

Introduction to Package 'animation' in R

Wang Ze Xian

Department of Statistics
the School of Economics, Xiamen University

Outline

- 1 Motivation
 - Why we need “animation” ?
- 2 Description
 - The Package “animation”
 - Basic Usage
- 3 Example

Outline

- 1 Motivation
 - Why we need “animation” ?
- 2 Description
 - The Package “animation”
 - Basic Usage
- 3 Example

The Advantage of Using Animation in R.

- Animation can easily show the dynamic change especially when there are so many Variables.
- Using animation we can clearly observe a 3D graph from every angle.
- Animation is more easily to explain and it is interesting.

The Advantage of Using Animation in R.

- Animation can easily show the dynamic change especially when there are so many Variables.
- Using animation we can clearly observe a 3D graph from every angle.
- Animation is more easily to explain and it is interesting.

The Advantage of Using Animation in R.

- Animation can easily show the dynamic change especially when there are so many Variables.
- Using animation we can clearly observe a 3D graph from every angle.
- Animation is more easily to explain and it is interesting.

Why Using R to Generate Animation?

- R is one of the most popular and useful tools to statistics.
- R has many powerful tools to make static graph (e.g ggplot).
- Animation is generate from many static graph.

Why Using R to Generate Animation?

- R is one of the most popular and useful tools to statistics.
- R has many powerful tools to make static graph (e.g ggplot).
- Animation is generate from many static graph.

Why Using R to Generate Animation?

- R is one of the most popular and useful tools to statistics.
- R has many powerful tools to make static graph (e.g ggplot).
- Animation is generate from many static graph.

Why Using Package “animation”?

- Easily to create animation with few commands.
- Support saving animation to various formats.(e.g. Flash, GIF, HTML pages, PDF and videos.)

Two wonderful examples(url)

- R : plot in Hans Rosling's Talk
- Population structure of Japan

Why Using Package “animation”?

- Easily to create animation with few commands.
- Support saving animation to various formats.(e.g. Flash, GIF, HTML pages, PDF and videos.)

Two wonderful examples(url)

- R : plot in Hans Rosling's Talk
- Population structure of Japan

Outline

- 1 Motivation
 - Why we need “animation” ?
- 2 Description
 - The Package “animation”
 - Basic Usage
- 3 Example

The Content of This Package.

- This package contains a variety functions for animations in statistics which could probably aid in teaching statistics and data analysis.
- It also has several utilities to export R animations to other formats.
- Author: Yihui Xie , Christian Mueller , Lijia Yu , Weicheng Zhu.
- HomePage

The Content of This Package.

- This package contains a variety functions for animations in statistics which could probably aid in teaching statistics and data analysis.
- It also has several utilities to export R animations to other formats.
- Author: Yihui Xie , Christian Mueller , Lijia Yu , Weicheng Zhu.
- HomePage

The Content of This Package.

- This package contains a variety functions for animations in statistics which could probably aid in teaching statistics and data analysis.
- It also has several utilities to export R animations to other formats.
- Author:Yihui Xie ,Christian Mueller ,Lijia Yu ,Weicheng Zhu.
- [HomePage](#)

The Content of This Package.

- This package contains a variety functions for animations in statistics which could probably aid in teaching statistics and data analysis.
- It also has several utilities to export R animations to other formats.
- Author:Yihui Xie ,Christian Mueller ,Lijia Yu ,Weicheng Zhu.
- HomePage

Install

Using this command to install it in R

- `install.packages(“animation”)`

System Requirements

- ImageMagick(url) or GraphicsMagick(url) or LyX(url) for saving GIF.
- (PDF)LaTeX for saving Latex.
- SWF Tools(url) for saving SWF.
- FFmpeg(url) or avconv(url) for saving video.

