code

```r
library(readxl)
library(stringr)
library(lubridate)
```

```

Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```r
library(ggplot2)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr   1.1.4     v readr   2.1.5
v forcats 1.0.0     v tibble  3.2.1
v purrr   1.0.2     v tidyr   1.3.1

-- 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 becom
```

```r
library(patchwork)
library(httr2)
library(gridExtra)
```

```
Attaching package: 'gridExtra'

The following object is masked from 'package:dplyr':

    combine
```

Data Cleaning

First, from reading https://data.cdc.gov/ we learn the endpoint `https://data.cdc.gov/resource/pwn4-m3yp.` provides state level data from COVID-19 cases.

```
api <- "https://data.cdc.gov/resource/pwn4-m3yp.json"
cases_raw <-  request(api) |>
  req_url_path_append("?$limit=10000000000") |>
  req_perform() |>
  resp_body_json(simplifyVector = TRUE) |>
  as_tibble()
```

Then wrangle the resulting data frame to produce a data frame with columns `state`, `date` (should be the end date), `year`, `MMWR week`, and `cases`. Make sure the cases are numeric and the dates are in `Date` ISO-8601 format. Finally, we need to filter the data to ensure it only includes the state level data and DC, PR.

```
cases <- cases_raw |>
  select(state, date = end_date, cases = new_cases) |>
  mutate(date = as.Date(date, format = "%Y-%m-%d"),
         cases = as.numeric(cases),
         mmwr_week = epiweek(date),
         year = year(date)) |>
  filter(state %in% state.abb | state %in% c("PR", "DC"))
```

Second, from reading https://data.cdc.gov/, we can also download the deaths data and save the data frames. Also wrangle the resulting data frame as above, to maintain consistency with cases data, change `state` to be abbreviated and round the date down to the nearest week, use the day of cases data showed as the first day. Finally, we need to filter the data to ensure it only includes the state level data and DC, PR.

```
api <- "https://data.cdc.gov/resource/r8kw-7aab.json"
deaths_raw <-  request(api) |>
  req_url_path_append("?$limit=10000000000") |>
  req_perform() |>
  resp_body_json(simplifyVector = TRUE) |>
  as_tibble()

deaths <- deaths_raw |>
  select(state_name = state, date = end_date, deaths = covid_19_deaths, mmwr_week) |>
  mutate(state = state.abb[match(state_name, state.name)]) |>
  mutate(state = case_when(state_name == "Puerto Rico" ~ "PR",
                           state_name == "District of Columbia" ~ "DC",
                           TRUE ~ state)) |>
  na.omit() |>
  mutate(date = as.Date(date, format = "%Y-%m-%d"),
         deaths = as.numeric(deaths),
         mmwr_week = parse_number(mmwr_week),
         year = year(date)) |>
  select(state, mmwr_week, deaths, year)
```

Then, combine these two data according to their state and date.

```
cases_deaths <- left_join(cases, deaths, by = c("state", "year", "mmwr_week")) |>
   na.omit()
```

Besides, we use the data from the United States Census Bureau to collect the population totals for each state from years 2020 to 2023. This data includes the Resident Population for the United States based on 2020-04-01, 2020-07-01, 2021-07-01, 2022-07-01, 2023-07-01, and we select the data based on 2020-07-01, 2021-07-01, 2022-07-01, 2023-07-01. And this data also includes the region-level data, we need to remove it.

```
population_raw <- read_excel("~/Desktop/BST260/BST260-Final-Project/data/raw-data/NST-EST2023
  slice(-c(1:8)) |> # Remove the region-level data
  select(-2) # Remove the data based on 2020-04-01

colnames(population_raw) <- c("state_name", "2020", "2021", "2022", "2023") # Rename each col
```

