

Mapas

Hanna Rodrigues Ferreira

13 junho,2021

Vamos reproduzir os resultados do capítulo 7 do livro **Data Visualization - A practical Introduction**. Os dados utilizados foram retirados deste github.

```
knitr::opts_chunk$set(echo = TRUE)

library(gapminder)
library(here)
library(tidyverse)
library(ggrepel)
library(socviz)
library(knitr)
library(statebins)
```

Choropleth maps mostra geograficamente regiões coloridas de acordo com certa variável. Além de Choropleth maps veremos outras formas de representar dados neste tipo na ggplot2.

Mapa dos EUA

Utilizaremos uma amostra dos dados de eleição presidencial dos EUA em 2016, ela possui diversas medidas do voto pela região e estado

```
election %>% select(state,
                      total_vote,
                      r_points,
                      pct_trump,
                      party,
                      census) %>%
  sample_n(5)

## # A tibble: 5 x 6
##   state      total_vote  r_points pct_trump party      census
##   <chr>        <dbl>     <dbl>     <dbl> <chr>      <chr>
## 1 New Jersey  3906723    -14.0      41   Democratic Northeast
## 2 New Hampshire 744296    -0.370     46.5 Democratic Northeast
## 3 Louisiana    2029032     19.6      58.1 Republican South
## 4 Oregon       2001336    -11.0      39.1 Democratic West
## 5 Virginia     3982752     -5.32     44.4 Democratic South
```

Antes de partir diretamente para a representação por mapas, podemos começar com um dotplot do resultado das eleições dos estados divididos pela região.

```

# cores : Democratas azul e Republicanos vermelho
party_colors <- c("#2E74C0", "#CB454A")

p0 <- ggplot(data = subset(election,
                           st %nin% "DC"),
              mapping = aes(x = r_points,
                            y = reorder(state, r_points),
                            color = party))

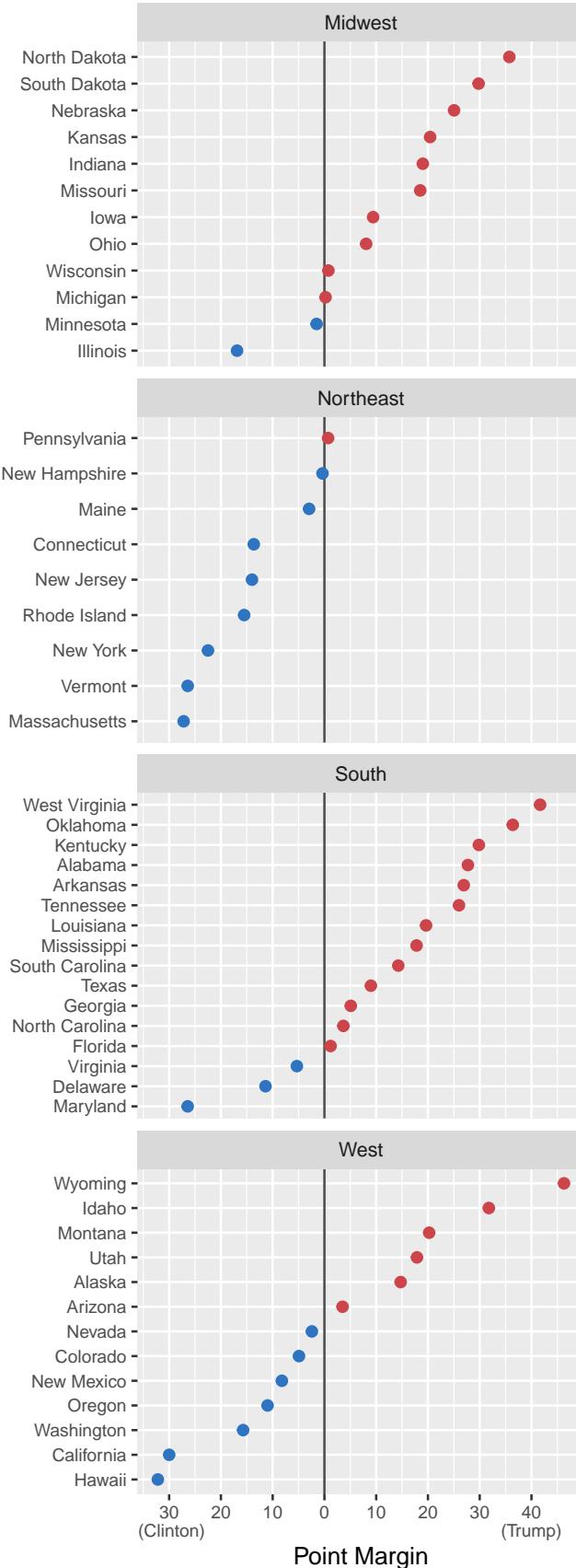
p1 <- p0 +
  geom_vline(xintercept = 0,
             color = "gray30") +
  geom_point(size = 2)

p2 <- p1 + scale_color_manual(values = party_colors)

p3 <- p2 + scale_x_continuous(breaks = c(-30, -20, -10, 0, 10, 20, 30, 40),
                               labels = c("30\n(Clinton)", "20", "10", "0",
                                         "10", "20", "30", "40\n(Trump)"))

p3 + facet_wrap(~ census, ncol=1, scales="free_y") +
  guides(color=FALSE) +
  labs(x = "Point Margin", y = "") +
  theme(axis.text=element_text(size=8))

