

Ggplot2 tutorial - Command lines Part 4

Ariane Ducellier

10/19/2023

Load R packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.3      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.3      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.0
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(Ecdat)
```

```
## Loading required package: Ecfun
##
## Attaching package: 'Ecfun'
##
## The following object is masked from 'package:base':
##
##     sign
##
## Attaching package: 'Ecdat'
##
## The following object is masked from 'package:datasets':
##
##     Orange
```

Part 4 - Miscellaneous functions

Setting the figure size in RMarkdown

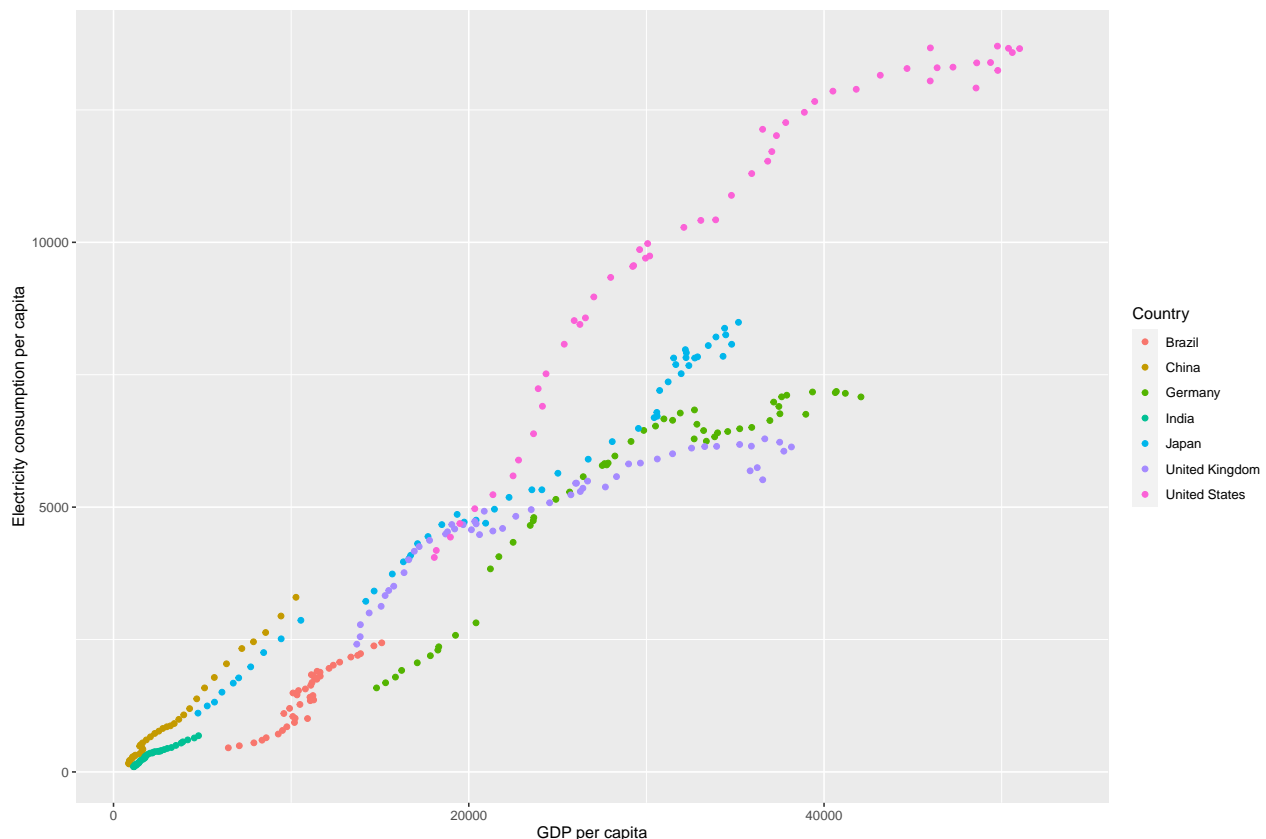
Define with knitr options at the beginning.

