Life Expectancy Example

Meshal AlQahtani

Introduction

Below are example graphs created using RStudio and published with Quarto. Over the next month, I will frequently share simple and easy-to-create visualizations using RStudio.

Loading My data

You can find the data I used here: Global Health Observatory

```
install.packages("devtools", repos = "https://cloud.r-project.org")
```

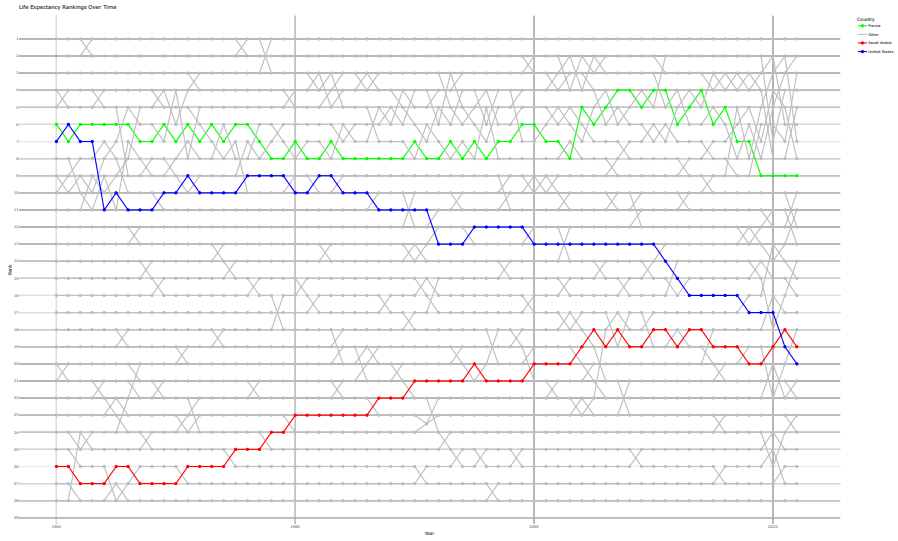
```
package 'devtools' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Admin\AppData\Local\Temp\RtmpW8FMij\downloaded_packages
```

```
library(streamgraph)
library(dplyr)
library(readr)

# Set the file path
library(here)
file_path <- here("data", "Long_Format_Life_Expectancy_Data_for_Streamgraph.csv")
data <- read_csv(file_path)
```

