Covid19_Report

Devadath C Prabhu

22/11/2020

1 Introduction

This is an analysis report of the Novel Coronavirus (COVID-19) around the world, to demonstrate data processing and visualization and insights. The report is carried out as a mini-project as part of the course *'Introduction to AI and Data Science (DSC513)'* of first semester MTech at IIIT Kottayam.

1.1 R, R Markdown and RStudio

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

R Markdown is a file format for making dynamic documents with R. An R Markdown document is written in markdown and contains chunks of embedded R code.

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux).

This report is generated using R Markdown and open source edition of RStudio in Mac.

1.2 Packages

The packages used for this analysis are mostly learnt as part of the course and are available in R. The core *tidyverse* includes the packages that are likely to be used in everyday data analysis. Out of the many available packages in it, *dplyr*, *tidyr* and *ggplot2* are used in this analysis. Package *lubridate* is used for date operations, *leaflet* for maps, *kableExtra* for displaying data in tables and *formatR* for formatting purposes.

```
library(tidyverse)
library(lubridate)
library(leaflet)
library(kableExtra)
library(formatR)
library(gridExtra)
```

2 Data

The data used for analysis is pulled from the COVID-19 repository of CSSEGISandData maintained by Johns Hopkins Whiting School of Engineering in GitHub.

Note: *All the analysis in this report are carried out on the data available during the date of generation of the report which is Nov 22, 2020.*

2.1 Data Loading

Three CSV files *time_series_covid19_confirmed_global.csv*, *time_series_covid19_deaths_global.csv* and *time_series_covid19_recovered_global.csv* contain confirmed, deaths and recovered coronavirus data respectively. These files are downloaded and loaded into the workspace.

```r
## three CSV files
filenames <- c("time_series_covid19_confirmed_global.csv",
    "time_series_covid19_deaths_global.csv", "time_series_covid19_recovered_global.csv")

## URL
url.path <- paste0("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/",
    "master/csse_covid_19_data/csse_covid_19_time_series/")

## download files to local
download <- function(filename) {
    url <- file.path(url.path, filename)
    dest <- file.path("./", filename)
    download.file(url, dest)
}
bin <- lapply(filenames, download)

## load data
raw.data.confirmed <- read.csv("time_series_covid19_confirmed_global.csv")
raw.data.deaths <- read.csv("time_series_covid19_deaths_global.csv")
raw.data.recovered <- read.csv("time_series_covid19_recovered_global.csv")
```

Lets have a peek at the raw data without any cleansing. *Table 1* shows the first four entries of confirmed raw data. *Table 2* shows the last four entries of deaths raw data. *Table 3* shows sample data that consists of province state as well. All the tables are restricted to first few columns for display purpose.

```r
## displaying the sample confirmed raw data
raw.data.confirmed[1:4, 1:8] %>% kable(format = "pipe",
    align = "c", caption = "Raw Data Confirmed (8 columns, 4 rows)",
    row.names = FALSE)
```

Table 1: Raw Data Confirmed (8 columns, 4 rows)

Province.State	Country.Region	Lat	Long	X1.22.20	X1.23.20	X1.24.20	X1.25.20
	Afghanistan	33.93911	67.70995	0	0	0	0
	Albania	41.15330	20.16830	0	0	0	0
	Algeria	28.03390	1.65960	0	0	0	0
	Andorra	42.50630	1.52180	0	0	0	0

```r
## displaying the sample deaths raw data
tail_raw_deaths <- tail(raw.data.deaths, 4)
tail_raw_deaths[1:4, 1:8] %>% kable(format = "pipe",
    align = "c", caption = "Raw Data Deaths (8 columns, 4 rows)",
    row.names = FALSE)
```

Table 2: Raw Data Deaths (8 columns, 4 rows)

Province.State	Country.Region	Lat	Long	X1.22.20	X1.23.20	X1.24.20	X1.25.20
	Western Sahara	24.21550	-12.88580	0	0	0	0
	Yemen	15.55273	48.51639	0	0	0	0
	Zambia	-13.13390	27.84933	0	0	0	0
	Zimbabwe	-19.01544	29.15486	0	0	0	0

```r
## displaying the sample recovered raw data
raw.data.recovered[9:12, 1:7] %>% kable(format = "pipe",
    align = "c", caption = "Raw Data Recovered (7 columns, 4 rows)",
    row.names = FALSE)
```

Table 3: Raw Data Recovered (7 columns, 4 rows)

Province.State	Country.Region	Lat	Long	X1.22.20	X1.23.20	X1.24.20
Australian Capital Territory	Australia	-35.4735	149.0124	0	0	0
New South Wales	Australia	-33.8688	151.2093	0	0	0
Northern Territory	Australia	-12.4634	130.8456	0	0	0
Queensland	Australia	-27.4698	153.0251	0	0	0

2.2 Data Cleaning

The three datasets are now cleaned. Unimportant fields are removed, few fields are renamed, then the dataset is converted from wide to long, date format is made readable and the data is grouped by country. Finally, these three datasets are merged into a single dataset.

```r
## method to clean the data
cleanData <- function(data) {
    data <- data %>% select(-c(Province.State, Lat, Long)) %>%
        rename(country = Country.Region)
    data <- data %>% gather(key = date, value = count,
        -country)
    data <- data %>% mutate(date = date %>% substr(2,
        9) %>% mdy())
    data <- data %>% group_by(country, date) %>% summarise(count = sum(count,
        na.rm = T))
    data <- data %>% as.data.frame()
    return(data)
}

## cleaning the datasets and renaming the count column
data.confirmed <- raw.data.confirmed %>% cleanData() %>%
```

```r
    rename(confirmed = count)
data.deaths <- raw.data.deaths %>% cleanData() %>% rename(deaths = count)
data.recovered <- raw.data.recovered %>% cleanData() %>%
    rename(recovered = count)

## merging the datasets
data <- data.confirmed %>% merge(data.deaths, all = T) %>%
    merge(data.recovered, all = T)

## displaying the first five entries after cleaning
head(data, 5) %>% kable(format = "pipe", align = "c",
    caption = "Sample Cleaned Data", row.names = FALSE)
```

Table 4: Sample Cleaned Data

country	date	confirmed	deaths	recovered
Afghanistan	2020-01-22	0	0	0
Afghanistan	2020-01-23	0	0	0
Afghanistan	2020-01-24	0	0	0
Afghanistan	2020-01-25	0	0	0
Afghanistan	2020-01-26	0	0	0

3 At A Glance

With the cleaned data we have, lets have sneak peek at it in this section.

3.1 Initial Data

We all know that the first case of coronavirus appeared in China way back in Dec 2019. Let us have a look at the initial data and the countries that are affected.

```
## filter dates with zero cases
non_zero_confirmed_data <- data %>% filter(confirmed != 0)

## filter to get data of first available date
non_zero_first_confirmed_data <- non_zero_confirmed_data %>%
    filter(non_zero_confirmed_data$date == min(non_zero_confirmed_data$date))

## print
non_zero_first_confirmed_data %>% kable(format = "pipe",
    align = "c", row.names = c(1:nrow(non_zero_first_confirmed_data)),
    caption = "Data of First Date Available in Dataset")
```

Table 5: Data of First Date Available in Dataset

	country	date	confirmed	deaths	recovered
1	China	2020-01-22	548	17	28
2	Japan	2020-01-22	2	0	0
3	Korea, South	2020-01-22	1	0	0
4	Taiwan*	2020-01-22	1	0	0
5	Thailand	2020-01-22	2	0	0
6	US	2020-01-22	1	0	0

We see from the above *Table 5* that the first available data is for Jan 22, 2020 and already 6 countries are affected from the virus by this time. Notice that there are already 17 deaths with 28 recoveries in China.

3.2 First and Latest Ten Countries

The following are the first 10 countries to be affected by coronavirus in order.

```
## order data by date
n_z_c_d_order_date <- non_zero_confirmed_data[order(non_zero_confirmed_data$date), ]

## print first 10 affected countries
head(unique(n_z_c_d_order_date[, 1:1]), 10)
```

```
##  [1] "China"        "Japan"        "Korea, South" "Taiwan*"      "Thailand"
##  [6] "US"           "Singapore"    "Vietnam"      "France"       "Malaysia"
```

Below are the latest 10 countries to be affected by the virus.

```r
## print last 10 affected countries
head(rev(unique(n_z_c_d_order_date[, 1:1])), 10)
```

```
##  [1] "Vanuatu"                "Marshall Islands"      "Solomon Islands"
##  [4] "Lesotho"                "Tajikistan"            "Comoros"
##  [7] "Yemen"                  "Sao Tome and Principe" "Western Sahara"
## [10] "South Sudan"
```

3.3 Total Cases

The virus which originated in China has spread across the globe and has affected almost every country. Let us now see the total number of countries affected by the virus.

```r
length(unique(n_z_c_d_order_date[, 1:1]))
```

```
## [1] 191
```

We see 191 countries are affected by the virus!