Then wrangle the resulting data frame to produce a data frame with columns `state`, `year` and `population`. Make sure the population is numeric. And the state should be abbreviated.

```
population <- population_raw |>
  mutate(state_name = str_remove(state_name, "^\\.")) |>
  mutate(across(-state_name, as.numeric)) |>
  pivot_longer(-state_name, names_to = "year", values_to = "population") |>
  mutate(state = state.abb[match(state_name, state.name)]) |>
  mutate(state = case_when(state_name == "Puerto Rico" ~ "PR",
                           state_name == "District of Columbia" ~ "DC",
                           TRUE ~ state)) |>
  select(-state_name) |>
  mutate(across(-state, as.numeric))
```

Finally, we need to combine these data according to their state and year.

```
data <- left_join(cases_deaths, population, by = c("state", "year"))
```

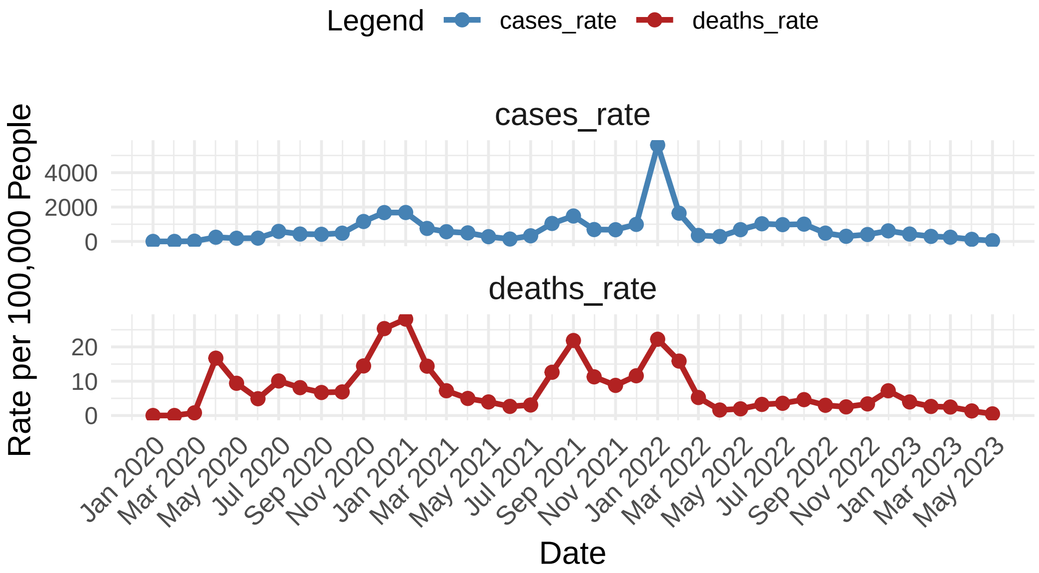
Task 1

I plan to divide the pandemic period into waves by the national level data. So I need to plot a trend plot for cases and deaths in the United States. Plot rates per 100,000 people.

```
# Calculate the total population in the United States for each year
population_country <- population |>
  group_by(year) |>
  summarise(population = sum(population),
            .groups = "drop")

# Calculate the total cases and deaths in the United States for each month
data_monthly <- data |>
  mutate(month = floor_date(date, "month")) |>
  group_by(month) |>
  summarise(cases = sum(cases),
            deaths = sum(deaths),
            year = unique(year),
            .groups = "drop") |>
  left_join(population_country, by = "year") |>
  mutate(cases_rate = (cases / population) * 100000,
         deaths_rate = (deaths / population) * 100000)

# Plot
figs_1 <- data_monthly |>
  pivot_longer(cols = c(cases_rate, deaths_rate),
               names_to = "two_data", values_to = "rates") |>
  filter(!is.na(rates)) |>
  ggplot(aes(x = month, y = rates, color = two_data)) +
  geom_line(linewidth = 1) +
```

```r
  geom_point(size = 2) +
  facet_wrap(~two_data, scales = "free_y", ncol = 1) +
  labs(
    title = "Figure 1: Trend Plot of Cases, and Deaths",
    x = "Date",
    y = "Rate per 100,000 People",
    color = "Legend") +
  scale_x_date(date_labels = "%b %Y", date_breaks = "2 months") +
  scale_color_manual(values = c("cases_rate" = "steelblue", "deaths_rate" = "firebrick")) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
    axis.title = element_text(size = 12),
    strip.text = element_text(size = 12),
    legend.position = "top"
  )
figs_1
```

Figure 1: Trend Plot of Cases, and Deaths



According to the graph, I divide the pandemic period into 4 waves, January 2020 to September 2020, October 2020 to June 2021, July 2021 to April 2022, May 2022 to May 2023.

