

Package ‘ggalign’

September 15, 2024

Title Align Multiple 'ggplot' Objects

Version 0.0.3

Description A 'ggplot2' extension offers various tools for organizing and arranging plots. It is designed to consistently align a specific axis across multiple 'ggplot' objects, making it especially useful for plots requiring data order manipulation. A typical use case includes organizing combinations like a dendrogram and a heatmap.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends ggplot2

Imports cli, ggh4x, grid, gtable, methods, rlang (>= 1.1.0), stats, tibble, tidyr

ByteCompile true

URL <https://github.com/Yunuuuu/ggalign>,
<https://yunuuuu.github.io/ggalign/>

BugReports <https://github.com/Yunuuuu/ggalign/issues>

Suggests gridGraphics, knitr, patchwork, rmarkdown, scales, testthat (>= 3.0.0), vdiff

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Yun Peng [aut, cre, cph] (<<https://orcid.org/0000-0003-2801-3332>>)

Maintainer Yun Peng <yunyunp96@163.com>

Repository CRAN

Date/Publication 2024-09-15 20:30:10 UTC

Contents

align_dendro	2
align_gg	5
align_group	8
align_kmeans	9
align_panel	10
align_plots	12
align_reorder	13
area	15
dendrogram_data	16
free_align	17
geom_draw	19
ggalignGrob	22
ggalign_stat	22
hclust2	23
hmanno	24
inset	25
is.ggheatmap	26
is.ggstack	27
layout-add	27
layout-operator	28
layout_heatmap	29
layout_stack	31
patch.formula	31
patch.ggplot	32
patch.grob	32
patch.Heatmap	33
patch_titles	34
read_example	35
stack_active	35
wrap	37
Index	38

align_dendro

Reorder or Group layout based on hierarchical clustering

Description

Reorder or Group layout based on hierarchical clustering

Usage

```
align_dendro(
  mapping = aes(),
  ...,
  distance = "euclidean",
  method = "complete",
  use_missing = "pairwise.complete.obs",
  reorder_group = FALSE,
  k = NULL,
  h = NULL,
  plot_dendrogram = TRUE,
  plot_cut_height = NULL,
  root = NULL,
  center = FALSE,
  type = "rectangle",
  size = NULL,
  data = NULL,
  free_labs = waiver(),
  free_spaces = waiver(),
  plot_data = waiver(),
  set_context = TRUE,
  order = NULL,
  name = NULL
)
```

Arguments

mapping	Additional default list of aesthetic mappings to use for plot.
...	Additional arguments passed to <code>geom_segment()</code> .
distance	A string of distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski". Correlation coefficient can be also used, including "pearson", "spearman" or "kendall". In this way, 1 - cor will be used as the distance. In addition, you can also provide a <code>dist</code> object directly or a function return a <code>dist</code> object.
method	A string of the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC). you can also provide a function which returns a <code>hclust</code> object.
use_missing	An optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs". Only used when distance is a correlation coefficient string.
reorder_group	A single boolean value, indicates whether we should do Hierarchical Clustering between groups, only used when previous groups have been established.
k	An integer scalar indicates the desired number of groups.

h	A numeric scalar indicates heights where the tree should be cut.
plot_dendrogram	A boolean value indicates whether plot the dendrogram tree.
plot_cut_height	A boolean value indicates whether plot the cut height.
root	A length one string or numeric indicates the root branch.
center	A boolean value. if TRUE, nodes are plotted centered with respect to the leaves in the branch. Otherwise (default), plot them in the middle of all direct child nodes.
type	A string indicates the plot type, "rectangle" or "triangle".
size	Plot size, can be an unit object.
data	<p>A matrix, a data frame, or even a simple vector that will be converted into a one-column matrix. If the data argument is set to NULL, the align_* will use the layout data. Additionally, the data argument can also accept a function (purrr-like lambda is also okay), which will be applied with the layout data, It is important to note that all align_* functions consider the rows as the observations. It means the NROW(data) must return the same number with the parallel layout axis.</p> <ul style="list-style-type: none"> • layout_heatmap: for column annotation, the layout data will be transposed before using (If data is a function, it will be applied with the transposed matrix). This is necessary because column annotation uses heatmap columns as observations, but we need rows. • layout_stack: the layout data will be used as it is since we place all plots along a single axis.
free_labs	A boolean value or a string containing one or more of "t","l","b",and "r" indicates which axis title should be free from alignment. If NULL, all axis title will be aligned. Default: "tlbr".
free_spaces	A character specifies the ggplot elements which won't count space sizes when alignment. If NULL (default), no space will be removed. See free_space() for allowed values.
plot_data	<p>A function used to transform the plot data before rendering. By default, it'll inherit from the parent layout. If no parent layout, the default is to not modify the data. Use NULL, if you don't want to modify anything.</p> <p>Used to modify the data after layout has been created, but before the data is handled of to the ggplot2 for rendering. Use this hook if the you needs change the default data for all geoms.</p>
set_context	A single boolean value indicates whether to set the active context to current plot. If TRUE, all subsequent ggplot elements will be added into this plot.
order	An single integer for the layout order.
name	A string of the plot name. Used to switch the active context in hmanno() or stack_active() .

Value

A new Align object.

ggplot2 specification

align_dendro initializes a ggplot data and mapping.

The internal will always use a default mapping of `aes(x = .data$x, y = .data$y)`.

The default ggplot data is the node coordinates, in addition, a `geom_segment` layer with a data of the tree segments edge coordinates will be added.

node and tree segments edge coordinates contains following columns:

- `index`: the original index in the tree for the current node
- `label`: node label text
- `x` and `y`: x-axis and y-axis coordinates for current node or the start node of the current edge.
- `xend` and `yend`: the x-axis and y-axis coordinates of the terminal node for current edge.
- `branch`: which branch current node or edge is. You can use this column to color different groups.
- `panel`: which panel current node is, if we split the plot into panel using `facet_grid`, this column will show which panel current node or edge is from. Note: some nodes may fall outside panel (between two panel), so there are possible NA values in this column. We also provide `.panel` column, which always give the right branch for usage of the ggplot facet.
- `.panel`: See `panel`, this is what we often used.
- `panel1` and `panel2`: The `panel1` and `panel2` variables have the same functionality as `panel`, but they are specifically for the edge data and correspond to both nodes of each edge.
- `leaf`: A logical value indicates whether current node is a leaf.

