DATA VISUALIZATION WITH GGPLOT2

Choropleths

Chapter Contents

- Maps
  - GIS = Geographic Information System
  - Choropleths
  - Cartographic maps
- Animations

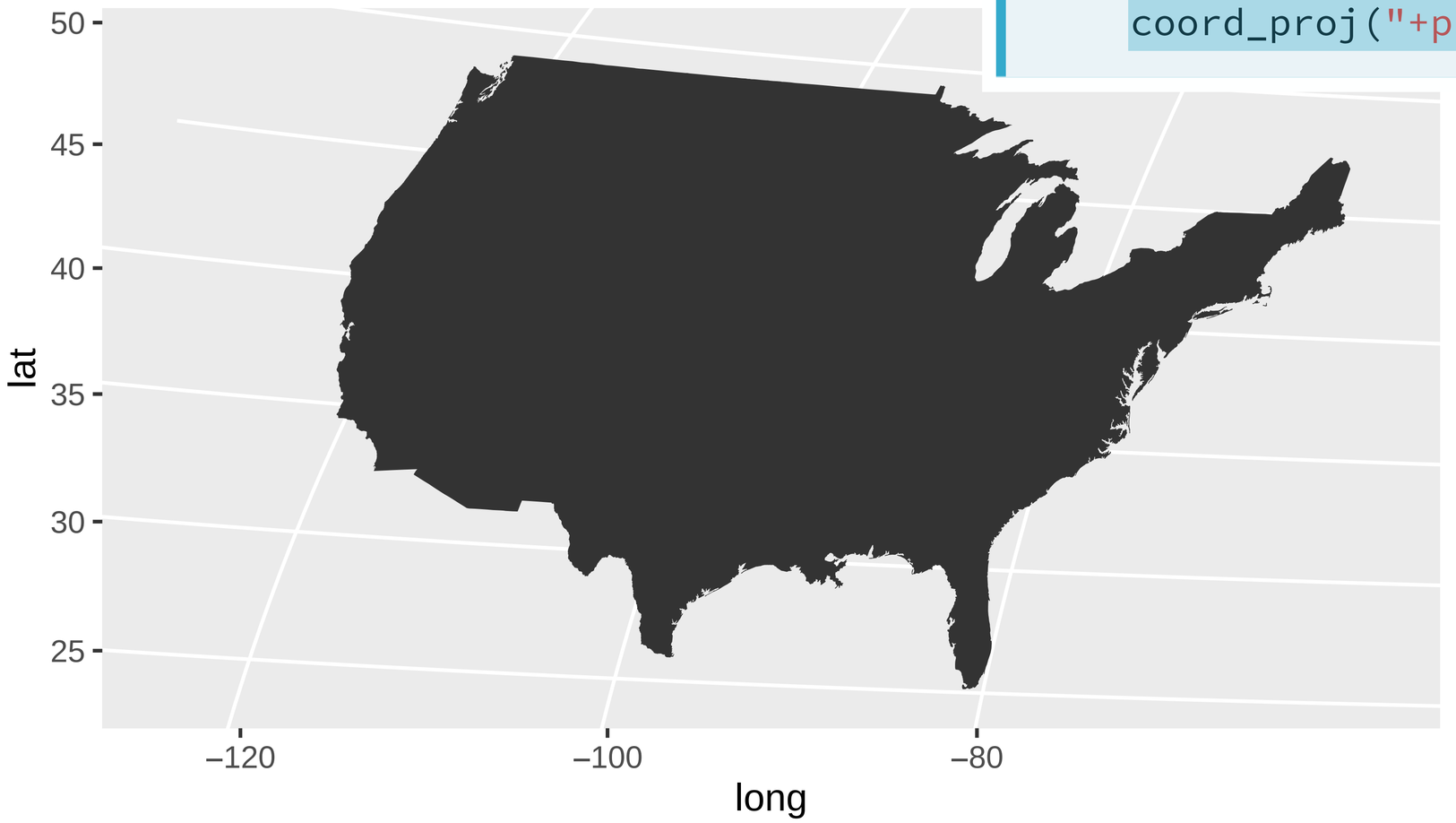
Choropleths

Bunch of polygons

```
> library(ggplot2)
> usa <- map_data("usa")
> ggplot(usa, aes(long, lat, group = group)) +
    geom_polygon() +
    coord_map()
```

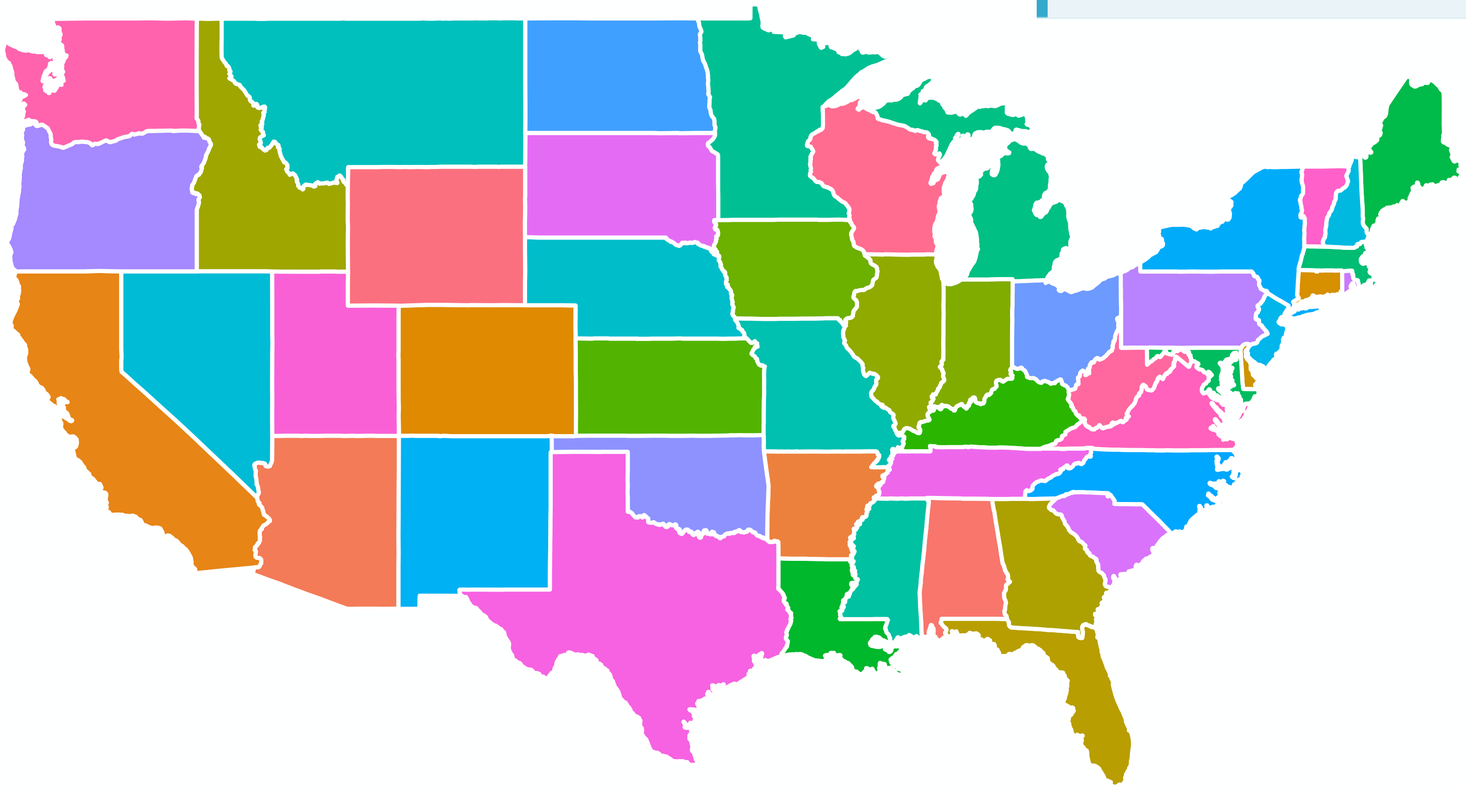
Choropleths

```
> library(ggplot2)
> library(ggalt)
> usa <- map_data("usa")
> ggplot(usa, aes(long, lat, group = group)) +
    geom_polygon() +
    coord_proj("+proj=wintri")
```

