

Package ‘ggalluvial’

September 2, 2019

Type Package

Title Alluvial Diagrams in 'ggplot2'

Version 0.10.0

Date 2019-09-02

Maintainer Jason Cory Brunson <cornelioid@gmail.com>

Description Alluvial diagrams use x-splines, sometimes augmented with stacked histograms, to visualize multi-dimensional or repeated-measures data with categorical or ordinal variables. They can be viewed as simplified and standardized Sankey diagrams; see Riehmman, Hanfler, and Froehlich (2005) <doi:10.1109/INFVIS.2005.1532152> and Rosvall and Bergstrom (2010) <doi:10.1371/journal.pone.0008694>. This package provides ggplot2 layers to produce alluvial diagrams from tidy data.

Depends R (>= 3.3.0), ggplot2 (>= 2.2.0)

Imports stats, dplyr (>= 0.7.0), tidyr (>= 0.7.0), lazyeval, rlang, tidyselect

Suggests grid, alluvial, testthat, knitr, babynames, sessioninfo, ggrepel, ggfittext (>= 0.6.0)

License GPL-3

URL <http://corybrunson.github.io/ggalluvial/>

BugReports <https://github.com/corybrunson/ggalluvial/issues>

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Jason Cory Brunson [aut, cre]

Repository CRAN

Date/Publication 2019-09-02 18:40:02 UTC

R topics documented:

alluvial-data	2
geom_alluvium	5
geom_flow	7
geom_lode	9
geom_stratum	12
lode-guidance-functions	14
majors	15
self-adjoin	15
stat_alluvium	16
stat_flow	20
stat_stratum	23
vaccinations	26

Index	27
--------------	-----------

alluvial-data	<i>Check for alluvial structure and convert between alluvial formats</i>
---------------	--

Description

Alluvial diagrams consist of multiple horizontally-distributed columns (axes) representing factor variables, vertical divisions (strata) of these axes representing these variables' values; and splines (alluvial flows) connecting vertical subdivisions (lodes) within strata of adjacent axes representing subsets or amounts of observations that take the corresponding values of the corresponding variables. This function checks a data frame for either of two types of alluvial structure:

Usage

```
is_lodes_form(data, key, value, id, weight = NULL, logical = TRUE,
              silent = FALSE)
```

```
is_alluvia_form(data, ..., axes = NULL, weight = NULL,
                logical = TRUE, silent = FALSE)
```

```
to_lodes_form(data, ..., axes = NULL, key = "x", value = "stratum",
              id = "alluvium", diffuse = FALSE, discern = FALSE)
```

```
to_alluvia_form(data, key, value, id, distill = FALSE)
```

Arguments

data	A data frame.
key, value, id	In <code>to_lodes_form</code> , handled as in <code>tidyr::gather()</code> and used to name the new axis (key), stratum (value), and alluvium (identifying) variables. In <code>to_alluvia_form</code> , handled as in <code>tidyr::spread()</code> and used to identify the fields of data to be used as the axis (key), stratum (value), and alluvium (identifying) variables.

weight	Optional field of data, handled using <code>rlang::enquo()</code> , to be used as heights or depths of the alluvia or lodes.
logical	Deprecated. Whether to return a logical value or a character string indicating the type of alluvial structure ("none", "lodes", or "alluvia").
silent	Whether to print messages.
...	Used in <code>is_alluvia_form</code> and <code>to_lodes_form</code> as in <code>dplyr::select()</code> to determine axis variables, as an alternative to <code>axes</code> . Ignored when <code>axes</code> is provided.
axes	In <code>*_alluvia_form</code> , handled as in <code>dplyr::select()</code> and used to identify the field(s) of data to be used as axes.
diffuse	Fields of data, handled using <code>tidyselect::vars_select()</code> , to merge into the reshaped data by id. They must be a subset of the axis variables. Alternatively, a logical value indicating whether to merge all (TRUE) or none (FALSE) of the axis variables.
discern	Logical value indicating whether to suffix values of the variables used as axes that appear at more than one variable in order to distinguish their factor levels. This forces the levels of the combined factor variable value to be in the order of the axes.
distill	A logical value indicating whether to include variables, other than those passed to <code>key</code> and <code>value</code> , that vary within values of <code>id</code> . Alternatively, a function (or its name) to be used to distill each such variable to a single value. In addition to existing functions, <code>distill</code> accepts the character values "first" (used if <code>distill</code> is TRUE), "last", and "most" (which returns the modal value).

Details

- One row per **lode**, wherein each row encodes a subset or amount of observations having a specific profile of axis values, a `key` field encodes the axis, a `value` field encodes the value within each axis, and a `id` column identifies multiple lodes corresponding to the same subset or amount of observations. `is_lodes_form` tests for this structure.
- One row per **alluvium**, wherein each row encodes a subset or amount of observations having a specific profile of axis values and a set `axes` of fields encodes its values at each axis variable. `is_alluvia_form` tests for this structure.

`to_lodes_form` takes a data frame with several designated variables to be used as axes in an alluvial diagram, and reshapes the data frame so that the axis variable names constitute a new factor variable and their values comprise another. Other variables' values will be repeated, and a row-grouping variable can be introduced. This function invokes `tidyr::gather()`.

`to_alluvia_form` takes a data frame with axis and axis value variables to be used in an alluvial diagram, and reshapes the data frame so that the axes constitute separate variables whose values are given by the value variable. This function invokes `tidyr::spread()`.