See Also

- [dendrogram_data\(\)](#)
- [hclust2\(\)](#)

Examples

```
ggheatmap(matrix(rnorm(81), nrow = 9)) +
  hmanno("top") +
  align_dendro()
ggheatmap(matrix(rnorm(81), nrow = 9)) +
  hmanno("top") +
  align_dendro(k = 3L)
```

align_gg

Create ggplot object with a customized data

Description

ggalign is just an alias of align_gg.

Usage

```
align_gg(
  data = NULL,
  mapping = aes(),
  size = NULL,
  free_labs = waiver(),
  free_spaces = waiver(),
  plot_data = waiver(),
  limits = TRUE,
  facet = TRUE,
  set_context = TRUE,
  order = NULL,
  name = NULL
)
```

```
ggalign(
  data = NULL,
  mapping = aes(),
  size = NULL,
  free_labs = waiver(),
  free_spaces = waiver(),
  plot_data = waiver(),
  limits = TRUE,
  facet = TRUE,
  set_context = TRUE,
  order = NULL,
  name = NULL
)
```

Arguments

data	<p>A matrix, a data frame, or even a simple vector that will be converted into a one-column matrix. If the data argument is set to NULL, the align_* will use the layout data. Additionally, the data argument can also accept a function (purrr-like lambda is also okay), which will be applied with the layout data, It is important to note that all align_* functions consider the rows as the observations. It means the NROW(data) must return the same number with the parallel layout axis.</p> <ul style="list-style-type: none"> • layout_heatmap: for column annotation, the layout data will be transposed before using (If data is a function, it will be applied with the transposed matrix). This is necessary because column annotation uses heatmap columns as observations, but we need rows. • layout_stack: the layout data will be used as it is since we place all plots along a single axis.
mapping	Additional default list of aesthetic mappings to use for plot.
size	Plot size, can be an unit object.

free_labs	A boolean value or a string containing one or more of "t", "l", "b", and "r" indicates which axis title should be free from alignment. If NULL, all axis title will be aligned. Default: "tlbr".
free_spaces	A character specifies the ggplot elements which won't count space sizes when alignment. If NULL (default), no space will be removed. See free_space() for allowed values.
plot_data	A function used to transform the plot data before rendering. By default, it'll inherit from the parent layout. If no parent layout, the default is to not modify the data. Use NULL, if you don't want to modify anything. Used to modify the data after layout has been created, but before the data is handled of to the ggplot2 for rendering. Use this hook if the you needs change the default data for all geoms.
limits	A boolean value indicates whether to set the layout limits for the plot.
facet	A boolean value indicates whether to set the layout facet for the plot. If this is FALSE, limits will always be FALSE too.
set_context	A single boolean value indicates whether to set the active context to current plot. If TRUE, all subsequent ggplot elements will be added into this plot.
order	An single integer for the layout order.
name	A string of the plot name. Used to switch the active context in hmanno() or stack_active() .

Value

A AlignGG object.

ggplot2 specification

align_gg initializes a ggplot data and mapping.

The internal will always use a default mapping of `aes(y = .data$.y)` or `aes(x = .data$.x)`.

For ggplot usage, matrix (including a simple vector) data is converted into a long-format data frame.

The data in the underlying ggplot object will contain following columns:

- `.panel`: the panel for current layout axis.
- `.x` or `.y`: the x or y coordinates
- `.row_names` and `.row_index`: A factor of the row names and an integer of row index of the original matrix or data frame.
- `.column_names` and `.column_index`: the column names and column index of the original matrix (only applicable if data is a matrix).
- `value`: the actual matrix value (only applicable if data is a matrix).

In the case where the input data is already a data frame, three additional columns (`.row_names`, `.row_index`, and `.panel`) are added to the data frame.

if data is inherit from the [heatmap](#) layout, an additional column will be added.

- `.extra_panel`: the panel information for column (left or right annotation) or row (top or bottom annotation).

Examples

```
ggheatmap(matrix(rnorm(81), nrow = 9)) +  
  hmanno("top") +  
  ggalign() +  
  geom_point(aes(y = value))
```

align_group

Group and align layout based on categorical or factor levels.

Description

Group and align layout based on categorical or factor levels.

Usage

```
align_group(group, set_context = FALSE, name = NULL)
```

Arguments

group	A character define the groups, this will split the axis into different panel.
set_context	A single boolean value indicates whether to set the active context to current plot. If TRUE, all subsequent ggplot elements will be added into this plot.
name	A string of the plot name. Used to switch the active context in hmanno() or stack_active() .

Value

A new Align object.

Examples

```
small_mat <- matrix(rnorm(81), nrow = 9)  
ggheatmap(small_mat) +  
  hmanno("top") +  
  align_group(sample(letters[1:4], ncol(small_mat), replace = TRUE))
```


align_kmeans

*Split layout by k-means clustering groups.***Description**

Split layout by k-means clustering groups.

Usage

```
align_kmeans(
  centers,
  iter.max = 10,
  nstart = 1,
  algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"),
  trace = FALSE,
  data = NULL,
  set_context = FALSE,
  name = NULL
)
```

Arguments

- | | |
|-----------|--|
| centers | either the number of clusters, say k , or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) rows in x is chosen as the initial centres. |
| iter.max | the maximum number of iterations allowed. |
| nstart | if centers is a number, how many random sets should be chosen? |
| algorithm | character: may be abbreviated. Note that "Lloyd" and "Forgy" are alternative names for one algorithm. |
| trace | logical or integer number, currently only used in the default method ("Hartigan-Wong"): if positive (or true), tracing information on the progress of the algorithm is produced. Higher values may produce more tracing information. |
| data | A matrix, a data frame, or even a simple vector that will be converted into a one-column matrix. If the data argument is set to NULL, the align_* will use the layout data. Additionally, the data argument can also accept a function (purrr-like lambda is also okay), which will be applied with the layout data, It is important to note that all align_* functions consider the rows as the observations. It means the <code>NROW(data)</code> must return the same number with the parallel layout axis. <ul style="list-style-type: none"> • layout_heatmap: for column annotation, the layout data will be transposed before using (If data is a function, it will be applied with the transposed matrix). This is necessary because column annotation uses heatmap columns as observations, but we need rows. • layout_stack: the layout data will be used as it is since we place all plots along a single axis. |

set_context	A single boolean value indicates whether to set the active context to current plot. If TRUE, all subsequent ggplot elements will be added into this plot.
name	A string of the plot name. Used to switch the active context in <code>hmanno()</code> or <code>stack_active()</code> .

