COVID-19 cases studies

Introduction

To find Rmd version, if needed, please, visit my github: https://github.com/IMosia/NYPD_shooting/blob/main/COVID_19.Rmd I was not sure which version is better to attach. Sorry for the possible inconvenience.

This project is dedicated to analysis of COVID-19 cases data.
The project is based on the data provided by Johns Hopkins University.
The main focus of the project is to look at the data in geographical and temporal dimensions.
It is a part of the course "Data Science as a Field" in Master of Science in Data Science at the University of Colorado.

Data Description

Data is provided from Johns Hopkins University. https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series
It is collected by the University and is available on GitHub.
Original data sources mostly country specific.
As data dedicated to the US was covered during lectures, the focus of this project is on the global data.

```
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(tidyr))
suppressPackageStartupMessages(library(lubridate))
suppressPackageStartupMessages(library(dplyr))
```

Data Import and Structure

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/4360e50239b4eb6b22f3a1759323748f367
cases <- "time_series_covid19_confirmed_global.csv"
deaths <- "time_series_covid19_deaths_global.csv"
recovered <- "time_series_covid19_recovered_global.csv"

global_cases <- read.csv(paste0(url_in, cases))
global_deaths <- read.csv(paste0(url_in, deaths))
global_recovered <- read.csv(paste0(url_in, recovered))


print(colnames(global_cases)[1:10])
```

```
##  [1] "Province.State" "Country.Region" "Lat"            "Long"
##  [5] "X1.22.20"       "X1.23.20"       "X1.24.20"       "X1.25.20"
##  [9] "X1.26.20"       "X1.27.20"
```

```r
print(colnames(global_deaths)[1:10])
```

```
##  [1] "Province.State" "Country.Region" "Lat"            "Long"
##  [5] "X1.22.20"       "X1.23.20"       "X1.24.20"      "X1.25.20"
##  [9] "X1.26.20"       "X1.27.20"
```

```r
print(colnames(global_recovered)[1:10])
```

```
##  [1] "Province.State" "Country.Region" "Lat"            "Long"
##  [5] "X1.22.20"       "X1.23.20"       "X1.24.20"      "X1.25.20"
##  [9] "X1.26.20"       "X1.27.20"
```

Originaly there are 3 dataframes: - global_cases - global_deaths - global_recovered Each dataframe has the following columns: - Province.State - Country.Region - Lat - Long - Series of columns with Date as header, each column contains the number of cases, deaths or recovered for that date.

Tidy and Transforming Data

Data Cleaning

We will omit Lat and Long columns.
The first two columns are factors.
To streamline the analysis we will pivot the dataframes to long format.
So each row will have data for one date, one country and one province/state.

```r
global_cases <- global_cases %>%
  rename_with(~ gsub("^X", "", .), starts_with("X"))
global_deaths <- global_deaths %>%
    rename_with(~ gsub("^X", "", .), starts_with("X"))
global_recovered <- global_recovered %>%
    rename_with(~ gsub("^X", "", .), starts_with("X"))

global_cases <- global_cases %>%
  select(-c(Lat, Long)) %>%
  pivot_longer(cols = -c(Province.State, Country.Region), names_to = "Date", values_to = "Cases") %>%
  mutate(Date = mdy(Date),
         Province.State = as.factor(Province.State),
         Country.Region = as.factor(Country.Region))

global_deaths <- global_deaths %>%
  select(-c(Lat, Long)) %>%
  pivot_longer(cols = -c(Province.State, Country.Region), names_to = "Date", values_to = "Deaths") %>%
  mutate(Date = mdy(Date),
         Province.State = as.factor(Province.State),
         Country.Region = as.factor(Country.Region))

global_recovered <- global_recovered %>%
  select(-c(Lat, Long)) %>%
  pivot_longer(cols = -c(Province.State, Country.Region), names_to = "Date", values_to = "Recovered") %
  mutate(Date = mdy(Date),
         Province.State = as.factor(Province.State),
```

```
         Country.Region = as.factor(Country.Region))

print(head(global_cases))
```

```
## # A tibble: 6 x 4
##   Province.State Country.Region Date       Cases
##   <fct>          <fct>          <date>     <int>
## 1 ""             Afghanistan    2020-01-22     0
## 2 ""             Afghanistan    2020-01-23     0
## 3 ""             Afghanistan    2020-01-24     0
## 4 ""             Afghanistan    2020-01-25     0
## 5 ""             Afghanistan    2020-01-26     0
## 6 ""             Afghanistan    2020-01-27     0
```

```
# Count NA for data column
print(sum(is.na(global_cases$Cases)))
```

```
## [1] 0
```

```
print(sum(is.na(global_deaths$Deaths)))
```

```
## [1] 0
```

```
print(sum(is.na(global_recovered$Recovered)))
```

```
## [1] 0
```

```
# Count NA for date column
print(sum(is.na(global_cases$Date)))
```

```
## [1] 0
```

```
print(sum(is.na(global_deaths$Date)))
```

```
## [1] 0
```

```
print(sum(is.na(global_recovered$Date)))
```

```
## [1] 0
```

```
# Total number of rows
print(nrow(global_cases))
```

```
## [1] 330327
```

```r
print(nrow(global_deaths))
```

```
## [1] 330327
```

```r
print(nrow(global_recovered))
```

```
## [1] 313182
```

There are no missing values, however dataframe with recovered cases has smaller number of values that the other two.

```r
# Print end and start dates
print(paste0("Start date Cases: ", min(global_cases$Date)))
```

```
## [1] "Start date Cases: 2020-01-22"
```

```r
print(paste0("End date Cases: ", max(global_cases$Date)))
```

```
## [1] "End date Cases: 2023-03-09"
```

```r
print(paste0("Start date Deaths: ", min(global_deaths$Date)))
```

```
## [1] "Start date Deaths: 2020-01-22"
```

```r
print(paste0("End date Deaths: ", max(global_deaths$Date)))
```

```
## [1] "End date Deaths: 2023-03-09"
```

```r
print(paste0("Start date Recovered: ", min(global_recovered$Date)))
```

```
## [1] "Start date Recovered: 2020-01-22"
```

```r
print(paste0("End date Recovered: ", max(global_recovered$Date)))
```

```
## [1] "End date Recovered: 2023-03-09"
```

All the dataframes has the same data range starting from 22 Jan 2020 till 9 March 2023.

```r
print(paste0("Number of countries: ", length(unique(global_cases$Country.Region))))
```

```
## [1] "Number of countries: 201"
```

```r
print(paste0("Number of countries: ", length(unique(global_deaths$Country.Region))))
```

```
## [1] "Number of countries: 201"
```

```
print(paste0("Number of countries: ", length(unique(global_recovered$Country.Region))))
```

```
## [1] "Number of countries: 201"
```

Set of countries has the same number of unique values for each dataframe.

Combining Dataframes

To streamline the analysis we will join the dataframes on columns Country.Region, Province.State and Date.

```
global_data <- global_cases %>%
  left_join(global_deaths, by = c("Country.Region", "Province.State", "Date")) %>%
  left_join(global_recovered, by = c("Country.Region", "Province.State", "Date"))
print(head(global_data))
```

```
## # A tibble: 6 x 6
##   Province.State Country.Region Date       Cases Deaths Recovered
##   <fct>          <fct>          <date>     <int>  <int>     <int>
## 1 ""             Afghanistan    2020-01-22     0      0         0
## 2 ""             Afghanistan    2020-01-23     0      0         0
## 3 ""             Afghanistan    2020-01-24     0      0         0
## 4 ""             Afghanistan    2020-01-25     0      0         0
## 5 ""             Afghanistan    2020-01-26     0      0         0
## 6 ""             Afghanistan    2020-01-27     0      0         0
```

Now each row represents a date, country and province/state and number of cases, deaths and recovered.

