Exploring Data Visualization and Graphics Tools in R

Stat 133, Fall 2017

Iris Yon

October 26, 2017

Table of Contents

Introduction

Going beyond the basic built-in graphing tools, the purpose of this post is to explore different data visualization tools available to best represent given data. This post mainly focuses on three techniques for data visualzation: image transformation, creating gifs, and plotting data through different frameworks. From manipulation and handling of images to animating plots, this post includes the many ways data can be processed and visualized in R.

Code

Load All Libraries

```
# library for animation, description in references
library("magick")
```

```
## Warning: package 'magick' was built under R version 3.4.2
```

```
## Linking to ImageMagick 6.9.9.14
## Enabled features: cairo, freetype, fftw, ghostscript, lcms, pango, rsvg, webp
## Disabled features: fontconfig, x11
```

```
# Get data
library(gapminder)
# ggplot library
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```r
#devtools::install_github("dgrtwo/gganimate")
library(gganimate)
# Charge the circlize library
library(circlize)
```

```
## Warning: package 'circlize' was built under R version 3.4.2
```

Load Images

```r
# Animated Stat133 Sign
stat133 <- image_read("../images/ucb_stat133.gif")
stat133 <- image_scale(stat133, "500")
stat133 <- image_animate(image_join(stat133))
image_write(stat133, "../images/ucb_stat133.gif")


# Static Image of Stat133 Sign
static_stat133 <- image_read("../images/ucb_image.png")
image_write(static_stat133, "../images/static_stat133.png")
```

1. Transform Images

```r
img_blur <- image_blur(static_stat133, 10, 5)
image_write(img_blur, "../images/img_blur.png")

img_noise <- image_noise(static_stat133)
image_write(img_noise, "../images/img_noise.png")

img_charcoal <- image_charcoal(static_stat133)
image_write(img_charcoal, "../images/img_charcoal.png")

img_oilpaint <- image_oilpaint(static_stat133)
image_write(img_oilpaint, "../images/img_oilpaint.png")

img_emboss <- image_emboss(static_stat133)
image_write(img_emboss, "../images/img_emboss.png")

img_edge <- image_edge(static_stat133)
image_write(img_edge, "../images/img_edge.png")

img_negate <- image_negate(static_stat133)
image_write(img_negate, "../images/img_negate.png")
```

2. Make Animated ggplot

```r
# Makes animated ggplot for life expectantcy of each continent over time
img <- image_graph(600, 400, res = 96)
datalist <- split(gapminder, gapminder$year)
out <- lapply(datalist, function(data){
  p <- ggplot(data, aes(continent, lifeExp, size = pop, color = continent)) +
    scale_size("population", limits = range(gapminder$pop)) + geom_point() + ylim(20, 90) +
    ggtitle(data$year) + theme_classic() + labs(y = "Life Expectantcy", x = "Continent")
  print(p)
})
dev.off()
```

```
## png
##   2
```

```r
img <- image_background(image_trim(img), 'white')
ani <- image_animate(img, fps = 2)

image_write(ani, "../images/graph.gif")
```

```r
# makes ggplot of gdp per capita vs life expectantcy
img <- image_graph(600, 400, res = 96)
datalist <- split(gapminder, gapminder$year)
out <- lapply(datalist, function(data){
  p <- ggplot(data, aes(gdpPercap, lifeExp, color = continent)) +
    geom_point() + ylim(20, 90) +
    scale_x_log10(limits = range(gapminder$gdpPercap)) + ggtitle(data$year) + theme_classic() + labs(y = "Life Exp
ectantcy", x = "GDP Per Capita")
  print(p)
})
dev.off()
```

```
## png
##   2
```

```
img <- image_background(image_trim(img), 'white')
ani <- image_animate(img, fps = 2)

image_write(ani, "../images/graph2.gif")
```

3. Circle Graph

```
# continent vs year in circlar data visualization tool
name=c(gapminder$continent)
feature= c(gapminder$year)
dat <- data.frame(name,feature)
dat <- with(dat, table(name, feature))

png("../images/circ_plot.png")
# Make the circular plot
circplot <- chordDiagram(as.data.frame(dat), transparency = 0.5)
dev.off()
```

```
## png
##   2
```