```



Agora vamos fazer uso do pacote maps que nos dará acesso a alguns dados de mapa pré-desenhados.

```
library(maps)

us_states <- map_data("state")
head(us_states)

##      long     lat group order  region subregion
## 1 -87.46201 30.38968     1     1 alabama    <NA>
## 2 -87.48493 30.37249     1     2 alabama    <NA>
## 3 -87.52503 30.37249     1     3 alabama    <NA>
## 4 -87.53076 30.33239     1     4 alabama    <NA>
## 5 -87.57087 30.32665     1     5 alabama    <NA>
## 6 -87.58806 30.32665     1     6 alabama    <NA>

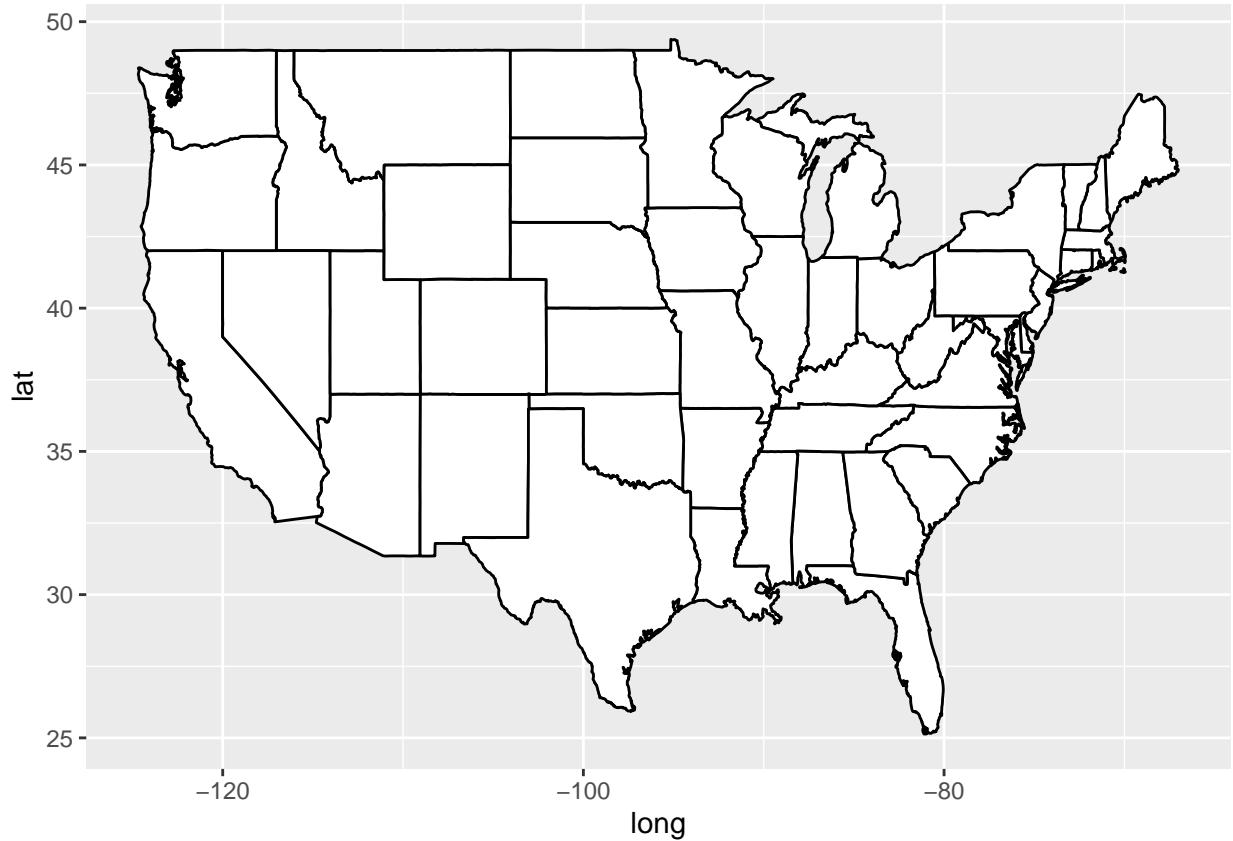
dim(us_states)

## [1] 15537     6
```

Perceba o tamanho deste dataframe, é necessário muitas linhas para se desenhar com robustez um mapa. Podemos utilizar estes dados para plotar um mapa ainda em branco através da função geom_polygon().

```
p <- ggplot(data = us_states,
             mapping = aes(x = long,
                           y = lat,
                           group = group))

p + geom_polygon(fill = "white",
                  color = "black")
```