```
df <- read_csv("../data/gapminder-data.csv")

## New names:
## Rows: 1512 Columns: 10
## -- Column specification
## ----- Delimiter: "," chr
## (1): Country dbl (9): ...1, Year, gdp_per_capita,
## Electricity_consumption_per_capita, und...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

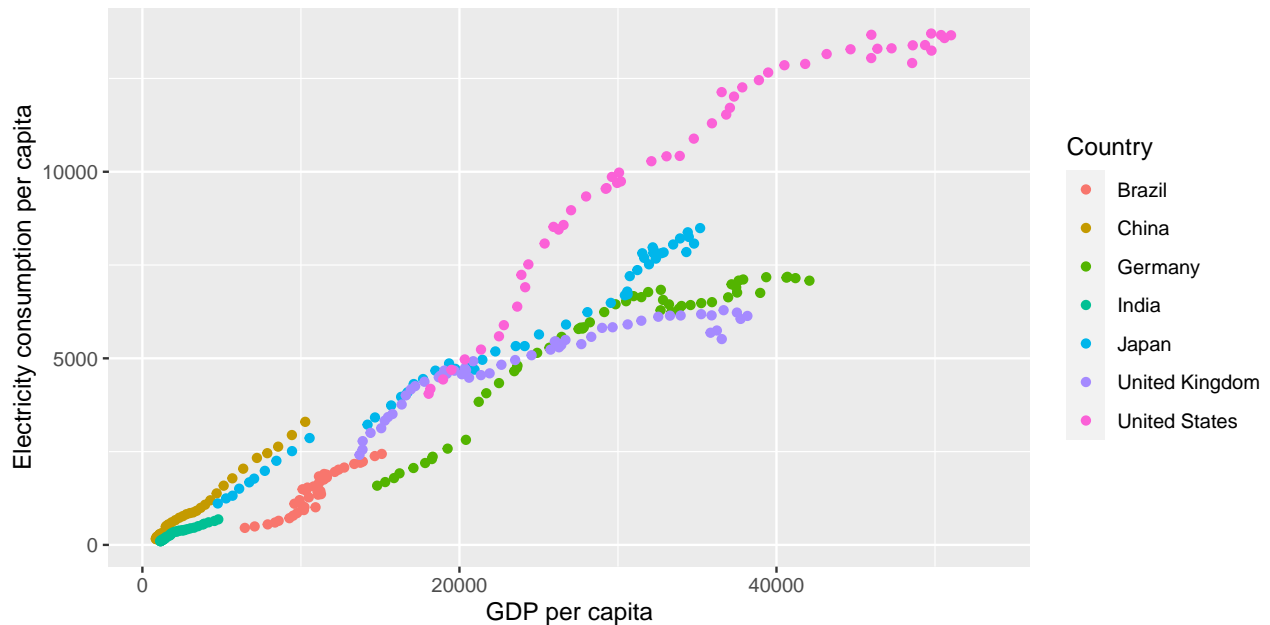
```
ggplot(df, aes(gdp_per_capita, Electricity_consumption_per_capita)) +
  geom_point(aes(color=Country)) +
  xlab("GDP per capita") +
  ylab("Electricity consumption per capita")
```

Warning: Removed 1181 rows containing missing values (`geom_point()`).



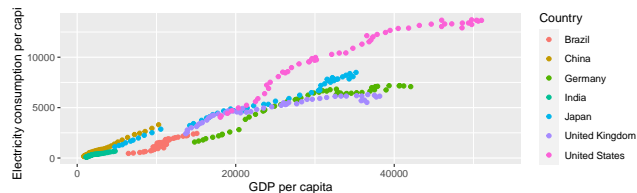
```
ggplot(df, aes(gdp_per_capita, Electricity_consumption_per_capita)) +
  geom_point(aes(color=Country)) +
  xlab("GDP per capita") +
  ylab("Electricity consumption per capita")
```

```
## Warning: Removed 1181 rows containing missing values (`geom_point()`).
```



```
ggplot(df, aes(gdp_per_capita, Electricity_consumption_per_capita)) +  
  geom_point(aes(color=Country)) +  
  xlab("GDP per capita") +  
  ylab("Electricity consumption per capita")
```

```
## Warning: Removed 1181 rows containing missing values (`geom_point()`).
```



