COVID-19 Analysis Report

R. Rhoads

2022-12-05

Purpose

The purpose of this report is to analyze the COVID-19 global data provided by Johns Hopkins University. This data is publicly available for anyone to use and is available on their Github page (https://github.com/CSSEGISandData). The global data comprises of confirmed, recovered, and death cases. Each data set contains the count on a per day basis since 22 January 2020 with additional information about the country or region, the province or state, latitude, and longitude.

The setup of this report will include all details necessary for anyone to quickly and easily reproduce the results. The sections are broken down for important libraries used, the exact data set used and where it was downloaded from, how the data was processed, overall analysis, and a discussion on any bias that should be considered when working with this data.

Given the wealth and type of information in the data set, this report will focus on the following areas.

1. How have the confirmed cases been trending since the beginning of 2020? Have they been steadily increasing or plateauing?
2. How have the deaths been trending since the beginning of 2020?
3. How has the recovery been trending? Was there a difference in recovery with the dispersion of the COVID-19 vaccine?

Libraries

The `tidyverse` and `lubridate` libraries are necessary to reproduce any work that has been done in this report. The `sessioninfo` library is not necessary for any analysis since the output is used to provide the information of the session for this report.

```
# Import necessary libraries
library(tidyverse)
library(lubridate)
# Library used to provide session information
library(sessioninfo)
```

Data Source and Structure

The data in its entirety can be found in at the following URL. The data is in a Comma-Separated Values (CSV) format, the `read.csv()` function is used to read in the data. Since there are three different time series data sets for global data comprised of cases, deaths, and recovery, the final URL of each data set is broken into a base URL and file name and concatenated when imported.

```
# Set the base URL address
base_url <-
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti
# Set a list of csv files
file_names <-
  c("time_series_covid19_confirmed_global.csv",
    "time_series_covid19_deaths_global.csv",
    "time_series_covid19_recovered_global.csv")
# Concatenate base URL with individual file names
urls <- str_c(base_url, file_names)
# Load individual file names into separate data structure
global_cases <- read_csv(urls[1])
```

```
## Rows: 289 Columns: 1053
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1051): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
global_deaths <- read_csv(urls[2])
```

```
## Rows: 289 Columns: 1053
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1051): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
global_recovery <- read_csv(urls[3])
```

```
## Rows: 274 Columns: 1053
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1051): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

Each data set has the same column names. These include Province/State, Country/Region, Lat (latitude), Long (longitude) and each date since 22 January 2022.

Data Processing

The first step in processing the data was to change the data structure format. The `pivot_longer()` was used to turn the column headers of all dates into a column of all dates Since the intent is to look at the

trends over the past several years, the Province/State, Lat, and Long columns are removed from the data structure.

```
# Pivot data to make data better to work with
# Note: cols -> names of columns to pivot
# c() -> pivot these columns
# -c() -> do NOT pivot these columns
# cases
global_cases <- global_cases %>%
  pivot_longer(cols = -c(`Province/State`,
                         `Country/Region`,
                         Lat,
                         Long),
               names_to = "date",
               values_to = "cases") %>%
  select(-c(`Province/State`, Lat, Long)) %>%
  rename(Country_Region = `Country/Region`)
# deaths
global_deaths <- global_deaths %>%
  pivot_longer(cols = -c(`Province/State`,
                         `Country/Region`,
                         Lat,
                         Long),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(`Province/State`, Lat, Long)) %>%
  rename(Country_Region = `Country/Region`)
# recovery
global_recovery <- global_recovery %>%
  pivot_longer(cols = -c(`Province/State`,
                         `Country/Region`,
                         Lat,
                         Long),
              names_to = "date",
              values_to = "recovery") %>%
  select(-c(`Province/State`, Lat, Long)) %>%
  rename(Country_Region = `Country/Region`)
```

The next step is to summarize all counts for the individual days. This is done by grouping each country and date and counting the total *cases*, *deaths*, and *recovery*.

```
# Sum up all counts for each country for the day
# cases
global_cases <- global_cases %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases))
```

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

```
# deaths
global_deaths <- global_deaths %>%
  group_by(Country_Region, date) %>%
  summarize(deaths = sum(deaths))
```

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

```
# recovery
global_recovery <- global_recovery %>%
  group_by(Country_Region, date) %>%
  summarize(recovery = sum(recovery))
```

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

The final step is to merge all previous data structures into one global data structure. Additionally, the date column is converted from a character type to a date type. This step will serve as the x values for the plots in the following section. Once the data is merge, individually, the cases, deaths, and recovery data is filtered to only contain rows of data that are greater than 0. That information is then used to plot the data. Below is a summary of the data structure.

