

more heatmaps

2025-05-03

```
{r} # # Ensure required packages are loaded # library(ggplot2)
# library(dplyr) # library(sf) # library(rnaturalearth) # library(rnaturalearthdata)
# # # Load dataset (if not already loaded) # data <- read.csv('/Users/dhanish/College/Spring 25/Linear Models/4355 Project/Life-Expectancy-Data-Updated.csv')
# # # Load world map # world <- ne_countries(scale = "medium",
returnclass = "sf") # # # Merge world map with your dataset
by country name # world_data <- world %>% # left_join(data,
by = c("name" = "Country")) # # # Heatmap 1: Life Expectancy #
ggplot(world_data) + # geom_sf(aes(fill = Life_expectancy))
+ # scale_fill_viridis_c(option = "plasma", na.value = "lightgrey")
+ # labs(title = "Life Expectancy by Country", fill = "Life
Expectancy") + # theme_minimal() # #
```

```
library(rworldmap)
```

```
## Loading required package: sp
```

```
## ### Welcome to rworldmap ###
```

```
## For a short introduction type : vignette('rworldmap')
```

```
library(RColorBrewer)
```

```
# Load dataset
```

```
data <- read.csv('/Users/dhanish/College/Spring 25/Linear Models/4355 Project/Life-Expectancy-Data-Updated.csv')
joinedData <- joinCountryData2Map(data, joinCode = "NAME", nameJoinColumn = "Country")
```

```
## 2768 codes from your data successfully matched countries in the map
```

```
## 96 codes from your data failed to match with a country code in the map
```

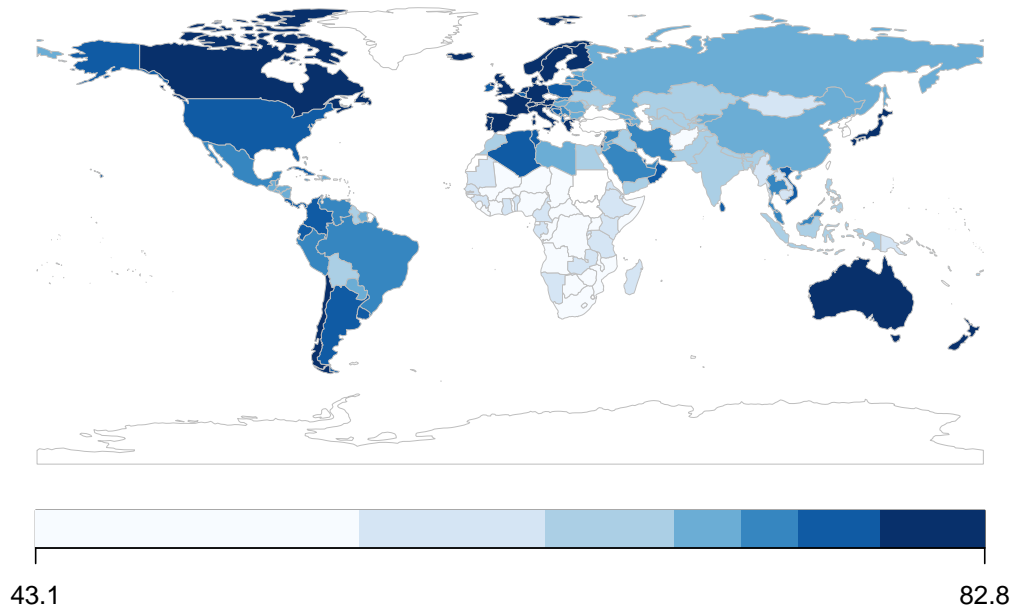
```
## 70 codes from the map weren't represented in your data
```

```
# 1. Life Expectancy - Blue
```

```
mapCountryData(joinedData,
  nameColumnToPlot = "Life_expectancy",
  mapTitle = "Life Expectancy by Country",
  colourPalette = colorRampPalette(brewer.pal(9, "Blues"))(100))
```

```
## Warning in rwmGetColours(colourPalette, numColours): 100 colours specified and
## 7 required, using interpolation to calculate colours
```