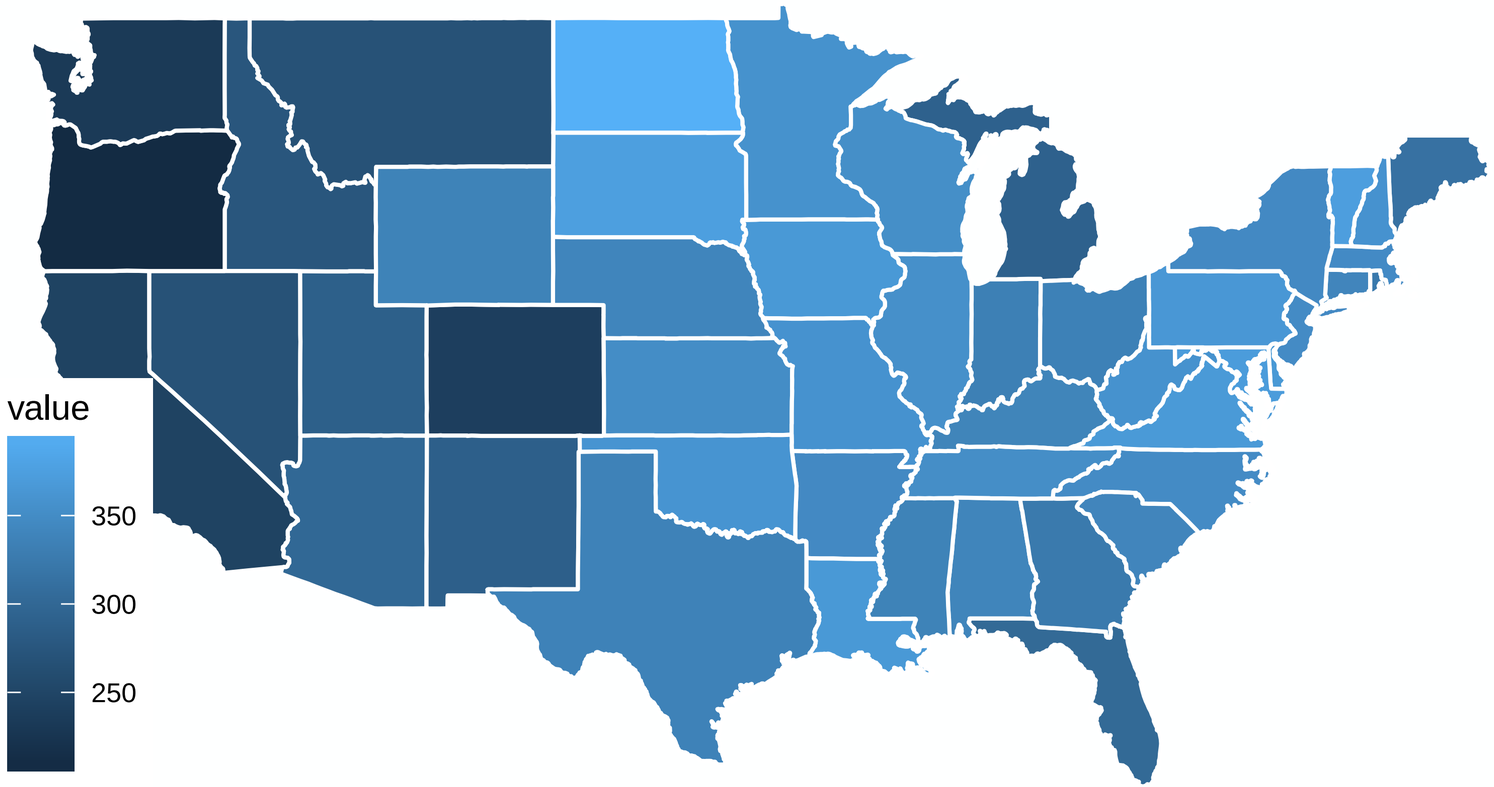


Many polygons

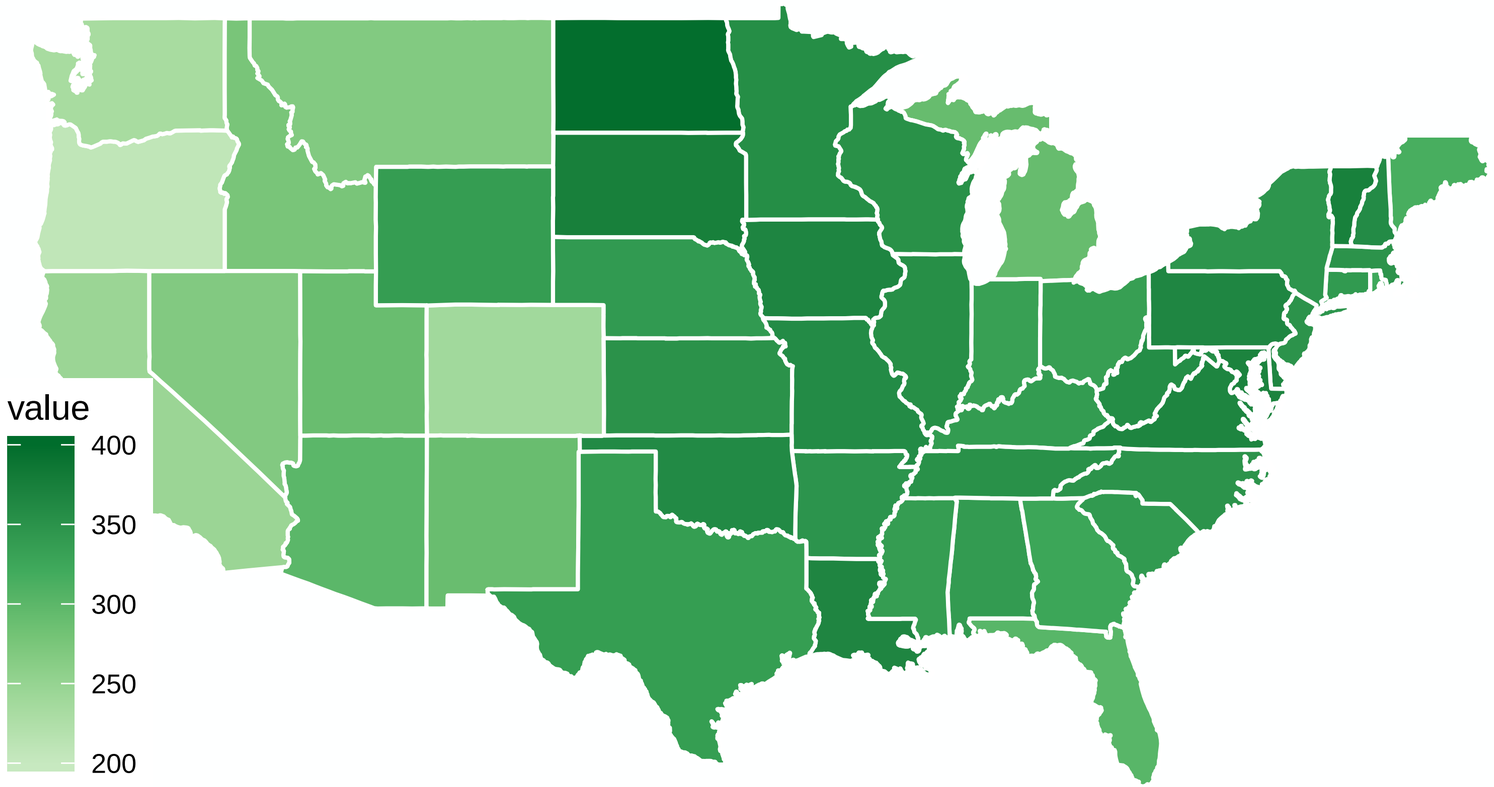
```
> states <- map_data("state")
> ggplot(states,
         aes(long, lat,
             fill = region, group = group)) +
    geom_polygon(color = "white") +
    coord_map()
```



