Covid Data

M Bailey

2024-03-02

Getting Data

We access the data by getting the four csv names and appending them to the original link. The data is made available by the Johns Hopkins Center for Systems Science and Engineering (CSSE). We are going to look at four datasets in total that we will combine into master dataset containing deaths and cases for the US and abroad.

```
base_url <- paste0("https://raw.githubusercontent.com/",
    "CSSEGISandData/COVID-19/master/", "csse_covid_19_data/csse_covid_19_time_series/")

csv_names <- c("time_series_covid19_confirmed_US.csv",
    "time_series_covid19_confirmed_global.csv", "time_series_covid19_deaths_US.csv",
    "time_series_covid19_deaths_global.csv")

lookup_csv <- paste0("https://raw.githubusercontent.com/",
    "CSSEGISandData/COVID-19/master/", "csse_covid_19_data/UID_ISO_FIPS_LookUp_Table.csv")

file_urls = str_c(base_url, csv_names)

us_cases_raw = read_csv(file_urls[1], show_col_types = FALSE)
global_cases_raw = read_csv(file_urls[2], show_col_types = FALSE)
us_deaths_raw = read_csv(file_urls[3], show_col_types = FALSE)
global_deaths_raw = read_csv(file_urls[4], show_col_types = FALSE)
lookup = read_csv(lookup_csv)
```

```
## Rows: 4321 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): iso2, iso3, FIPS, Admin2, Province_State, Country_Region, Combined_Key
## dbl (5): UID, code3, Lat, Long_, Population
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
us_cases_raw
```

```
## # A tibble: 3,342 x 1,154
##         UID iso2  iso3  code3  FIPS Admin2   Province_State Country_Region   Lat
##       <dbl> <chr> <chr> <dbl> <dbl> <chr>    <chr>          <chr>          <dbl>
##  1 84001001 US    USA     840  1001 Autauga  Alabama        US              32.5
##  2 84001003 US    USA     840  1003 Baldwin  Alabama        US              30.7
```

```
##  3 84001005 US    USA     840  1005 Barbour  Alabama        US              31.9
##  4 84001007 US    USA     840  1007 Bibb     Alabama        US              33.0
##  5 84001009 US    USA     840  1009 Blount   Alabama        US              34.0
##  6 84001011 US    USA     840  1011 Bullock  Alabama        US              32.1
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # i 3,332 more rows
## # i 1,145 more variables: Long_ <dbl>, Combined_Key <chr>, `1/22/20` <dbl>,
## #   `1/23/20` <dbl>, `1/24/20` <dbl>, `1/25/20` <dbl>, `1/26/20` <dbl>,
## #   `1/27/20` <dbl>, `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>,
## #   `1/31/20` <dbl>, `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>,
## #   `2/4/20` <dbl>, `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>,
## #   `2/8/20` <dbl>, `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>, ...
```

```
global_cases_raw
```

```
## # A tibble: 289 x 1,147
##    `Province/State`  `Country/Region`   Lat   Long `1/22/20` `1/23/20` `1/24/20`
##    <chr>             <chr>            <dbl>  <dbl>     <dbl>     <dbl>     <dbl>
##  1 <NA>              Afghanistan       33.9  67.7          0         0         0
##  2 <NA>              Albania           41.2  20.2          0         0         0
##  3 <NA>              Algeria           28.0   1.66         0         0         0
##  4 <NA>              Andorra           42.5   1.52         0         0         0
##  5 <NA>              Angola           -11.2  17.9          0         0         0
##  6 <NA>              Antarctica       -71.9  23.3          0         0         0
##  7 <NA>              Antigua and Bar~  17.1 -61.8          0         0         0
##  8 <NA>              Argentina        -38.4 -63.6          0         0         0
##  9 <NA>              Armenia           40.1  45.0          0         0         0
## 10 Australian Capit~ Australia        -35.5 149.           0         0         0
## # i 279 more rows
## # i 1,140 more variables: `1/25/20` <dbl>, `1/26/20` <dbl>, `1/27/20` <dbl>,
## #   `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>, `1/31/20` <dbl>,
## #   `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>, `2/4/20` <dbl>,
## #   `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>, `2/8/20` <dbl>,
## #   `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>, `2/12/20` <dbl>,
## #   `2/13/20` <dbl>, `2/14/20` <dbl>, `2/15/20` <dbl>, `2/16/20` <dbl>, ...
```

```
us_deaths_raw
```

```
## # A tibble: 3,342 x 1,155
##         UID iso2  iso3  code3  FIPS Admin2   Province_State Country_Region   Lat
##       <dbl> <chr> <chr> <dbl> <dbl> <chr>    <chr>          <chr>          <dbl>
##  1 84001001 US    USA     840  1001 Autauga  Alabama        US              32.5
##  2 84001003 US    USA     840  1003 Baldwin  Alabama        US              30.7
##  3 84001005 US    USA     840  1005 Barbour  Alabama        US              31.9
##  4 84001007 US    USA     840  1007 Bibb     Alabama        US              33.0
##  5 84001009 US    USA     840  1009 Blount   Alabama        US              34.0
##  6 84001011 US    USA     840  1011 Bullock  Alabama        US              32.1
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
```

```
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # i 3,332 more rows
## # i 1,146 more variables: Long_ <dbl>, Combined_Key <chr>, Population <dbl>,
## #   `1/22/20` <dbl>, `1/23/20` <dbl>, `1/24/20` <dbl>, `1/25/20` <dbl>,
## #   `1/26/20` <dbl>, `1/27/20` <dbl>, `1/28/20` <dbl>, `1/29/20` <dbl>,
## #   `1/30/20` <dbl>, `1/31/20` <dbl>, `2/1/20` <dbl>, `2/2/20` <dbl>,
## #   `2/3/20` <dbl>, `2/4/20` <dbl>, `2/5/20` <dbl>, `2/6/20` <dbl>,
## #   `2/7/20` <dbl>, `2/8/20` <dbl>, `2/9/20` <dbl>, `2/10/20` <dbl>, ...
```

```
global_deaths_raw
```

```
## # A tibble: 289 x 1,147
##    `Province/State`  `Country/Region`   Lat   Long `1/22/20` `1/23/20` `1/24/20`
##    <chr>             <chr>            <dbl>  <dbl>     <dbl>     <dbl>     <dbl>
##  1 <NA>              Afghanistan       33.9  67.7          0         0         0
##  2 <NA>              Albania           41.2  20.2          0         0         0
##  3 <NA>              Algeria           28.0   1.66         0         0         0
##  4 <NA>              Andorra           42.5   1.52         0         0         0
##  5 <NA>              Angola           -11.2  17.9          0         0         0
##  6 <NA>              Antarctica       -71.9  23.3          0         0         0
##  7 <NA>              Antigua and Bar~  17.1 -61.8          0         0         0
##  8 <NA>              Argentina        -38.4 -63.6          0         0         0
##  9 <NA>              Armenia           40.1  45.0          0         0         0
## 10 Australian Capit~ Australia        -35.5 149.           0         0         0
## # i 279 more rows
## # i 1,140 more variables: `1/25/20` <dbl>, `1/26/20` <dbl>, `1/27/20` <dbl>,
## #   `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>, `1/31/20` <dbl>,
## #   `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>, `2/4/20` <dbl>,
## #   `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>, `2/8/20` <dbl>,
## #   `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>, `2/12/20` <dbl>,
## #   `2/13/20` <dbl>, `2/14/20` <dbl>, `2/15/20` <dbl>, `2/16/20` <dbl>, ...
```