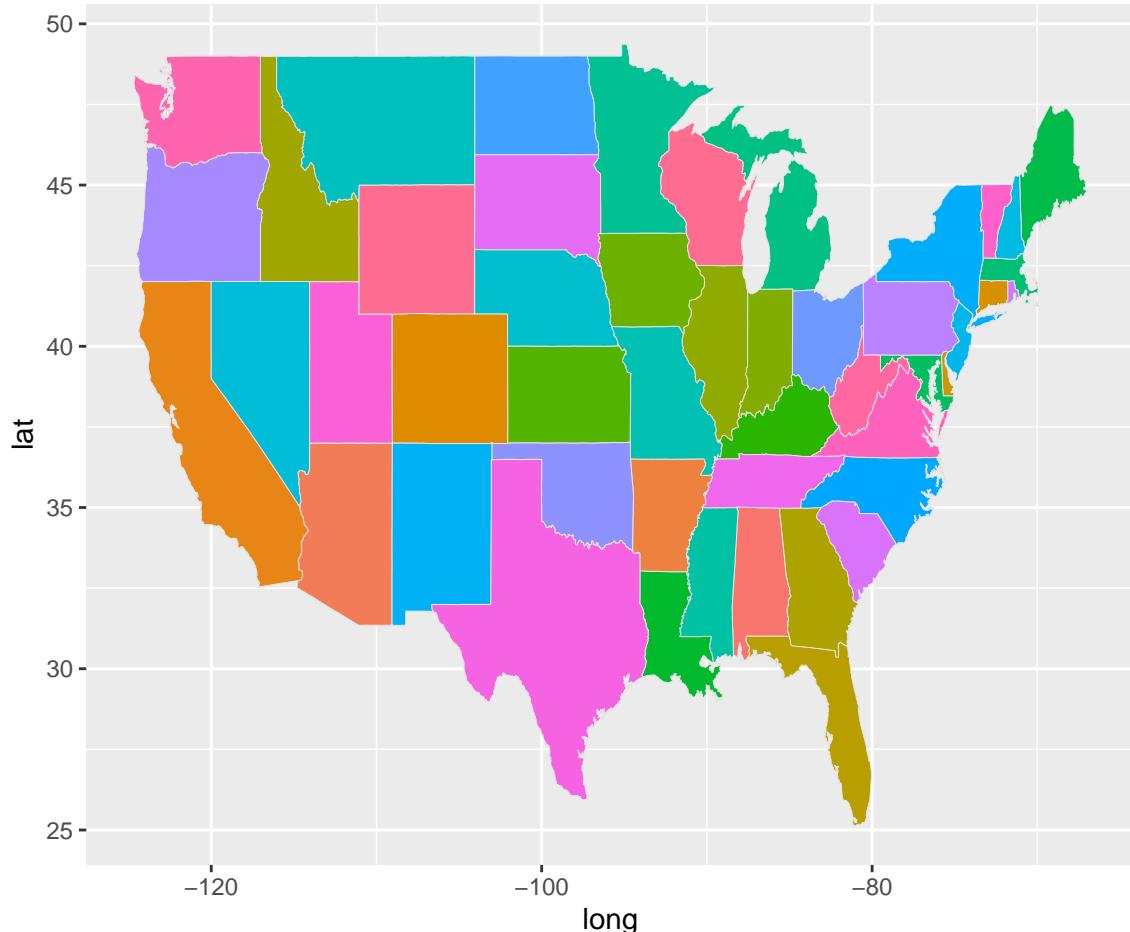
Podemos adicionar cor aos estados definindo a variável `region` ao `fill` e às fronteiras redefinindo `color` e `size`.

```
library('mapproj')

## Warning: package 'mapproj' was built under R version 4.0.5

p <- ggplot(data = us_states,
            aes(x = long,
                y = lat,
                group = group,
                fill = region))

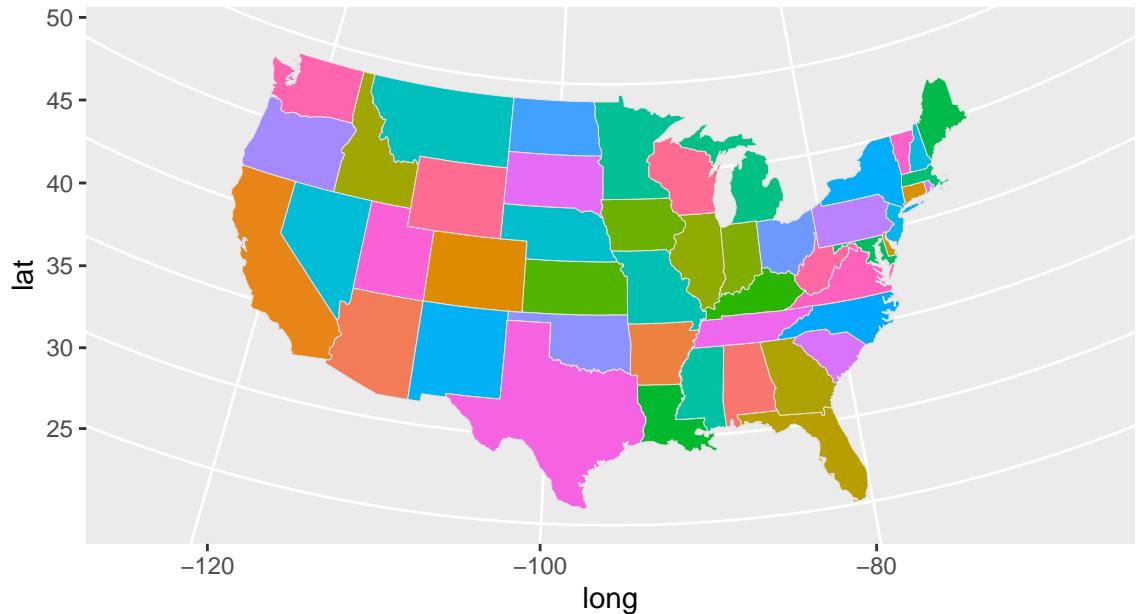
p + geom_polygon(color = "gray90",
                  size = 0.1) +
  guides(fill = FALSE)
```



Por padrão a projeção do mapa é “mercator”, existem diversos tipos de técnicas cartográficas para projetar mapas e optaremos por redefinir a projeção e seus parâmetros para a “albers” através da função coord_map().

```
p <- ggplot(data = us_states,
             mapping = aes(x = long,
                           y = lat,
                           group = group,
                           fill = region))

p + geom_polygon(color = "gray90", size = 0.1) +
  coord_map("albers", lat0 = 39, lat1 = 45) +
  guides(fill = FALSE)
```



Agora vamos plotar nossos dados no mapa, faremos um merge dos dados do pacote maps com nossos dados das eleições.

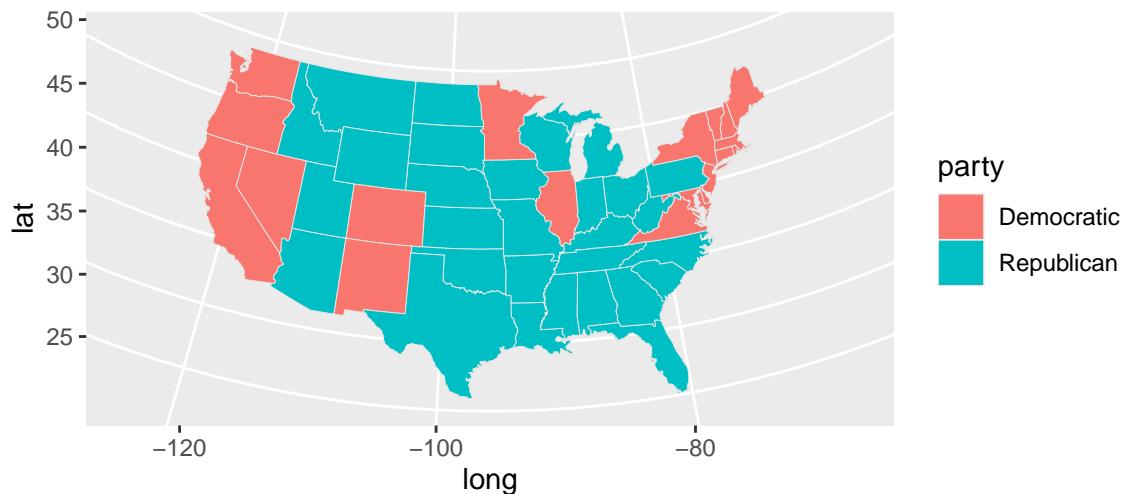
```
election$region <- tolower(election$state)
us_states_elec <- left_join(us_states, election)

## Joining, by = "region"
```