Setting the figure size when saving

```
ggsave(filename = "figures/small_memory_size.png", ggplot(mtcars, aes(x=wt, y=mpg)) +  
  geom_point(size=2, shape=23) + theme_bw(base_size = 10),  
  width = 5, height = 4, dpi = 150, units = "in", device='png')  
ggsave(filename = "figures/big_memory_size.png", ggplot(mtcars, aes(x=wt, y=mpg)) +  
  geom_point(size=2, shape=23) + theme_bw(base_size = 10),  
  width = 5, height = 4, dpi = 300, units = "in", device='png')  
ggsave(filename = "figures/big_figure.png", ggplot(mtcars, aes(x=wt, y=mpg)) +  
  geom_point(size=2, shape=23) + theme_bw(base_size = 10),  
  width = 10, height = 8, dpi = 300, units = "in", device='png')
```

You need to scale the figures to get the same size of dots and fonts.

Highlighting parts of the map

We are interested in the gross state product (GSP) of the three states: New York, New Jersey and Connecticut.

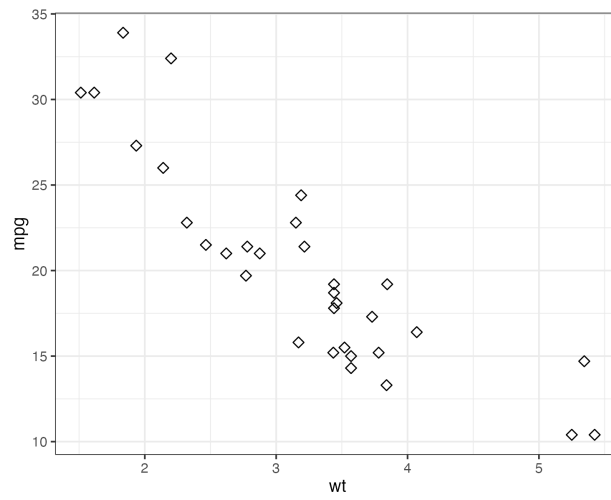


Figure 1: Small width and height

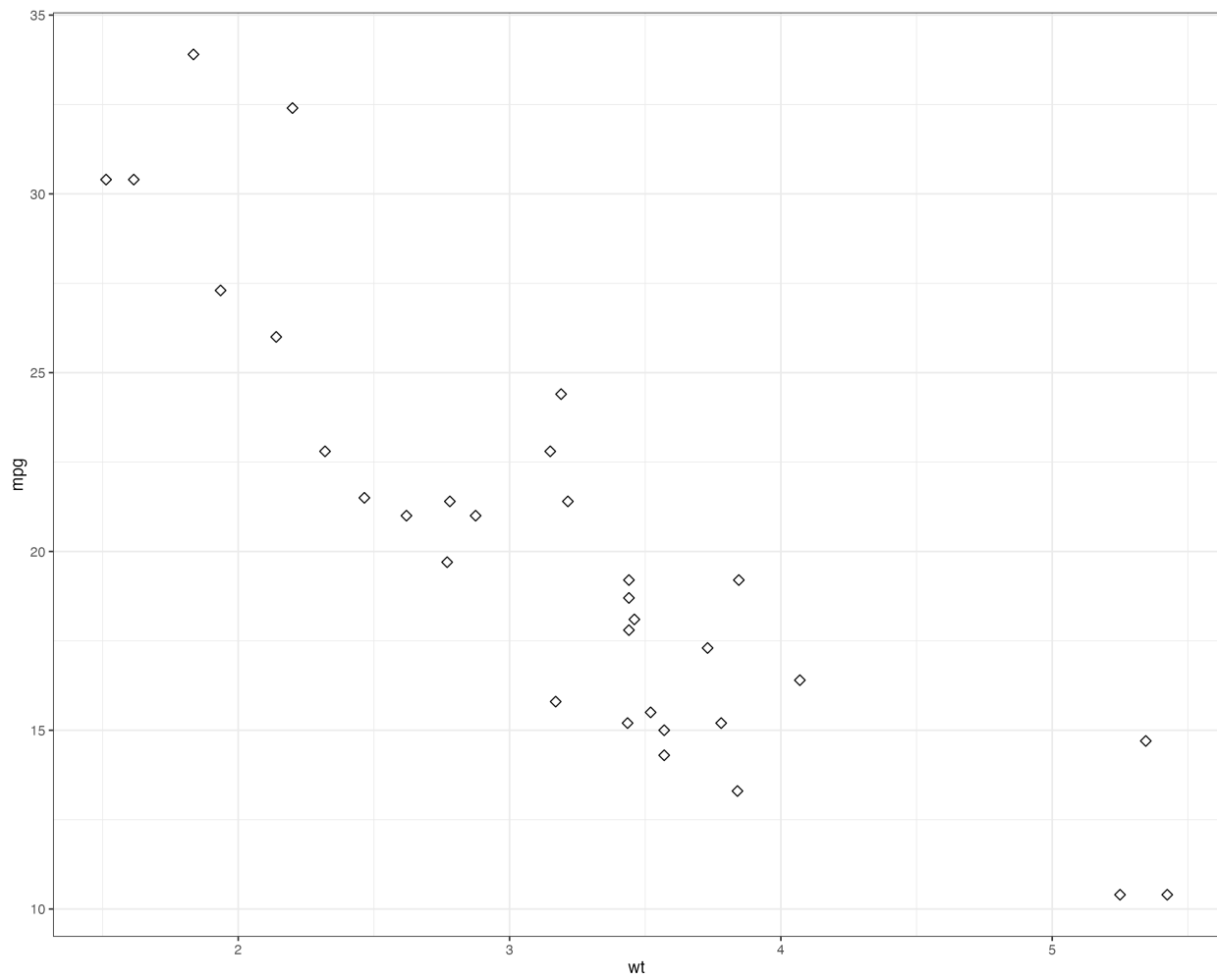
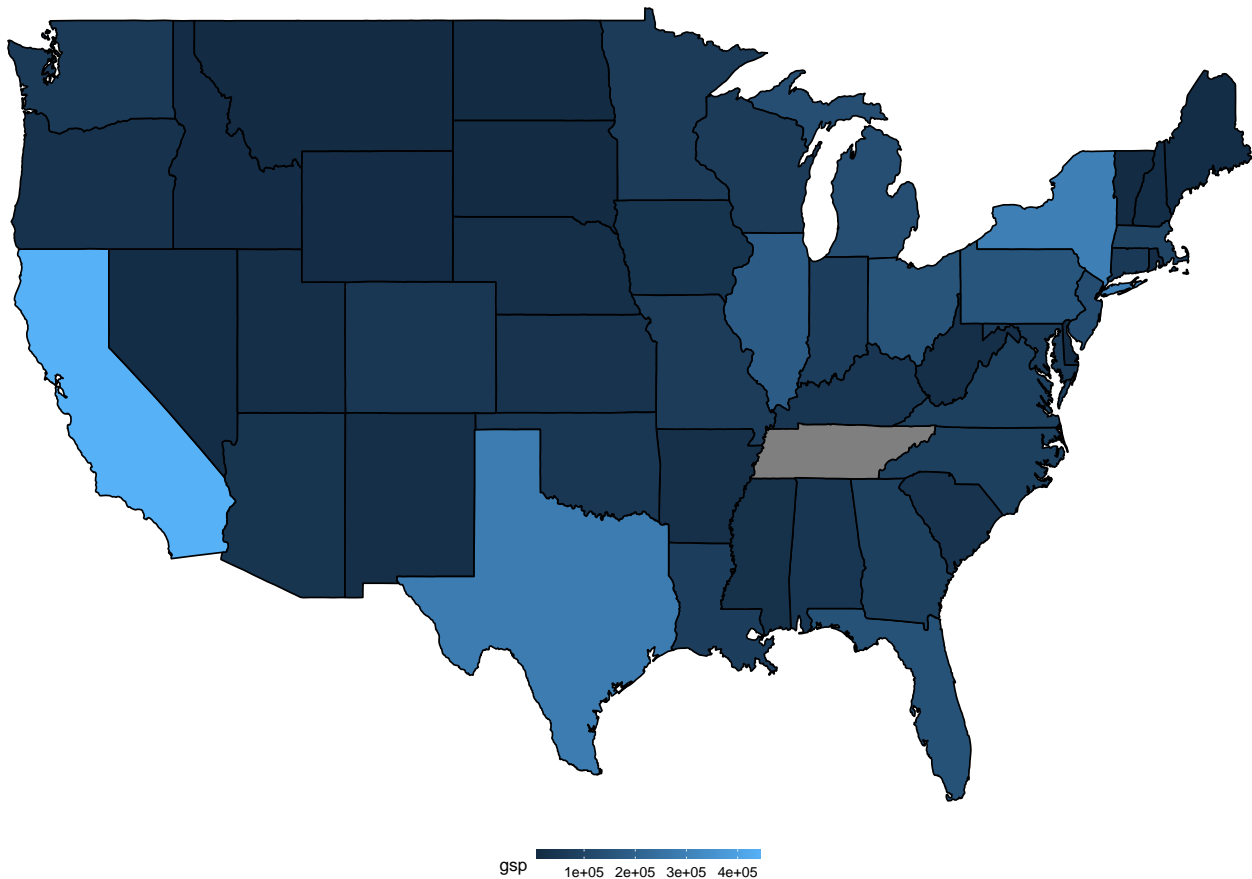


