Package ‘loon.shiny’

June 10, 2021

Type Package

Title Automatically Create a 'Shiny' App Based on Interactive 'Loon' Widgets

Version 1.0.0

Date 2021-05-10

Description Package 'shiny' provides interactive web applications in R. Package 'loon' is an interactive toolkit engaged in open-ended, creative and unscripted data exploration. The 'loon.shiny' package can take 'loon' widgets and display a selfsame 'shiny' app.

License GPL-2

Depends R (>= 3.4.0), loon (>= 1.3.0), tcltk, methods, shiny

Imports stats, grDevices, grid, gridExtra, loon.ggplot (>= 1.1.0), colourpicker, base64enc

Suggests magrittr, dplyr, ggmulti, tools, testthat, knitr, rmarkdown, tidyverse, covr, png

RoxygenNote 7.1.1

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Author Zehao Xu [aut, cre],
R. Wayne Oldford [aut]

Maintainer Zehao Xu <z267xu@uwaterloo.ca>

Repository CRAN

Date/Publication 2021-06-10 16:30:06 UTC

R topics documented:

loon.shiny ........................................................... 2

Index .............................. 5
**Description**

Interactive loon widgets displayed in a shiny app

**Usage**

```r
loon.shiny(
  widgets,
  selectBy = c("byDefault", "brushing", "sweeping"),
  showWorldView = TRUE,
  plotRegionWidth = "500px",
  plotRegionHeight = "500px",
  plotRegionBackground = "gray92",
  layoutMatrix = NULL,
  nrow = NULL,
  ncol = NULL,
  widths = NULL,
  heights = NULL,
  displayedPanel = ",",
  colorList = loon::l_getColorList(),
  inspectorLocation = c("auto", "auto", "60px", "20px"),
  inspectorWidth = "350px",
  inspectorHeight = "auto",
  options = list(),
  ...
)
```

**Arguments**

- `widgets`: A loon widget or a list of loon widgets.
- `selectBy`: The way to brush, can be ‘brushing’ (keep the brush whenever the plot is updated), ‘sweeping’ (clear the brush whenever the plot is updated) or ‘byDefault’ (determined by loon widget ‘selectBy’, if ‘selectBy’ is sweeping, then `selectBy` will be assigned to ‘sweeping’, vice versa)
- `showWorldView`: Logical; whether to show the world view.
- `plotRegionWidth`: Plot region width. Must be a valid CSS unit (like ’100’ which will be coerced to a string and have ’px’ appended. More details, check `plotOutput`.
- `plotRegionHeight`: Plot region height.
- `plotRegionBackground`: Plot region background color
**loon.shiny**

layoutMatrix  
Optional layout matrix to place loon widgets. See layout_matrix in grid.arrange.

nrow  
Number of rows, see grid.arrange.

ncol  
Number of columns, see grid.arrange.

widths  
A unit vector giving the width of each plot.

heights  
A unit vector giving the height of each plot.

displayedPanel  
A string vector. The default is an empty string "" so that none inspector components (Plot, Linking, Select, etc) are open automatically. The available strings are c("Plot","Select","Linking","Modify","Layer","Glyph")

colorList  
A list of colors displayed on modify panel.

inspectorLocation  
A length four vector representing the distance between the bottom, left, top and right of the inspector panel and the bottom, left, top and right of the page or parent container.

inspectorWidth  
Width of the inspector panel.

inspectorHeight  
Height of the inspector panel.

options  
shinyApp argument that should be passed to the runApp call, see shinyApp.

...  
Named arguments to modify shiny app.

Details

- Useful hints for a loon.shiny app
  - The inspector can be switched either by “toggling tabpanel” in the bar menu or the last mouse gesture input (<double-click>) on the plot region
  - To downlight the selected elements, one has to double click on the plot region
  - In loon, holding down the <shift> key while pressing the left button keeps the current selection states. In loon.shiny app, <shift> key is replaced by a 'sticky' radiobutton. If the 'sticky' mode is on, while sweeping, current selection states remain; else new selection will eliminate the previous selection states.

- Useful hints for a loon.shiny markdown file
  - Based on our experience, setting ‘out.width’ or ‘out.height’ (try "10px") in the chunk could give a better layout
  - To modify the app size, set ‘options = list(height = **, width = **)’ in loon.shiny()

Value

A shiny.appobj object that represents the loon.shiny app. Printing the object or passing it to runApp will run the app.

Examples

```r
## Only run this example in interactive R sessions
if(interactive()) {
  ############### Basic ###############
  p <- l_plot(iris,
```
loon.shiny(p)

#################### Link multiple plots ####################
p1 <- l_plot(iris,
    linkingGroup = 'iris',
    showLabels = FALSE)
p2 <- l_hist(iris$Sepal.Length,
    linkingGroup = 'iris',
    showLabels = FALSE,
    showStackedColors = TRUE)
p3 <- l_hist(iris$Sepal.Width, linkingGroup = 'iris',
    color = iris$Species, sync = 'push',
    showLabels = FALSE, swapAxes = TRUE,
    showStackedColors = TRUE)
loon.shiny(list(p1, p2, p3),
    layoutMatrix = matrix(c(2,NA,1,3),
    nrow = 2, byrow = TRUE))
}

## Not run:
if (requireNamespace('loon.ggplot', quietly = TRUE)) {
p <- ggplot(mpg, aes(displ, hwy)) +
    geom_point(data = transform(mpg, class = NULL), colour = 'grey85') +
    geom_point() +
    facet_wrap(~class)
g <- loon.ggplot(p, activeGeomLayers = 2) # active the second layer
loon.shiny(g)
}

## End(Not run)
Index

grid.arrange, 3  
loon.shiny, 2  
plotOutput, 2  
runApp, 3  
shinyApp, 3