Com os dados compilados portanto plotar o mapa colorindo segundo o partido.

```
p <- ggplot(data = us_states_elec,
             aes(x = long,
                 y = lat,
                 group = group,
                 fill = party))

p + geom_polygon(color = "gray90", size = 0.1) +
  coord_map(projection = "albers", lat0 = 39, lat1 = 45)
```



Adicionando certas configurações ao tema, como as cores pre-definidas anteriormente e remover o grid.

```
theme_map <- function(base_size=9, base_family="") {
  require(grid)
  theme_bw(base_size=base_size, base_family=base_family) %+replace%
    theme(axis.line=element_blank(),
          axis.text=element_blank(),
          axis.ticks=element_blank(),
          axis.title=element_blank(),
          panel.background=element_blank(),
          panel.border=element_blank(),
          panel.grid=element_blank(),
          panel.spacing=unit(0, "lines"),
          plot.background=element_blank(),
          legend.justification = c(0,0),
          legend.position = c(0,0)
        )
}
p0 <- ggplot(data = us_states_elec,
              mapping = aes(x = long,
                            y = lat,
                            group = group,
```

```

      fill = party))

p1 <- p0 + geom_polygon(color = "gray90", size = 0.1) +
  coord_map(projection = "albers", lat0 = 39, lat1 = 45)

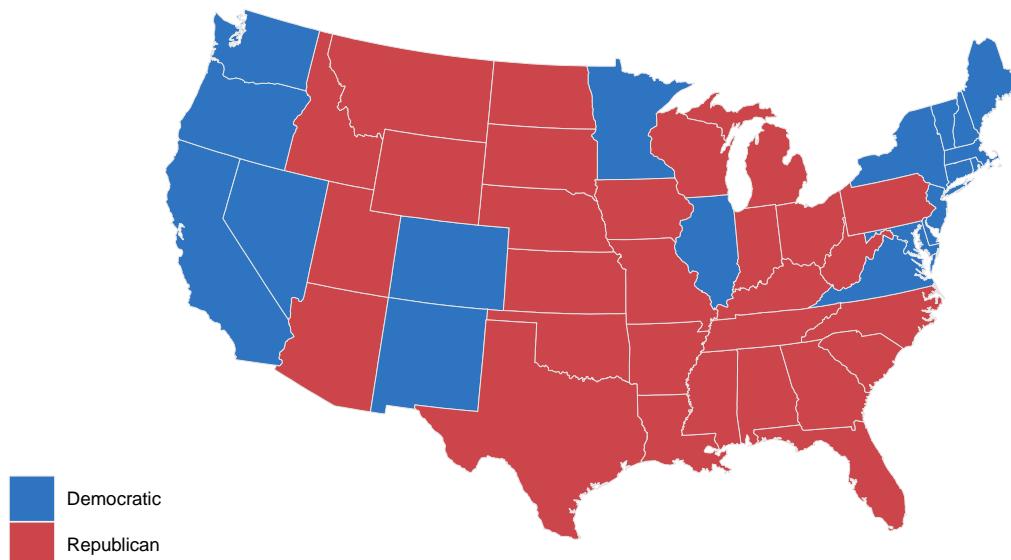
p2 <- p1 + scale_fill_manual(values = party_colors) +
  labs(title = "Election Results 2016", fill = NULL)

p2 + theme_map()

```

Loading required package: grid

Election Results 2016



Agora a partir deste podemos colorir utilizando outras variáveis nosso mapa, como a variável contínua da porcentagem de votos em Donald Trump.

Como por padrão a cor de mapeamento é azul, em p2 especificamos a escala manualmente através da função `scale_fill_gradient()`.

```

p0 <- ggplot(data = us_states_elec,
              mapping = aes(x = long,
                            y = lat,
                            group = group,
                            fill = pct_trump))

p1 <- p0 + geom_polygon(color = "gray90", size = 0.1) +

```

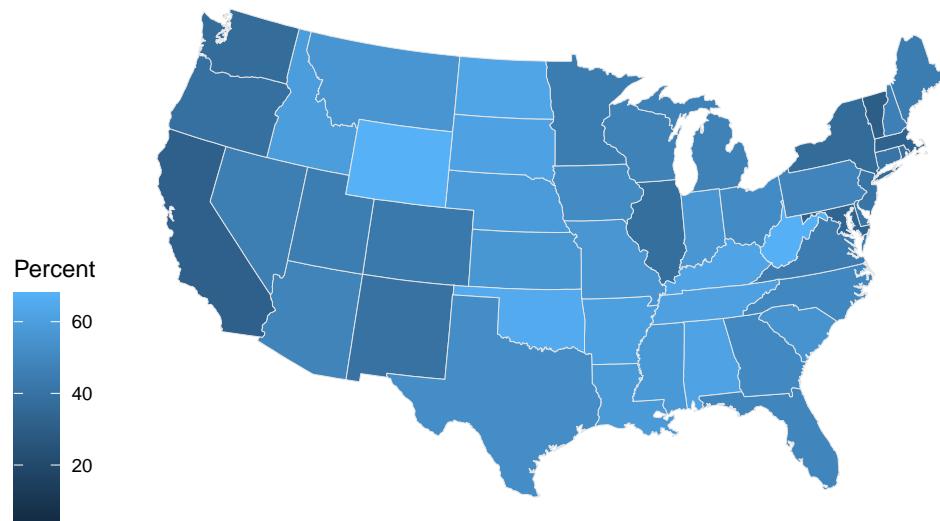
```
coord_map(projection = "albers", lat0 = 39, lat1 = 45)

p1 + labs(title = "Trump vote") +
  theme_map() +
  labs(fill = "Percent")

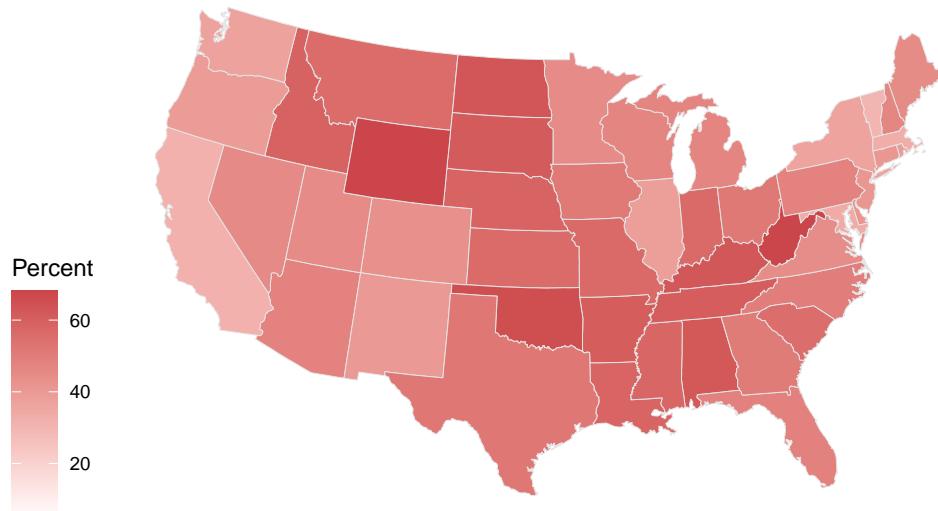
p2 <- p1 + scale_fill_gradient(low = "white",
                                high = "#CB454A") +
  labs(title = "Trump vote")

p2 + theme_map() + labs(fill = "Percent")
```