Install

Using this command to install it in R

- `install.packages(“animation”)`

System Requirements

- ImageMagick(url) or GraphicsMagick(url) or LyX(url) for saving GIF.
- (PDF)LaTeX for saving Latex.
- SWF Tools(url) for saving SWF.
- FFmpeg(url) or avconv(url) for saving video.

Outline

- 1 Motivation
 - Why we need “animation” ?
- 2 Description
 - The Package “animation”
 - Basic Usage
- 3 Example

Generate Animation

ani.option()

ani.options()

- There are various parameters that control the behaviour of the animation, such as time interval, maximum number of animation frames, height and width, etc.
- Use this command to set the options of animation before generating.

Generate Animation

`ani.option()`

Example

- `ani.options(interval = 0.05, nmax = 30)`
 - The time interval of the animation to be generated is 0.05s, and the maximum number of steps in a loop to create animation frames is 30.
- `ani.options(ani.width=100, ani.height=100)`
 - Width and height of image frames both are 100.
- `ani.options()`
 - return all the parameters of function `ani.options()`.

Generate Animation

`ani.pause()`

`ani.pause()`

- If this function is called in an interactive graphics device, it will pause for a time interval (by default specified in `ani.options('interval')`) and flush the current device; otherwise it will do nothing.

example

- `ani.pause()`
 - pause for a time interval the same as which is specified in `ani.options('interval')`.
- `ani.pause(interval = 0.05)`
 - pause for 0.05s.

Generate Animation

`ani.pause()`

`ani.pause()`

- If this function is called in an interactive graphics device, it will pause for a time interval (by default specified in `ani.options('interval')`) and flush the current device; otherwise it will do nothing.

example

- `ani.pause()`
 - pause for a time interval the same as which is specified in `ani.options('interval')`.
- `ani.pause(interval = 0.05)`
 - pause for 0.05s.

Generate Animation

`ani.record()`

Block: `ani.record()`

- These two functions use `recordPlot` and `replayPlot` to record image frames and replay the animation respectively.

example

- `ani.record(reset = True)`
 - clear history before recording.
- `ani.record(replay.cur = True)`
 - whether to replay the current plot (we can set both `reset` and `replay.cur` to `TRUE` so that low-level plotting changes can be captured by off-screen graphics devices without storing all the plots in memory).

Generate Animation

`ani.record()`

Block: `ani.record()`

- These two functions use `recordPlot` and `replayPlot` to record image frames and replay the animation respectively.

example

- `ani.record(reset = True)`
 - clear history before recording.
- `ani.record(replay.cur = True)`
 - whether to replay the current plot (we can set both `reset` and `replay.cur` to `TRUE` so that low-level plotting changes can be captured by off-screen graphics devices without storing all the plots in memory).

Generate Animation

Setup a simple animation

Make a animation using loop

- To generate a simple animation, we can use loop (e.g. “for”) to plot several graphs then use `ani.pause` to play them in a correct order.

Example

- `ani.options(interval = 0.2, nmax = 10)`
- `for (i in 1:ani.options("nmax")) {plot(rnorm(30)); ani.pause()}`
 - generate a animation in which 30 random points follow normal distribution change their position every 0.2 second lasting for 2 second.

Generate Animation

Setup a simple animation

Make a animation using loop

- To generate a simple animation, we can use loop (e.g. "for") to plot several graphs then use `ani.pause` to play them in a correct order.

Example

- `ani.options(interval = 0.2, nmax = 10)`
- `for (i in 1:ani.options("nmax")) {plot(rnorm(30)); ani.pause()}`
 - generate a animation in which 30 random points follow normal distribution change their position every 0.2 second lasting for 2 second.

Generate Animation

Animations built in “animation” package

Animations built in “animation” package and demo

- In this package, there are many animations built in already, we can view them by some simply commands.
- I have converted some of these animations to html files which can be found in the folder containing this pdf.

