ST237 Assignment 3

u2112531

Packages

Load any packages you need at the top of the notebook in this code chunk.

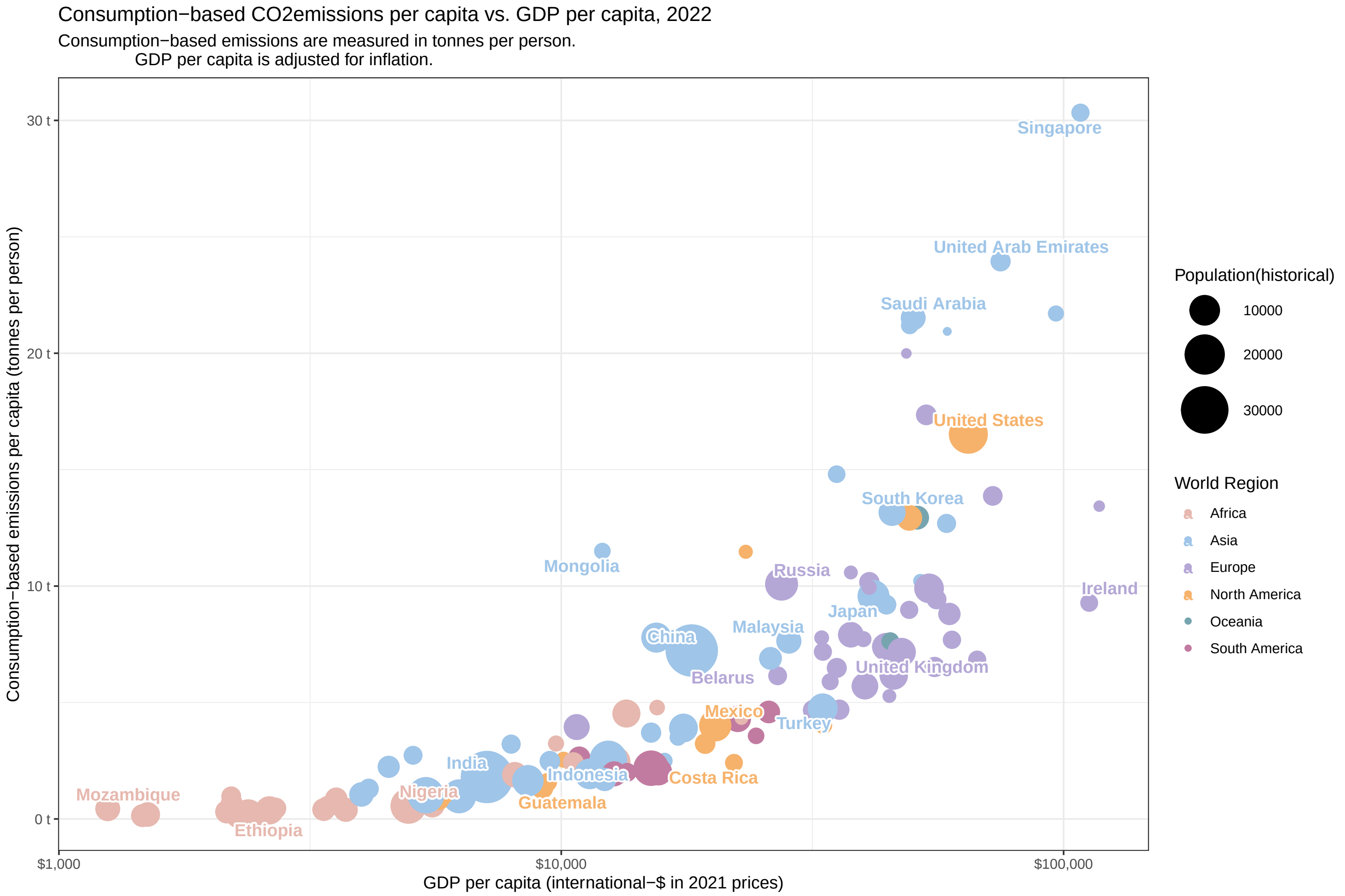
```
library(readxl)
library(ggplot2)
library(dplyr)
library(readr)
library(scales)
library(tidyr)
library(ggrepel)
library(tidyverse)
library(zoo)
library(lubridate)
library(ggstream)
```