Table 6 shows the total confirmed cases, deaths and recoveries in the world as on the date of report generation.

```r
## get the data for last available date
total_cases_world <- data %>% filter(data$date ==
    max(date))

## sum the cases in all countries
total_cases_world <- total_cases_world %>%
    summarise(confirmed = sum(total_cases_world$confirmed),
        deaths = sum(total_cases_world$deaths),
        recovered = sum(total_cases_world$recovered))

## print
total_cases_world %>% kable(format = "pipe",
    align = "c", row.names = FALSE, caption = "Total Cases in the World")
```

Table 6: Total Cases in the World

confirmed	deaths	recovered
58143122	1380436	37214376

3.4 Latest Trend

Let us now see total new cases, new deaths and new recoveries in the world on the date of report generation.

```r
## get data for previous day and sum all cases
total_cases_world_previous_day <- data %>% filter(data$date ==
    max(date - 1))
total_cases_world_previous_day <- total_cases_world_previous_day %>%
    summarise(confirmed = sum(total_cases_world_previous_day$confirmed),
```

```r
        deaths = sum(total_cases_world_previous_day$deaths),
        recovered = sum(total_cases_world_previous_day$recovered))

## calculate new cases and print
total_new_cases_world_today = total_cases_world - total_cases_world_previous_day
total_new_cases_world_today %>% kable(format = "pipe",
    align = "c", row.names = FALSE, caption = "Latest Trend")
```

Table 7: Latest Trend

confirmed	deaths	recovered
579039	8254	339301

3.5 Recovery Rate and Death Rate

Recovery rate and death rate among the closed cases gives us some idea about the deadliness of the virus.

```r
## recovery rate in the world among closed cases
(total_cases_world$recovered/(total_cases_world$recovered + total_cases_world$deaths)) *
    100
```

```
## [1] 96.42326
```

```r
## death rate in the world among closed cases
(total_cases_world$deaths/(total_cases_world$recovered + total_cases_world$deaths)) *
    100
```

```
## [1] 3.57674
```

Coronavirus pandemic has about more than 96% recovery rate, which is very positive.

Total active cases in the world can be calculated by subtracting total deaths and recovered cases from total confirmed cases.

```r
## active cases in the world
total_cases_world$confirmed - total_cases_world$deaths - total_cases_world$recovered
```

```
## [1] 19548310
```

We still have more than 19.5M active cases in the world and are far away from the day where cases will be 0.

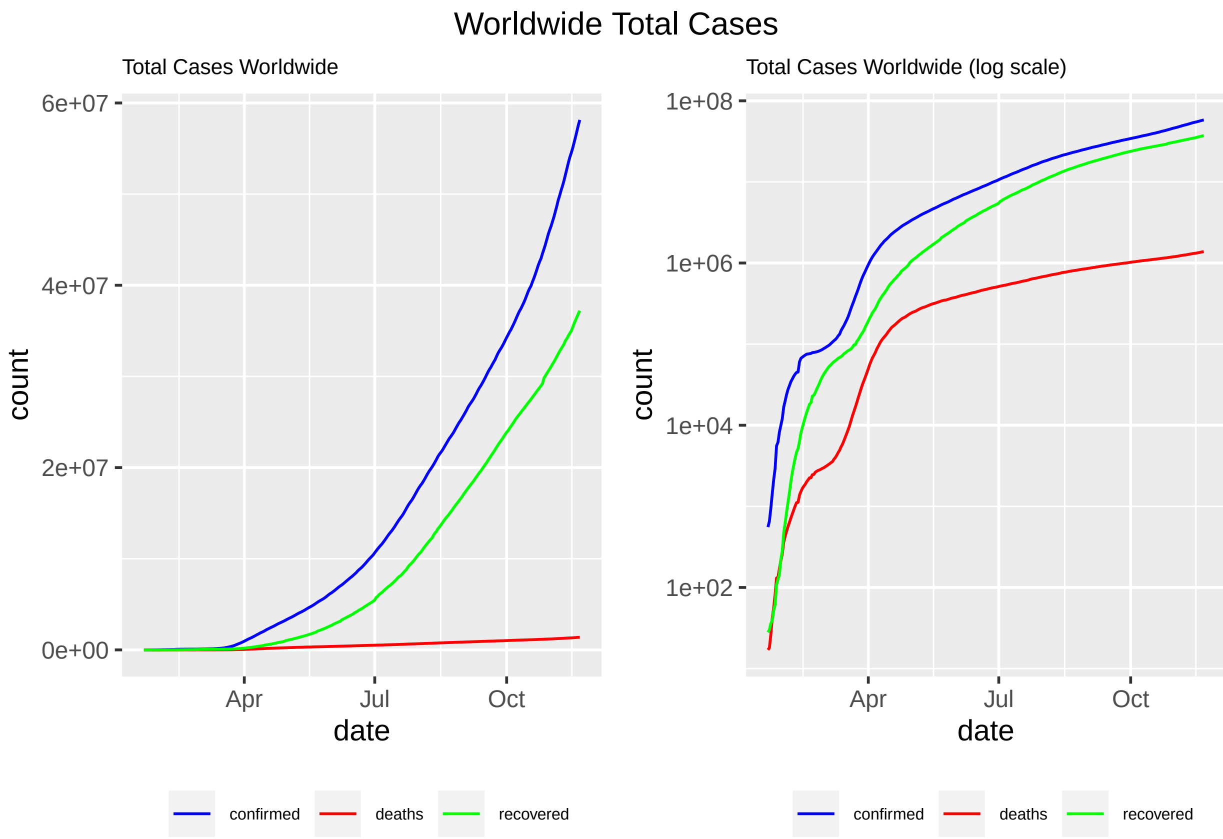
4 Worldwide

We had a quick peek at the data until now. In this section, we analyze the data of various countries worldwide and also visualize the data and the patterns.

4.1 Total Cases

Let us see how the total cases has increased worldwide. Flatter the curve of confirmed cases, better for us and the world.

```
## data of world for each day
data_world <- data %>% group_by(date) %>% summarise(confirmed = sum(confirmed,
    na.rm = T), deaths = sum(deaths, na.rm = T), recovered = sum(recovered,
    na.rm = T))

## gather data to long format for graph
data_world_for_graph <- data_world %>% gather(key = type,
    value = count, -c(date))

## create graph
world_graph <- data_world_for_graph %>% ggplot(aes(x = date,
    y = count, color = type)) + geom_line() + labs(title = "Total Cases Worldwide") +
    scale_color_manual(values = c("blue", "red", "green")) +
    theme(legend.position = "bottom", legend.title = element_blank(),
        legend.text = element_text(size = 6), plot.title = element_text(size = 8))
world_graph_log <- data_world_for_graph %>% ggplot(aes(x = date,
    y = count, color = type)) + geom_line() + scale_y_log10() +
    labs(title = "Total Cases Worldwide (log scale)") +
    scale_color_manual(values = c("blue", "red", "green")) +
    theme(legend.position = "bottom", legend.title = element_blank(),
        legend.text = element_text(size = 6), plot.title = element_text(size = 8))

## display graph
grid.arrange(world_graph, world_graph_log, ncol = 2, widths = c(6,
    6), top = "Worldwide Total Cases")
```



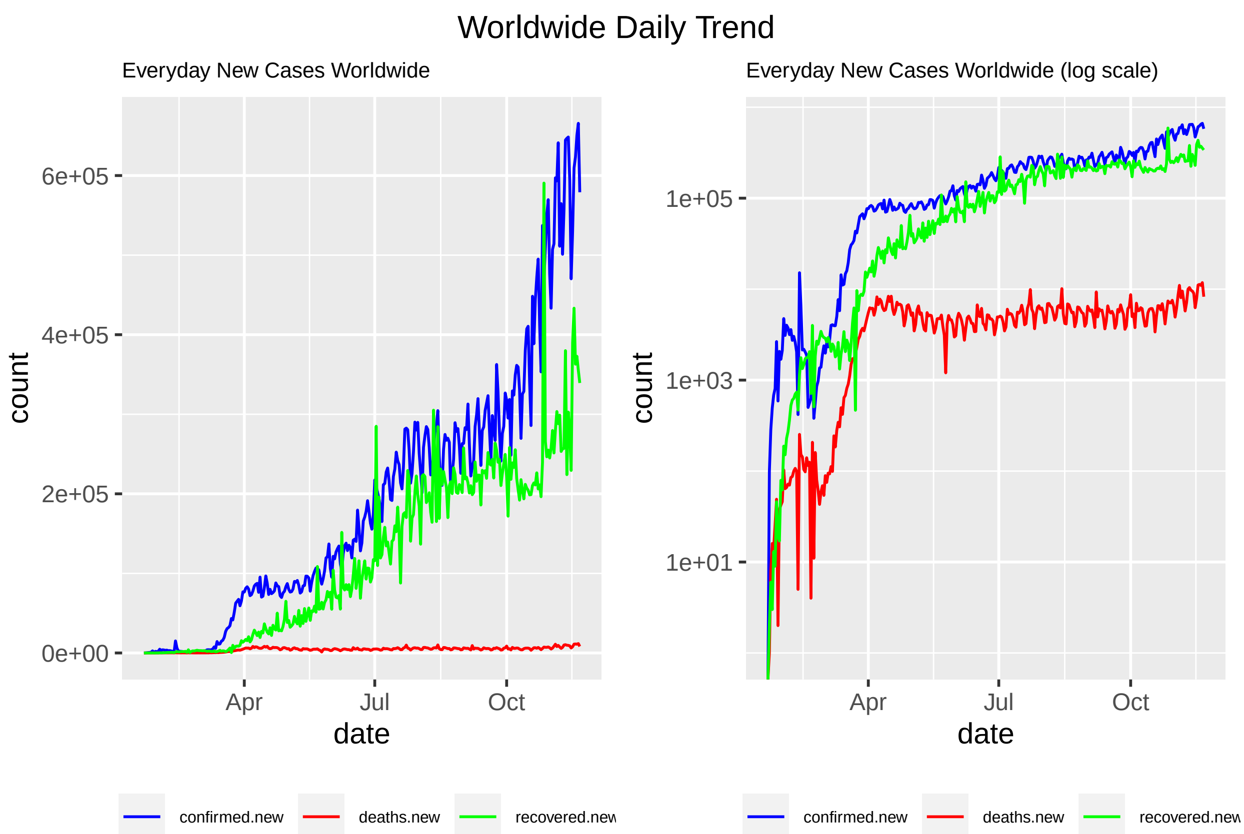
Notice that the confirmed cases are constantly increasing and we are yet far away from flattening the curve.