Value

A new Align object.

Examples

```
ggheatmap(matrix(rnorm(81), nrow = 9)) +
  hmanno("t") +
  align_kmeans(3L)
```

align_panel	<i>Create ggplot object with layout panel data</i>
-------------	--

Description

This is similar with `ggalign()` function, but it will always use the layout panel data. `ggpanel` is just an alias of `align_panel`.

Usage

```
align_panel(
  mapping = aes(),
  size = NULL,
  free_labs = waiver(),
  free_spaces = waiver(),
  plot_data = waiver(),
  limits = TRUE,
  facet = TRUE,
  set_context = TRUE,
  order = NULL,
  name = NULL
)
```

```
ggpanel(
  mapping = aes(),
  size = NULL,
  free_labs = waiver(),
  free_spaces = waiver(),
  plot_data = waiver(),
  limits = TRUE,
  facet = TRUE,
  set_context = TRUE,
```

```

    order = NULL,
    name = NULL
  )

```

Arguments

mapping	Additional default list of aesthetic mappings to use for plot.
size	Plot size, can be an unit object.
free_labs	A boolean value or a string containing one or more of "t", "l", "b", and "r" indicates which axis title should be free from alignment. If NULL, all axis title will be aligned. Default: "tlbr".
free_spaces	A character specifies the ggplot elements which won't count space sizes when alignment. If NULL (default), no space will be removed. See free_space() for allowed values.
plot_data	A function used to transform the plot data before rendering. By default, it'll inherit from the parent layout. If no parent layout, the default is to not modify the data. Use NULL, if you don't want to modify anything. Used to modify the data after layout has been created, but before the data is handled of to the ggplot2 for rendering. Use this hook if the you needs change the default data for all geoms.
limits	A boolean value indicates whether to set the layout limits for the plot.
facet	A boolean value indicates whether to set the layout facet for the plot. If this is FALSE, limits will always be FALSE too.
set_context	A single boolean value indicates whether to set the active context to current plot. If TRUE, all subsequent ggplot elements will be added into this plot.
order	An single integer for the layout order.
name	A string of the plot name. Used to switch the active context in hmanno() or stack_active() .

Value

A AlignPanel object.

ggplot2 specification

align_panel initializes a ggplot data and mapping.

The internal will always use a default mapping of `aes(y = .data$.y)` or `aes(x = .data$.x)`.

The data in the underlying ggplot object contains following columns:

- `.panel`: the panel for current layout axis.
- `.index`: the index of the original layout data.
- `.x` or `.y`: the x or y coordinates

Examples

```
ggheatmap(matrix(rnorm(81), nrow = 9)) +
  hmanno("top") +
  ggalign() +
  geom_point(aes(y = value))
```

align_plots

*Arrange multiple plots into a grid***Description**

Arrange multiple plots into a grid

Usage

```
align_plots(
  ...,
  ncol = NULL,
  nrow = NULL,
  byrow = TRUE,
  widths = NA,
  heights = NA,
  design = NULL,
  guides = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  theme = NULL
)
```

Arguments

...	<dyn-dots> A list of plots, usually the ggplot object. Use NULL to indicate an empty spacer.
ncol, nrow	The dimensions of the grid to create - if both are NULL it will use the same logic as facet_wrap() to set the dimensions
byrow	If FALSE the plots will be filled in in column-major order.
widths, heights	The relative widths and heights of each column and row in the grid. Will get repeated to match the dimensions of the grid. The special value of NA will behave as 1null unit unless a fixed aspect plot is inserted in which case it will allow the dimension to expand or contract to match the aspect ratio of the content.
design	Specification of the location of areas in the layout. Can either be specified as a text string or by concatenating calls to area() together.
guides	Which guide should be collected? A string containing one or more of "t", "l", "b", and "r".

title	The text for the title.
subtitle	The text for the subtitle for the plot which will be displayed below the title.
caption	The text for the caption which will be displayed in the bottom-right of the plot by default.
theme	A <code>theme()</code> object to rendering the guides, title, subtitle, caption, margins and background.

Value

A `alignpatches` object.

Examples

```
# directly copied from patchwork
p1 <- ggplot(mtcars) +
  geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) +
  geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) +
  geom_bar(aes(gear)) +
  facet_wrap(~cyl)
p4 <- ggplot(mtcars) +
  geom_bar(aes(carb))
p5 <- ggplot(mtcars) +
  geom_violin(aes(cyl, mpg, group = cyl))

# Either add the plots as single arguments
align_plots(p1, p2, p3, p4, p5)

# Or use bang-bang-bang to add a list
align_plots(!!!list(p1, p2, p3), p4, p5)

# Match plots to areas by name
design <- "#BB
        AA#"
align_plots(B = p1, A = p2, design = design)

# Compare to not using named plot arguments
align_plots(p1, p2, design = design)
```

align_reorder

Reorders layout observations based on weights or summary statistics.

Description

Reorders layout observations based on weights or summary statistics.

Usage

```
align_reorder(
  fun = rowMeans,
  ...,
  strict = TRUE,
  decreasing = FALSE,
  data = NULL,
  set_context = FALSE,
  name = NULL
)
```

Arguments

fun	A summary function. It should take a data and return the weights for the layout observations.
...	Additional arguments passed to fun.
strict	A boolean value indicates whether the order should be strict. If previous groups has been established, and strict is FALSE, this will reorder the observations in each group.
decreasing	A boolean value. Should the sort order be increasing or decreasing?
data	A matrix, a data frame, or even a simple vector that will be converted into a one-column matrix. If the data argument is set to NULL, the align_* will use the layout data. Additionally, the data argument can also accept a function (purrr-like lambda is also okay), which will be applied with the layout data, It is important to note that all align_* functions consider the rows as the observations. It means the NROW(data) must return the same number with the parallel layout axis. <ul style="list-style-type: none"> • layout_heatmap: for column annotation, the layout data will be transposed before using (If data is a function, it will be applied with the transposed matrix). This is necessary because column annotation uses heatmap columns as observations, but we need rows. • layout_stack: the layout data will be used as it is since we place all plots along a single axis.
set_context	A single boolean value indicates whether to set the active context to current plot. If TRUE, all subsequent ggplot elements will be added into this plot.
name	A string of the plot name. Used to switch the active context in hmanno() or stack_active() .

