Covid Analysis and Models

Cody S

```
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 stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- 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
```

Goal

My main goal will be to see if I can get a general sense of how the number of deaths relative to the number of cases changed as vaccines were developed and rolled out.

Import Data

```
# set urls

global_cases_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/mas

global_deaths_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/ma

US_cases_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/

US_death_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/
```

```
#import to dataframes

global_cases <- read.csv(global_cases_url)
global_deaths <- read.csv(global_deaths_url)
US_cases <- read.csv(US_cases_url)
US_deaths <- read.csv(US_death_url)

all_data <- list(global_cases, global_deaths, US_cases, US_deaths)
```

```
lapply(all_data, head)
```

Start cleaning and organizing

The data is broken down into pretty small regions, and the regions are the rows and the dates are the columns. My main goal will be consolidated all of the global data into a single time series of cases and deaths, and all of the USA data in the same way for comparison. This will mostly involve dropping the geographic breakdown and pivoting the data so that the dates are the rows.

```
geo_rename <- function(df) {
  df <- df %>%
    rename(Province.State = Province_State,
           Country.Region = Country_Region) %>%
    select(-Admin2, -UID, -FIPS, -code3, -Combined_Key, -starts_with("iso"))

  return(df)
}

US_cases_renamed <- geo_rename(US_cases)
US_deaths_renamed <- geo_rename(US_deaths)
```

```
# drop some columns and pivot to dates
pivot_organize <- function(df) {
  df <- df %>%
    select(-Lat, -contains("Long"), -contains("Population")) %>%
    pivot_longer(
        cols = -c(Province.State, Country.Region),
        names_to = "Date",
        values_to = "Cases"
    ) %>%
    select(Date, Country.Region, Province.State, Cases
    ) %>%
    mutate(Date = sub("^X", "", Date)
    ) %>%
     mutate(Date = as.Date(Date,format = "%m.%d.%y"))

  return(df)
}

# apply the pivot and reorganization to all data sets
global_cases_clean <- pivot_organize(global_cases)
global_deaths_clean <- pivot_organize(global_deaths)
us_cases_clean <- pivot_organize(US_cases_renamed)
us_deaths_clean <- pivot_organize(US_deaths_renamed)

# make sure to keep cases and deaths straight
global_deaths_clean <- global_deaths_clean %>%
  rename(Deaths = Cases)
us_deaths_clean <- us_deaths_clean %>%
  rename(Deaths = Cases)

all_data_clean <- list(global_cases_clean, global_deaths_clean, us_cases_clean, us_deaths_cle
```

Now we need to consolidate all of the sub-region data into overal totals. The next few steps will be complicated so I'm going to do it with global data first.

```
# make a simple global daily cases df with a weekly column as well
global_daily_cases <- global_cases_clean %>%
    group_by(Date) %>%
    summarise(Total_cases = sum(Cases)) %>%
    mutate(New_daily_cases = Total_cases - lag(Total_cases, n = 1, default = NA)) %>%
    mutate(Week = floor_date(Date, unit = "week", week_start = 1))

global_daily_deaths <- global_deaths_clean %>%
    group_by(Date) %>%
    summarise(Total_deaths = sum(Deaths)) %>%
    mutate(New_daily_deaths = Total_deaths - lag(Total_deaths, n = 1, default = NA)) %>%
    mutate(Week = floor_date(Date, unit = "week", week_start = 1))

tail(global_daily_cases)
```

```
# A tibble: 6 x 4
```

```
  Date       Total_cases New_daily_cases Week
  <date>           <int>           <int> <date>
1 2023-03-04   675968775           54195 2023-02-27
2 2023-03-05   676024901           56126 2023-02-27
3 2023-03-06   676082941           58040 2023-03-06
4 2023-03-07   676213378          130437 2023-03-06
5 2023-03-08   676392824          179446 2023-03-06
6 2023-03-09   676570149          177325 2023-03-06
```

```
tail(global_daily_deaths)
```

```
# A tibble: 6 x 4
  Date       Total_deaths New_daily_deaths Week
  <date>            <int>            <int> <date>
1 2023-03-04      6877601              276 2023-02-27
2 2023-03-05      6877749              148 2023-02-27
3 2023-03-06      6878115              366 2023-03-06
4 2023-03-07      6879038              923 2023-03-06
5 2023-03-08      6880483             1445 2023-03-06
6 2023-03-09      6881802             1319 2023-03-06
```

```
# combine daily deaths and cases
daily_global_combo <- left_join(global_daily_cases, global_daily_deaths, by = "Date")

daily_global_combo <- daily_global_combo %>%
    select(-Week.x) %>%
    rename(Week = Week.y)

# create a weekly global death/cases df
weekly_global_combo <- daily_global_combo %>%
    group_by(Week) %>%
    summarise(
        g_Weekly_total_cases = last(Total_cases),
        g_Weekly_total_deaths = last(Total_deaths),
        g_Weekly_new_cases = sum(New_daily_cases, na.rm = TRUE),
        g_Weekly_new_deaths = sum(New_daily_deaths, na.rm = TRUE)
)

tail(daily_global_combo)
```

```
# A tibble: 6 x 6
```

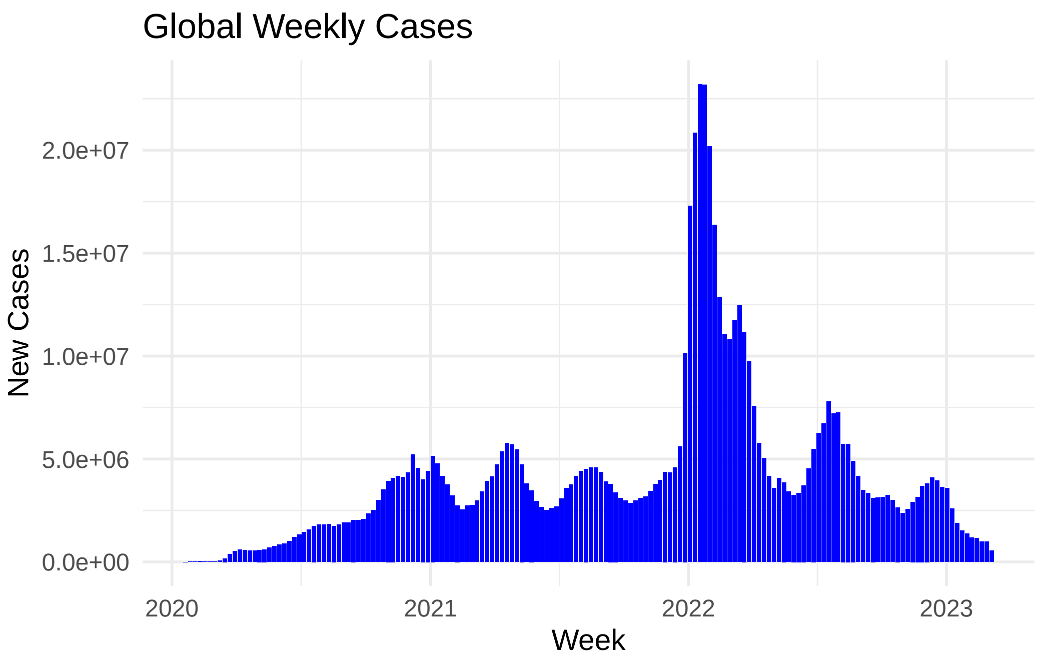
```
  Date       Total_cases New_daily_cases Total_deaths New_daily_deaths
  <date>           <int>           <int>        <int>            <int>
1 2023-03-04   675968775           54195      6877601              276
2 2023-03-05   676024901           56126      6877749              148
3 2023-03-06   676082941           58040      6878115              366
4 2023-03-07   676213378          130437      6879038              923
5 2023-03-08   676392824          179446      6880483             1445
6 2023-03-09   676570149          177325      6881802             1319
# i 1 more variable: Week <date>
```

```
tail(weekly_global_combo)
```