4.2 Daily Trends

This section talks about the daily numbers. Lesser the values for confirmed and deaths, better. Since the dataset we had is the cumulative dataset, we will first calculate the daily numbers and then visualize them.

```
## daily new numbers
data_world <- data_world %>% mutate(confirmed.new = ifelse(date ==
    min(data_world$date), 0, confirmed - lag(confirmed,
    n = 1)), deaths.new = ifelse(date == min(data_world$date),
    0, deaths - lag(deaths, n = 1)), recovered.new = ifelse(date ==
    min(data_world$date), 0, recovered - lag(recovered,
    n = 1)))
data_world <- data_world %>% mutate(confirmed.new = ifelse(confirmed.new <
    0, 0, confirmed.new), deaths.new = ifelse(deaths.new <
    0, 0, deaths.new), recovered.new = ifelse(recovered.new <
    0, 0, recovered.new))

## gather data to long format for graph
data_world_daily_for_graph <- data_world %>% select(c(date,
    confirmed.new, deaths.new, recovered.new)) %>%
    gather(key = type, value = count, -c(date))

## create graph
world_graph_daily <- data_world_daily_for_graph %>%
```

```r
    ggplot(aes(x = date, y = count, color = type)) +
    geom_line() + labs(title = "Everyday New Cases Worldwide") +
    scale_color_manual(values = c("blue", "red", "green")) +
    theme(legend.position = "bottom", legend.title = element_blank(),
        legend.text = element_text(size = 6), plot.title = element_text(size = 8))
world_graph_daily_log <- data_world_daily_for_graph %>%
    ggplot(aes(x = date, y = count, color = type)) +
    geom_line() + labs(title = "Everyday New Cases Worldwide (log scale)") +
    scale_y_log10() + scale_color_manual(values = c("blue",
    "red", "green")) + theme(legend.position = "bottom",
    legend.title = element_blank(), legend.text = element_text(size = 6),
    plot.title = element_text(size = 8))

## display graph
grid.arrange(world_graph_daily, world_graph_daily_log,
    ncol = 2, widths = c(6, 6), top = "Worldwide Daily Trend")
```

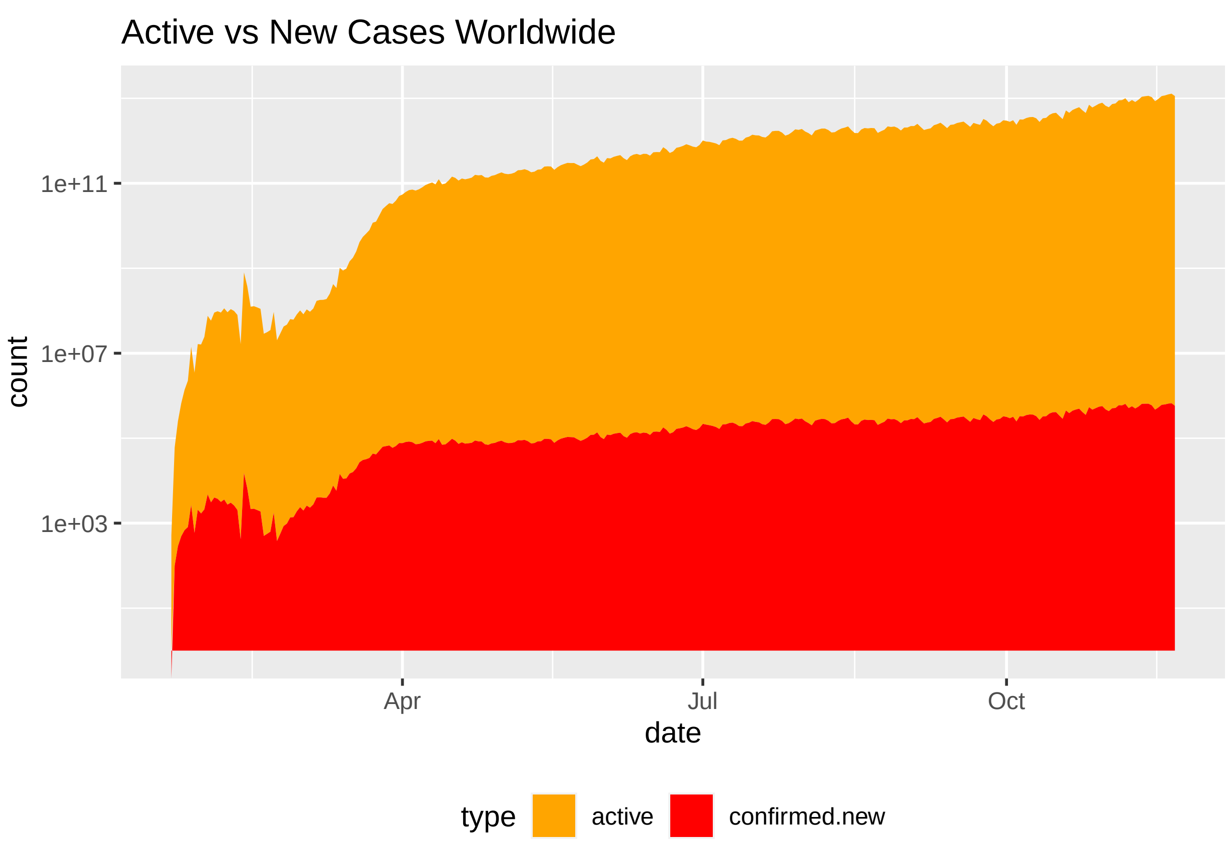


4.3 Active vs New

This section gives us a brief about the active cases and the daily new cases.

```r
## calculate active cases
data_world <- data_world %>% mutate(active = confirmed -
    deaths - recovered)
```

```r
## convert dataset to required format
data_world_ative_new <- data_world %>% select(c(date, confirmed.new,
    active)) %>% gather(key = type, value = count, -c(date))

## create graph
data_world_ative_new_graph <- data_world_ative_new %>% ggplot(aes(x = date,
    y = count)) + geom_area(aes(fill = type)) + scale_y_log10() +
    labs(title = "Active vs New Cases Worldwide") + scale_fill_manual(values = c("orange",
    "red")) + theme(legend.position = "bottom")

## display graph
data_world_ative_new_graph
```

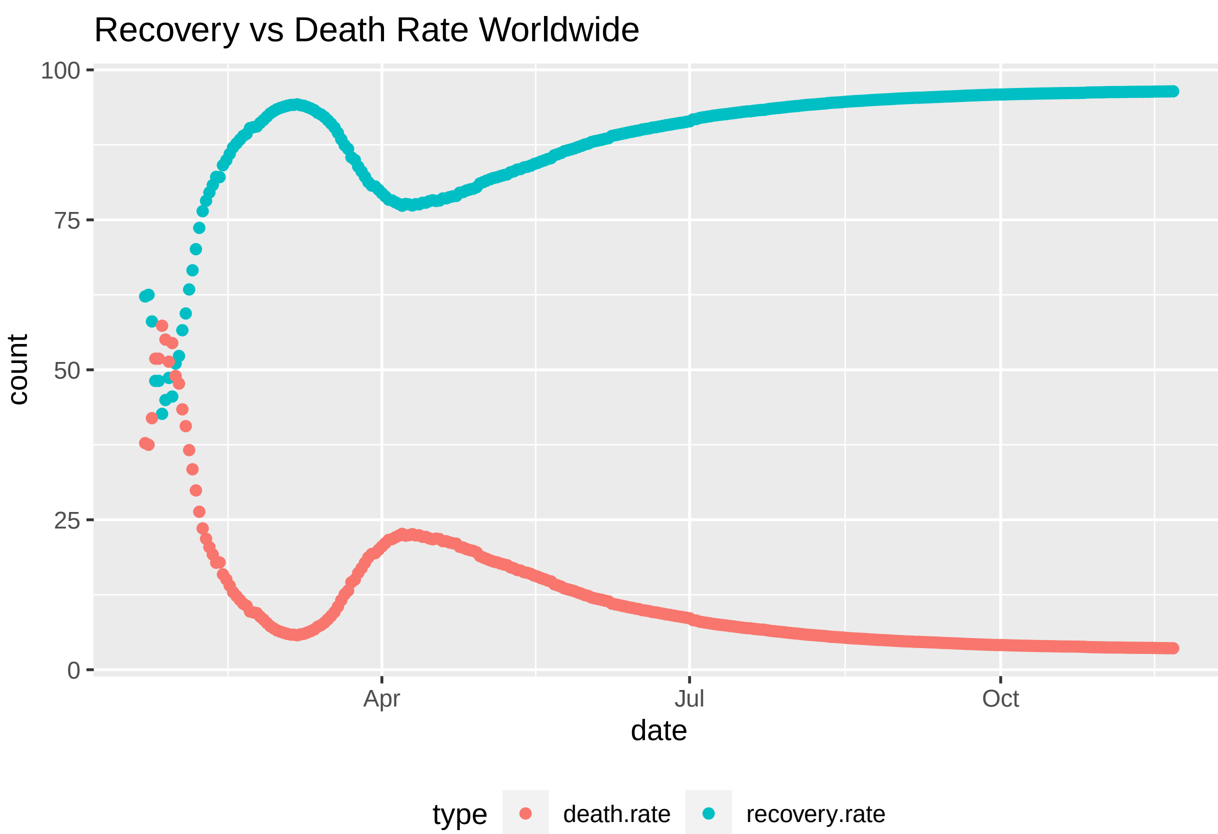


Active cases generally depend on new cases and we see that through the pattern in above graph.