In order to get the data into a friendly format, I needed to do some wrangling. This was a good test of my skills, as it wasn't super hard, but I did need to be mindful of joining the global tables by both date and country. After a fairly large amount of tidying and joining (shown below), I ended up with some good tables of data. One important thing to note is that the populations for the countries are static, so this is a decent representation of per capita, but not precise enough for high-accuracy use cases. After recognizing the presence of outlier data, I needed to locate and remove the impact of those numbers. My solution in such cases was to remove that row entirely so it would not affect the data. Due to lack of reporting, I drop North Korea, for instance.

```
us_cases <- us_cases_raw %>%
    select(Country_Region, Province_State, 12:last_col()) %>%
    pivot_longer(cols = 3:last_col(), names_to = "date",
        values_to = "cases") %>%
    mutate(date = mdy(date)) %>%
    group_by(date, Country_Region) %>%
    summarize(cases = sum(cases))

global_cases <- global_cases_raw %>%
    select("Country/Region", 5:last_col()) %>%
    pivot_longer(cols = 2:last_col(), names_to = "date",
        values_to = "cases") %>%
```

```
    mutate(date = mdy(date)) %>%
    filter(!`Country/Region` == "Korea, North") %>%
    group_by(date, `Country/Region`) %>%
    summarize(cases = sum(cases))

us_deaths <- us_deaths_raw %>%
    select(Population, 13:last_col()) %>%
    pivot_longer(cols = 3:last_col(), names_to = "date",
        values_to = "deaths") %>%
    mutate(date = mdy(date)) %>%
    group_by(date) %>%
    summarize(deaths = sum(deaths), population = sum(Population))

lookup <- lookup %>%
    select(Combined_Key, Population)

global_deaths_with_pop <- global_deaths_raw %>%
    inner_join(lookup, by = join_by(`Country/Region` ==
        Combined_Key))

global_deaths <- global_deaths_with_pop %>%
    select(!c(`Province/State`, Lat, Long)) %>%
    pivot_longer(cols = 2:last_col(offset = 1), names_to = "date",
        values_to = "deaths") %>%
    mutate(date = mdy(date)) %>%
    group_by(date, `Country/Region`, Population) %>%
    summarize(deaths = sum(deaths)) %>%
    filter(Population > 0)

all_US_deaths_and_cases <- us_cases %>%
    inner_join(us_deaths, by = "date")

all_global_deaths_and_cases <- global_cases %>%
    inner_join(global_deaths, by = join_by(`Country/Region`,
        "date")) %>%
    rename(population = Population, Country_Region = `Country/Region`)

summary(all_US_deaths_and_cases)
```

