

# Package: d3Tree (via r-universe)

September 25, 2024

**Version** 0.3.0

**Date** 2024-01-23

**Title** Create Interactive Collapsible Trees with the JavaScript 'D3' Library

**Description** Create and customize interactive collapsible 'D3' trees using the 'D3' JavaScript library and the 'htmlwidgets' package. These trees can be used directly from the R console, from 'RStudio', in Shiny apps and R Markdown documents. When in Shiny the tree layout is observed by the server and can be used as a reactive filter of structured data.

**Depends** R (>= 4.0.0)

**Imports** htmlwidgets, tibble, tidyselect, dplyr, utils

**Suggests** rmarkdown

**License** MIT + file LICENSE

**URL** <https://github.com/yonicd/d3Tree>

**BugReports** <https://github.com/yonicd/d3Tree/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**Repository** <https://yonicd.r-universe.dev>

**RemoteUrl** <https://github.com/yonicd/d3tree>

**RemoteRef** HEAD

**RemoteSha** d8b98ca7d720f2f21434564c2fa5981d7d4b0b71

## Contents

d3tree . . . . .	2
d3tree-shiny . . . . .	3
df2tree . . . . .	4
stan.models . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

d3tree

*d3tree***Description**

Htmlwidget that binds to d3js trees. When used in Shiny environment the widget returns a data.frame of logical expressions that represent the current state of the tree.

**Usage**

```
d3tree(
  data,
  name = "name",
  value = "value",
  direction = "horizontal",
  activeReturn = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

**Arguments**

data	named list containing hierarchy structure of data created by df2tree and the layout of the tree (collapse,radial,cartesian)
name	character containing the names of the nodes
value	character containing the name of the tooltip column that are used in the leafs
direction	charater containing the direction the collapsible tree layout will grow to horizontal or vertical (can be 'h','v')
activeReturn	character vector of node attributes to observe and return to shiny.
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.
elementId	The input slot that will be used to access the element.

**Details**

activeReturn is set to NULL by default, but can return any attributes that are strings or numeric such as: name,value,depth,id.

Any node attributes requested that are not found in the node keys are ignored.

**Examples**

```
if(interactive()){
  d3tree(list(root = df2tree(
```

```

        rootname='Titanic',
        struct=as.data.frame(Titanic)
      ),
      layout = 'collapse')
    )

d3tree(list(
  root = df2tree(
    rootname = 'Titanic',
    struct = as.data.frame(Titanic),
    tool_tip = letters[1:(ncol(as.data.frame(Titanic))+1)]
  ),
  layout = 'collapse')
)

d3tree(list(
  root = df2tree(
    rootname = 'book',
    struct = stan.models),
  layout = 'collapse')
)
}

```

---

d3tree-shiny

*Shiny bindings for d3tree*


---

## Description

Output and render functions for using d3tree within Shiny applications and interactive Rmd documents.

## Usage

```
d3treeOutput(outputId, width = "100%", height = "400px")
```

```
renderD3tree(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 d3tree
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.

df2tree

*df2tree*

---

**Description**

converts dataframe to json to send to javascript

**Usage**

```
df2tree(struct, rootname = "root", tool_tip = NULL)
```

**Arguments**

struct	data.frame containing the structure the tree will represent
rootname	character name of the root node
tool_tip	character vector of the label to give to the nodes in each hierarchy

**Examples**

```
titanic_df <- as.data.frame(Titanic)

df2tree(struct = titanic_df, rootname = 'Titanic')

df2tree(
  struct = titanic_df,
  rootname = 'Titanic',
  tool_tip = letters[1:5]
)
```

---

stan.models*stan.models*

---

**Description**

data.frame containing the structure of the github repository <https://github.com/stan-dev/example-models> that contains examples to run STAN models in R from the book by Gelman and Hill 'Data Analysis Using Regression Analysis and Multilevel/Hierarchical Models'.

**Usage**

```
stan.models
```

**Format**

An object of class "data.frame"

**Examples**

```
data(stan.models)  
stan.models
```

# Index

\* **datasets**

stan.models, 4

d3tree, 2

d3tree-shiny, 3

d3treeOutput (d3tree-shiny), 3

df2tree, 4

renderD3tree (d3tree-shiny), 3

stan.models, 4