```
print(paste0("Number of rows: ", nrow(global_data)))
```

```
## [1] "Number of rows: 330327"
```

```
print(paste0("Number of NA in Country.Region: ", sum(is.na(global_data$Country.Region))))
```

```
## [1] "Number of NA in Country.Region: 0"
```

```
print(paste0("Number of NA in Province.State: ", sum(is.na(global_data$Province.State))))
```

```
## [1] "Number of NA in Province.State: 0"
```

```
print(paste0("Number of NA in Date: ", sum(is.na(global_data$Date))))
```

```
## [1] "Number of NA in Date: 0"
```

```
print(paste0("Number of NA in Cases: ", sum(is.na(global_data$Cases))))
```

```
## [1] "Number of NA in Cases: 0"
```

```r
print(paste0("Number of NA in Deaths: ", sum(is.na(global_data$Deaths))))
```

```
## [1] "Number of NA in Deaths: 0"
```

```r
print(paste0("Number of NA in Recovered: ", sum(is.na(global_data$Recovered))))
```

```
## [1] "Number of NA in Recovered: 18288"
```

The only column wits missing values so far is Recovered.
We will deal with it later.

```r
print(paste0("Number of unique values in Country.Region: ", length(unique(global_data$Country.Region))))
```

```
## [1] "Number of unique values in Country.Region: 201"
```

```r
print(paste0("Number of unique values in Province.State: ", length(unique(global_data$Province.State))))
```

```
## [1] "Number of unique values in Province.State: 92"
```

```r
# number of occurances of each value sorted from most to least frequent
print(global_data %>%
  group_by(Country.Region) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(10))
```

```
## # A tibble: 10 x 2
##    Country.Region     n
##    <fct>          <int>
##  1 China          38862
##  2 Canada         18288
##  3 United Kingdom 17145
##  4 France         13716
##  5 Australia       9144
##  6 Netherlands     5715
##  7 Denmark         3429
##  8 New Zealand     3429
##  9 Afghanistan     1143
## 10 Albania         1143
```

```r
print(global_data %>%
    group_by(Province.State) %>%
    summarise(n = n()) %>%
    arrange(desc(n)) %>%
    head(10))
```

```
## # A tibble: 10 x 2
##    Province.State                    n
##    <fct>                         <int>
```

```
##  1 ""                                  226314
##  2 "Alberta"                             1143
##  3 "Anguilla"                            1143
##  4 "Anhui"                               1143
##  5 "Aruba"                               1143
##  6 "Australian Capital Territory"        1143
##  7 "Beijing"                             1143
##  8 "Bermuda"                             1143
##  9 "Bonaire, Sint Eustatius and Saba"    1143
## 10 "British Columbia"                    1143
```

There are only a few counties with enlarged ammount of entries, which is related to the fact that for them Provincies were taking into account. We will combine the data for each country and remove Province.State column.

```
global_data <- global_data %>%
  group_by(Country.Region, Date) %>%
  summarise(Cases = sum(Cases, na.rm = TRUE),
            Deaths = sum(Deaths, na.rm = TRUE),
            Recovered = sum(Recovered, na.rm = TRUE)) %>%
  ungroup()
```

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

```
print(head(global_data))
```

```
## # A tibble: 6 x 5
##   Country.Region Date       Cases Deaths Recovered
##   <fct>          <date>     <int>  <int>     <int>
## 1 Afghanistan    2020-01-22     0      0         0
## 2 Afghanistan    2020-01-23     0      0         0
## 3 Afghanistan    2020-01-24     0      0         0
## 4 Afghanistan    2020-01-25     0      0         0
## 5 Afghanistan    2020-01-26     0      0         0
## 6 Afghanistan    2020-01-27     0      0         0
```

```
print(paste0("Number of rows: ", nrow(global_data)))
```

```
## [1] "Number of rows: 229743"
```

Summary

```
str(global_data)
```

```
## tibble [229,743 x 5] (S3: tbl_df/tbl/data.frame)
##  $ Country.Region: Factor w/ 201 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date          : Date[1:229743], format: "2020-01-22" "2020-01-23" ...
##  $ Cases         : int [1:229743] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Deaths        : int [1:229743] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Recovered     : int [1:229743] 0 0 0 0 0 0 0 0 0 0 ...
```

```
summary(global_data)
```

```
##      Country.Region        Date                Cases
##  Afghanistan:  1143   Min.   :2020-01-22   Min.   :        0
##  Albania    :  1143   1st Qu.:2020-11-02   1st Qu.:     3831
##  Algeria    :  1143   Median :2021-08-15   Median :    52933
##  Andorra    :  1143   Mean   :2021-08-15   Mean   :  1379412
##  Angola     :  1143   3rd Qu.:2022-05-28   3rd Qu.:   499592
##  Antarctica :  1143   Max.   :2023-03-09   Max.   :103802702
##  (Other)    :222885
##      Deaths          Recovered
##  Min.   :      0   Min.   :      -1
##  1st Qu.:     46   1st Qu.:       0
##  Median :    786   Median :       0
##  Mean   :  19238   Mean   :  101101
##  3rd Qu.:   7227   3rd Qu.:    3564
##  Max.   :1123836   Max.   :30974748
##
```

Now there is set ammount of data for each country.
Recovered has surprising -1 value as min, and median is 0 so this is rather non informative column. Max cases is rather big, we can take a look on it:

```
print(global_data %>%
  filter(Cases == max(Cases, na.rm = TRUE)) %>%
  select(Country.Region, Date, Cases, Deaths, Recovered))
```

```
## # A tibble: 1 x 5
##   Country.Region Date           Cases  Deaths Recovered
##   <fct>          <date>         <int>   <int>     <int>
## 1 US             2023-03-09 103802702 1123836         0
```

It is in allignment with the data from the US.

Joining population data

For further analysis it would be important to have total population for each country.
We will join the data with population data from the same source.

```
table_with_population <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/4360e50239
print(head(table_with_population))
```

```
##   UID iso2 iso3 code3 FIPS Admin2 Province_State Country_Region       Lat
## 1   4   AF  AFG     4   NA                          Afghanistan  33.93911
## 2   8   AL  ALB     8   NA                              Albania  41.15330
## 3  10   AQ  ATA    10   NA                           Antarctica -71.94990
## 4  12   DZ  DZA    12   NA                              Algeria  28.03390
## 5  20   AD  AND    20   NA                              Andorra  42.50630
## 6  24   AO  AGO    24   NA                               Angola -11.20270
##      Long_ Combined_Key Population
```

```
## 1 67.70995  Afghanistan   38928341
## 2 20.16830      Albania    2877800
## 3 23.34700   Antarctica         NA
## 4  1.65960      Algeria   43851043
## 5  1.52180      Andorra      77265
## 6 17.87390       Angola   32866268
```

```
# only need 'Country_Region', 'Population'
table_with_population <- table_with_population %>%
  select(Country_Region, Population) %>%
  rename(Country.Region = Country_Region)
print(head(table_with_population))
```

```
##   Country.Region Population
## 1    Afghanistan   38928341
## 2        Albania    2877800
## 3     Antarctica         NA
## 4        Algeria   43851043
## 5        Andorra      77265
## 6         Angola   32866268
```

As we only need country and its population - rest can be ommited.