Value

A new Align object.

Examples

```
ggheatmap(matrix(rnorm(81), nrow = 9)) +
  hmanno("1") +
  align_reorder()
```

area	<i>Define the plotting areas in align_plots</i>
------	---

Description

This is a small helper used to specify a single area in a rectangular grid that should contain a plot. Objects constructed with `area()` can be concatenated together with `c()` in order to specify multiple areas.

Usage

```
area(t, l, b = t, r = l)
```

Arguments

t, b	The top and bottom bounds of the area in the grid
l, r	The left and right bounds of the area in the grid

Details

The grid that the areas are specified in reference to enumerate rows from top to bottom, and columns from left to right. This means that `t` and `l` should always be less or equal to `b` and `r` respectively. Instead of specifying area placement with a combination of `area()` calls, it is possible to instead pass in a single string

```
areas <- c(area(1, 1, 2, 1),  
           area(2, 3, 3, 3))
```

is equivalent to

```
areas <- "A##  
        A#B  
        ##B"
```

Value

A `align_area` object.

Examples

```
p1 <- ggplot(mtcars) +  
  geom_point(aes(mpg, disp))  
p2 <- ggplot(mtcars) +  
  geom_boxplot(aes(gear, disp, group = gear))  
p3 <- ggplot(mtcars) +  
  geom_bar(aes(gear)) +  
  facet_wrap(~cyl)
```

```

layout <- c(
  area(1, 1),
  area(1, 3, 3),
  area(3, 1, 3, 2)
)

# Show the layout to make sure it looks as it should
plot(layout)

# Apply it to a patchwork
align_plots(p1, p2, p3, design = layout)

```

dendrogram_data *Dendrogram x and y coordinates*

Description

Dendrogram x and y coordinates

Usage

```

dendrogram_data(
  tree,
  priority = "right",
  center = FALSE,
  type = "rectangle",
  leaf_pos = NULL,
  leaf_braches = NULL,
  branch_gap = NULL,
  root = NULL
)

```

Arguments

tree	A hclust or a dendrogram object.
priority	A string of "left" or "right". if we draw from right to left, the left will override the right, so we take the "left" as the priority. If we draw from left to right, the right will override the left, so we take the "right" as priority. This is used by align_dendro() to provide support of facet operation in ggplot2 .
center	A boolean value. if TRUE, nodes are plotted centered with respect to the leaves in the branch. Otherwise (default), plot them in the middle of all direct child nodes.
type	A string indicates the plot type, "rectangle" or "triangle".
leaf_pos	The x-coordinates of the leaf node. Must be the same length of the number of observations in tree.
leaf_braches	Branches of the leaf node. Must be the same length of the number of observations in tree. Usually come from cutree .

branch_gap	A single numeric value indicates the gap between different branches.
root	A length one string or numeric indicates the root branch.

Value

A list of 2 data.frame. One for node coordinates, another for edge coordinates. node and tree segments edge coordinates contains following columns:

- index: the original index in the tree for the current node
- label: node label text
- x and y: x-axis and y-axis coordinates for current node or the start node of the current edge.
- xend and yend: the x-axis and y-axis coordinates of the terminal node for current edge.
- branch: which branch current node or edge is. You can use this column to color different groups.
- panel: which panel current node is, if we split the plot into panel using `facet_grid`, this column will show which panel current node or edge is from. Note: some nodes may fall outside panel (between two panels), so there are possible NA values in this column. We also provide `ggpanel` column, which always give the right branch for usage of the `ggplot` facet.
- ggpanel: See panel, this is what we often used.
- panel1 and panel2: The `panel1` and `panel2` variables have the same functionality as `panel`, but they are specifically for the edge data and correspond to both nodes of each edge.
- leaf: A logical value indicates whether current node is a leaf.

Examples

```
dendrogram_data(hclust(dist(USArrests), "ave"))
```

free_align	<i>Free from alignment</i>
------------	----------------------------

Description

`align_plots` will try to align plot panels, and every elements of the plot, following functions remove these restrictions:

- `free_align`: if we want to compose plots without alignment of some panel axes (panel won't be aligned). we can wrap the plot with `free_align`.
- `free_border`: If we want to compose plots without alignment of the panel borders (but still align the panels themselves), we can wrap the plot with `free_border`.
- `free_lab`: If we want to compose plots without alignment of the axis title, we can wrap the plot with `free_lab`.
- `free_space`: Removing the `ggplot` element sizes when aligning.

Usage

```
free_align(plot, axes = "tlbr")
```

```
free_border(plot, borders = "tlbr")
```

```
free_lab(plot, labs = "tlbr")
```

```
free_space(plot, ...)
```

Arguments

plot	A ggplot or alignpatches object.
axes	Which axes shouldn't be aligned? A string containing one or more of "t","l","b",and "r".
borders	Which border shouldn't be aligned? A string containing one or more of "t","l","b",and "r".
labs	Which axis labs to be free? A string containing one or more of "t","l","b",and "r".
...	<p>What sizes of the ggplot2 elements to remove? Allowed values are: "title", "subtitle", "xlab-t", "axis-t", "strip-t", "patch-title-top", "margin-t", "ylab-l", "axis-l", "strip-l", "patch-title-left", "margin-l", "caption", "xlab-b", "axis-b", "strip-b", "patch-title-bottom", "margin-b", "ylab-r", "axis-r", "strip-r", "patch-title-right", and "margin-r". Some unions also allowed:</p> <ul style="list-style-type: none"> • "t"/"top": "title", "subtitle", "xlab-t", "axis-t", "strip-t", "patch-title-top", "margin-t". • "l"/"left": "ylab-l", "axis-l", "strip-l", "patch-title-left", "margin-l". • "b"/"bottom": "caption", "xlab-b", "axis-b", "strip-b", "patch-title-bottom", "margin-b". • "r"/"right": "ylab-r", "axis-r", "strip-r", "patch-title-right", "margin-r". • "x": "xlab-t", "axis-t", "strip-t", "xlab-b", "axis-b", "strip-b". • "y": "ylab-l", "axis-l", "strip-l", "ylab-r", "axis-r", "strip-r". • "xlab"/"xlabs": "xlab-t", "xlab-b". • "ylab"/"ylabs": "ylab-l", "ylab-r". • "lab"/"labs": "xlab-t", "xlab-b", "ylab-l", "ylab-r". • "axis"/"axes": "axis-t", "axis-b", "axis-l", "axis-r". • "strip"/"strips": "strip-t", "strip-b", "strip-l", "strip-r". • "patch-title"/"patch-titles": "patch-title-top", "patch-title-left", "patch-title-bottom", "patch-title-right". • "margin"/"margins": "margin-t", "margin-l", "margin-b", "margin-r".

