Statistical Methods for Data Science
CS 6313.001: Mini Project #2

Due on Thursday February 16, 2017 at 4pm

Instructor: Pankaj Choudhary

Hanlin He / Lizhong Zhang (hxh160630 / lxz160730)

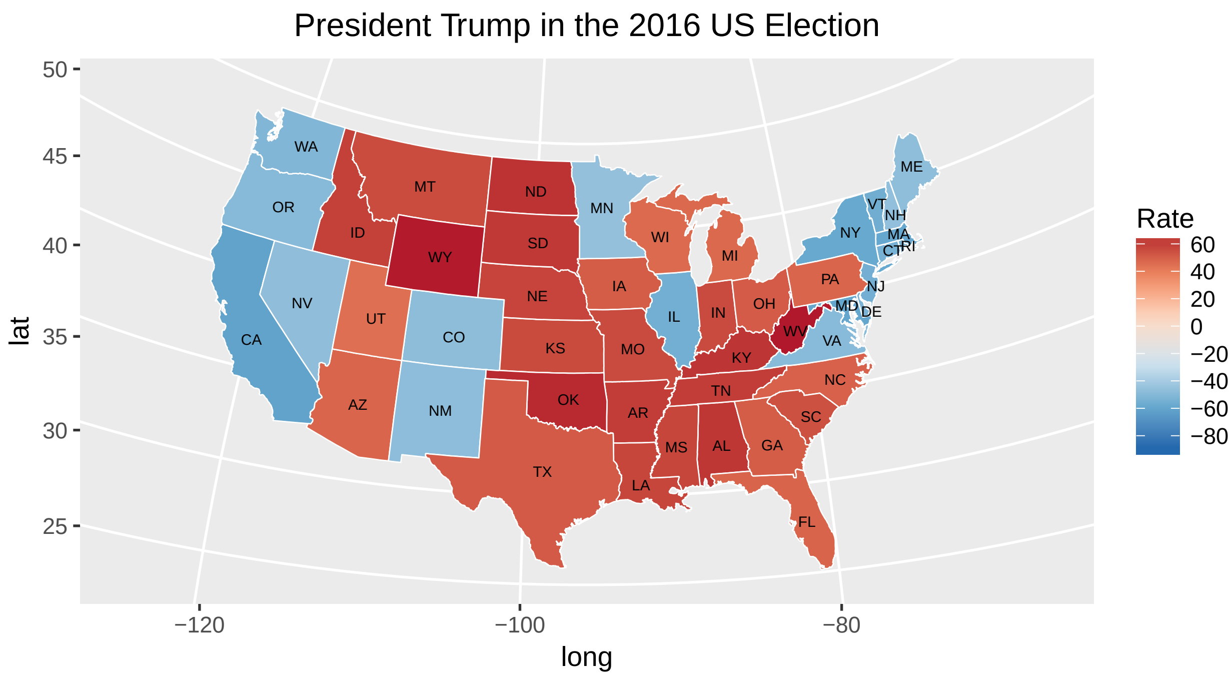
Contribution

Both team members made the same contribution in this project.

Section 1 The Map

The first approach to draw the map is shown in fig. 1. The color is based on the “RdBu” palette of *ggplot*. However, since Alaska and Hawaii is not in the map data, mapping them independently may result in inconsistency in color. Therefore, these two states are not shown in this version of the map.

