Post 01: Data Visualization and Graphics

Lisa Zhou

Introduction

R has the ability to create many different customizable data visualizations and graphics, because of the various packages that are constantly being developed. ggplot2 is package that creates very customizable graphs using many different layers. The purpose of this post is to explore the different ways of creating graphs with ggplot2 and also focus on making map visualizations. The post will go through different interesting examples of using the external packages ggplot2 and maps to create interactive graphics in R.

Motivation

After working briefly with ggplot2 to create graphs, I was amazed at all the different visualizations encompassed in the package. With one package, we gained the ability to create beautiful scatterplots, boxplots, bar charts among other visualizations yet to be explored such as bubble charts, violin plots, heat maps and many more. As I was researching ggplot2, I discovered the extension maps, which allows you to create maps to visualize trends and data worldwide.

Background

The package ggplot2 was created by Hadley Wickham in 2005 as a way convenient way to customize graphics. The maps package created by Richard A. Becker and Allan R. Wilks allows us to draw geographical maps. The package maps provides lots of different map outlines and ways to use maps.

Installing and loading packages

First, we have to install the gganimate package from github and load packages.

```
# installing the package gganimate using devtools
install.packages("maps")
```

```r
#load packages to use
library(ggplot2)
library(maps)
library(dplyr)
```

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

```
##
## Attaching package: 'dplyr'
```

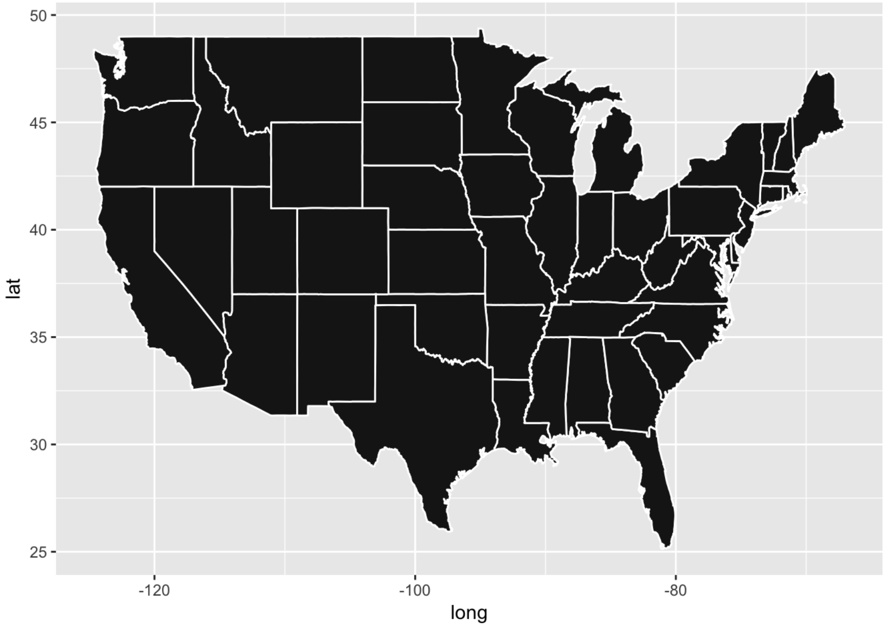
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Basic Graphic Examples

