

Package: linkagg (via r-universe)

October 11, 2026

Title Linked Selection Across Aggregate Views

Version 0.1.0

Description Brush a row-level display and see aggregate displays fill in proportion to the rows selected, with the row-to-group mapping retained. Aggregate views such as bar charts summarise many rows into one mark, so a selection made on individual rows is resolved back through the row to group mapping to fill each mark partially. Output is an 'htmlwidget' that works inside 'shiny' or as a single self-contained HTML file with no server, so an interactive figure can be archived or shared like a static one.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports utils, htmlwidgets (>= 1.6.0)

Suggests shiny (>= 1.7.0), bslib (>= 0.5.0), testthat (>= 3.0.0),
htmltools, survival, pharmaverseadam

Config/testthat/edition 3

URL <https://github.com/renit12345-ship-it/linkagg>

BugReports <https://github.com/renit12345-ship-it/linkagg/issues>

Config/pak/sysreqs cmake make libuv1-dev

Repository <https://renit12345-ship-it.r-universe.dev>

Date/Publication 2026-08-17 17:23:55 UTC

RemoteUrl <https://github.com/renit12345-ship-it/linkagg>

RemoteRef HEAD

RemoteSha b545e59bf50fb033f3938b801b3139fcb9c55590

Contents

as_linkagg_widget	2
linkagg	3
linkagg-shiny	4
run_linkagg_app	5
view_bars	5
view_hist	7
view_points	8
view_table	9
view_volcano	9
Index	11

as_linkagg_widget	<i>Render a linked figure</i>
-------------------	-------------------------------

Description

Builds the htmlwidget. Save it with `htmlwidgets::saveWidget()` and `selfcontained = TRUE` to get a single HTML file that opens offline with no server, which is the point of the package.

Usage

```
as_linkagg_widget(  
  spec,  
  width = NULL,  
  height = NULL,  
  caption = NULL,  
  stamp = TRUE,  
  elementId = NULL  
)
```

Arguments

spec	A linkagg_spec, or an already-built widget, returned unchanged.
width, height	Widget dimensions. height defaults to a size that fits the displays you added, including one sub-bar per treatment arm.
caption	Optional free text shown in the footer, for example the dataset and any subset applied.
stamp	Add a provenance line recording the render time, package version, row count and per-arm denominators. Keep this on for anything you intend to share or archive.
elementId	Optional DOM id.

Value

An object of class htmlwidget.

linkagg

*Start a linked figure***Description**

Begins a specification. Add displays with `view_points()`, `viewBars()` and `viewTable()`, then render with `as_linkagg_widget()`.

Usage

```
linkagg(
  data,
  key,
  threads = TRUE,
  thread_cap = 160L,
  points = c("auto", "svg", "canvas"),
  canvas_threshold = 6000L,
  palette = NULL
)
```

Arguments

<code>data</code>	A data frame with one row per unit of selection. For clinical safety use this is a subject-level dataset such as ADSL, one row per subject, restricted to the analysis population.
<code>key</code>	Column uniquely identifying each row, such as USUBJID. Bare name or string.
<code>threads</code>	Draw animated threads from selected rows to the groups they belong to. This is what makes the row-to-aggregate mapping visible. Set FALSE for a quieter figure.
<code>thread_cap</code>	Maximum threads drawn. Above this, threads are sampled evenly so the display stays legible.
<code>points</code>	Renderer for the row-level display: "auto" switches to canvas above <code>canvas_threshold</code> rows, "svg" and "canvas" force one.
<code>canvas_threshold</code>	Row count above which "auto" uses canvas.
<code>palette</code>	Named list of colours overriding the default light theme. Recognised names: ground, panel, data, select, zone, text, dim, rule, mute for marks outside the selection, and arms for a vector of one colour per treatment arm. Arm colours default to an Okabe-Ito derived set that survives colour blindness and greyscale printing, with a neutral first colour for placebo.

Value

An object of class `linkagg_spec`.

Examples

```
set.seed(1)
adsl <- data.frame(
  USUBJID = sprintf("01-%03d", 1:120),
  ARM      = rep(c("Placebo", "Drug A 50mg", "Drug A 100mg"), each = 40),
  ALT      = exp(rnorm(120)),
  TBILI     = exp(rnorm(120))
)
adsl$SOC <- replicate(120, sample(c("Hepatobiliary disorders",
                                   "Investigations",
                                   "Gastrointestinal disorders"),
                                   sample(0:3, 1)), simplify = FALSE)

linkagg(adsl, USUBJID) |>
  view_points(TBILI, ALT, log_x = TRUE, log_y = TRUE) |>
  viewBars(SOC, by = ARM) |>
  viewTable(cols = c("USUBJID", "ARM", "ALT", "TBILI"))
```

linkagg-shiny

Shiny bindings for linkagg

Description

The current selection is reported as `input$<outputId>_selected`, a character vector of key values, or `NULL` when nothing is selected.