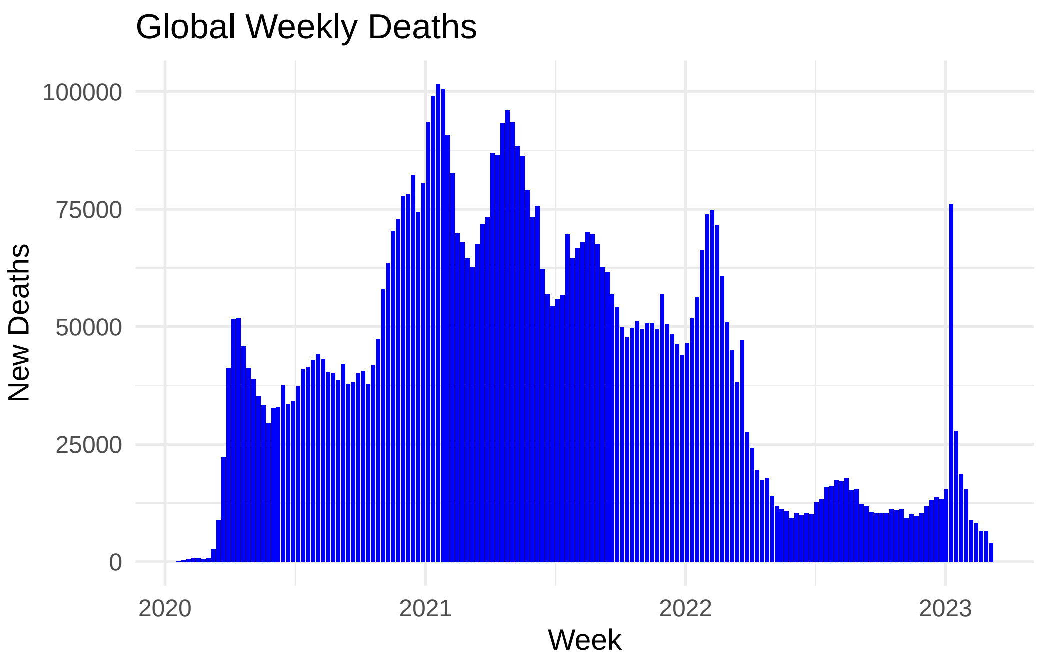
```
# A tibble: 6 x 5
  Week       g_Weekly_total_cases g_Weekly_total_deaths g_Weekly_new_cases
  <date>                    <int>                 <int>              <int>
1 2023-01-30            671721476               6847646            1377422
2 2023-02-06            672906177               6856419            1184701
3 2023-02-13            674056229               6864711            1150052
4 2023-02-20            675044414               6871268             988185
5 2023-02-27            676024901               6877749             980487
6 2023-03-06            676570149               6881802             545248
# i 1 more variable: g_Weekly_new_deaths <int>
```

```
# start plotting soon for testing, so this will set global fig dimensions
options(repr.plot.width = 10, repr.plot.height = 6)
```

```
ggplot(data = weekly_global_combo, aes(x = Week, y = g_Weekly_new_cases)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "Global Weekly Cases",
    x = "Week",
    y = "New Cases"
  ) +
  theme_minimal()
```



```r
ggplot(data = weekly_global_combo, aes(x = Week, y = g_Weekly_new_deaths)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "Global Weekly Deaths",
    x = "Week",
    y = "New Deaths"
  ) +
  theme_minimal()
```