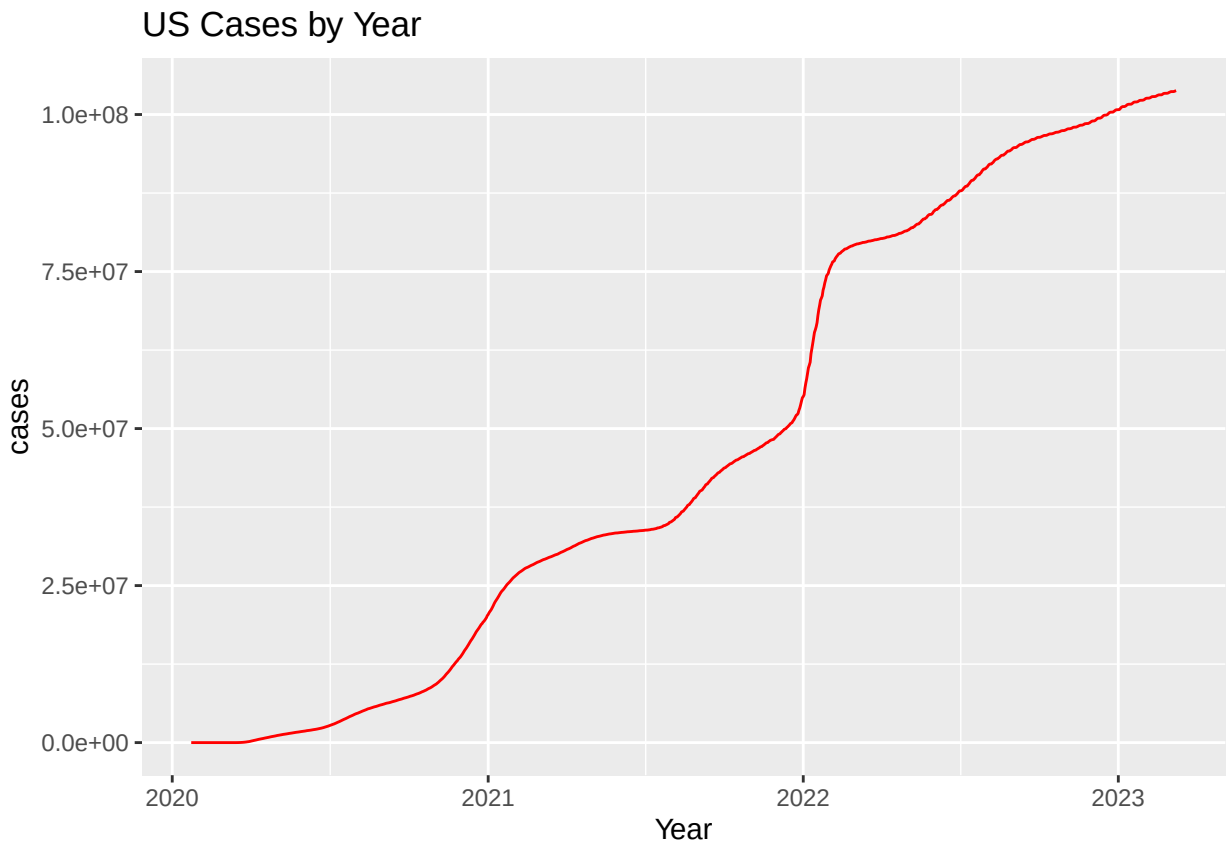
```
##       date            Country_Region         cases                deaths
##  Min.   :2020-01-23   Length:1142        Min.   :        1   Min.   :      1
##  1st Qu.:2020-11-03   Class :character   1st Qu.:  9490848   1st Qu.: 233654
##  Median :2021-08-15   Mode  :character   Median : 36942563   Median : 618552
##  Mean   :2021-08-15                      Mean   : 47122021   Mean   : 625110
##  3rd Qu.:2022-05-27                      3rd Qu.: 84087246   3rd Qu.:1006637
##  Max.   :2023-03-09                      Max.   :103802702   Max.   :1123836
##    population
##  Min.   :332875137
##  1st Qu.:332875137
##  Median :332875137
##  Mean   :332875137
##  3rd Qu.:332875137
##  Max.   :332875137
```

```r
summary(all_global_deaths_and_cases)
```

