Package 'shinydlplot'

November 18, 2022

Type Package

Title Add a Download Button to a 'shiny' Plot or 'plotly'

Version 0.1.4

Author Alex Pickering

Maintainer Alex Pickering <alexvpickering@gmail.com>

Description Add a download button to a 'shiny' plot or 'plotly' that appears when the plot is hovered. A tooltip, styled to resemble 'plotly' buttons, is displayed on hover of the download button. The download button can be used to allow users to download the dataset used for a plot.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.2

Suggests ggplot2 (>= 3.3.2)

Imports shinyBS (>= 0.61), shiny (>= 1.4.0), shinyjs (>= 1.1), plotly (>= 4.9.2), htmlwidgets (>= 1.5.1), htmltools (>= 0.5.0), methods, utils

NeedsCompilation no

Repository CRAN

Date/Publication 2022-11-18 18:00:02 UTC

R topics documented:

downloadablePlot 2
downloadablePlotly 3
downloadablePlotlyUI 4
downloadablePlotUI 5

Index **6**

`downloadablePlot` *Server-side logic for plot with download data button*

Description

Download button appears on hover in top right.

Usage

```
downloadablePlot(input, output, session, plot, filename, content, ...)
```

Arguments

`input, output, session`	standard `shiny` boilerplate.
`plot`	A `ggplot2` object or a function or reactive that generates a plot.
`filename`	A string of the filename, including extension, that the user's web browser should default to when downloading the file; or a function that returns such a string. (Reactive values and functions may be used from this function.)
`content`	A function that takes a single argument `file` that is a file path (string) of a nonexistent temp file, and writes the content to that file path. (Reactive values and functions may be used from this function.)
`...`	additional named arguments passed to `renderPlot`.

Value

No return value, called to generate server logic.

See Also

`downloadablePlotUI`, `renderPlot`.

Examples

```
library(shiny)
library(shinyjs)
library(shinydlplot)
library(ggplot2)

ui <- fluidPage(
  useShinyjs(),
  downloadablePlotUI(id = 'iris_plot')
)

server <- function(input, output, session) {

  plot <- ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) + geom_point()
```

```
  callModule(downloadablePlot,
             id = 'iris_plot',
             plot = plot,
             filename = 'iris.csv',
             content = function(file) {write.csv(iris, file)})
}

## Not run: shinyApp(ui, server)
```

`downloadablePlotly` *Server-side logic for plotly with download data button in modebar*

Description

Server-side logic for plotly with download data button in modebar

Usage

```
downloadablePlotly(
  input,
  output,
  session,
  plot,
  filename,
  content,
  title = "Download plot data"
)
```

Arguments

`input, output, session`	standard `shiny` boilerplate.
`plot`	Object of class `'plotly'` or a function or reactive that generates one.
`filename`	A string of the filename, including extension, that the user's web browser should default to when downloading the file; or a function that returns such a string. (Reactive values and functions may be used from this function.)
`content`	A function that takes a single argument `file` that is a file path (string) of a nonexistent temp file, and writes the content to that file path. (Reactive values and functions may be used from this function.)
`title`	Text for plotly tooltip.

Value

No return value, called to generate server logic.

See Also

downloadablePlotlyUI

---

downloadablePlotlyUI *UI for plotly with download data button in modebar*

Description

UI for plotly with download data button in modebar

Usage

```
downloadablePlotlyUI(id, width = "100%", height = "auto", inline = FALSE)
```

Arguments

id	id string that gets namespaced by shiny::NS.
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. Note that, for height, using "auto" or "100%" generally will not work as expected, because of how height is computed with HTML/CSS.
inline	use an inline (span()) or block container (div()) for the output

Value

an HTML tag object corresponding to the UI for downloadablePlotly.

See Also

NS, downloadablePlotly

Examples

```
library(shiny)
library(shinyjs)
library(shinydlplot)
library(plotly)
ui <- fluidPage(
  useShinyjs(),
  downloadablePlotlyUI(id = 'iris')
)
server <- function(input, output, session) {

  plot <- plot_ly(data = iris, x = ~Sepal.Length, y = ~Petal.Length)

  callModule(downloadablePlotly,
             id = 'iris',
             plot = plot,
```

```
              filename = 'iris.csv',
              content = function(file) {write.csv(iris, file)})
}

## Not run: shinyApp(ui, server)
```

`downloadablePlotUI` *UI for plot with download data button*

Description

UI for plot with download data button

Usage

```
downloadablePlotUI(
  id,
  title = "Download plot data",
  width = "100%",
  height = "400px",
  zoom = FALSE
)
```

Arguments

`id`	id string that gets namespaced by `shiny::NS`.
`title`	Text to display on hover of download button.
`width, height`	Image 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. These two arguments are ignored when `inline = TRUE`, in which case the width/height of a plot must be specified in `renderPlot()`. Note that, for height, using `"auto"` or `"100%"` generally will not work as expected, because of how height is computed with HTML/CSS.
`zoom`	if `TRUE` brush then double-click to zoom.

Value

an HTML tag object corresponding to the UI for `downloadablePlot`.

See Also

`NS`, `downloadablePlot`, `plotOutput`

Index

downloadablePlot, 2, *5*
downloadablePlotly, 3, *4*
downloadablePlotlyUI, *4*, 4
downloadablePlotUI, *2*, 5

NS, *4, 5*

plotOutput, *5*

renderPlot, *2*