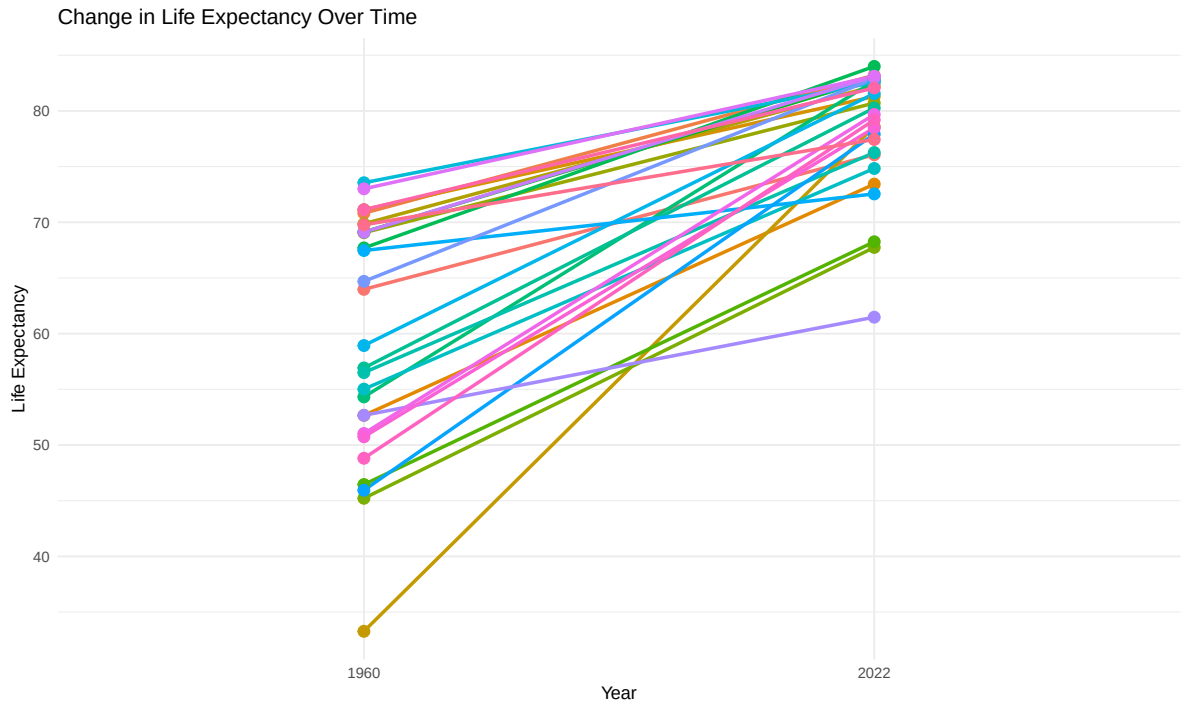
the LE over the years among G20 countries

```r
library(ggplot2)
library(ggrepel)
library(dplyr)
library(readr)

# Load your data
library(here)
file_path <- here("data", "Long_Format_Life_Expectancy_Data_for_Streamgraph.csv")
data <- read_csv(file_path)

# Rank countries by Life Expectancy for each year
data_ranked <- data %>%
  group_by(year) %>%
  mutate(Rank = rank(-Life_Expectancy)) %>%  # Rank descending
  ungroup()

# Highlight specific countries
highlight_countries <- c("Saudi Arabia", "United States", "France")
data_ranked <- data_ranked %>%
  mutate(Highlight = ifelse(Country %in% highlight_countries, Country, "Other"))

# Create the bump chart
p <- ggplot(data_ranked, aes(x = year, y = Rank, group = Country, color = Highlight)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  scale_y_reverse(breaks = 1:nrow(data_ranked), minor_breaks = NULL) +  # Reverse ranks
  theme_minimal() +
  scale_color_manual(values = c("Saudi Arabia" = "red", "United States" = "blue", "France" =
  labs(
    title = "Life Expectancy Rankings Over Time",
    x = "Year",
    y = "Rank",
    color = "Country"
  )

# Display with ggrepel for labeling (optional)
p <- p + geom_text_repel(data = filter(data_ranked, year == max(year)),
                         aes(label = Country),
                         nudge_x = 0.5, size = 3)

# Make it interactive with plotly
```

