Package 'nivo.bubblechart'

January 19, 2026

Title Interactive Bubble Charts for 'Shiny' Using 'Nivo'

Version 0.1.1

Description Creates interactive bubble chart visualizations for 'Shiny' applications
using the 'Nivo' circle packing library. Provides an 'htmlwidgets' wrapper around
the 'Nivo' circle packing chart, enabling hierarchical data visualization with
customizable colors, labels, and interactive features including click and hover events.
For more information about 'Nivo', see <https://nivo.rocks/>.

License MIT + file LICENSE

URL https://github.com/DataRacerEdu/nivo.bubblechart

BugReports https://github.com/DataRacerEdu/nivo.bubblechart/issues

Encoding UTF-8

RoxygenNote 7.3.3

Config/Needs/website pkgdown

Imports htmltools, htmlwidgets, jsonlite, reactR

Suggests knitr, rmarkdown, shiny, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Anastasiia Kostiv [aut, cre]

Maintainer Anastasiia Kostiv <akostiv11@gmail.com>

Repository CRAN

Date/Publication 2026-01-19 18:40:02 UTC

Contents

---

bubblechart	*Create an Interactive Bubble Chart*

Description

Creates an interactive circle packing (bubble chart) visualization using the Nivo library. This function produces a hierarchical bubble chart that supports click and hover interactions in Shiny applications. Bubble size is determined by the `value` field in your data, and colors are fully customizable.

Usage

```
bubblechart(
  element_id,
  main_color,
  label_color,
  on_hover_title_color,
  data = NULL,
  width = NULL,
  isInteractive = TRUE,
  height = "400px",
  activeColor = "transparent",
  borderWidth = 3
)
```

Arguments

element_id	Character string specifying the Shiny input ID for click events. Click events will be accessible in Shiny as input$<element_id>_clicked. For example, if element_id = "my_chart", click events are available at input$my_chart_clicked.
main_color	Character string specifying the default color for bubbles (hex color code, e.g., "#ff5f56" for red, "#3498db" for blue). This color is used as the base color for the chart border and as a fallback.
label_color	Character string specifying the default label text color (hex color code). Use high-contrast colors for better readability (e.g., "#ffffff" for white text on dark bubbles, "#000000" for black text on light bubbles).
on_hover_title_color	
	Character string specifying the label color when hovering over a bubble (hex color code). This color is applied to the label text when the user hovers their mouse over a bubble, providing visual feedback.
data	A nested list representing hierarchical data. Must have a "name" field at the root and a "children" field containing child nodes. Each child should have:

- `name`: Display name for the bubble (character)
- `value`: Numeric value determining bubble size (must be positive)
- `color`: Hex color code for the bubble fill (character, e.g., "#3498db")
- `labelColor`: Hex color code for the label text (character, e.g., "#ffffff")

	Use [prepare_bubble_data](#) to convert data frames to this format.
width	The width of the widget. Can be a valid CSS unit (e.g., "100 percent", "400px", "50vw") or NULL for automatic sizing. NULL is recommended for responsive layouts.
isInteractive	Logical. If TRUE (default), enables click and hover interactions. Set to FALSE for static charts in reports or presentations where interactivity is not needed.
height	The height of the widget. Can be a valid CSS unit (e.g., "400px", "600px", "50vh") or NULL. Default is "400px". Adjust based on your layout needs and the amount of data being displayed.
activeColor	Character string specifying the color of selected/active bubbles (hex color code). When a user clicks a bubble, it changes to this color. Default is "transparent", which makes the selected bubble see-through.
borderWidth	Numeric value specifying the border width of bubbles in pixels. Default is 3. Use larger values (e.g., 5) for prominent borders, smaller values (e.g., 1) for subtle borders, or 0 for no borders.

Details

The bubble chart (circle packing diagram) is a powerful way to visualize hierarchical data where the size of each circle represents a quantitative value. This implementation uses the Nivo visualization library to create responsive, interactive charts that work seamlessly in both static R documents and dynamic Shiny applications.