Value

- free_align: A modified version of plot with a free_align class.

- free_border: A modified version of plot with a free_border class.
- free_lab: A modified version of plot with a free_lab class.
- free_space: A modified version of plot with a free_space class.

Examples

```
# directly copied from patchwork
# Sometimes you have a plot that defies good composition alignment, e.g. due
# to long axis labels
p1 <- ggplot(mtcars) +
  geom_bar(aes(y = factor(gear), fill = factor(gear))) +
  scale_y_discrete(
    "",
    labels = c(
      "3 gears are often enough",
      "But, you know, 4 is a nice number",
      "I would def go with 5 gears in a modern car"
    )
  )
)

# When combined with other plots it ends up looking bad
p2 <- ggplot(mtcars) +
  geom_point(aes(mpg, disp))

align_plots(p1, p2, ncol = 1L)

# We can fix this by using free (here, with the default "panel" type)
align_plots(free_align(p1), p2, ncol = 1L)

# If we still want the panels to be aligned to the right, we can choose to
# free only the left side
align_plots(free_align(p1, axes = "l"), p2, ncol = 1L)

# We could use "label" to fix the layout in a different way
align_plots(p1, free_lab(p2), ncol = 1L)

# Another issue is that long labels are not using already available free
# space.
align_plots(NULL, p1, p2, p2)

# This can be fixed with the "space" type
align_plots(NULL, free_space(p1, "l"), p2, p2)
```

Description

Draw ggplot2 layer with a grob or function.

Usage

```
geom_draw(
  draw = grid::nullGrob(),
  ...,
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  inherit.aes = TRUE
)
```

Arguments

draw	Either a grob object or a function (can be purrr-style) which accepts two arguments (data and coords) and returns a grob . when draw is a function, it is used as the draw_group function in a Geom ggproto object.
...	Additional arguments passed to draw.
mapping	Set of aesthetic mappings created by 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	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot() . A data.frame, 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 data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
stat	The statistical transformation to use on the data for this layer. When using a geom_*() function to construct a layer, the stat argument can be used to override the default coupling between geoms and stats. The stat argument accepts the following: <ul style="list-style-type: none"> • A Stat ggproto subclass, for example StatCount. • A string naming the stat. To give the stat as a string, strip the function name of the stat_ prefix. For example, to use stat_count(), give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.

position	<p>A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:</p> <ul style="list-style-type: none"> • The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
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> .

Details

If you want to combine the functionality of multiple geoms it can usually be achieved by preparing the data for each of the geoms inside the `draw_*()` call and send it off to the different geoms, collecting the output in a `grid::gList` (a list of grobs) if the call is `draw_group()` or a `grid::gTree` (a grob containing multiple children grobs) if the call is `draw_panel()`.

Value

A `ggplot2` layer.

See Also

<https://ggplot2.tidyverse.org/reference/ggplot2-ggproto.html>

Examples

```
ggdraw_text <- grid::textGrob(
  "ggdraw",
  x = c(0, 0, 0.5, 1, 1),
  y = c(0, 1, 0.5, 0, 1),
  hjust = c(0, 0, 0.5, 1, 1),
  vjust = c(0, 1, 0.5, 0, 1)
)
ggplot2::ggplot(data.frame(x = 1, y = 2)) +
  geom_draw(ggdraw_text)
```

ggalignGrob	<i>Generate a plot grob.</i>
-------------	------------------------------

Description

Generate a plot grob.

Usage

```
ggalignGrob(x)

## Default S3 method:
ggalignGrob(x)

## S3 method for class 'patchwork'
ggalignGrob(x)

## S3 method for class 'patch'
ggalignGrob(x)

## S3 method for class 'alignpatches'
ggalignGrob(x)
```

Arguments

`x` An object to be converted into a [grob](#).

Value

A [grob\(\)](#) object.

ggalign_stat	<i>Get the statistics from the layout</i>
--------------	---

Description

Get the statistics from the layout

Usage

```
ggalign_stat(x, ...)
```

```
## S3 method for class 'HeatmapLayout'
ggalign_stat(x, ..., position, what)
```

```
## S3 method for class 'StackLayout'
ggalign_stat(x, ..., what)
```

Arguments

x	A <code>layout_heatmap()</code> or <code>layout_stack()</code> object.
...	Not used currently.
position	A string of "top", "left", "bottom", or "right".
what	A single number or string of the plot elements in the stack layout.

Value

The statistics

hclust2	<i>Generate Tree Structures with Hierarchical Clustering</i>
---------	--

Description

Generate Tree Structures with Hierarchical Clustering

Usage

```
hclust2(
  matrix,
  distance = "euclidean",
  method = "complete",
  use_missing = "pairwise.complete.obs"
)
```

Arguments

matrix	A numeric matrix, or data frame.
distance	A string of distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski". Correlation coefficient can be also used, including "pearson", "spearman" or "kendall". In this way, 1 - cor will be used as the distance. In addition, you can also provide a <code>dist</code> object directly or a function return a <code>dist</code> object.
method	A string of the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC). you can also provide a function which returns a <code>hclust</code> object.
use_missing	An optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs". Only used when distance is a correlation coefficient string.

Value

A `hclust` object.