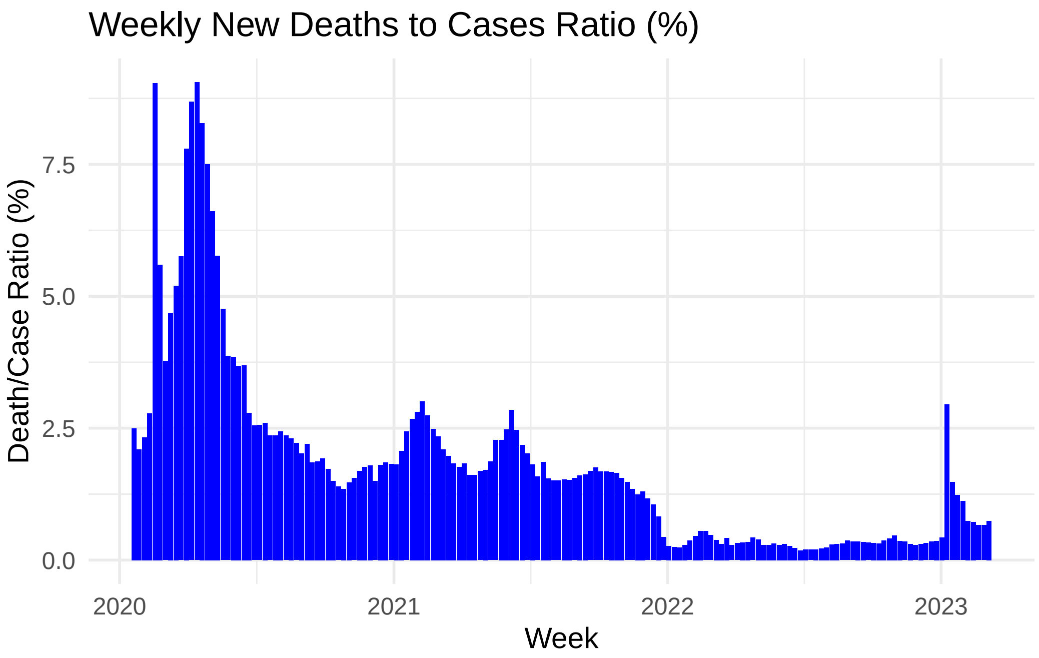


That looks good, and I can see some general trends in deaths over time, so I think I can move forward with calculated the ratio of deaths to cases, which will also allow me to compare it on the same scale to the US data later.

```
weekly_global_combo <- weekly_global_combo %>%
    mutate(g_Weekly_death_to_case_ratio = (g_Weekly_new_deaths / g_Weekly_new_cases)* 100)

tail(weekly_global_combo)
```

```
# A tibble: 6 x 6
  Week       g_Weekly_total_cases g_Weekly_total_deaths g_Weekly_new_cases
  <date>                    <int>                 <int>              <int>
1 2023-01-30            671721476               6847646            1377422
2 2023-02-06            672906177               6856419            1184701
3 2023-02-13            674056229               6864711            1150052
4 2023-02-20            675044414               6871268             988185
5 2023-02-27            676024901               6877749             980487
6 2023-03-06            676570149               6881802             545248
# i 2 more variables: g_Weekly_new_deaths <int>,
#   g_Weekly_death_to_case_ratio <dbl>
```

```r
ggplot(data = weekly_global_combo, aes(x = Week, y = g_Weekly_death_to_case_ratio)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "Weekly New Deaths to Cases Ratio (%)",
    x = "Week",
    y = "Death/Case Ratio (%)"
  ) +
  theme_minimal()
```



I'm using pretty general vaccine availability dates and using as somewhat arbitrary one year span as the "rollout" duration. The post- to pre-vaccine comparison will be the most important in the end.

```r
# set phase dates
vac_rollout_start <- as.Date("2020-12-01")
vac_rollout_end <- as.Date("2021-12-01")

# add vaccine phase code
weekly_global_combo <- weekly_global_combo %>%
    mutate(Phase = case_when(
            Week < vac_rollout_start ~ "1. Pre",
            Week >= vac_rollout_start & Week < vac_rollout_end ~ "2. Rollout",
            Week >= vac_rollout_end ~ "3. Post"
    ))

tail(weekly_global_combo)
```

```
# A tibble: 6 x 7
  Week       g_Weekly_total_cases g_Weekly_total_deaths g_Weekly_new_cases
  <date>                    <int>                 <int>              <int>
1 2023-01-30            671721476               6847646            1377422
2 2023-02-06            672906177               6856419            1184701
3 2023-02-13            674056229               6864711            1150052
4 2023-02-20            675044414               6871268             988185
5 2023-02-27            676024901               6877749             980487
6 2023-03-06            676570149               6881802             545248
# i 3 more variables: g_Weekly_new_deaths <int>,
#   g_Weekly_death_to_case_ratio <dbl>, Phase <chr>
```

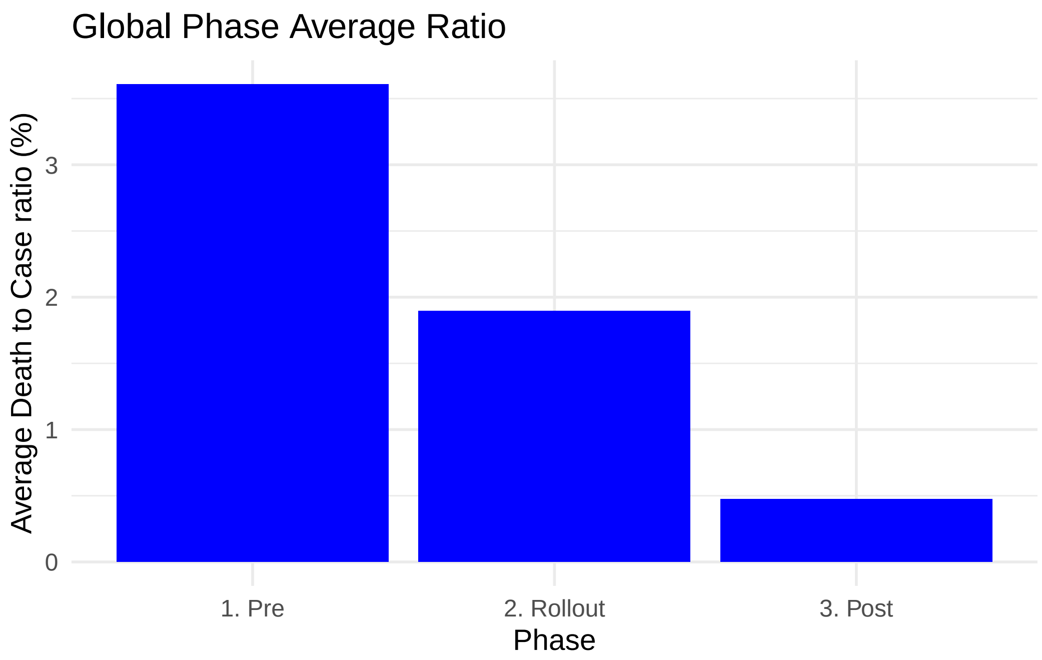
That looks good, so now I'll just calculate some rudimentary death/case ratio averages for each phase.

```
global_phase_means <- weekly_global_combo %>%
    group_by(Phase) %>%
    summarise(global_phase_avg = mean(g_Weekly_death_to_case_ratio, na.rm = TRUE))

global_phase_means
```

```
# A tibble: 3 x 2
  Phase      global_phase_avg
  <chr>                 <dbl>
1 1. Pre                3.61
2 2. Rollout            1.90
3 3. Post               0.474
```

```
global_phase_means_plot <- ggplot(data = global_phase_means, aes(x = Phase, y = global_phase_
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "Global Phase Average Ratio",
    x = "Phase",
    y = "Average Death to Case ratio (%)"
  ) +
  theme_minimal()
```

```r
global_phase_means_plot
```



Model development

The visuals and simple checks all indicate that the ratio of deaths to cases were much lower after the vaccine rollout. Now I'll model that with a multi-part (or interrupted) linear analysis for each phase.