```
# Join the data structures into a single data structure
global <- global_cases %>%
  full_join(global_deaths) %>%
  full_join(global_recovery) %>%
  mutate(date = mdy(date))
```

```
## Joining, by = c("Country_Region", "date")
## Joining, by = c("Country_Region", "date")
```

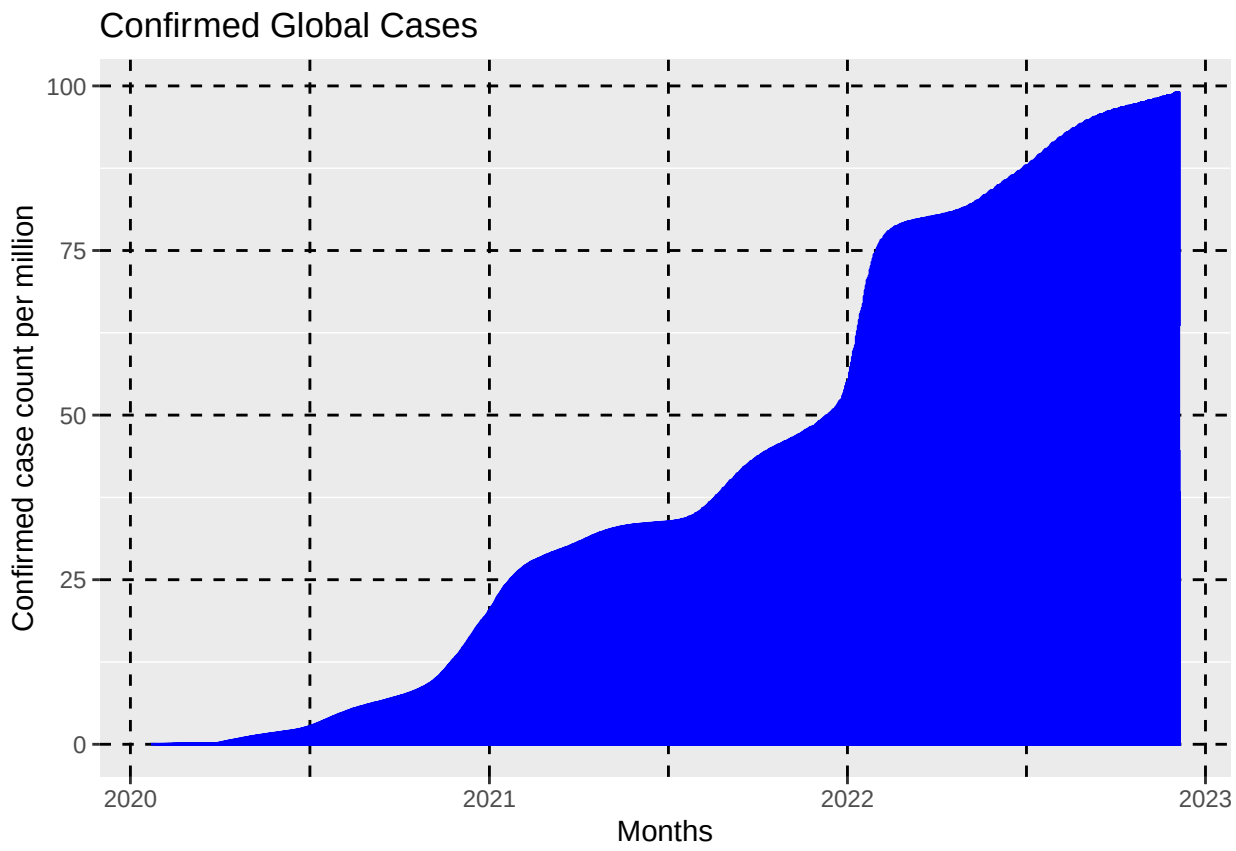
```
# Filter to contain data greater than 0
global_cases_filter <- global %>%
    filter(cases > 0)
global_deaths_filter <- global %>%
  filter(deaths > 0)
global_recovery_filter <- global %>%
  filter(recovery > 0)
# Overall summary of global data
summary(global)
```

