Animation with R

Spencer Graves

Abstract

This presentation will describe an "Animate" function in the "Ecfun" package on R-Forge. This function provides a way to create large numbers of image files (in, e.g., png at 25, 30 or 60 frames per second) that can then be mated with audio to produce a movie. The appearance of motion can be obtained with any capability for producing the requisite number of images. The "Animate" function makes it easier in some cases to produce various effects. Examples include a series of plots that add points or change colors as time progresses. This can be used to remix images created elsewhere, e.g., panning and zooming on a photograph or a map can create the illusion of motion. It can be used with other video software such as FFmpeg that supports merging image files with audio. The presentation will also discuss general concepts for functional programming such as parsing, modifying and executing variations of a function. It will also discuss concepts of defensive programming to produce diagnostics to help a user find and fix problems.

Overview

- This is about creating videos / movies,
  - NOT interactive data analysis

Outline:

- Basics of video production
- Functional programming
- Defensive programming

Basics of video production

- Movie =
  - audio +
  - video = stream of image files, e.g, png, at 25 - 60 frames / second
- Example that motivated this development

Challenge / Opportunity

Create a 1 minute YouTube video or filler for television (e.g., public access),

to advertise a blog (or anything else)

using FOSS tools easily accessible to R users

For short videos:

1. Draft narrative
2. Record audio (e.g., with *Audacity*)
3. Time segments
4. Storyboard (slides)
5. Create image files (e.g., png with Animate {Ecfun} in R)
6. Merge images with sound (e.g. with *FFmpeg*)

Alternative Video Software

- Slide show software
  - e.g., MS PowerPoint, LO Impress, Google Slides
  - Record a narrative with the slides

Limited

- FOSS:
  - Blender (3D, *used in open movie projects*)
  - Avidemux
  - ***FFmpeg*** *(command line)*
  - **R to produce image files for, e.g., FFmpeg**

Lots to learn

- Trendalyzer used in Rosling TED talk
- Final Cut Pro: popular Apple product

Animation in R

- animation package:
  - includes simulations of, e.g., CLT …
  - uses FFmpeg
- animate{raster} sequentially plots layers of a RasterStack or RasterBrick
- shiny: sliderInput(..., animate=TRUE, …)
- Animate{Ecfun}
  - Write a function to produce an image
  - Modify arguments so “Animate” produces multiple variants

Animate{Ecfun}

- Animate(plotFn, nFrames, iFrames, …) for (iFrame in iFrames):
- Animate1(plotFn, nFrames, iFrame, …) for each frame
  - plotFn = function or a list of function calls
  - iFrame = which frame
  - nFrames = max allowable iFrame

Example 1

Traverse

```
CRANfn1 <- function(){
    plot(Date, Pkgs, log='y')
    rasterImageAdj(image=Rlogo, xleft=x[iFrame],
        ybottom = y[iFrame+1], xright = x[iFrame+1],
        ytop = y[iFrame])
}
Animate(CRANfn1, nFrames=nFr, iFrames=1:nFr)
```

Example 2

Adjust

Layer
3 = blue
1 = red
2 = green

```
CRANfn2 <- function(){
    plot(Date, Pkgs, log='y')
    Rlg <- Rlogo
    Rlg[,,3] <- ((iFrame/nFrames)*Rlogo[,,3])
    rasterImageAdj(image=Rlg, xleft=x[iFrame], ybottom =
        y[iFrame+1], xright = x[iFrame+1], ytop = y[iFrame],
        angle=ang[iFrame])
}
Animate(CRANfn2, nFrames=nFr, iFrames=1:nFr)
```

Interpolation via optional arguments

- firstFrame, lastFrame:
  - display each observation in this range

- *.0, *.1: Interpolate between
  - Character strings on nchar

Example 3

```
CRANfn3a <- function(){
  plot(firstFrame=seq(1, nFr, length=length(Date)),
    x.0=Date, x.1=x.0, y.0=Pkgs, y.1=y.0, log='y',
    xlim=range(Date), ylim=range(Pkgs),
    main=main5[iFrame])
__ __  }
Animate(CRANfn3a, nFrames=nFr, iFrames=1:
nFr)
```

Animate -> files

```
iFr <- 1:nFr
str(CRANfiles3a <-
    sprintf('../frames/%04dCRANfiles3.png', iFr)

Animate(CRANfn3a, nFrames=nFr,
        iFrames=1:nFr, filenames=CRANfiles3)
```

FFmpeg to combine audio with images

Animate(..., filenames = …, framesFile = '**fs.txt**')

```
ffmpeg -y -i audio.mp2^
        -f concat -i fs.txt^
        -vf fps=25 -pix_fmt yuv420p output.mov
```

Functional Programming

- Ref: Wickham (2014, second section)
- base::body extracts the body of a function
  - Examine, modify the body like a list

- Wickham's 3 chapters on "Functional programming"

Defensive Programming

Wickham (2014):

- section on “Defensive programming”
  - “Fail fast”
  - with informative error messages
  - … may conflict with making the code easy to use -> give the user something sensible when possible

References

Hadley Wikckham (2014) *Advanced R* (http://adv-r.had.co.nz) fortune(298): "Don't do as I say, do as Hadley does."

Paul Murrell (2011) *R Graphics*, 2nd ed. (CRC Press)

install.packages('Ecfun', repos = 'http://R-forge.R-project.org')