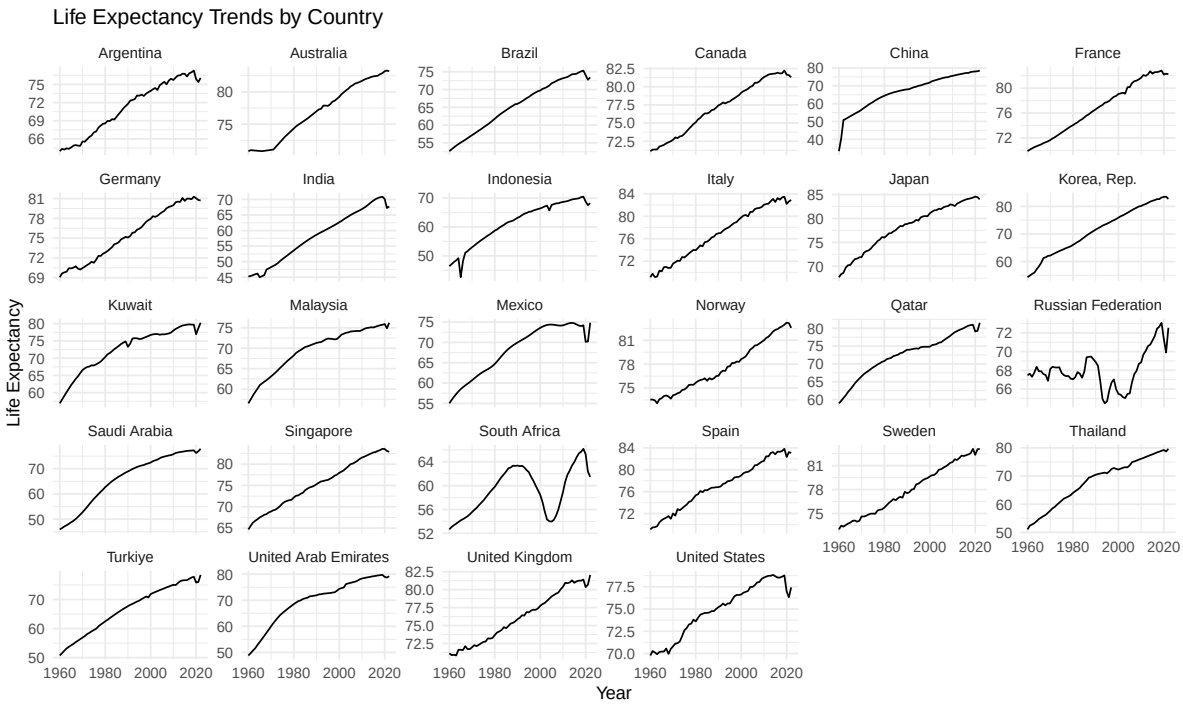
```
library(plotly)
ggplotly(p)
```



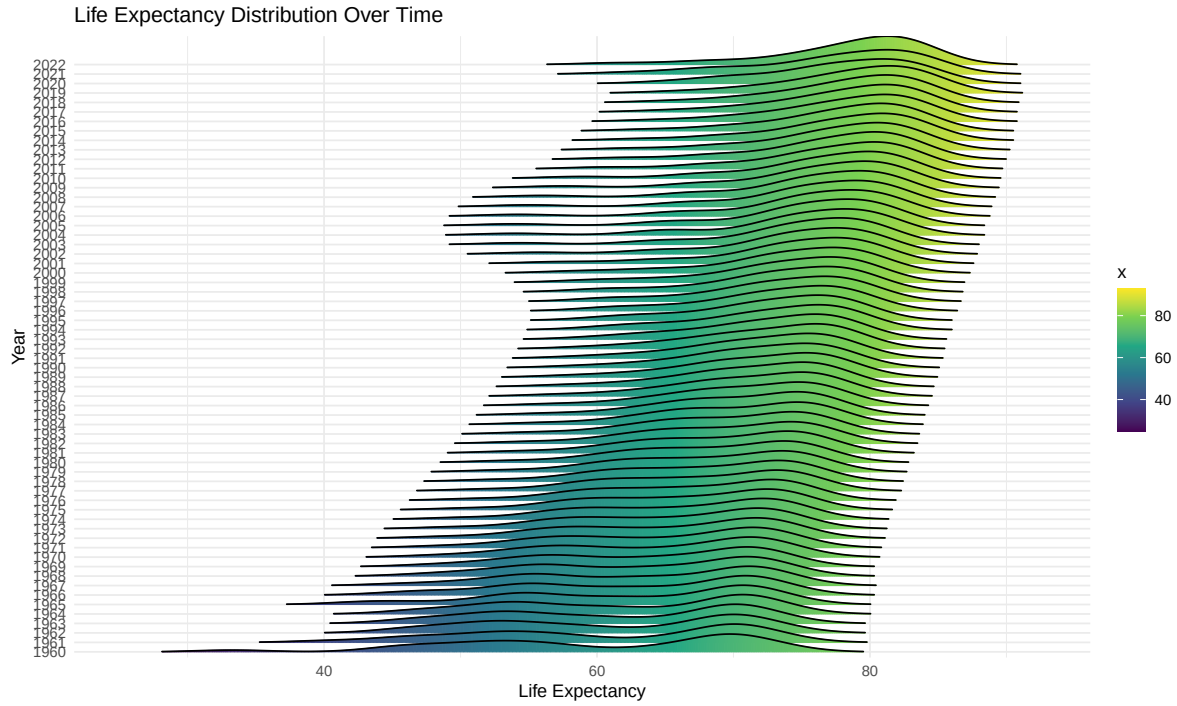

```
library(ggplot2)
library(dplyr)

# Select two years (e.g., first and last available years)
selected_years <- data %>%
  filter(year %in% c(min(year), max(year))) %>%
  mutate(year = as.factor(year))

ggplot(selected_years, aes(x = year, y = Life_Expectancy, group = Country, color = Country))
  geom_line(size = 1) +
  geom_point(size = 3) +
  theme_minimal() +
  labs(
    title = "Change in Life Expectancy Over Time",
    x = "Year",
    y = "Life Expectancy"
  ) +
  theme(legend.position = "none")
```




```
library(ggplot2)
library(ggrepel)

ggplot(data, aes(x = year, y = Life_Expectancy, group = Country)) +
  geom_line() +
  facet_wrap(~Country, scales = "free_y") +
  theme_minimal() +
  labs(title = "Life Expectancy Trends by Country", x = "Year", y = "Life Expectancy")
```