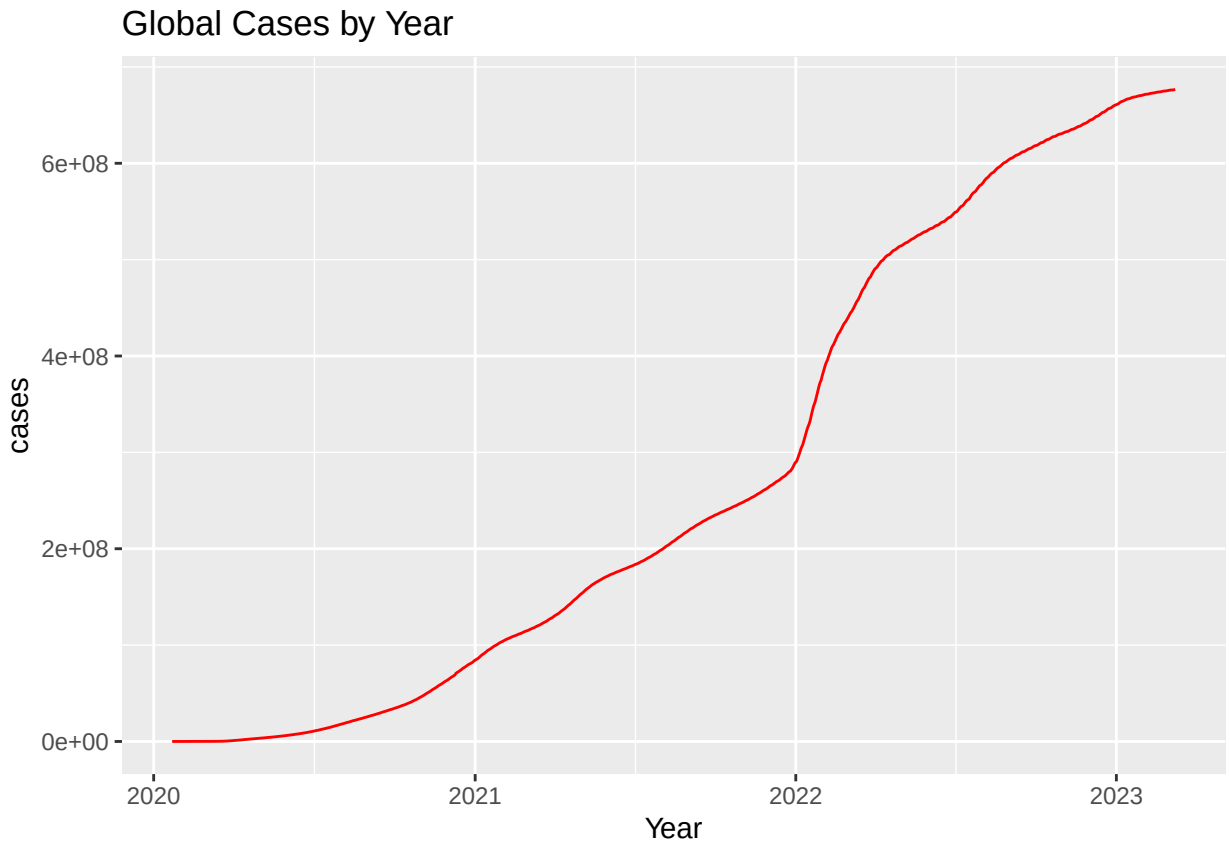
```
##       date             Country_Region         cases
##  Min.   :2020-01-22   Length:222885      Min.   :        0
##  1st Qu.:2020-11-02   Class :character   1st Qu.:     5349
##  Median :2021-08-15   Mode  :character   Median :    61901
##  Mean   :2021-08-15                      Mean   :  1421849
##  3rd Qu.:2022-05-28                      3rd Qu.:   523710
##  Max.   :2023-03-09                      Max.   :103802702
##    population            deaths
##  Min.   :8.090e+02   Min.   :      0
##  1st Qu.:1.886e+06   1st Qu.:     65
##  Median :8.737e+06   Median :    857
##  Mean   :3.958e+07   Mean   :  19830
##  3rd Qu.:2.914e+07   3rd Qu.:   7651
##  Max.   :1.412e+09   Max.   :1123836
```