Figure 2: Big width and height

```

us_map <- map_data("state")
us_prod <- Produc$year==1985,]
us_prod$region <- gsub("_", " ", tolower(us_prod$state))
merged_data <- left_join(us_map, us_prod[, c("region", "gsp")], by="region")
choropleth <- ggplot(merged_data) +
  geom_polygon(aes(x=long, y=lat, group=group, fill=gsp), color="black") +
  theme_void() +
  theme(legend.position="bottom") +
  scale_fill_continuous(guide=guide_colorbar(barheight=unit(2, units="mm"),
                                             barwidth=unit(5, units="cm")))
choropleth

```

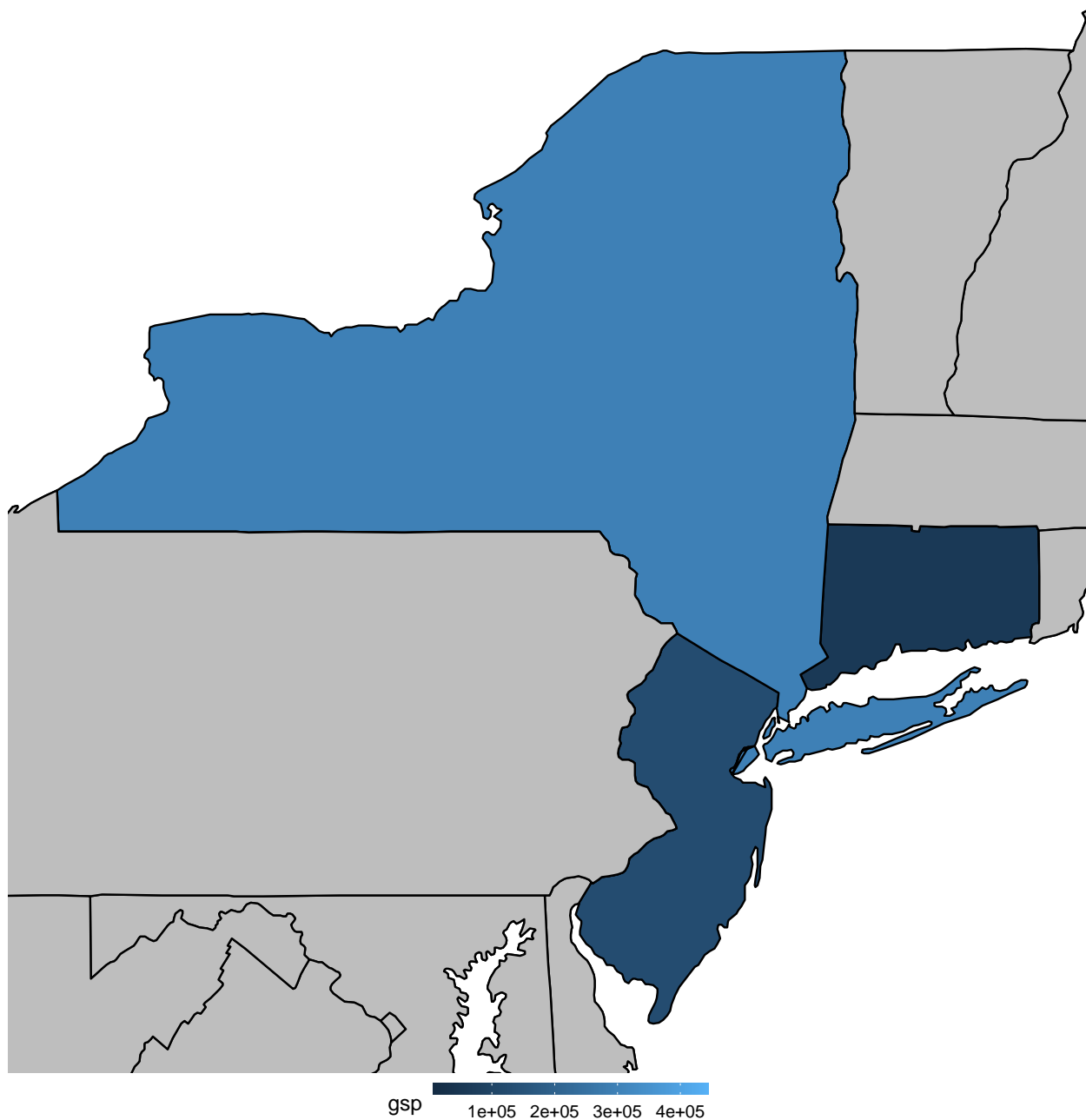


```

tristate <- c("new york", "new jersey", "connecticut")
long_lim <- merged_data$long[merged_data$region %in% tristate]
lat_lim <- merged_data$lat[merged_data$region %in% tristate]
outer <- merged_data[!merged_data$region %in% tristate,]

choropleth +
  geom_polygon(data=outer, aes(x=long, y=lat, group=group),
    fill="gray", color="black") +
  coord_fixed(xlim=c(min(long_lim), max(long_lim)),
    ylim=c(min(lat_lim), max(lat_lim)),
    ratio=1.3)

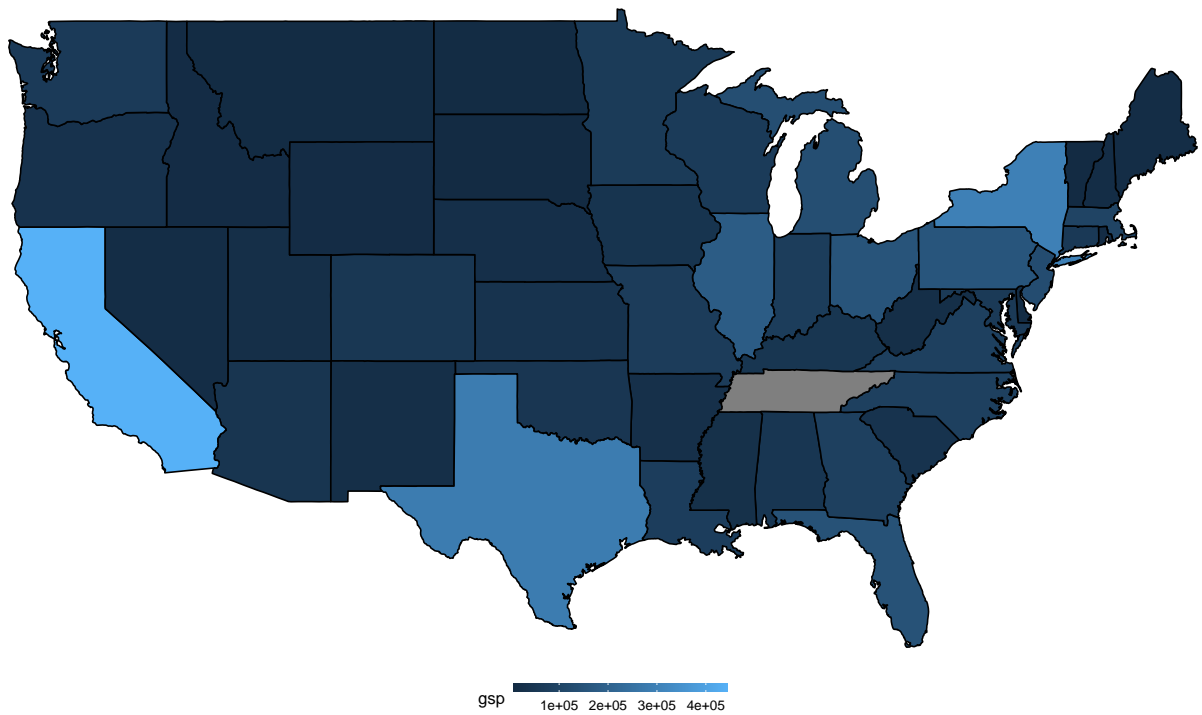
```