Q1(a) Reproduce the plot

```
# Load dataset
data <- read_csv("ST237_co2_gdp_cleaned.csv", show_col_types = FALSE)

# Filter for year 2022
data_2022 <- data %>%
  filter(Year == 2022) %>%
  drop_na(Per.capita.consumption.based.CO..emissions,
          GDP.per.capita..PPP..constant.2017.international...,
          Population..historical.) %>%
  mutate(Population_scaled = sqrt(Population..historical.))
```

```
# Define highlighted countries
highlight_countries <- c("Singapore", "United Arab Emirates", "Saudi Arabia", "United States",
                         "South Korea", "Ireland", "Japan", "Russia", "Mongolia", "China",
                         "Malaysia", "United Kingdom", "Belarus", "Turkey", "Mexico",
                         "India", "Indonesia", "Costa Rica", "Mozambique", "Ethiopia",
                         "Nigeria", "Guatemala")

# Define color palette
soft_palette <- c("Africa" = "#e6b8af", "Asia" = "#9fc5e8",
                  "Europe" = "#b4a7d6", "North America" = "#f6b26b",
                  "Oceania" = "#76a5af", "South America" = "#c27ba0")
```

```r
# Create scatter plot
ggplot(data_2022, aes(x = GDP.per.capita..PPP..constant.2017.international...,
                       y = Per.capita.consumption.based.CO..emissions,
                       size = Population_scaled,
                       color = World.regions.according.to.OWID)) +

  geom_point(alpha = 1) +

  geom_text_repel(data = data_2022 %>% filter(Entity %in% highlight_countries),
                  aes(label = Entity),
                  size = 4,
                  box.padding = 0.3,
                  segment.color = "gray50",
                  segment.size = 0.3,
                  fontface = "bold",
                  bg.color = "white",
                  bg.r = 0.15) +

  scale_x_log10(labels = scales::label_dollar()) +

  scale_y_continuous(labels = scales::label_number(suffix = " t")) +

  scale_size(range = c(2, 15)) +

  scale_color_manual(values = soft_palette) +

  theme_bw() +
  labs(
    title = "Consumption-based CO emissions per capita vs. GDP per capita, 2022",
    subtitle = "Consumption-based emissions are measured in tonnes per person.
                GDP per capita is adjusted for inflation.",
    x = "GDP per capita (international-$ in 2021 prices)",
    y = "Consumption-based emissions per capita (tonnes per person)",
    size = "Population(historical)",
    color = "World Region"
  ) +
  theme(legend.position = "right")
```



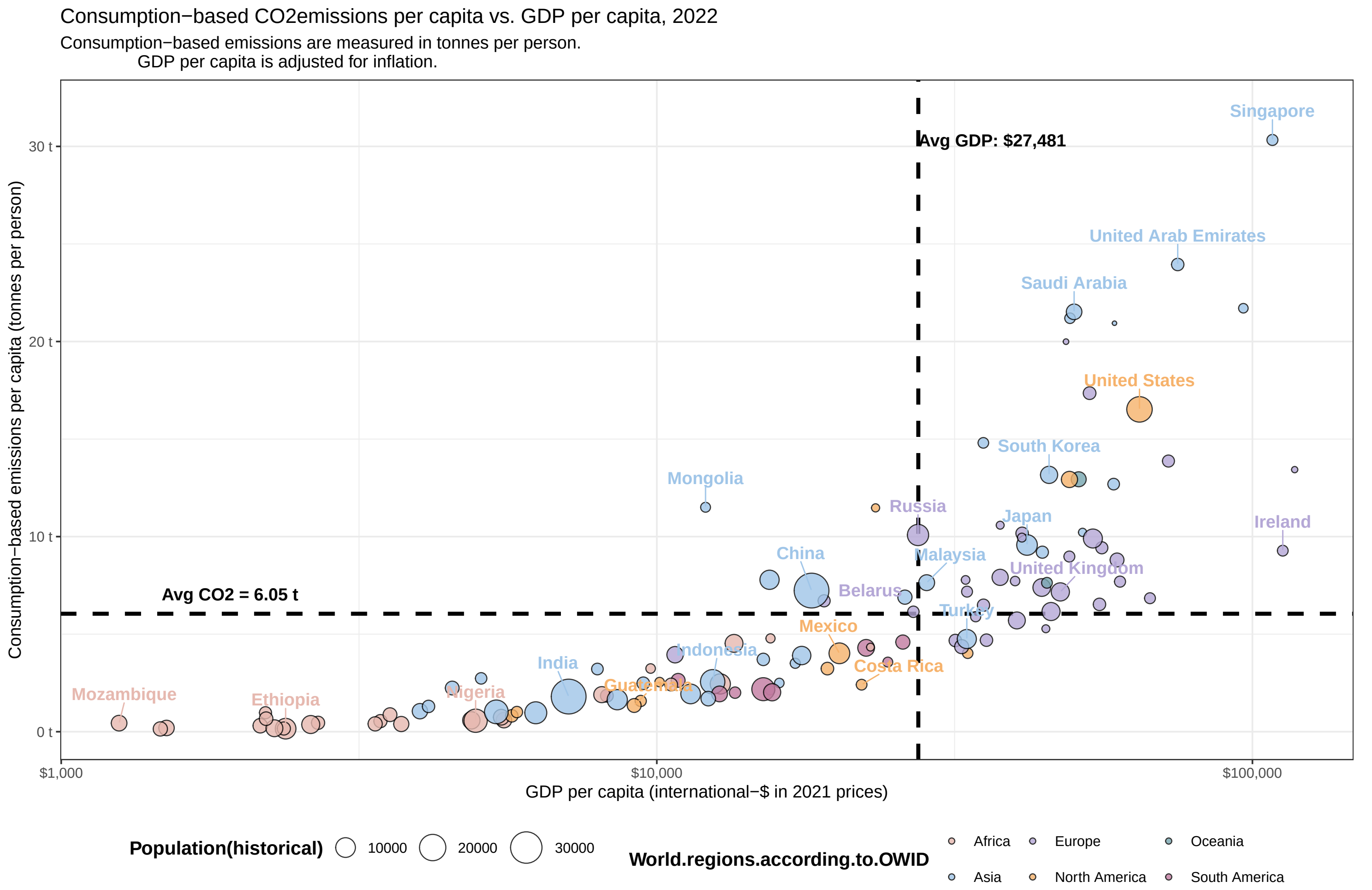
This plot shows a clear positive correlation between GDP per capita and CO emissions per capita. High-income countries generally produce more CO emissions per person.