See Also

Other alluvial data manipulation: [self-adjoin](#)

Examples

```

# Titanic data in alluvia format
titanic_alluvia <- as.data.frame(Titanic)
head(titanic_alluvia)
is_alluvia_form(titanic_alluvia,
                weight = "Freq")
# Titanic data in lodes format
titanic_lodes <- to_lodes_form(titanic_alluvia,
                              key = "x", value = "stratum", id = "alluvium",
                              axes = 1:4)

head(titanic_lodes)
is_lodes_form(titanic_lodes,
              key = "x", value = "stratum", id = "alluvium",
              weight = "Freq")
# again in lodes format, this time diffusing the `Class` variable
titanic_lodes2 <- to_lodes_form(titanic_alluvia,
                               key = variable, value = value,
                               id = passenger,
                               1:3, diffuse = Class)

head(titanic_lodes2)
is_lodes_form(titanic_lodes2,
              key = variable, value = value, id = passenger,
              weight = Freq)

# curriculum data in lodes format
data(majors)
head(majors)
is_lodes_form(majors,
              key = "semester", value = "curriculum", id = "student")
# curriculum data in alluvia format
majors_alluvia <- to_alluvia_form(majors,
                                  key = "semester", value = "curriculum",
                                  id = "student")

head(majors_alluvia)
is_alluvia_form(majors_alluvia, tidyselect::starts_with("CURR"))

# distill variables that vary within `id` values
set.seed(1)
majors$hypo_grade <- LETTERS[sample(5, size = nrow(majors), replace = TRUE)]
majors_alluvia2 <- to_alluvia_form(majors,
                                  key = "semester", value = "curriculum",
                                  id = "student",
                                  distill = "most")

head(majors_alluvia2)

# options to distinguish strata at different axes
gg <- ggplot(majors_alluvia,
             aes(axis1 = CURR1, axis2 = CURR7, axis3 = CURR13))
gg +
  geom_alluvium(aes(fill = as.factor(student)), width = 2/5, discern = TRUE) +
  geom_stratum(width = 2/5, discern = TRUE) +
  geom_text(stat = "stratum", discern = TRUE, label.strata = TRUE)

```

```
gg +
  geom_alluvium(aes(fill = as.factor(student)), width = 2/5, discern = FALSE) +
  geom_stratum(width = 2/5, discern = FALSE) +
  geom_text(stat = "stratum", discern = FALSE, label.strata = TRUE)
# warning when inappropriate
ggplot(majors[majors$semester %in% paste0("CURR", c(1, 7, 13)), ],
  aes(x = semester, stratum = curriculum, alluvium = student,
    label = curriculum)) +
  geom_alluvium(aes(fill = as.factor(student)), width = 2/5, discern = TRUE) +
  geom_stratum(width = 2/5, discern = TRUE) +
  geom_text(stat = "stratum", discern = TRUE)
```

geom_alluvium	<i>Alluvia across strata</i>
---------------	------------------------------

Description

geom_alluvium receives a dataset of the horizontal (x) and vertical (y, ymin, ymax) positions of the **lodes** of an alluvial diagram, the intersections of the alluvia with the strata. It plots both the lodes themselves, using [geom_lode\(\)](#), and the flows between them, using [geom_flow\(\)](#).

Usage

```
geom_alluvium(mapping = NULL, data = NULL, stat = "alluvium",
  position = "identity", width = 1/3, knot.pos = 1/6,
  na.rm = FALSE, show.legend = NA, inherit.aes = TRUE, ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to ggplot() . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
stat	The statistical transformation to use on the data; override the default.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
width	Numeric; the width of each stratum, as a proportion of the distance between axes. Defaults to 1/3.

knot.pos	The horizontal distance between a stratum (width/2 from its axis) and the knot of the x-spline, as a proportion of the separation between strata. Defaults to 1/6.
na.rm	Logical: if FALSE, the default, NA lodes are not included; if TRUE, NA lodes constitute a separate category, plotted in grey (regardless of the color scheme).
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .
...	Additional arguments passed to <code>ggplot2::layer()</code> .

Aesthetics

`geom_alluvium`, `geom_flow`, `geom_lode`, and `geom_stratum` understand the following aesthetics (required aesthetics are in bold):

- x
- y
- ymin
- ymax
- alpha
- colour
- fill
- linetype
- size
- group

group is used internally; arguments are ignored.

Defunct parameters

The previously deprecated parameters `axis_width` and `ribbon_bend` are now defunct. Use `width` and `knot.pos` instead.

See Also

`ggplot2::layer()` for additional arguments and `stat_alluvium()` and `stat_flow()` for the corresponding stats.

Other alluvial geom layers: `geom_flow`, `geom_lode`, `geom_stratum`

Examples

```
# basic
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex, axis3 = Age,
           fill = Survived)) +
  geom_alluvium() +
  scale_x_discrete(limits = c("Class", "Sex", "Age"))

gg <- ggplot(alluvial::Refugees,
             aes(y = refugees, x = year, alluvium = country))
# time series bump chart
gg + geom_alluvium(aes(fill = country, colour = country),
                  width = 1/4, alpha = 2/3, decreasing = FALSE)
# time series line plot of refugees data, sorted by country
gg + geom_alluvium(aes(fill = country, colour = country),
                  decreasing = NA, width = 0, knot.pos = 0)
```

geom_flow

*Flows between lodes or strata***Description**

geom_flow receives a dataset of the horizontal (x) and vertical (y, ymin, ymax) positions of the **lodes** of an alluvial diagram, the intersections of the alluvia with the strata. It reconfigures these into alluvial segments connecting pairs of corresponding lodes in adjacent strata and plots filled x-splines between each such pair, using a provided knot position parameter knot.pos, and filled rectangles at either end, using a provided width.

Usage

