Covid-19 John Hopkins Data Analysis

In this document I will explore the Covid-19 from John Hopkins in order to better understand the impact of Covid-19 across the world.

The data from JH includes data from countries around the world and also specific to the US. In this analysis, I will focus on world data including cases, deaths, population, countries, and states.

First we will view the top 10 countries with the most total deaths. After, we will factor in population to calculate deaths per million to get a more accurate measure of the impact.

We will begin by importing time series data from Github. There will be 4 files.

```
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 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- 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 become errors
```

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_covid_19_da
file_names <- c("time_series_covid19_confirmed_US.csv","time_series_covid19_confirmed_global.csv","time_
urls <- str_c(url_in,file_names)
```

We will read in the data

```
global_cases <- read_csv(urls[2], show_col_types = FALSE)
global_deaths <- read_csv(urls[4], show_col_types = FALSE)
US_cases <- read_csv(urls[1], show_col_types = FALSE)
US_deaths <- read_csv(urls[3], show_col_types = FALSE)
```

We will start tidying the global cases data by pivoting dates to rows and keeping only relevant columns Province, Country, Date, Cases

```
global_cases <- global_cases %>% pivot_longer(cols = -c(`Province/State`,`Country/Region`,Lat,Long), nam
```

We will tidy up the data for global deaths

```
global_deaths <- global_deaths %>% pivot_longer(cols = -c(`Province/State`,`Country/Region`,Lat,Long), n
```

We can now join the global cases and global deaths tables

```
library(lubridate)
global_data <- global_cases %>% full_join(global_deaths) %>% rename(Country_Region = `Country/Region`, P
```

```
## Joining with `by = join_by(`Province/State`, `Country/Region`, date)`
```

Next we can remove rows where cases are 0

```
global_data <- global_data %>% filter(cases > 0)
```

For the final data preparation step, we will add in population.

We will import the population CSV file

```
global_population <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/mas
```

Before being able to join, we will create a combined key for global_data which we can use for the join

```
global_data <- global_data %>% unite("Combined_Key", c(Province_State,Country_Region), sep = ", ",na.rm
```

Now we join population_data to global_data

```
global_data <- global_data %>% left_join(global_population, by = c("Province_State","Country_Region")) %
```

Grouping data by country and retrieving the total number of deaths per country

```
global_data_by_country <- global_data %>% group_by(Country_Region) %>% summarize(cases = max(cases), dea
```

Filtering to top 10 countries with most deaths

```
global_top_10_deaths <- global_data_by_country %>% group_by(Country_Region) %>% summarize(cases = max(ca
```

Top 10 Countries with Most Covid-19 Deaths

Creating bar chart