Q1(b) Create your own improved visualisation

```
# Compute mean values
avg_gdp <- mean(data_2022$GDP.per.capita..PPP..constant.2017.international..., na.rm = TRUE)
avg_co2 <- mean(data_2022$Per.capita.consumption.based.CO..emissions, na.rm = TRUE)
```

```
# Create improved scatter plot
ggplot(data_2022, aes(x = GDP.per.capita..PPP..constant.2017.international...,
                       y = Per.capita.consumption.based.CO..emissions,
                       size = Population_scaled,
                       fill = World.regions.according.to.OWID,
                       color = World.regions.according.to.OWID)) +

  geom_hline(yintercept = avg_co2, linetype = "dashed", color = "black", size = 1.2) +
  geom_vline(xintercept = avg_gdp, linetype = "dashed", color = "black", size = 1.2) +

  annotate("text", x = min(data_2022$GDP.per.capita..PPP..constant.2017.international...) * 2,
           y = avg_co2 +1, label = paste("Avg CO  =", round(avg_co2, 2), "t"),
           hjust = 1, color = "black", size = 4, fontface = "bold") +
```

```r
  annotate("text", x = avg_gdp,
           y = max(data_2022$Per.capita.consumption.based.CO..emissions),
           label = paste("Avg GDP:", scales::dollar(round(avg_gdp, 0))),
           hjust = 0, color = "black", size = 4, fontface = "bold") +

  geom_point(shape = 21, color = "black", stroke = 0.5, alpha = 0.8) +

geom_text_repel(data = data_2022 %>% filter(Entity %in% highlight_countries),
                aes(label = Entity),
                size = 4, fontface = "bold",
                box.padding = 0.5,
                nudge_y = 1.5,
                segment.size = 0.4,
                max.overlaps = 100,
                force = 10) +

  scale_x_log10(labels = scales::label_dollar()) +
  scale_y_continuous(labels = scales::label_number(suffix = " t")) +

  scale_size(range = c(1, 10)) +

  scale_fill_manual(values = soft_palette) +
  scale_color_manual(values = soft_palette) +

  theme_bw() +
  theme(legend.position = "bottom", legend.title = element_text(size = 12, face = "bold")) +

  labs(
    title = "Consumption-based CO emissions per capita vs. GDP per capita, 2022",
    subtitle = "Consumption-based emissions are measured in tonnes per person.
                GDP per capita is adjusted for inflation.",
    x = "GDP per capita (international-$ in 2021 prices)",
    y = "Consumption-based emissions per capita (tonnes per person)",
    size = "Population(historical)",
    color = "World Region"
  ) +
  guides(color = "none") +
  theme(legend.position = "bottom")
```



The dashed lines indicate the average global CO emissions and GDP per capita. Countries above and to the right of these lines have above-average emissions and economic output.