Trump vote



Trump vote



Para um para com o resultado das eleições optaremos uma função de gradiente que diverge de um ponto central, a `scale_fill_gradient2()`. Ela por padrão nos dá um gradiente azul-vermelho que passa pelo branco como vemos em p1, podemos redefinir que passe pelo roxo como vemos em p2:

```
p0 <- ggplot(data = us_states_elec,
               mapping = aes(x = long,
                             y = lat,
                             group = group,
                             fill = d_points))

p1 <- p0 + geom_polygon(color = "gray90", size = 0.1) +
  coord_map(projection = "albers", lat0 = 39, lat1 = 45)

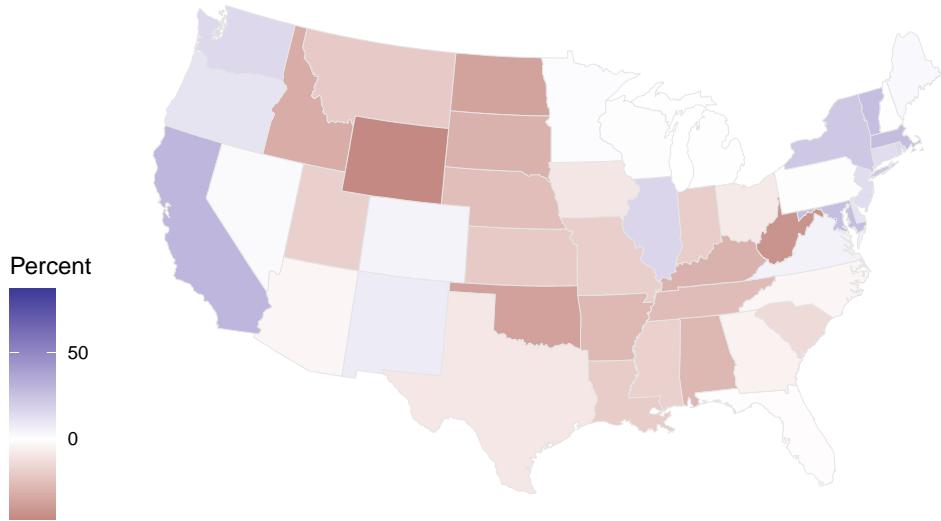
p2 <- p1 + scale_fill_gradient2() +
  labs(title = "Winning margins")

p2 + theme_map() + labs(fill = "Percent")

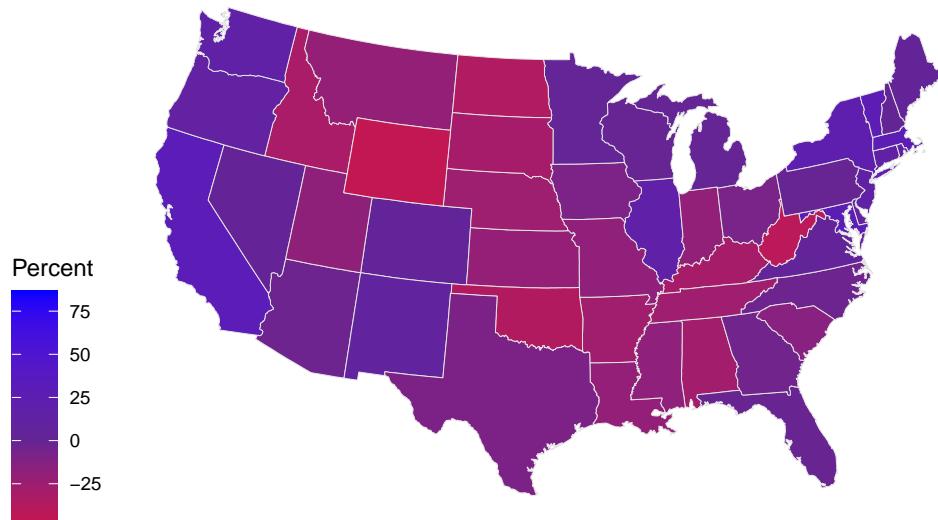
p3 <- p1 + scale_fill_gradient2(low = "red",
                                 mid = scales::muted("purple"),
                                 high = "blue",
                                 breaks = c(-25, 0, 25, 50, 75)) +
  labs(title = "Winning margins")
```

```
p3 + theme_map() + labs(fill = "Percent")
```

Winning margins



Winning margins



Perceba que o gradiente do último mapa apresenta estar muito mais para os tons azuis, isso é devido ao fato de Washington ter disparado os maiores votos a favor dos Democratas do que qualquer outro estado do mapa. Para resolver este problema de escala, vamos omitir essa parte dos dados.