Test model with global data

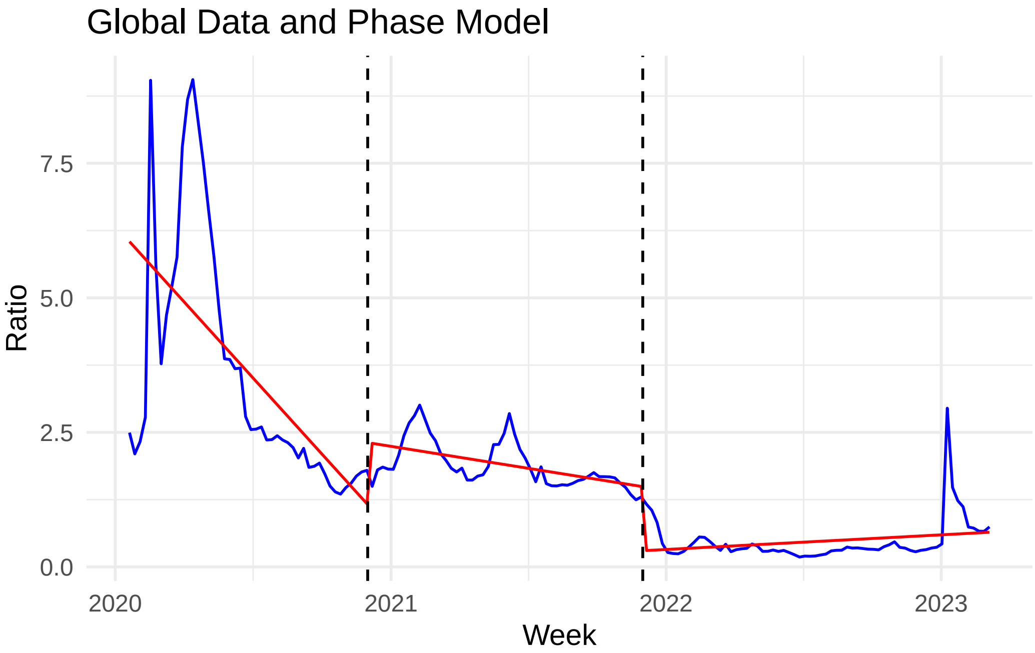
```r
# do a simple phased linear regression
global_phase_model <- lm(g_Weekly_death_to_case_ratio ~ Week * factor(Phase), data = weekly_g

summary(global_phase_model)
```

```
Call:
lm(formula = g_Weekly_death_to_case_ratio ~ Week * factor(Phase),
    data = weekly_global_combo)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-3.8365 -0.2591 -0.1333  0.1012  4.3100 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  2.888e+02  2.932e+01   9.850  < 2e-16 ***
Week                        -1.546e-02  1.590e-03  -9.727  < 2e-16 ***
factor(Phase)2. Rollout     -2.447e+02  3.843e+01  -6.369 1.98e-09 ***
factor(Phase)3. Post        -3.026e+02  3.427e+01  -8.829 1.90e-15 ***
Week:factor(Phase)2. Rollout  1.322e-02  2.068e-03   6.393 1.75e-09 ***
Week:factor(Phase)3. Post     1.621e-02  1.839e-03   8.813 2.09e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.002 on 158 degrees of freedom
Multiple R-squared:  0.6974,    Adjusted R-squared:  0.6878 
F-statistic: 72.81 on 5 and 158 DF,  p-value: < 2.2e-16
```

```r
weekly_global_combo$g_Modeled <- predict(global_phase_model)
```

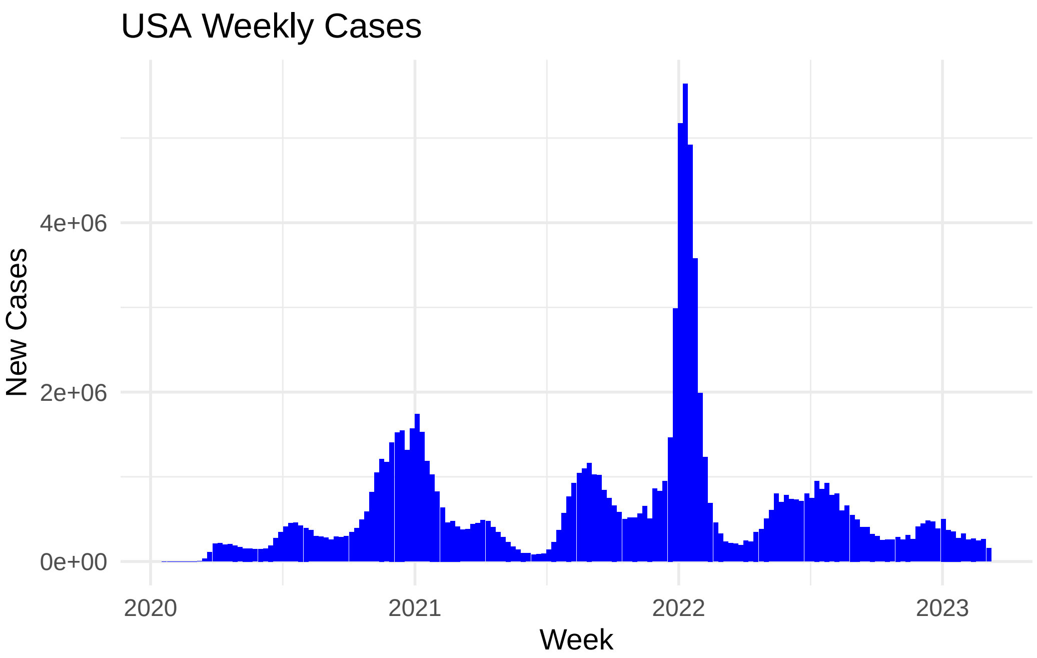
```r
global_ratio_data_model <- ggplot(weekly_global_combo, aes(x = Week)) +
  geom_line(aes(y = g_Weekly_death_to_case_ratio), color = "blue") +
  geom_line(aes(y = g_Modeled), color = "red") +
  labs(title = "Global Data and Phase Model",
       x = "Week", y = "Ratio") +
  geom_vline(xintercept = vac_rollout_start, linetype = "dashed", color = "black") +
  geom_vline(xintercept = vac_rollout_end, linetype = "dashed", color = "black") +
  theme_minimal()

global_ratio_data_model
```