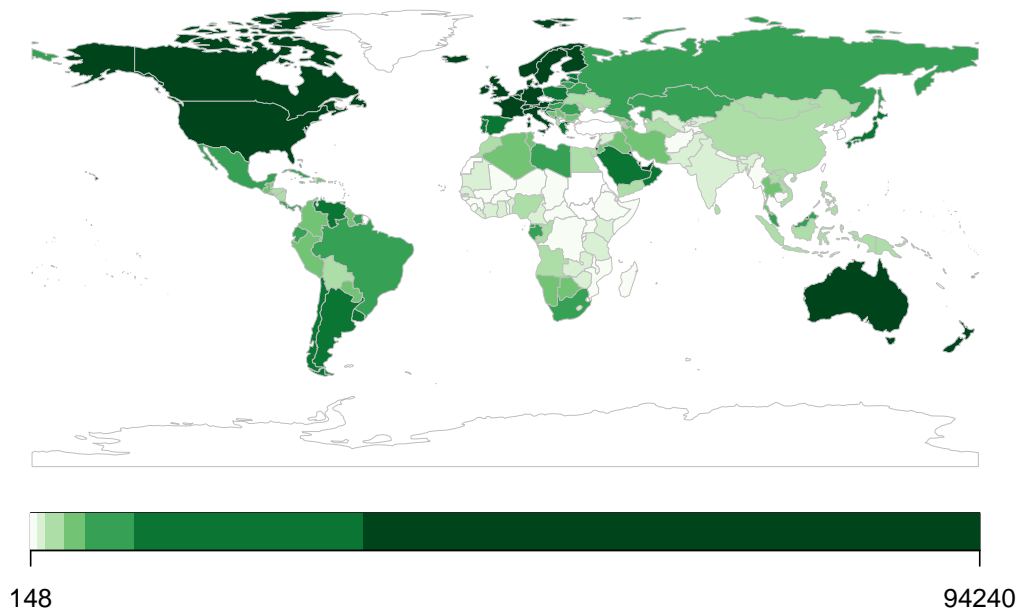
Life Expectancy by Country



```
# 2. GDP per Capita - Green
mapCountryData(joinedData,
  nameColumnToPlot = "GDP_per_capita",
  mapTitle = "GDP per Capita by Country",
  colourPalette = colorRampPalette(brewer.pal(9, "Greens"))(100))
```

```
## Warning in rwmGetColours(colourPalette, numColours): 100 colours specified and
## 7 required, using interpolation to calculate colours
```

GDP per Capita by Country

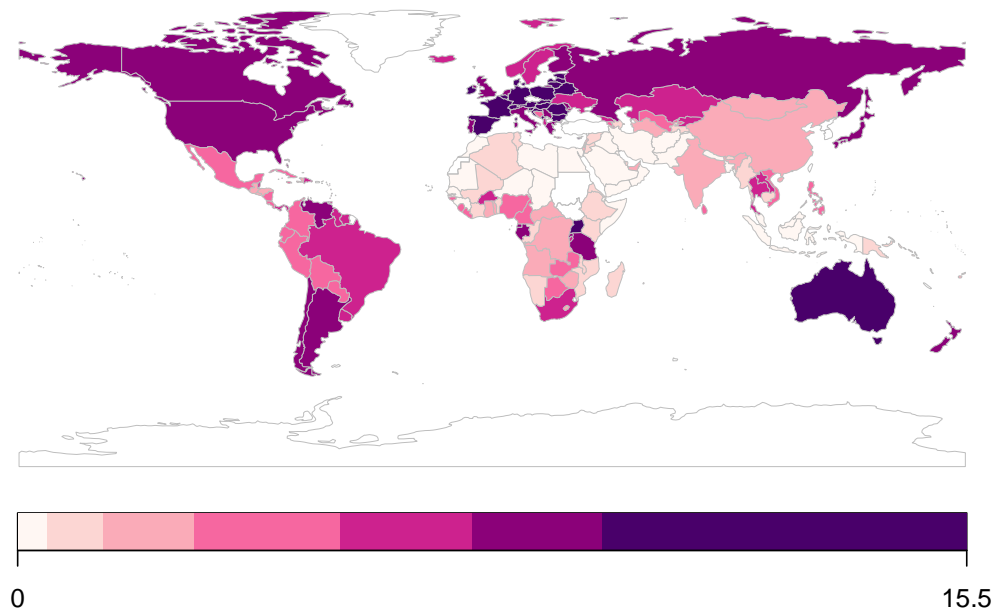


```
# 3. Alcohol Consumption - Pinkish (RdPu)
```

```
mapCountryData(joinedData,  
  nameColumnToPlot = "Alcohol_consumption",  
  mapTitle = "Alcohol Consumption by Country",  
  colourPalette = colorRampPalette(brewer.pal(9, "RdPu"))(100))
```

```
## Warning in rwmGetColours(colourPalette, numColours): 100 colours specified and  
## 7 required, using interpolation to calculate colours
```