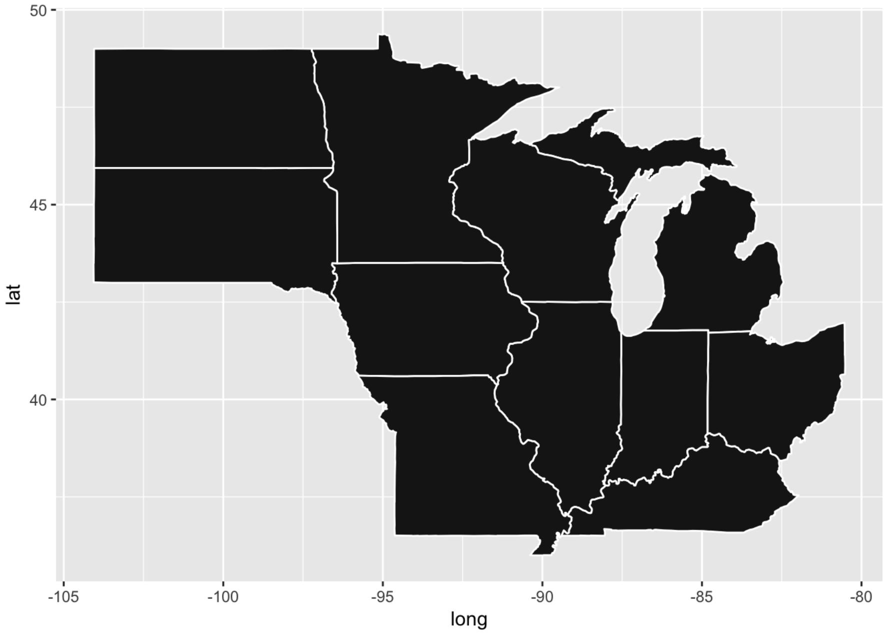
Let's start off by creating a basic example of a map. Let's start by creating a visualization of the US map.

```r
library(ggplot2)
library(maps)
#load us map data
all_states <- map_data("state")
#plot all states with ggplot
p <- ggplot()
p <- p + geom_polygon( data=all_states, aes(x=long, y=lat, group = group),colour="white", fill="grey10" )
p
```



We can create subset of states or just plot one state all well.

```
#create subset of midwest states
states <- subset(all_states, region %in% c( "illinois", "indiana", "iowa", "kentucky", "michigan", "minnesota","missouri", "north dakota", "ohio", "south dakota", "wisconsin" ) )
p <- ggplot()
p <- p + geom_polygon( data=states, aes(x=long, y=lat, group = group),colour="white", fill="grey10" )
p
```



Visualization Using ggplot

To explore the possibilities to create beautiful grahs, we are going to explore a simple example using the gapminder data set available in Jenny Bryan's gapminder package. The gapminder data set provides values for life expectancy, GDP per capita, population, every five years from 1952 to 2007 for 147 different countries.

```
# install package
install.packages("gapminder")
```

```
#load and inspect the data
library(gapminder)
head(gapminder)
```

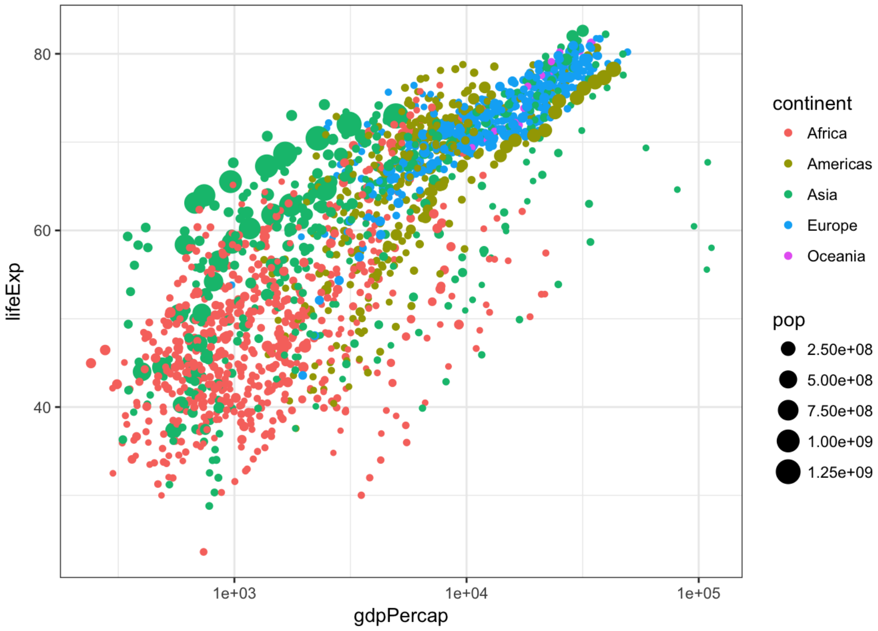
```
## # A tibble: 6 x 6
##       country continent  year lifeExp      pop gdpPercap
##        <fctr>    <fctr> <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan      Asia  1952  28.801  8425333  779.4453
## 2 Afghanistan      Asia  1957  30.332  9240934  820.8530
## 3 Afghanistan      Asia  1962  31.997 10267083  853.1007
## 4 Afghanistan      Asia  1967  34.020 11537966  836.1971
## 5 Afghanistan      Asia  1972  36.088 13079460  739.9811
## 6 Afghanistan      Asia  1977  38.438 14880372  786.1134
```