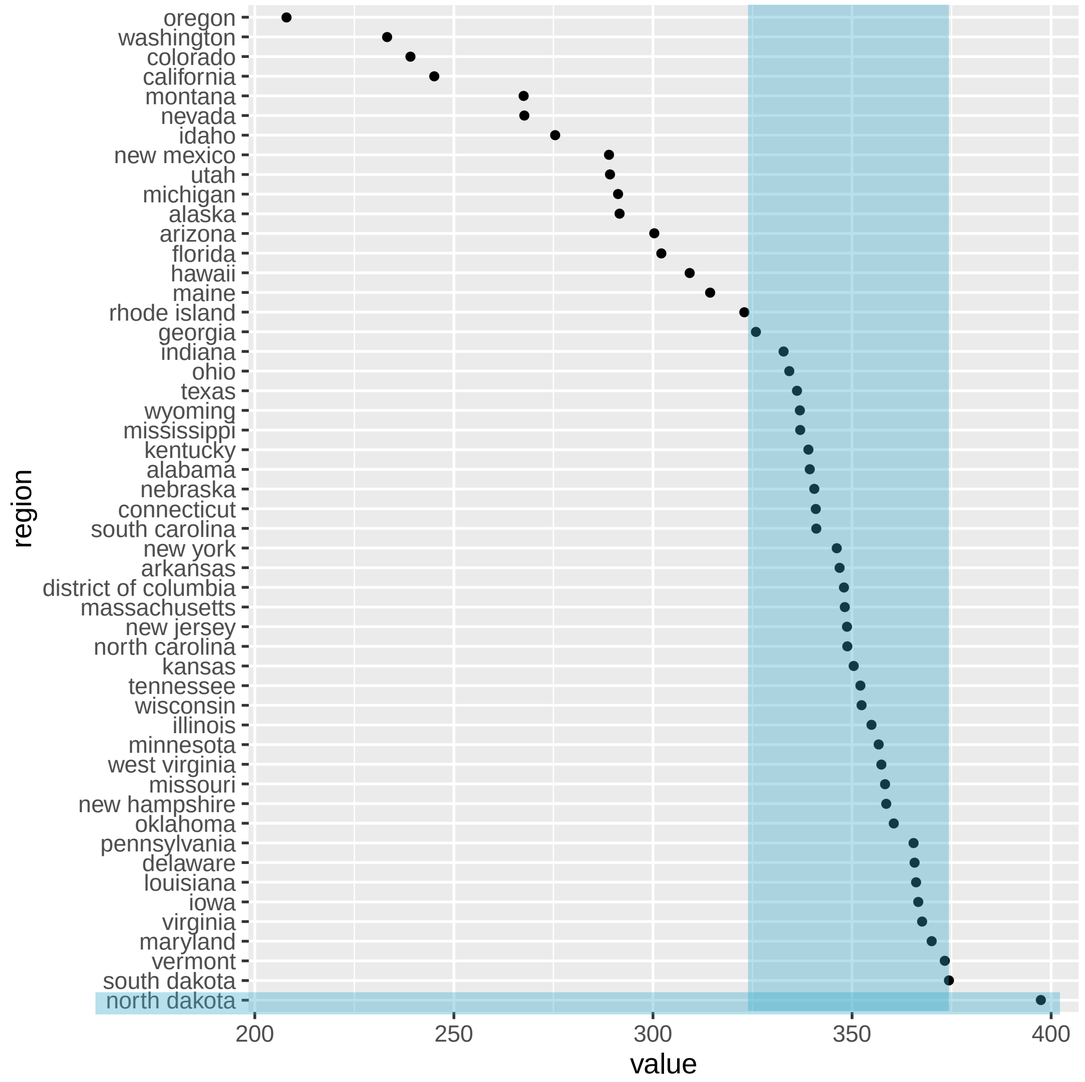
Weed prices



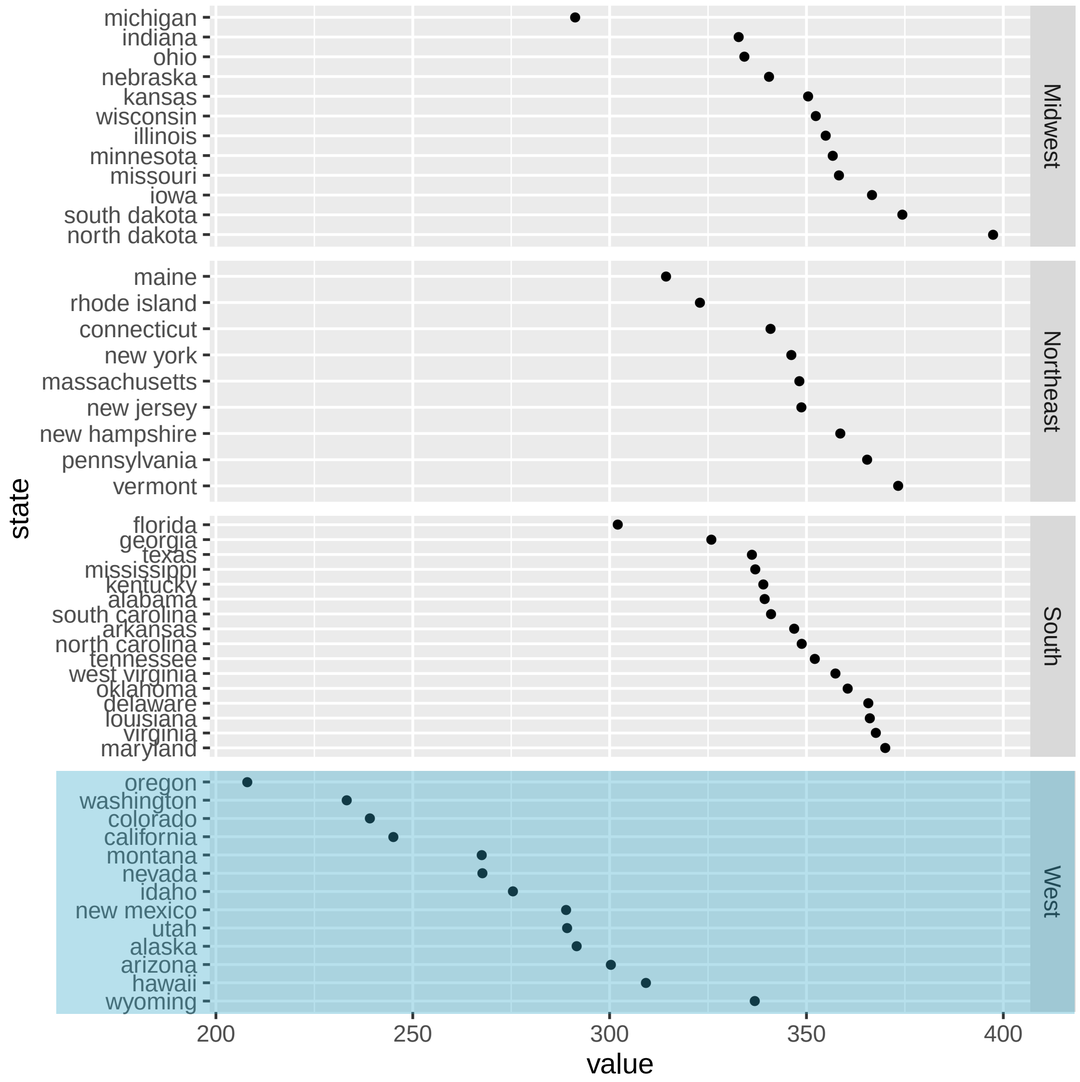
Weed prices



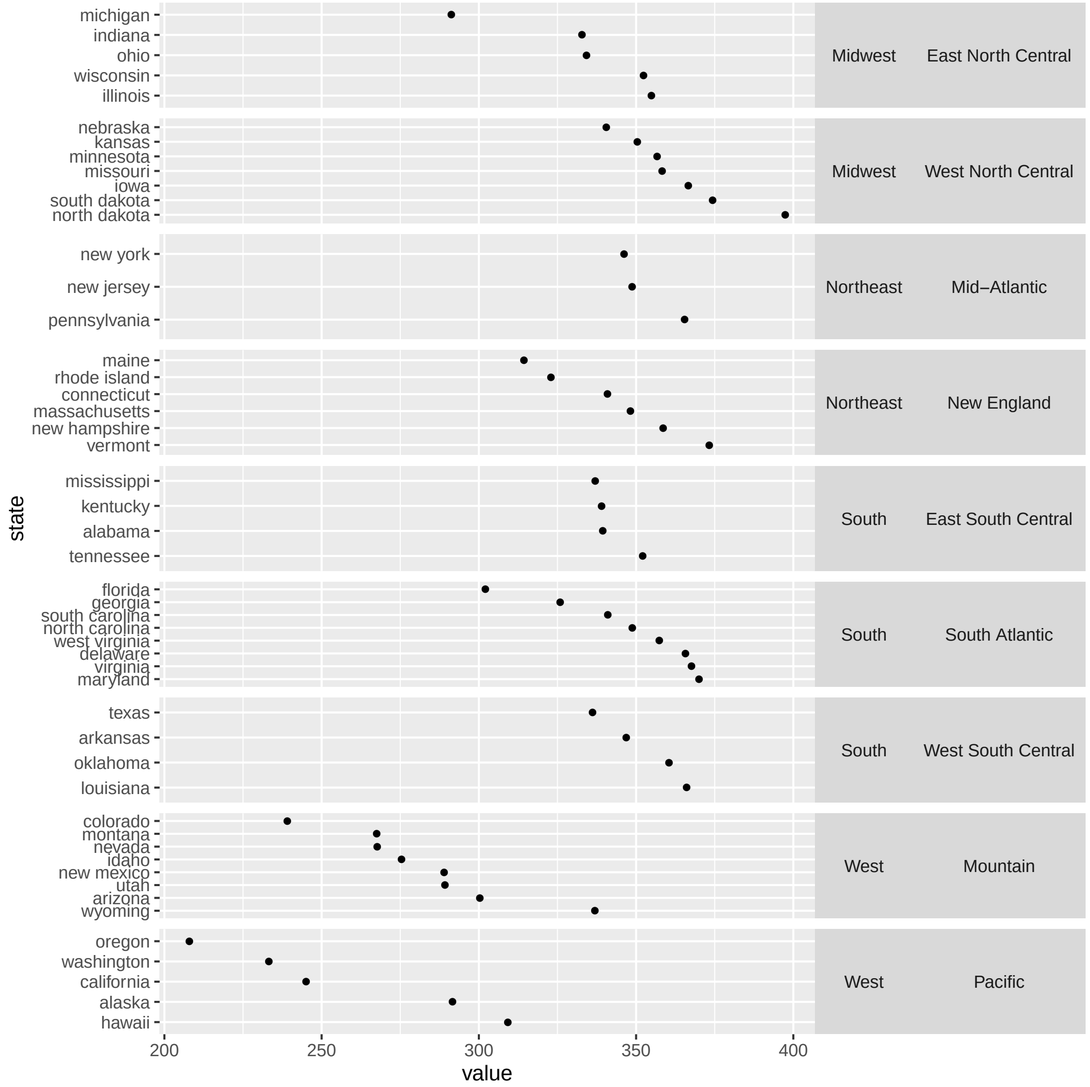
Alternatives



Alternatives



Alternatives



DATA VISUALIZATION WITH GGPLOT2

Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

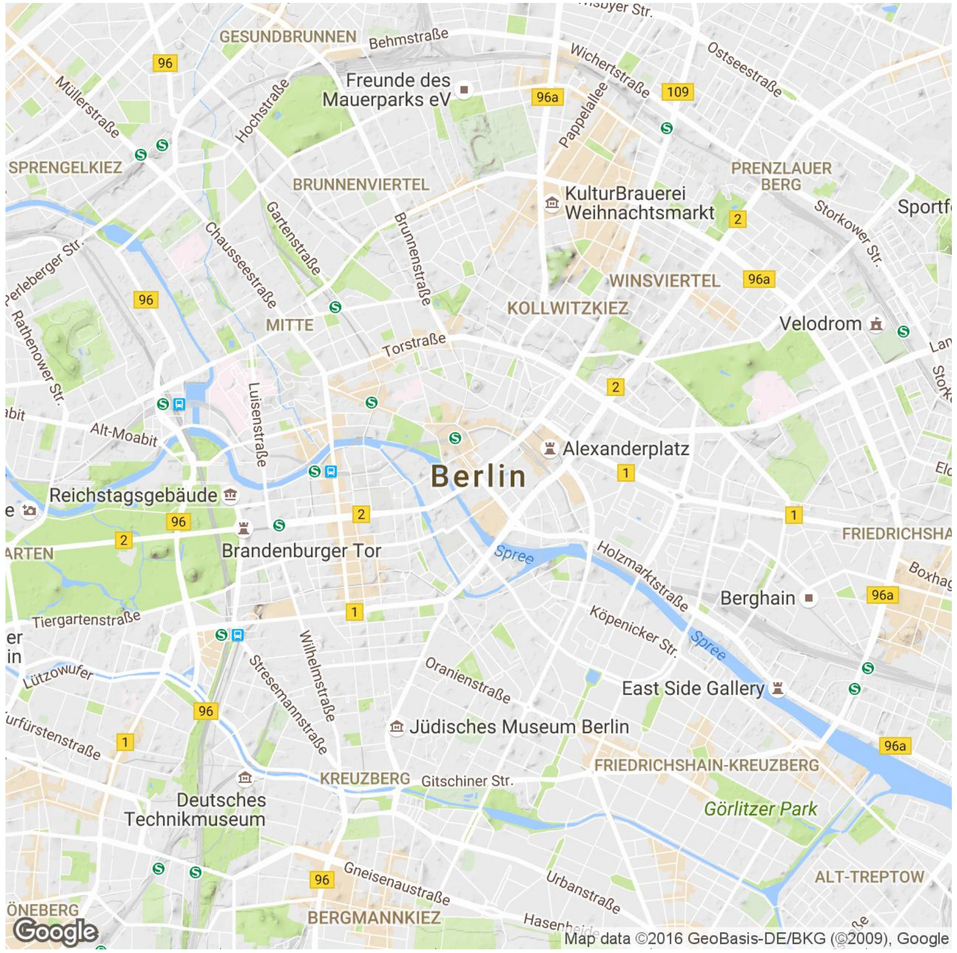
Cartographic Maps

Cartographic map

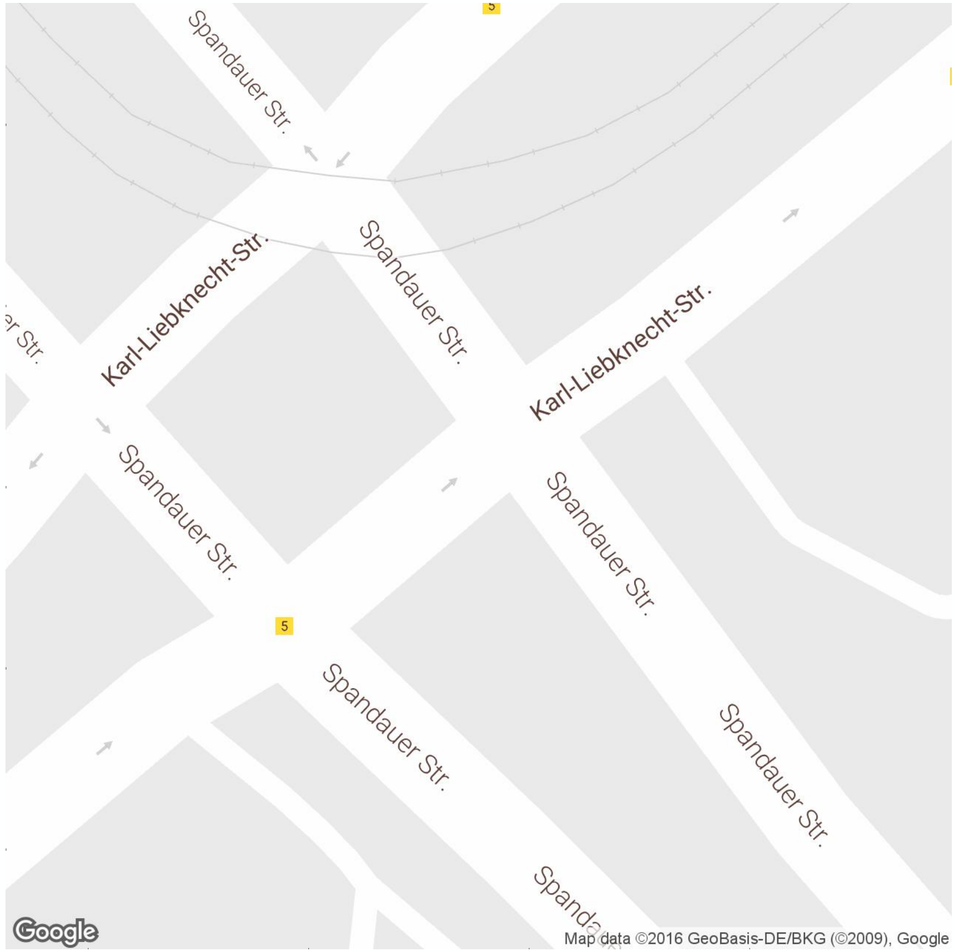
- Drawn
  - Topographical maps
  - Altitude, infrastructure ...
- Photographic
  - Satellite images
- Hybrid
- ggmap

```
> # Default style - zoom = 3
> library(ggmap)
> def_03 <- get_map(location = "Berlin, Germany", zoom = 3)
> ggmap(def_03, extent = "device")
```

```
> # Default style - zoom = 13
> library(ggmap)
> def_13 <- get_map(location = "Berlin, Germany", zoom = 13)
> ggmap(def_13, extent = "device")
```