When used in Shiny, the chart generates click events that can be captured via input$<element_id>_clicked, where <element_id> is the ID you specify. Clicking a bubble returns its name, and clicking it again returns "DESELECT_EVENT".

Value

An htmlwidget object of class c("bubblechart", "htmlwidget") that can be:

- Rendered directly in R console or RStudio Viewer
- Included in R Markdown documents
- Used in Shiny applications with [bubblechartOutput](#) and [renderBubblechart](#)
- Saved to HTML with htmlwidgets::saveWidget()

See Also

[prepare_bubble_data](#) for converting data frames to the required format, [bubblechartOutput](#) and [renderBubblechart](#) for Shiny integration.

Examples

```
if (interactive()) {
library(nivo.bubblechart)

# Example 1: Basic bubble chart with manual data
sample_data <- list(
```

```
    name = "root",
    children = list(
      list(name = "Category A", color = "#ff5f56", labelColor = "#ffffff", value = 100),
      list(name = "Category B", color = "#ffbd2e", labelColor = "#ffffff", value = 200),
      list(name = "Category C", color = "#27c93f", labelColor = "#ffffff", value = 150)
    )
)

bubblechart(
  element_id = "my_chart",
  main_color = "#ff5f56",
  label_color = "#ffffff",
  on_hover_title_color = "#000000",
  data = sample_data,
  height = "500px"
)

# Example 2: Using prepare_bubble_data for data frame conversion
df <- data.frame(
  product = c("Laptop", "Phone", "Tablet"),
  sales = c(45000, 38000, 22000),
  category_color = c("#3498db", "#e74c3c", "#f39c12"),
  text_color = rep("#ffffff", 3)
)

chart_data <- prepare_bubble_data(
  df,
  name_col = "product",
  value_col = "sales",
  color_col = "category_color",
  label_color_col = "text_color"
)

bubblechart(
  element_id = "sales_chart",
  main_color = "#3498db",
  label_color = "#ffffff",
  on_hover_title_color = "#2c3e50",
  data = chart_data,
  height = "600px"
)

# Example 3: Customized appearance
bubblechart(
  element_id = "custom_chart",
  main_color = "#2c3e50",
  label_color = "#ecf0f1",
  on_hover_title_color = "#f39c12",
  activeColor = "#e74c3c",  # Red when selected
  borderWidth = 5,          # Thicker borders
  data = sample_data,
  isInteractive = TRUE
)
```

```
}
```

---

bubblechart-shiny   *Shiny bindings for bubblechart*

Description

Output and render functions for using bubblechart within Shiny applications and interactive Rmd documents.

Usage

```
bubblechartOutput(outputId, width = "100%", height = "400px")

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

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100 percent', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a bubblechart
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

Value

For bubblechartOutput, a Shiny UI element that can be included in a Shiny UI definition. For renderBubblechart, a Shiny render function that can be assigned to an output slot in a Shiny server function.

---

prepare_bubble_data   *Prepare Data Frame for Bubble Chart*

Description

Converts a data frame into the hierarchical list structure required by bubblechart. This helper function bridges the gap between tabular data (data frames) and the nested format expected by the Nivo circle packing visualization, making it easy to create bubble charts from your existing data.

Usage