Changing the map projection

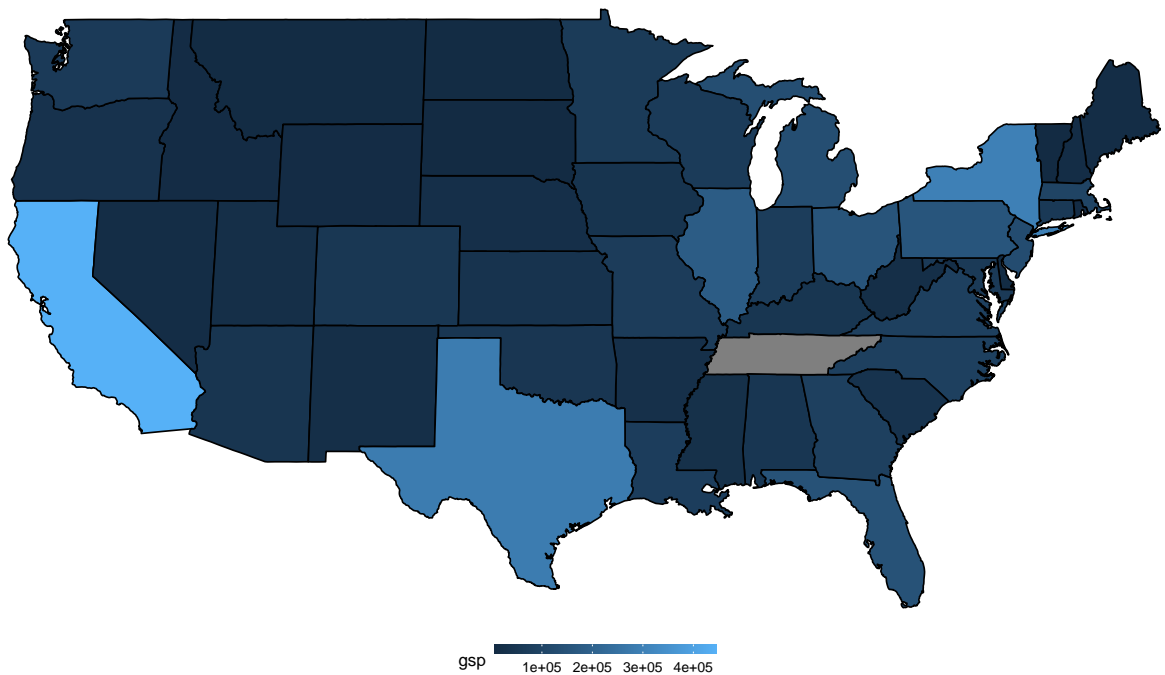
```
choropleth +  
  coord_map() +  
  ggtitle("US Map - mercator projection")
```