```
> # Default style - zoom = 20
> library(ggmap)
> def_20 <- get_map(location = "Berlin, Germany", zoom = 20)
> ggmap(def_20, extent = "device")
```

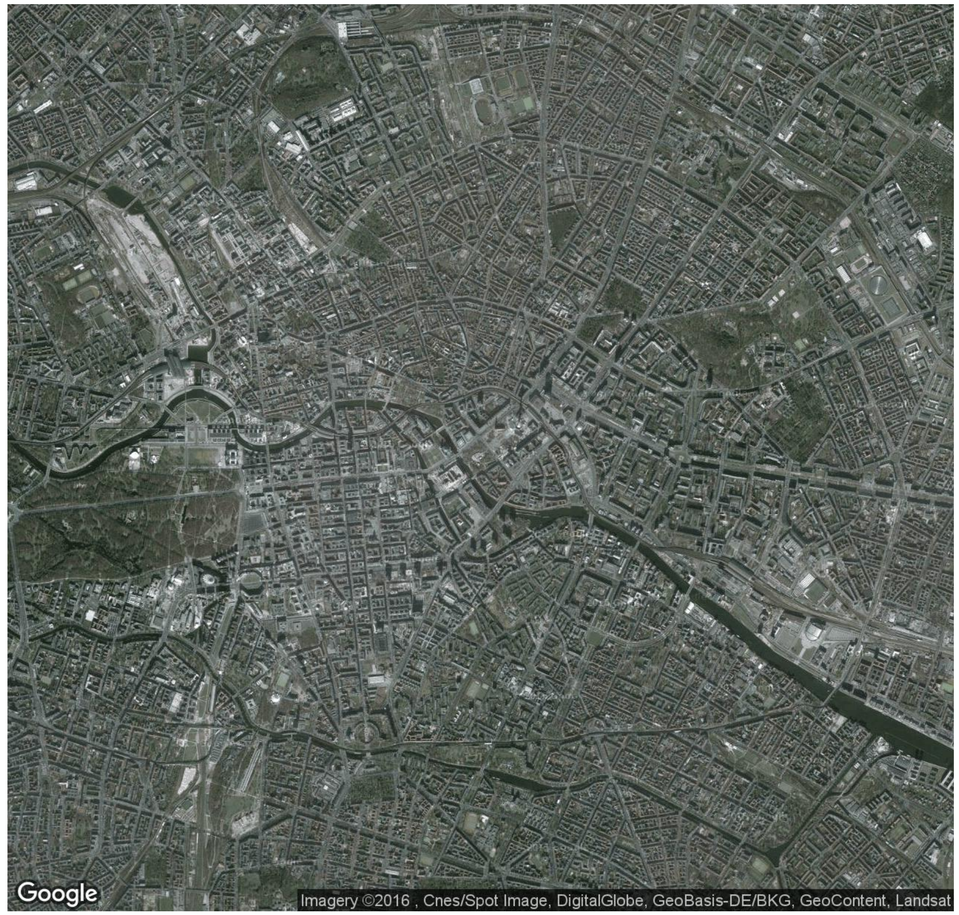


```
> # stamen/watercolor - zoom = 13
> library(ggmap)
> wc_13 <- get_map(location = "Berlin, Germany", zoom = 13,
                   source = "stamen", maptype = "watercolor")
> ggmap(wc_13, extent = "device")
```

```
> # stamen/toner - zoom = 13
> library(ggmap)
> ton_13 <- get_map(location = "Berlin, Germany", zoom = 13,
                    source = "stamen", maptype = "toner")
> ggmap(ton_13, extent = "device")
```

```
> # stamen/hybrid - zoom = 13
> library(ggmap)
> hyb_13 <- get_map(location = "Berlin, Germany", zoom = 13,
                    source = "stamen", maptype = "hybrid")
> ggmap(ton_13, extent = "device")
```