```
geom_flow(mapping = NULL, data = NULL, stat = "flow",
          position = "identity", width = 1/3, knot.pos = 1/6,
          aes.flow = "forward", na.rm = FALSE, show.legend = NA,
          inherit.aes = TRUE, ...)
```

Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> or <code>aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created.

A function will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A function can be created from a formula (e.g. `~ head(.x, 10)`).

<code>stat</code>	The statistical transformation to use on the data; override the default.
<code>position</code>	Position adjustment, either as a string, or the result of a call to a position adjustment function.
<code>width</code>	Numeric; the width of each stratum, as a proportion of the distance between axes. Defaults to 1/3.
<code>knot.pos</code>	The horizontal distance between a stratum (<code>width/2</code> from its axis) and the knot of the x-spline, as a proportion of the separation between strata. Defaults to 1/6.
<code>aes.flow</code>	Character; how inter-lode flows assume aesthetics from lodes. Options are "forward" and "backward".
<code>na.rm</code>	Logical: if FALSE, the default, NA lodes are not included; if TRUE, NA lodes constitute a separate category, plotted in grey (regardless of the color scheme).
<code>show.legend</code>	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
<code>inherit.aes</code>	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .
<code>...</code>	Additional arguments passed to <code>ggplot2::layer()</code> .

Aesthetics

`geom_alluvium`, `geom_flow`, `geom_lode`, and `geom_stratum` understand the following aesthetics (required aesthetics are in bold):

- `x`
- `y`
- `ymin`
- `ymax`
- `alpha`
- `colour`
- `fill`
- `linetype`
- `size`
- `group`

`group` is used internally; arguments are ignored.

Defunct parameters

The previously deprecated parameters `axis_width` and `ribbon_bend` are now defunct. Use `width` and `knot.pos` instead.

See Also

`ggplot2::layer()` for additional arguments and `stat_alluvium()` and `stat_flow()` for the corresponding stats.

Other alluvial geom layers: `geom_alluvium`, `geom_lode`, `geom_stratum`

Examples

```
# use of strata and labels
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex, axis3 = Age)) +
  geom_flow() +
  scale_x_discrete(limits = c("Class", "Sex", "Age")) +
  geom_stratum() + geom_text(stat = "stratum", label.strata = TRUE) +
  ggtitle("Alluvial diagram of Titanic passenger demographic data")

# use of facets
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex)) +
  geom_flow(aes(fill = Age), width = .4) +
  geom_stratum(width = .4) +
  geom_text(stat = "stratum", label.strata = TRUE, size = 3) +
  scale_x_discrete(limits = c("Class", "Sex")) +
  facet_wrap(~ Survived, scales = "fixed")

# time series alluvia of WorldPhones data
wph <- as.data.frame(as.table(WorldPhones))
names(wph) <- c("Year", "Region", "Telephones")
ggplot(wph,
       aes(x = Year, alluvium = Region, y = Telephones)) +
  geom_flow(aes(fill = Region, colour = Region), width = 0)

# rightward flow aesthetics for vaccine survey data
data(vaccinations)
levels(vaccinations$response) <- rev(levels(vaccinations$response))
ggplot(vaccinations,
       aes(x = survey, stratum = response, alluvium = subject,
           y = freq, fill = response, label = round(a, 3))) +
  geom_lode() + geom_flow() +
  geom_stratum(alpha = 0) +
  geom_text(stat = "stratum")
```

Description

`geom_alluvium` receives a dataset of the horizontal (x) and vertical (y , $ymin$, $ymax$) positions of the **lodes** of an alluvial diagram, the intersections of the alluvia with the strata. It plots rectangles for these lodes of a provided width.

Usage

```
geom_lode(mapping = NULL, data = NULL, stat = "alluvium",
          position = "identity", width = 1/3, na.rm = FALSE,
          show.legend = NA, inherit.aes = TRUE, ...)
```

Arguments

<code>mapping</code>	Set of aesthetic mappings created by <code>aes()</code> or <code>aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply <code>mapping</code> if there is no plot mapping.
<code>data</code>	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
<code>stat</code>	The statistical transformation to use on the data; override the default.
<code>position</code>	Position adjustment, either as a string, or the result of a call to a position adjustment function.
<code>width</code>	Numeric; the width of each stratum, as a proportion of the distance between axes. Defaults to <code>1/3</code> .
<code>na.rm</code>	Logical: if <code>FALSE</code> , the default, NA lodes are not included; if <code>TRUE</code> , NA lodes constitute a separate category, plotted in grey (regardless of the color scheme).
<code>show.legend</code>	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
<code>inherit.aes</code>	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .
<code>...</code>	Additional arguments passed to <code>ggplot2::layer()</code> .

Aesthetics

`geom_alluvium`, `geom_flow`, `geom_lode`, and `geom_stratum` understand the following aesthetics (required aesthetics are in bold):

- **x**

- y
- ymin
- ymax
- alpha
- colour
- fill
- linetype
- size
- group

group is used internally; arguments are ignored.

Defunct parameters

The previously deprecated parameters `axis_width` and `ribbon_bend` are now defunct. Use `width` and `knot.pos` instead.

See Also

[ggplot2::layer\(\)](#) for additional arguments and [stat_alluvium\(\)](#) and [stat_stratum\(\)](#) for the corresponding stats.

Other alluvial geom layers: [geom_alluvium](#), [geom_flow](#), [geom_stratum](#)

Examples