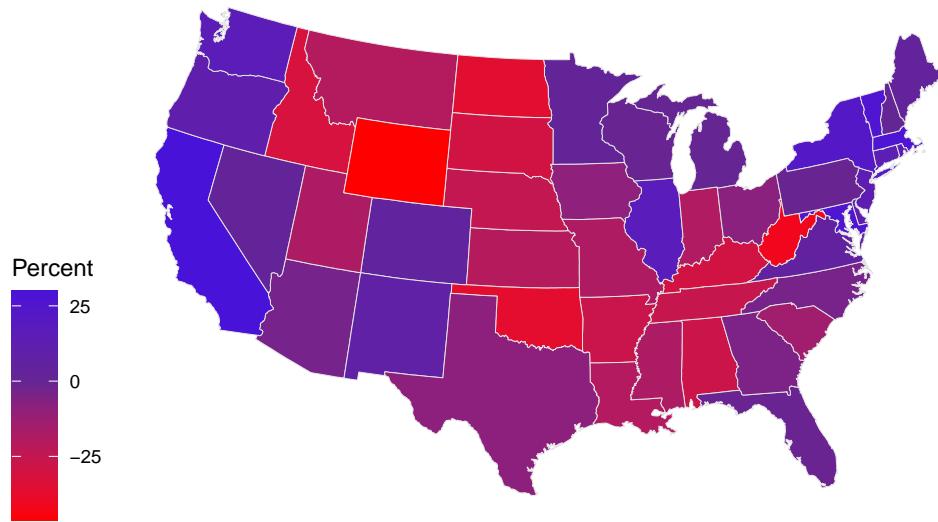
```
p0 <- ggplot(data = subset(us_states_elec,
                           region %in% "district of columbia"),
              aes(x = long,
                  y = lat,
                  group = group,
                  fill = d_points))

p1 <- p0 + geom_polygon(color = "gray90", size = 0.1) +
  coord_map(projection = "albers", lat0 = 39, lat1 = 45)

p2 <- p1 + scale_fill_gradient2(low = "red",
                                 mid = scales::muted("purple"),
                                 high = "blue") +
  labs(title = "Winning margins")

p2 + theme_map() + labs(fill = "Percent")
```

Winning margins



choropleths dos EUA

Vamos usar agora outro dataset um pouco maior que o anterior, que é dividido em pequenas regiões:

```
county_map %>% sample_n(5)
```

```
##           long        lat  order hole piece      group     id
## 1  1518262.41 -613894.173  66725 FALSE      1 0500000US21127.1 21127
## 2   57084.37 -1975621.388 165849 FALSE      1 0500000US48505.1 48505
## 3   434231.58 -608272.749 103661 FALSE      1 0500000US29165.1 29165
## 4   765802.47  241634.471  87077 FALSE      1 0500000US26053.1 26053
## 5 -1190732.67  -6928.735  49097 FALSE      1 0500000US16037.1 16037
```

E usaremos este dataset que possui informações das pequenas regiões:

```
county_data %>%
  select(id, name, state, pop_dens, pct_black) %>%
  sample_n(5)
```

```
##       id      name state  pop_dens  pct_black
## 1 0500000US21127.1  West Virginia 21127  10.000000
```

```

## 1 08079 Mineral County      CO [     0,    10) [ 0.0, 2.0)
## 2 28013 Calhoun County     MS [    10,    50) [25.0,50.0)
## 3 18009 Blackford County   IN [    50,   100) [ 0.0, 2.0)
## 4 37195 Wilson County      NC [   100,   500) [25.0,50.0)
## 5 36021 Columbia County    NY [    50,   100) [ 5.0,10.0)

```

Vamos unir os dois datasets utilizando a coluna de FIPS id.

```
county_full <- left_join(county_map, county_data, by = "id")
```

Com os dados reunidos, podemos plotar o mapa da densidade populacional por milha quadrada.

```

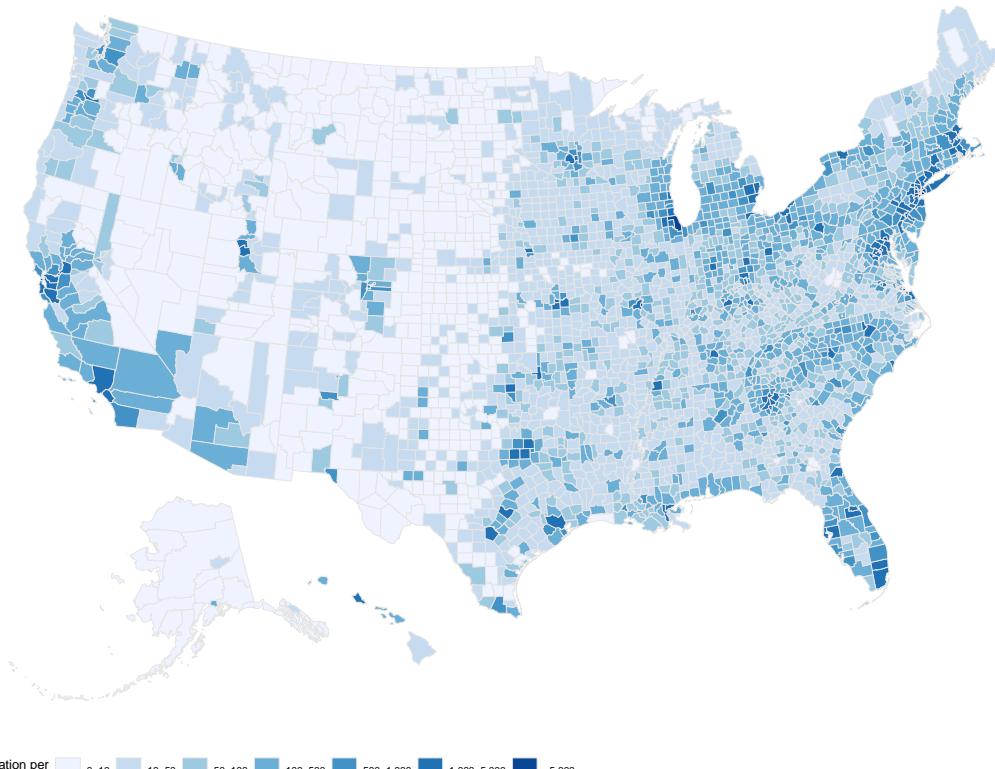
p <- ggplot(data = county_full,
             mapping = aes(x = long, y = lat,
                           fill = pop_dens,
                           group = group))

p1 <- p + geom_polygon(color = "gray90", size = 0.05) + coord_equal()

p2 <- p1 + scale_fill_brewer(palette="Blues",
                             labels = c("0-10", "10-50", "50-100", "100-500",
                                       "500-1,000", "1,000-5,000", ">5,000"))

p2 + labs(fill = "Population per\nsquare mile") +
  theme_map() +
  guides(fill = guide_legend(nrow = 1)) +
  theme(legend.position = "bottom")