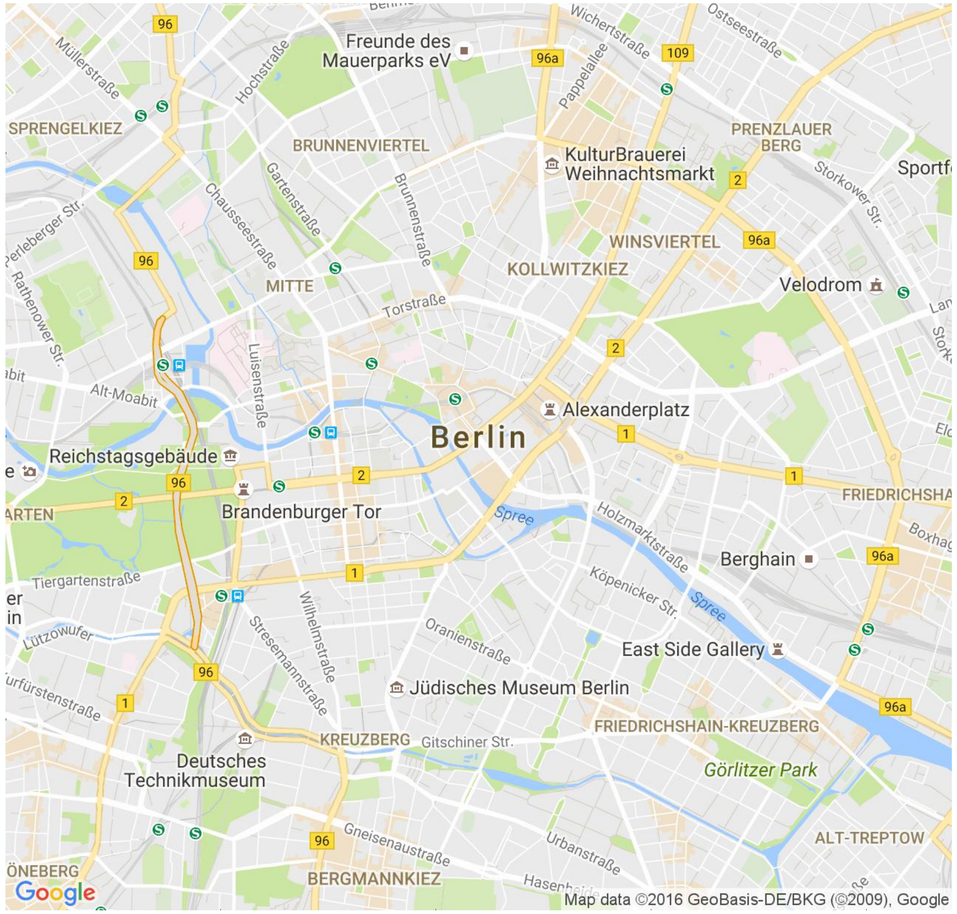
```
> # google/satellite - zoom = 13
> library(ggmap)
> sat_13 <- get_map(location = "Berlin, Germany", zoom = 13,
                    source = "google", maptype = "satellite")
> ggmap(ton_13, extent = "device")
```



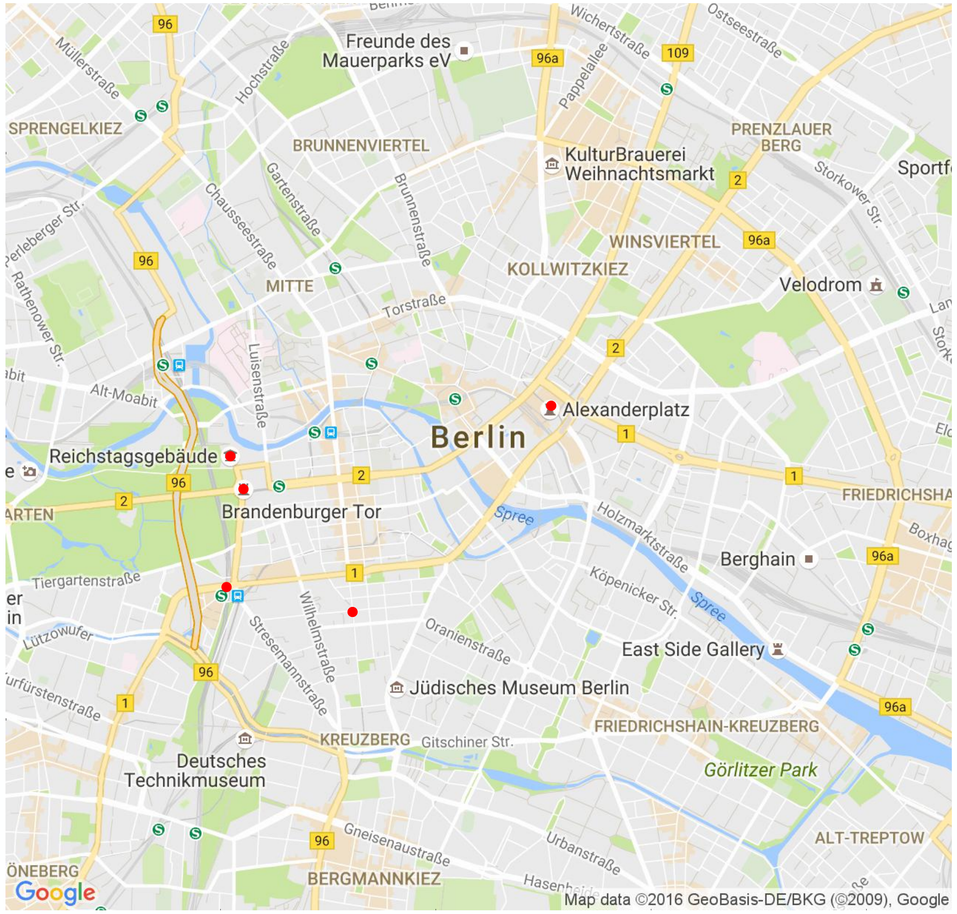
Get coordinates

```
> berlin_sites <- c("Brandenburger Tor", "Potsdamer Platz",
                    "Victory Column Berlin", "Checkpoint Charlie",
                    "Reichstag Berlin", "Alexander Platz")

> xx <- geocode(berlin_sites)
Information from URL : http://maps.googleapis.com/maps/...
Information from URL : ...

> # Add column with cleaned up names
> xx$location <- sub(" Berlin", "", berlin_sites)

> str(xx)
'data.frame':6 obs. of  3 variables:
 $ lon     : num  13.4 13.4 13.4 13.4 13.4 ...
 $ lat     : num  52.5 52.5 52.5 52.5 52.5 ...
 $ location: chr  "Brandenburger Tor" "Potsdamer Platz" ...
```

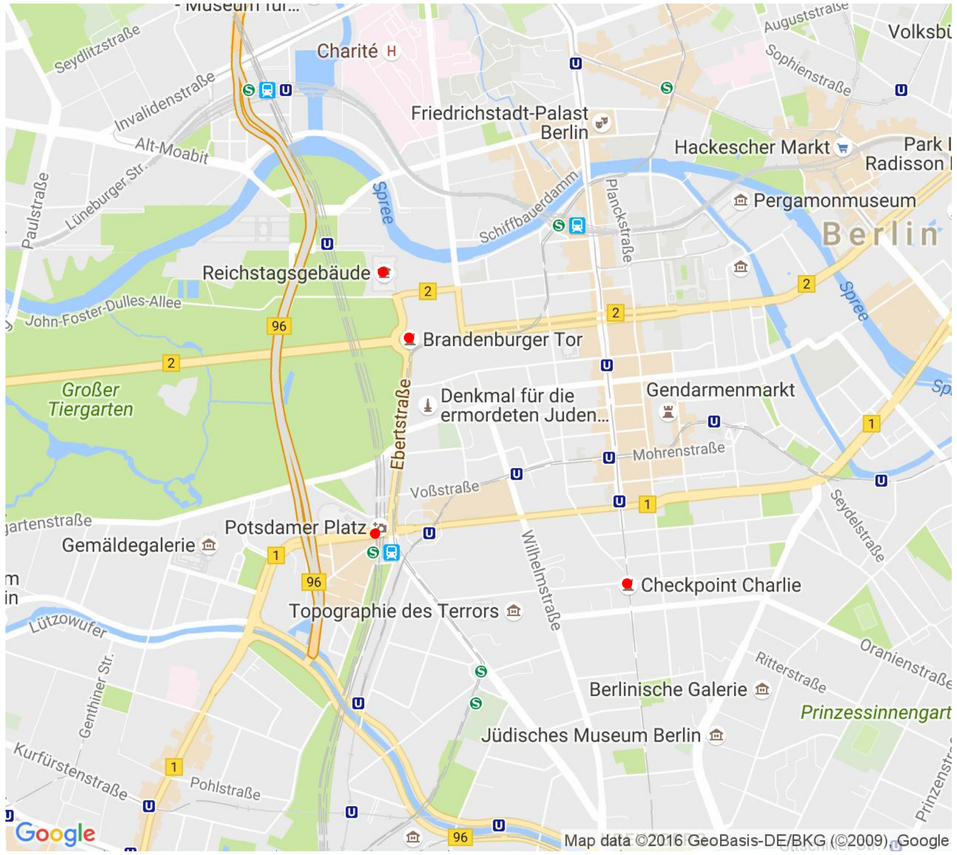
```
> # google/roadmap - zoom = 13
> road_13 <- get_map(location = "Berlin, Germany", zoom = 13,
                     source = "google", maptype = "roadmap")
> ggmap(road_13, extent = "device")
```



