Package 'GWalkR'

June 10, 2025

Title Interactive Exploratory Data Analysis Tool

Version 0.2.1

Maintainer Yue Yu <yue.yu@connect.ust.hk>

Description Simplify your R data analysis and data visualization workflow by turning your data frame into an interactive 'Tableau'-like interface, leveraging the 'graphic-walker' JavaScript library and the 'htmlwidgets' package.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

URL https://github.com/Kanaries/GWalkR/

BugReports https://github.com/Kanaries/GWalkR/issues

Imports htmlwidgets, jsonlite, openssl, shiny, shinycssloaders, DBI

Suggests duckdb

NeedsCompilation no

Author Yue Yu [aut, cre] (ORCID: <https://orcid.org/0000-0002-9302-0793>), Kanaries Data Inc. [cph, fnd]

Repository CRAN

Date/Publication 2025-06-10 09:50:02 UTC

Contents

gwalkr *Create GWalkR Interface in "Viewer"*

Description

Use this function to create a GWalkR interface from a given data frame in your "Viewer" window, and start your data exploration! Please make sure the width and the height of your "Viewer" window are large enough.

Usage

```
gwalkr(
  data,
  lang = "en",
  dark = "light",
  columnSpecs = list(),
  visConfig = NULL,
  visConfigFile = NULL,
  toolbarExclude = list(),
  kernelComputation = FALSE
)
```

Arguments

data	A data frame to be visualized in the GWalkR. The data frame should not be empty.
lang	A character string specifying the language for the widget. Possible values are "en" (default), "ja", "zh".
dark	A character string specifying the dark mode preference. Possible values are "light" (default), "dark", "media".
columnSpecs	An optional list of lists to manually specify the types of some columns in the data frame. Each top level element in the list corresponds to a column, and the list assigned to each column should have two elements: `analyticalType` and `semanticType`. `analyticalType` can only be one of "measure" or "dimension". `semanticType` can only be one of "quantitative", "temporal", "nominal" or "ordinal". For example: `list( "gender" = list(analyticalType = "dimension", semanticType = "nominal"), "age" = list(analyticalType = "measure", semanticType = "quantitative") )`
visConfig	An optional config string to reproduce your chart. You can copy the string by clicking "export config" button on the GWalkR interface.
visConfigFile	An optional config file path to reproduce your chart. You can download the file by clicking "export config" button then "download" button on the GWalkR interface.
toolbarExclude	An optional list of strings to exclude the tools from toolbar UI. However, Kanaries brand info is not allowed to be removed or changed unless you are granted with special permission.

kernelComputation
: An optional boolean to enable the kernel mode computation which is much more efficient. Requires duckdb package installed. Default is FALSE.

Value

An htmlwidget object that can be rendered in R environments

Examples

```
data(mtcars)
gwalkr(mtcars)
```

---

gwalkr-shiny *Shiny bindings for gwalkr*

Description

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

Usage

```
gwalkrOutput(outputId, width = "100%", height = "100%")

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

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a gwalkr
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

- gwalkrOutput: A shinyWidgetOutput object for the root HTML element.
- renderGwalkr: A server-side function to help Shiny display the GWalkR visualization.

Examples

```
# !formatR
library(GWalkR)
library(shiny)
data(mtcars)
app <- shinyApp(
  ui = fluidPage(
    titlePanel("Explore the data here: "),
    gwalkrOutput("mygraph")
  ),
  server = function(input, output, session) {
    output$mygraph = renderGwalkr(
      gwalkr(mtcars)
    )
  }
)
if (interactive()) app
```

Index