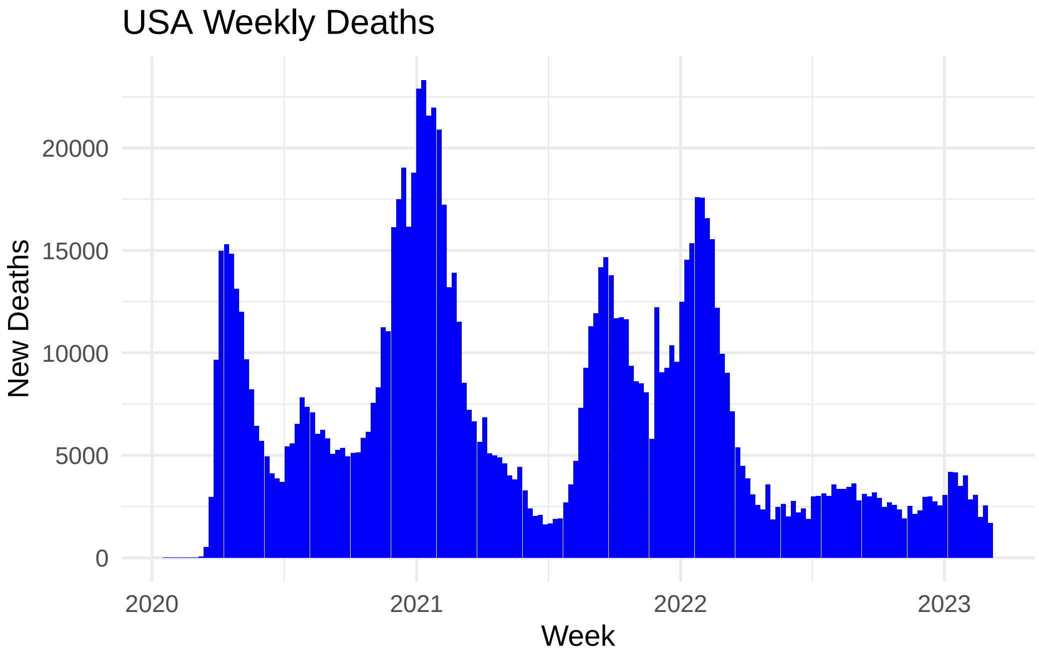
US Data clean and model prep

Now I want to do the same thing for the USA data, then I'll replot everything at the end

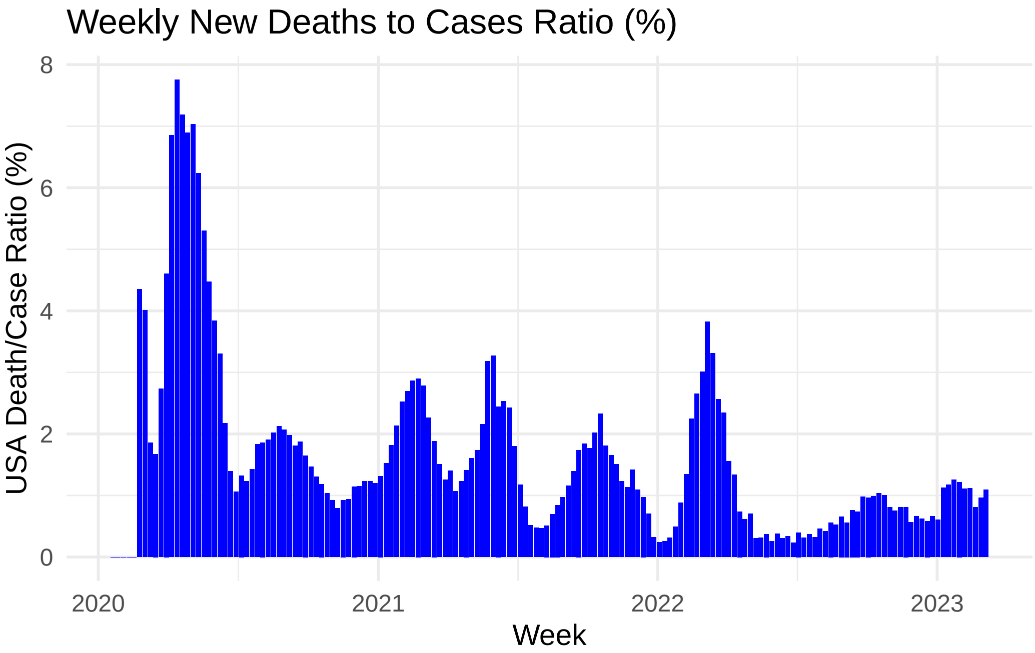
```
# make a simple us daily cases df with a weekly column as well
us_daily_cases <- us_cases_clean %>%
    group_by(Date) %>%
    summarise(Total_cases = sum(Cases)) %>%
    mutate(New_daily_cases = Total_cases - lag(Total_cases, n = 1, default = NA)) %>%
    mutate(Week = floor_date(Date, unit = "week", week_start = 1))

us_daily_deaths <- us_deaths_clean %>%
    group_by(Date) %>%
    summarise(Total_deaths = sum(Deaths)) %>%
    mutate(New_daily_deaths = Total_deaths - lag(Total_deaths, n = 1, default = NA)) %>%
    mutate(Week = floor_date(Date, unit = "week", week_start = 1))

# combine daily deaths and cases
daily_us_combo <- left_join(us_daily_cases, us_daily_deaths, by = "Date")

daily_us_combo <- daily_us_combo %>%
    select(-Week.x) %>%
    rename(Week = Week.y)

# create a weekly US death/cases df
weekly_us_combo <- daily_us_combo %>%
    group_by(Week) %>%
    summarise(
        us_Weekly_total_cases = last(Total_cases),
        us_Weekly_total_deaths = last(Total_deaths),
        us_Weekly_new_cases = sum(New_daily_cases, na.rm = TRUE),
        us_Weekly_new_deaths = sum(New_daily_deaths, na.rm = TRUE)
)


ggplot(data = weekly_us_combo, aes(x = Week, y = us_Weekly_new_cases)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "USA Weekly Cases",
    x = "Week",
    y = "New Cases"
  ) +
  theme_minimal()
```



```r
ggplot(data = weekly_us_combo, aes(x = Week, y = us_Weekly_new_deaths)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "USA Weekly Deaths",
    x = "Week",
    y = "New Deaths"
  ) +
  theme_minimal()
```



```r
weekly_us_combo <- weekly_us_combo %>%
    mutate(us_Weekly_death_to_case_ratio = (us_Weekly_new_deaths / us_Weekly_new_cases)* 100

ggplot(data = weekly_us_combo, aes(x = Week, y = us_Weekly_death_to_case_ratio)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "Weekly New Deaths to Cases Ratio (%)",
    x = "Week",
    y = "USA Death/Case Ratio (%)"
  ) +
  theme_minimal()
```