See Also

- `cor()`
- `dist()`
- `hclust()`

Examples

```
hclust2(dist(USArrests), method = "ward.D")
```

hmanno

Determine the active context of heatmap layout

Description

Determine the active context of heatmap layout

Usage

```
hmanno(
  position = NULL,
  size = NULL,
  width = NULL,
  height = NULL,
  guides = NA,
  free_labs = NA,
  free_spaces = NA,
  plot_data = NA,
  theme = NULL,
  what = waiver()
)
```

Arguments

- | | |
|-----------------------|---|
| <code>position</code> | Which heatmap annotation should get activated? Possible values are follows: <ul style="list-style-type: none"> • A string of "top", "left", "bottom", or "right". • NULL: means set the active context into the heatmap itself. |
| <code>size</code> | A <code>unit</code> object to set the total size of the heatmap annotation. This will only be used if <code>what</code> is a string of "top", "left", "bottom", or "right". <ul style="list-style-type: none"> • If position is "top" or "bottom", size set the total height of the annotation. • If position is "left" or "right", size set the total width of the annotation. |

width, height	Heatmap body width/height, can be a unit object. Only used when position is NULL.
guides	A boolean value or a string containing one or more of "t", "l", "b", and "r" indicates Which guide should be collected. If NULL, no guides will be collected. Default: "tlbr".
free_labs	A boolean value or a string containing one or more of "t", "l", "b", and "r" indicates which axis title should be free from alignment. If NULL, all axis title will be aligned. Default: "tlbr".
free_spaces	A character specifies the ggplot elements which won't count space sizes when alignment. If NULL (default), no space will be removed. See free_space() for allowed values.
plot_data	A function used to transform the plot data before rendering. By default, it'll inherit from the parent layout. If no parent layout, the default is to not modify the data. Use NULL, if you don't want to modify anything. Used to modify the data after layout has been created, but before the data is handled of to the ggplot2 for rendering. Use this hook if the you needs change the default data for all geoms.
theme	A theme() object to rendering the guides, title, subtitle, caption, margins and background. Only used when position is NULL.
what	What should get activated for the anntoation stack? Only used when position is not NULL. See stack_active for details.

Value

A heatmap_active object which can be added into [HeatmapLayout](#).

Examples

```
ggheatmap(matrix(rnorm(81), nrow = 9)) +
  hmanno("top") +
  align_dendro()
```

inset

Create a ggplot inset

Description

Create a ggplot inset

Usage

```
inset(plot, ..., align = "panel", on_top = TRUE, clip = TRUE, vp = NULL)
```

Arguments

plot	Any graphic that can be converted into a grob using <code>patch()</code> .
...	Additional arguments passed to the <code>patch()</code> method.
align	A string specifying the area to place the plot: "full" for the full area, "plot" for the full plotting area (including the axis label), or "panel" for only the actual area where data is drawn.
on_top	A single boolean value indicates whether the graphic plot should be put front-most. Note: the graphic plot will always put above the background.
clip	A single boolean value indicating whether the grob should be clipped if they expand outside their designated area.
vp	A viewport object, you can use this to define the plot area.

Value

A `patch_inset` object, which can be added in `ggplot`.

See Also

- [patch.grob](#)
- [patch.ggplot](#)
- [patch.formula](#)
- [patch.Heatmap](#)
- [patch.HeatmapList](#)
- [patch.HeatmapAnnotation](#)

is.ggheatmap

Reports whether x is a HeatmapLayout object

Description

Reports whether x is a HeatmapLayout object

Usage

```
is.ggheatmap(x)
```

Arguments

x	An object to test
---	-------------------

Value

A boolean value

Examples

```
is.ggheatmap(ggheatmap(1:10))
```

is.ggstack	<i>Reports whether x is a StackLayout object</i>
------------	--

Description

Reports whether x is a StackLayout object

Usage

```
is.ggstack(x)
```

Arguments

x An object to test

Value

A boolean value

Examples

```
is.ggstack(ggstack(1:10))
```

layout-add	<i>Add components to Layout</i>
------------	---------------------------------

Description

Add components to Layout

Usage

```
## S4 method for signature 'Layout,ANY'  
e1 + e2
```

Arguments

e1 A [layout_heatmap\(\)](#) or [layout_stack\(\)](#) object.
e2 An object to be added to the plot, including [gg](#) elements or [align](#) object.

Value

A modified Layout object.

Examples

```
ggheatmap(matrix(rnorm(81), nrow = 9)) +
  hmanno("t") +
  ggalignment() +
  geom_point(aes(y = value))
```

 layout-operator

Plot arithmetic

Description

Plot arithmetic

Usage

```
## S4 method for signature 'Layout,ANY'
e1 & e2
```

```
## S4 method for signature 'Layout,ANY'
e1 - e2
```

Arguments

e1 A [layout_heatmap\(\)](#) or [layout_stack\(\)](#) object.

e2 An object to be added to the plot.

Details

In order to reduce code repetition `ggalign` provides two operators for adding ggplot elements (geoms, themes, facets, etc.) to multiple/all plots in [layout_heatmap\(\)](#) or [layout_stack\(\)](#) object.

Like `patchwork`, `&` add the element to all plots in the plot. If the element is a [theme](#), this will also modify the layout theme.

Unlike `patchwork`, the `-` operator adds ggplot2 elements (geoms, themes, facets, etc.) rather than a ggplot plot. The key difference between `&` and `-` is in how they behave in [layout_heatmap\(\)](#). The `-` operator only applies the element to the current active context in [layout_heatmap\(\)](#). Using `-` might seem unintuitive if you think of the operator as "subtract", the underlying reason is that `-` is the only operator in the same precedence group as `+`.

Value

A modified Layout object.

Examples

```
mat <- matrix(rnorm(81), nrow = 9)
ggheatmap(mat) +
  hmanno("top") +
  align_dendro() &
  theme(panel.border = element_rect(
    colour = "red", fill = NA, linewidth = unit(2, "mm")
  ))
ggheatmap(mat) +
  hmanno("top") +
  align_dendro() -
  theme(panel.border = element_rect(
    colour = "red", fill = NA, linewidth = unit(2, "mm")
  ))
```

layout_heatmap	<i>Arrange plots around a Heatmap</i>
----------------	---------------------------------------

Description

ggheatmap is an alias of layout_heatmap.

Usage

```
layout_heatmap(
  data,
  mapping = aes(),
  ...,
  filling = TRUE,
  set_context = TRUE,
  order = NULL,
  name = NULL,
  environment = parent.frame()
)

ggheatmap(
  data,
  mapping = aes(),
  ...,
  filling = TRUE,
  set_context = TRUE,
  order = NULL,
  name = NULL,
  environment = parent.frame()
)
```

Arguments

data	A numeric or character vector, a data frame, and any other data which can be converted into a matrix. Simple vector will be converted into a one column matrix.
mapping	Default list of aesthetic mappings to use for plot. If NULL, will use <code>aes(.data\$.x, .data\$.y)</code> .
...	Additional arguments passed to <code>geom_tile</code> . Only used when <code>filling = TRUE</code> .
filling	A boolean value indicates whether to fill the heatmap. If you want to custom the filling style, you can set to FALSE.
set_context	A single boolean value indicates whether to set the active context to current plot. If TRUE, all subsequent ggplot elements will be added into this plot.
order	An single integer for the layout order.
name	A string of the plot name. Used to switch the active context in <code>hmanno()</code> or <code>stack_active()</code> .
environment	[Deprecated] Used prior to tidy evaluation.