```
##  Country_Region          date                cases              deaths
##  Length:210849      Min.   :2020-01-22   Min.   :       0   Min.   :      0
##  Class :character   1st Qu.:2020-10-10   1st Qu.:    2730   1st Qu.:     35
##  Mode  :character   Median :2021-06-29   Median :   42932   Median :    680
##                     Mean   :2021-06-29   Mean   : 1206271   Mean   :  17941
##                     3rd Qu.:2022-03-18   3rd Qu.:  420336   3rd Qu.:   6445
##                     Max.   :2022-12-05   Max.   :99023619   Max.   :1081638
##     recovery
##  Min.   :      -1
##  1st Qu.:       0
##  Median :       0
##  Mean   :  111414
##  3rd Qu.:    5820
##  Max.   :30974748
```

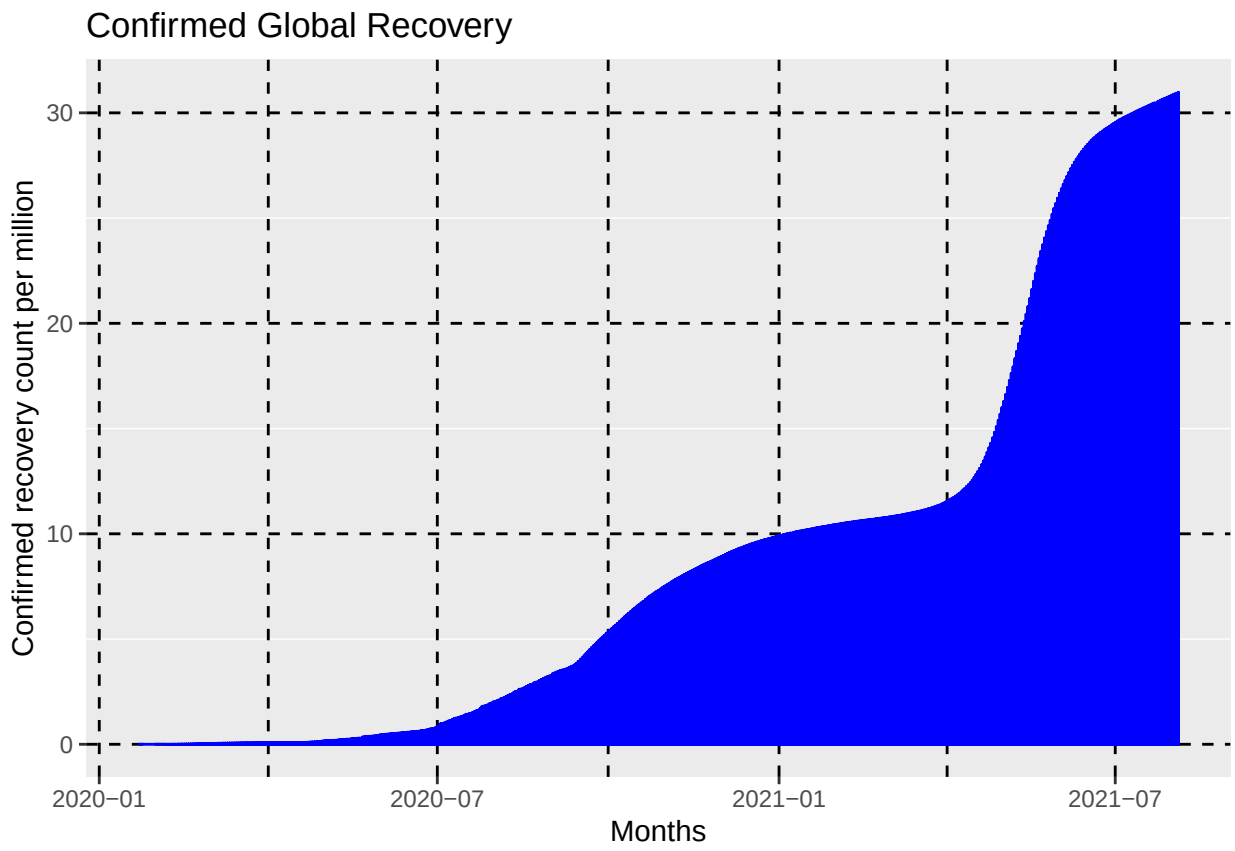
Analysis

To answer the question stated in the *Purpose* section, the global data is plotted. The following graphs represent the global case, death, and recovery count. Please not that although all graphs look similar, the death count graph has a different y-axis range to better visualize the data and the y-axis values are count per million.

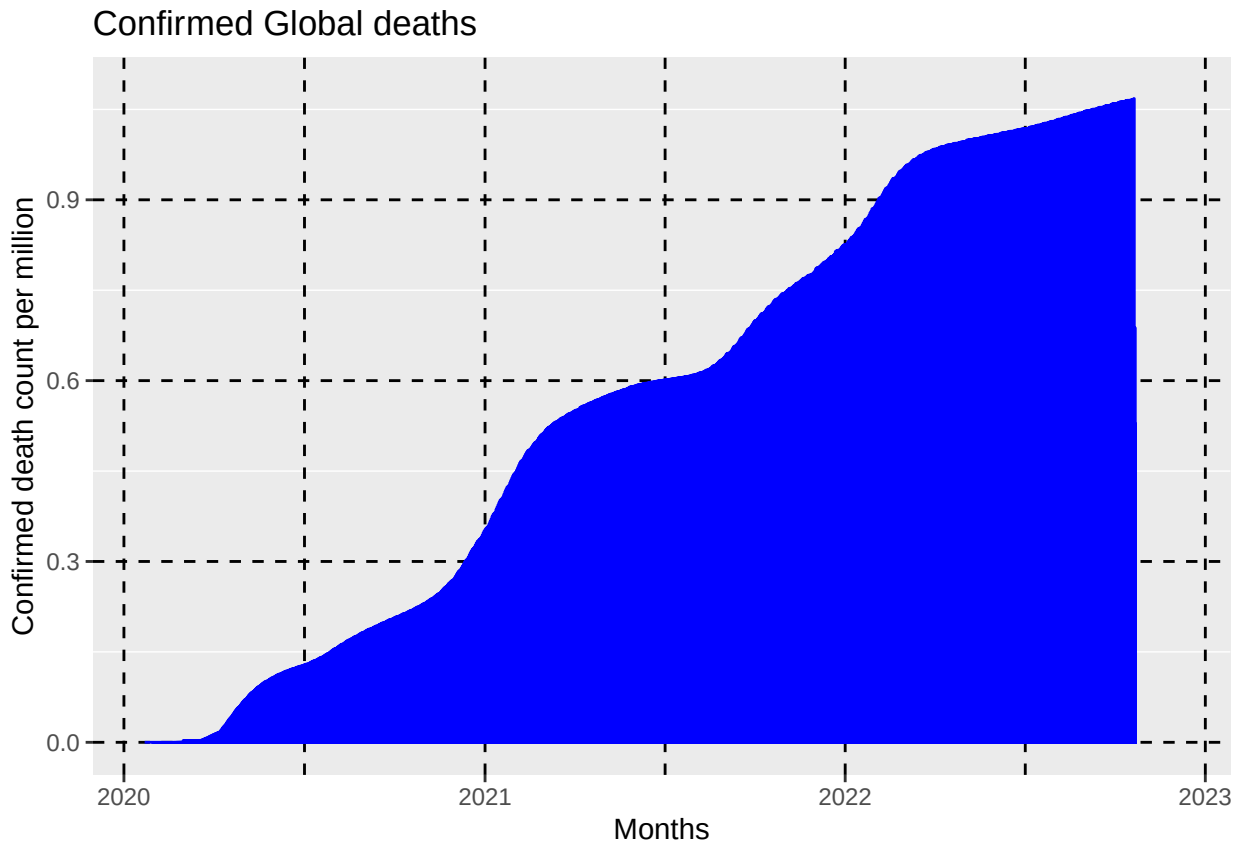
```
# Create the cases trend
cases_plot <- ggplot(data = global_cases_filter,
                     mapping = aes(x = date, y = cases/1e6)) +
              geom_line(color = "blue") +
              labs(title = "Confirmed Global Cases",
                   x = "Months",
                   y = "Confirmed case count per million") +
              theme(panel.grid.major = element_line(color = "black",
                                                    linewidth = 0.5,
                                                    linetype = 2),
              panel.grid.minor.x = element_line(color = "black",
                                                linewidth = 0.5,
                                                linetype = 2))
# recovery
recovery_plot <- ggplot(data = global_recovery_filter,
                        mapping = aes(x = date, y = recovery/1e6)) +
                 geom_line(color = "blue") +
                 labs(title = "Confirmed Global Recovery",
                      x = "Months",
                      y = "Confirmed recovery count per million") +
                 theme(panel.grid.major = element_line(color = "black",
                                                       linewidth = 0.5,
                                                       linetype = 2),
                 panel.grid.minor.x = element_line(color = "black",
                                                   linewidth = 0.5,
                                                   linetype = 2))