Data Visualization

```r
all_US_deaths_and_cases %>%
    ggplot(aes(x = date, y = cases)) + geom_line(color = "red") +
    xlab("Year") + labs(title = "US Cases by Year")
```



```r
all_global_deaths_and_cases %>%
    group_by(date) %>%
    summarize(cases = sum(cases)) %>%
    ggplot(aes(x = date, y = cases)) + geom_line(color = "red") +
    xlab("Year") + labs(title = "Global Cases by Year")
```



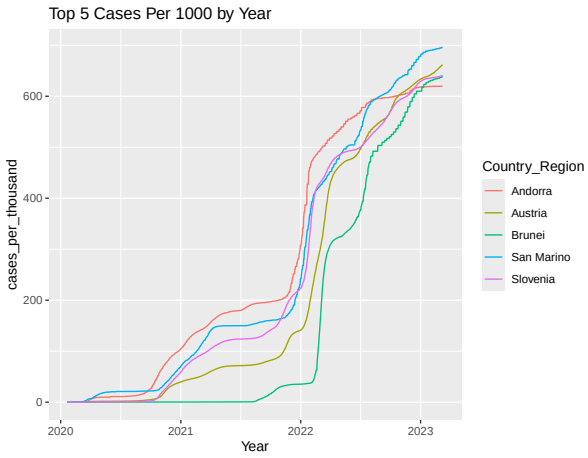
Data Analysis

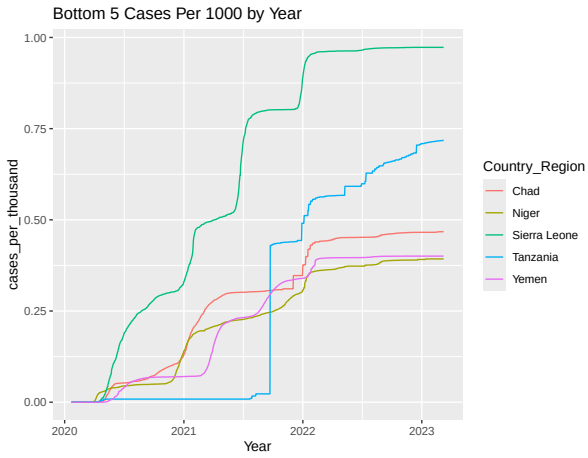
There are lots of interesting metrics that can be developed on top of the raw data that can give us a different perspective. For the sake of this project, I decided to add a column for the deaths and cases per 1000 people. After that, I ranked the top 5 and bottom 5 for both of these metrics, and plotted their rise over the duration of the data.

```r
all_US_deaths_and_cases <- all_US_deaths_and_cases %>%
    mutate(cases_per_thousand = (cases/population) *
        1000, deaths_per_thousand = (deaths/population) *
        1000)

all_global_deaths_and_cases <- all_global_deaths_and_cases %>%
    ungroup() %>%
    mutate(cases_per_thousand = (cases/population) *
        1000, deaths_per_thousand = (deaths/population) *
        1000)
```