4.4 Recovery vs Death Rate

```r
## calculate recovery and death rate
data_world <- data_world %>% mutate(recovery.rate = (recovered/(recovered +
    deaths)) * 100)
data_world <- data_world %>% mutate(death.rate = (deaths/(recovered +
    deaths)) * 100)

## convert dataset to required format
world_deaths_recovery <- data_world %>% select(c(date,
```

```
    recovery.rate, death.rate)) %>% gather(key = type,
    value = count, -c(date))

## create graph
world_deaths_recovery_graph <- world_deaths_recovery %>%
    ggplot(aes(x = date, y = count, color = type)) +
    geom_point() + labs(title = "Recovery vs Death Rate Worldwide") +
    theme(legend.position = "bottom")

## display graph
world_deaths_recovery_graph
```

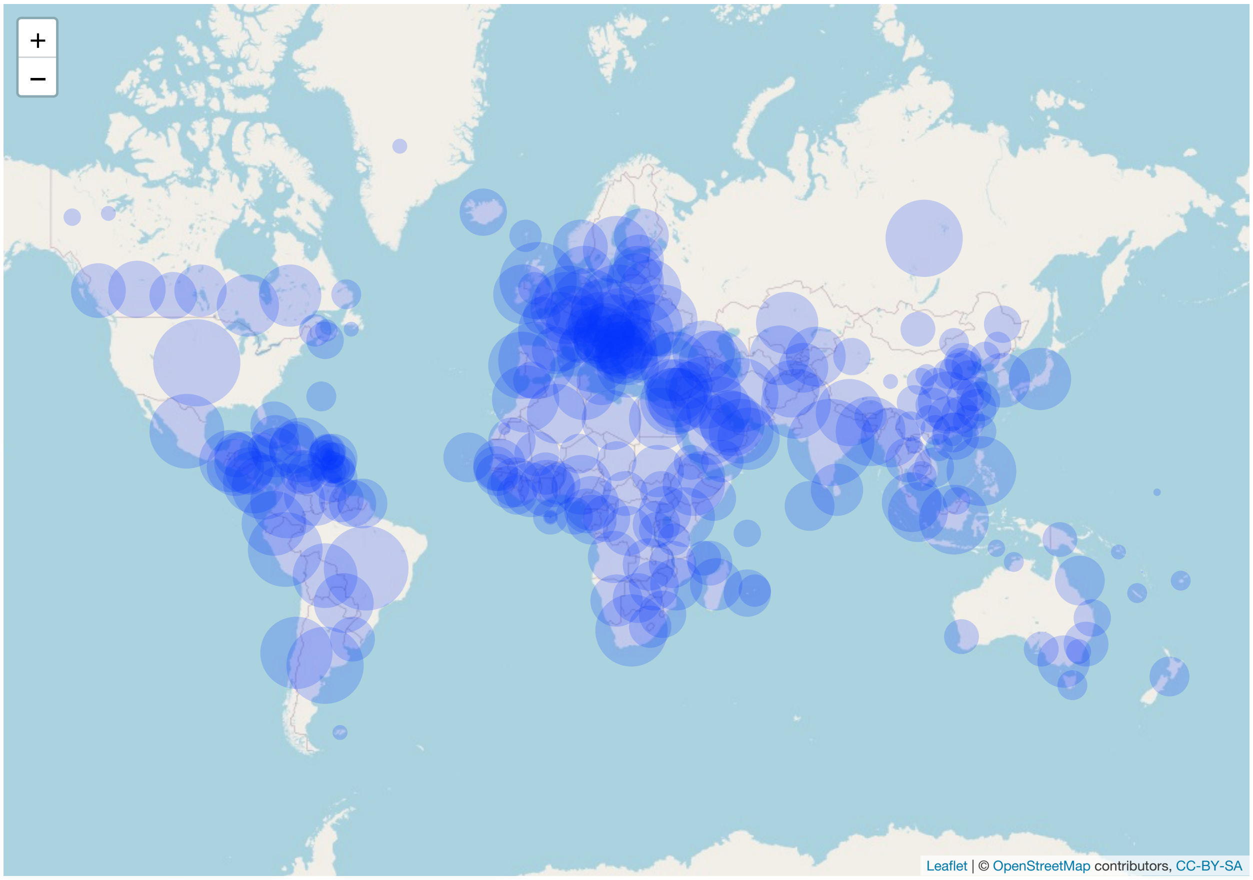


Apart from the initial days where death rate and recovery rate were both around 50%, we see that both recovery and death rates have stabilized now with recovery rate hovering close to 96.5% and death rate about 3.5%. Note that these rates are calculated with respect to closed cases.

4.5 Cases Worldwide

```
## prepare the dataset
raw_data_confirmed_map <- raw.data.confirmed
raw_data_confirmed_map <- raw_data_confirmed_map %>%
    mutate(confirmed = raw_data_confirmed_map[, ncol(raw_data_confirmed_map)])
raw_data_confirmed_map <- raw_data_confirmed_map %>%
    select(c(Country.Region, Lat, Long, confirmed))
```

```r
## create and display map
leaflet(width = 1000, height = 700) %>% addTiles() %>%
    addCircleMarkers(raw_data_confirmed_map$Long, raw_data_confirmed_map$Lat,
        radius = 1.5 * log2(raw_data_confirmed_map$confirmed),
        stroke = F)
```



The coronavirus has spread to all corners of the world and its evident from the map above.

4.6 Countries with Zero Deaths

Coronavirus has low death rate as we already saw. It could be possible that few small countries might have recorded 0 deaths for coronavirus. Lets check if there are any such countries and if yes, some of their stats.

```r
## filter countries with zero deaths
zero_deaths_countries <- data %>% filter(date == max(date),
    deaths == 0) %>% mutate(active = confirmed - deaths -
    recovered) %>% select(-c(date))

## print
zero_deaths_countries %>% kable(format = "pipe", align = "c",
    row.names = c(1:nrow(zero_deaths_countries)), caption = "Countries with Zero Deaths")
```

Table 8: Countries with Zero Deaths

	country	confirmed	deaths	recovered	active
1	Bhutan	379	0	360	19
2	Cambodia	306	0	295	11
3	Dominica	72	0	55	17
4	Eritrea	551	0	473	78
5	Grenada	41	0	30	11
6	Holy See	27	0	15	12
7	Laos	25	0	23	2
8	Marshall Islands	4	0	1	3
9	Mongolia	608	0	337	271
10	Saint Kitts and Nevis	19	0	19	0
11	Saint Vincent and the Grenadines	84	0	78	6
12	Seychelles	163	0	159	4
13	Solomon Islands	16	0	5	11
14	Timor-Leste	30	0	30	0
15	Vanuatu	1	0	0	1

There are 15 countries with zero deaths as of today as shown in *Table 8*.

4.7 Countries with Less Active Cases

Lets first see if there are countries with zero active cases.

```
## filter countries with zero active cases
zero_active_countries <- data %>% filter(date == max(date)) %>%
    mutate(active = confirmed - deaths - recovered) %>%
    filter(active == 0) %>% select(-c(date))

## print
zero_active_countries %>% kable(row.names = c(1:nrow(zero_active_countries)),
    format = "pipe", caption = "Countries with Zero Active Cases",
    align = "c")
```

Table 9: Countries with Zero Active Cases

	country	confirmed	deaths	recovered	active
1	Brunei	148	3	145	0
2	Fiji	35	2	33	0
3	Saint Kitts and Nevis	19	0	19	0
4	Timor-Leste	30	0	30	0

There are four countries! *Next destination, may be?*

Lets now check for the countries with less than 100 active cases.

```
## filter countries with <= 100 active cases
less_active_countries <- data %>% filter(date == max(date)) %>%
    mutate(active = confirmed - deaths - recovered) %>%
```

```r
    filter(active <= 100) %>% select(-c(date))

## print
less_active_countries %>% kable(row.names = c(1:nrow(less_active_countries)),
    format = "pipe", caption = "Countries with Less than 100 Active Cases",
    align = "c")
```

Table 10: Countries with Less than 100 Active Cases

	country	confirmed	deaths	recovered	active
1	Antigua and Barbuda	139	4	128	7
2	Barbados	255	7	241	7
3	Bhutan	379	0	360	19
4	Brunei	148	3	145	0
5	Burundi	656	1	575	80
6	Cambodia	306	0	295	11
7	Chad	1633	101	1469	63
8	Comoros	596	7	572	17
9	Diamond Princess	712	13	659	40
10	Djibouti	5661	61	5552	48
11	Dominica	72	0	55	17
12	Equatorial Guinea	5130	85	4975	70
13	Eritrea	551	0	473	78
14	Fiji	35	2	33	0
15	Gabon	9131	59	8976	96
16	Gambia	3726	123	3582	21
17	Grenada	41	0	30	11
18	Guinea-Bissau	2421	43	2286	92
19	Holy See	27	0	15	12
20	Laos	25	0	23	2
21	Marshall Islands	4	0	1	3
22	Mauritius	494	10	433	51
23	Monaco	581	3	507	71
24	MS Zaandam	9	2	0	7
25	New Zealand	2028	25	1953	50
26	Papua New Guinea	604	7	586	11
27	Saint Kitts and Nevis	19	0	19	0
28	Saint Vincent and the Grenadines	84	0	78	6
29	Sao Tome and Principe	979	17	920	42
30	Senegal	15865	330	15465	70
31	Seychelles	163	0	159	4
32	Singapore	58148	28	58064	56
33	Solomon Islands	16	0	5	11
34	Suriname	5295	116	5166	13
35	Taiwan*	611	7	546	58
36	Thailand	3913	60	3761	92
37	Timor-Leste	30	0	30	0
38	Vanuatu	1	0	0	1
39	Western Sahara	10	1	8	1
40	Yemen	2093	608	1441	44

There are whopping 40 countries with active cases less than 100! The entry of Singapore in this table is commendable as it has active cases less than 100 with confirmed cases above 58k.