Value

A HeatmapLayout object.

ggplot2 specification

The data input in `ggheatmap` will be converted into the long formatted data frame when drawing. The default mapping will use `aes(.data$.x, .data$.y)`, you can use `mapping` argument to control it. The data in the underlying ggplot object contains following columns:

- `.xpanel` and `.ypanel`: the column and row panel
- `.x` and `.y`: the x and y coordinates
- `.row_names` and `.column_names`: A factor of the row and column names of the original matrix (only applicable when names exist).
- `.row_index` and `.column_index`: the row and column index of the original matrix.
- `value`: the actual matrix value.

Examples

```
ggheatmap(1:10)
ggheatmap(letters)
```

layout_stack	<i>Put plots horizontally or vertically</i>
--------------	---

Description

ggstack is an alias of layout_stack.

Usage

```
layout_stack(data, direction = NULL, ..., environment = parent.frame())
```

```
ggstack(data, direction = NULL, ..., environment = parent.frame())
```

Arguments

data	A numeric or character vector, a data frame, or a matrix.
direction	A string of "horizontal" or "vertical", indicates the direction of the stack layout.
...	Not used currently.
environment	[Deprecated] Used prior to tidy evaluation.

Value

A StackLayout object.

Examples

```
ggstack(matrix(rnorm(100L), nrow = 10L)) + align_dendro()
```

patch.formula	<i>Convert Object into a Grob</i>
---------------	-----------------------------------

Description

The patch() function is used by wrap() to convert objects into a grob, enabling their alignment within align_plots().

Usage

```
## S3 method for class 'formula'
patch(x, ..., device = NULL)
```

Arguments

x	An object to be converted into a grob .
...	Not used currently.
device	A function that opens a graphics device for <code>grid.echo()</code> to work on. By default this is an off-screen, in-memory device based on the pdf device. This default device may not be satisfactory when using custom fonts.

Value

A [grob](#) object.

patch.ggplot	<i>Convert Object into a Grob</i>
--------------	-----------------------------------

Description

The `patch()` function is used by `wrap()` to convert objects into a [grob](#), enabling their alignment within `align_plots()`.

Usage

```
## S3 method for class 'ggplot'
patch(x, ...)
```

Arguments

x	An object to be converted into a grob .
...	Not used currently.

Value

A [grob](#) object.

patch.grob	<i>Convert Object into a Grob</i>
------------	-----------------------------------

Description

The `patch()` function is used by `wrap()` to convert objects into a [grob](#), enabling their alignment within `align_plots()`.

Usage

```
## S3 method for class 'grob'
patch(x, ...)
```


Arguments

x An object to be converted into a [grob](#).
 ... Not used currently.

Value

A [grob](#) object.

patch.Heatmap	<i>Convert Object into a Grob</i>
---------------	-----------------------------------

Description

The patch() function is used by [wrap\(\)](#) to convert objects into a [grob](#), enabling their alignment within [align_plots\(\)](#).

Usage

```
## S3 method for class 'Heatmap'
patch(x, ..., device = NULL)

## S3 method for class 'HeatmapList'
patch(x, ..., device = NULL)

## S3 method for class 'HeatmapAnnotation'
patch(x, ..., device = NULL)
```

Arguments

x An object to be converted into a [grob](#).
 ... Additional arguments passed to [draw\(\)](#).
 device A function that opens a graphics device for temporary rendering. By default this is an off-screen, in-memory device based on the pdf device, but this default device may not be satisfactory when using custom fonts.

Value

A [grob](#) object.

patch_titles	<i>Add patch titles to plot borders</i>
--------------	---

Description

This function extends `ggplot2`'s title functionality, allowing you to add titles to each border of the plot: top, left, bottom, and right.

Usage

```
patch_titles(  
  top = waiver(),  
  left = waiver(),  
  bottom = waiver(),  
  right = waiver()  
)
```

Arguments

top, left, bottom, right

A string specifying the title to be added to the top, left, bottom, and right border of the plot.

Details

You can also use `labs()` to specify titles for the top, left, bottom, and right borders of the plot.

The appearance and alignment of these patch titles can be customized using `theme()`:

- `plot.patch_title/plot.patch_title.*`: Controls the text appearance of patch titles. By default, `plot.patch_title` inherit from `plot.title`, and settings for each border will inherit from `plot.patch_title`, with the exception of the `angle` property, which is not inherited.
- `plot.patch_title.position/plot.patch_title.position.*`: Determines the alignment of the patch titles. By default, `plot.patch_title.position` inherit from `plot.title.position`, and settings for each border will inherit from `plot.patch_title`. The value "panel" aligns the patch titles with the plot panels. Setting this to "plot" aligns the patch title with the entire plot (excluding margins and plot tags).

Value

A `labs` object to be added to `ggplot`.

`read_example`*Read Example Data*

Description

This function reads example data from the file. If no file is specified, it returns a list of available example files.

Usage

```
read_example(file = NULL)
```

Arguments

`file` A string representing the name of the example file to be read. If `NULL`, the function will return a list of available example file names.

Value

If `file` is `NULL`, returns a character vector of available example file names. Otherwise, returns the contents of the specified example file, read as an R object.

