

Package ‘SimpleUpset’

September 1, 2026

Type Package

Title Create Upset Plots

Version 0.1.6

Description Create Upset plots using a combination of 'ggplot2' and 'patchwork'.

URL <https://github.com/smped/SimpleUpset>

BugReports <https://github.com/smped/SimpleUpset/issues>

License GPL-3

Encoding UTF-8

Depends ggplot2 (>= 4.0.0), patchwork (>= 1.3.2), R (>= 4.1.0),

Imports dplyr, methods, rlang (>= 1.1.6), S7, scales, tidyr,
tidyselect

Suggests knitr, pander, pkgdown, testthat (>= 3.0.0), tidyverse (>= 2.0.0),

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Stevie Pederson [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-8197-3303>>),
Lachlan Baer [ctb]

Maintainer Stevie Pederson <stephen.pederson.au@gmail.com>

Repository CRAN

Date/Publication 2026-09-01 15:50:02 UTC

Contents

default_set_layers	2
simpleUpSet	5

Index	9
--------------	----------

default_set_layers	<i>Define default layers for individual UpSet components</i>
--------------------	--

Description

Define and modify default layers for individual UpSet components

Usage

```
default_set_layers(  
  ...,  
  fill = NULL,  
  labels = "size",  
  f = comma,  
  expand = c(0.2, 0),  
  hjust = 1.1,  
  vjust = 0.5,  
  angle = 0,  
  lineheight = 1.2,  
  fontface = 1,  
  label_size = 3.5,  
  name = "Set Size",  
  dry_run = FALSE  
)  
  
default_intersect_layers(  
  ...,  
  fill = NULL,  
  labels = "size",  
  f = comma,  
  expand = c(0, 0.05),  
  hjust = 0.5,  
  vjust = -0.5,  
  angle = 0,  
  fontface = 1,  
  lineheight = 1.2,  
  label_size = 3.5,  
  name = "Intersection Size",  
  dry_run = FALSE  
)  
  
default_grid_layers(  
  ...,  
  colour = NULL,  
  fill = NULL,  
  light = "grey80",  
  dark = "grey23",
```

```

    shape = 19,
    size = 4,
    name = NULL,
    dry_run = FALSE
)

```

Arguments

...	additional layers to include alongside default layers. Will be added after the default layers
fill	Column to fill set bars by. Can be 'set' or another column within the main data object
labels	Choose either size or prop to label bars with totals or the proportion of all intersections
f	Function for labelling set or intersection sizes
expand	Multiplicative axis expansion passed to expansion()
hjust, vjust, lineheight, angle, fontface	Passed to respective elements for simple adjustment of either set or intersection sizes
label_size	Passed to labels for sets and intersections
name	Main axis title
dry_run	Set as TRUE to view the unevaluated layers which are defined as the defaults. Additional layers passed through the ellipsis will not be included as part of a dry_run
colour	Primarily used for highlighting points and segments in the intersections matrix
light, dark	default colours for empty intersections (light) and for both non-empty intersections and segments (dark)
shape, size	Point shape/size passed to the intersections matrix

Details

These functions define the default layers for inclusion on UpSet plots.

The returned object is a list with a series of layers, scales, themes etc which represent the default plotting layers for each of the sets, intersections and intersections matrix (grid) panels.

A series of common arguments have been defined to enable common modifications without recreating the list from scratch. These include modifying the mapping to fill, axis expansion to better accommodate labels, labelling functions for set/intersection sizes, and axis titles.

Additional layers, such as `scale_fill_*` elements, guides or themes, can be simply included by passing to the function, without any requirement for naming, and are handled by the ellipsis.

The entire command used to create default layers can be shown by calling each function using the argument `dry_run = TRUE`. This can be helpful for creating custom layers, by starting with then modifying the defaults.

The returned object is a simple list, and are easily modifiable using simple list operations. Each list of default layers is described clearly below. If passing additional scales, themes, layers or

guides using the ellipsis, these additional elements will automatically be placed after the defaults. Importantly, these will be created as lists, then can be re-ordered using standard list manipulation.