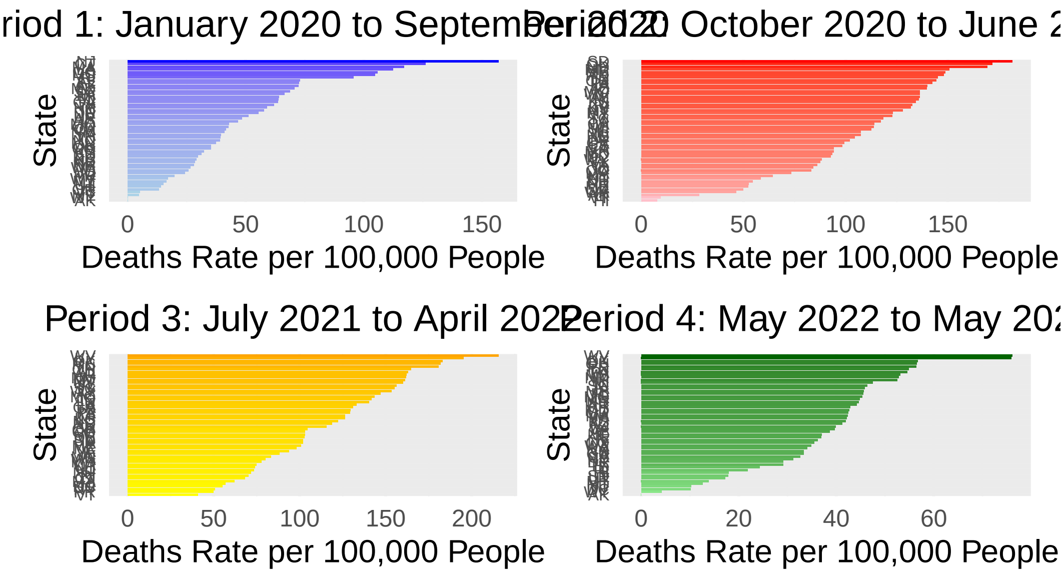
Task 2

For each period compute the deaths rates per 100,000 people by state.

```r
# Calculate the deaths rates by state in the first period, January 2020 to September 2020
deaths_rate_period1 <- data |>
  filter(date <= as.Date("2020-09-30")) |>
  group_by(state) |>
  summarise(deaths = sum(deaths),
            population = mean(population),
            .groups = "drop") |>
  mutate(deaths_rate_period1 = deaths / population * 100000)

# Calculate the deaths rates by state in the second period, October 2020 to June 2021
deaths_rate_period2 <- data |>
  filter(date >= as.Date("2020-10-01") & date <= as.Date("2021-06-30")) |>
  group_by(state) |>
  summarise(deaths = sum(deaths),
            population = mean(population),
            .groups = "drop") |>
  mutate(deaths_rate_period2 = deaths / population * 100000)

# Calculate the deaths rates by state in the third period, July 2021 to April 2022
deaths_rate_period3 <- data |>
  filter(date >= as.Date("2021-07-01") & date <= as.Date("2022-04-30")) |>
  group_by(state) |>
  summarise(deaths = sum(deaths),
            population = mean(population),
            .groups = "drop") |>
  mutate(deaths_rate_period3 = deaths / population * 100000)

# Calculate the deaths rates by state in the fourth period, May 2022 to May 2023
deaths_rate_period4 <- data |>
  filter(date >= as.Date("2022-05-01")) |>
  group_by(state) |>
  summarise(deaths = sum(deaths),
            population = mean(population),
            .groups = "drop") |>
  mutate(deaths_rate_period4 = deaths / population * 100000)

# Combine all death rates in different periods to one dataset
deaths_rate <- left_join(deaths_rate_period1, deaths_rate_period2, by = "state") |>
  left_join(deaths_rate_period3, by = "state") |>
  left_join(deaths_rate_period4, by = "state") |>
  select(state, deaths_rate_period1, deaths_rate_period2,
         deaths_rate_period3, deaths_rate_period4)

# Plot for period 1
figs_2_1 <- deaths_rate |>
  mutate(state = reorder(state, deaths_rate_period1)) |>
  ggplot(aes(x = deaths_rate_period1, y = state, fill = deaths_rate_period1)) +
  geom_col(show.legend = FALSE) +
  scale_fill_gradient(low = "lightblue", high = "blue") +