```
table_with_population <- table_with_population %>%
  group_by(Country.Region) %>%
  summarise(Population = sum(Population, na.rm = TRUE)) %>%
  ungroup()

print(table_with_population %>%
  arrange(Population) %>%
  head(50))
```

```
## # A tibble: 50 x 2
##    Country.Region       Population
##    <chr>                     <dbl>
##  1 Antarctica                    0
##  2 Diamond Princess              0
##  3 MS Zaandam                    0
##  4 Summer Olympics 2020          0
##  5 Winter Olympics 2022          0
##  6 Holy See                    809
##  7 Nauru                     10834
##  8 Tuvalu                    11792
##  9 Palau                     18008
## 10 San Marino                33938
## # i 40 more rows
```

There are countries with 0 population.
We will also remove countries with population less than 1 million as it creates a lot of noise in the data.

```
# just join the data
global_data <- global_data %>%
```

```
  left_join(table_with_population, by = "Country.Region")

# get read of countries with Population < 1000000
global_data <- global_data %>%
  filter(Population > 1000000)

print(head(global_data))
```

```
## # A tibble: 6 x 6
##   Country.Region Date       Cases Deaths Recovered Population
##   <chr>          <date>     <int>  <int>     <int>      <dbl>
## 1 Afghanistan    2020-01-22     0      0         0   38928341
## 2 Afghanistan    2020-01-23     0      0         0   38928341
## 3 Afghanistan    2020-01-24     0      0         0   38928341
## 4 Afghanistan    2020-01-25     0      0         0   38928341
## 5 Afghanistan    2020-01-26     0      0         0   38928341
## 6 Afghanistan    2020-01-27     0      0         0   38928341
```

Summary on Tidy and Transforming Data

The data was combined and brought to pivot format.
Necessary data transformation were performed: Country to factor, Date to date.
Redundant columns (Lat, Long) were removed.
Data was combined over provinces/states and this column was removed as well.
Population data was joined to the data. Countries with small population were removed from consideration.
The data is now ready for analysis.

Visualising & Analyzing

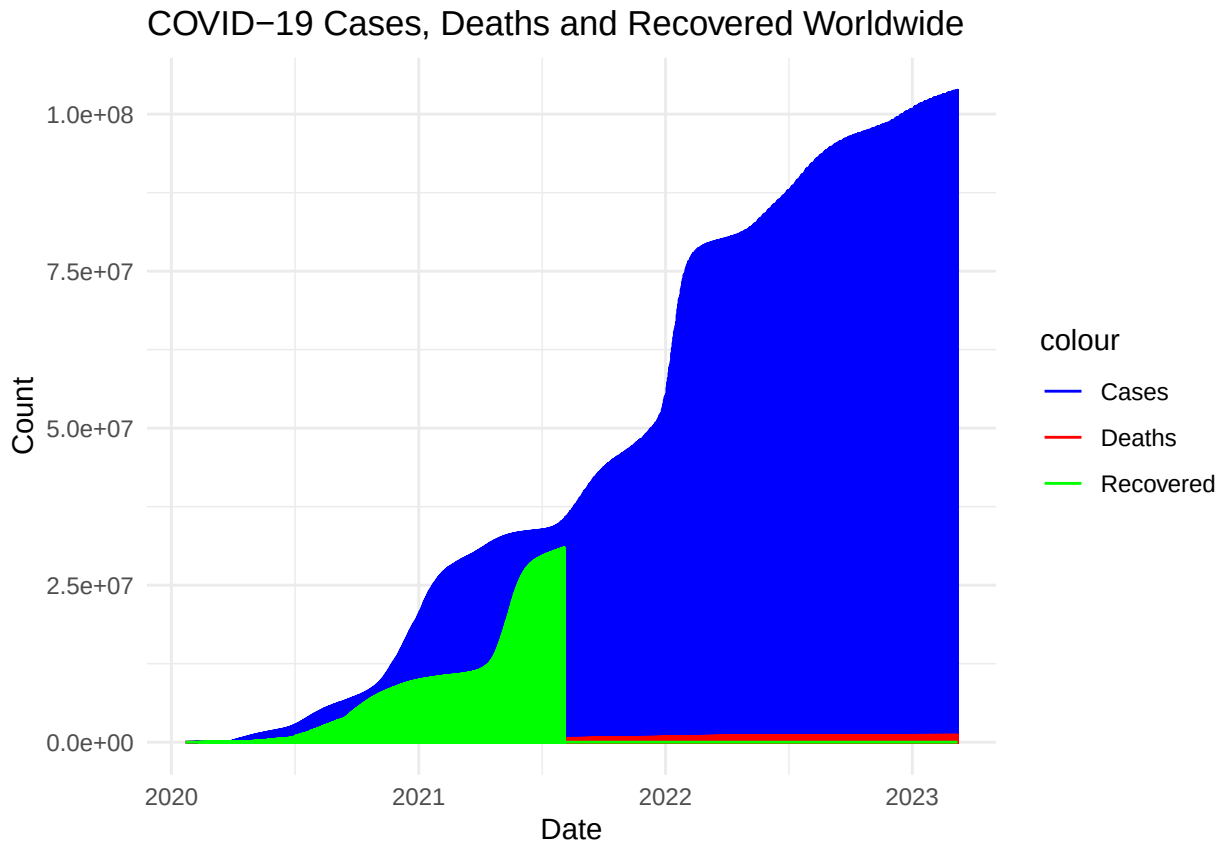
Worldwide Data over time

First lets' take a look on worldwide data.

```
df_combined_world <- global_data %>%
  group_by(Date) %>%
  summarise(Cases = sum(Cases, na.rm = TRUE),
            Deaths = sum(Deaths, na.rm = TRUE),
            Recovered = sum(Recovered, na.rm = TRUE)) %>%
  ungroup()
print(head(df_combined_world)[, 1:2])
```

```
## # A tibble: 6 x 2
##   Date       Cases
##   <date>     <int>
## 1 2020-01-22   557
## 2 2020-01-23   657
## 3 2020-01-24   944
## 4 2020-01-25  1437
## 5 2020-01-26  2120
## 6 2020-01-27  2929
```