5 India

We all know that on this day, India is one of the worst affected country by coronavirus. Let us analyze data about India in this section.

5.1 First Case

```
## get the date of first reported case
first_case <- head(data %>% filter(country == "India", confirmed !=
    0) %>% select(c(date, confirmed)), 1)

## print
first_case %>% kable(format = "pipe", align = "c", row.names = FALSE,
    caption = "First Case in India")
```

Table 11: First Case in India

date	confirmed
2020-01-30	1

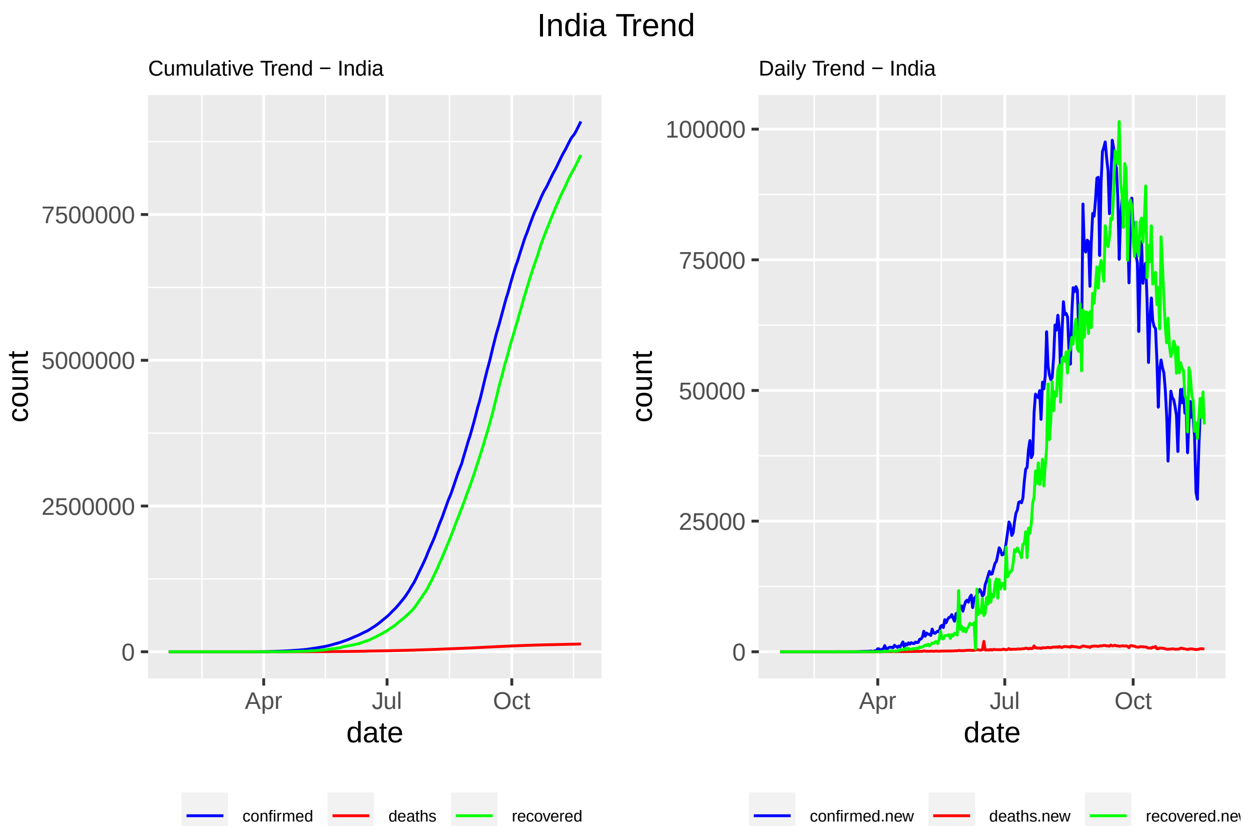
First case in India was confirmed on Jan 30, 2020 and only one case was reported on this day.

5.2 Cumulative and Daily Trends

This section gives trends of cumulative and daily cases in India.

```
## filter data of India
data_india <- data %>% filter(country == "India") %>%
    group_by(date) %>% summarise(confirmed = sum(confirmed,
    na.rm = T), deaths = sum(deaths, na.rm = T), recovered = sum(recovered,
    na.rm = T))

## gather data to long format for graph
data_india_for_graph <- data_india %>% gather(key = type,
    value = count, -c(date))

## daily new numbers
data_india <- data_india %>% mutate(confirmed.new = ifelse(date ==
    min(data_india$date), 0, confirmed - lag(confirmed,
    n = 1)), deaths.new = ifelse(date == min(data_india$date),
    0, deaths - lag(deaths, n = 1)), recovered.new = ifelse(date ==
    min(data_india$date), 0, recovered - lag(recovered,
    n = 1)))
data_india <- data_india %>% mutate(confirmed.new = ifelse(confirmed.new <
    0, 0, confirmed.new), deaths.new = ifelse(deaths.new <
    0, 0, deaths.new), recovered.new = ifelse(recovered.new <
    0, 0, recovered.new))

## gather data to long format for graph
```

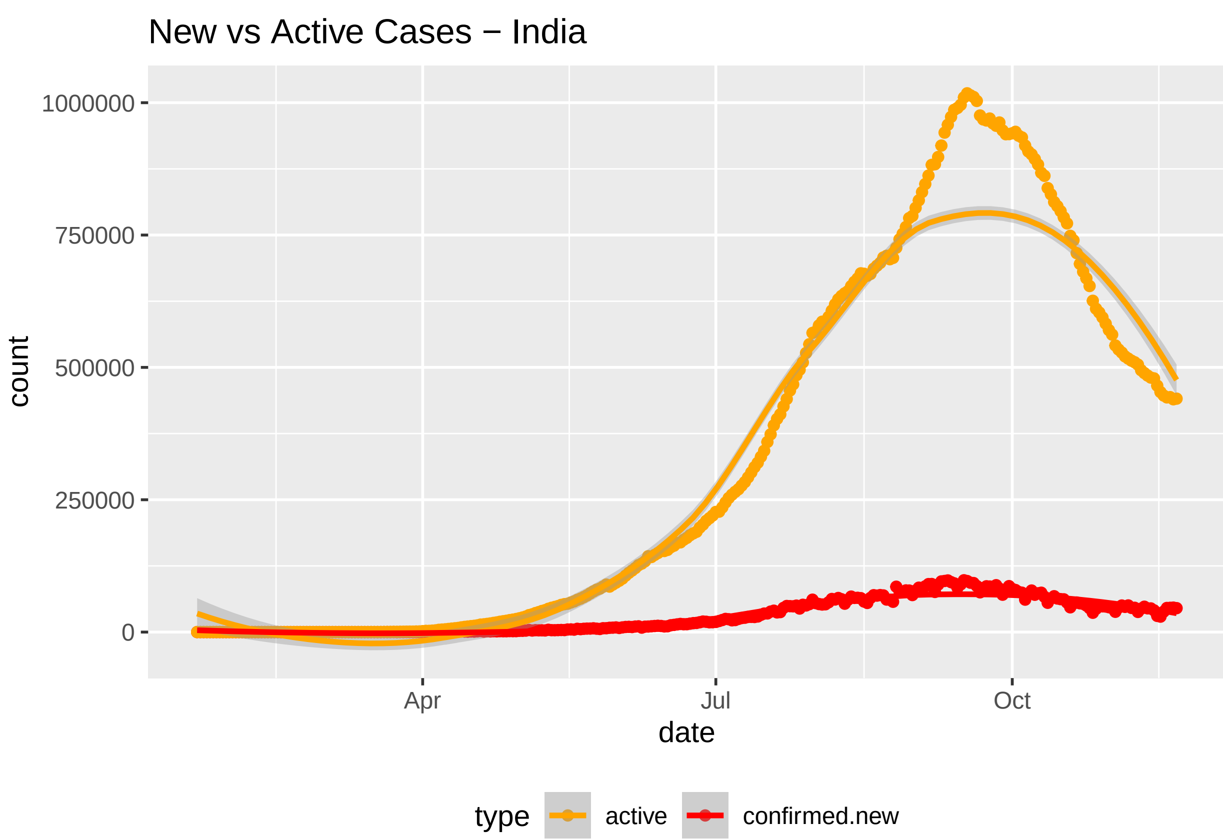
```
data_india_daily_for_graph <- data_india %>% select(c(date,
    confirmed.new, deaths.new, recovered.new)) %>%
    gather(key = type, value = count, -c(date))

## create graph
india_graph <- data_india_for_graph %>% ggplot(aes(x = date,
    y = count, color = type)) + geom_line() + labs(title = "Cumulative Trend - India") +
    scale_color_manual(values = c("blue", "red", "green")) +
    theme(legend.position = "bottom", legend.title = element_blank(),
        legend.text = element_text(size = 6), plot.title = element_text(size = 8))
india_graph_daily <- data_india_daily_for_graph %>%
    ggplot(aes(x = date, y = count, color = type)) +
    geom_line() + labs(title = "Daily Trend - India") +
    scale_color_manual(values = c("blue", "red", "green")) +
    theme(legend.position = "bottom", legend.title = element_blank(),
        legend.text = element_text(size = 6), plot.title = element_text(size = 8))

## display graph
grid.arrange(india_graph, india_graph_daily, ncol = 2,
    widths = c(5, 5), top = "India Trend")
```



From the plot above, we see that there is a slow dip in daily new cases from approximately last one month. Another interesting fact is that daily recoveries have increased over daily new cases in the same time period.