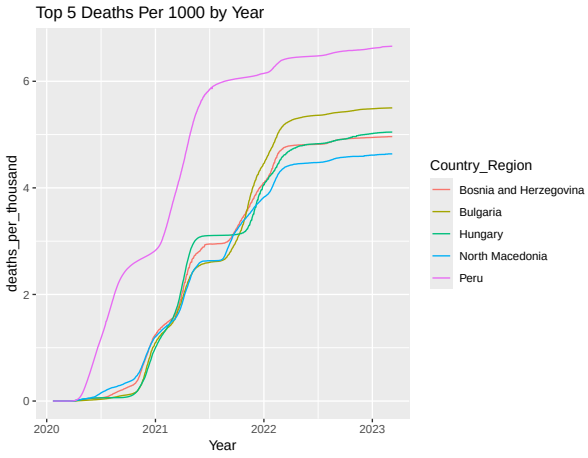
```r
top_five_death_countries <- all_global_deaths_and_cases %>%
    filter(date == max(all_global_deaths_and_cases$date)) %>%
    slice_max(order_by = deaths_per_thousand, n = 5) %>%
    select(c(Country_Region, deaths_per_thousand, population))

bottom_five_death_countries <- all_global_deaths_and_cases %>%
    filter(date == max(all_global_deaths_and_cases$date)) %>%
    filter(deaths_per_thousand > 0.01) %>%
    slice_min(order_by = deaths_per_thousand, n = 5) %>%
    select(c(Country_Region, deaths_per_thousand, population))

top_five_case_countries <- all_global_deaths_and_cases %>%
    filter(date == max(all_global_deaths_and_cases$date)) %>%
    slice_max(order_by = cases_per_thousand, n = 5) %>%
    select(c(Country_Region, cases_per_thousand, population))

bottom_five_case_countries <- all_global_deaths_and_cases %>%
    filter(date == max(all_global_deaths_and_cases$date)) %>%
    filter(cases_per_thousand > 0.01) %>%
    slice_min(order_by = cases_per_thousand, n = 5) %>%
    select(c(Country_Region, cases_per_thousand, population))
```