```



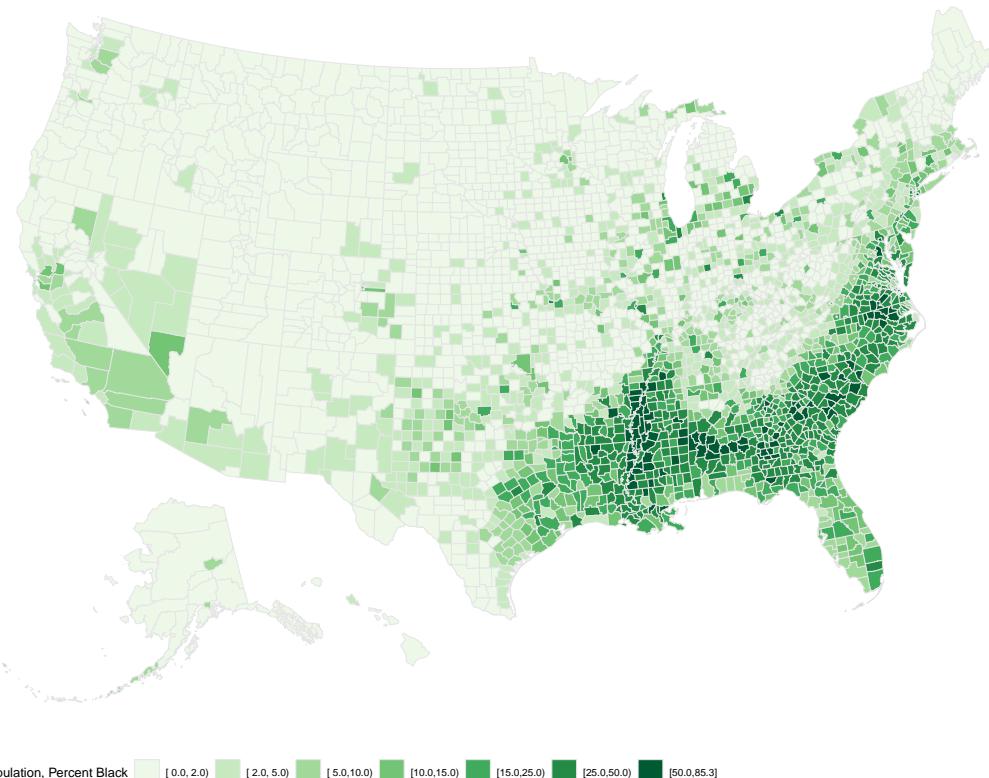
Desta vez vamos alterar a paleta do mapa para uma verde.

```
p <- ggplot(data = county_full,
             mapping = aes(x = long,
                           y = lat,
                           fill = pct_black,
                           group = group))

p1 <- p + geom_polygon(color = "gray90", size = 0.05) + coord_equal()

p2 <- p1 + scale_fill_brewer(palette="Greens")

p2 + labs(fill = "US Population, Percent Black") +
  guides(fill = guide_legend(nrow = 1)) +
  theme_map() + theme(legend.position = "bottom")
```



Podemos definir novas paletas do pacote Color Brewer, observe que utilizamos a função rev() para inverter o vetor de cores.

```
orange_pal <- RColorBrewer::brewer.pal(n = 6, name = "Oranges")
orange_pal
```

```
## [1] "#FEEEDDE" "#FDD0A2" "#FDAE6B" "#FD8D3C" "#E6550D" "#A63603"
```

```
orange_rev <- rev(orange_pal)
orange_rev
```

```
## [1] "#A63603" "#E6550D" "#FD8D3C" "#FDAE6B" "#FDD0A2" "#FEEDDE"
```

Agora aplicamos no mapa de densidade dos suicídios relacionados a armas a paleta customizada e a paleta invertida.

```
gun_p <- ggplot(data = county_full,
                  mapping = aes(x = long,
                                 y = lat,
                                 fill = su_gun6,
                                 group = group))

gun_p1 <- gun_p + geom_polygon(color = "gray90", size = 0.05) + coord_equal()

gun_p2 <- gun_p1 + scale_fill_manual(values = orange_rev)

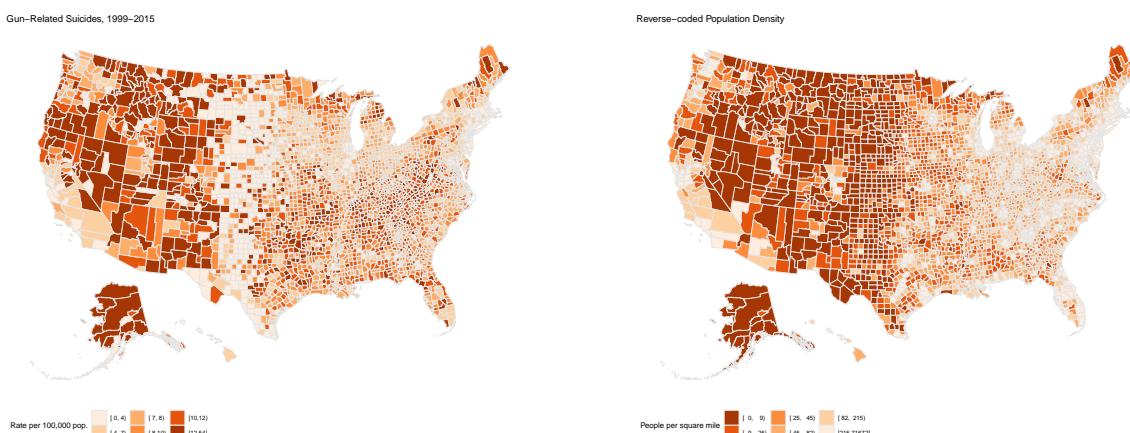
gun_p2 + labs(title = "Gun-Related Suicides, 1999–2015",
              fill = "Rate per 100,000 pop.") +
  theme_map() +
  theme(legend.position = "bottom")

pop_p <- ggplot(data = county_full,
                  mapping = aes(x = long,
                                 y = lat,
                                 fill = pop_dens6,
                                 group = group))

pop_p1 <- pop_p + geom_polygon(color = "gray90", size = 0.05) + coord_equal()

pop_p2 <- pop_p1 + scale_fill_manual(values = orange_rev)

pop_p2 + labs(title = "Reverse-coded Population Density",
              fill = "People per square mile") +
  theme_map() +
  theme(legend.position = "bottom")
```