```
library(scales)
```

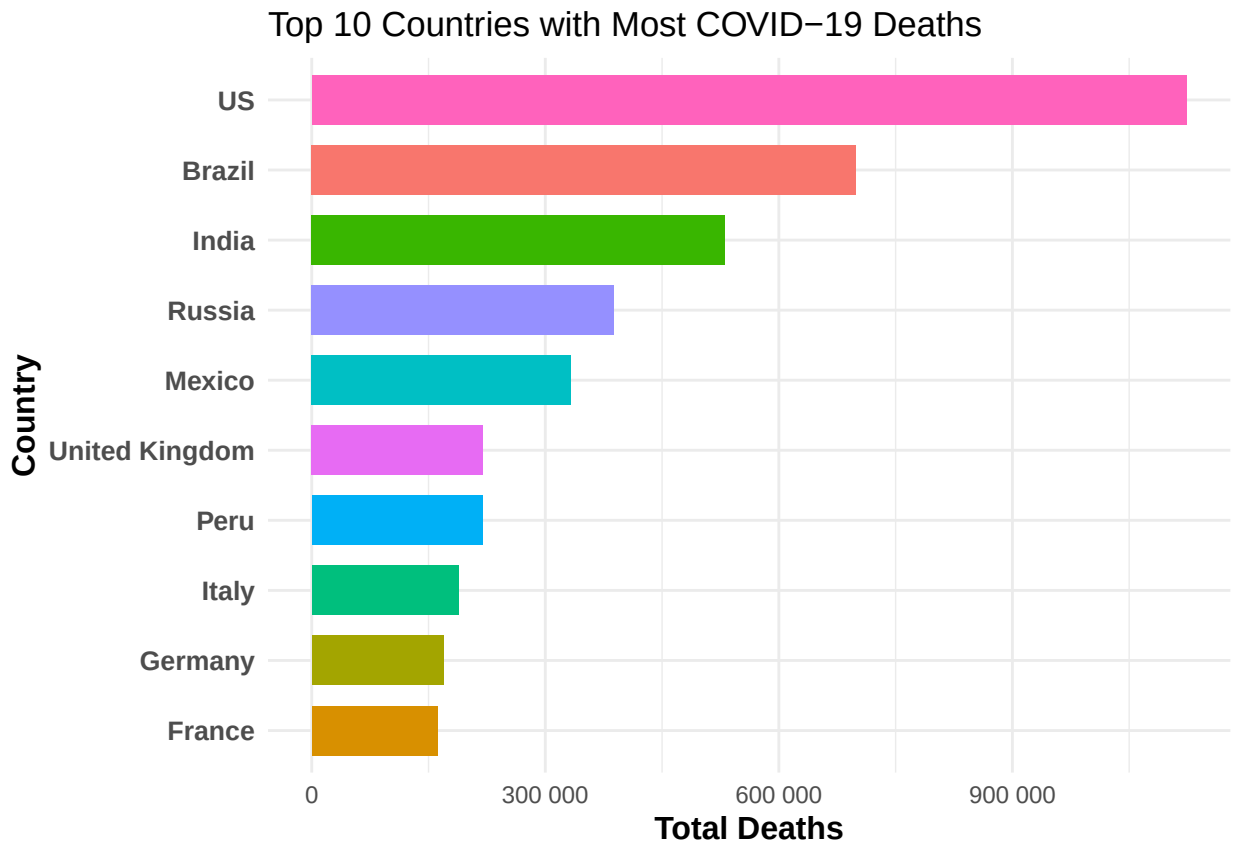
```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
ggplot(global_top_10_deaths, aes(x = reorder(Country_Region, deaths), y = deaths, fill = Country_Region)
  geom_bar(stat = "identity", width = 0.7, show.legend = FALSE) +
  labs(title = "Top 10 Countries with Most COVID-19 Deaths",
       x = "Country",
       y = "Total Deaths") +
  scale_y_continuous(labels = label_number(scale = 1)) +  # Format Y-axis only
```

```
  coord_flip() +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 10, face = "bold"),
        axis.title = element_text(size = 12, face = "bold"))
```



Visualize deaths per million per Country

Group data by country and year, and create a deaths per million variable.

```
global_death_perm_year <- global_data %>% mutate(year = year(date)) %>% group_by(Country_Region, year(da
```

```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

After doing a quick review of the data, it appears that the population data does not change yearly. I wanted to analyze if there was a change in deaths per million YOY but I will use the max year, population and deaths instead.

```
(global_death_perm_year)
```

```
## # A tibble: 790 x 7
## # Groups:   Country_Region [201]
```

```
##    Country_Region `year(date)` deaths  cases Population deaths_perm cases_perm
##    <chr>                 <dbl>  <dbl>  <dbl>      <dbl>       <dbl>      <dbl>
##  1 Afghanistan            2020   2189  52330   38928341        56.2      1344.
##  2 Afghanistan            2021   7356 158084   38928341       189.       4061.
##  3 Afghanistan            2022   7849 207559   38928341       202.       5332.
##  4 Afghanistan            2023   7896 209451   38928341       203.       5380.
##  5 Albania                2020   1181  58316    2877800       410.      20264.
##  6 Albania                2021   3217 210224    2877800      1118.      73050.
##  7 Albania                2022   3595 333806    2877800      1249.     115993.
##  8 Albania                2023   3598 334457    2877800      1250.     116220.
##  9 Algeria                2020   2756  99610   43851043        62.8      2272.
## 10 Algeria                2021   6276 218432   43851043       143.       4981.
## # i 780 more rows
```

We will tidy the global data to view deaths per million per country

```
global_death_perm <- global_data %>% group_by(Country_Region) %>% summarize(deaths = max(deaths), cases
global_death_perm
```

```
## # A tibble: 201 x 6
##    Country_Region      deaths    cases Population deaths_perm cases_perm
##    <chr>                <dbl>    <dbl>      <dbl>       <dbl>      <dbl>
##  1 Afghanistan           7896   209451   38928341       203.       5380.
##  2 Albania               3598   334457    2877800      1250.     116220.
##  3 Algeria               6881   271496   43851043       157.       6191.
##  4 Andorra                165    47890      77265      2136.     619815.
##  5 Angola                1933   105288   32866268        58.8      3204.
##  6 Antarctica               0       11         NA        NA          NA
##  7 Antigua and Barbuda    146     9106      97928      1491.      92987.
##  8 Argentina           130472 10044957   45195777      2887.     222254.
##  9 Armenia               8727   447308    2963234      2945.     150953.
## 10 Australia             7370  3915992    8118000       908.     482384.
## # i 191 more rows
```

Lets view the top countries by deaths per million.