Let's use ggplot to create a color coded scatterplot for the gapminder data. In this example, we are going to see how ggplot can be used to create extensive, custom graphs.

```
# Basic scatter plot
theme_set(theme_bw())

p <- ggplot(gapminder, mapping = aes(x = gdpPercap,
              y = lifeExp,
              size = pop, color = continent)) +
  geom_point() +
  scale_x_log10()

p
```



Plotting Maps

Let's explore an example of producing a map, where the size of a point on a location indicates the amount of Twitter followers for that group of Twitter users.

```
#preparing the data
library(readr)
library(dplyr)

url_csv <- 'https://raw.githubusercontent.com/d4tagirl/R-Ladies-growth-maps/master/rladies.csv'
rladies <- read_csv(url(url_csv)) %>%
  select(-1)
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   screen_name = col_character(),
##   location = col_character(),
##   created_at = col_date(format = ""),
##   followers = col_integer(),
##   age_days = col_double(),
##   lon = col_double(),
##   lat = col_double()
## )
```

```
library(DT)

datatable(rladies, rownames = FALSE,
          options = list(pageLength = 5))
```

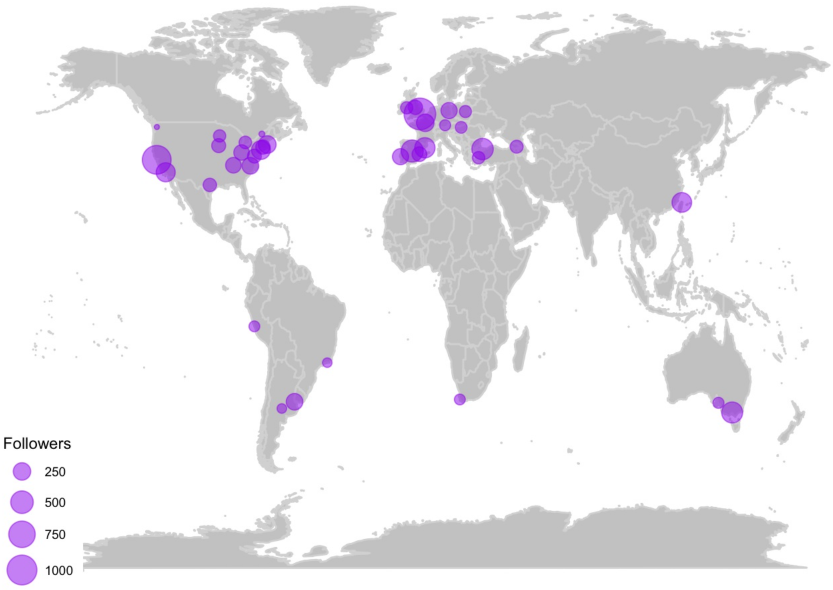
After loading the data and organizing it into a dataframe, we are going to first create a map using ggplot.

```
# plot the map using ggplot
library(ggplot2)
library(maps)
library(ggthemes)

#creating a gray map
world <- ggplot() +
  borders("world", colour = "gray85", fill = "gray80") +
  theme_map()

#adding points on the map to represent Twitter followers
map <- world +
  geom_point(aes(x = lon, y = lat, size = followers),
             data = rladies,
             colour = 'purple', alpha = .5) +
  scale_size_continuous(range = c(1, 8),
                        breaks = c(250, 500, 750, 1000)) +
  labs(size = 'Followers')

map
```



This map provides a visualization depicting locations in the world where groups of users have more Twitter followers. The size of the dots represent the number of followers. A larger dot means more followers.

Conclusion

There are many powerful visualizations that R is able to create from external packages. The maps package provides a tool to visualize data for the world in one compact set.

References

1. http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html 2.http://eriqande.github.io/rep-res-web/lectures/making-maps-with-R.html 3.https://cran.r-project.org/web/packages/maps/index.html 4.https://github.com/jennybc/gapminder https://uchicagoconsulting.wordpress.com/tag/r-ggplot2-maps-visualization/ 6.http://relevantmisc.com/ggplot/instagram/2016/02/14/ig-over-time/ 7.https://d4tagirl.com/2017/05/how-to-plot-animated-maps-with-gganimate
2. https://cran.r-project.org/web/packages/gapminder/index.html