```r
ggplot(global_data, aes(x = Date)) +
  geom_line(aes(y = Cases, color = "Cases")) +
  geom_line(aes(y = Deaths, color = "Deaths")) +
  geom_line(aes(y = Recovered, color = "Recovered")) +
  labs(title = "COVID-19 Cases, Deaths and Recovered Worldwide",
       x = "Date",
       y = "Count") +
  scale_color_manual(values = c("Cases" = "blue", "Deaths" = "red", "Recovered" = "green")) +
  theme_minimal()
```



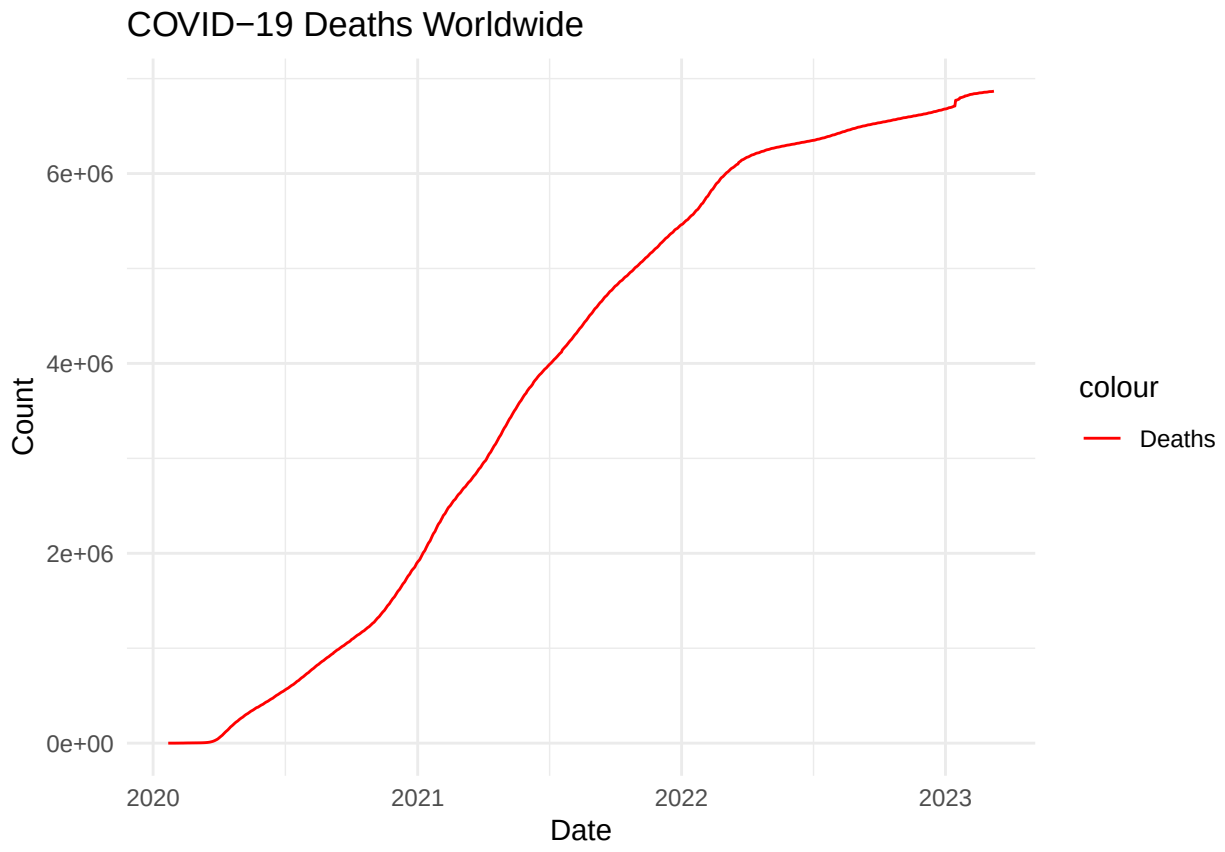
It is clear that data on number of recovered cases is not reliable.
This data was stopped to be collected at some point and there is a huge difference between sum of deaths and recovered vs number of cases. So, we will not use this data for further analysis.

```r
global_data  <- global_data  %>%
  select(-Recovered)
df_combined_world <- df_combined_world %>%
  select(-Recovered)
```

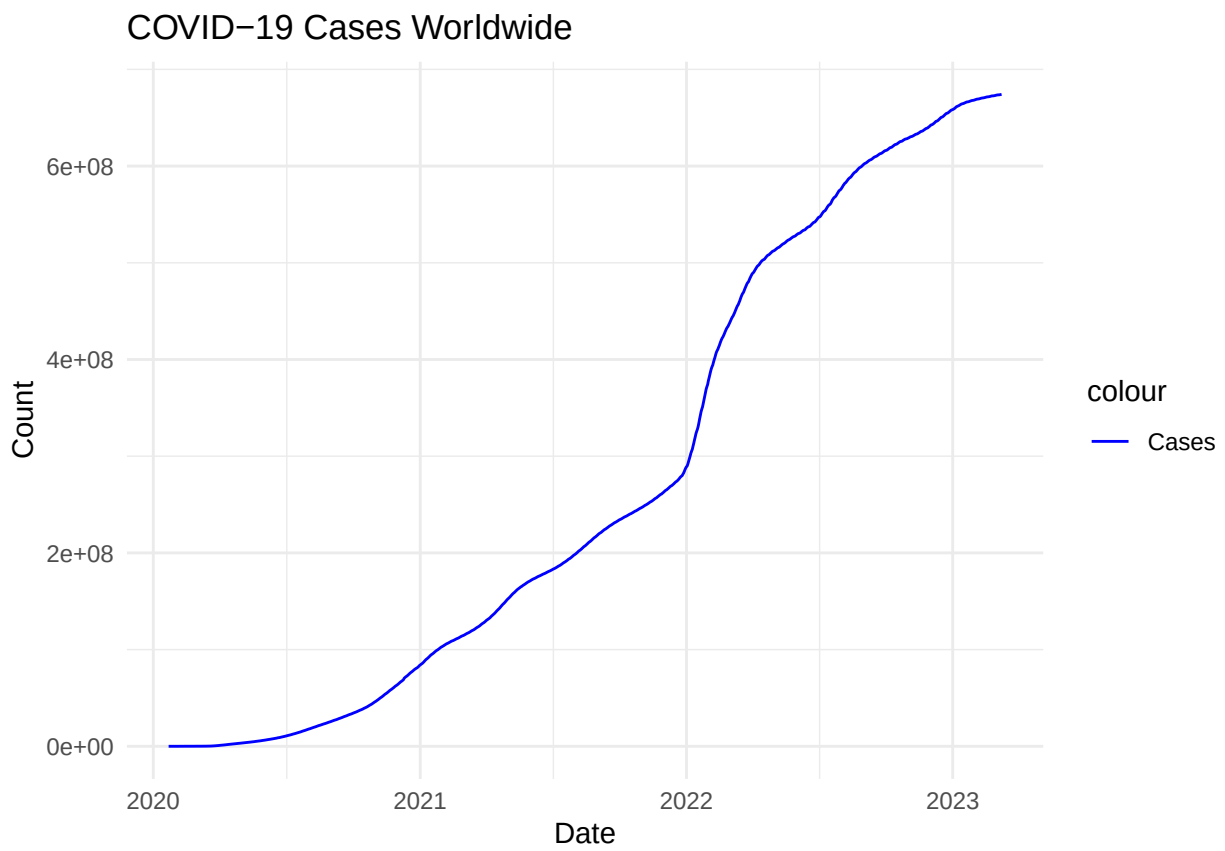
Plotting cases and deaths separately to see the trends.

```r
ggplot(df_combined_world, aes(x = Date)) +
  geom_line(aes(y = Deaths, color = "Deaths")) +
  labs(title = "COVID-19 Deaths Worldwide",
       x = "Date",
```

```
       y = "Count") +
  scale_color_manual(values = c("Deaths" = "red")) +
  theme_minimal()
```




