

Package ‘autoplotly’

August 22, 2026

Type Package

Title Automatic Generation of Interactive Visualizations for
Statistical Results

Version 0.1.7

Description Functionalities to automatically generate interactive visualizations for statistical results supported by 'ggfortify', such as time series, PCA, clustering and survival analysis, with 'plotly.js' <<https://plotly.com/>> and 'ggplot2' style. The generated visualizations can also be easily extended using 'ggplot2' and 'plotly' syntax while staying interactive.

License GPL-2

URL <https://github.com/terrytangyuan/autoplotly>

BugReports <https://github.com/terrytangyuan/autoplotly/issues>

Encoding UTF-8

Depends R (>= 3.1)

Imports methods, ggplot2, plotly, ggfortify

RoxygenNote 7.0.2

VignetteBuilder knitr

Suggests testthat, knitr, markdown

NeedsCompilation no

Author Yuan Tang [aut, cre] (ORCID: <<https://orcid.org/0000-0001-5243-233X>>)

Maintainer Yuan Tang <terrytangyuan@gmail.com>

Repository CRAN

Date/Publication 2026-08-22 15:10:52 UTC

Contents

autoplotly	2
Index	5

autoplotly	<i>Automatic Visualization of Popular Statistical Results Using 'plotly.js' and 'ggplot2'</i>
------------	---

Description

This function provides functionality to automatically generate interactive plot for many popular statistical results supported by 'ggfortify' package using 'plotly.js' and 'ggplot2'.

This package provides functionalities to automatically generate interactive visualizations for many popular statistical results supported by ggfortify package with 'plotly.js' and 'ggplot2' style. The generated plot can also be easily extended using 'ggplot2' and 'plotly' syntax while staying interactive.

Usage

```
autoplotly(
  object,
  ...,
  width = NULL,
  height = NULL,
  tooltip = "all",
  dynamicTicks = FALSE,
  layerData = 1,
  originalData = TRUE,
  source = "A",
  widths = NULL,
  heights = NULL,
  margin = 0.02,
  shareX = TRUE,
  shareY = TRUE,
  titleX = shareX,
  titleY = shareY,
  which_layout = "merge"
)
```

Arguments

object	The object that represents your statistical result, e.g. <code>stats::prcomp(iris[-5])</code> .
...	Arguments passed to <code>ggfortify::autoplot</code> function for the applied object. For example, if your object is constructed from <code>stats::prcomp(iris[-5])</code> , you can find the documentation for the list of additional arguments via <code>?ggfortify::autoplot.prcomp</code> .
width	Width of the plot in pixels (optional, defaults to automatic sizing).
height	Height of the plot in pixels (optional, defaults to automatic sizing).
tooltip	A character vector specifying which aesthetic mappings or columns from the generated plot data to show in the tooltip. The default, "all", shows all aesthetic

	mappings. For example, <code>tooltip = c("x", "y", "colour", "Sepal.Length")</code> also shows the <code>Sepal.Length</code> column when it is present in the fortified data.
<code>dynamicTicks</code>	should plotly.js dynamically generate axis tick labels? Dynamic ticks are useful for updating ticks in response to zoom/pan interactions; however, they can not always reproduce labels as they would appear in the static <code>ggplot2</code> image.
<code>layerData</code>	data from which layer should be returned?
<code>originalData</code>	should the "original" or "scaled" data be returned?
<code>source</code>	a character string of length 1. Match the value of this string with the source argument in <code>event_data()</code> to retrieve the event data corresponding to a specific plot (shiny apps can have multiple plots).
<code>widths</code>	relative width of each column on a 0-1 scale. By default all columns have an equal relative width.
<code>heights</code>	relative height of each row on a 0-1 scale. By default all rows have an equal relative height.
<code>margin</code>	either a single value or four values (all between 0 and 1). If four values are provided, the first is used as the left margin, the second is used as the right margin, the third is used as the top margin, and the fourth is used as the bottom margin. If a single value is provided, it will be used as all four margins.
<code>shareX</code>	should the x-axis be shared amongst the subplots?
<code>shareY</code>	should the y-axis be shared amongst the subplots?
<code>titleX</code>	should x-axis titles be retained?
<code>titleY</code>	should y-axis titles be retained?
<code>which_layout</code>	adopt the layout of which plot? If the default value of "merge" is used, layout options found later in the sequence of plots will override options found earlier in the sequence. This argument also accepts a numeric vector specifying which plots to consider when merging.

Details

With `ggplot2` 4.0, direct composition using `+` requires R 4.3 or later. On R 4.1 and R 4.2, add components as a list, for example `p + list(ggplot2::labs(title = "Title"))`. Compatibility with `ggplot2` 3.x is unchanged.

Examples

```
# Automatically generate interactive plot for results produced by `stats::prcomp`
p <- autoplotly(prcomp(iris[c(1, 2, 3, 4)]), data = iris,
  colour = 'Species', label = TRUE, label.size = 3, frame = TRUE)

# Include additional columns from the data in the tooltip
autoplotly(prcomp(iris[c(1, 2, 3, 4)]), data = iris,
  colour = "Species",
  tooltip = c("x", "y", "colour", "Sepal.Length", "Petal.Length"))

# You can apply additional ggplot2 elements to interactive plot built using `autoplotly()`
p +
```

```
ggplot2::ggtitle("Principal Components Analysis") +  
ggplot2::labs(y = "Second Principal Components", x = "First Principal Components")  
  
# Or apply additional plotly elements to the generated interactive plot  
p %>% plotly::layout(annotations = list(  
  text = "Example Text",  
  font = list(  
    family = "Courier New, monospace",  
    size = 18,  
    color = "black"),  
  x = 0,  
  y = 0,  
  showarrow = TRUE))
```

Index

`autoplotly`, [2](#)

`event_data()`, [3](#)