```
# one axis
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis = Class)) +
  geom_lode(aes(fill = Class, alpha = Survived)) +
  scale_x_discrete(limits = c("Class")) +
  scale_alpha_manual(values = c(.25, .75))

gg <- ggplot(as.data.frame(Titanic),
            aes(y = Freq,
                axis1 = Class, axis2 = Sex, axis3 = Age,
                fill = Survived))
# alluvia and lodes
gg + geom_alluvium() + geom_lode()
# lodes as strata
gg + geom_alluvium() +
  geom_stratum(stat = "alluvium")
```

geom_stratum

*Strata at axes***Description**

geom_stratum receives a dataset of the horizontal (x) and vertical (y, ymin, ymax) positions of the strata of an alluvial diagram. It plots rectangles for these strata of a provided width.

Usage

```
geom_stratum(mapping = NULL, data = NULL, stat = "stratum",
             position = "identity", show.legend = NA, inherit.aes = TRUE,
             width = 1/3, na.rm = FALSE, ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to ggplot() . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
stat	The statistical transformation to use on the data; override the default.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders() .
width	Numeric; the width of each stratum, as a proportion of the distance between axes. Defaults to 1/3.
na.rm	Logical: if <code>FALSE</code> , the default, <code>NA</code> lodes are not included; if <code>TRUE</code> , <code>NA</code> lodes constitute a separate category, plotted in grey (regardless of the color scheme).
...	Additional arguments passed to ggplot2::layer() .

Aesthetics

geom_alluvium, geom_flow, geom_lode, and geom_stratum understand the following aesthetics (required aesthetics are in bold):

- x
- y
- ymin
- ymax
- alpha
- colour
- fill
- linetype
- size
- group

group is used internally; arguments are ignored.

Defunct parameters

The previously deprecated parameters `axis_width` and `ribbon_bend` are now defunct. Use `width` and `knot.pos` instead.

See Also

[ggplot2::layer\(\)](#) for additional arguments and [stat_stratum\(\)](#) for the corresponding stat.

Other alluvial geom layers: [geom_alluvium](#), [geom_flow](#), [geom_lode](#)

Examples

```
# full axis width
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex, axis3 = Age, axis4 = Survived)) +
  geom_stratum(width = 1) + geom_text(stat = "stratum", label.strata = TRUE) +
  scale_x_discrete(limits = c("Class", "Sex", "Age", "Survived"))

# use of facets
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex)) +
  geom_flow(aes(fill = Survived)) +
  geom_stratum() + geom_text(stat = "stratum", label.strata = TRUE) +
  scale_x_discrete(limits = c("Class", "Sex")) +
  facet_wrap(~ Age, scales = "free_y")
```

lode-guidance-functions

Lode guidance functions

Description

These functions control the order of lodes within strata in an alluvial diagram. They are invoked by `stat_alluvium()` and can be passed to the `lode.guidance` parameter.

Usage

`lode_zigzag(n, i)`

`lode_forward(n, i)`

`lode_rightward(n, i)`

`lode_backward(n, i)`

`lode_leftward(n, i)`

`lode_frontback(n, i)`

`lode_rightleft(n, i)`

`lode_backfront(n, i)`

`lode_leftright(n, i)`

Arguments

<code>n</code>	Numeric, a positive integer
<code>i</code>	Numeric, a positive integer at most <code>n</code>

Details

Each function orders the numbers 1 through `n`, starting at index `i`. The choice of function made in `stat_alluvium()` determines the order in which the other axes contribute to the sorting of lodes within each index axis. After starting at `i`, the functions order the remaining axes as follows:

- `zigzag`: Zigzag outward from `i`
- `forward`: Increasing order (alias `rightward`)
- `backward`: Decreasing order (alias `leftward`)
- `frontback`: Proceed forward from `i` to `n`, then backward to 1 (alias `rightleft`)
- `backfront`: Proceed backward from `i` to 1, then forward to `n` (alias `leftright`)

majors	<i>Student curricula across semesters</i>
--------	---

Description

This dataset, kindly contributed by Dario Bonaretti, follows the curricula of 10 students across 8 academic semesters.

Format

An alluvial data frame in lodes form.

self-adjoin	<i>Adjoin a dataset to itself</i>
-------------	-----------------------------------

Description

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

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::join functions.
link	Character vector of variables to adjoin. Will be replaced by pairs of variables suffixed by <code>suffix</code> .
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 <code>by</code> or <code>link</code> .
suffix	Suffixes to add to the adjoined link variables; passed to dplyr::join functions.

Details

`self_adjoin` invokes [dplyr::join](#) 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.

See Also

Other alluvial data manipulation: [alluvial-data](#)

Examples

```
# self-adjoin `majors` data
data(majors)
major_changes <- self_adjoin(majors, key = semester,
                             by = "student", link = c("semester", "curriculum"))
major_changes$change <- major_changes$curriculum.x == major_changes$curriculum.y
head(major_changes)

# self-adjoin `vaccinations` data
data(vaccinations)
vaccination_steps <- self_adjoin(vaccinations, key = survey, by = "subject",
                                 link = c("survey", "response"),
                                 keep.x = c("freq", "a"))

head(vaccination_steps)
vaccination_steps <- self_adjoin(vaccinations, key = survey, by = "subject",
                                 link = c("survey", "response"),
                                 keep.x = c("freq", "a"), keep.y = "a")

head(vaccination_steps)
```

stat_alluvium

Alluvial positions

Description

Given a dataset with alluvial structure, `stat_alluvium` calculates the centroids (x and y) of the **lodes**, the intersections of the alluvia with the strata, together with their weights (heights; `ymin` and `ymax`). It leverages the group aesthetic for plotting purposes (for now).

