COVID 19 CASES

TT

2024-05-07

About the Report

We will be continuing with the covid-19 dataset using in the Lecture. I will be adding two visualizations to check:

- which country has the **most** recorded covid-19 deaths in the dataset.
- List the Top 20 countries with the **most** recorded covid-19 deaths.
- List of 20 countries in order of death-to-cases ratio.

The libraries used:

- tidyverse
- lubridate
- knitr

The datasets used:

- https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv
- https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv
- https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_US.csv
- https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_US.csv
- https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UID_ISO_FIPS_LookUp_Table.csv

import data

```
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 become errors
```

```
library(lubridate)
library(knitr)

main_url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_co

file_names = c("time_series_covid19_confirmed_global.csv",
"time_series_covid19_deaths_global.csv",
"time_series_covid19_confirmed_US.csv",
"time_series_covid19_deaths_US.csv")

urls <- str_c(main_url, file_names)

global_cases <- read_csv(urls[1])
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): 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: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): 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.
```

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

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## 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.
```

```
us_deaths <- read_csv(urls[4])
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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.
```

tidy the data

```
global_cases <- global_cases %>%
  pivot_longer(cols = -c('Province/State',
                         'Country/Region', Lat, Long),
               names_to = "date",
               values_to = "cases") %>%
  select(-c(Lat,Long))

global_deaths <- global_deaths %>%
  pivot_longer(cols = -c('Province/State',
                         'Country/Region', Lat, Long),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat,Long))

global <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = `Country/Region`,
         Province_State = `Province/State`) %>%
  mutate(date = mdy(date))
```

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

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

summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:306827      Length:306827      Min.   :2020-01-22   Min.   :        1
##  Class :character   Class :character   1st Qu.:2020-12-12   1st Qu.:     1316
##  Mode  :character   Mode  :character   Median :2021-09-16   Median :    20365
##                                        Mean   :2021-09-11   Mean   :  1032863
##                                        3rd Qu.:2022-06-15   3rd Qu.:   271281
##                                        Max.   :2023-03-09   Max.   :103802702
##      deaths
##  Min.   :      0
##  1st Qu.:      7
##  Median :    214
```

```
##  Mean   :  14405
##  3rd Qu.:   3665
##  Max.   :1123836
```

```r
# check if maximum values seem correct
# global %>% filter(cases > 100000000)


# tidy us data
us_cases <- us_cases %>%
  pivot_longer(cols = -(UID:Combined_Key),
               names_to = "date",
               values_to = "cases") %>%
  select(Admin2:cases) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))

us_deaths <- us_deaths %>%
  pivot_longer(cols = -(UID:Population),
               names_to = "date",
               values_to = "death") %>%
  select(Admin2:death) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))

us <- us_cases %>%
  full_join(us_deaths)
```

```
## Joining with `by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, date)`
```

```r
global <- global %>%
  unite("Combined_Key",
        c(Province_State, Country_Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)

uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/U

uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
```

```
## 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.
```