Usage

```
linkaggOutput(outputId, width = "100%", height = "760px")
```

```
renderLinkagg(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

<code>outputId</code>	Output variable name.
<code>width, height</code>	Passed to the container.
<code>expr</code>	An expression producing a <code>linkagg_spec</code> or widget.
<code>env</code>	Environment in which to evaluate <code>expr</code> .
<code>quoted</code>	Is <code>expr</code> already quoted?

Value

`linkaggOutput()` returns a Shiny output element; `renderLinkagg()` returns a Shiny render function.

run_linkagg_app	<i>Run the Shiny demo app</i>
-----------------	-------------------------------

Description

Launches a small app showing the figure wired into Shiny, with the current selection read back in R from `input$fig_selected` and summarised below the figure. Use it as a working reference for the Shiny bindings, which are documented at [linkagg-shiny](#).

Usage

```
run_linkagg_app(...)
```

Arguments

... Passed to `shiny::runApp()`, for example `port` or `launch.browser`.

Value

Called for its side effect of running the app.

Examples

```
if (interactive()) {
  run_linkagg_app()
}
```

viewBars	<i>Add an aggregate bar display, optionally split by treatment arm</i>
----------	--

Description

One bar per group, each standing for many rows. When a selection is active each bar fills to the share of its own rows that are selected, which is the behaviour existing linked-brushing tools in R do not provide.

Usage

```
viewBars(
  spec,
  group,
  by = NULL,
  drill = NULL,
  group_levels = NULL,
  by_levels = NULL,
```

```

denominator = c("population", "count"),
population = NULL,
maxBars = 12L,
label = NULL
)

```

Arguments

spec	A linkagg_spec.
group	Grouping column. Either an ordinary column with one value per row, or a list-column where each element is a character vector, for rows in several groups at once, such as a subject with events in several system organ classes.
by	Optional column giving the treatment arm, one value per row.
drill	Optional finer terms to drill into, given coarse to fine. One column drills a single level, such as preferred term below system organ class; several give a hierarchy, drill = c(PT, LLT) taking system organ class to preferred term to lowest level term. Clicking a bar label steps down one level and the breadcrumb steps back up, to any depth. Every drill column must be shaped like group: if group is a list-column, each must be a list-column whose elements pair up positionally, so element j of drill[[i]] is the term for element j of group[[i]]. A length mismatch is an error naming the row, since a silent misalignment would file events under the wrong organ class.
group_levels	Optional character vector fixing the order of groups. Defaults to descending overall count.
by_levels	Optional character vector fixing the order of arms. Defaults to the factor levels, or sorted unique values.
denominator	"population" gives percentages of each arm's analysis population; "count" gives raw counts. Defaults to "population" when by is supplied and "count" otherwise.
population	Denominator source. NULL counts rows per arm in data, which is correct when data is the analysis population. Otherwise pass a named numeric vector such as c(Placebo = 80, "Drug A" = 82), or the full population data frame to count from.
maxBars	Most bars to draw at any one level, largest first. Drilling into a system organ class can turn up dozens of preferred terms, most of them seen in one or two subjects, and drawing all of them buries the terms worth reading. Terms beyond the cap are counted in a note under the display rather than dropped silently. Set Inf to draw everything.
label	Display title. Defaults to the column name.

Details

Supply by to split every group by treatment arm, which is how safety displays are actually read. With by set, bar length is the percentage of that arm's analysis population, matching the denominator convention of a standard adverse event summary. Without by, bar length is a raw count.

Value

The updated linkagg_spec.

view_hist	<i>Add a linked histogram</i>
-----------	-------------------------------

Description

Bins a continuous column and draws it as an aggregate display. This is the case crosstalk's documentation names as unsupported, since each bar stands for many rows. Here a selection fills each bar from the baseline up, in proportion to the rows of that bin which are selected.

Usage

```
view_hist(
  spec,
  x,
  bins = 24L,
  by = NULL,
  log = FALSE,
  denominator = c("count", "population"),
  population = NULL,
  label = NULL
)
```

Arguments

spec	A linkagg_spec.
x	Numeric column to bin. Bare name or string.
bins	Number of equal-width bins.
by	Optional treatment arm column, drawn as one series per arm.
log	Bin on the log10 scale, for skewed measures such as lab ratios. Non-positive values are dropped, and the count of dropped rows is checked rather than passed over silently.
denominator	"count" or "population", as in viewBars() .
population	Denominator source, as in viewBars() .
label	Display title. Defaults to the column name.

Value

The updated linkagg_spec.

view_points

*Add a row-level scatter display***Description**

One mark per row, and the display you brush to make a selection. For liver safety this is the eDISH shape: peak ALT against peak total bilirubin on log scales, with a reference region.

Usage