```
ggplot(df_combined_world, aes(x = Date)) +
  geom_line(aes(y = Cases, color = "Cases")) +
  labs(title = "COVID-19 Cases Worldwide",
       x = "Date",
       y = "Count") +
  scale_color_manual(values = c("Cases" = "blue")) +
  theme_minimal()
```



Both total number of cases and deaths are increasing over time.
However it is clarly visible that the number of deaths is not increasing as fast as the number of cases after the beggining of the pandemic.

Deaths and Cases relation

To get a better understanding we can take a look on number of new cases and deaths per day.

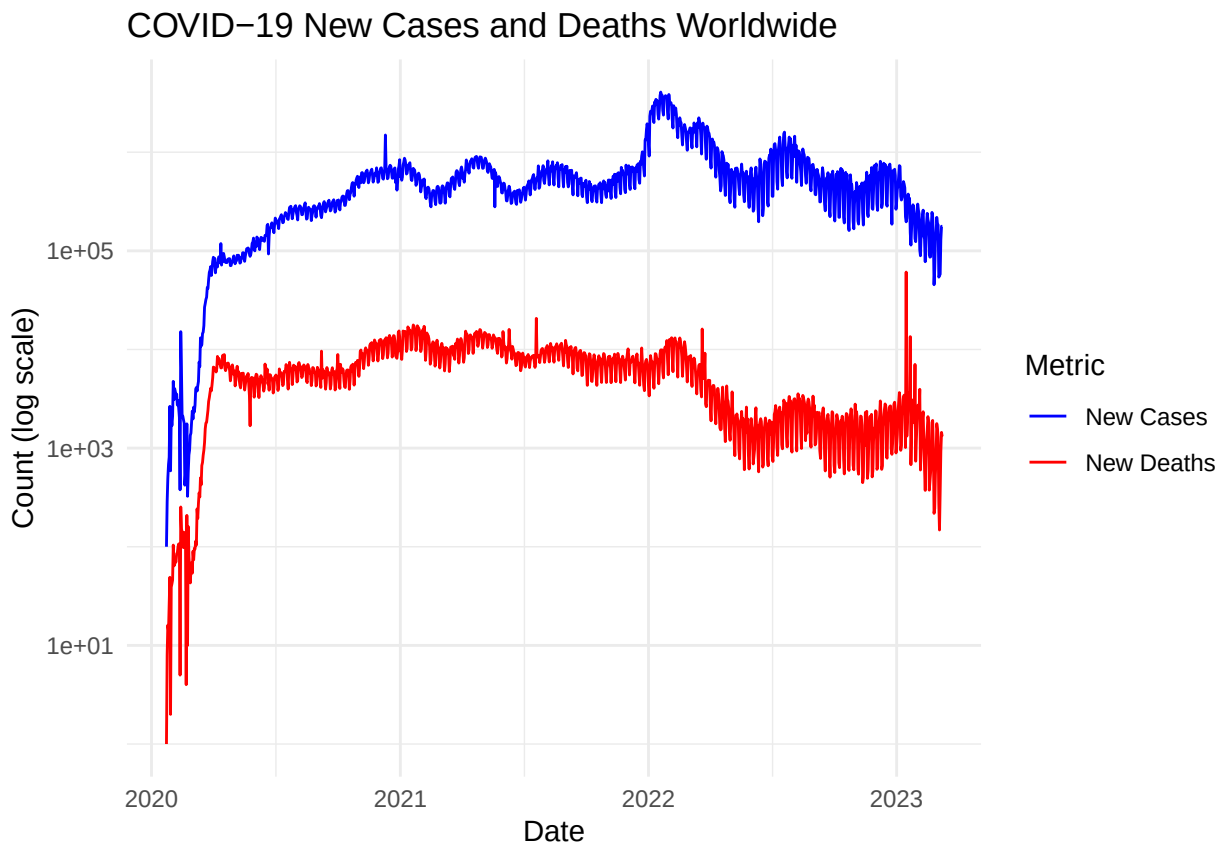
```
df_combined_world <- df_combined_world %>%
  mutate(New.Cases = Cases - lag(Cases, 1),
         New.Deaths = Deaths - lag(Deaths, 1))
global_data <- global_data %>%
  mutate(New.Cases = Cases - lag(Cases, 1),
         New.Deaths = Deaths - lag(Deaths, 1))
```

Logarithmic scale is used to see the trends better.

```
ggplot(df_combined_world, aes(x = Date)) +
  geom_line(aes(y = New.Cases, color = "New Cases")) +
  geom_line(aes(y = New.Deaths, color = "New Deaths")) +
  labs(title = "COVID-19 New Cases and Deaths Worldwide",
       x = "Date",
       y = "Count (log scale)",
       color = "Metric") +
  scale_color_manual(values = c("New Cases" = "blue", "New Deaths" = "red")) +
```

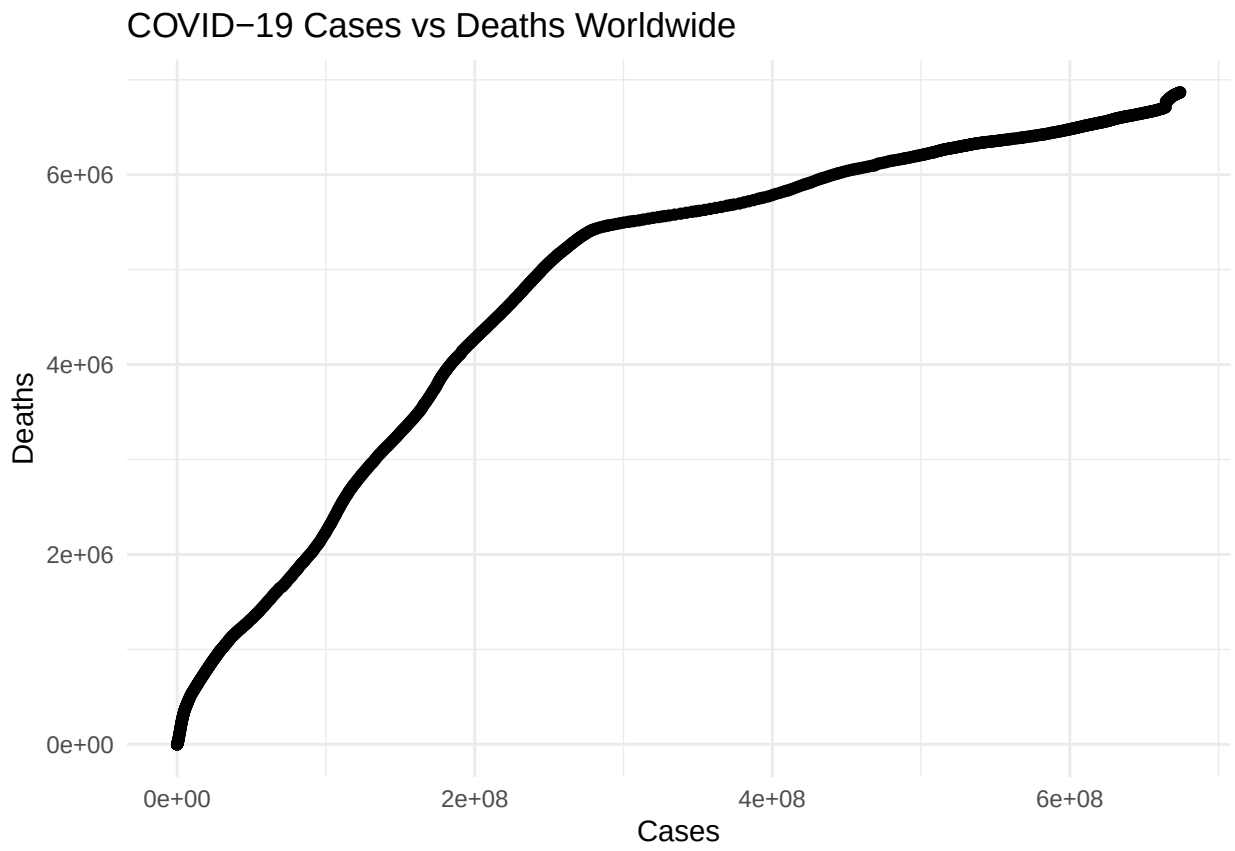
```
  theme_minimal() +
  scale_y_log10()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