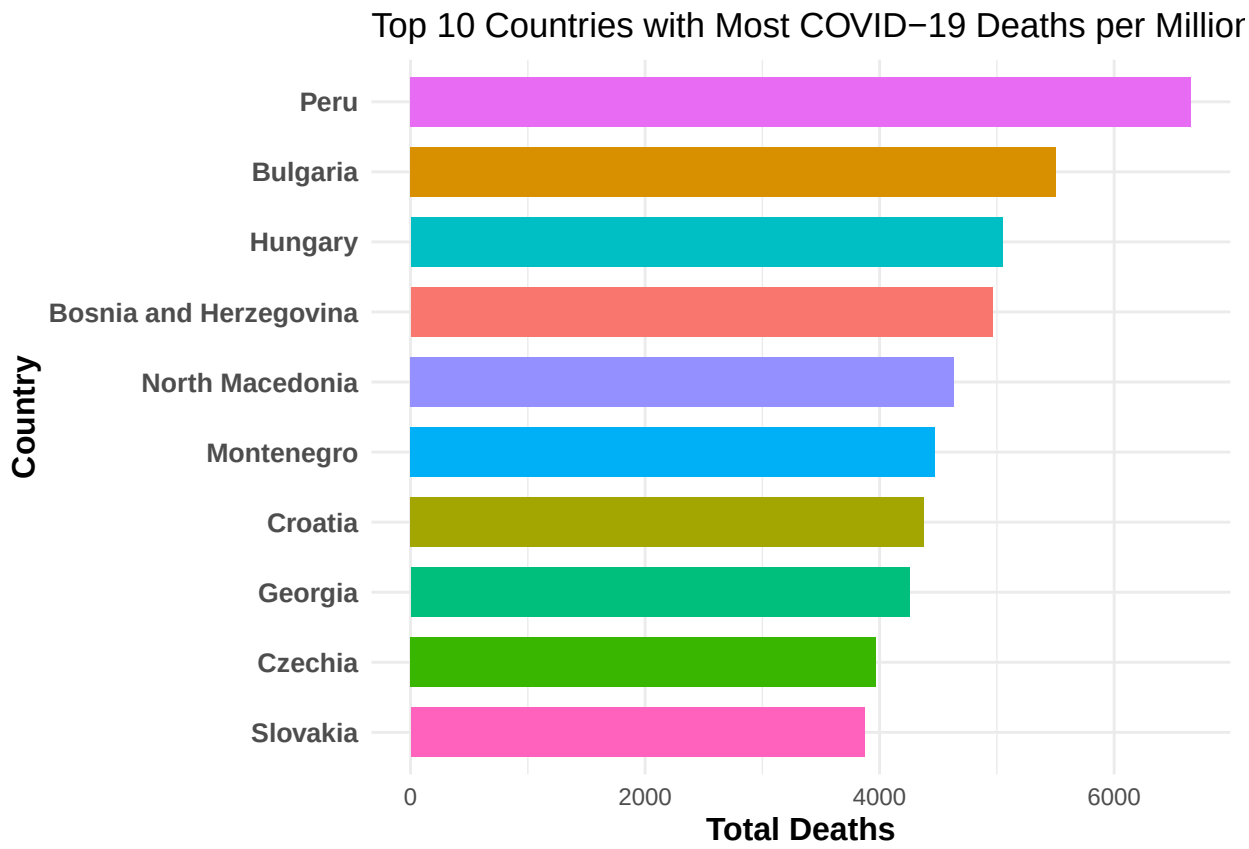
We will start by creating a data set to group the top 10 countries, sorted by descending order.

```
global_top_10_deathsperm <- global_death_perm %>% group_by(Country_Region) %>% summarize(deaths= deaths
```