  labs(
    title = "Period 1: January 2020 to September 2020",
    x = "Deaths Rate per 100,000 People",
    y = "State"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_text(size = 5.5),
    axis.title = element_text(size = 12),
    plot.title = element_text(hjust = 0.5, size = 14))

# Plot for period 2
figs_2_2 <- deaths_rate |>
  mutate(state = reorder(state, deaths_rate_period2)) |>
  ggplot(aes(x = deaths_rate_period2, y = state, fill = deaths_rate_period2)) +
  geom_col(show.legend = FALSE) +
  scale_fill_gradient(low = "pink", high = "red") +
  labs(
    title = "Period 2: October 2020 to June 2021",
    x = "Deaths Rate per 100,000 People",
    y = "State"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_text(size = 5.5),
    axis.title = element_text(size = 12),
    plot.title = element_text(hjust = 0.5, size = 14))

# Plot for period 3
figs_2_3 <- deaths_rate |>
  mutate(state = reorder(state, deaths_rate_period3)) |>
  ggplot(aes(x = deaths_rate_period3, y = state, fill = deaths_rate_period3)) +
```

```r
  geom_col(show.legend = FALSE) +
  scale_fill_gradient(low = "yellow", high = "orange") +
  labs(
    title = "Period 3: July 2021 to April 2022",
    x = "Deaths Rate per 100,000 People",
    y = "State"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_text(size = 5.5),
    axis.title = element_text(size = 12),
    plot.title = element_text(hjust = 0.5, size = 14))

# Plot for period 4
figs_2_4 <- deaths_rate |>
  mutate(state = reorder(state, deaths_rate_period4)) |>
  ggplot(aes(x = deaths_rate_period4, y = state, fill = deaths_rate_period4)) +
  geom_col(show.legend = FALSE) +
  scale_fill_gradient(low = "lightgreen", high = "darkgreen") +
  labs(
    title = "Period 4: May 2022 to May 2023",
    x = "Deaths Rate per 100,000 People",
    y = "State"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_text(size = 5.5),
    axis.title = element_text(size = 12),
    plot.title = element_text(hjust = 0.5, size = 14))

# Combine four plots
(figs_2_1 | figs_2_2) / (figs_2_3 | figs_2_4) +
  plot_annotation(
    title = "Figure 2: Deaths Rates by State for Each Period",
    theme = theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5)))
```