Examples

```
read_example()
```

`stack_active`*Determine the active context of stack layout*

Description

Determine the active context of stack layout

Usage

```
stack_active(  
  sizes = NULL,  
  guides = NA,  
  free_labs = NA,  
  free_spaces = NA,  
  plot_data = NA,  
  theme = NULL,  
  what = NULL  
)
```

Arguments

sizes	A numeric or unit object of length 3 indicates the relative widths (direction = "horizontal") / heights (direction = "vertical").
guides	A boolean value or a string containing one or more of "t", "l", "b", and "r" indicates Which guide should be collected. If NULL, no guides will be collected. Default: "tlbr".
free_labs	A boolean value or a string containing one or more of "t", "l", "b", and "r" indicates which axis title should be free from alignment. If NULL, all axis title will be aligned. Default: "tlbr".
free_spaces	A character specifies the ggplot elements which won't count space sizes when alignment. If NULL (default), no space will be removed. See free_space() for allowed values.
plot_data	A function used to transform the plot data before rendering. By default, it'll inherit from the parent layout. If no parent layout, the default is to not modify the data. Use NULL, if you don't want to modify anything. Used to modify the data after layout has been created, but before the data is handled of to the ggplot2 for rendering. Use this hook if the you needs change the default data for all geoms.
theme	A theme() object to rendering the guides, title, subtitle, caption, margins and background. Only used when position is NULL.
what	What should get activated for the stack layout? Possible values are follows: <ul style="list-style-type: none"> • A single number or string of the plot elements in the stack layout. Usually you are waive to use this, since the adding procedure can be easily changed. • NULL: Remove any active context, this is useful when the active context is a layout_heatmap() object, where any Align objects will be added into the heatmap. By removing the active context, we can add Align object into the layout_stack() .

Value

A `stack_active` object which can be added into [StackLayout](#).

Examples

```
ggstack(matrix(1:9, nrow = 3L)) +
  ggheatmap() +
  # ggheatmap will set the active context, directing following addition
  # into the heatmap plot area. To remove the heatmap active context,
  # we can use `stack_active()` which will direct subsequent addition into
  # the stack
  stack_active() +
  # here we add a dendrogram to the stack.
  align_dendro()
```

`wrap`*Wrap Arbitrary Graphics for Alignment*

Description

The `wrap()` function allows non-ggplot2 elements to be converted into a compliant representation for use with `align_plots()`. This is useful for adding any graphics that can be converted into a `grob` with the `patch()` method.

Usage

```
wrap(plot, ..., align = "panel", on_top = TRUE, clip = TRUE, vp = NULL)
```

Arguments

<code>plot</code>	Any graphic that can be converted into a <code>grob</code> using <code>patch()</code> .
<code>...</code>	Additional arguments passed to the <code>patch()</code> method.
<code>align</code>	A string specifying the area to place the plot: "full" for the full area, "plot" for the full plotting area (including the axis label), or "panel" for only the actual area where data is drawn.
<code>on_top</code>	A single boolean value indicates whether the graphic plot should be put front-most. Note: the graphic plot will always put above the background.
<code>clip</code>	A single boolean value indicating whether the grob should be clipped if they expand outside their designated area.
<code>vp</code>	A <code>viewport</code> object, you can use this to define the plot area.

Value

A `wrapped_plot` object that can be directly placed into `align_plots()`.

See Also

- [patch.grob](#)
- [patch.ggplot](#)
- [patch.formula](#)
- [patch.Heatmap](#)
- [patch.HeatmapList](#)
- [patch.HeatmapAnnotation](#)

Index

`+`, Layout, ANY-method (layout-add), 27
`+`.HeatmapLayout (layout-add), 27
`+`.Layout (layout-add), 27
`+`.StackLayout (layout-add), 27
`+`.ggheatmap (layout-add), 27
`+`.ggstack (layout-add), 27
`-`, Layout, ANY-method (layout-operator), 28
`-`.HeatmapLayout (layout-operator), 28
`-`.Layout (layout-operator), 28
`-`.StackLayout (layout-operator), 28
`-`.ggheatmap (layout-operator), 28
`-`.ggstack (layout-operator), 28
`&`, Layout, ANY-method (layout-operator), 28
`&`.HeatmapLayout (layout-operator), 28
`&`.Layout (layout-operator), 28
`&`.StackLayout (layout-operator), 28
`&`.ggheatmap (layout-operator), 28
`&`.ggstack (layout-operator), 28

`aes()`, 20
`align`, 27
`align_dendro`, 2
`align_dendro()`, 16
`align_gg`, 5
`align_group`, 8
`align_kmeans`, 9
`align_panel`, 10
`align_plots`, 12, 17
`align_plots()`, 31–33, 37
`align_reorder`, 13
`alignpatches`, 18
`area`, 15
`area()`, 12

`borders()`, 21

`cor()`, 24
`cutree`, 16

`dendrogram`, 16
`dendrogram_data`, 16
`dendrogram_data()`, 5
`dist`, 3, 23
`dist()`, 24
`draw()`, 33
`dyn-dots`, 12

`facet_grid`, 5, 17
`facet_wrap()`, 12
`fortify()`, 20
`free_align`, 17
`free_border` (free_align), 17
`free_lab` (free_align), 17
`free_space` (free_align), 17
`free_space()`, 4, 7, 11, 25, 36

Geom, 20
`geom_draw`, 19
`geom_segment`, 5
`geom_segment()`, 3
`geom_tile`, 30
`gg`, 27
`ggalign` (align_gg), 5
`ggalign_stat`, 22
`ggalignGrob`, 22
`ggheatmap` (layout_heatmap), 29
`ggpanel` (align_panel), 10
`ggplot`, 18
`ggplot()`, 20
`ggstack` (layout_stack), 31
`grid::gList`, 21
`grid::gTree`, 21
`grob`, 20, 22, 26, 31–33, 37
`grob()`, 22

`hclust`, 3, 16, 23, 24
`hclust()`, 24
`hclust2`, 23
`hclust2()`, 5

heatmap, 7
HeatmapLayout, 25
hmanno, 24
hmanno(), 4, 7, 8, 10, 11, 14, 30

inset, 25
is.ggheatmap, 26
is.ggstack, 27

labs, 34
labs(), 34
layer position, 21
layer stat, 20
layout-add, 27
layout-operator, 28
layout_heatmap, 29
layout_heatmap(), 23, 27, 28, 36
layout_stack, 31
layout_stack(), 23, 27, 28, 36

patch(), 26, 37
patch.formula, 26, 31, 37
patch.ggplot, 26, 32, 37
patch.grob, 26, 32, 37
patch.Heatmap, 26, 33, 37
patch.HeatmapAnnotation, 26, 37
patch.HeatmapAnnotation
 (patch.Heatmap), 33
patch.HeatmapList, 26, 37
patch.HeatmapList (patch.Heatmap), 33
patch_titles, 34

read_example, 35

stack_active, 25, 35
stack_active(), 4, 7, 8, 10, 11, 14, 30
StackLayout, 36

theme, 28
theme(), 13, 25, 34, 36

unit, 4, 6, 11, 24, 25, 36

viewport, 26, 37

wrap, 37
wrap(), 31–33