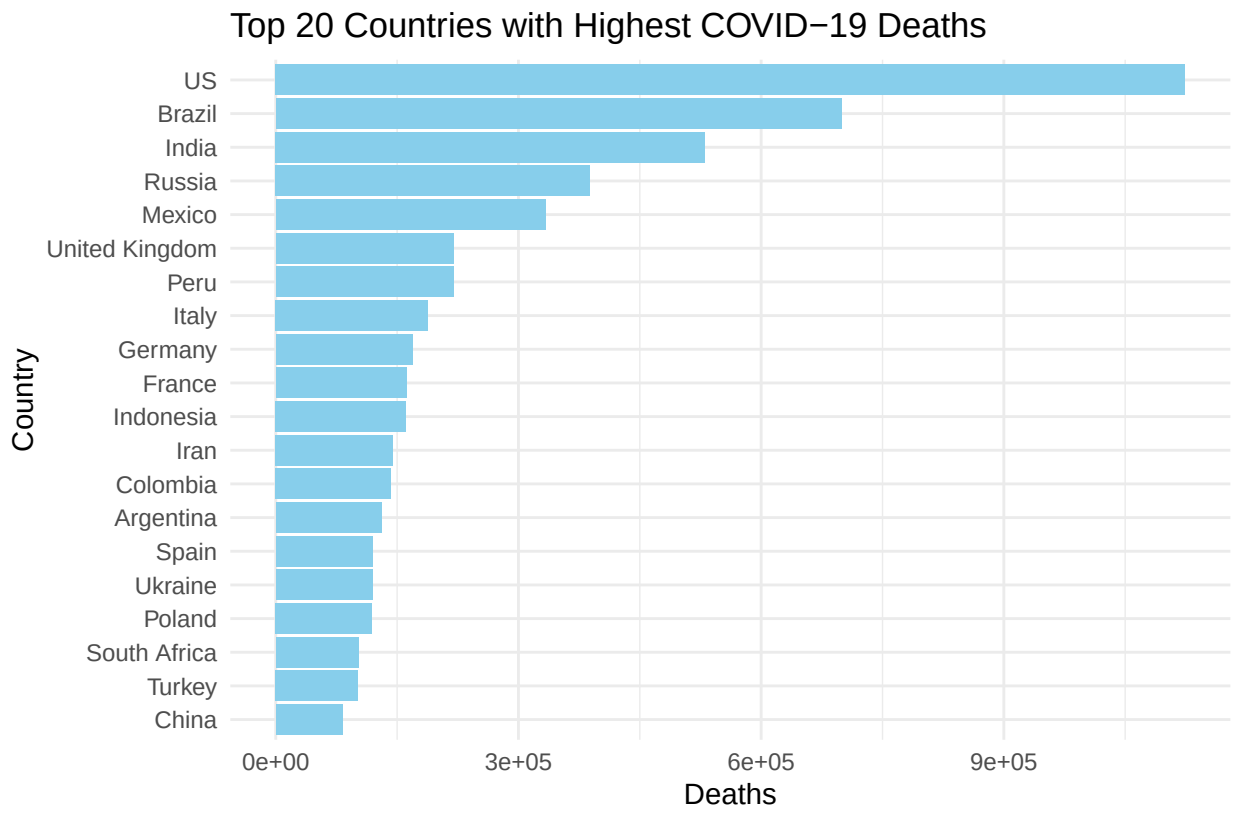
```r
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date,
         cases, deaths, Population,
         Combined_Key)
```

Top 20 Countries most Death

```r
# let's show top 20 results, more results is hard to see on the graph.
top_20_countries <- global %>%
  group_by(Country_Region) %>%
  summarize(max_deaths = max(deaths, na.rm = TRUE)) %>%
  top_n(20, max_deaths)

# Sort the top 20 countries by maximum deaths in descending order
top_20_countries <- top_20_countries[order(-top_20_countries$max_deaths),]

# Create a plot
ggplot(top_20_countries, aes(x = reorder(Country_Region, max_deaths), y = max_deaths)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  labs(x = "Country", y = "Deaths", title = "Top 20 Countries with Highest COVID-19 Deaths",
       caption = "visualization 1") +
  theme_minimal()
```



visualization 1

Below is the modal of the data (most Deaths):

The model of data showing countries with the most recorded covid-19 deaths:

```
kable(top_20_countries)
```

Country_Region	max_deaths
US	1123836
Brazil	699276
India	530779
Russia	388478
Mexico	333188
United Kingdom	219948
Peru	219539
Italy	188322
Germany	168935
France	161512
Indonesia	160941
Iran	144933
Colombia	142339
Argentina	130472
Spain	119479
Ukraine	119283
Poland	119010
South Africa	102595
Turkey	101492