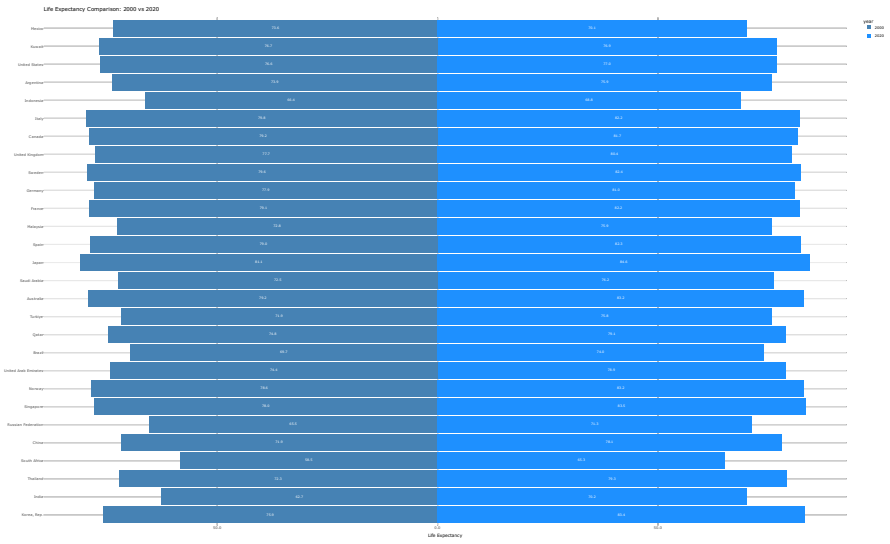



```r
library(ggridges)

ggplot(data, aes(x = Life_Expectancy, y = as.factor(year), fill = ..x..)) +
  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01) +
  scale_fill_viridis_c() +
  theme_minimal() +
  labs(title = "Life Expectancy Distribution Over Time", x = "Life Expectancy", y = "Year")
```




```
library(ggplot2)
library(dplyr)
library(readr)
library(plotly)  # For interactivity

# Load data
library(here)
file_path <- here("data", "Long_Format_Life_Expectancy_Data_for_Streamgraph.csv")

data <- read_csv(file_path)

# Filter for 2000 and 2020
selected_years <- data %>%
  filter(year %in% c(2000, 2020)) %>%
  mutate(year = as.factor(year),
         Life_Expectancy = ifelse(year == "2000", -Life_Expectancy, Life_Expectancy))  # Mir

# Create face-to-face bar chart
p <- ggplot(selected_years,
            aes(x = reorder(Country, -Life_Expectancy),
                y = Life_Expectancy,
                fill = year,
```

```
                text = paste("Country:", Country, "<br>Year:", year, "<br>Life Expectancy:",
  geom_bar(stat = "identity") +
  geom_text(aes(label = sprintf("%.1f", abs(Life_Expectancy))),  # Format to 1 decimal place
            position = position_stack(vjust = 0.5),
            size = 3, color = "white") +  # Display LE inside bars
  coord_flip() +
  theme_minimal() +
  labs(title = "Life Expectancy Comparison: 2000 vs 2020",
       x = "",
       y = "Life Expectancy") +
  scale_fill_manual(values = c("2000" = "steelblue", "2020" = "dodgerblue")) +  # Blue shades
  theme(legend.position = "top") +
  scale_y_continuous(labels = function(x) sprintf("%.1f", abs(x)))  # Format y-axis labels to

# Convert to interactive plot
interactive_plot <- ggplotly(p, tooltip = "text")

# Display
interactive_plot
```