Generate Animation

Animations built in "animation" package

Example

- `BM.circle(n = 20, col = rainbow(n))`
 - several points moving randomly in a circle.
- `brownian.motion(n = 10, xlim = c(-20, 20), ylim = c(-20, 20))`
 - Brownian motion, or random walk, can be regarded as the trace of some cumulative normal random numbers.
- `buffon.needle(l = 0.8, d = 1, redraw = TRUE, mat = matrix(c(1, 3, 2, 3), 2), heights = c(3, 2), col = c("lightgray", "red", "gray", "red", "blue", "black", "red"), expand = 0.4, type = "l")`
 - a simulation for the problem of Buffon's Needle, which is one of the oldest problems in the field of geometrical probability.

Saving A Animation

Saving a animation

- Animations played in R can't be easily shown to other people especially when they didn't install R.
- Package “animation” provides some useful functions to save your animations to several formats.
- Animations can be saved into GIFs,HTML,Latex,SWF and Video,
- using `saveGIF()`,`saveHTML()`,`saveLatex()`,`saveSWF()` and `saveVideo`.

Saving A Animation

Example - Save to GIF

- ```
ani.options("C:/Software/LyX/etc/ImageMagick/convert.exe")
```

  - set the path to the software which will help to convert animation to gif.
- ```
saveGIF({ brownian.motion(pch = 21, cex = 5, col = "red",  
bg = "yellow") }, movie.name = "brownian_motion.gif",  
interval = 0.1, nmax = 30, ani.width = 600, ani.height = 600)
```

 - save the "brownian.motion" to "brownian_motion.gif".

Saving A Animation

Example - Save to HTML

```
saveHTML(  
  { par(mar = c(3, 3, 1, 0.5), mgp = c(2, 0.5, 0), tcl = -0.3,  
    cex.axis = 0.8, cex.lab = 0.8, cex.main = 1) ;ani.options(  
    interval = 0.05, nmax = ifelse(interactive(), 150, 2)) ;  
    buffon.needle(l = 0.8, d = 1, redraw = TRUE,  
    mat = matrix(c(1, 3, 2, 3), 2), heights = c(3, 2), col =  
    c("lightgray", "red", "gray", "red", "blue", "black", "red"),  
    expand = 0.4, type = "l") }, img.name = "buffon.needle",  
    htmlfile = "buffon.needle.html",navigator = FALSE,  
    description = c("a simulation for the problem of Buffon's  
    Needle", "(without the navigation panel)"))
```


Saving A Animation

Parameters - Save to HTML

- which in `{}` is the animation we made before,
- `img.name` sets the name of the image,
- `htmlfile` sets the name of the html file which will be generating,
- `navigator` determines whether to show the navigator,
- `description` sets what will be shown as the description in the html.

A Simple Example

- Here is a simple example written by myself to show the dynamic sine wave.
- The HTML file are contained in the folder which containing this pdf.

A Simple Example

- Here is a simple example written by myself to show the dynamic sine wave.
- The HTML file are contained in the folder which containing this pdf.

A Simple Example

Sine Wave-first

```
generate a animation  
library(animation);library(ggplot2);  
x<-seq(from=0,to=4*pi,by=0.01);  
y<-sin(x);m <- data.frame(x,y);  
p<-ggplot(data=m,aes(x=m$x,y=m$y));  
ani.options(interval=0.02);  
for(i in 1:100){m$y<-sin(m$x+i);  
k<-p+geom_point(color="pink");print(k);  
ani.pause()}}
```

A Simple Example

Sine Wave-second

save to HTML file

```
saveHTML({x<-seq(from=0,to=4*pi,by=0.01);  
y<-sin(x);m <- data.frame(x,y);  
p<-ggplot(data=m,aes(x=m$x,y=m$y));  
ani.options(interval=0.02);  
for(i in 1:100){m$y<-sin(m$x+i);  
k<-p+geom_point(color="pink");print(k);  
ani.pause()}} ,img.name="Sinewave",  
htmlfile = "Sinewave.html")
```

For Further Reading I



Yihui Xie

Package 'animation'



Github