Usage

```
stat_alluvium(mapping = NULL, data = NULL, geom = "alluvium",
              position = "identity", decreasing = NA, reverse = TRUE,
              discern = FALSE, aes.bind = FALSE, aggregate.y = FALSE,
              aggregate.wts = NULL, lode.guidance = "zigzag",
              lode.ordering = NULL, na.rm = FALSE, show.legend = NA,
              inherit.aes = TRUE, ...)
```

Arguments

`mapping` Set of aesthetic mappings created by `aes()` or `aes_()`. If specified and `inherit.aes` = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

data	<p>The data to be displayed in this layer. There are three options:</p> <p>If NULL, the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code>.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p>
geom	The geometric object to use display the data; override the default.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
decreasing	Logical; whether to arrange the strata at each axis in the order of the variable values (NA, the default), in ascending order of total weight (largest on top, FALSE), or in descending order of total weight (largest on bottom, TRUE).
reverse	Logical; if <code>decreasing</code> is NA, whether to arrange the strata at each axis in the reverse order of the variable values, so that they match the order of the values in the legend. Ignored if <code>decreasing</code> is not NA. Defaults to TRUE.
discern	Passed to <code>to_lodes_form()</code> if data is in alluvia format.
aes.bind	Whether to prioritize aesthetics before axes (other than the index axis) when ordering the lodes within each stratum. Defaults to FALSE.
aggregate.y	Whether to aggregate weights across otherwise equivalent rows before computing lode and flow positions. Set to TRUE to group observations into cohorts. Warning: This is currently an expensive operation.
aggregate.wts	Defunct alias for <code>aggregate.y</code> .
lode.guidance	The function to prioritize the axis variables for ordering the lodes within each stratum, or else a character string identifying the function. Character options are "zigzag", "frontback", "backfront", "forward", and "backward" (see lode-guidance-functions).
lode.ordering	A list (of length the number of axes) of integer vectors (each of length the number of rows of data) or NULL entries (indicating no imposed ordering), or else a numeric matrix of corresponding dimensions, giving the preferred ordering of alluvia at each axis. This will be used to order the lodes within each stratum by sorting the lodes first by stratum and then by the provided vectors.
na.rm	Logical: if FALSE, the default, NA lodes are not included; if TRUE, NA lodes constitute a separate category, plotted in grey (regardless of the color scheme).
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .
...	Additional arguments passed to <code>ggplot2::layer()</code> .

Aesthetics

stat_alluvium, stat_flow, and stat_stratum require one of two sets of aesthetics:

- x and at least one of alluvium and stratum
- any number of axis[0-9]* (axis1, axis2, etc.)

Use x, alluvium, and/or stratum for data in lodes format and axis[0-9]* for data in alluvia format (see [alluvial-data](#)). Arguments to parameters inconsistent with the format will be ignored. Additionally, each stat_* layer accepts the following optional aesthetics:

- y
- group

y controls the heights of the alluvia and may be aggregated across equivalent observations. group is used internally; arguments are ignored. Finally, stat_stratum accepts the following optional aesthetic:

- label

label is used to label the strata and must take a unique value across the observations within each stratum. These and any other aesthetics are aggregated as follows: Numeric aesthetics, including y, are summed. Character and factor aesthetics, including label, are assigned to strata provided they take unique values across the observations within each stratum (otherwise NA is assigned).

See Also

[ggplot2::layer\(\)](#) for additional arguments and [geom_alluvium\(\)](#), [geom_lode\(\)](#), and [geom_flow\(\)](#) for the corresponding geoms.

Other alluvial stat layers: [stat_flow](#), [stat_stratum](#)