# deaths
deaths_plot <- ggplot(data = global_deaths_filter,
                      mapping = aes(x = date, y = deaths/1e6)) +
               geom_line(color = "blue") +
               labs(title = "Confirmed Global deaths",
                    x = "Months",
                    y = "Confirmed death count per million") +
               theme(panel.grid.major = element_line(color = "black",
                                                     linewidth = 0.5,
                                                     linetype = 2),
               panel.grid.minor.x = element_line(color = "black",
                                                 linewidth = 0.5,
                                                 linetype = 2))
cases_plot
```




```
recovery_plot
```


Confirmed Global Recovery
Confirmed recovery count per million
30
20
10
0
2020−01
2020−07
2021−01
2021−07
Months

```
deaths_plot
```



According to plots and the data used, the number of cases seem to be continuously increasing which overall does not seem likely. Additionally, the recovery trend seems to stop having data past August 2021 which is interesting. When plotting that information without the filters, the recovery count seems to be near 0 after August 2021, which also does not sound correct. The death count followed similar trends as the total cases, however, the y-scale is much lower when compared to total case count. The global data summary indicates that the maximum death count value is just over one million. The US alone has had that death count over the past several years so a maximum death count globally does not seem likely. Additionally, the start of the vaccine roll out in December 2020/January 2021, the recovery rate seems to be increasing. However, due to the missing data past August 2021, it is not possible to see how well the vaccine contributed to recovery.

Future Analysis

Overall, these graphs and data that was compiled does not seem to be fully representative of reality. Better and different analysis is needed to capture a better picture of what has happen globally due to COVID-19. Perhaps a better analysis would have been to look at a rolling average of the three data sets. Additionally, further investigation is needed as to why the recovery data seems to be missing data past August 2021. However, due to time constraints this option was not further investigated and is left for any future work.

Bias Considerations

The COVID-19 data set is comprised of a lot of data points. Some biases to consider are how individual countries reported COVID-19 cases when looking on the global scale. The United States followed followed CDC guidelines but even on a individual state level, these systems were not always connected properly. A lot of the world also followed their own guidelines and that of the World Health Organization (WHO) if

they were members of that organization. There were countries that were not reporting any cases or very few which seemed very unlikely given that the entire world was effected by COVID-19. The other bias to consider is the roll out of the vaccination. It was not available immediately and certainly not to all countries at once. This information would effect the recovery information about patients.

Appendix

Session Information

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] sessioninfo_1.2.2 lubridate_1.9.0   timechange_0.1.1  forcats_0.5.2
##  [5] stringr_1.4.1     dplyr_1.0.10      purrr_0.3.5       readr_2.1.3
##  [9] tidyr_1.2.1       tibble_3.1.8      ggplot2_3.4.0     tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.2.1    digest_0.6.30       utf8_1.2.2
##  [4] R6_2.5.1            cellranger_1.1.0    backports_1.4.1
##  [7] reprex_2.0.2        evaluate_0.18       highr_0.9
## [10] httr_1.4.4          pillar_1.8.1        rlang_1.0.6
## [13] googlesheets4_1.0.1 curl_4.3.3          readxl_1.4.1
## [16] rstudioapi_0.14     rmarkdown_2.18      labeling_0.4.2
## [19] googledrive_2.0.0   bit_4.0.4           munsell_0.5.0
## [22] broom_1.0.1         compiler_4.2.2      modelr_0.1.10
## [25] xfun_0.34           pkgconfig_2.0.3     htmltools_0.5.3
## [28] tidyselect_1.2.0    fansi_1.0.3         crayon_1.5.2
## [31] tzdb_0.3.0          dbplyr_2.2.1        withr_2.5.0
## [34] grid_4.2.2          jsonlite_1.8.3      gtable_0.3.1
## [37] lifecycle_1.0.3     DBI_1.1.3           magrittr_2.0.3
## [40] scales_1.2.1        cli_3.4.1           stringi_1.7.8
## [43] vroom_1.6.0         farver_2.1.1        fs_1.5.2
## [46] xml2_1.3.3          ellipsis_0.3.2      generics_0.1.3
## [49] vctrs_0.5.0         tools_4.2.2         bit64_4.0.5
## [52] glue_1.6.2          hms_1.1.2           parallel_4.2.2
## [55] fastmap_1.1.0       yaml_2.3.6          colorspace_2.0-3
## [58] gargle_1.2.1        rvest_1.0.3         knitr_1.41
## [61] haven_2.5.1
```