Figure 2: Deaths Rates by State for Each Period



Task 3

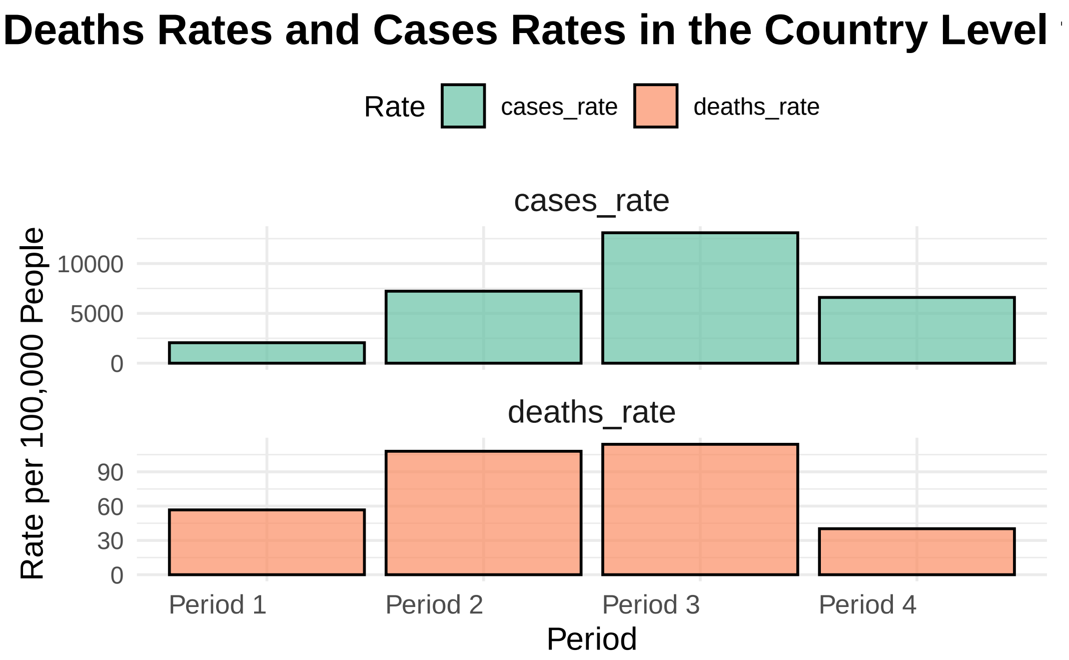
Calculate the deaths rate and cases rate per 100,000 people in the country level to determine if COVID-19 became less or more virulent across the different periods.

```
# First calculate the population for each period, some period spans two years,
# so the population should be calculated by the average of two years.
population_period_1 <- population_country[1,2]
population_period_2 <- (population_country[1,2] + population_country[2,2])/2
population_period_3 <- (population_country[2,2] + population_country[3,2])/2
population_period_4 <- (population_country[3,2] + population_country[4,2])/2

# Calculate the deaths and cases rates in the country level in the first period, January 202
rate_country_period1 <- data |>
  filter(date <= as.Date("2020-09-30")) |>
  summarise(deaths = sum(deaths),
            cases = sum(cases),
            deaths_rate = deaths / population_period_1 * 100000,
            cases_rate = cases / population_period_1 * 100000)

# Calculate the deaths and cases rates in the country level in the second period, October 20
rate_country_period2 <- data |>
```

```r
  filter(date >= as.Date("2020-10-01") & date <= as.Date("2021-06-30")) |>
  summarise(deaths = sum(deaths),
            cases = sum(cases),
            deaths_rate = deaths / population_period_2 * 100000,
            cases_rate = cases / population_period_2 * 100000)

# Calculate the deaths and cases rates in the country level in the third period, July 2021 to
rate_country_period3 <- data |>
  filter(date >= as.Date("2021-07-01") & date <= as.Date("2022-04-30")) |>
  summarise(deaths = sum(deaths),
            cases = sum(cases),
            deaths_rate = deaths / population_period_3 * 100000,
            cases_rate = cases / population_period_3 * 100000)

# Calculate the deaths and cases rates in the country level in the fourth period, May 2022 to
rate_country_period4 <- data |>
  filter(date >= as.Date("2022-05-01")) |>
  summarise(deaths = sum(deaths),
            cases = sum(cases),
            deaths_rate = deaths / population_period_4 * 100000,
            cases_rate = cases / population_period_4 * 100000)

# Combine all periods data into one dataset
rate_country <- rbind(rate_country_period1, rate_country_period2,
                      rate_country_period3, rate_country_period4)
rate_country$period <- c("Period 1", "Period 2", "Period 3", "Period 4")
rate_country$deaths_rate <- rate_country$deaths_rate$population
rate_country$cases_rate <- rate_country$cases_rate$population
```