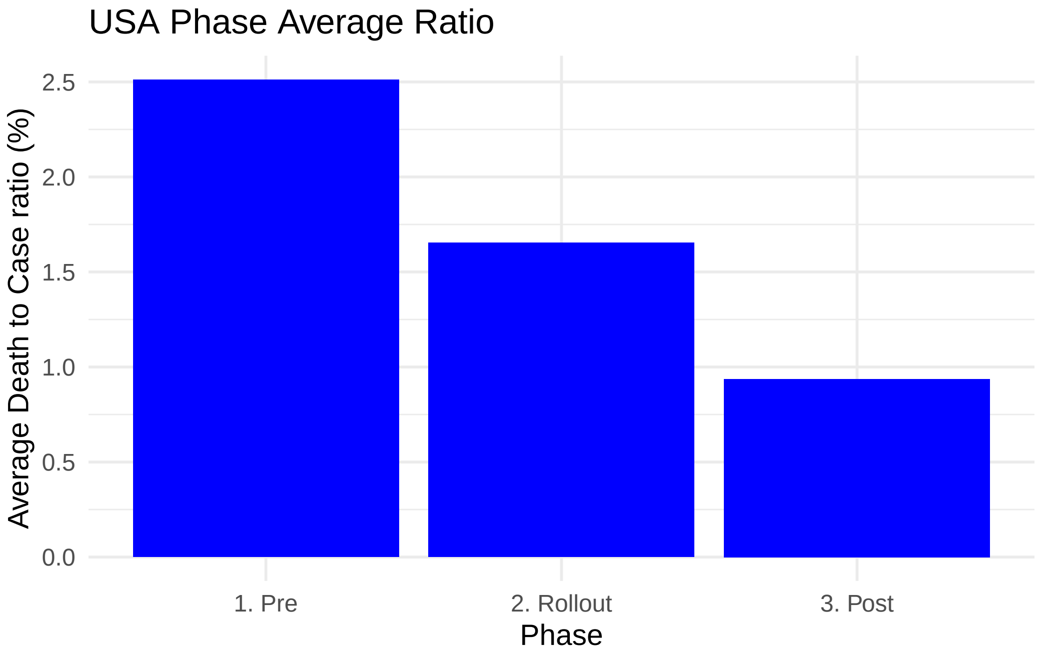
```
weekly_us_combo <- weekly_us_combo %>%
    mutate(Phase = case_when(
            Week < vac_rollout_start ~ "1. Pre",
            Week >= vac_rollout_start & Week < vac_rollout_end ~ "2. Rollout",
            Week >= vac_rollout_end ~ "3. Post"
    ))

# check that the phase and ratio calcs all worked correctly
tail(weekly_us_combo)
```

```
# A tibble: 6 x 7
  Week       us_Weekly_total_cases us_Weekly_total_deaths us_Weekly_new_cases
  <date>                     <int>                  <int>               <int>
1 2023-01-30             102607114                1111698              328687
2 2023-02-06             102862879                1114537              255765
3 2023-02-13             103136077                1117590              273198
4 2023-02-20             103382763                1119588              246686
5 2023-02-27             103646975                1122134              264212
6 2023-03-06             103802702                1123836              155727
# i 3 more variables: us_Weekly_new_deaths <int>,
#   us_Weekly_death_to_case_ratio <dbl>, Phase <chr>
```

```r
# calculate phase means
us_phase_means <- weekly_us_combo %>%
    group_by(Phase) %>%
    summarise(us_phase_avg = mean(us_Weekly_death_to_case_ratio, na.rm = TRUE))

# display the phase means
us_phase_means
```

```
# A tibble: 3 x 2
  Phase      us_phase_avg
  <chr>             <dbl>
1 1. Pre            2.51
2 2. Rollout        1.65
3 3. Post           0.937
```

```r
# plot phase means
us_phase_means_plot <- ggplot(data = us_phase_means, aes(x = Phase, y = us_phase_avg)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(
    title = "USA Phase Average Ratio",
    x = "Phase",
    y = "Average Death to Case ratio (%)"
  ) +
  theme_minimal()

us_phase_means_plot
```

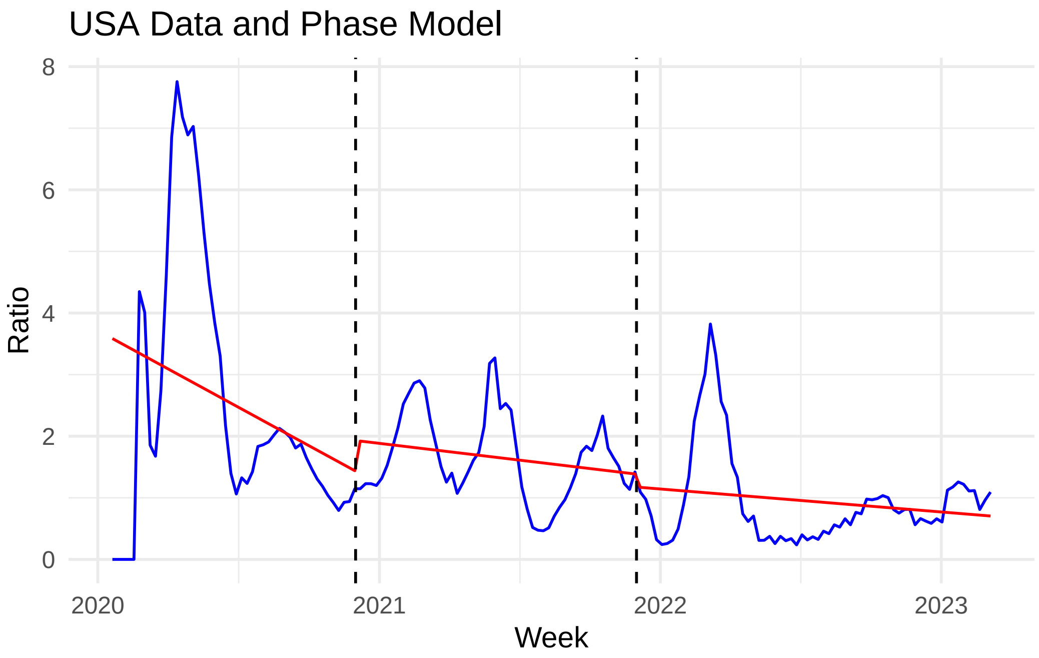


```
# do a simple phased linear regression
us_phase_model <- lm(us_Weekly_death_to_case_ratio ~ Week * factor(Phase), data = weekly_us_

summary(us_phase_model)
```

```
Call:
lm(formula = us_Weekly_death_to_case_ratio ~ Week * factor(Phase),
    data = weekly_us_combo)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5852 -0.6086 -0.1722  0.3683  4.7438

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                   1.281e+02  3.684e+01   3.478 0.000653 ***
Week                         -6.812e-03  1.998e-03  -3.410 0.000825 ***
factor(Phase)2. Rollout      -9.838e+01  4.829e+01  -2.037 0.043281 *
factor(Phase)3. Post         -1.076e+02  4.307e+01  -2.498 0.013528 *
Week:factor(Phase)2. Rollout  5.317e-03  2.599e-03   2.046 0.042435 *
Week:factor(Phase)3. Post     5.790e-03  2.311e-03   2.505 0.013256 *
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.259 on 158 degrees of freedom
Multiple R-squared:  0.2608,    Adjusted R-squared:  0.2374
F-statistic: 11.15 on 5 and 158 DF,  p-value: 3.255e-09
```

```
weekly_us_combo$us_Modeled <- predict(us_phase_model)

us_ratio_data_model <- ggplot(weekly_us_combo, aes(x = Week)) +
  geom_line(aes(y = us_Weekly_death_to_case_ratio), color = "blue") +
  geom_line(aes(y = us_Modeled), color = "red") +
  labs(title = "USA Data and Phase Model",
       x = "Week", y = "Ratio") +
  geom_vline(xintercept = vac_rollout_start, linetype = "dashed", color = "black") +
  geom_vline(xintercept = vac_rollout_end, linetype = "dashed", color = "black") +
  theme_minimal()

us_ratio_data_model
```



```
# join the global and USA data to see if I can plot them together nicely
combined_means <- inner_join(global_phase_means, us_phase_means, by = "Phase")
combined_means
```

```
# A tibble: 3 x 3
  Phase      global_phase_avg us_phase_avg
  <chr>                 <dbl>        <dbl>
1 1. Pre                3.61         2.51
2 2. Rollout            1.90         1.65
3 3. Post               0.474        0.937
```