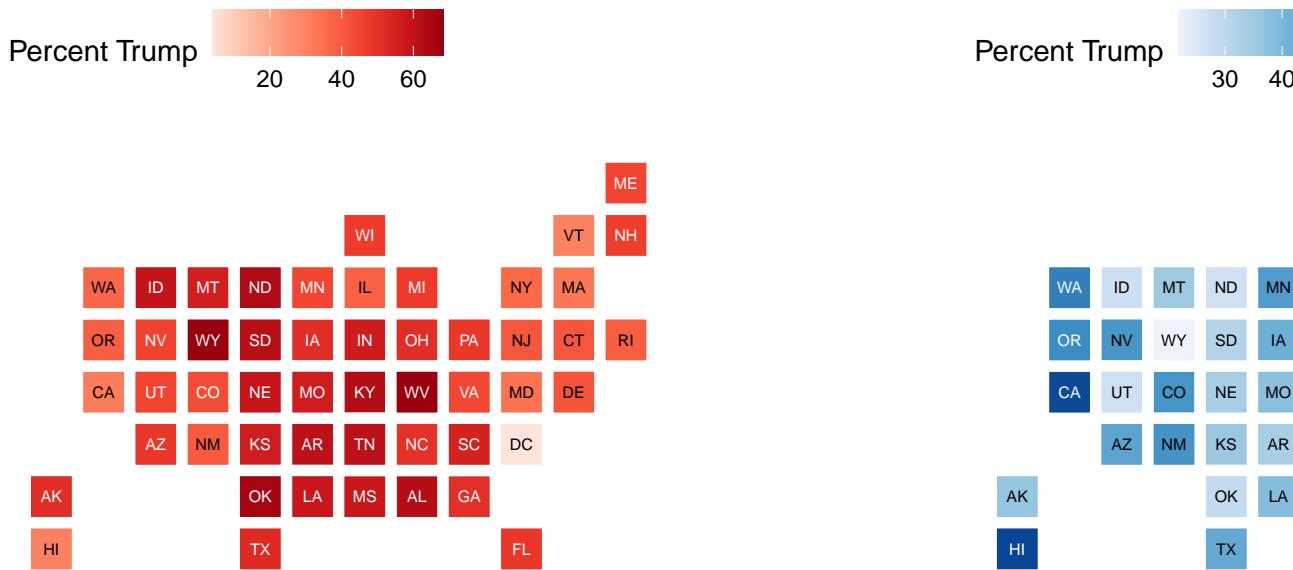


Statebins

Como uma alternativa, podemos considerar o pacote statebins:

```
statebins(election,
          state_col = "state",
          value_col = "pct_trump",
          name = "Percent Trump",
          palette = "Reds",
          direction = 1,
          font_size = 2) +
  theme_statebins(legend_position = "top")

statebins(state_data = subset(election, st %in% "DC"),
          state_col = "state",
          value_col = "pct_clinton",
          name = "Percent Trump",
          palette = "Blues",
          direction = 1,
          font_size = 2) +
  theme_statebins(legend_position = "top")
```



Múltiplos mapas

Múltiplos mapas é uma alternativa para representar dados geográficos ao longo do tempo. Utilizaremos o dataset opiates que foi dados sobre morte por drogas entre 1999 e 2014.

```
opiates
```

```
## # A tibble: 800 x 11
##   year state    fips deaths population crude adjusted adjusted_se region abbr
##   <int> <chr>   <int>  <int>     <dbl>    <dbl>      <dbl> <ord>  <chr>
## 1 1999 Alabama     1     37  4430141  0.8     0.8       0.1 South  AL
## 2 1999 Alaska      2     27  624779   4.3     4        0.8 West   AK
## 3 1999 Arizona     4    229  5023823  4.6     4.7       0.3 West  AZ
## 4 1999 Arkans~     5     28  2651860  1.1     1.1       0.2 South AR
## 5 1999 Califo~     6    1474 33499204  4.4     4.5       0.1 West  CA
## 6 1999 Colora~     8    164  4226018   3.9     3.7       0.3 West CO
## 7 1999 Connec~     9    151  3386401   4.5     4.4       0.4 North~ CT
## 8 1999 Delawa~    10     32  774990   4.1     4.1       0.7 South DE
## 9 1999 Distri~    11     28  570213   4.9     4.9       0.9 South DC
## 10 1999 Florida    12    402  15759421  2.6     2.6       0.1 South FL
## # ... with 790 more rows, and 1 more variable: division_name <chr>
```

```
opiates$region <- tolower(opiates$state)
opiates_map <- left_join(us_states, opiates)
```

```
## Joining, by = "region"
```

Utilizaremos a paleta de cores do pacote viridis, que possui opções de cores balanceadas:

```
library(viridis)

p0 <- ggplot(data = subset(opiates_map,
                           year > 1999),
              mapping = aes(x = long,
                            y = lat,
                            group = group,
                            fill = adjusted))

p1 <- p0 +
  geom_polygon(color = "gray90",
               size = 0.05) +
  coord_map(projection = "albers",
            lat0 = 39,
            lat1 = 45)

p2 <- p1 + scale_fill_viridis_c(option = "plasma")

p2 + theme_map() +
  facet_wrap(~ year, ncol = 3) +
  theme(legend.position = "bottom",
        strip.background = element_blank()) +
  labs(fill = "Death rate per 100,000 population ",
       title = "Opiate Related Deaths by State, 2000-2014")
```

Opiate Related Deaths by State, 2000–2014