Alcohol Consumption by Country

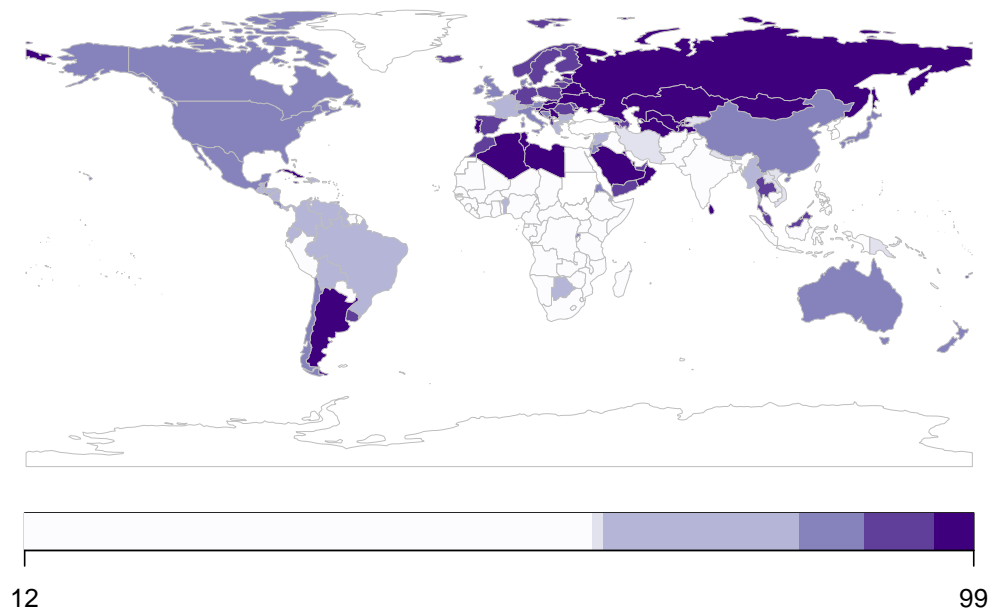


```
# 4. Measles - Purple
mapCountryData(joinedData,
  nameColumnToPlot = "Measles",
  mapTitle = "Measles Cases by Country",
  colourPalette = colorRampPalette(brewer.pal(9, "Purples"))(100))
```

```
## You asked for 7 quantiles, only 6 could be created in quantiles classification
```

```
## Warning in rwmGetColours(colourPalette, numColours): 100 colours specified and
## 6 required, using interpolation to calculate colours
```

Measles Cases by Country

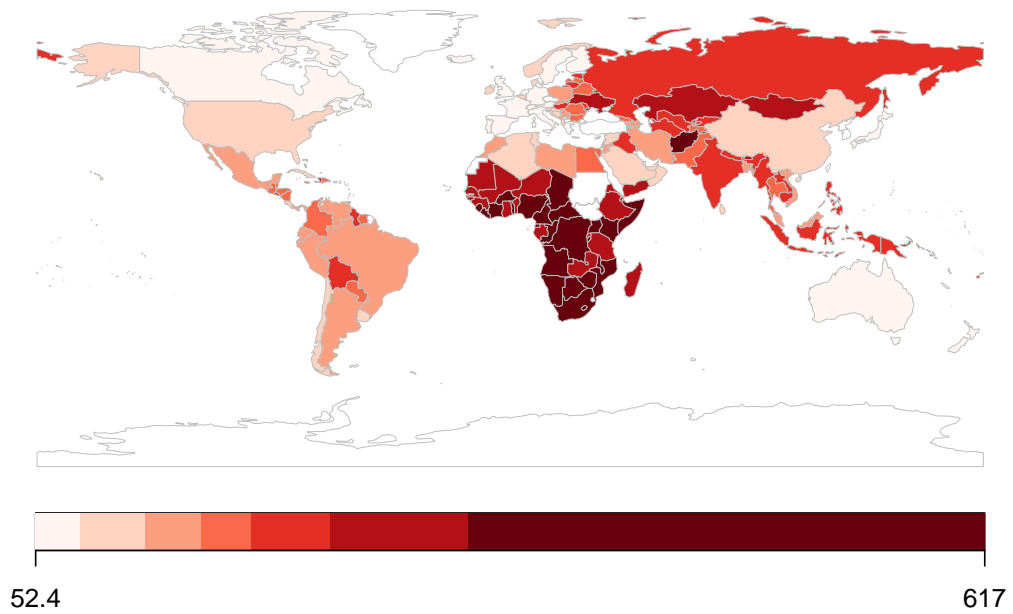


```
# 5. Adult Mortality - Red
```

```
mapCountryData(joinedData,  
  nameColumnToPlot = "Adult_mortality",  
  mapTitle = "Adult Mortality by Country",  
  colourPalette = colorRampPalette(brewer.pal(9, "Reds"))(100))
```

```
## Warning in rwmGetColours(colourPalette, numColours): 100 colours specified and  
## 7 required, using interpolation to calculate colours
```

Adult Mortality by Country



```
# 6. BMI - Lighter Yellow-Orange (YlOrBr)
mapCountryData(joinedData,
  nameColumnToPlot = "BMI",
  mapTitle = "BMI by Country",
  colourPalette = colorRampPalette(brewer.pal(9, "RdBu"))(100))
```

```
## Warning in rwmGetColours(colourPalette, numColours): 100 colours specified and
## 7 required, using interpolation to calculate colours
```

BMI by Country

