

# Package ‘svgPanZoom’

September 26, 2016

**Title** R 'Htmlwidget' to Add Pan and Zoom to Almost any R Graphic

**Version** 0.3.3

**Maintainer** Kent Russell <kent.russell@timelyportfolio.com>

**Description** This 'htmlwidget' provides pan and zoom interactivity to R graphics, including 'base', 'lattice', and 'ggplot2'. The interactivity is provided through the 'svg-pan-zoom.js' library. Various options to the widget can tailor the pan and zoom experience to nearly any user desire.

**URL** <https://github.com/timelyportfolio/svgPanZoom>

**BugReports** <https://github.com/timelyportfolio/svgPanZoom/issues>

**License** MIT + file LICENSE

**Depends** R (>= 3.1.2)

**Imports** htmlwidgets (>= 0.3.2)

**Suggests** htmltools, svglite

**Enhances** gridSVG, knitr, XML, xml2

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Anders Riutta et. al. [aut, cph] (svg-pan-zoom.js BSD-licensed library in htmlwidgets/lib, <https://github.com/ariutta/svg-pan-zoom>),  
Jorik Tangelder [aut, cph] (hammer.js MIT-licensed touch library in htmlwidgets/lib, <https://github.com/hammerjs/hammer>),  
Kent Russell [aut, cre] (R interface to svg-pan-zoom.js)

**Repository** CRAN

**Date/Publication** 2016-09-26 21:27:03

## R topics documented:

svgPanZoom . . . . .	2
svgPanZoom-shiny . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

svgPanZoom

*Pan and Zoom R graphics***Description**

Add panning and zooming to almost any R graphics from base graphics, lattice, and ggplot2 by using the JavaScript library [svg-pan-zoom](#).

**Usage**

```
svgPanZoom(svg, viewBox = TRUE, ..., width = NULL, height = NULL,
            elementId = NULL)
```

**Arguments**

svg	one of <ul style="list-style-type: none"> <li>• svg - SVG as XML or xml2, such as return from <a href="#">xmlSVG</a></li> <li>• lattice plot - trellis object, such as l in <code>l=xypplot(...)</code></li> <li>• ggplot2 plot - ggplot object, such as g in <code>g=ggplot(...)</code> + <code>geom_line()</code></li> <li>• filename or connection of a SVG file</li> </ul>
viewBox	logical to add back the viewBox to the SVG. The default is TRUE to fit the svgPanZoom in its container.
...	other configuration options for <code>svg-pan-zoom.js</code> . See <a href="#">svg-pan-zoom How To Use</a> for a full description of the options available. As an example to turn on <code>controlIconsEnabled</code> and turn on, do <code>svgPanZoom(..., controlIconsEnabled = TRUE, panEnabled</code>
width, height	valid CSS unit (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended
elementId	string id for the <code>svgPanZoom</code> container. Since <code>svgPanZoom</code> does not display its container, this is very unlikely to be anything other than the default NULL.

**Examples**

```
# svgPanZoom tries to be very flexible with its first argument

# in this first example use SVG as a character string
# this is probably the least likely use case
library(svgPanZoom)
svgPanZoom('
  <svg style="height:300px;width:300px;">
    <circle cx="60" cy="60" r="50" style="fill:none;stroke:blue;"/>
  </svg>
')
```

```
## Not run:
library(svgPanZoom)
```

```

# first let's demonstrate a base plot
# use svglite for now
library(svglite)
library(lattice)
svgPanZoom( svglite:::inlineSVG( plot(1:10) ) )

svgPanZoom(svglite:::inlineSVG(show( xyplot( y~x, data.frame(x=1:10,y=1:10) ) )))

# the package gridSVG is highly recommended for lattice and ggplot2
# second let's demonstrate a lattice plot
library(lattice)
svgPanZoom( xyplot( y~x, data.frame(x=1:10,y=1:10) ) )

# third with a ggplot2 plot
library(ggplot2)
svgPanZoom( ggplot( data.frame(x=1:10,y=1:10), aes(x=x,y=y) ) + geom_line() )

#Of course as a good htmlwidget should, it works with Shiny also.
library(shiny)
library(svglite)
library(svgPanZoom)
library(ggplot2)

ui <- shinyUI(bootstrapPage(
  svgPanZoomOutput(outputId = "main_plot")
))

server = shinyServer(function(input, output) {
  output$main_plot <- renderSvgPanZoom({
    p <- ggplot() +
      geom_point(
        data=data.frame(faithful),aes(x=eruptions,y=waiting)
      ) +
      stat_density2d(
        data=data.frame(faithful)
        ,aes(x=eruptions,y=waiting ,alpha =..level..)
        ,geom="polygon") +
      scale_alpha_continuous(range=c(0.05,0.2))

    svgPanZoom(p, controlIconsEnabled = T)
  })
})

runApp(list(ui=ui,server=server))

## End(Not run)

```

**Description**

Shiny bindings for svgPanZoom

**Usage**

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

```
renderSvgPanZoom(expr, env = parent.frame(), quoted = FALSE)
```

**Arguments**

outputId	output variable to read from
width, height	must be a valid CSS unit (like "100" which will be coerced to a string and have "px" appended)
expr	expression that generates a svgPanZoom htmlwidget
env	environment in which to evaluate expr
quoted	logical is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

# Index

`renderSvgPanZoom` (`svgPanZoom-shiny`), [3](#)

`svgPanZoom`, [2](#)

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

`svgPanZoomOutput` (`svgPanZoom-shiny`), [3](#)

`xmlSVG`, [2](#)