Examples

```
# illustrate positioning
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex, axis3 = Age,
           color = Survived)) +
  stat_stratum(geom = "errorbar") +
  geom_line(stat = "alluvium") +
  stat_alluvium(geom = "pointrange") +
  geom_text(stat = "stratum", label.strata = TRUE) +
  scale_x_discrete(limits = c("Class", "Sex", "Age"))

gg <- ggplot(as.data.frame(Titanic),
             aes(y = Freq,
                 axis1 = Class, axis2 = Sex, axis3 = Age)) +
  geom_stratum() + geom_text(stat = "stratum", label.strata = TRUE) +
  scale_x_discrete(limits = c("Class", "Sex", "Age"))
# use of lode controls
gg + geom_flow(aes(fill = Survived, alpha = Sex), stat = "alluvium",
              lode.guidance = "forward")
```

```

# prioritize aesthetic binding
gg + geom_flow(aes(fill = Survived, alpha = Sex), stat = "alluvium",
               aes.bind = TRUE, lode.guidance = "forward")
# use of lode ordering
lode_ord <- replicate(n = 3, expr = sample(x = 32), simplify = FALSE)
print(lode_ord)
gg + geom_flow(aes(fill = Survived, alpha = Sex), stat = "alluvium",
               lode.ordering = lode_ord)
# fixed lode ordering across axes
gg + geom_flow(aes(fill = Survived, alpha = Sex), stat = "alluvium",
               lode.ordering = lode_ord[[1]])
# use of custom lode guidance function
lode_custom <- function(n, i) {
  stopifnot(n == 3)
  switch(
    i,
    `1` = 1:3,
    `2` = c(2, 3, 1),
    `3` = 3:1
  )
}
gg + geom_flow(aes(fill = Survived, alpha = Sex), stat = "alluvium",
               aes.bind = TRUE, lode.guidance = lode_custom)
gg + geom_flow(aes(fill = Survived, alpha = Sex), stat = "alluvium",
               aes.bind = TRUE, lode.guidance = "custom")

data(majors)
# omit missing lodes and incident flows
ggplot(majors,
       aes(x = semester, stratum = curriculum, alluvium = student)) +
  geom_alluvium(fill = "darkgrey", na.rm = TRUE) +
  geom_stratum(aes(fill = curriculum), color = NA, na.rm = TRUE) +
  theme_bw()
# reverse the vertical axis (requires an explicit `y` aesthetic)
ggplot(majors,
       aes(x = semester, stratum = curriculum, alluvium = student, y = 1)) +
  geom_alluvium(fill = "darkgrey", na.rm = TRUE) +
  geom_stratum(aes(fill = curriculum), color = NA, na.rm = TRUE) +
  theme_bw() +
  scale_y_reverse()

gg <- ggplot(majors,
            aes(x = semester, stratum = curriculum, alluvium = student,
                fill = curriculum)) +
  geom_stratum()
# diagram with outlined alluvia and forward-colored flows
gg + geom_flow(stat = "alluvium", lode.guidance = "frontback",
               color = "black")
# same diagram with students are aggregated into cohorts
gg + geom_flow(stat = "alluvium", lode.guidance = "frontback",
               color = "black", aggregate.y = TRUE)

# irregular spacing between axes of a continuous variable

```

```

data(Refugees, package = "alluvial")
refugees_sub <- subset(Refugees, year %in% c(2003, 2005, 2010, 2013))
ggplot(data = refugees_sub,
       aes(x = year, y = refugees, alluvium = country)) +
  geom_alluvium(aes(fill = country),
               alpha = .75, decreasing = FALSE, knot.pos = 1) +
  geom_stratum(aes(stratum = country), decreasing = FALSE, width = 1/2) +
  theme_bw() +
  scale_fill_brewer(type = "qual", palette = "Set3")

## Not run:
data(babynames, package = "babynames")
# a discontinuous alluvium
bn <- dplyr::filter(babynames,
                   prop >= .01 & sex == "F" &
                   year > 1962 & year < 1968)
ggplot(data = bn,
       aes(x = year, alluvium = name, y = prop)) +
  geom_alluvium(aes(fill = name, color = name == "Tammy"),
               decreasing = TRUE, show.legend = FALSE) +
  scale_color_manual(values = c("#00000000", "#000000"))
# filling in missing zeros
bn2 <- merge(bn,
             expand.grid(year = unique(bn$year), name = unique(bn$name)),
             all = TRUE)
bn2$prop[is.na(bn2$prop)] <- 0
ggplot(data = bn2,
       aes(x = year, alluvium = name, y = prop)) +
  geom_alluvium(aes(fill = name, color = name == "Tammy"),
               decreasing = TRUE, show.legend = FALSE) +
  scale_color_manual(values = c("#00000000", "#000000"))

## End(Not run)

```

stat_flow

Flow positions

Description

Given a dataset with alluvial structure, `stat_flow` calculates the centroids (x and y) and weights (heights; `ymin` and `ymax`) of alluvial flows between each pair of adjacent axes.

Usage

```

stat_flow(mapping = NULL, data = NULL, geom = "flow",
          position = "identity", decreasing = NA, reverse = TRUE,
          discern = FALSE, aes.bind = FALSE, na.rm = FALSE,
          show.legend = NA, inherit.aes = TRUE, ...)

```

Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> or <code>aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
geom	The geometric object to use display the data; override the default.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
decreasing	Logical; whether to arrange the strata at each axis in the order of the variable values (<code>NA</code> , the default), in ascending order of total weight (largest on top, <code>FALSE</code>), or in descending order of total weight (largest on bottom, <code>TRUE</code>).
reverse	Logical; if <code>decreasing</code> is <code>NA</code> , whether to arrange the strata at each axis in the reverse order of the variable values, so that they match the order of the values in the legend. Ignored if <code>decreasing</code> is not <code>NA</code> . Defaults to <code>TRUE</code> .
discern	Passed to <code>to_lodes_form()</code> if data is in alluvia format.
aes.bind	Whether to prioritize aesthetics before axes (other than the index axis) when ordering the lodes within each stratum. Defaults to <code>FALSE</code> .
na.rm	Logical: if <code>FALSE</code> , the default, <code>NA</code> lodes are not included; if <code>TRUE</code> , <code>NA</code> lodes constitute a separate category, plotted in grey (regardless of the color scheme).
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .
...	Additional arguments passed to <code>ggplot2::layer()</code> .

Aesthetics

`stat_alluvium`, `stat_flow`, and `stat_stratum` require one of two sets of aesthetics:

- `x` and at least one of `alluvium` and `stratum`
- any number of `axis[0-9]*` (`axis1`, `axis2`, etc.)