Default Layers For Sets:

ggplot2 element

```
aes(y = set)
geom_bar(bar_aes)
geom_text(aes(x = size, label = f(size)), hjust = hjust, size = label_size)
scale_x_reverse(expand = c(expand, 0, 0, 0), name = name, labels = f)
scale_y_discrete(position = "right", name = NULL, labels = NULL)
theme(axis.text.y.right = element_text(hjust = 0.5), axis.ticks.y.right = element_blank(), margins = margin(5.5, 5.5, 0, 0))
```

Default Layers For Intersections:

ggplot2 element

```
aes(x = intersect)
geom_bar(bar_aes)
geom_text(aes(y = size, label = f(size)), vjust = vjust, size = label_size)
scale_x_discrete(name = NULL, labels = NULL)
scale_y_continuous(name = name, expand = c(0, 0, expand, 0), labels = f)
theme(axis.ticks.x.bottom = element_blank(), margins = margin(5.5, 5.5, 0, 0))
```

Comment

Intersections are placed along the x-axis.
If fill = NULL, bar_aes is the layer.
Adds intersection totals using the 'size' column.
Tidies up the x-axis, hiding intersection labels.
Standard y-axis with name 'prop'.
Ensures margins and tick marks are consistent.

Default Layers For Intersections Matrix (i.e. Grid):

ggplot2 element

```
aes(x = intersect, y = set)
if (!is.null(colour)) geom_point(mapping = points_aes, size = size, shape = shape) else geom_point(mapping = points_aes, size = size, shape = shape, colour = light)
geom_point(size = size, shape = shape, colour = light)
if (!is.null(colour)) geom_segment(segment_aes) else geom_segment(segment_aes, colour = dark)
scale_y_discrete(name = NULL)
scale_x_discrete(name = name, labels = NULL)
guides(colour = guide_none())
theme(margins = margin(5.5, 5.5, 5.5, 0), axis.text.y = element_text(hjust = 0.5), axis.ticks = element_blank())
```

Panel Internals:

Internally, the supplied data.frame has the additional columns 'intersect', 'degree' added, along with the optional 'highlight' column. This object is used to directly create bars using geom_bar() and as such, any of the additional columns can be passed to geom_bar() as mapping aesthetics, along with all original columns.

For both the sets and intersection totals (i.e. labels), separate tables are created specifically for printing totals at the top (or left) of each bar, and these tables are specifically passed to those layers. Totals are included as 'size' and the proportion of all intersections is also included as the column 'prop' for both the sets and intersections panel. Whilst default labels are added using 'size', changing this to 'prop' and using `scales::percent()` will work and is supported.

Value

List of ggplot2 elements

Examples

```
# View the un-evaluated list of default layers for the sets
default_set_layers(dry_run = TRUE)

# Create set layers colouring by set name, and hiding the legend
set_list <- default_set_layers(
  fill = "set", scale_fill_brewer(palette = "Set1"), guides(fill = guide_none())
)
sapply(set_list, is)
```

simpleUpSet

Make simple UpSet plots

Description

Make simple UpSet plots using ggplot2 and patchwork

Usage

```
simpleUpSet(
  x,
  sets = NULL,
  sort_sets = size,
  sort_intersect = list(desc(size), degree, set),
  n_intersect = 20,
  min_size = 0,
  min_degree = 1,
  max_degree = length(sets),
  set_layers = default_set_layers(),
  intersect_layers = default_intersect_layers(),
  grid_layers = default_grid_layers(),
  highlight = NULL,
  highlight_levels = NULL,
  annotations = list(),
  width = 0.75,
  height = 0.75,
  vjust_ylab = 0.8,
  stripe_colours = c("grey90", "white"),
  keep_empty = FALSE,
  guides = "keep",
  top_left = NULL,
```

```

    show_sets = TRUE,
    ...,
    na.rm = TRUE
  )