US Map – mercator projection



```
choropleth +  
  coord_map(projection = "gilbert") +  
  ggtitle("US Map - gilbert projection")
```

US Map – gilbert projection

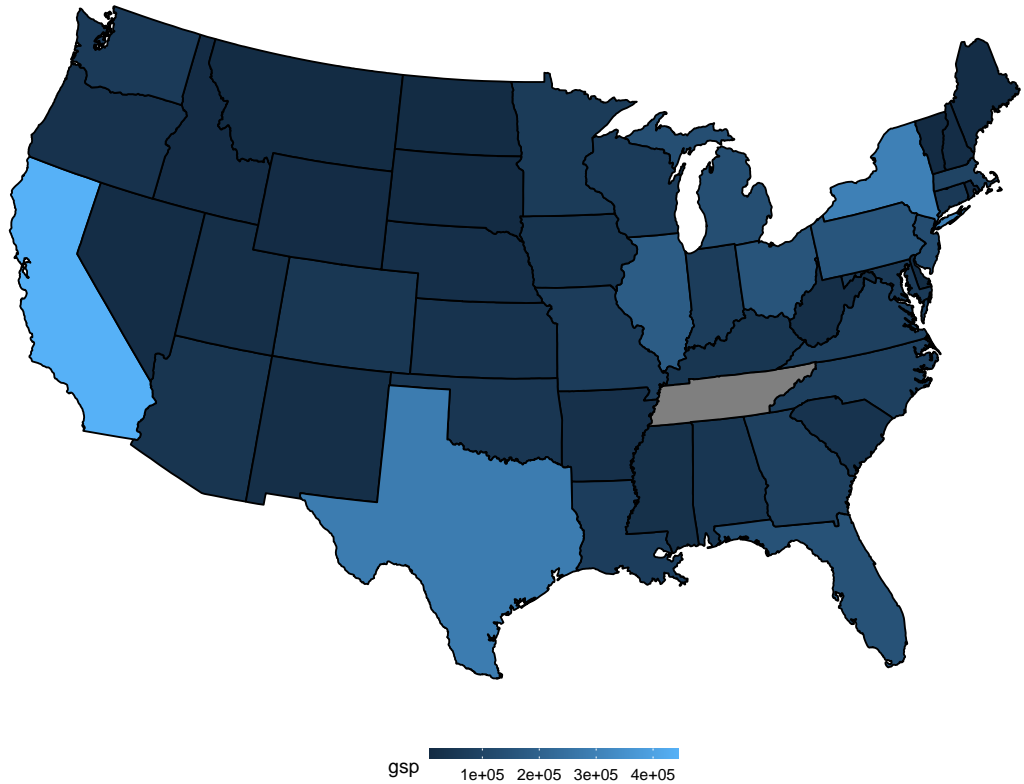


```
choropleth +  
  coord_map(projection = "conic", lat0 = 50) +  
  labs(title = "US Map - conic projection",
```

```
subtitle = "latitude zero = 50")
```

US Map – conic projection

latitude zero = 50



Contour plots

```
p <- ggplot(faithfuld, aes(waiting, eruptions, z = density))
p1 <- p + geom_contour()
p2 <- p + geom_contour_filled()
p3 <- p + geom_raster(aes(fill = density)) +
  geom_contour(color="white")
plot_grid(p1, p2, p3, align = "v", nrow = 3, rel_heights = c(1/3, 1/3, 1/3))
```

Warning: Graphs cannot be vertically aligned unless the axis parameter is set.

Placing graphs unaligned.

