New Features and Enhancements of the animation Package

Yihui Xie

Department of Statistics, Iowa State University

2011-03-04
The winter break

- nice weather in Ames
- + my limited driving skills ⇒

⇒ substantial updates to the animation package

see GitHub: https://github.com/yihui/animation/graphs/impact
The winter break

- nice weather in Ames
- + my limited driving skills ⇒
- substantial updates to the animation package
- see GitHub:
  https://github.com/yihui/animation/graphs/impact
Introduction

- a package for fun
- very naive ideas! (see the basic schema later)
- to escape the world of $\alpha, \beta, \gamma \ldots$ but who knows...
Introduction

▷ a package for fun
▷ very naive ideas! (see the basic schema later)
▷ to escape the world of $\alpha, \beta, \gamma$... but who knows...
▷ it won the John Chambers Software award in 2009
  ▷ felt as if I won the lottery
  ▷ it turned out that... :-(
Introduction

▶ a package for fun
▶ very naive ideas! (see the basic schema later)
▶ to escape the world of $\alpha, \beta, \gamma$... but who knows...
▶ it won the John Chambers Software award in 2009
  ▶ felt as if I won the lottery
  ▶ it turned out that... :-(
▶ seriously, people are using this package! (feedback from old friends, teachers, ...)


Introduction

- a package for fun
- very naive ideas! (see the basic schema later)
- to escape the world of $\alpha, \beta, \gamma$... but who knows...
- it won the John Chambers Software award in 2009
  - felt as if I won the lottery
  - it turned out that... :-(
- seriously, people are using this package! (feedback from old friends, teachers, ...)
- are all these a joke?!
The Holy Truth

```r
## set some options first

ani.options(interval = 0.2, nmax = 10)
## use a loop to create plots one by one

for (i in 1:ani.options("nmax")) {
    draw_plot()  # may need calculations beforehand
    ani.pause()  # pause for a while
}

## that's it!
```

The rest of things

- once you know the holy truth, there is only one thing to do
The rest of things

- once you know the holy truth, there is only one thing to do
- programming like crazy, using as wild imagination as possible
- looks like "labor work"
A dirty example

```r
library(animation)

brownian.motion

function (n = 10, xlim = c(-20, 20), ylim = c(-20, 20), ...) {
  x = rnorm(n)
  y = rnorm(n)
  for (i in seq_len(ani.options("nmax"))) {
    dev.hold()
    plot(x, y, xlim = xlim, ylim = ylim, ...)
    text(x, y)
    x = x + rnorm(n)
    y = y + rnorm(n)
    ani.pause()
  }
}

<bytecode: 0x7fba196d9e28>
<environment: namespace:animation>
```
Standing on the shoulders of giants

Starting with:

- none of the exporting utilities is really my contribution, except the first version of the HTML animations
- the workhorses:
  - `saveHTML()` – using SciAnimator (a JS library I found right before I started to rewrite my JavaScript)
  - `saveGIF()` – using GraphicsMagick or ImageMagick
  - `saveSWF()` – using SWF Tools
  - `saveLatex()` – using the \LaTeX\ package animate
  - `saveVideo()` – using FFmpeg
this approach does not require additional software; a web browser is enough

give me a code chunk to produce plots, and I’ll return you a web page with animations
saveHTML()

```
saveHTML({

  par(mar = c(4, 4, 0.5, 0.5))

  for (i in 1:20) {

    plot(runif(20), ylim = c(0, 1))

    ani.pause()

  }

}, img.name = "unif_plot", htmlfile = "random.html")
```
The interface in the HTML page

▶ like a movie player (written in JavaScript)
▶ see the quincunx() example (change the loop mode and you will find real fun!)
A crazy extension: Rweb

- `saveHTML()` is so flexible that you can submit some R code to a server, and it will immediately generate the animation for you!
- see `system.file('misc', 'Rweb', 'demo.html', package = 'animation')`
- this feature is experimental (subject to the server admin)
saveGIF(), saveSWF(), saveVideo()

- they require additional software packages
- similar usage to saveHTML()
- did really hard work under the all-mighty Windows system
saveLatex()

- this package was listed in the CRAN Task View “ReproducibleResearch” before it is really prepared to be listed there!
  - why things always come earlier than I expected?
- http://cran.r-project.org/web/views/ReproducibleResearch.html
- so saveLatex() was updated to incorporate with Sweave for reproducible research
- it was originally written to insert animations in \LaTeX documents only, using a \LaTeX package animate
- see demo('Sweave_animation', package = 'animation')
- we can watch PDF animations only with Adobe Reader
Another crazy extension

- we can capture arbitrary animations in \texttt{\LaTeX} with \texttt{save\LaTeX{}} (in fact, with any \texttt{save*{}} functions)
- e.g. the \texttt{rgl} 3D animations
- \texttt{demo('rgl\_animation', package = 'animation')}
Yet another crazy extension

- we can download images from the internet and create an animation
- `demo('flowers', package = 'animation')` shows some flower photos I took a few years ago
A gory detail that puzzled me for years

▶ low-level plotting functions in base graphics cannot be recorded by R graphics devices as new images
▶ e.g. add points one by one to a plot – you only get one image file under a `png()` device
▶ this brings us ugly solutions – redraw the whole plot in order to be recorded
▶ now we have a new solution
The animation recorder

- R graphics can be recorded by recordPlot() (as a list)
- and the recorded plot can be replayed
- ani.record() and ani.replay() extended these functions a little bit, so we can record a sequence of images and replay later
An example of the recorder

```r
library(animation)
n = 20
x = sort(rnorm(n))
y = rnorm(n)
## set up an empty frame; add points one by one
plot(x, y, type = "n")
ani.record(reset = TRUE)  # clear history first
for (i in 1:n) {
  points(x[i], y[i], pch = 19, cex = 2)
  ani.record()  # record the current frame
}
ani.options(interval = 0.5)
ani.replay()
```
Topics in statistics

▶ the bisection method, the gradient descent algorithm, Newton-Raphson
▶ Brownian motion
▶ bootstrapping, cross-validation
▶ Buffon’s needle
▶ CLT, LLN
▶ flipping coins
▶ least squares
▶ k-Means clustering, k-Nearest neighbors
▶ survey sampling methods
▶ simulating QQ plots
▶ Monte Carlo integration
▶ (see the JSS manuscript for examples)
see demo(package = 'animation')

some are amusing and entertaining

let’s watch a few of them

- the NBA game
- Mandelbrot set
- Xmas, Xmas2, Xmas_card
- busybees: in memory of my childhood
- simulating the fire
- the Game of Life (cat faces?)
- Tower of Hanoi
- the Jumping Horse
Final notes

- this talk is based on animation 2.0-2, which is not released to CRAN yet, but most things should work
- if you are really brave and eager to follow the latest version, go to https://github.com/yihui/animation
- otherwise install.packages('animation')
- a web collection of animations: http://animation.yihui.name
Conclusions

- finally I feel less guilty for winning the Chambers award
Conclusions

- finally I feel less guilty for winning the Chambers award
- life should be fun
Conclusions

▶ finally I feel less guilty for winning the Chambers award
▶ life should be fun
▶ seriously speaking, I think we are too serious
▶ looking forward to more alive statistical reports!