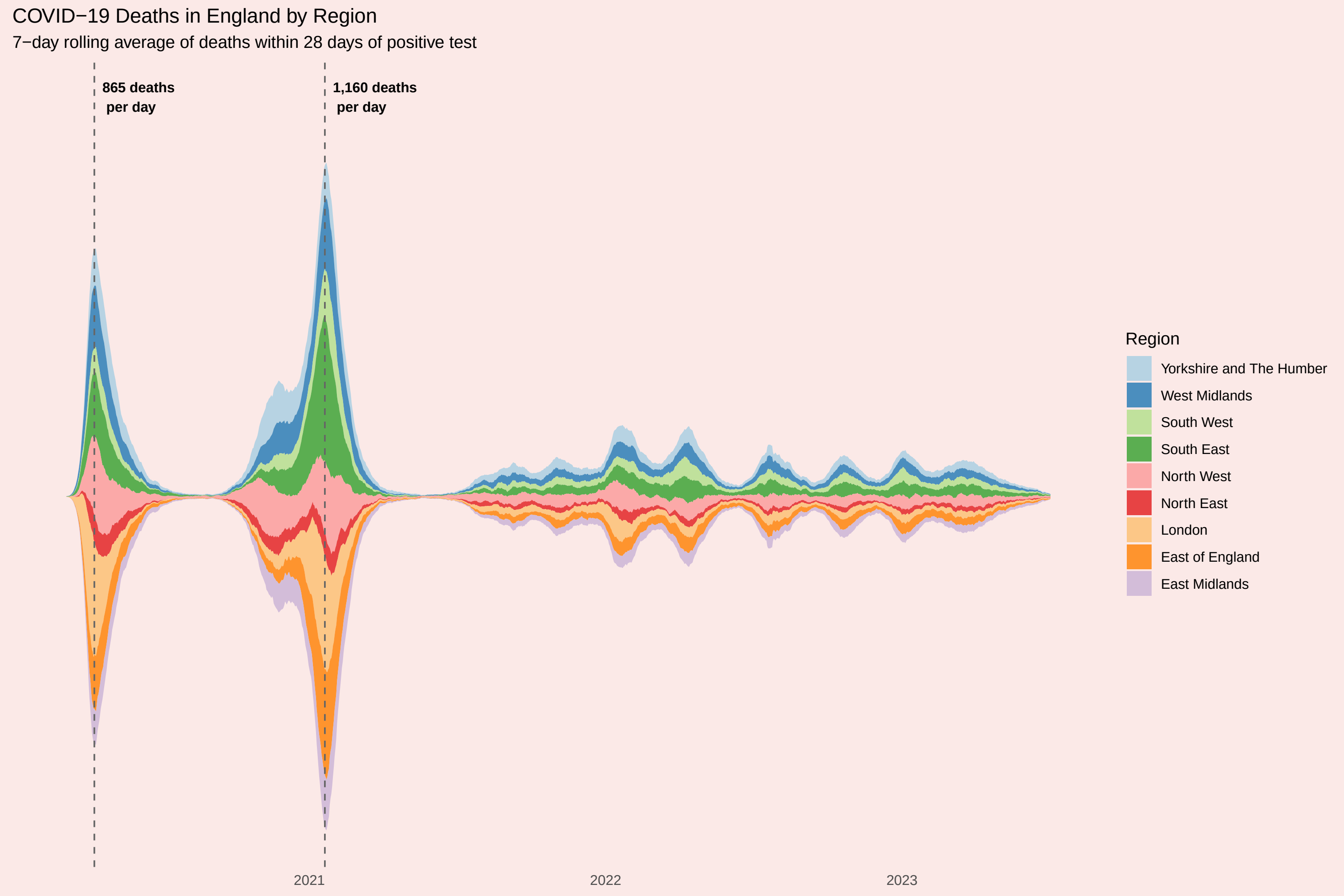
Q2(a) Reproduce the plot

```r
# Compute 7-day rolling average
full_data <- full_data %>%
  group_by(area_name) %>%
  arrange(date) %>%
  mutate(value_smooth = rollmean(value, k = 7, fill = 0, align = "right")) %>%
  ungroup()
```

```r
# Compute stacked ribbon chart
full_data <- full_data %>%
  group_by(date) %>%
  arrange(desc(area_name)) %>%
  mutate(y_offset = sum(value_smooth) / 2,
         ymin = y_offset - cumsum(value_smooth),
         ymax = ymin + value_smooth) %>%
  ungroup()
```

```r
# Define key events
highlight_dates <- as.Date(c("2020-04-11", "2021-01-20"))
highlight_values <- full_data %>%
  filter(date %in% highlight_dates) %>%
  group_by(date) %>%
  summarise(national_rolling_avg = sum(value_smooth, na.rm = TRUE))
```

```r
# Plot COVID-19 deaths with key events
ggplot(full_data, aes(x = date, ymin = ymin, ymax = ymax,
                      fill = factor(area_name, levels = unique(area_name)))) +

  geom_ribbon(alpha = 0.8) +

  geom_vline(xintercept = highlight_dates, linetype = "dashed", color = "gray40") +

  annotate("text", x = highlight_dates[1]+10, y = 700,
           label = paste(format(round(highlight_values$national_rolling_avg[1], 0),
                                big.mark=","), "deaths", "\n", "per day"),
           color = "black", hjust = 0, size = 3, fontface = "bold") +
  annotate("text", x = highlight_dates[2]+10, y = 700,
           label = paste(format(round(highlight_values$national_rolling_avg[2], 0),
                                big.mark=","), "deaths", "\n", "per day"),
           color = "black", hjust = 0, size = 3, fontface = "bold") +

  scale_fill_brewer(palette = "Paired") +

  scale_x_date(labels = scales::date_format("%Y"), breaks = scales::pretty_breaks(n = 5)) +

  theme_minimal() +
  theme(
    panel.background = element_rect(fill = "#FBE9E7", color = NA),
    plot.background = element_rect(fill = "#FBE9E7", color = NA),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.title.y = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.title.x = element_blank(),
    legend.position = "right"
  ) +

  labs(
    title = "COVID-19 Deaths in England by Region",
    subtitle = "7-day rolling average of deaths within 28 days of positive test",
    fill = "Region"
  )
```