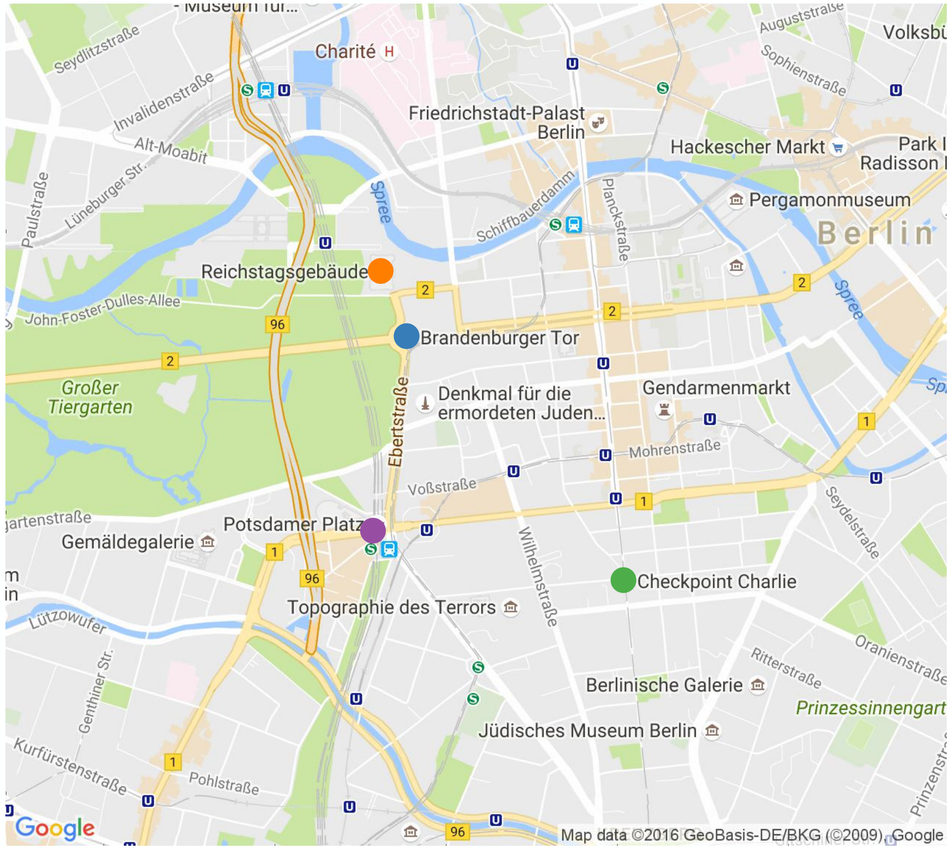
```
> # google/roadmap - zoom = 13
> road_13 <- get_map(location = "Berlin, Germany", zoom = 13,
                     source = "google", maptype = "roadmap")
> ggmap(road_13, extent = "device") +
    geom_point(data = xx, col = "red")
```



```
> bbox <- make_bbox(lon = xx$lon, lat = xx$lat, f = .1)
> boxed_14 <- get_map(location = bbox, zoom = 14,
                      source = "google", maptype = "roadmap")
> ggmap(boxed_14, extent = "device") +
    geom_point(data = xx, col = "red")
```



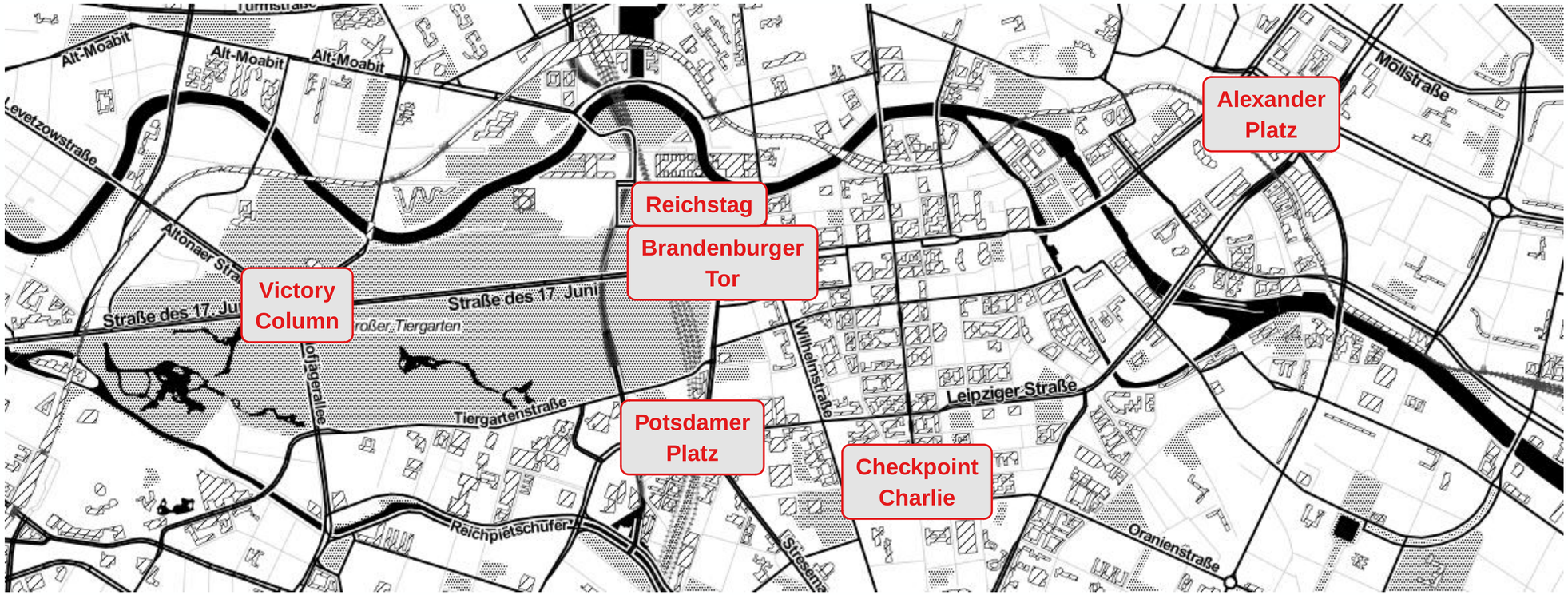
```
> bbox <- make_bbox(lon = xx$lon, lat = xx$lat, f = .1)
> boxed_14 <- get_map(location = bbox, zoom = 14,
                      source = "google", maptype = "roadmap")
> ggmap(boxed_14, extent = "device") +
    geom_point(data = xx, aes(col = location), size = 3) +
    scale_colour_brewer(palette = "Set1")
```



location

- Alexander Platz
- Brandenburger Tor
- Checkpoint Charlie
- Potsdamer Platz
- Reichstag
- Victory Column

Final Plot


Alt-Moabit
Alt-Moabit
Alt-Moabit
Levetzowstraße
Straße des 17. Juni
Straße des 17. Juni
großer Tiergarten
Tiergartenstraße
Reichpietschufer
Wilhelmstraße
Leipziger Straße
Oranienstraße
Moltkestraße
Victory Column
Reichstag
Brandenburger Tor
Potsdamer Platz
Checkpoint Charlie
Alexander Platz

DATA VISUALIZATION WITH GGPLOT2

Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

Animations

Animations

- Dense temporal data
- Great exploratory tool
- Several ways
  - for loop to produce gif
  - `animation`
  - `gganimate`