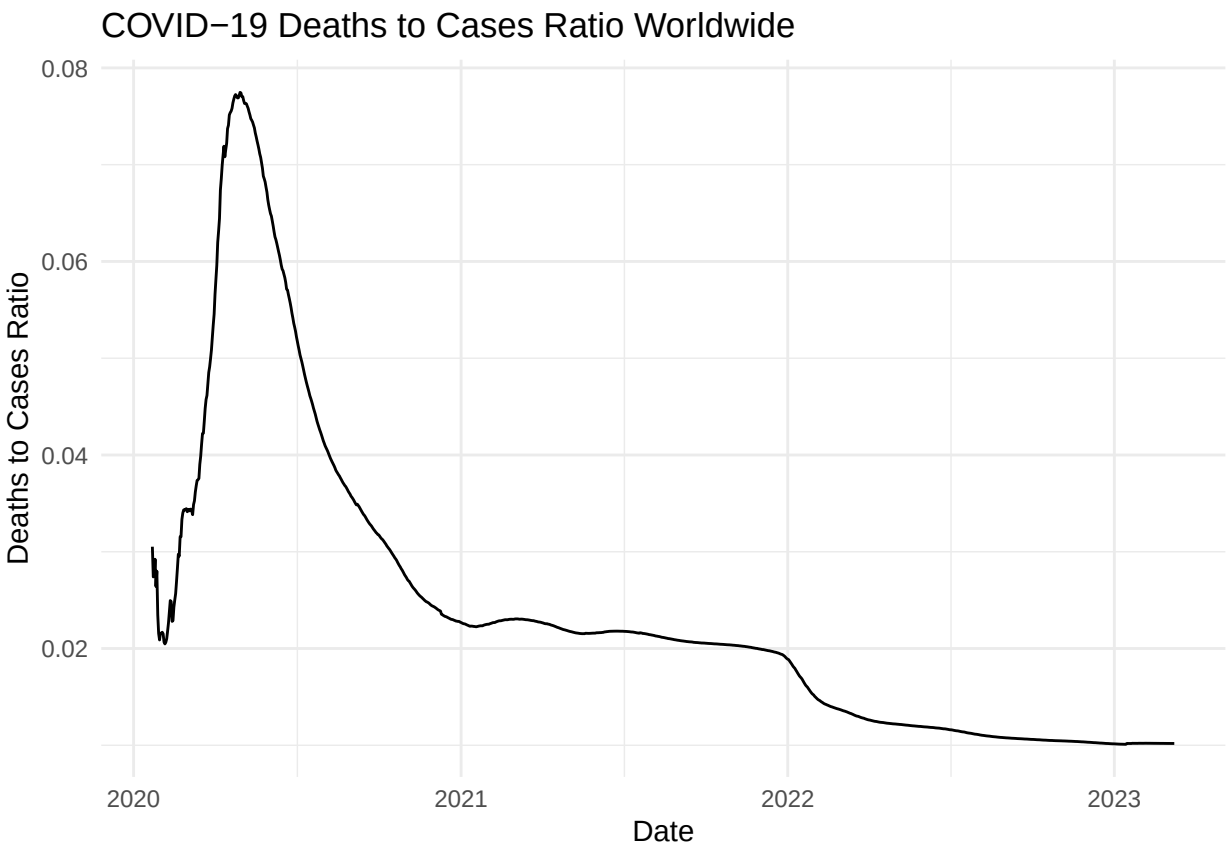
The number of both new cases and deaths has stabilized after spring 2020 and was constant for the next 3 years.

```
ggplot(df_combined_world, aes(x = Cases, y = Deaths)) +
  geom_point() +
  labs(title = "COVID-19 Cases vs Deaths Worldwide",
       x = "Cases",
       y = "Deaths") +
  theme_minimal()
```



The change of trend of relation of number of deaths and cases is clearly visible.

```
df_combined_world$deaths_to_cases <- df_combined_world$Deaths / df_combined_world$Cases
ggplot(df_combined_world, aes(x = Date, y = deaths_to_cases)) +
  geom_line() +
  labs(title = "COVID-19 Deaths to Cases Ratio Worldwide",
       x = "Date",
       y = "Deaths to Cases Ratio") +
  theme_minimal()
```



There are 3 distinct periods in the data 1. First period is from the beginning of the pandemic till the end of 2020.
The number of deaths to cases ratio is high and unstable.
2. Second period is from the beginning of 2021 till the end of 2021.
The number of deaths to cases ratio is decreasing and is around 0.02. 3. Third period is from the beginning of 2022 onwards.
The number of deaths to cases ratio has decreased significantly and goes to 0.

The first period is likely related to the fact that the virus was new and there was no vaccine.
Not only methods of treatment were not known, but also the virus was not well studied.
In addition, limited supply of testing equipement would bring high noise to the data.

The second period is likely related to the fact that the vaccine was introduced and the virus was better studied. So, it is probably close to the real ratio of deaths to cases.

While the time from 2022 onwords may be related to both advances in treatment and evolution of the virus.
The virus may have mutated and became less deadly, but more contagious.
This is also in agremeent with spike in number of cases in the beggining of 2022.

Data by countries

Lets' take a look on number of cases in different countries.

```
# Get the data for the top 10 countries by number of cases and plot number of new cases over time
top_10_countries <- global_data %>%
  group_by(Country.Region) %>%
  summarise(Total.Cases = max(Cases)) %>%
```

```
  arrange(desc(Total.Cases)) %>%
  slice(1:10)
print(top_10_countries)
```

```
## # A tibble: 10 x 2
##    Country.Region Total.Cases
##    <chr>                <int>
##  1 US                103802702
##  2 India              44690738
##  3 France             39866718
##  4 Germany            38249060
##  5 Brazil             37081209
##  6 Japan              33320438
##  7 Korea, South       30615522
##  8 Italy              25603510
##  9 United Kingdom     24658705
## 10 Russia             22075858
```

The same with relative number of cases to population.

```
# Get the data for the top 10 countries by number of cases and plot number of new cases over time
global_data$relative_cases <- global_data$Cases / global_data$Population
top_10_countries <- global_data %>%
  group_by(Country.Region) %>%
  summarise(Total.Cases = max(relative_cases)) %>%
  arrange(desc(Total.Cases)) %>%
  slice(1:10)
print(top_10_countries)
```

```
## # A tibble: 10 x 2
##    Country.Region Total.Cases
##    <chr>                <dbl>
##  1 Austria              0.662
##  2 Slovenia             0.641
##  3 Korea, South         0.597
##  4 France               0.585
##  5 Denmark              0.581
##  6 Israel               0.555
##  7 Portugal             0.546
##  8 Cyprus               0.539
##  9 Greece               0.532
## 10 Latvia               0.518
```

```
# same bottom 10
bottom_10_countries <- global_data %>%
  group_by(Country.Region) %>%
  summarise(Total.Cases = max(relative_cases)) %>%
  arrange(Total.Cases) %>%
  slice(1:10)
print(bottom_10_countries)
```

```
## # A tibble: 10 x 2
```

```
##    Country.Region    Total.Cases
##    <chr>                   <dbl>
##  1 Korea, North     0.0000000388
##  2 Niger            0.000393
##  3 Yemen            0.000400
##  4 Chad             0.000467
##  5 Nigeria          0.000667
##  6 Tanzania         0.000718
##  7 Sierra Leone     0.000973
##  8 Burkina Faso     0.00106
##  9 Congo (Kinshasa) 0.00107
## 10 Sudan            0.00146
```

The highest number of cases is in the US.
However, the relative number of cases is more representative.
For Austria it is as much as 2/3 of population!

