

# Turning Statistical Ideas into Animations

Yihui Xie

Department of Statistics, Iowa State University

September 21, 2009

# Outline

## 1 Why?

- Personal concerns
- Looking inside methods
- Entertaining other people beside teaching

## 2 What?

- Simulation (Prob Theory, Math Stat)
- Explanation (Machine Learning, Stat Computation)

## 3 How?

- Sketch of R package animation
- Output formats
- Other packages

# I'm Confused Sometimes...

*It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way.*

— Charles Dickens

# $k$ -means Clustering When Outliers Exist

Figure 1: How an outlier influenced  $k$ -means clustering ([expl1-kmeans.r](#))

# $k$ -means Clustering (cont'd)

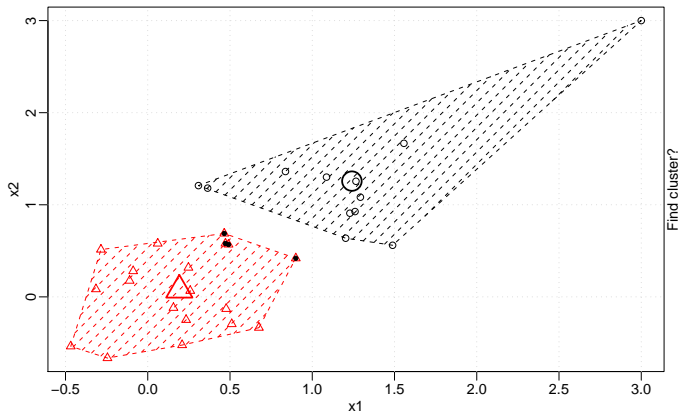


Figure 2: Four mislabeled points

# Must statisticians always be serious?



Stone (40,000 B.C.)



iPhone 3G (2008)

MMS	X	X
Video recording	X	X
Videocall	X	X
Changeable memory cards	X	X
Touchscreen	X	✓

Figure 3: Do we really need to prove everthing exactly?

# Buffon's Needle – Estimating $\pi$

Figure 4: Scenario for Buffon's Needle ([expl2-buffon.r](#))

# Simulation for A Coin-Flipping Problem

Figure 5: How many steps are needed for HTT and HTH to appear in a coin-flipping sequence? ([expl3-coin.r](#))



# Yet Another Presentation of CLT

Figure 6: People often think the “bell-shaped” curve has demonstrated normality well, but I don’t think so ([expl4-clt.r](http://expl4-clt.r))

# $k$ -Nearest Neighbor Algorithm

Figure 7: A point looked around and decided to join the majority  
([expl5-knn.r](#))

# Monte-Carlo Integration $\int_0^1 f(x)dx$

Figure 8: Draw  $x_i \sim U(0, 1)$ , we know  $\int_0^1 f(x) \cdot 1dx \approx \sum f(x_i)/n$  by LLN  
([expl6-mcint.r](http://expl6-mcint.r))

# A Sketch for Creating Animations in R

## Listing 1: Sketch of animation

```

1 ani.fun <- function(args.for.stat.method,
2   args.for.graphics, ...) {
3   {stat.calculation.for.preparation.here}
4   i = 1
5   while (i <= ani.options("nmax") &
6     other.conditions.for.stat.method) {
7     {stat.calculation.for.animation}
8     {plot.results.in.ith.step}
9     # pause for a while in this step
10    Sys.sleep(ani.options("interval"))
11    i = i + 1
12  }
13  # (i - 1) frames produced in the loop
14  ani.options("nmax") = i - 1
15  {return.something}
16 }
```

# Installing the animation Package

- CRAN:

<http://cran.r-project.org/web/packages/animation/>

- R-Forge: `install.packages("animation", repos = "http://r-forge.r-project.org")` (daily built; bug fixes and new features included)

# Dirty Examples

## Text Rotation, Bouncing Balls and Burning Fire

```
library(animation)
demo("wordrotation")
demo("busybees")
demo("fire")
```

# HTML Pages

- save animation frames as static (bitmap) images, e.g. 1.png, 2.png, ...
- use `cat()` to write HTML and JavaScript code (pure text) into a file (\*.html)
- JavaScript controls CSS attributes of layers (`style = display:none` or `display:block`) to show images one by one
- R functions: `ani.start()`, `ani.stop()`
- advantage of the HTML format: a web browser with JavaScript enabled is enough for viewing the output
- see [expl8-html.r](#)
- AniWiki: <http://animation.yihui.name>

## GIF and MPEG (by ImageMagick)

- the convert utility: `convert -delay [...] -loop [...] *.png filename.gif`
- use `system()` (or `shell()` under Windows) to call convert
- pass arguments in the R function `saveMovie()` to convert
- GIF is a common image format, but we need additional software ImageMagick to create it
- MPEG is bad (according to my experience)
- example: `saveMovie(for(i in 1:10) plot(runif(10), ylim = 0:1))` (and open it with `shell.exec(normalizePath(file.path(tempdir()), "movie.gif"))`)



# Flash (by SWF Tools)

- the utilities pdf2swf, png2swf, jpeg2swf, ...
- e.g. `png2swf *.png -o output.swf -r fps`
- usually pdf is a better choice but pdf2swf has limitations
- e.g.

```
ani.options(interval = 0, nmax = 50)
saveSWF(knn.ani(test = matrix(rnorm(16), ncol = 2),
  cl.pch = c(16, 2)), 1.5, dev = "png",
  para = list(mar = c(3, 3, 1, 1.5),
  mgp = c(1.5, 0.5, 0)), swfname = "kNN.swf")
# shell.exec(normalizePath(file.path(tempdir(), "kNN.swf")))
```

# SVGAnnotation (SVG animations)

- Scalable Vector Graphics (XML)
- SVG supports animations
- Opera supports SVG completely; Firefox partially (unfortunately still does not support animation but hopefully it will in future); IE does not support without additional plug-in
- e.g. the `animate()` function: animating a scatterplot (with points' colors and radii changing)

# FlashMXML (Flash animations)

- Flash is a much more popular format in web pages
- can be created with MXML (XML again!)
- e.g. `mxmlDevice()`: record R graphics in XML and compile with the `mxmlc` utility in Adobe Flex
- seems to be a little far away from animations (currently no functions directly for animations)

# Future Work

- more statistical ideas
- more tools (personally prefer SVG)
- function *saveLatex()* ( $\text{\LaTeX}$  package `animate`)
- rewrite (replace) the messy JavaScript code in HTML animations (use **Dojo**?)
- a formal PDF document as a gallery of animations (using **Sweave** so that the examples can be easily changed)

# Thanks!

- Questions and comments?
- Email: `sprintf("%s@s", "xie", "yihui.name")`
- slides and R code will be available at  
<http://yihui.name/en/vitae> after this talk