5.3 New vs Active

```r
## calculate active cases
data_india <- data_india %>% mutate(active = confirmed -
    deaths - recovered)

## arrange data for plot
data_india_ative_new <- data_india %>% select(c(date,
    confirmed.new, active)) %>% gather(key = type,
    value = count, -c(date))

## create graph
data_india_ative_new_graph <- data_india_ative_new %>%
    ggplot(aes(x = date, y = count, color = type)) +
    geom_point() + geom_smooth() + labs(title = "New vs Active Cases - India") +
    scale_color_manual(values = c("orange", "red")) +
    theme(legend.position = "bottom")

## display graph
data_india_ative_new_graph
```



The active cases have started to dip down from past month along with the new cases as seen in the graph.

5.4 Recovery vs Death Rate

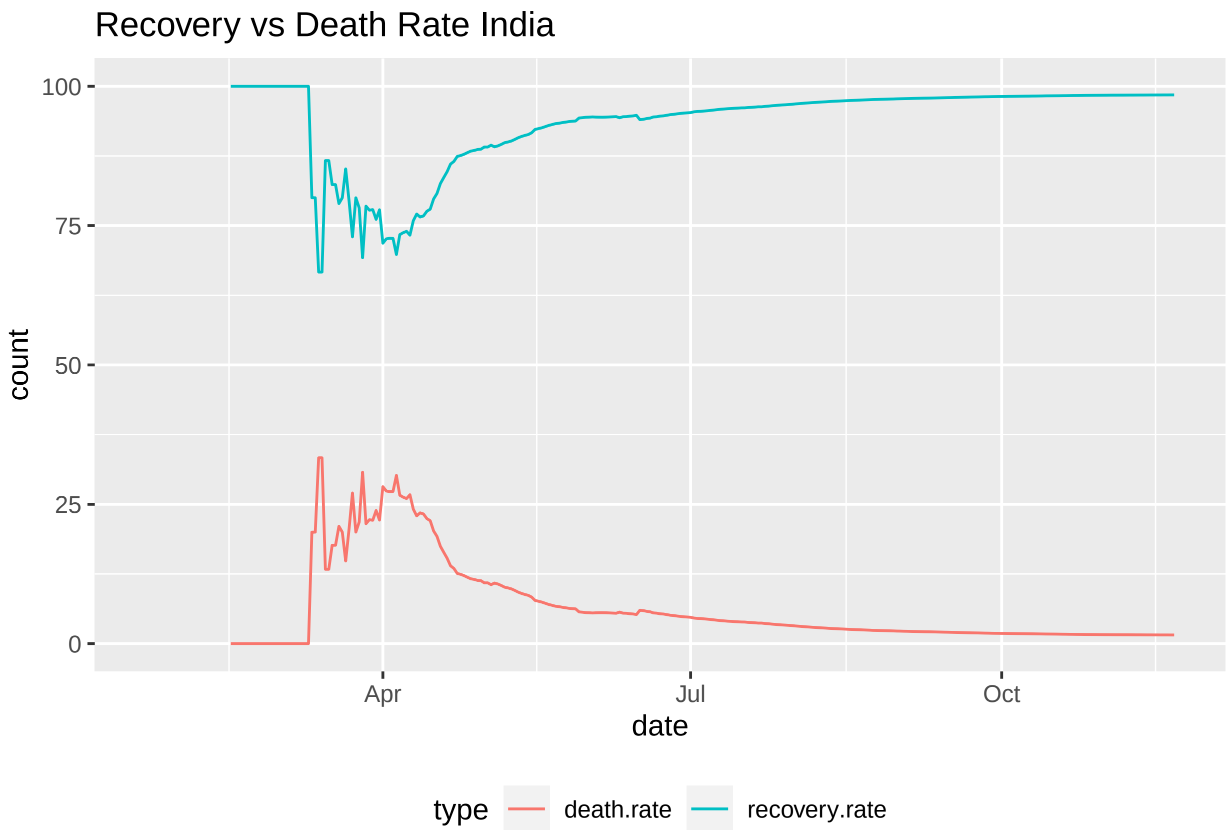
Recovery rate is significant as it implies the deadliness of the virus. Higher the recovery rate, lower will be the death rate and is better for the country. Both the recovery and death rates are calculated with respect to closed cases. Active cases are not taken into consideration for calculation of recovery and death rates.

```
## calculate recovery and
## death rate
data_india <- data_india %>%
    mutate(recovery.rate = (data_india$recovered/(data_india$recovered +
        data_india$deaths)) *
        100)
data_india <- data_india %>%
    mutate(death.rate = (data_india$deaths/(data_india$recovered +
        data_india$deaths)) *
        100)

## gather data to plot
india_deaths_recovery <- data_india %>%
    select(c(date, recovery.rate,
        death.rate)) %>% gather(key = type,
    value = count, -c(date))

## recovery and death rate
india_deaths_recovery %>% filter(date ==
    max(date)) %>% select(-c(date)) %>%
    kable(format = "pipe", align = "c",
        row.names = FALSE, caption = "Recovery and Death Rates in India")
```

Table 12: Recovery and Death Rates in India

type	count
recovery.rate	98.460666
death.rate	1.539335

```
## create graph
india_deaths_recovery_graph <- india_deaths_recovery %>%
    ggplot(aes(x = date, y = count,
        color = type)) + geom_line() +
    labs(title = "Recovery vs Death Rate India") +
    theme(legend.position = "bottom")

## display graph
india_deaths_recovery_graph
```

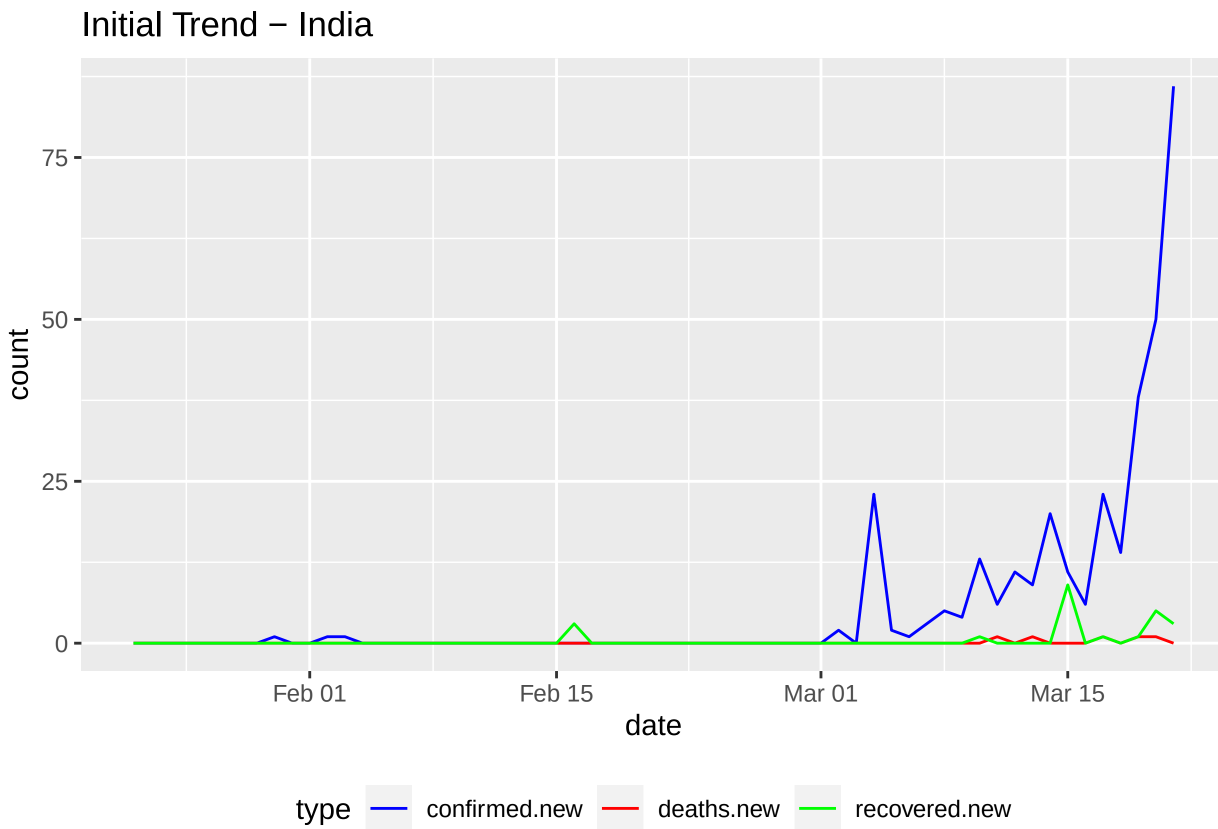


Like worldwide, India as well we see that apart from initial variations, recovery and death rates are consistently hovering around 98% and 2% respectively.