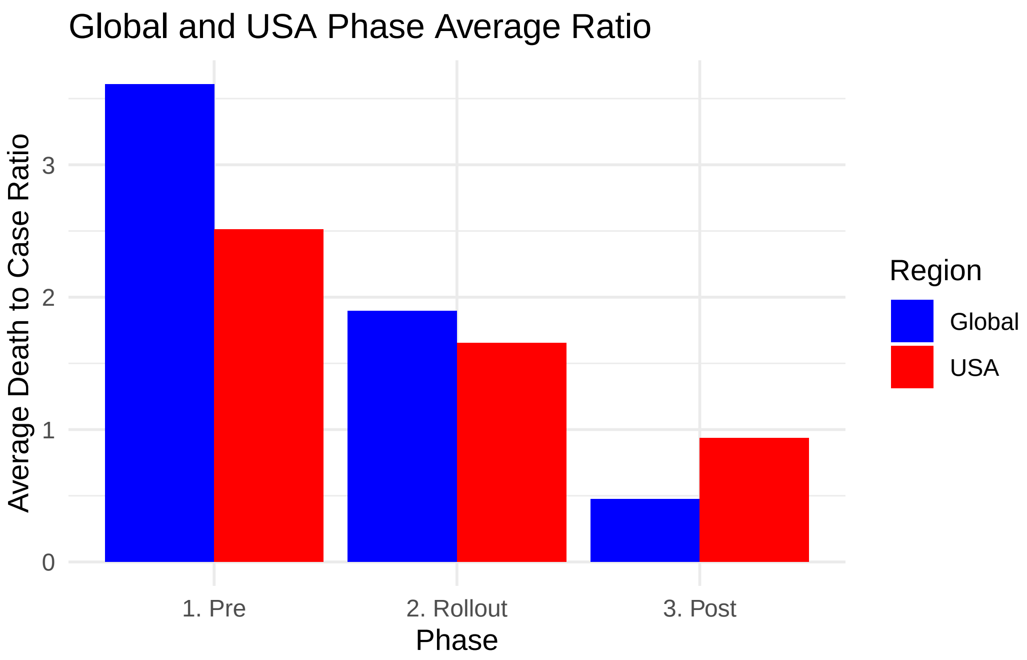
Final Models and Visuals

Combined phase averages

```
# Reshape the data to long format for plotting
long_combined_means <- combined_means %>%
  pivot_longer(cols = c(global_phase_avg, us_phase_avg),
               names_to = "Region",
               values_to = "Phase_avg") %>%
  mutate(Region = ifelse(Region == "global_phase_avg", "Global", "USA"))

long_combined_means
```

```
# A tibble: 6 x 3
  Phase      Region Phase_avg
  <chr>      <chr>      <dbl>
1 1. Pre     Global     3.61
2 1. Pre     USA        2.51
3 2. Rollout Global     1.90
4 2. Rollout USA        1.65
5 3. Post    Global     0.474
6 3. Post    USA        0.937
```

```
combined_means_plot <- ggplot(data = long_combined_means,
    aes(x = Phase, y = Phase_avg, fill = Region)) +
    geom_bar(stat = "identity", position = "dodge") +
    scale_fill_manual(values = c("Global" = "blue", "USA" = "red")) +
    labs(
        title = "Global and USA Phase Average Ratio",
        x = "Phase",
        y = "Average Death to Case Ratio") +
    theme_minimal()

print(combined_means_plot)
```

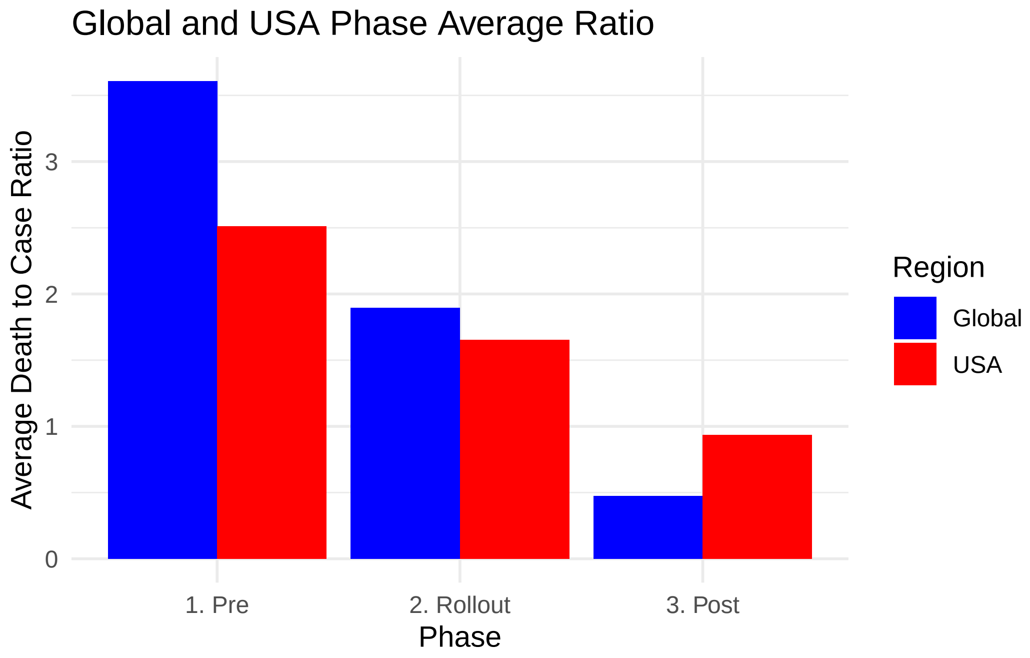


```r
combined_model_df <- inner_join(weekly_global_combo, weekly_us_combo, by = "Week")

tail(combined_model_df)
```