Use `x`, `alluvium`, and/or `stratum` for data in lodes format and `axis[0-9]*` for data in alluvia format (see [alluvial-data](#)). Arguments to parameters inconsistent with the format will be ignored. Additionally, each `stat_*` layer accepts the following optional aesthetics:

- y
- group

y controls the heights of the alluvia and may be aggregated across equivalent observations. group is used internally; arguments are ignored. Finally, stat_stratum accepts the following optional aesthetic:

- label

label is used to label the strata and must take a unique value across the observations within each stratum. These and any other aesthetics are aggregated as follows: Numeric aesthetics, including y, are summed. Character and factor aesthetics, including label, are assigned to strata provided they take unique values across the observations within each stratum (otherwise NA is assigned).

See Also

[ggplot2::layer\(\)](#) for additional arguments and [geom_alluvium\(\)](#) and [geom_flow\(\)](#) for the corresponding geoms.

Other alluvial stat layers: [stat_alluvium](#), [stat_stratum](#)

Examples

```
# illustrate positioning
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex, axis3 = Age,
           color = Survived)) +
  stat_stratum(geom = "errorbar") +
  geom_line(stat = "flow") +
  stat_flow(geom = "pointrange") +
  geom_text(stat = "stratum", label.strata = TRUE) +
  scale_x_discrete(limits = c("Class", "Sex", "Age"))

# use of lode controls
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex, axis3 = Age)) +
  geom_flow(aes(fill = Survived), aes.bind = TRUE, reverse = FALSE) +
  geom_stratum(reverse = FALSE) +
  geom_text(stat = "stratum", label.strata = TRUE, reverse = FALSE) +
  scale_x_discrete(limits = c("Class", "Sex", "Age"))

data(vaccinations)
gg <- ggplot(vaccinations,
            aes(x = survey, stratum = response, alluvium = subject,
               y = freq, fill = response)) +
  geom_stratum(alpha = .5) +
  geom_text(aes(label = response), stat = "stratum")
# rightward alluvial aesthetics for vaccine survey data
gg + geom_flow(stat = "alluvium", lode.guidance = "forward")
# memoryless flows for vaccine survey data
gg + geom_flow()
```

```

# aesthetics that vary between and within strata
data(vaccinations)
vaccinations$subgroup <- LETTERS[1:2][rbinom(
  n = length(unique(vaccinations$subject)), size = 1, prob = .5
) + 1][vaccinations$subject]
ggplot(vaccinations,
  aes(x = survey, stratum = response, alluvium = subject,
      y = freq, fill = response, label = response)) +
  geom_flow(aes(alpha = subgroup)) +
  scale_alpha_discrete(range = c(1/3, 2/3)) +
  geom_stratum(alpha = .5) +
  geom_text(stat = "stratum")
# can even set aesthetics that vary both ways
ggplot(vaccinations,
  aes(x = survey, stratum = response, alluvium = subject,
      y = freq, label = response)) +
  geom_flow(aes(fill = interaction(response, subgroup)), aes.bind = TRUE) +
  scale_alpha_discrete(range = c(1/3, 2/3)) +
  geom_stratum(alpha = .5) +
  geom_text(stat = "stratum")

data(majors)
# omit missing lodes and incident flows
ggplot(majors,
  aes(x = semester, stratum = curriculum, alluvium = student)) +
  geom_flow(fill = "darkgrey", na.rm = TRUE) +
  geom_stratum(aes(fill = curriculum), color = NA, na.rm = TRUE) +
  theme_bw()

```

stat_stratum

Stratum positions

Description

Given a dataset with alluvial structure, `stat_stratum` calculates the centroids of the strata at each axis, together with their weights (heights).

Usage

```

stat_stratum(mapping = NULL, data = NULL, geom = "stratum",
  position = "identity", decreasing = NA, reverse = TRUE,
  discern = FALSE, label.strata = FALSE, min.height = NULL,
  max.height = NULL, show.legend = NA, inherit.aes = TRUE,
  na.rm = FALSE, ...)

```

Arguments

`mapping` Set of aesthetic mappings created by `aes()` or `aes_()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

<code>data</code>	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code>.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p>
<code>geom</code>	The geometric object to use display the data; override the default.
<code>position</code>	Position adjustment, either as a string, or the result of a call to a position adjustment function.
<code>decreasing</code>	Logical; whether to arrange the strata at each axis in the order of the variable values (<code>NA</code> , the default), in ascending order of total weight (largest on top, <code>FALSE</code>), or in descending order of total weight (largest on bottom, <code>TRUE</code>).
<code>reverse</code>	Logical; if <code>decreasing</code> is <code>NA</code> , whether to arrange the strata at each axis in the reverse order of the variable values, so that they match the order of the values in the legend. Ignored if <code>decreasing</code> is not <code>NA</code> . Defaults to <code>TRUE</code> .
<code>discern</code>	Passed to <code>to_lodes_form()</code> if data is in alluvia format.
<code>label.strata</code>	Logical; whether to assign the values of the axis variables to the strata. Defaults to <code>FALSE</code> , and requires that no <code>label</code> aesthetic is assigned.
<code>min.height, max.height</code>	Numeric; bounds on the heights (weights) of the strata to be rendered. Use these bounds to exclude strata outside a certain range, for example when labeling strata using <code>ggplot2::geom_text()</code> .
<code>show.legend</code>	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
<code>inherit.aes</code>	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .
<code>na.rm</code>	Logical: if <code>FALSE</code> , the default, <code>NA</code> lodes are not included; if <code>TRUE</code> , <code>NA</code> lodes constitute a separate category, plotted in grey (regardless of the color scheme).
<code>...</code>	Additional arguments passed to <code>ggplot2::layer()</code> .