```
view_points(
  spec,
  x,
  y,
  log_x = FALSE,
  log_y = FALSE,
  x_lab = NULL,
  y_lab = NULL,
  zone = NULL,
  facet = NULL,
  facet_levels = NULL,
  facet_row = NULL,
  facet_row_levels = NULL
)
```

Arguments

spec	A linkagg_spec.
x, y	Numeric columns. Bare names or strings.
log_x, log_y	Use a log scale on that axis.
x_lab, y_lab	Axis labels. Default to the column names.
zone	Optional threshold region drawn behind the data, as a list with x and y numeric cut-points and an optional label. Marks above both cut-points fall inside the region.
facet	Optional column giving one small multiple per level, drawn on shared scales so panels are comparable. Brushing acts within one panel. With facet_row also set, this becomes the column variable of a grid.
facet_levels, facet_row_levels	Optional character vectors fixing the order of columns and of rows.
facet_row	Optional second column, laid out down the rows to give a full grid of panels: one column per level of facet, one row per level of facet_row, such as arm across and sex down. Scales stay shared across the whole grid, so every panel is comparable with every other, and a brush still selects only the rows in the one cell you dragged over.

Value

The updated linkagg_spec.

view_table

Add a row listing

Description

Shows every row, and filters to the selection when one is active.

Usage

```
view_table(spec, cols = NULL, labels = NULL, max_rows = 400L)
```

Arguments

spec	A linkagg_spec.
cols	Character vector of columns to show. Defaults to all columns that are not list-columns.
labels	Optional column headings, same length as cols.
max_rows	Rows rendered at once. The count shown is always the full selection size.

Value

The updated linkagg_spec.

view_volcano

Add a linked volcano plot of adverse event terms

Description

One point per term, positioned by how much the two arms differ. The x axis is the risk difference, the comparison arm's incidence minus the reference arm's, in percentage points. The y axis is $-\log_{10}(p)$ from Fisher's exact test on that term's two-by-two table. Terms far right are more frequent on the comparison arm, terms high up separate the arms most sharply.

Usage

```
view_volcano(
  spec,
  group,
  by,
  ref = NULL,
  comp = NULL,
  min_n = 2L,
  alpha = 0.05,
  label = NULL
)
```

Arguments

spec	A linkagg_spec.
group	Term column, such as preferred term. Either one value per row or a list-column, as in viewBars() .
by	Treatment arm column.
ref, comp	Reference and comparison arm. Default to the first and last levels of by. Subjects on any other arm take no part in the comparison.
min_n	Drop terms with fewer than this many subjects across the two arms. The count dropped is shown on the display.
alpha	Significance level for the reference line. Drawn as a guide to the eye, with no multiplicity adjustment implied.
label	Display title. Defaults to the column name.

Details

Each point stands for many subjects, so it is an aggregate mark in the sense this package is built around: with a selection active every point fills from the bottom in proportion to the share of its own subjects selected. Brushing the liver corner of an eDISH plot and reading the volcano therefore answers a question a static safety pack cannot: which adverse event signals are actually carried by those subjects.

Point area is proportional to the number of subjects contributing to the term. A volcano plot shows an effect estimate without showing its precision, which is its recognised weakness: a term seen in two subjects can sit as far out as one seen in fifty. Sizing by subject count keeps that visible, and `min_n` drops the sparsest terms while reporting how many were dropped rather than passing over them silently.

Value

The updated `linkagg_spec`.

Index

`as_linkagg_widget`, [2](#)
`as_linkagg_widget()`, [3](#)

`htmlwidgets::saveWidget()`, [2](#)

`linkagg`, [3](#)
`linkagg-shiny`, [4](#), [5](#)
`linkaggOutput(linkagg-shiny)`, [4](#)

`renderLinkagg(linkagg-shiny)`, [4](#)
`run_linkagg_app`, [5](#)

`shiny::runApp()`, [5](#)

`viewBars`, [5](#)
`viewBars()`, [3](#), [7](#), [10](#)
`viewHist`, [7](#)
`viewPoints`, [8](#)
`viewPoints()`, [3](#)
`viewTable`, [9](#)
`viewTable()`, [3](#)
`viewVolcano`, [9](#)