

Package ‘gganime’

August 30, 2026

Title Animate 'ggplot2' Plots with 'Anime.js'

Version 0.1.0

Description Renders 'ggplot2' plots written with 'gganimate' syntax as animated Scalable Vector Graphics (SVG) drawn in the browser by the 'Anime.js' JavaScript library (<<https://animejs.com>>), through the 'animejs' package. The result is a single self-contained, resolution-independent 'htmlwidget' with one timeline over the plot's elements.

License MIT + file LICENSE

URL <https://github.com/long39ng/gganime>,
<https://long39ng.github.io/gganime/>

BugReports <https://github.com/long39ng/gganime/issues>

Depends R (>= 4.1.0)

Imports animejs (>= 1.1.0), cli, gganimate (>= 1.0.11), ggplot2 (>= 4.0.0), grDevices, grid, gridSVG (>= 1.7-5), rlang, utils, xml2

Suggests jsonlite, knitr, rmarkdown, shiny, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

NeedsCompilation no

Author Long Nguyen [aut, cre] (ORCID: <<https://orcid.org/0000-0001-8878-7386>>)

Maintainer Long Nguyen <nguyen@dezim-institut.de>

Repository CRAN

Date/Publication 2026-08-30 09:20:02 UTC

Contents

anime	2
gganime-shiny	3

anime

*Render a gganimate plot as an animated-SVG widget***Description**

Builds a ggplot2 plot written with gganimate syntax and renders it as a self-contained, resolution-independent animated-SVG htmlwidget powered by Anime.js, instead of gganimate's frame-by-frame output.

Usage

```
anime(
  plot,
  nframes = 100,
  fps = 10,
  duration = NULL,
  width = 640,
  height = 480,
  precision = 2,
  loop = TRUE,
  controls = TRUE,
  elementId = NULL,
  ...
)
```

Arguments

plot	A gganim plot: a ggplot2 plot with a <code>transition_*</code> () added.
nframes	Number of frames to sample the animation at.
fps	Frames per second, setting the playback speed.
duration	Optional total duration in seconds; overrides fps.
width, height	Widget size in pixels, defaulting to 640 by 480. The plot is rendered at these dimensions, so their ratio sets the plot's aspect ratio.
precision	Decimal places to round animated coordinates to.
loop	Loop the animation: TRUE, FALSE, or a number of iterations.
controls	Show a play/pause and scrub control bar.
elementId	Optional htmlwidget element id.
...	Unused; reserved for future arguments.

Value

An htmlwidget of class gganime.

Examples

```
library(ggplot2)

p <- ggplot(iris, aes(Sepal.Width, Sepal.Length, size = Petal.Length)) +
  geom_point(colour = "steelblue") +
  labs(title = "{closest_state}") +
  transition_states(Species)

anime(p, nframes = 20)
```

gganime-shiny

Shiny bindings for gganime

Description

Output and render functions for using gganime widgets within Shiny applications and interactive R Markdown documents.

Usage

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

renderGganime(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 gganime widget, typically a call to anime() .
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with <code>quote()</code>)? This is useful if you want to save an expression in a variable.

Value

`gganimeOutput()` returns a Shiny output function for a UI definition; `renderGganime()` returns a Shiny render function to assign to an output slot.

Examples

```
if (interactive() && rlang::is_installed("shiny")) {  
  library(shiny)  
  library(ggplot2)  
  
  ui <- fluidPage(gganimeOutput("plot"))  
  
  server <- function(input, output, session) {  
    output$plot <- renderGganime({  
      anime(  
        ggplot(mtcars, aes(mpg, wt)) +  
        geom_point() +  
        transition_states(gear)  
      )  
    })  
  }  
  
  shinyApp(ui, server)  
}
```

Index

`anime`, [2](#)

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

`gganime-shiny`, [3](#)

`gganimeOutput (gganime-shiny)`, [3](#)

`renderGganime (gganime-shiny)`, [3](#)