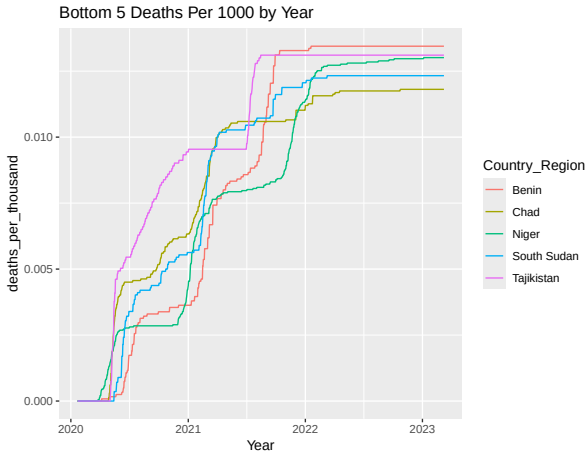
```r
# Top Five Case Plot
all_global_deaths_and_cases %>%
    filter(Country_Region == top_five_case_countries$Country_Region[1] |
        Country_Region == top_five_case_countries$Country_Region[2] |
        Country_Region == top_five_case_countries$Country_Region[3] |
        Country_Region == top_five_case_countries$Country_Region[4] |
        Country_Region == top_five_case_countries$Country_Region[5]) %>%
    ggplot(aes(x = date, y = cases_per_thousand)) +
    geom_line(aes(color = Country_Region)) + xlab("Year") +
    labs(title = "Top 5 Cases Per 1000 by Year")

# Bottom 5 Case Plot
all_global_deaths_and_cases %>%
    filter(Country_Region == bottom_five_case_countries$Country_Region[1] |
        Country_Region == bottom_five_case_countries$Country_Region[2] |
        Country_Region == bottom_five_case_countries$Country_Region[3] |
        Country_Region == bottom_five_case_countries$Country_Region[4] |
        Country_Region == bottom_five_case_countries$Country_Region[5]) %>%
    ggplot(aes(x = date, y = cases_per_thousand)) +
    geom_line(aes(color = Country_Region)) + xlab("Year") +
    labs(title = "Bottom 5 Cases Per 1000 by Year")
```






```r
# Top 5 Death Plot
all_global_deaths_and_cases %>%
    filter(Country_Region == top_five_death_countries$Country_Region[1] |
        Country_Region == top_five_death_countries$Country_Region[2] |
        Country_Region == top_five_death_countries$Country_Region[3] |
        Country_Region == top_five_death_countries$Country_Region[4] |
        Country_Region == top_five_death_countries$Country_Region[5]) %>%
    ggplot(aes(x = date, y = deaths_per_thousand)) +
    geom_line(aes(color = Country_Region)) + xlab("Year") +
    labs(title = "Top 5 Deaths Per 1000 by Year")

# Bottom 5 Death Plot
all_global_deaths_and_cases %>%
    filter(Country_Region == bottom_five_death_countries$Country_Region[1] |
        Country_Region == bottom_five_death_countries$Country_Region[2] |
        Country_Region == bottom_five_death_countries$Country_Region[3] |
        Country_Region == bottom_five_death_countries$Country_Region[4] |
        Country_Region == bottom_five_death_countries$Country_Region[5]) %>%
    ggplot(aes(x = date, y = deaths_per_thousand)) +
    geom_line(aes(color = Country_Region)) + xlab("Year") +
    labs(title = "Bottom 5 Deaths Per 1000 by Year")
```

