Package 'lineupjs'

July 22, 2025

Type Package

Title 'HTMLWidget' Wrapper of 'LineUp' for Visual Analysis of Multi-Attribute Rankings

Description 'LineUp' is an interactive technique designed to create, visualize and explore rankings of items based on a set of heterogeneous attributes.
This is a 'htmlwidget' wrapper around the JavaScript library 'LineUp.js'.
It is designed to be used in 'R Shiny' apps and 'R Markddown' files.
Due to an outdated 'webkit' version of 'RStudio' it won't work in the integrated viewer.

Version 4.6.0

Date 2022-08-10

Maintainer Samuel Gratzl <sam@sgratzl.com>

URL https://github.com/lineupjs/lineup_htmlwidget/

BugReports https://github.com/lineupjs/lineup_htmlwidget/issues

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

Imports htmlwidgets

Suggests crosstalk, knitr, rmarkdown, testthat, lintr, remotes, styler, shiny

RoxygenNote 7.2.0

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Samuel Gratzl [aut, cre]

Repository CRAN

Date/Publication 2022-08-19 12:20:08 UTC

Contents

`buildLineUp` *factory for LineUp HTMLWidget based on a LineUpBuilder*

Description

factory for LineUp HTMLWidget based on a LineUpBuilder

Usage

```
buildLineUp(
  x,
  width = "100%",
  height = NULL,
  elementId = NULL,
  dependencies = .crosstalkLineUpLibs()
)
```

Arguments

`x`	LineUpBuilder object
`width`	width of the element
`height`	height of the element
`elementId`	unique element id
`dependencies`	include crosstalk dependencies

Value

lineup html widget

Examples

```
## Not run:
lineupBuilder(iris) |> buildLineUp()

## End(Not run)
```

buildTaggle *factory for LineUp HTMLWidget based on a LineUpBuilder*

Description

factory for LineUp HTMLWidget based on a LineUpBuilder

Usage

```
buildTaggle(
  x,
  width = "100%",
  height = NULL,
  elementId = NULL,
  dependencies = .crosstalkLineUpLibs()
)
```

Arguments

x	LineUpBuilder object
width	width of the element
height	height of the element
elementId	unique element id
dependencies	include crosstalk dependencies

Value

taggle html widget

Examples

```
## Not run:
lineupBuilder(iris) |> buildTaggle()

## End(Not run)
```

`lineup` *lineup - factory for LineUp HTMLWidget*

Description

lineup - factory for LineUp HTMLWidget

Usage

```
lineup(
  data,
  width = "100%",
  height = NULL,
  elementId = NULL,
  options = c(.lineupDefaultOptions),
  ranking = NULL,
  dependencies = .crosstalkLineUpLibs(),
  ...
)
```

Arguments

`data`	data frame like object i.e. also crosstalk shared data frame
`width`	width of the element
`height`	height of the element
`elementId`	unique element id
`options`	LineUp options
`ranking`	ranking definition created using `lineupRanking`
`dependencies`	include crosstalk dependencies
`...`	additional ranking definitions like 'ranking1=...' due to restrictions in converting parameters

Value

lineup html widget

LineUp options

filterGlobally whether filter within one ranking applies to all rankings (default: TRUE)

singleSelection restrict to single item selection (default: FALSE

noCriteriaLimits allow more than one sort and grouping criteria (default: FALSE)

animated use animated transitions (default: TRUE)

sidePanel show side panel (TRUE, FALSE, 'collapsed') (default: 'collapsed')

hierarchyIndicator show sorting and grouping hierarchy indicator (TRUE, FALSE) (default: TRUE)

labelRotation how many degrees should a label be rotated in case of narrow columns (default: 0)

summaryHeader show summary histograms in the header (default: TRUE)

overviewMode show overview mode in Taggle by default (default: FALSE)

expandLineOnHover expand to full row height on mouse over (default: FALSE)

defaultSlopeGraphMode default slope graph mode: item,band (default: 'item')

ignoreUnsupportedBrowser ignore unsupported browser detection at own risk (default: FALSE)

rowHeight height of a row in pixel (default: 18)

rowPadding padding between two rows in pixel (default: 2)

groupHeight height of an aggregated group in pixel (default: 40)

groupPadding padding between two groups in pixel (default: 5)

Examples

```
## Not run:
lineup(iris)

## End(Not run)
```

`lineupBuilder` *lineup builder pattern function*

Description

lineup builder pattern function

Usage

```
lineupBuilder(data, options = c(.lineupDefaultOptions), ranking = NULL, ...)
```

Arguments

`data`	data frame like object i.e. also crosstalk shared data frame
`options`	LineUp options
`ranking`	ranking definition created using `lineupRanking`
`...`	additional ranking definitions like 'ranking1=...' due to restrictions in converting parameters

Value

lineup builder object

LineUp options

filterGlobally whether filter within one ranking applies to all rankings (default: TRUE)

singleSelection restrict to single item selection (default: FALSE

noCriteriaLimits allow more than one sort and grouping criteria (default: FALSE)

animated use animated transitions (default: TRUE)

sidePanel show side panel (TRUE, FALSE, 'collapsed') (default: 'collapsed')

hierarchyIndicator show sorting and grouping hierarchy indicator (TRUE, FALSE) (default: TRUE)

labelRotation how many degrees should a label be rotated in case of narrow columns (default: 0)

summaryHeader show summary histograms in the header (default: TRUE)

overviewMode show overview mode in Taggle by default (default: FALSE)

expandLineOnHover expand to full row height on mouse over (default: FALSE)

defaultSlopeGraphMode default slope graph mode: item,band (default: 'item')

ignoreUnsupportedBrowser ignore unsupported browser detection at own risk (default: FALSE)

rowHeight height of a row in pixel (default: 18)

rowPadding padding between two rows in pixel (default: 2)

groupHeight height of an aggregated group in pixel (default: 40)

groupPadding padding between two groups in pixel (default: 5)