Aesthetics

`stat_alluvium`, `stat_flow`, and `stat_stratum` require one of two sets of aesthetics:

- `x` and at least one of `alluvium` and `stratum`
- any number of `axis[0-9]*` (`axis1`, `axis2`, etc.)

Use `x`, `alluvium`, and/or `stratum` for data in lodes format and `axis[0-9]*` for data in alluvia format (see [alluvial-data](#)). Arguments to parameters inconsistent with the format will be ignored. Additionally, each `stat_*` layer accepts the following optional aesthetics:

- y
- group

y controls the heights of the alluvia and may be aggregated across equivalent observations. group is used internally; arguments are ignored. Finally, stat_stratum accepts the following optional aesthetic:

- label

label is used to label the strata and must take a unique value across the observations within each stratum. These and any other aesthetics are aggregated as follows: Numeric aesthetics, including y, are summed. Character and factor aesthetics, including label, are assigned to strata provided they take unique values across the observations within each stratum (otherwise NA is assigned).

See Also

[ggplot2::layer\(\)](#) for additional arguments and [geom_stratum\(\)](#) for the corresponding geom.

Other alluvial stat layers: [stat_alluvium](#), [stat_flow](#)

Examples

```
# only `stratum` assignment is necessary to generate strata
data(vaccinations)
ggplot(vaccinations,
  aes(y = freq,
      x = survey, stratum = response,
      fill = response)) +
  stat_stratum(width = .5)

# lode data: positioning with weight labels
ggplot(vaccinations,
  aes(y = freq,
      x = survey, stratum = response, alluvium = subject,
      label = freq)) +
  stat_stratum(geom = "errorbar") +
  geom_text(stat = "stratum")
# lode data: positioning with stratum labels
ggplot(vaccinations,
  aes(y = freq,
      x = survey, stratum = response, alluvium = subject,
      label = response)) +
  stat_stratum(geom = "errorbar") +
  geom_text(stat = "stratum")

# alluvium data: positioning with weight labels
ggplot(as.data.frame(Titanic),
  aes(y = Freq,
      axis1 = Class, axis2 = Sex, axis3 = Age, axis4 = Survived,
      label = Freq)) +
  geom_text(stat = "stratum") +
  stat_stratum(geom = "errorbar") +
  scale_x_discrete(limits = c("Class", "Sex", "Age", "Survived"))
```

```
# alluvium data: positioning with stratum labels
ggplot(as.data.frame(Titanic),
       aes(y = Freq,
           axis1 = Class, axis2 = Sex, axis3 = Age, axis4 = Survived)) +
  geom_text(stat = "stratum", label.strata = TRUE) +
  stat_stratum(geom = "errorbar") +
  scale_x_discrete(limits = c("Class", "Sex", "Age", "Survived"))

# omit labels for strata outside a weight range
ggplot(vaccinations,
       aes(y = freq,
           x = survey, stratum = response,
           fill = response, label = response)) +
  stat_stratum(width = .5) +
  geom_text(stat = "stratum", min.height = 100)
```

vaccinations

Influenza vaccination rates

Description

This is a sample from the RAND ALP surveys on influenza vaccination, kindly contributed by Raffaele Vardavas.

Format

An alluvial data frame in lodes form.

Source

RAND American Life Panel <https://alpdata.rand.org/>

Index

`aes()`, 5, 7, 10, 12, 16, 21, 23
`aes_()`, 5, 7, 10, 12, 16, 21, 23
alluvial-data, 2

`borders()`, 6, 8, 10, 12, 17, 21, 24

`dplyr:::join`, 15
`dplyr:::select()`, 3

`fortify()`, 5, 7, 10, 12, 17, 21, 24

`geom_alluvium`, 5, 9, 11, 13
`geom_alluvium()`, 18, 22
`geom_flow`, 6, 7, 11, 13
`geom_flow()`, 5, 15, 18, 22
`geom_lode`, 6, 9, 9, 13
`geom_lode()`, 5, 18
`geom_stratum`, 6, 9, 11, 12
`geom_stratum()`, 25
`ggplot()`, 5, 7, 10, 12, 17, 21, 24
`ggplot2::geom_text()`, 24
`ggplot2::layer()`, 6, 8–13, 17, 18, 21, 22, 24, 25

`is_alluvia_form` (alluvial-data), 2
`is_lodes_form` (alluvial-data), 2

lode-guidance-functions, 14
`lode_backfront`
 (lode-guidance-functions), 14
`lode_backward`
 (lode-guidance-functions), 14
`lode_forward` (lode-guidance-functions), 14
`lode_frontback`
 (lode-guidance-functions), 14
`lode_leftright`
 (lode-guidance-functions), 14
`lode_leftward`
 (lode-guidance-functions), 14
`lode_rightleft`
 (lode-guidance-functions), 14
`lode_rightward`
 (lode-guidance-functions), 14
`lode_zigzag` (lode-guidance-functions), 14

majors, 15

`rlang::enquo()`, 3

self-adjoin, 15
`self_adjoin` (self-adjoin), 15
`stat_alluvium`, 16, 22, 25
`stat_alluvium()`, 6, 9, 11, 14
`stat_flow`, 18, 20, 25
`stat_flow()`, 6, 9
`stat_stratum`, 18, 22, 23
`stat_stratum()`, 11, 13

`tidyr::gather()`, 2, 3
`tidyr::spread()`, 2, 3, 15
`tidyselect::vars_select()`, 3
`to_alluvia_form` (alluvial-data), 2
`to_lodes_form` (alluvial-data), 2
`to_lodes_form()`, 17, 21, 24

vaccinations, 26