Figure 1: The First Approach

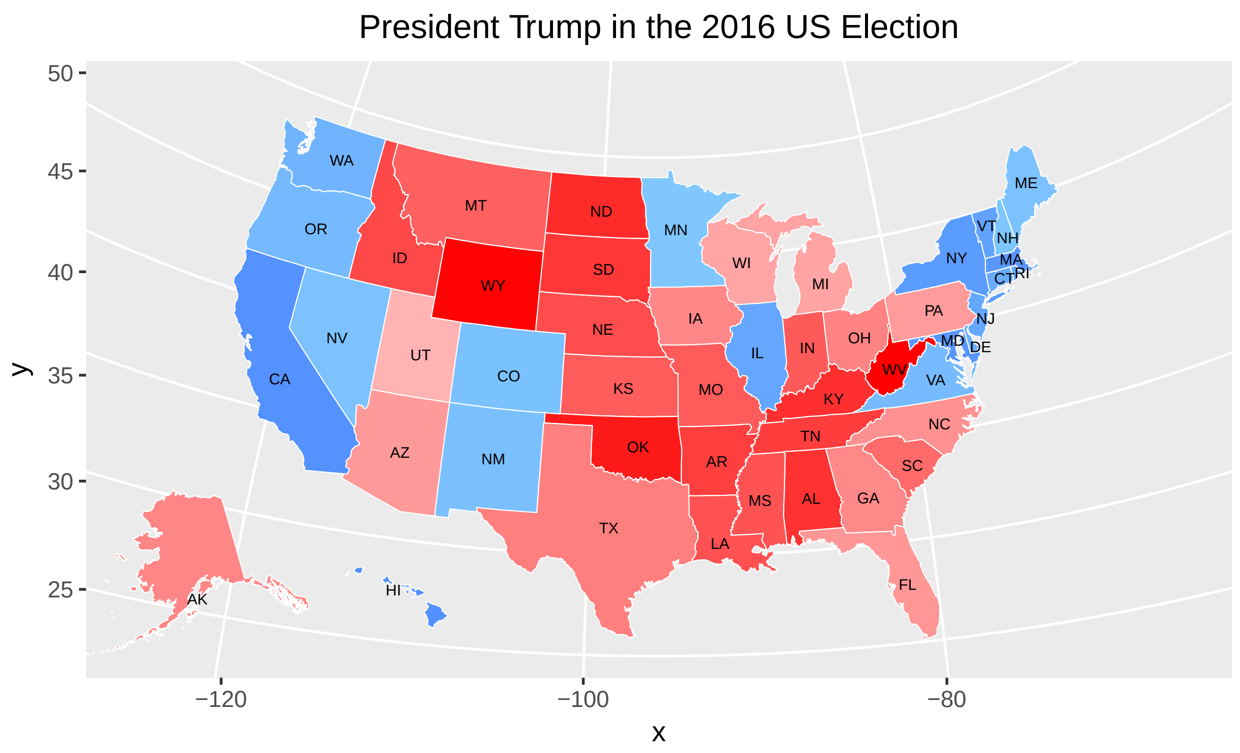


Based on the above map, it uses shades of red for Donald Trump and shades of blue for Hillary Clinton. And due to different vote percentages, we know that with the increasing of vote percentage, the colors will be deeper, i.e. for Donald Trump, he won in Texas and Arkansas, so the color of Texas and Arkansas in this map is Red, but vote percentage in Arkansas is bigger than in Texas, so that color of Arkansas is deeper.

The second approach to draw the map is shown in fig. 2. To solve the problem in fig. 1, we introduced the library of *mapdata*. And to make the color brighter, and to get rid of the predefine “RdBu” palette, we directly mapped the support rates into RGB color. In this way, making the color of Alaska and Hawaii consistent with the other states. And finally, we used a *layout* function, joined the plots together into one single graph.

However, the second approach still brought in new problem. Since the color was directly mapped from rates to RGB, no mapping was used in the **aes**. Thus, the guide of legend failed to generate.

