

Package ‘upsetly’

August 7, 2026

Type Package

Title Interactive UpSet Plots Using 'plotly'

Version 0.1.1

Description Creates interactive UpSet-style visualizations for exploring intersections among multiple sets, following the layout of Lex et al. (2014) <[doi:10.1109/TVCG.2014.2346248](https://doi.org/10.1109/TVCG.2014.2346248)>. 'plotly' widgets combine set-size bars, intersection-size bars, and membership matrices, and can synchronize complete intersection details with an HTML element for copying in 'Quarto', 'R Markdown', and 'Shiny' documents.

URL <https://github.com/Rbunensi/upsetly>,
<https://rbunensi.github.io/upsetly/>

BugReports <https://github.com/Rbunensi/upsetly/issues>

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.6.0)

Imports htmltools, htmlwidgets, jsonlite, plotly

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Author Anbunensi [aut, cre]

Maintainer Anbunensi <an.bunengsi@qq.com>

Repository CRAN

Date/Publication 2026-08-07 10:40:02 UTC

Contents

upsetly	2
Index	4

Description

upsetly() creates an interactive UpSet-style visualization with three coordinated parts: set-size bars, intersection-size bars, and a membership matrix. Intersection tooltips can include member identifiers. In HTML documents, complete intersection details can optionally be synchronized to a separate element so that readers can select and copy the text.

Usage

```
upsetly(
  x,
  set_cols = NULL,
  id_col = NULL,
  min_intersection_size = 1,
  max_n_intersections = NULL,
  point_size = 8,
  bar_color_sets = "black",
  bar_color_inters = "black",
  active_color = "black",
  inactive_color = "lightgray",
  line_color = "black",
  title = "UpSet (plotly)",
  height = 500,
  width = NULL,
  members_per_line = 20,
  box_id = NULL
)
```

Arguments

x	A data frame containing set-membership columns and, optionally, an identifier column.
set_cols	A non-empty character vector naming the membership columns in x. Logical values and non-zero numeric values are treated as members. For character and factor columns, empty strings and common false values such as "0", "false", and "no" are treated as non-members; other non-empty values are treated as members. If NULL, all columns other than id_col are used.
id_col	An optional single column name containing element identifiers. If NULL, sequential identifiers are generated.
min_intersection_size	A positive integer giving the smallest intersection to retain.

<code>max_n_intersections</code>	An optional positive integer giving the maximum number of intersections to display, ordered by decreasing size. For compatibility with earlier versions, it also limits the number of member identifiers shown in each hover tooltip. Complete details synchronized to <code>box_id</code> are not truncated.
<code>point_size</code>	A positive number giving the marker size in the membership matrix.
<code>bar_color_sets</code>	Color used for the set-size bars.
<code>bar_color_inters</code>	Color used for the intersection-size bars.
<code>active_color</code>	Color used for active membership markers.
<code>inactive_color</code>	Color used for inactive membership markers.
<code>line_color</code>	Color used for lines joining active membership markers.
<code>title</code>	A single character string used as the plot title.
<code>height</code>	A positive number giving the widget height in pixels.
<code>width</code>	An optional positive number giving the widget width in pixels.
<code>members_per_line</code>	A positive integer giving the number of member identifiers placed on each tooltip line.
<code>box_id</code>	An optional single character string containing the HTML id of an element, commonly a <code>pre</code> element. Hovering over an intersection bar updates the element, clicking locks its content, and double-clicking the plot unlocks and clears it.

Value

A plotly `htmlwidget`.

Examples

```
set.seed(1)
memberships <- data.frame(
  gene = paste0("g", seq_len(40)),
  A = stats::rbinom(40, 1, 0.35),
  B = stats::rbinom(40, 1, 0.40),
  C = stats::rbinom(40, 1, 0.25),
  stringsAsFactors = FALSE
)

upsetly(
  memberships,
  set_cols = c("A", "B", "C"),
  id_col = "gene",
  max_n_intersections = 12
)
```

Index

upsetly, [2](#)