Another important observation is the bias of data.
Countries with the most cases per population are relativly rich. While those with the least cases are poor.
Which is more likely related to efficiency of testing and reporting other then real number of cases.

Summary on Data Analysis

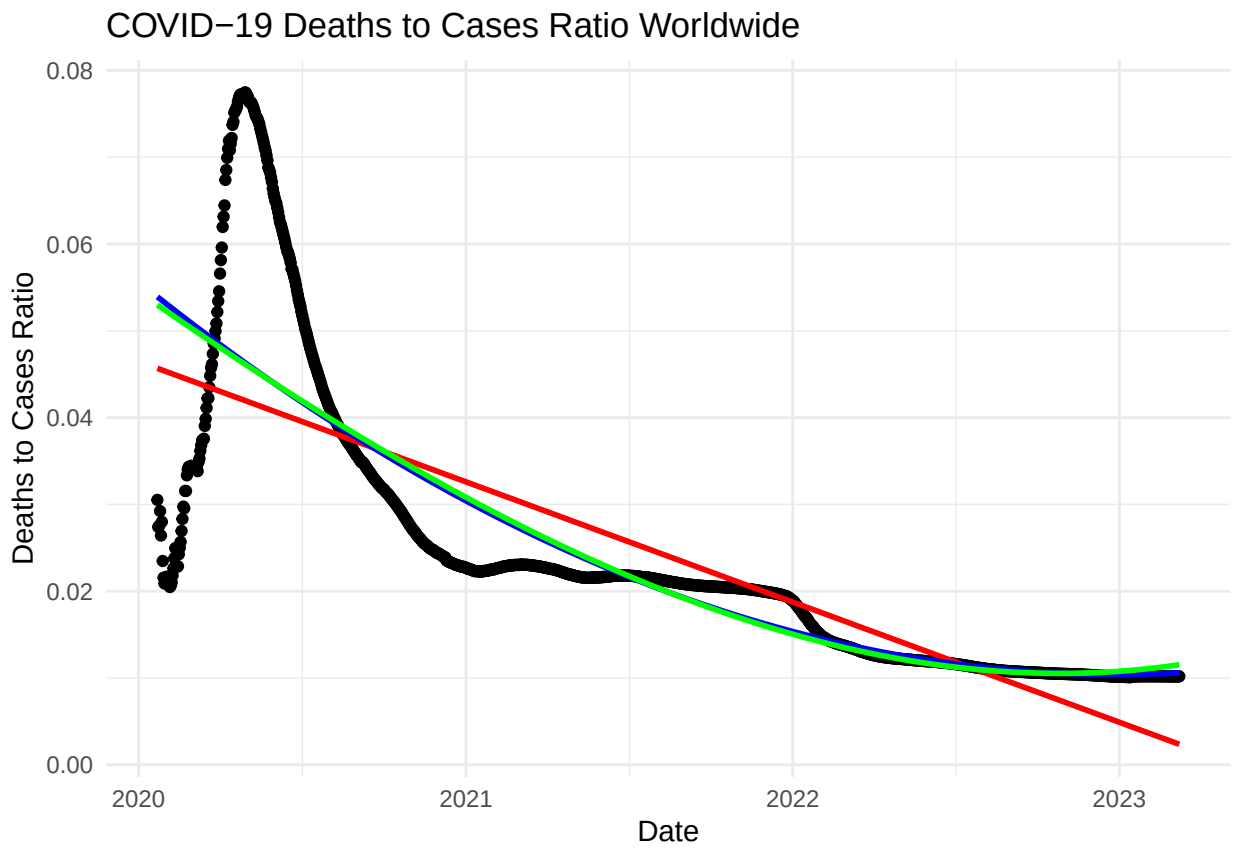
Data analysis allowed to take a look on the data from different angles.
First of all it was concluded that the data on recovered cases is not reliable.
New number of cases and deaths appered to be stable over time in exception of the initial period.
Intrigingly, the number of deaths to cases ratio over time has 3 distinct periods. The first period is from the beginning of the pandemic till the end of 2020 with high and unstable ratio. The second period is from the beginning of 2021 till the end of 2021 with ratio of deaths around 2%. The third period is from the beginning of 2022 onwards with decreasing ratio of deaths to cases going to 0. Country specific analysis is is not reliable due to data bias.

Modeling Data

To support our findings about the deaths to cases ratio we can study modeling for this data.

```
# Fit a linear model to the data
model <- lm(deaths_to_cases ~ Date, data = df_combined_world)
model_poly2 <- lm(deaths_to_cases ~ poly(Date, 2), data = df_combined_world)
model_poly3 <- lm(deaths_to_cases ~ poly(Date, 3), data = df_combined_world)

ggplot(df_combined_world, aes(x = Date, y = deaths_to_cases)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE, color = "blue") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 3), se = FALSE, color = "green") +
  labs(title = "COVID-19 Deaths to Cases Ratio Worldwide",
       x = "Date",
       y = "Deaths to Cases Ratio") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

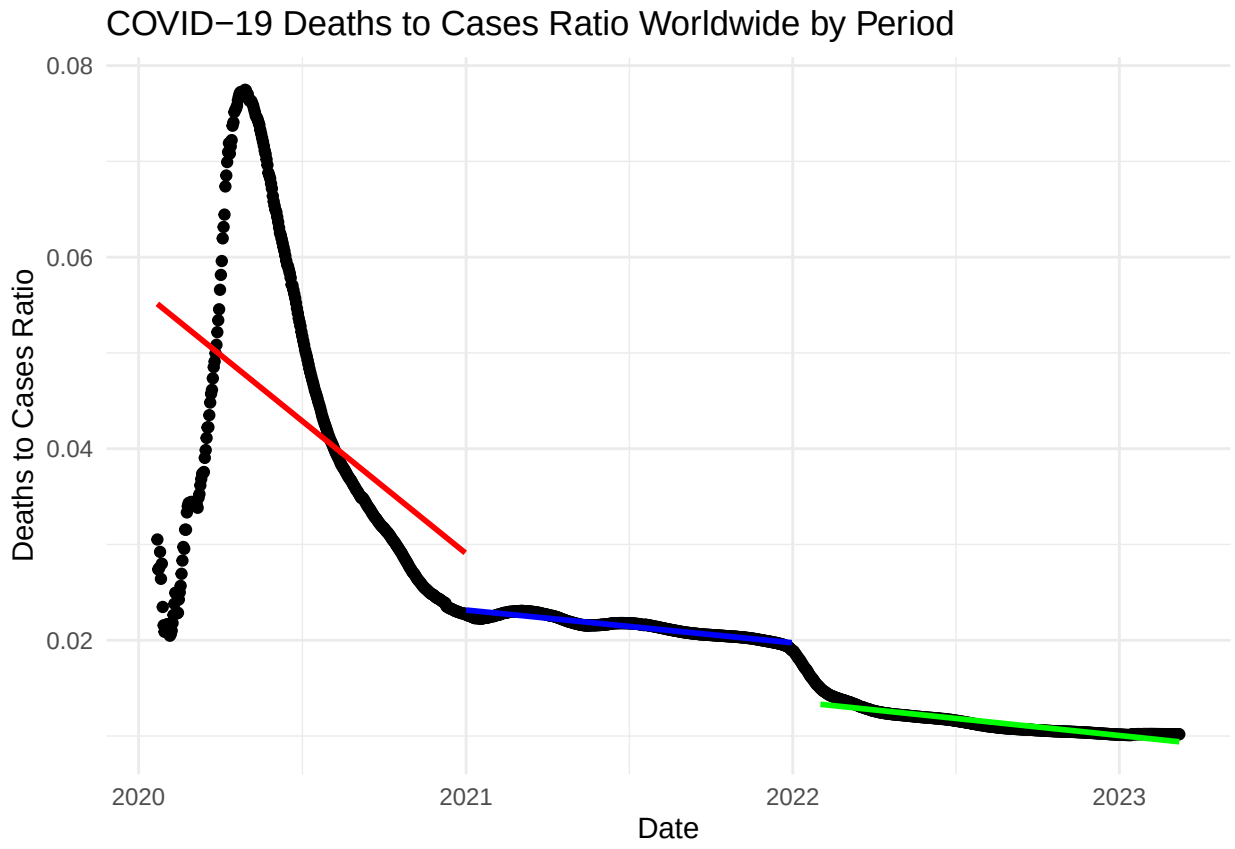


It can be claerly seen that feeting while time period does not work properly.