Examples

```
## Not run:
lineupBuilder(iris) |> buildLineUp()

## End(Not run)
```

`lineupjs` *LineUpjs module*

Description

a htmlwidget wrapper around LineUpJS (https://lineup.js.org)

`lineupOutput` *Shiny bindings for lineup*

Description

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

Usage

```
lineupOutput(outputId, width = "100%", height = "800px")

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

Arguments

`outputId`	output variable to read from
`width, height`	Must be a valid CSS unit (like `'100%'`, `'800px'`, `'auto'`) or a number, which will be coerced to a string and have `'px'` appended.
`expr`	An expression that generates a taggle
`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

An output or render function that enables the use of the widget within Shiny applications.

Examples

```
# !formatR
library(shiny)
app <- shinyApp(
  ui = fluidPage(lineupOutput("lineup")),
  server = function(input, output) {
    lineup <- lineupBuilder(iris) |> buildLineUp()
    output$lineup <- renderLineup(lineup)
  }
)


if (interactive()) app
```

`lineupRanking` *helper function for creating a LineUp ranking definition as used by* `lineup`

Description

helper function for creating a LineUp ranking definition as used by `lineup`

Usage

```
lineupRanking(columns = c("_*", "*"), sortBy = c(), groupBy = c(), ...)
```

Arguments

`columns`	list of columns shown in this ranking, besides *column names of the given data frame* following special columns are available
`sortBy`	list of columns to sort this ranking by, grammar: `"<column name>[:desc]"`
`groupBy`	list of columns to group this ranking by
`...`	additional ranking combination definitions as lists (`list(type = 'min', columns = c('a', 'b'), label = NULL)`), possible types

Value

a configured lineup ranking config

Special columns

'* ' include all data frame columns

'_* ' add multiple support columns (_aggregate, _rank, _selection)

'_aggregate' add a column for collapsing groups

'_rank' add a column for showing the rank of the item

'_selection' add a column with checkboxes for selecting items

'_group' add a column showing the current grouping title

'$data.frame column$' add the specific column

'$def column$' add defined column given as additional parameter to this function, see below

Ranking definition types

weightedSum a weighted sum of multiple numeric columns, extras `list(weights = c(0.4, 0.6))`

min minimum of multiple numeric columns

max maximum of multiple numeric columns

mean mean of multiple numeric columns

median median of multiple numeric columns

nested group multiple columns

script scripted (JS code) combination of multiple numeric columns, extras `list(code = '...')`

impose color a numerical column (column) with the color of a categorical column (categoricalColumn), changed `list(column = 'a', categoricalColumn = 'b')`

Examples

```
lineupRanking(columns = c("*"))
lineupRanking(columns = c("*"), sortBy = c("hp"))
lineupRanking(
  columns = c("*", "sum"),
  sum = list(type = "weightedSum", columns = c("hp", "wt"), weights = c(0.7, 0.3))
)
```

`taggle` *taggle - factory for Taggle HTMLWidget*

Description

taggle - factory for Taggle HTMLWidget

Usage

```
taggle(
  data,
  width = "100%",
  height = NULL,
  elementId = NULL,
  options = c(.lineupDefaultOptions),
  ranking = NULL,
  dependencies = .crosstalkLineUpLibs(),
  ...
)
```

Arguments

`data`	data frame like object i.e. also crosstalk shared data frame
`width`	width of the element
`height`	height of the element
`elementId`	unique element id
`options`	LineUp options
`ranking`	ranking definition created using `lineupRanking`
`dependencies`	include crosstalk dependencies
`...`	additional ranking definitions like 'ranking1=...' due to restrictions in converting parameters

Value

taggle html widget

LineUp options

filterGlobally whether filter within one ranking applies to all rankings (default: TRUE)

singleSelection restrict to single item selection (default: FALSE

noCriteriaLimits allow more than one sort and grouping criteria (default: FALSE)

animated use animated transitions (default: TRUE)

sidePanel show side panel (TRUE, FALSE, 'collapsed') (default: 'collapsed')

hierarchyIndicator show sorting and grouping hierarchy indicator (TRUE, FALSE) (default: TRUE)

labelRotation how many degrees should a label be rotated in case of narrow columns (default: 0)

summaryHeader show summary histograms in the header (default: TRUE)

overviewMode show overview mode in Taggle by default (default: FALSE)

expandLineOnHover expand to full row height on mouse over (default: FALSE)

defaultSlopeGraphMode default slope graph mode: item,band (default: 'item')

ignoreUnsupportedBrowser ignore unsupported browser detection at own risk (default: FALSE)

rowHeight height of a row in pixel (default: 18)

rowPadding padding between two rows in pixel (default: 2)

groupHeight height of an aggregated group in pixel (default: 40)

groupPadding padding between two groups in pixel (default: 5)

Examples

```
## Not run:
taggle(iris)

## End(Not run)
```

---

taggleOutput *Shiny bindings for taggle*

Description

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

Usage

```
taggleOutput(outputId, width = "100%", height = "800px")

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

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '800px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a taggle
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

An output or render function that enables the use of the widget within Shiny applications.

Examples

```
# !formatR
library(shiny)
app <- shinyApp(
  ui = fluidPage(taggleOutput("taggle")),
  server = function(input, output) {
    taggle <- lineupBuilder(iris) |> buildTaggle()
    output$taggle <- renderTaggle(taggle)
  }
)


if (interactive()) app
```

Index