Motion Chart

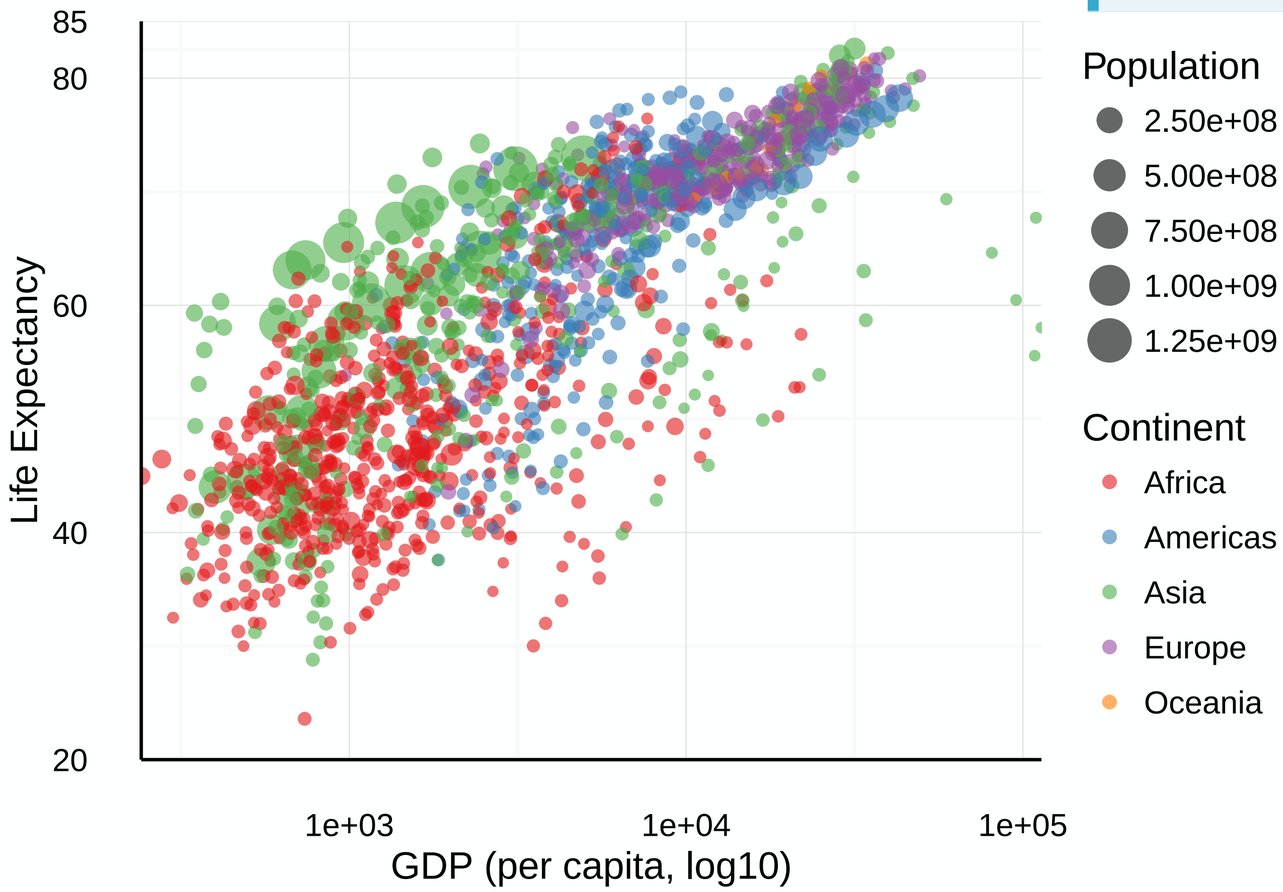
- Hans Rosling
- Karolinska Institute in Stockholm
- Founder of Gapminder
  - UN data
  - Life expectancy, GDP ...

Gapminder data

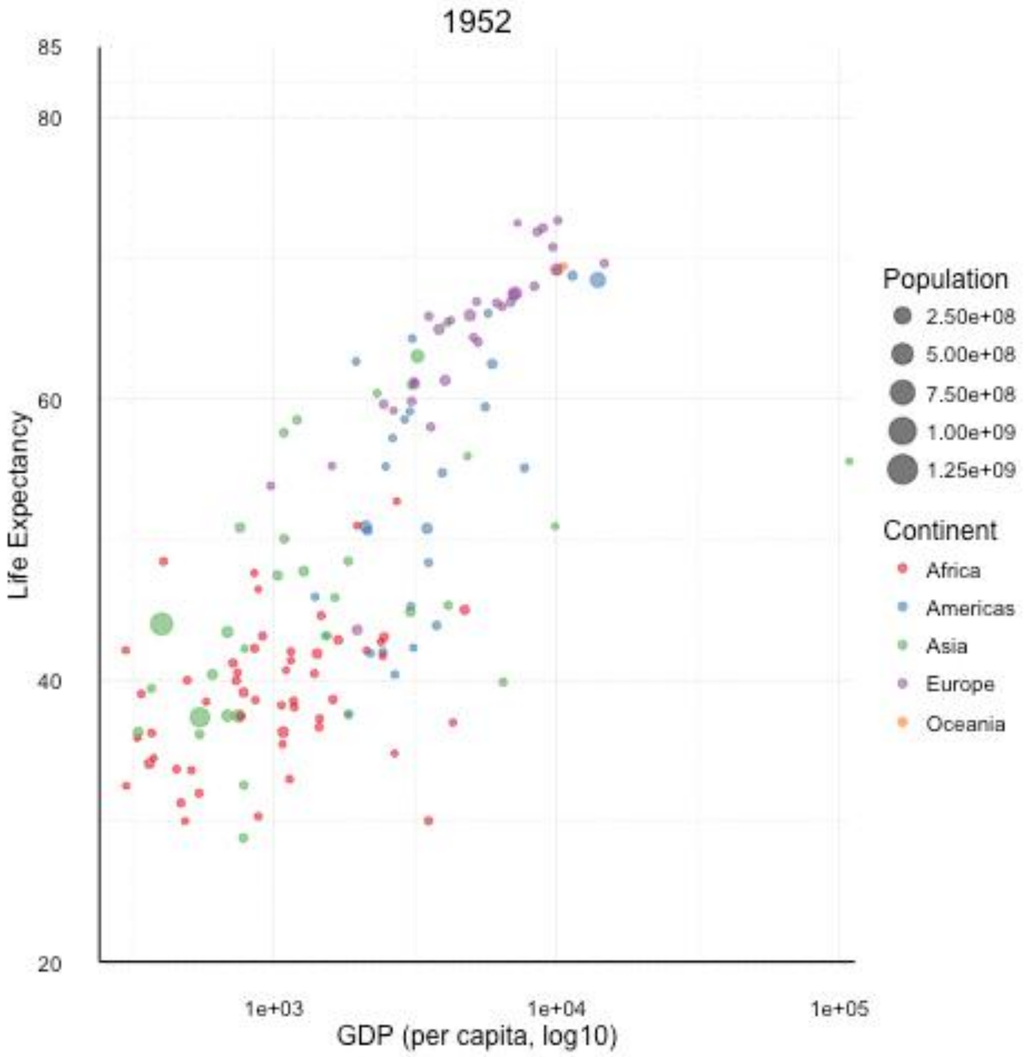
```
> # import tab-delimited data
> gapminder <- read.delim("gapminder.tsv",
                          stringsAsFactors = FALSE)

> str(gapminder)
'data.frame':1704 obs. of  6 variables:
 $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" ...
 $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 ...
 $ pop      : num  8425333 9240934 10267083 11537966 13079460 ...
 $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
 $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
 $ gdpPercap: num  779 821 853 836 740 ...
```

Static plot

```
> ggplot(gapminder, aes(x = gdpPercap,
                        y = lifeExp,
                        colour = continent,
                        size = pop)) +
    geom_point(alpha = 0.6)  # details omitted
```