5.5 Initial Trend

This section shows the initial daily trend of the cases in India. First 60 days of data is used to analyze the initial trend in India since corona cases started increasing only in mid of March.

```
## first 60 days of data
initial_data_india <- head(data_india, 60) %>% select(c(date,
    confirmed.new, deaths.new, recovered.new)) %>%
    gather(key = type, value = count, -c(date))

## create plot
initial_graph_india <- initial_data_india %>% ggplot(aes(x = date,
    y = count, color = type)) + geom_line() + labs(title = "Initial Trend - India") +
    scale_color_manual(values = c("blue", "red", "green")) +
    theme(legend.position = "bottom")

## display graph
initial_graph_india
```

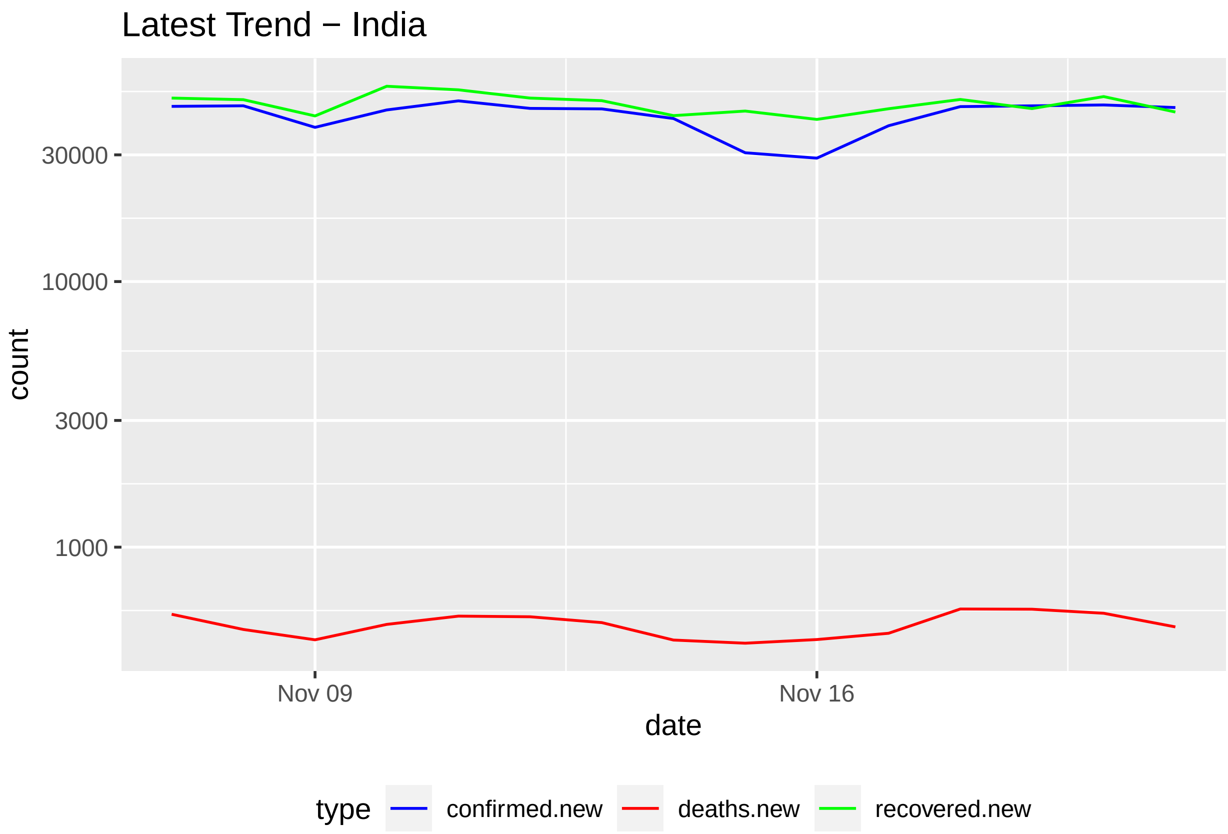


The graph tells us that there were rarely any cases until March 01, 2020 in India. The cases slowly started to increase from March in India.

5.6 Latest Trend

For latest daily trend, we analyze the last 15 days of data for India.

```
## last 15 days of data
latest_data_india <- tail(data_india, 15) %>% select(c(date,
    confirmed.new, deaths.new, recovered.new)) %>%
    gather(key = type, value = count, -c(date))

## create graph
latest_graph_india <- latest_data_india %>% ggplot(aes(x = date,
    y = count, color = type)) + geom_line() + scale_y_log10() +
    labs(title = "Latest Trend - India") + scale_color_manual(values = c("blue",
    "red", "green")) + theme(legend.position = "bottom")

## display graph
latest_graph_india
```



We see that for last 15days in India, daily deaths are below 1000 and recoveries are higher than new cases in general.

6 Top Ten Countries

Here we see top 10 countries with various parameters and compare them.

6.1 Top Ten Countries - Total Confirmed Cases

```
## current cases by country
data_by_country <- data %>% filter(data$date == max(date))

## top 10 countries with confirmed cases
data_top_ten <- head(data_by_country[rev(order(data_by_country$confirmed)),
    ], 10)

## display data
data_top_ten %>% select(c(country, confirmed, deaths,
    recovered)) %>% kable(format = "pipe", align = "c",
    row.names = FALSE, caption = "Top 10 Countries")
```

Table 13: Top 10 Countries

country	confirmed	deaths	recovered
US	12088410	255861	4529700
India	9095806	133227	8521617
Brazil	6052786	168989	5437189
France	2178023	48593	156755
Russia	2047563	35442	1564897
Spain	1556730	42619	150376
United Kingdom	1497135	54721	3310
Italy	1380531	49261	539524
Argentina	1366182	36902	1187053
Colombia	1240493	35104	1144923

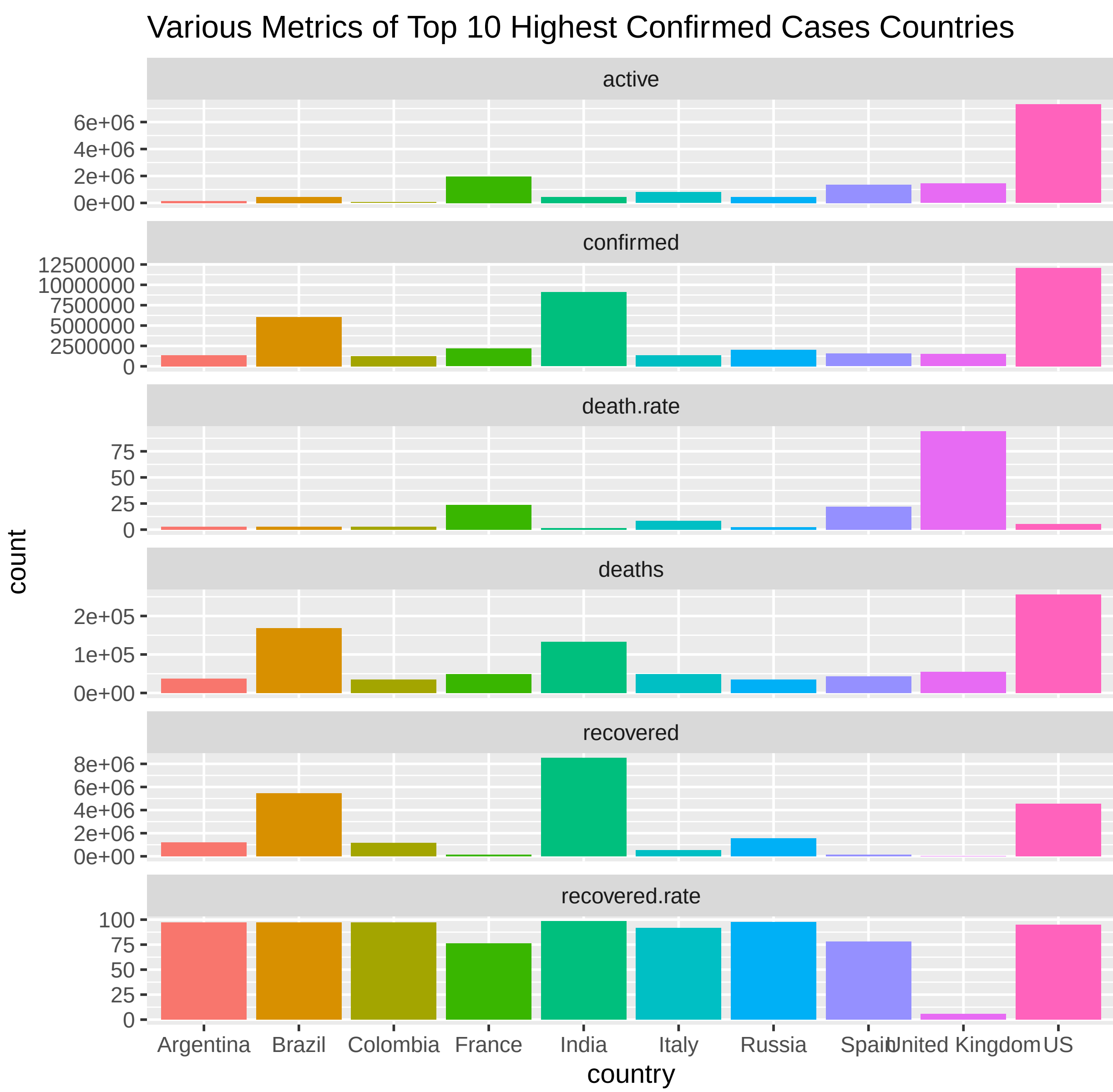
Table 13 shows the top 10 countries in the world with most confirmed corona cases. US has already breached the 10M mark and India is at second with 8.7M.