The graph shows the trends in COVID-19 deaths across England's regions. The two vertical lines indicate key peaks in deaths in 2020 and 2021.

Q2(b) Create your own improved visualisation

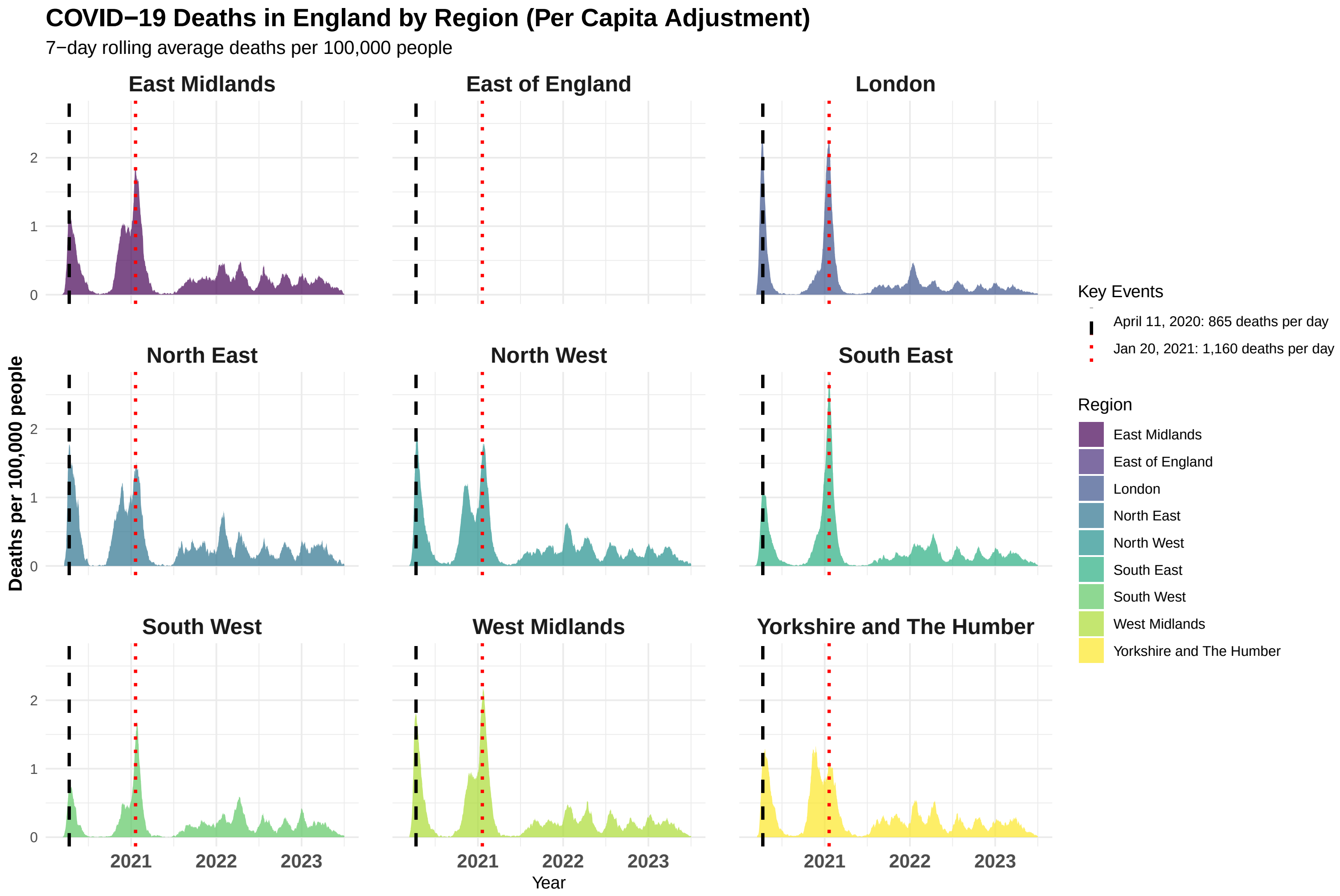
```r
# Load regional population estimates
pop_data <- read_excel("regionalpopestimatesenglandandwales19712022.xlsx",
                       sheet = "Table 4", col_names = TRUE, skip = 2) %>%
  filter(Year == 2022) %>%
  select(Government_Office_Regions = `Government Office Regions`, population = `All Persons`)
```

```r
# Merge COVID-19 data with population data
merged_data <- covid_data %>%
  left_join(pop_data, by = c("area_name" = "Government_Office_Regions")) %>%
  mutate(per_capita_deaths = (value / population) * 100000)
```

```r
# Compute rolling average per capita
merged_data <- merged_data %>%
  group_by(area_name) %>%
  arrange(date) %>%
  mutate(rolling_avg = zoo::rollmean(per_capita_deaths, k = 7, fill = 0, align = "right")) %>%
  ungroup()
```

```r
# Key dates for annotations
highlight_dates <- as.Date(c("2020-04-11", "2021-01-20"))
highlight_values <- merged_data %>%
  group_by(date) %>%
  summarise(national_rolling_avg = sum(rolling_avg, na.rm = TRUE)) %>%
  filter(date %in% highlight_dates)
```

```r
# Improved visualization with per capita adjustment
ggplot(merged_data, aes(x = date, y = rolling_avg, fill = area_name)) +
  geom_area(alpha = 0.7, na.rm = TRUE) +
  scale_fill_viridis_d(option = "viridis") +

  facet_wrap(~area_name, scales = "fixed") +