Country_Region	max_deaths
China	82195

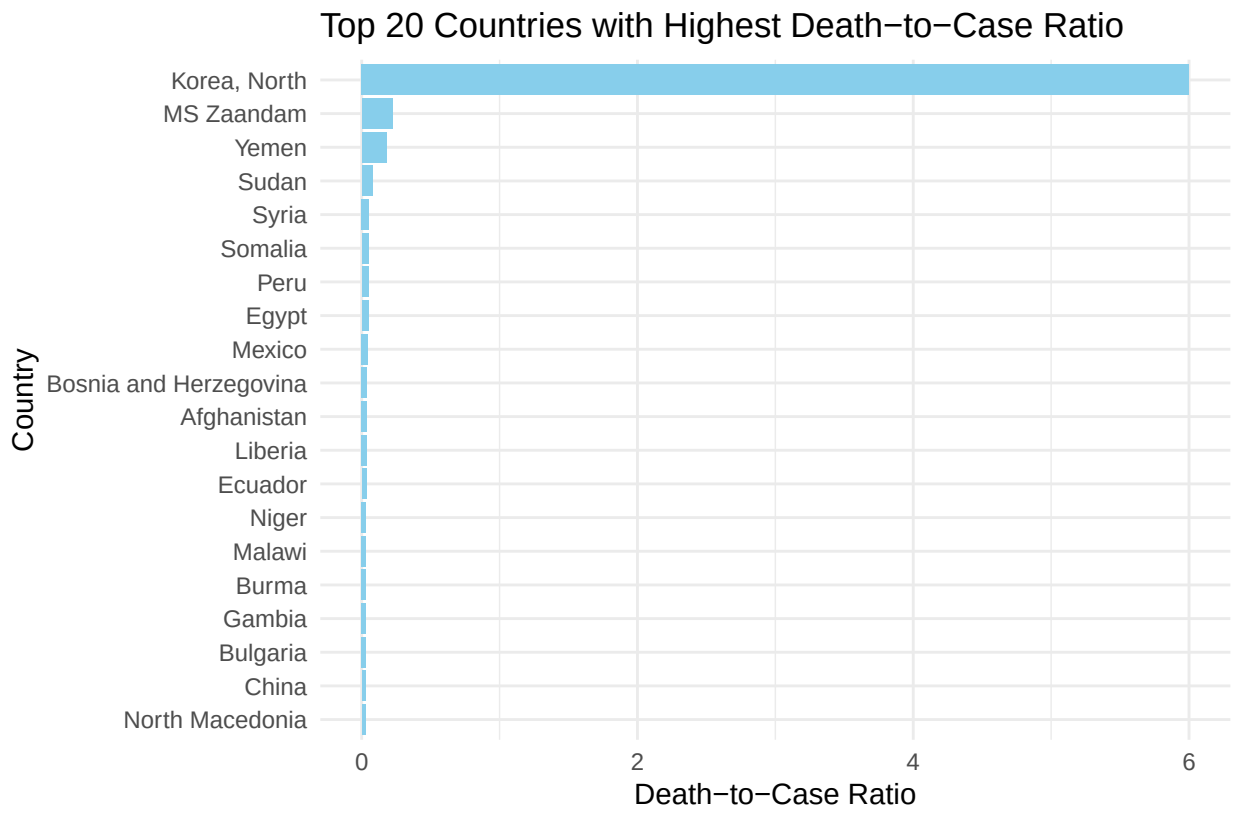
Top Countries Death-to-cases Ratio

```
country_summary <- global %>%
  group_by(Country_Region) %>%
  summarize(max_deaths = max(deaths, na.rm = TRUE),
            max_cases = max(cases, na.rm = TRUE)) %>%
  ungroup()

# Calculate the ratio of deaths divided by cases
country_summary <- country_summary %>%
  mutate(death_case_ratio = max_deaths / max_cases)

# Filter the top 20 countries with the highest death-to-case ratio
top_20_ratio_countries <- country_summary %>%
  top_n(20, death_case_ratio) %>%
  arrange(desc(death_case_ratio))

# Create a bar plot for the top 20 countries with highest death-to-case ratio
ggplot(top_20_ratio_countries, aes(x = reorder(Country_Region, death_case_ratio), y = death_case_ratio)
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  labs(x = "Country", y = "Death-to-Case Ratio",
       title = "Top 20 Countries with Highest Death-to-Case Ratio",
       caption = "visualization 2") +
  theme_minimal()
```



visualization 2

Below is the modal of the data (most Deaths to cases ratio):