We will create the bar graph

```
library(scales)
ggplot(global_top_10_deathsperm, aes(x = reorder(Country_Region, deaths), y = deaths, fill = Country_Reg
  geom_bar(stat = "identity", width = 0.7, show.legend = FALSE) +
  labs(title = "Top 10 Countries with Most COVID-19 Deaths per Million",
       x = "Country",
       y = "Total Deaths") +
  #scale_y_continuous(labels = label_number(scale = 1)) +  # Format Y-axis only

  coord_flip() +
    theme_minimal() +
  theme(axis.text.y = element_text(size = 10, face = "bold"),
        axis.title = element_text(size = 12, face = "bold"))
```



Observation

While the US had the most overall deaths, we can see that in terms of deaths per million, the US is not in the top 10. Peru was the most affected country, appearing at the top of deaths per million and also being present in top 10 countries with total deaths.

For comparison, we can check where the US ranks:

```
global_death_perm %>% group_by(Country_Region) %>% summarize(deaths= deaths_perm, .groups = "drop") %>%
```

```
## # A tibble: 1 x 3
##   Country_Region deaths  Rank
##   <chr>           <dbl> <int>
## 1 US              3411.    14
```

Modeling our Data

We will do a linear model to check predictions based on cases per million and deaths per million

```
mod <- lm(deaths_perm ~ cases_perm,data = global_death_perm)
summary(mod)
```

```
##
## Call:
```

```
## lm(formula = deaths_perm ~ cases_perm, data = global_death_perm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2403.4  -610.3  -389.2   471.8  5541.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.292e+02  1.116e+02   5.637 6.10e-08 ***
## cases_perm  3.580e-03  4.383e-04   8.167 4.13e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1151 on 192 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.2578, Adjusted R-squared:  0.254
## F-statistic:  66.7 on 1 and 192 DF,  p-value: 4.132e-14
```

I can see there are deletions due to data missing, we will drop rows where cases, deaths, or population is 0

```
global_death_perm <- global_death_perm %>% filter(Population > 0, cases >0, deaths >0)
```

We will run the model again

```
mod <- lm(deaths_perm ~ cases_perm,data = global_death_perm)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_perm ~ cases_perm, data = global_death_perm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2415.0  -621.3  -397.1   464.8  5530.2
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.410e+02  1.121e+02   5.718 4.12e-08 ***
## cases_perm  3.580e-03  4.389e-04   8.156 4.62e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1150 on 190 degrees of freedom
## Multiple R-squared:  0.2593, Adjusted R-squared:  0.2554
## F-statistic: 66.51 on 1 and 190 DF,  p-value: 4.623e-14
```

We can now generate a prediction variable which will allow us to visualize the model

```
global_death_perm %>% mutate(pred = predict(mod))
```