  labs(
    title = "COVID-19 Deaths in England by Region (Per Capita Adjustment)",
    subtitle = "7-day rolling average deaths per 100,000 people",
    x = "Year",
    y = "Deaths per 100,000 people",
    fill = "Region"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 14, face = "bold"),
    legend.position = "none",
    panel.spacing = unit(1.5, "lines"),
    axis.title.y = element_text(size = 12, face = "bold")
  ) +

  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(axis.text.x = element_text(size = 12, face = "bold")) +
  theme(
    legend.position = "right",
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12)
  ) +

  geom_vline(aes(xintercept = as.Date("2020-04-11"),
                 linetype = "April 11, 2020: 865 deaths per day"),
             color = "black", size = 1) +
  geom_vline(aes(xintercept = as.Date("2021-01-20"),
                 linetype = "Jan 20, 2021: 1,160 deaths per day"),
             color = "red", size = 1) +

  scale_linetype_manual(values = c("April 11, 2020: 865 deaths per day" = "dashed",
                                   "Jan 20, 2021: 1,160 deaths per day" = "dotted")) +

  guides(linetype = guide_legend(title = "Key Events"))
```



COVID−19 Deaths in England by Region (Per Capita Adjustment)

7−day rolling average deaths per 100,000 people

This visualization adjusts for population size, making comparisons across regions fairer. The per capita approach reveals that some regions were disproportionately affected by COVID-19 deaths.