The model of data showing countries with the most recorded covid-19 death-to-case ratio:

```
kable(top_20_ratio_countries)
```

Country_Region	max_deaths	max_cases	death_case_ratio
Korea, North	6	1	6.0000000
MS Zaandam	2	9	0.2222222
Yemen	2159	11945	0.1807451
Sudan	5017	63829	0.0786006
Syria	3164	57467	0.0550577
Somalia	1361	27324	0.0498097
Peru	219539	4487553	0.0489218
Egypt	24812	515759	0.0481077
Mexico	333188	7483444	0.0445234
Bosnia and Herzegovina	16280	401729	0.0405248
Afghanistan	7896	209451	0.0376986
Liberia	295	8090	0.0364648
Ecuador	36014	1057121	0.0340680
Niger	315	9508	0.0331300
Malawi	2896	88707	0.0326468
Burma	19490	633950	0.0307437
Gambia	372	12598	0.0295285
Bulgaria	38228	1297523	0.0294623
China	82195	2876106	0.0285786

Country_Region	max_deaths	max_cases	death_case_ratio
North Macedonia	9662	346852	0.0278563

2.Statement of Question

Q. During the recorded period of 2020 Jan until 2023 March which Country recorded the **most** Deaths?

A. The US has the most recorded no. of Deaths with a total count of 1,123,836.

Conclusion & Bias

My bias:

- The Cases & Death data is only accurate if every country made the same effort in recording the cases.
- Countries will differ in Logistics, Infrastructure which can and will alter the data for the cases.

This was a quick analysis using R to play around with the covid-19 dataset. The findings in this analysis is no where near thorough or accurate.

For example, the US has the highest no. of recorded covid-19 deaths in this dataset but it's difficult to make any conclusions from this result because of these **bias**:

- we have not accounted for the population of each country
- we have not accounted for the **age** of each individual with cases and deaths.
- each country had different regulations regarding quarantine, testing & vaccination.
- different countries had access to different vaccines at different intervals.

For the second visualization of death-to-cases ratio. It is clear that the first two results (N.Korea & MS Zaandam) are anomalies and don't represent the same characteristics with the results from the other countries.

Bias: Did N.Korea really only have 1 case?

Note: Also the y-axis in both of the visuals are not the most ideal:

- The 1st visual's y-axis can be changed to show the full numbers with comma separators.
- The 2nd visual's y-axis can be shown as a decimal or a percentage. Installing the library(scales) seems like one solution, but wasn't added for reproducible purposes.

```
"ran sessioninfo in console:
 sessionInfo()
R version 4.4.0 (2024-04-24 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 11 x64 (build 22631)

Matrix products: default


locale:
[1] LC_COLLATE=English_United States.utf8  LC_CTYPE=English_United States.utf8
```

```
[3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

time zone: Asia/Hong_Kong
tzcode source: internal

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

loaded via a namespace (and not attached):
 [1] compiler_4.4.0    fastmap_1.1.1     cli_3.6.2         htmltools_0.5.8.1 tools_4.4.0
 [6] yaml_2.3.8        tinytex_0.50      rmarkdown_2.26    knitr_1.46        digest_0.6.35
[11] xfun_0.43         rlang_1.1.3       evaluate_0.23"
```

```
## [1] "ran sessioninfo in console:\n sessionInfo()\nR version 4.4.0 (2024-04-24 ucrt)\nPlatform: x86_6
```