GDPR Visualization from :
https://github.com/davidsjoberg/ggbump?tab=readme-ov-file

```
options(repos = c(CRAN = "https://cloud.r-project.org"))

# Load necessary packages
install.packages("pacman")
```

```
package 'pacman' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Admin\AppData\Local\Temp\RtmpW8FMij\downloaded_packages
```

```
install.packages("rnaturalearthdata")
```

```
package 'rnaturalearthdata' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\Admin\AppData\Local\Temp\RtmpW8FMij\downloaded_packages
```

```
pacman::p_load(BBmisc, tidyverse, hablar, ggbump, sf, rnaturalearth, feather, janitor, lubric
options(stringsAsFactors = F)

# Load GDPR violations data
gdpr_violations <- readr::read_tsv('https://raw.githubusercontent.com/rfordatascience/tidytu

# Process data
df <- gdpr_violations %>%
  group_by(name) %>%
  summarise(price = sum(price)) %>%
  ungroup()

sdf <- rnaturalearthdata::countries50 %>%
  st_as_sf() %>%
  st_crop(xmin = -24, xmax = 31, ymin = 33, ymax = 73) %>%
  filter(admin %in% df$name) %>%
  left_join(df, by = c("admin" = "name")) %>%
  mutate(price_cap  = price / pop_est,
         admin = case_when(admin == "United Kingdom" ~ "UK",
                           admin == "Czech Republic" ~ "Czech",
```

```
                            TRUE ~ admin))

# Create ranking data
ranking <- st_geometry(sdf) %>%
  st_point_on_surface() %>%
  st_coordinates() %>%
  as_tibble() %>%
  bind_cols(tibble(fine_cap = normalize(rank(sdf$price_cap), range = c(40.12161, 66.12161), r
                   country = sdf$admin,
                   xend = 60,
                   x_axis_start = xend + 10,
                   fine_cap_x = normalize(sdf$price_cap, range = c(first(x_axis_start), 100)
                   val_txt = paste0(format(sdf$price_cap, digits = 2, nsmall = 2)),
                   val_txt2 = if_else(country == "Austria", paste0(val_txt, "€ per capita"),

sdf <- sdf %>%
  bind_cols(ranking %>% select(fine_cap))

# Create the plot
ggplot() +
  geom_sf(data = sdf, size = .3, fill = "transparent", color = "gray17") +
  geom_sigmoid(data = ranking,
               aes(x = X, y = Y, xend = x_axis_start - .2, yend = fine_cap, group = country,
               alpha = .6, smooth = 10, size = 1) +
  geom_segment(data = ranking,
               aes(x = x_axis_start, y = fine_cap, xend = fine_cap_x, yend = fine_cap, color
               lineend = "round") +
  geom_segment(data = ranking,
               aes(x = x_axis_start, y = 40, xend = x_axis_start, yend = 67), alpha = .6, siz
  geom_point(data = ranking,
             aes(x = X, y = Y, color = fine_cap), size = 2) +
  geom_text(data = ranking, aes(x = x_axis_start-.5, y = fine_cap, label = country, color = f
  geom_text(data = ranking, aes(x = fine_cap_x, y = fine_cap, label = val_txt2, color = fine_
  coord_sf(clip = "off") +
  scale_fill_viridis_c() +
  scale_color_viridis_c() +
  theme_void() +
  labs(title = "GDPR fines per capita",
       subtitle = str_wrap("The General Data Protection Regulation (EU) 2016/679 (GDPR) is a
       caption = "Source: TidyTuesday & Wikipedia") +
  theme(plot.margin = margin(.5, 1, .5, .5, "cm"),
        legend.position = "none",
```

```
        plot.background = element_rect(fill = "black"),
        plot.caption = element_text(color = "gray40"),
        plot.title = element_text(color = "gray40", size = 16, family = "Helvetica", face = "
        plot.subtitle = element_text(color = "gray40", size = 8))
```