Animated plot using animation package

Tian Zheng

October 7, 2015

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Step 1: first you would need to install imageMagick. I used an installer from http://cactuslab.com/imagemagick/ and that worked well for me.

Step 2: load the required libraries and map.

```
require(animation)
```

```
## Loading required package: animation
```

```
## Warning: package 'animation' was built under R version 3.1.3
```

```
require(RColorBrewer)
```

```
## Loading required package: RColorBrewer
```

```
require(maps)
```

```
## Loading required package: maps
```

```
## Warning: package 'maps' was built under R version 3.1.3
```

```
##
##  # ATTENTION: maps v3.0 has an updated 'world' map.        #
##  # Many country borders and names have changed since 1990. #
##  # Type '?world' or 'news(package="maps")'. See README_v3. #
```

```
require(maptools)
```

```
## Loading required package: maptools
## Loading required package: sp
```

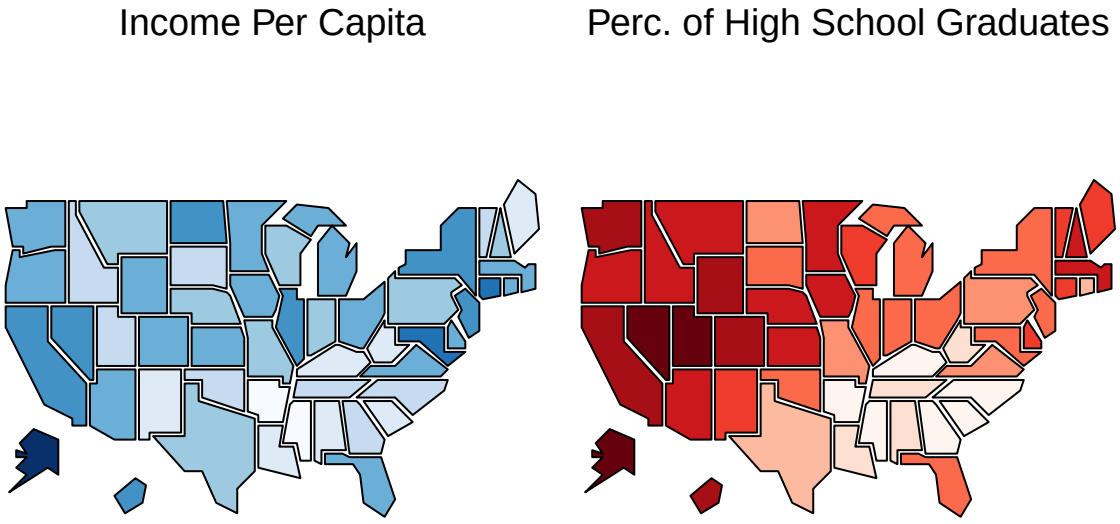
```
## Warning: package 'sp' was built under R version 3.1.3
```

```
## Checking rgeos availability: FALSE
##      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpcl
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

```
data(state.vbm)
```

Step 3: Decide a squence of plots you will make. For this example, we will visualize the variables `Income` and `HS Grad` from the `state.x77` data set.

```
  fg.col="black"
  bg.col="transparent"

  par(mfrow=c(1,2), pty="s", font.main=1,
      bg=bg.col, fg=fg.col,
      col.axis=fg.col, col.main=fg.col,
      col.lab=fg.col)
  par(mar=c(0,0,2,0), mgp=c(0,0,0))

  tmp.x <- state.x77[,'Income']
  tmp2.x <- cut(tmp.x,

  seq(min(tmp.x),max(tmp.x), length.out=10),
              include.lowest=TRUE)

  tmp.y <- state.x77[,'HS Grad']
  tmp2.y <- cut(tmp.y, seq(min(tmp.y),max(tmp.y), length.out=10),
              include.lowest=TRUE)

  col.fg.x=brewer.pal(9,"Blues")[tmp2.x]
  col.fg.y=brewer.pal(9,"Reds")[tmp2.y]

  plot(state.vbm,col=col.fg.x,
         main="Income Per Capita")
  plot(state.vbm,col=col.fg.y,
         main="Perc. of High School Graduates")
```



Step 4: Using `saveGIF` to create an animated version of the above figure.

```
## Set the time delay between each slide
ani.options(interval=.2)
```

```
# Set foreground and background color.
fg.col="black"
bg.col="transparent"

saveGIF({
  #layout(matrix(c(1,2,3,3, 3, 3), 3, 2, byrow = TRUE))
  par(mfrow=c(1,2), pty="s", font.main=1,
      bg=bg.col, fg=fg.col,
      col.axis=fg.col, col.main=fg.col,
      col.lab=fg.col)
  par(mar=c(0,0,2,0), mgp=c(0,0,0))

  tmp.x <- state.x77[,'Income']
  tmp2.x <- cut(tmp.x, seq(min(tmp.x),max(tmp.x), length.out=10),
              include.lowest=TRUE)

  tmp.y <- state.x77[,'HS Grad']
  tmp2.y <- cut(tmp.y, seq(min(tmp.y),max(tmp.y), length.out=10),
              include.lowest=TRUE)

  # a radnom order of states to plot
  order.plot=sample(1:length(tmp.y))

  col.bg=rep(bg.col, length(tmp.y))

  # generate the color of each state depending on the values of
  # x variable and y varialbe.
  col.fg.x=brewer.pal(9,"Blues")[tmp2.x]
  col.fg.y=brewer.pal(9,"Reds")[tmp2.y]

  for(i in 1:length(order.plot)){
    col.use.x=col.bg
    col.use.y=col.bg
    col.use.x[order.plot[1:i]]=col.fg.x[order.plot[1:i]]
    col.use.y[order.plot[1:i]]=col.fg.y[order.plot[1:i]]

    plot(state.vbm,col=col.use.x,
         main="Income Per Capita")
    plot(state.vbm,col=col.use.y,
         main="Perc. of High School Graduates")

#     par(mar=c(5,4,2,1), mgp=c(3,2,1))
#
#     plot(state.x77[, "Income"], state.x77[,"HS Grad"],
#          xlab="Income",
#          ylab="Perc. of high school graduate",
#          type="n")
#     text(state.x77[order.plot[1:i], "Income"],
#          state.x77[order.plot[1:i],"HS Grad"],
#          state.abb[order.plot[1:i]])
  }
}, movie.name="maps1.gif", ani.width=600, ani.height=300)
```

```
## Executing:
## 'convert' -loop 0 -delay 20 Rplot1.png Rplot2.png Rplot3.png
##     Rplot4.png Rplot5.png Rplot6.png Rplot7.png Rplot8.png
##     Rplot9.png Rplot10.png Rplot11.png Rplot12.png Rplot13.png
##     Rplot14.png Rplot15.png Rplot16.png Rplot17.png Rplot18.png
##     Rplot19.png Rplot20.png Rplot21.png Rplot22.png Rplot23.png
##     Rplot24.png Rplot25.png Rplot26.png Rplot27.png Rplot28.png
##     Rplot29.png Rplot30.png Rplot31.png Rplot32.png Rplot33.png
##     Rplot34.png Rplot35.png Rplot36.png Rplot37.png Rplot38.png
##     Rplot39.png Rplot40.png Rplot41.png Rplot42.png Rplot43.png
##     Rplot44.png Rplot45.png Rplot46.png Rplot47.png Rplot48.png
##     Rplot49.png Rplot50.png 'maps1.gif'
## Output at: maps1.gif
```

The final product