```
period1 <- df_combined_world %>%
  filter(Date < as.Date("2021-01-01"))

period2 <- df_combined_world %>%
  filter(Date >= as.Date("2021-01-01") & Date < as.Date("2022-01-01"))

period3 <- df_combined_world %>%
  filter(Date >= as.Date("2022-02-01"))

model1 <- lm(deaths_to_cases ~ Date, data = period1)
model2 <- lm(deaths_to_cases ~ Date, data = period2)
model3 <- lm(deaths_to_cases ~ Date, data = period3)

ggplot() +
  geom_point(data = df_combined_world, aes(x = Date, y = deaths_to_cases)) +
  geom_smooth(data = period1, aes(x = Date, y = deaths_to_cases),
              method = "lm", se = FALSE, color = "red") +
  geom_smooth(data = period2, aes(x = Date, y = deaths_to_cases),
              method = "lm", se = FALSE, color = "blue") +
  geom_smooth(data = period3, aes(x = Date, y = deaths_to_cases),
              method = "lm", se = FALSE, color = "green") +
  labs(title = "COVID-19 Deaths to Cases Ratio Worldwide by Period",
       x = "Date",
       y = "Deaths to Cases Ratio") +
```

```
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Aside from first highly fluctuating period, the data is well fitted by linear models.

```
summary(model)
```

```
##
## Call:
## lm(formula = deaths_to_cases ~ Date, data = df_combined_world)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.024674 -0.005232 -0.001563  0.002445  0.035515
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.392e-01  1.648e-02   44.87   <2e-16 ***
## Date        -3.793e-05  8.737e-07  -43.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.009747 on 1141 degrees of freedom
## Multiple R-squared:  0.6229, Adjusted R-squared:  0.6226
## F-statistic:  1885 on 1 and 1141 DF,  p-value: < 2.2e-16
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = deaths_to_cases ~ Date, data = period1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.033638 -0.007006 -0.003284  0.010098  0.029769
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.437e+00  1.574e-01   9.128   <2e-16 ***
## Date        -7.557e-05  8.529e-06  -8.860   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01578 on 343 degrees of freedom
## Multiple R-squared:  0.1863, Adjusted R-squared:  0.1839
## F-statistic: 78.51 on 1 and 343 DF,  p-value: < 2.2e-16
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = deaths_to_cases ~ Date, data = period2)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -7.799e-04 -1.569e-04  1.690e-06  2.475e-04  5.183e-04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.978e-01  2.820e-03   70.13   <2e-16 ***
## Date        -9.374e-06  1.499e-07  -62.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0003018 on 363 degrees of freedom
## Multiple R-squared:  0.915,  Adjusted R-squared:  0.9148
## F-statistic:  3910 on 1 and 363 DF,  p-value: < 2.2e-16
```

```
summary(model3)
```

```
##
## Call:
## lm(formula = deaths_to_cases ~ Date, data = period3)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0004852 -0.0002736 -0.0001971  0.0001709  0.0016683
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.989e-01  3.593e-03   55.36   <2e-16 ***
## Date        -9.755e-06  1.869e-07  -52.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0004348 on 400 degrees of freedom
## Multiple R-squared:  0.872,  Adjusted R-squared:  0.8717
## F-statistic:  2725 on 1 and 400 DF,  p-value: < 2.2e-16
```

Bias

Bias from Data

The data is collected from different sources from different countries.
Moreover the data on COVID-19 was of great political importance at the time it was collected. Thus, extreme influence of country of origin can be anticipated.
Another bias source is lack of testing equipement in some countries at various times.

Personal Bias

I have witnessed the COVID-19 pandemic and it has affected my life and took lives of people I know.
Thus I have strong feelings about the data and the pandemic.
It is not affecting preliminary stages of analysis, but may influence the conclusions.

Conclusion

This project is dedicated to analysis of COVID-19 cases data provided by Johns Hopkins University.
Original data sources mostly country specific which leads to high bias, thsu the focus of this project is on the global data.

On the prepartion step data was cleaned and transformed.
Prodcuedre resulted combined pivoted dataframe with columns: - Country.Region - Date - Cases - Deaths - Population

During the exploratory data analysis it was concluded that the data on recovered cases is not reliable. New number of cases and deaths appered to be stable over time in exception of the initial period. Intrigingly, the number of deaths to cases ratio over time has 3 distinct periods. The first period is from the beginning of the pandemic till the end of 2020 with high and unstable ratio. The second period is from the beginning of 2021 till the end of 2021 with ratio of deaths around 2%. The third period is from the beginning of 2022 onwards with decreasing ratio of deaths to cases going to 0.

This was additionaly supported by modeling the data with linear models over whole time and selected periods.

It must be noted that the data is biased due to the fact that it was collected from different sources from different countries and the topic itself was of great political importance.
Wealthy countries with better testing and reporting systems are overrepresented in the data.

P.S.

Thank you for reading this report.

```
sessionInfo()
```

```
## R version 4.4.3 (2025-02-28)
## Platform: aarch64-apple-darwin23.6.0
## Running under: macOS Sequoia 15.3.2
##
## Matrix products: default
## BLAS:   /opt/homebrew/Cellar/openblas/0.3.29/lib/libopenblasp-r0.3.29.dylib
## LAPACK: /opt/homebrew/Cellar/r/4.4.3_1/lib/R/lib/libRlapack.dylib;  LAPACK version 3.12.0
##
## 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/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_1.1.4     lubridate_1.9.4 tidyr_1.3.1     ggplot2_3.5.1
##
## loaded via a namespace (and not attached):
##  [1] Matrix_1.7-2      gtable_0.3.6      compiler_4.4.3    tidyselect_1.2.1
##  [5] splines_4.4.3     scales_1.3.0      yaml_2.3.10       fastmap_1.2.0
##  [9] lattice_0.22-6    R6_2.6.1          labeling_0.4.3    generics_0.1.3
## [13] knitr_1.50        tibble_3.2.1      munsell_0.5.1     pillar_1.10.1
## [17] rlang_1.1.5       utf8_1.2.4        xfun_0.51         timechange_0.3.0
## [21] cli_3.6.4         withr_3.0.2       magrittr_2.0.3    mgcv_1.9-1
## [25] digest_0.6.37     grid_4.4.3        rstudioapi_0.17.1 lifecycle_1.0.4
## [29] nlme_3.1-167      vctrs_0.6.5       evaluate_1.0.3    glue_1.8.0
## [33] farver_2.1.2      colorspace_2.1-1  rmarkdown_2.29    purrr_1.0.4
## [37] tools_4.4.3       pkgconfig_2.0.3   htmltools_0.5.8.1
```