```

Arguments

<code>x</code>	Input data frame
<code>sets</code>	Character vector listing columns of <code>x</code> to plot
<code>sort_sets</code>	<data-masking> specification for set order, using variables such as <code>size</code> , <code>desc(size)</code> or <code>NULL</code> . Passed internally to <code>dplyr::arrange()</code> . The only possible options are <code>size</code> , <code>desc(size)</code> or <code>NULL</code> (for sets in the order passed). Can additionally accept the arguments "ascending", "descending" or "none"
<code>sort_intersect</code>	list of <data-masking> specifications for intersection order. Passed internally to <code>dplyr::arrange()</code> . The available columns are <code>size</code> , <code>degree</code> and <code>set</code> , along with <code>highlight</code> if specified. Any other column names will cause an error. The default order is in descending sizes, using degree and set to break ties.
<code>n_intersect</code>	Maximum number of intersections to show
<code>min_size</code>	Only show intersections larger than this value
<code>min_degree, max_degree</code>	Only show intersections within this range
<code>set_layers</code>	List of ggplot2 layers, scales and themes to define the appearance of the sets panel. Can be obtained and extended using <code>default_set_layers()</code>
<code>intersect_layers</code>	List of ggplot2 layers, scales and themes to define the appearance of the intersections panel. Can be obtained and extended using <code>default_intersect_layers()</code>
<code>grid_layers</code>	List of ggplot2 layers, scales & themes
<code>highlight</code>	<code>case_when()</code> statement defining all intersections to highlight using <code>geom_intersect</code> and <code>scale_fill/colour_intersect</code> . Will add a column named <code>highlight</code> which can be called from any geom passed to the intersections barplot or matrix
<code>highlight_levels</code>	Given the <code>highlight</code> column will be coerced to a factor when setting colours etc, levels can be manually set here for finer control.
<code>annotations</code>	list where each element is a list of ggplot2 layers. Each element will be added as an upper annotation panel above the intersections plot. All layer types (geom, scale, aes, stat, labs etc) can be passed with the exception of facets.
<code>width, height</code>	Proportional width and height of the intersection panel
<code>vjust_ylab</code>	Used to nudge the y-axis labels closer to the axis
<code>stripe_colours</code>	Colours for background stripes in the lower two panels. For no stripes, set as <code>NULL</code>
<code>keep_empty</code>	Keep empty sets in the figure
<code>guides</code>	Passed to <code>plot_layout()</code>
<code>top_left</code>	Optional ggplot object to show in the top left panel. Will default to an empty ggplot object

show_sets	logical(1). Show the sets panel (default: TRUE)
...	Not used
na.rm	NA handling

Details

Taking a subset of columns from a data.frame, create an UpSet plot showing all intersections as specified. Columns chosen for the sets and intersections must contain logical values or be strictly 0/1 values.

Internally, data objects will have the variables `set` and `intersect` which can be referred to when passing custom `aes()` mappings to various layers. If specifying highlights, the column `highlight` will also be added as a column to the data.frame containing intersections data, following the `case_when` output provided as the argument. Scales can be passed to the intersections and grid panels, taking this structure into account.

Any additional layers passed using `annotations()` will have layers added after an initial, internal call to `ggplot(data, aes(x = intersect))`. Additional columns can be used where appropriate for creating boxplots etc.

A list of ggplot2 layers, scales, guides and themes is expected in each of the `set_layers`, `intersect_layers` or `grid_layers` arguments, with defaults generated by calls to `default_set_layers()`, `default_intersect_layers()` or `default_grid_layers()`. These can be used as templates to full customisation by creating a custom list object, or modified directly using the ellipsis

Value

Object of class 'patchwork' containing multiple ggplot panels

Examples

```
## Use a modified version of the movies data provided with the package UpSetR
library(tidyverse)
theme_set(theme_bw())
sets <- c("Action", "Comedy", "Drama", "Thriller", "Romance")
movies <- system.file("extdata", "movies.tsv.gz", package = "SimpleUpset") %>%
  read_tsv() %>%
  mutate(
    Decade = fct_inorder(Decade) %>% fct_rev()
  )
simpleUpSet(movies, sets)

## Add a detailed upper plot
simpleUpSet(
  movies, sets, n_intersect = 10,
  annotations = list(
    list(
      aes(y = AvgRating),
      geom_jitter(aes(colour = Decade), height = 0, width = 0.3, alpha = 0.5),
      geom_violin(fill = NA, quantiles = 0.5, quantile.linetype = 1),
      scale_colour_brewer(palette = "Paired"),
      guides(colour = guide_legend(nrow = 2, reverse = TRUE))
    )
  )
)
```

```
    )  
  ), guides = "collect"  
) &  
  theme(legend.position = "bottom")  
  
## Modify set colours  
set_cols <- c(  
  Action = "red", Comedy = "grey23", Drama = "red",  
  Romance = "grey23", Thriller = "grey23"  
)  
simpleUpSet(  
  movies, sets,  
  set_layers = default_set_layers(  
    fill = "set", scale_fill_manual(values = set_cols), guides(fill = guide_none())  
  )  
)
```

Index

`case_when()`, [6](#)

`default_grid_layers`
 (`default_set_layers`), [2](#)

`default_grid_layers()`, [7](#)

`default_intersect_layers`
 (`default_set_layers`), [2](#)

`default_intersect_layers()`, [6](#), [7](#)

`default_set_layers`, [2](#)

`default_set_layers()`, [6](#), [7](#)

`dplyr::arrange()`, [6](#)

`expansion()`, [3](#)

`plot_layout()`, [6](#)

`scales::percent()`, [4](#)

`simpleUpSet`, [5](#)