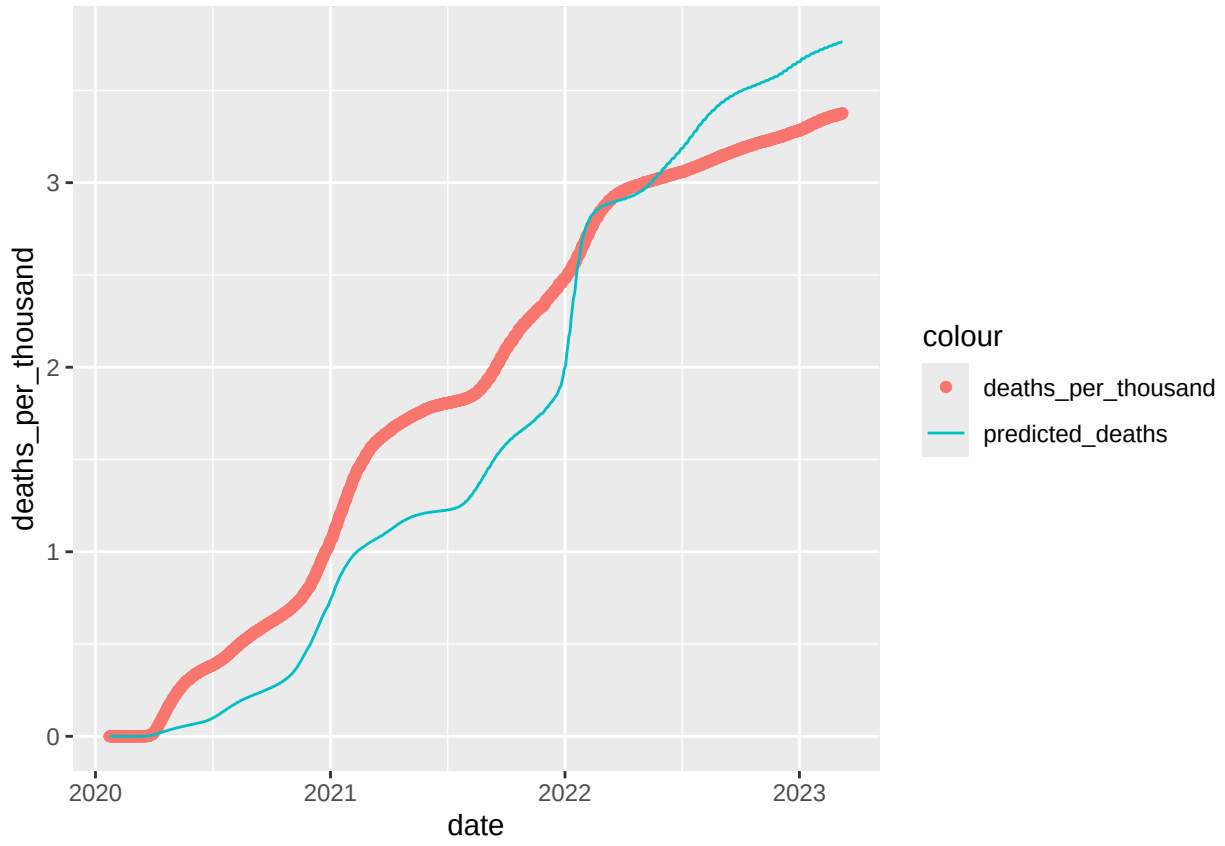




Data Model - Predicting Deaths By Cases in US

I decided to model the aggregate USA data to find a relationship between cases and deaths. This was done via a simple linear model, assuming that there's a certain fixed rate of lethality over a population. I think this model fails to be more accurate because it does not account for the appearance of new, less lethal variants of Covid.

```
mod = lm(deaths_per_thousand ~ cases_per_thousand -
    1, data = all_US_deaths_and_cases, )

x_max = max(all_US_deaths_and_cases$cases)
x_min = 1
x_points = x_min:x_max

all_US_deaths_and_cases <- all_US_deaths_and_cases %>%
    ungroup() %>%
    mutate(predicted_deaths = predict(mod))

all_US_deaths_and_cases %>%
    ggplot(aes(x = date)) + geom_point(aes(y = deaths_per_thousand,
    color = "deaths_per_thousand")) + geom_line(aes(y = predicted_deaths,
    color = "predicted_deaths"))
```



Bias Identification and Conlcusion

So this concludes my analysis of the covid data published by Johns Hopkins. Before concluding, I would like to address some of the main biases. The most glaring example would be reporting bias. Some countries clearly didn't report accurately (looking at you, North Korea), others seem to stop reporting (reference the bottom 5 cases graph), which results in a flat line of cases. These were the main issues I encountered with my data.

Most of the data wasn't particularly surprising since the covid pandemic was covered so widely. One fun suspicion to confirm was that people stopped caring in 2022 and it shows, as there were massive spikes in both the largest and smallest nations righ around the turn of that year. Despite that, it appears the 2022 was less lethal, probably due to the less lethal variant spreading at that time.

It's always fun to play with data pertaining to an event you experienced, as it allows you to confirm or debunk views you held about that particular experience. This concludes my report on Covid data from Johns Hopkins, thank you for reading!

Session Info

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.2.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## 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] ggplot2_3.5.0   lubridate_1.9.3 dplyr_1.1.4     readr_2.1.5
## [5] stringr_1.5.1   tidyr_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] bit_4.0.5         gtable_0.3.4      crayon_1.5.2      compiler_4.3.2
##  [5] tidyselect_1.2.0  parallel_4.3.2    scales_1.3.0      yaml_2.3.8
##  [9] fastmap_1.1.1     R6_2.5.1          labeling_0.4.3    generics_0.1.3
## [13] curl_5.2.0        knitr_1.45        tibble_3.2.1      munsell_0.5.0
## [17] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.3       utf8_1.2.4
## [21] stringi_1.8.3     xfun_0.42         bit64_4.0.5       timechange_0.3.0
## [25] cli_3.6.2         formatR_1.14      withr_3.0.0       magrittr_2.0.3
## [29] digest_0.6.34     grid_4.3.2        vroom_1.6.5       rstudioapi_0.15.0
## [33] hms_1.1.3         lifecycle_1.0.4   vctrs_0.6.5       evaluate_0.23
## [37] glue_1.7.0        farver_2.1.1      codetools_0.2-19  fansi_1.0.6
## [41] colorspace_2.1-0  rmarkdown_2.25    purrr_1.0.2       tools_4.3.2
```

```
## [45] pkgconfig_2.0.3   htmltools_0.5.7
```