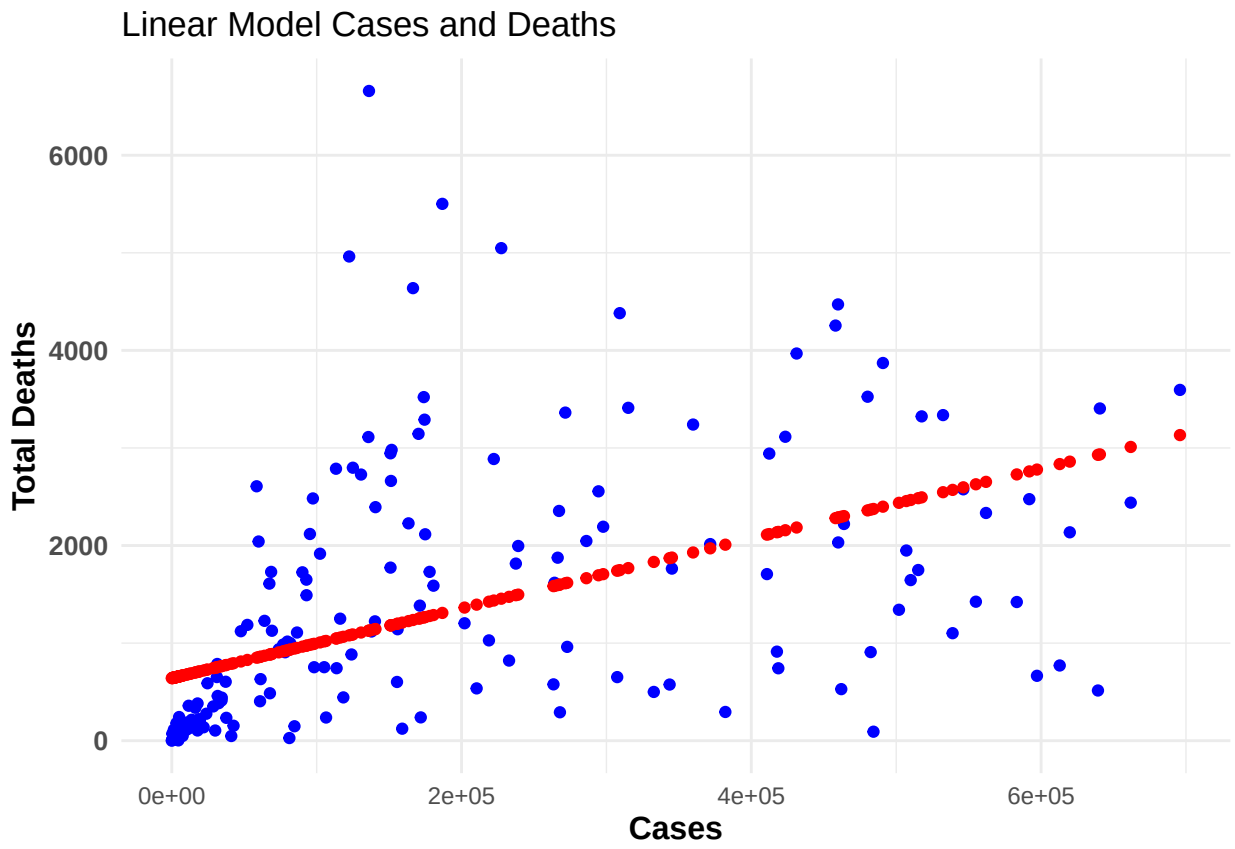
```
## # A tibble: 192 x 7
##    Country_Region      deaths    cases Population deaths_perm cases_perm  pred
```

```
##    <chr>                <dbl>    <dbl>      <dbl>       <dbl>      <dbl> <dbl>
##  1 Afghanistan           7896   209451   38928341       203.       5380.  660.
##  2 Albania               3598   334457    2877800      1250.     116220. 1057.
##  3 Algeria               6881   271496   43851043       157.       6191.  663.
##  4 Andorra                165    47890      77265      2136.     619815. 2860.
##  5 Angola                1933   105288   32866268        58.8      3204.  652.
##  6 Antigua and Barbuda    146     9106      97928      1491.      92987.  974.
##  7 Argentina           130472 10044957   45195777      2887.     222254. 1437.
##  8 Armenia               8727   447308    2963234      2945.     150953. 1181.
##  9 Australia             7370  3915992    8118000       908.     482384. 2368.
## 10 Austria              21970  5961143    9006400      2439.     661879. 3010.
## # i 182 more rows
```

```
global_deaths_pred <- global_death_perm %>% mutate(pred = predict(mod))
```

```
global_deaths_pred %>% ggplot() + geom_point(aes(x = cases_perm, y = deaths_perm), color = "blue") + ge
labs(title = "Linear Model Cases and Deaths",
       x = "Cases",
       y = "Total Deaths") +
  #scale_y_continuous(labels = label_number(scale = 1)) +  # Format Y-axis only

    theme_minimal() +
  theme(axis.text.y = element_text(size = 10, face = "bold"),
        axis.title = element_text(size = 12, face = "bold"))
```



Bias

From an initial look, it was easy to make an assumption on which countries were impacted the most. Some of the biggest countries with large populations like the US and Brazil show the most deaths. However, after adding variables to the data like cases and deaths per million, we can see that some of the countries most heavily impacted by Covid-19 were smaller countries like Peru and Bulgaria. This data doesnt tell the whole story so I searched online for information specific to Peru.

From what I found, the main reason for such a heavy impact was "... the collapse of an underfunded public health care system with low coverage among the population and a lack of adequate health care facilities, including enough hospitals to treat patients requiring intensive care (see more in Olivera, 2021). Levels of public investments in health have been lower in Peru than in other countries with similar economic development (Economic Commission for Latin America and the Caribbean (ECLAC), 2019)."

Source: https://pmc.ncbi.nlm.nih.gov/articles/PMC10271852/