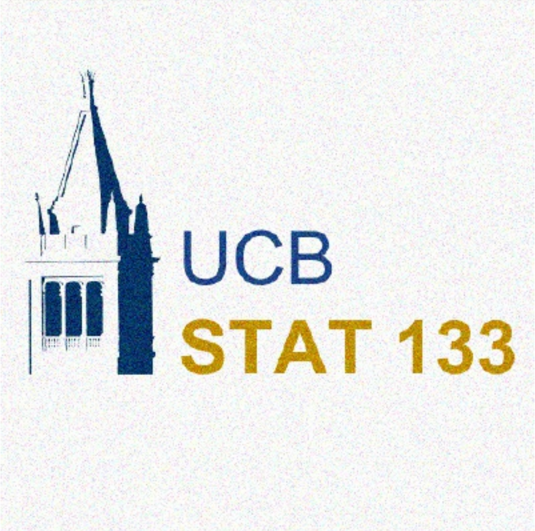
Graphics

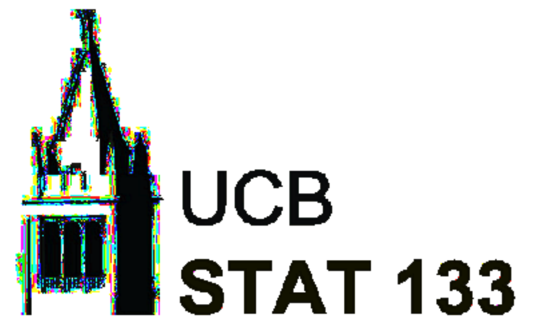
Original Image

UCB
STAT 133

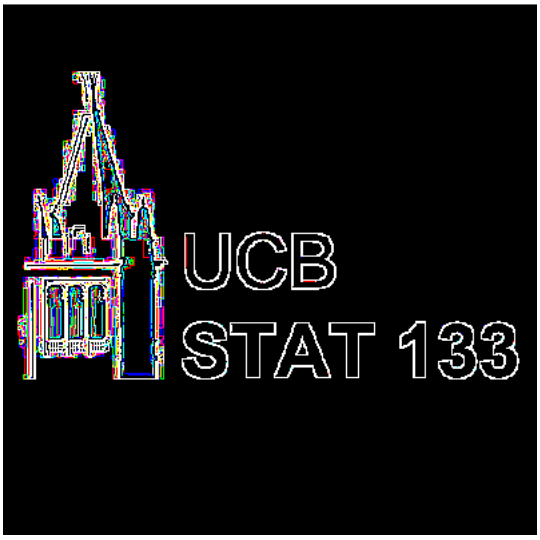
1. **Transformations**

UCB
STAT 133





UCB
STAT 133

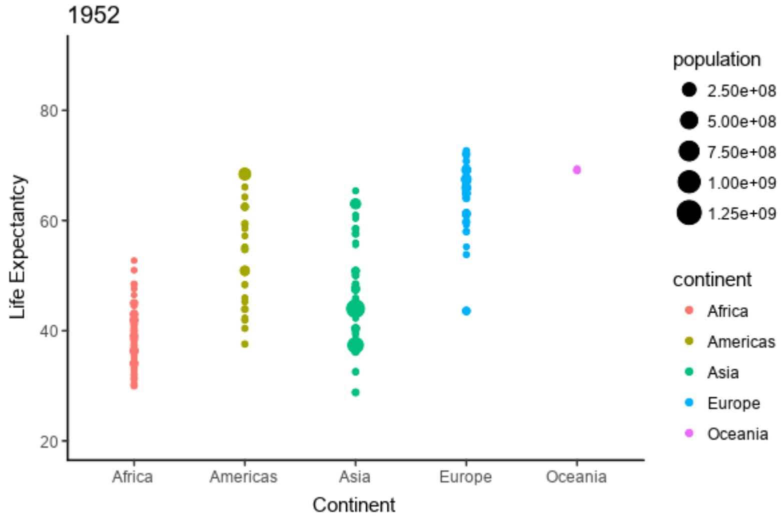


UCB
STAT 133

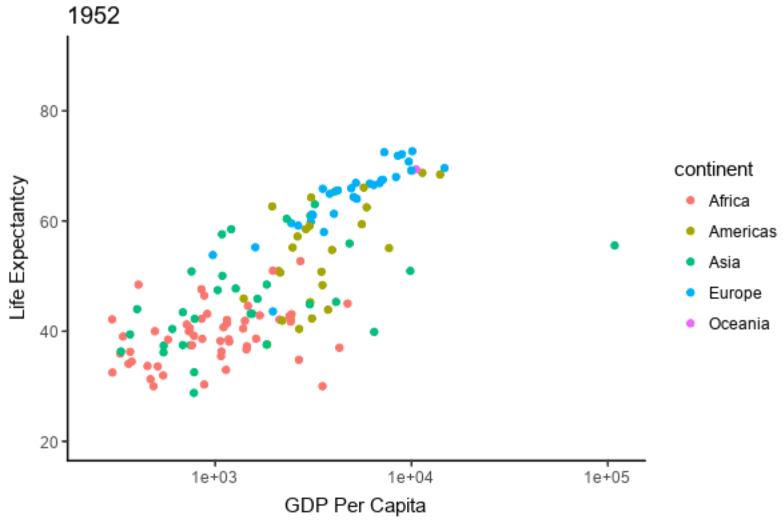
UCB
STAT 133

2. **Animated Graphs**

Life Expectantcy of Country Over Time

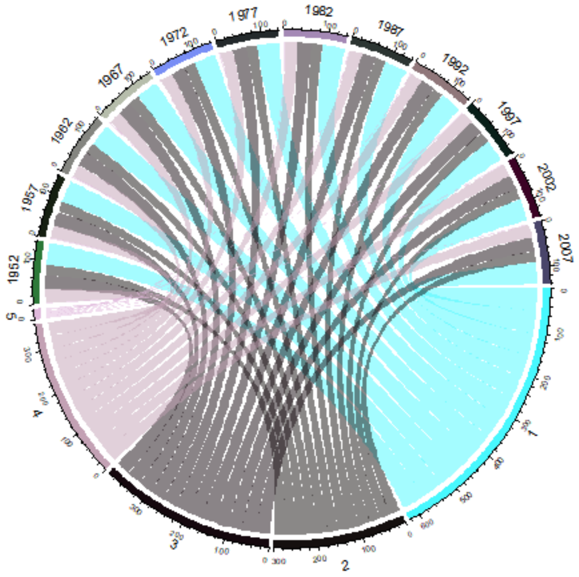


GDP Per Capita vs Life Expectantcy Over Time



3. **Circlized Graph**

Continents vs Years



Message & Discussion

Using data packages such as gapminder to visualize real world data and its implications, the different data visualization techniques such as magick and circlize provide an effective way to understand and process the information.

Therefore, although it is sufficient to use basic graphing tools to understanding data, the way the information is represented is almost as important as the data itself. Providing clear and engaging methods to capturing important statistics allow for audience to best interact with the data.

Conclusion

In the process of making this post, it was clear that I had only scratched the surface of the many data visualization tools available in conjunction with R. It seems that for each unique set of data, there exists a number of methods that could accommodate the user and the data itself to yield great results. In the end, it’s up to the data analyst to prepare the information accordingly and effectively.

References

1. Gapminder Library

- gapminder is a library that has data and statistics on global developments (ie. population growth)
- includes documentation and description of tools for gapminder library

2. R Graph Gallery

- a collection of crowd sourced graphs/projects in R

3. Data Visualization Packages

- article about best libraries and packages for data visualizations in R
- includes uses, examples, short description of each R package

4. Magick Package

- allow for animation of images or lists
- substitute for gganimate

5. Magick Functions

- different magick functions and capabilities to manipulating data/images

6. Circlize Functions

- different circlize functions that go into making the circular plots

7. Circlize Graph Reference

- inspiration for the circular plot included in this post