Figure 2: The Second Approach



Section 2 R Code

```
############################################################################
# Basic data operation
############################################################################

library(maps)
library(mapdata)
library(ggplot2) # for plotting and miscellaneuous things
library(ggmap) # for plotting
library(plyr) # for merging datasets

##
## Attaching package: 'plyr'
## The following object is masked from 'package:maps':
##
##     ozone

# Fetch map data for the States.
usa <- map_data("state")
# Fetch map data for Alaska.
```

```r
alaska <- map_data("world2Hires", "USA:Alaska")
# Fetch map data for Hawaii.
hawaii <- map_data("world2Hires", "Hawaii")

# Change states names into lower case.
hawaii$region <- tolower(hawaii$region)
alaska$subregion <- tolower(alaska$subregion)

# Remove useless column and unify column names.
usa$subregion <- NULL
hawaii$subregion <- NULL
alaska$region <- NULL
colnames(alaska)[5] <- "region"

# Fetch abbreviations for each states.
statesname <- data.frame(state.center, state.abb)

# Unify column names.
colnames(statesname)[3] <- "region"

# Extract abbr for Hawaii and Alaska, modify the coordinates.
hiname <- statesname[statesname$region == "HI",]
hiname[1,1] <- mean(hawaii[,1])
hiname[1,2] <- mean(hawaii[,2])
akname <- statesname[statesname$region == "AK",]
akname[1,1] <- mean(alaska[,1])
akname[1,2] <- mean(alaska[,2])

# Remove Hawaii and Alaska from list.
statesname <- statesname[statesname$region != "HI",]
statesname <- statesname[statesname$region != "AK",]

# Fetch pool data.
data <- read.table("us_2016_election_data.csv", header = T, sep = ",")

# Convert "xx%" to float number.
data[,2] <- as.numeric(sub("%","",data[,2]))
data[,3] <- as.numeric(sub("%","",data[,3]))

# Change states names into lower case.
data$State <- tolower(data$State)
```

```
data[,5] <- apply(data, 1,
    (function(a) if(a[2]>a[3]) -as.numeric(a[2]) else as.numeric(a[3])))

# Modify column for convenience.
colnames(data)[2] <- "Clinton"
colnames(data)[3] <- "Trump"
colnames(data)[1] <- "region"
colnames(data)[5] <- "Rate"

# Extract data for Hawaii and Alaska, and remove them from main list.
hdata <- data[data$region == "hawaii",]
adata <- data[data$region == "alaska",]
data  <- data[data$region != "hawaii",]
data  <- data[data$region != "alaska",]

# Join the data with map.
usa.df <- join(usa, data, by = "region", type = "inner")
alaska.df <- join(alaska, data, by = "region", type = "inner")
hawaii.df <- join(hawaii, data, by = "region", type = "inner")

############################################################################
# Approach 1
############################################################################

# Generate map for States.
state.maps <- ggplot() +
    geom_polygon(data  = usa.df,
                 aes(x = long,
                     y = lat,
                     group = group,
                     fill = Rate),
                 color = "white",
                 size = 0.25) +
    scale_fill_distiller(palette = "RdBu",
                         breaks = seq(-80, 80, by = 20)) +
    geom_text(data = statesname,
              aes(x = x, y = y, label = region),
              size=2.0) +
    coord_map("polyconic") +
    ggtitle("President Trump in the 2016 US Election") +
    theme(plot.title = element_text(hjust = 0.5))

state.maps
```

```
############################################################################
# Approach 2
############################################################################

# Generate map for the States except Hawaii and Alaska.
# Conduct 'ggplot' on 'data' based on 'region'.
# To generate map for States, call: gMap(data, usa, statesname)
# To generate map for Hawaii, call: gMap(hdata, hawaii, hiname)
# To generate map for Alaska, call: gMap(adata, alaska, akname)
gMap <- function(data, states, statesname) {
    maps <- ggplot(data, aes(map_id = region))
    # Add map for each state in data iteratively.
    for (i in 1:dim(data)[1]) { # for each state in 'data'
        # Extract the target state.
        targetstate <- states[states$region == data[i,]$region,]
        # Depends on who won, specify the color of the state in RGB.
        if(data[i,]$Clinton > data[i,]$Trump) {
            r <- (1   - ((data[i,]$Clinton/100 - 0.464)/0.445))*0.5
            g <- (1.3 - ((data[i,]$Clinton/100 - 0.464)/0.445))*0.6
            b <- 1
        } else {
            r <- 1
            g <- (1 - ((data[i,]$Trump/100 - 0.455)/0.231))*0.7
            b <- (1 - ((data[i,]$Trump/100 - 0.455)/0.231))*0.7
        }
        # Add map with color filling.
        maps <- maps +
            geom_map(fill   = rgb(r,g,b),
                     color  = 'white',
                     size   = 0.2,
                     map    = targetstate) +
            expand_limits(x = targetstate$long,
                          y = targetstate$lat)
    }
    # Add some more information to the map.
    maps <- maps +
        # Add abbreviations for each state.
        geom_text(data  = statesname,
                  aes(x = x,
                      y = y,
                      label = region),
                  size  = 2.0) +
```

```r
        # Make map shape more realistic.
        coord_map("polyconic")
    # Return the result
    return(maps)
}

# This function was used to combine three generated map into one file,
# coming from an early version of https://github.com/cran, its url is
# https://github.com/cran/wq/blob/8223da687d8daff2ad612f9a07926f412a08ba82/R/layOut.R
layOut = function(...) {
     require(grid)

     x <- list(...)
     n <- max(sapply(x, function(x) max(x[[2]])))
     p <- max(sapply(x, function(x) max(x[[3]])))
     pushViewport(viewport(layout = grid.layout(n, p)))

     for (i in seq_len(length(x))) {
         print(x[[i]][[1]], vp = viewport(layout.pos.row = x[[i]][[2]],
             layout.pos.col = x[[i]][[3]]))
     }
}

# This function calls the previous gMap function to generate three maps
# from data, and then use layOut function to combine them together.
print_map <- function() {
    layOut(
        list(gMap(data, usa, statesname) +
            ggtitle("President Trump in the 2016 US Election") +
            theme(plot.title = element_text(hjust = 0.5)), 1:200, 10:300),
        list(gMap(hdata, hawaii, hiname)+theme_nothing(), 90:160, 10:120),
        list(gMap(adata, alaska, akname)+theme_nothing(), 90:170, 10:90))
}

print_map()
```