```
prepare_bubble_data(
  df,
  name_col = "name",
  value_col = "value",
  color_col = "color",
  label_color_col = "labelColor",
  root_name = "root",
  default_color = "#ff5f56",
  default_label_color = "#ffffff"
)
```

Arguments

df	A data frame containing your data. Must have at least two columns: one for names and one for values. Color columns are optional.
name_col	Character string specifying the column name for bubble names. This will be the text displayed on each bubble. Default is "name".
value_col	Character string specifying the column name for bubble values. These values determine the size of each bubble - larger values create larger bubbles. Values must be positive numbers. Default is "value".
color_col	Character string specifying the column name for bubble colors, or NULL. If provided, this column should contain hex color codes (e.g., "#3498db"). If NULL or if the column doesn't exist, `default_color` is used. Default is "color".
label_color_col	
	Character string specifying the column name for label colors, or NULL. If provided, this column should contain hex color codes for the text labels. If NULL or if the column doesn't exist, `default_label_color` is used. Default is "labelColor".
root_name	Character string for the root node name. This is used internally by the visualization and typically doesn't need to be changed. Default is "root".
default_color	Character string for default bubble color (hex code). Used when `color_col` is NULL or when a row has a missing color value. Default is "#ff5f56" (a warm red).
default_label_color	
	Character string for default label text color (hex code). Used when `label_color_col` is NULL or when a row has a missing label color. Default is "#ffffff" (white). Use "#000000" for black text on light backgrounds.

Details

Most R users work with tabular data in data frames, but bubble charts require hierarchical (nested list) structures. This function handles the conversion automatically, allowing you to specify which columns contain the bubble names, values, and colors.

The function is particularly useful when:

- You have data in a data frame and want to visualize it as a bubble chart

- Your data comes from a database query, CSV file, or API that returns tabular data
- You want to apply conditional coloring based on data values
- You need to dynamically generate chart data in Shiny applications

If your data frame doesn't have color columns, the function will use the default_color and default_label_color parameters for all bubbles. You can also mix and match - for example, providing a color column but letting the function use the default label color.

Value

A named list with the hierarchical structure required by bubblechart. The structure is:

```
list(
  name = <root_name>,
  children = list(
    list(name = ..., value = ..., color = ..., labelColor = ...),
    list(name = ..., value = ..., color = ..., labelColor = ...),
    ...
  )
)
```

This output can be passed directly to the data parameter of bubblechart.

See Also

bubblechart for creating the visualization.

Examples

```
# Example 1: Basic conversion with all columns present
df <- data.frame(
  name = c("Apple", "Banana", "Cherry"),
  value = c(100, 200, 150),
  color = c("#ff5f56", "#ffbd2e", "#27c93f"),
  labelColor = c("#ffffff", "#ffffff", "#ffffff")
)

bubble_data <- prepare_bubble_data(df)
str(bubble_data)

# Example 2: Using custom column names
sales_df <- data.frame(
  product = c("Laptop", "Phone", "Tablet"),
  revenue = c(45000, 38000, 22000),
  brand_color = c("#3498db", "#e74c3c", "#f39c12")
)

chart_data <- prepare_bubble_data(
  sales_df,
  name_col = "product",
```

```r
    value_col = "revenue",
    color_col = "brand_color",
    default_label_color = "#ffffff"
)

# Example 3: Using default colors (no color columns)
simple_df <- data.frame(
  category = c("A", "B", "C", "D"),
  amount = c(50, 120, 80, 200)
)

bubble_data <- prepare_bubble_data(
  simple_df,
  name_col = "category",
  value_col = "amount",
  default_color = "#9b59b6",        # Purple
  default_label_color = "#ffffff"   # White
)

# Example 4: Conditional coloring based on values
data_df <- data.frame(
  item = c("Small", "Medium", "Large", "X-Large"),
  size = c(50, 150, 300, 500)
)

# Add colors based on size
data_df$color <- ifelse(data_df$size > 250, "#27ae60",  # Green for large
                 ifelse(data_df$size > 100, "#f39c12",  # Orange for medium
                        "#e74c3c"))                      # Red for small
data_df$labelColor <- "#ffffff"

bubble_data <- prepare_bubble_data(data_df, name_col = "item", value_col = "size")

# Example 5: Use with bubblechart
if (interactive()) {
library(nivo.bubblechart)

df <- data.frame(
  fruit = c("Apples", "Oranges", "Bananas", "Grapes"),
  sales = c(450, 280, 320, 150)
)

chart_data <- prepare_bubble_data(
  df,
  name_col = "fruit",
  value_col = "sales",
  default_color = "#3498db",
  default_label_color = "#ffffff"
)

bubblechart(
  element_id = "fruit_sales",
  main_color = "#3498db",
```

```
        label_color = "#ffffff",
        on_hover_title_color = "#2c3e50",
        data = chart_data,
        height = "500px"
    )
}
```

Index