Then plot the rates for each period.

```r
# Plot
figs_3 <- rate_country |>
  pivot_longer(cols = c(cases_rate, deaths_rate),
               names_to = "two_data", values_to = "rates") |>
  ggplot(aes(x = period, y = rates, fill = two_data)) +
  geom_col(position = "dodge", color = "black", alpha = 0.7) +
  facet_wrap(~two_data, scales = "free_y", ncol = 1) +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    axis.text.x = element_text(hjust = 1, size = 10),
    axis.title = element_text(size = 12),
    strip.text = element_text(size = 12),
    legend.position = "top") +
  labs(
    title = "Figure 3: Deaths Rates and Cases Rates in the Country Level for Each Period",
    x = "Period",
    y = "Rate per 100,000 People",
    fill = "Rate")
figs_3
```



Supplementary Methods

Show the box plot to see the death rates distribution for each state by period.

```
figs_4 <- deaths_rate |>
  pivot_longer(cols = starts_with("deaths_rate_period"),
               names_to = "period",
               values_to = "deaths_rate") |>
  mutate(period = recode(period,
                         "deaths_rate_period1" = "Period 1",
                         "deaths_rate_period2" = "Period 2",
                         "deaths_rate_period3" = "Period 3",
                         "deaths_rate_period4" = "Period 4")) |>
  ggplot(aes(x = period, y = deaths_rate, fill = period)) +
  geom_boxplot(outlier.color = "red", outlier.size = 2) +
  labs(title = "Figure 4: Death Rates Distribution by Period",
       x = "Period",
       y = "Death Rate per 100,000 People") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    axis.text.x = element_text(size = 12),
    axis.title = element_text(size = 14))

figs_4
```

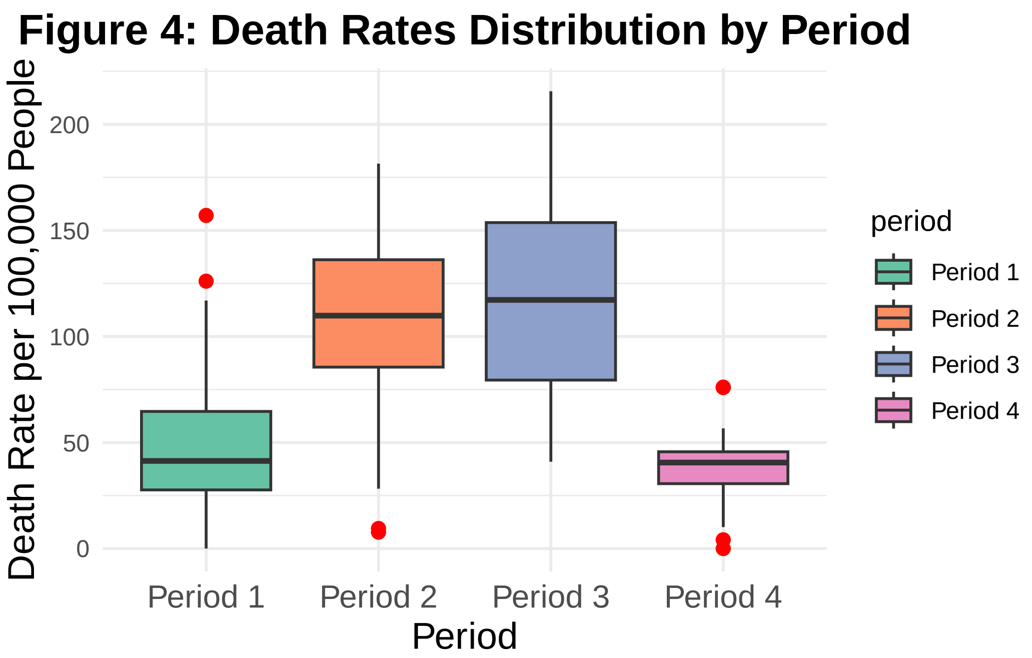


Figure 4: Death Rates Distribution by Period
Death Rate per 100,000 People
Period 1
Period 2
Period 3
Period 4
Period
period
Period 1
Period 2
Period 3
Period 4