

Package ‘rpivotTable’

August 24, 2026

Title Build Powerful Pivot Tables and Dynamically Slice & Dice your Data

Date 2026-08-06

Version 0.4.0

Maintainer Lily Clements <lily@idems.international>

Description Build powerful pivot tables (aka Pivot Grid, Pivot Chart, Cross-Tab) and dynamically slice & dice / drag 'n' drop your data. 'rpivotTable' is a wrapper of 'pivottable', a powerful open-source Pivot Table library implemented in 'JavaScript' by Nicolas Kruchten. Aligned to 'pivottable' v2.19.0.

Depends R (>= 3.2.0)

Imports htmlwidgets (>= 0.5)

Suggests data.table, devtools, dplyr, htmltools, knitr, rmarkdown, testthat, tibble

Enhances shiny (>= 0.12)

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/smartinsightsfromdata>,
<https://github.com/idemsinternational>

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Author Enzo Martoglio [aut],
Nicolas Kruchten [ctb, cph],
Nagarajan Chinnasamy [ctb, cph],
Kenton Russell [ctb],
Lily Clements [cre]

Repository CRAN

Date/Publication 2026-08-24 18:50:10 UTC

Contents

change_locale	2
renderRpivotTable	3
rpivotTable	3
rpivotTableOutput	6
Index	7

change_locale	<i>Localization</i>
---------------	---------------------

Description

Change localization for rpivotTable to support non-English language. For more discussion, see [pivottable Localization](#).

Usage

```
change_locale(pivottable = NULL, locale = NULL)
```

Arguments

- | | |
|------------|---|
| pivottable | rpivotTable object to modify localization |
| locale | character of locale to use. Valid locale options are <ul style="list-style-type: none">• cs• da• de• es• fr• it• nl• pl• pt• ru• sq• tr• zh |

Value

rpivotTable object

Examples

```
library(rpivotTable)
change_locale(rpivotTable(mtcars),"pt")
```

renderRpivotTable	<i>Widget render function for use in Shiny</i>
-------------------	--

Description

Widget render function for use in Shiny

Usage

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

Arguments

expr	rpivotTable expression
env	environment
quoted	logical, default = FALSE

Examples

```
# A simple example - this goes in the server part of a shiny application

# output$pivot <- renderRpivotTable({
#   rpivotTable(data = canadianElections , rows = c( "Province"),cols="Party",
#   vals = "votes", aggregatorName = "Sum", rendererName = "Table",
#   width="100%", height="500px")
# })
```

rpivotTable	<i>pivottable.js in R</i>
-------------	---------------------------

Description

Use pivottable.js in R with the power and convenience of a htmlwidget.

Usage

```
rpivotTable(
  data,
  rows = NULL,
  cols = NULL,
  aggregatorName = NULL,
  vals = NULL,
  rendererName = NULL,
```

```

    sorter = NULL,
    exclusions = NULL,
    inclusions = NULL,
    locale = "en",
    subtotals = FALSE,
    ...,
    width = 800,
    height = 600,
    elementId = NULL
  )

```

Arguments

data	data.frame or data.table (R>=1.9.6 for safety) with data to use in the pivot table
rows	String name of the column in the data.frame to prepopulate the rows of the pivot table.
cols	String name of the column in the data.frame to prepopulate the columns of the pivot table.
aggregatorName	String name of the rpivottable.js aggregator to prepopulate the pivot table.
vals	String name of the column in the data.frame to use with aggregatorName. Must be additive (i.e a number).
rendererName	List name of the renderer selected, e.g. Table, Heatmap, Treemap etc.
sorter	String name this allows to implement a javascript function to specify the ad hoc sorting of certain values. See vignette for an example. It is especially useful with time divisions like days of the week or months of the year (where the alphabetical order does not work).
exclusions	String this optional parameter allows to filter the members of a particular dimension "by exclusion". Using the 'Titanic' example, to display only the "1st", "2nd" and "3rd" members in the "Class" dimension, it is convenient to filter by exclusion using 'exclusions = list(Class = list("Crew"))'. Please note that this only pre-selects the visible filter(s) on the pivot table: the other dimension members are still available for selection if needed.
inclusions	List this optional parameter allows to filter the members of a particular dimension "by inclusion". Using the 'Titanic' example, to display only the "Crew" member in the "Class" dimension, it is convenient to filter by inclusion using 'inclusions = list(Class = list("Crew"))'. Please note that this only pre-selects the visible filter(s) on the pivot table: the other dimension members are still available for selection if needed.
locale	character of locale to use. Valid locale options are • cs • da • de • es • fr • it

	• nl • pl • pt • ru • sq • tr • zh
subtotals	Logical this optional parameter allows use of the pivottable subtotal plugin . Using this parameter will set all renderer names to the english locale.
...	list other parameters that can be passed to pivotUI. See Nicolas's Wiki for more details. If sorters is not supplied here, it is generated automatically from the factor levels of any factor columns in data, so that rows/columns for those variables sort in factor-level order rather than alphabetically. Pass your own sorters to override this default. A further example of parameter is onRefresh. This parameters (shiny-only) introduces a JS function that allows to get back server side the list of parameters selected by the user. An example is: onRefresh=htmlwidgets::JS("function(config) \{ Shiny.onInputChange('myPivotData', config); \}") This setting makes available server-side a function input\$myPivotData that gives back a list (of lists) with all the slice & dice parameters offered by pivottable. See the example onRefresh-shiny.R for an example of how to use this feature. Example of usage could be: These parameters could be saved and re-resent to the user. Alternative they could be used to subset the data item for saving as csv.
width	width parameter
height	height parameter
elementId	String valid CSS selector id for the rpivotTable container.

Examples

```
# use Titanic dataset provided in base R - simple creation with just data

rpivotTable( Titanic )

# prepopulate multiple columns and multiple rows

rpivotTable( Titanic, rows = c("Class","Sex"), cols = c("Age","Survived" ) )

# A more complete example:

rpivotTable(
  Titanic,
  rows = "Survived",
  cols = c("Class","Sex"),
  aggregatorName = "Sum as Fraction of Columns",
  vals = "Freq",
  rendererName = "Table Barchart"
)
```

```
# An example with inclusions and exclusions filters:
```

```
rpivotTable(  
  Titanic,  
  rows = "Survived",  
  cols = c("Class", "Sex"),  
  aggregatorName = "Sum as Fraction of Columns",  
  inclusions = list( Survived = list("Yes")),  
  exclusions = list( Class = list( "Crew")),  
  vals = "Freq",  
  rendererName = "Table Barchart"  
)
```

rpivotTableOutput	<i>Widget output function for use in Shiny</i>
-------------------	--

Description

Widget output function for use in Shiny

Usage

```
rpivotTableOutput(outputId, width = "100%", height = "500px")
```

Arguments

outputId	Shiny output ID
width	width default '100%'
height	height default '500px'

Examples

```
# A simple example - this goes in the ui part of a shiny application  
  
# rpivotTableOutput("pivot")
```

Index

`change_locale`, [2](#)

`renderRpivotTable`, [3](#)

`rpivotTable`, [3](#)

`rpivotTableOutput`, [6](#)