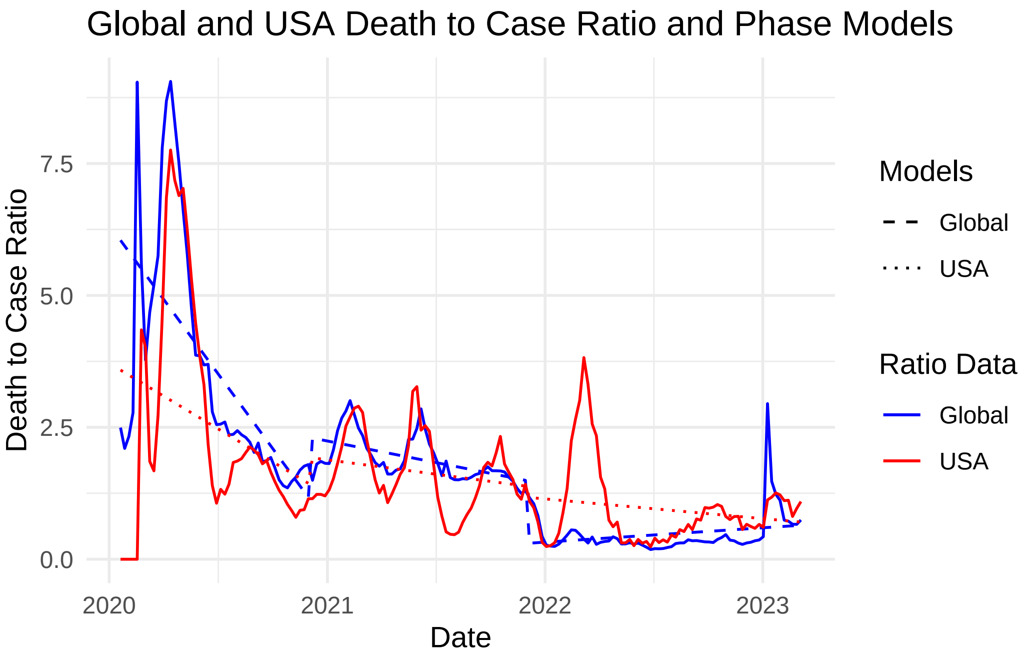
```
# A tibble: 6 x 15
  Week       g_Weekly_total_cases g_Weekly_total_deaths g_Weekly_new_cases
  <date>                    <int>                 <int>              <int>
1 2023-01-30            671721476               6847646            1377422
2 2023-02-06            672906177               6856419            1184701
3 2023-02-13            674056229               6864711            1150052
4 2023-02-20            675044414               6871268             988185
5 2023-02-27            676024901               6877749             980487
6 2023-03-06            676570149               6881802             545248
# i 11 more variables: g_Weekly_new_deaths <int>,
#   g_Weekly_death_to_case_ratio <dbl>, Phase.x <chr>, g_Modeled <dbl>,
#   us_Weekly_total_cases <int>, us_Weekly_total_deaths <int>,
#   us_Weekly_new_cases <int>, us_Weekly_new_deaths <int>,
#   us_Weekly_death_to_case_ratio <dbl>, Phase.y <chr>, us_Modeled <dbl>
```

Final Combined Models and Visuals

```
combined_model_plot <- ggplot(combined_model_df, aes(x = Week)) +
  # Global lines
  geom_line(aes(y = g_Weekly_death_to_case_ratio, color = "Global")) +
  geom_line(aes(y = g_Modeled, color = "Global", linetype = "Global")) +
  # USA lines
  geom_line(aes(y = us_Weekly_death_to_case_ratio, color = "USA")) +
  geom_line(aes(y = us_Modeled, color = "USA", linetype = "USA")
  ) +
  labs(
    title = "Global and USA Death to Case Ratio and Phase Models",
    x = "Date",
    y = "Death to Case Ratio"
  ) +
  scale_color_manual(name = "Ratio Data", values = c("Global" = "blue", "USA" = "red")) +
  scale_linetype_manual(name = "Models", values = c("Global" = "dashed", "USA" = "dotted")) +

theme_minimal()

print(combined_means_plot)
```



```
print(combined_model_plot)
```



I think that accomplishes the goal. With these two figures we can see the changes in the death to cases ratios before and after (and during) the vaccine rollout for the global and USA datasets, as well as the general linear trend during each phase.

Bias statement

There are a few potential sources of bias or inaccuracy in this data and the analysis.

- Collection bias
  - This data was taken from the Johns Hopkins Covid project source without modification. Johns Hopkins is assumed to be a credible source with good data handling practices, but the accuracy of this data is still dependent on the accuracy of the data recorded and reporting by each region. Countries or sub-regions with less developed public health infrastructured may be missing or under-reported.
- Analysis bias
  - A single start and end date were used to define the vaccine rollout phase. In reality, several countries (mostly more wealthy) had access to the vaccines months before poorer countries and were able to distribute them to the general population much more rapidly.

- Death counts are likely to causally lag case counts on the order of a few weeks. That lag wasn't considered when calculating the death to case ratios. It was assumed that over periods as long as the vaccine phases that the ratios would stabilize so that the averages would be meaningful representations of those phases. More granular analysis would require account for the case to death time lag.
- Specific vaccine rates were not considered as they weren't included in the data. The demonstrated decrease in the death to case ratios can be correlated to the general availability of vaccines, but further analysis of those ratios for specific regions or over smaller time or any quantitative assessment of vaccine effectiveness would require adjusting for specific vaccination rates.

```r
sessionInfo()
```

```
R version 4.4.1 (2024-06-14)
Platform: aarch64-apple-darwin20
Running under: macOS Sonoma 14.6.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Denver
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] lubridate_1.9.3 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
 [5] purrr_1.0.2     readr_2.1.5     tidyr_1.3.1     tibble_3.2.1
 [9] ggplot2_3.5.1   tidyverse_2.0.0

loaded via a namespace (and not attached):
 [1] gtable_0.3.5      jsonlite_1.8.8    compiler_4.4.1    tidyselect_1.2.1
 [5] scales_1.3.0      yaml_2.3.10       fastmap_1.2.0     R6_2.5.1
 [9] labeling_0.4.3    generics_0.1.3    knitr_1.48        munsell_0.5.1
[13] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.4       utf8_1.2.4
[17] stringi_1.8.4     xfun_0.46         timechange_0.3.0  cli_3.6.3
[21] withr_3.0.1       magrittr_2.0.3    digest_0.6.36     grid_4.4.1
```

```
[25] rstudioapi_0.16.0 hms_1.1.3         lifecycle_1.0.4   vctrs_0.6.5
[29] evaluate_0.24.0   glue_1.7.0        farver_2.1.2      fansi_1.0.6
[33] colorspace_2.1-1  rmarkdown_2.28    tools_4.4.1       pkgconfig_2.0.3
[37] htmltools_0.5.8.1
```