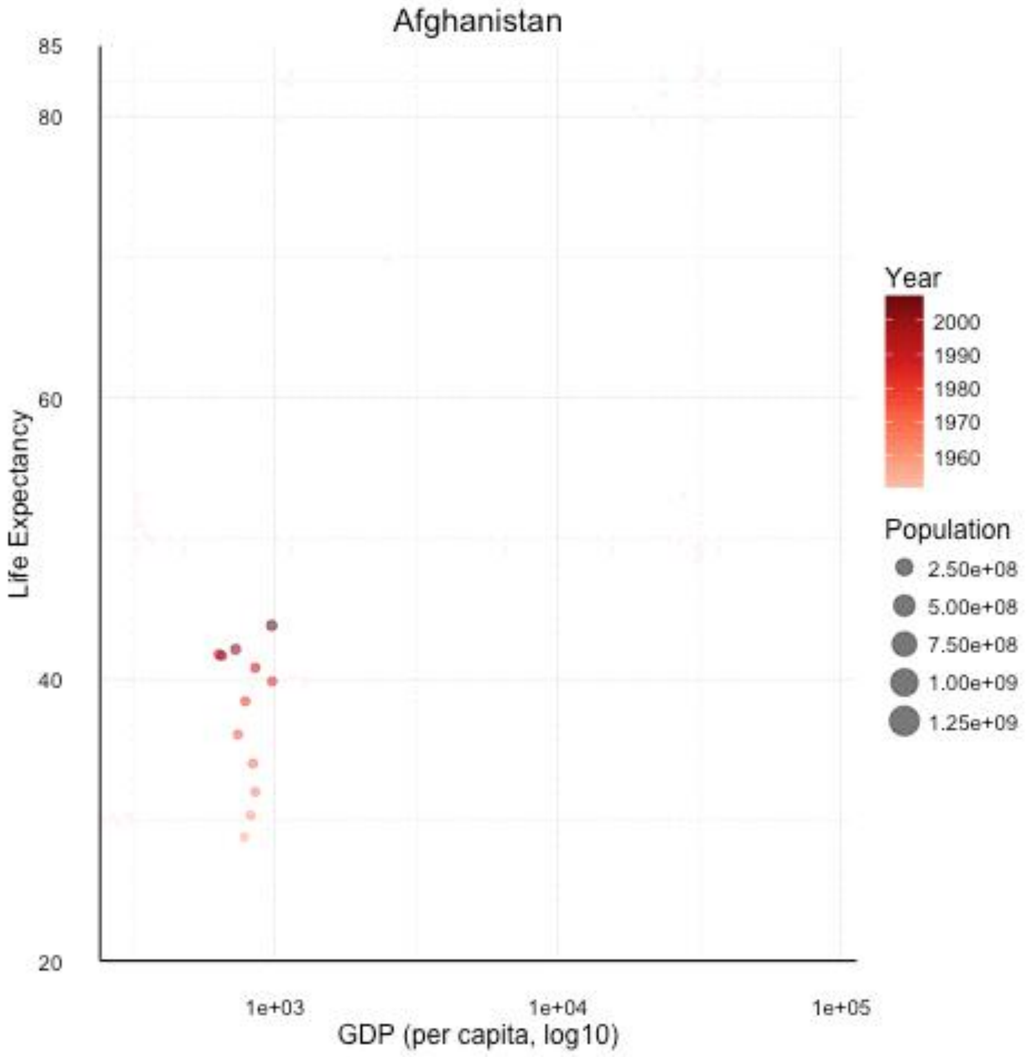


Motion chart (1)



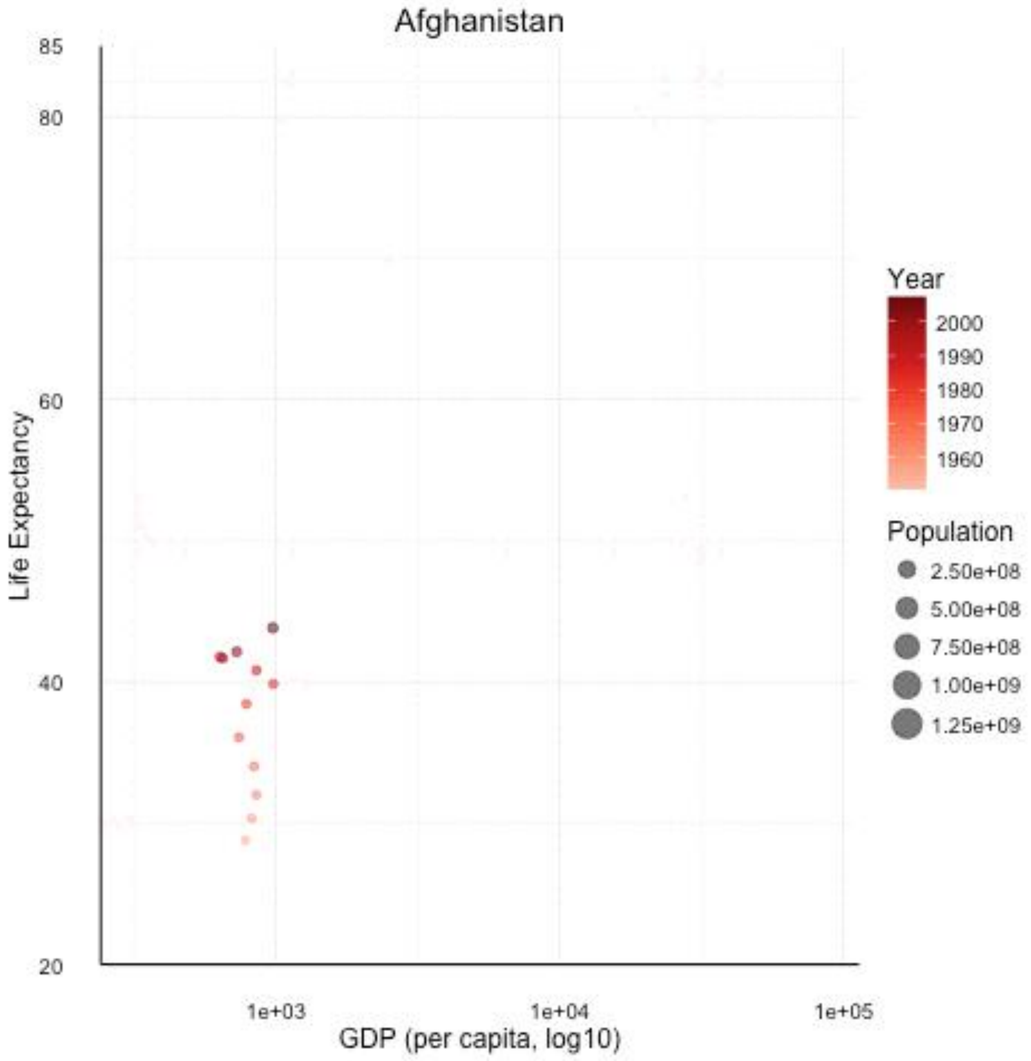

```
> p <- ggplot(gapminder, aes(x = gdpPercap,
                             y = lifeExp,
                             colour = continent,
                             size = pop,
                             frame = year)) +
    geom_point(alpha = 0.6)  # details omitted
> gg_animate(p, "chart1.gif")
```

Motion chart (2)



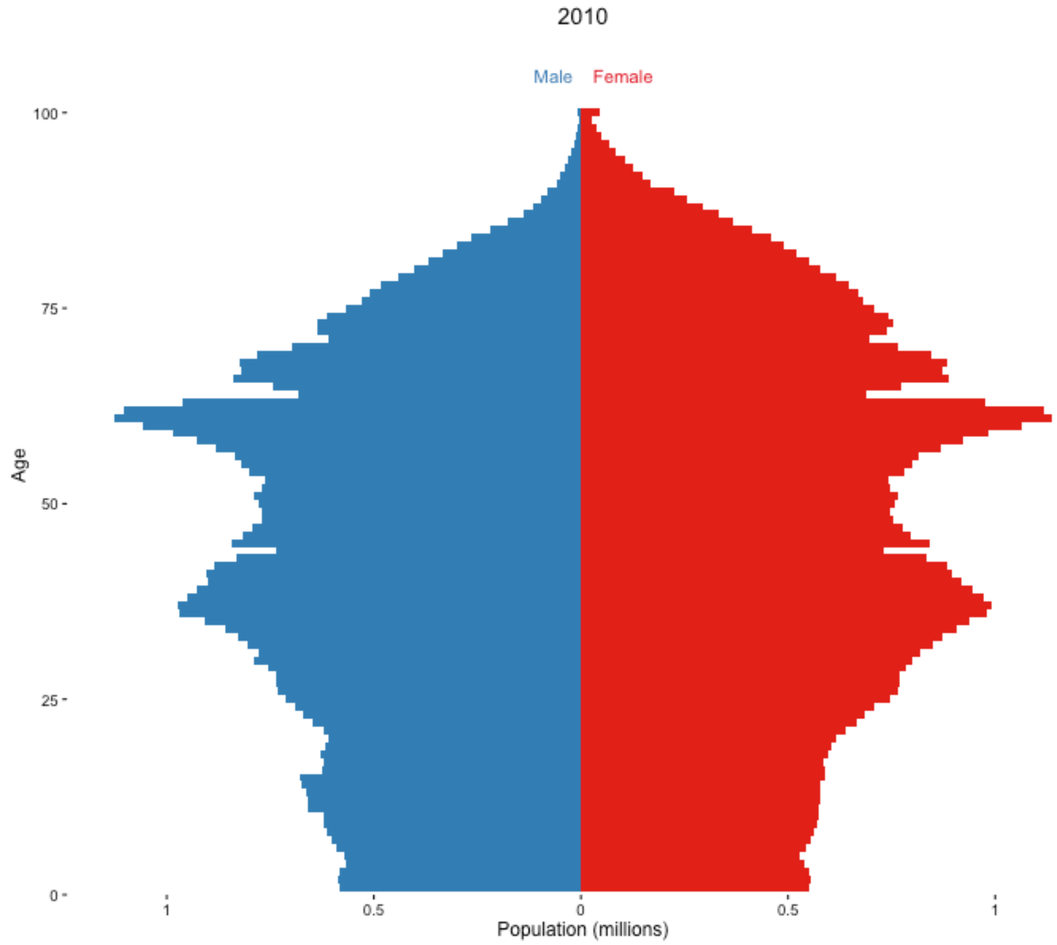

```
> p <- ggplot(gapminder, aes(x = gdpPercap,
                             y = lifeExp,
                             colour = continent,
                             size = pop,
                             frame = country)) +
    geom_point(alpha = 0.6)  # details omitted
> gg_animate(p, "chart2.gif")
```

Motion chart (3)




```
> p <- ggplot(gapminder, aes(x = gdpPercap,
                             y = lifeExp,
                             colour = continent,
                             size = pop,
                             frame = country)) +
    geom_point(alpha = 0.6)  # details omitted
> gg_animate(p, "chart3.gif", interval = 3.0)
```

Population size



Original visualization by Kyle Walker

DATA VISUALIZATION WITH GGPLOT2

Let’s practice!