We can also notice that the data for United Kingdom looks invalid. On verifying this in the dataset used and online, it can be confirmed that the data for recovered cases of UK is corrupt.

Lets visualize various parameters of top 10 countries with highest cases.

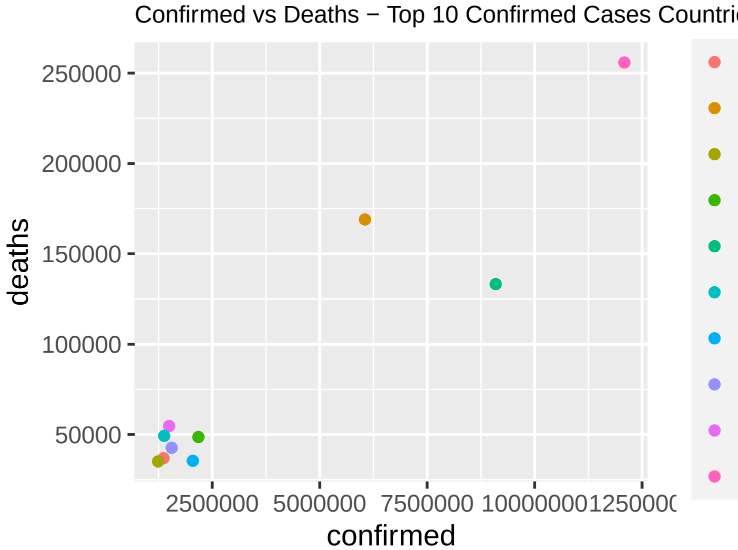
```
## prepare date by countries
data_by_country <- data_by_country %>% mutate(active = confirmed -
    deaths - recovered)
data_by_country <- data_by_country %>% mutate(death.rate = ifelse(recovered >
    0, deaths/(recovered + deaths) * 100, 0))
data_by_country <- data_by_country %>% mutate(recovered.rate = ifelse(recovered >
    0, recovered/(recovered + deaths) * 100, NA))

## select top 10 countries with confirmed cases
data_by_country_top_ten <- head(data_by_country[rev(order(data_by_country$confirmed)),
    ], 10, row.names = FALSE)
```

```r
## gather data to plot
data_top_ten_for_graph <- data_by_country_top_ten %>%
    gather(key = type, value = count, -c(country, date)) %>%
    select(c(country, type, count))

## create plot
top_ten_graph <- data_top_ten_for_graph %>% ggplot(aes(x = country,
    y = count, fill = country, group = country)) +
    geom_bar(stat = "identity") + theme(legend.position = "none") +
    labs(title = "Various Metrics of Top 10 Highest Confirmed Cases Countries") +
    facet_wrap(~type, ncol = 1, scales = "free_y")

## display plot
top_ten_graph
```

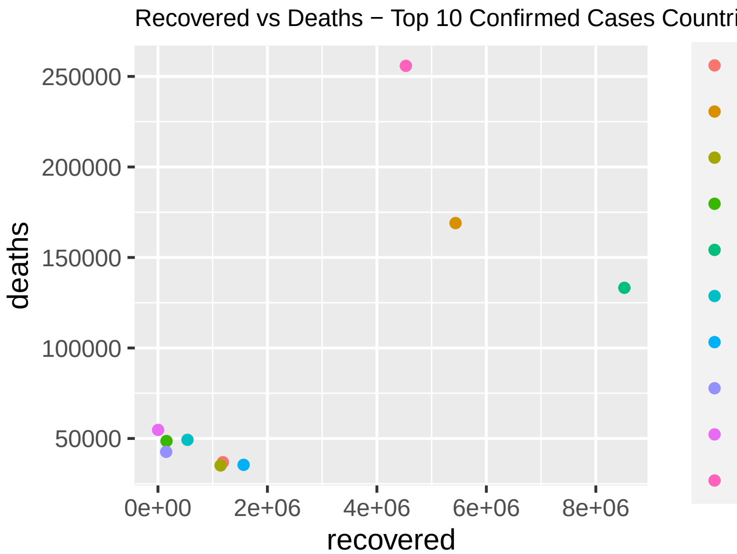


The bar charts above show the number of active, confirmed, death, recovered cases and death, recovered rates for top 10 countries with highest confirmed cases.

```
## confirmed vs deaths plot
data_by_country_top_ten %>% ggplot(aes(x = confirmed,
    y = deaths, group = country)) + geom_point(aes(color = country)) +
    labs(title = "Confirmed vs Deaths - Top 10 Confirmed Cases Countries") +
    theme(legend.title = element_blank(), plot.title = element_text(size = 9))
```

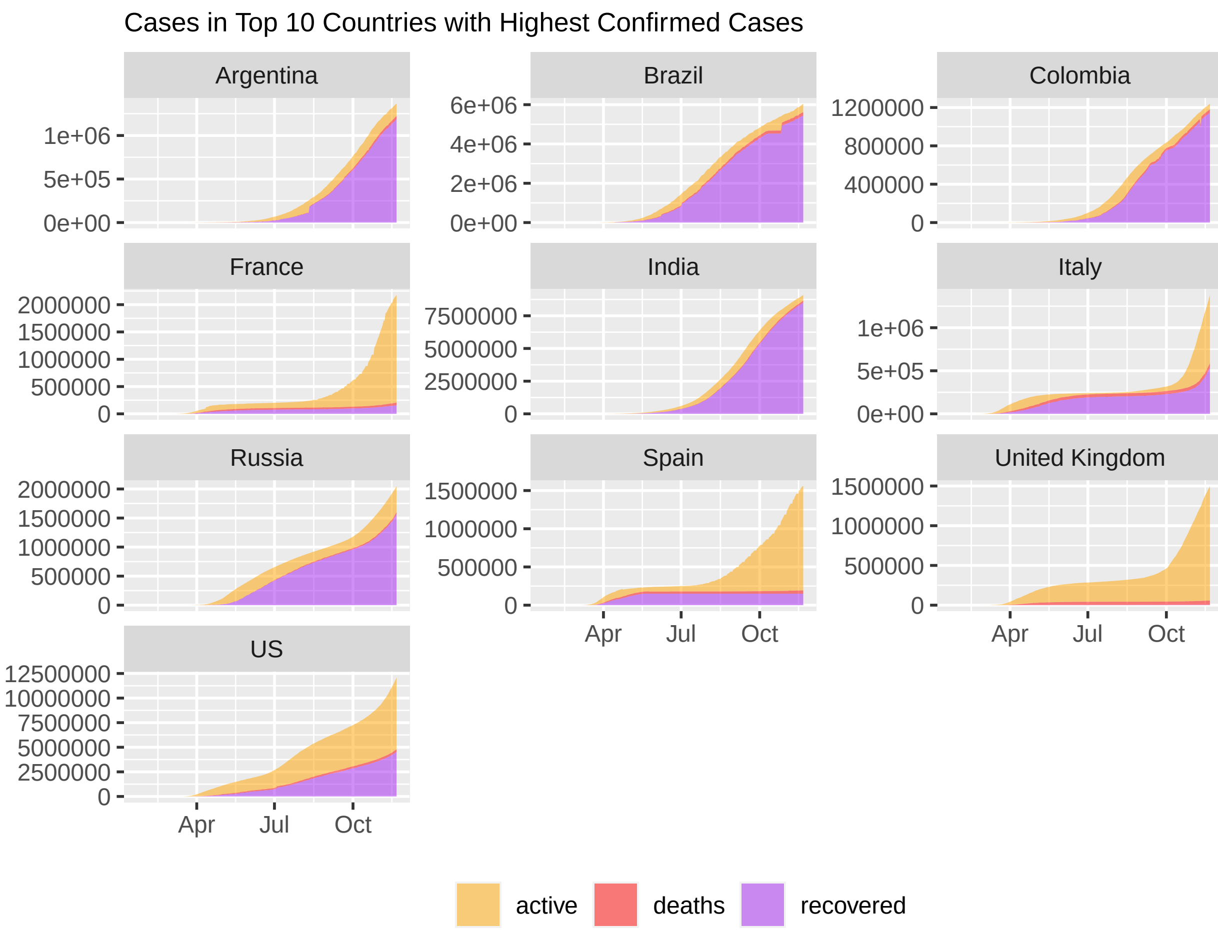



```
## recovered vs deaths plot
data_by_country_top_ten %>% ggplot(aes(x = recovered,
    y = deaths, group = country)) + geom_point(aes(color = country)) +
    labs(title = "Recovered vs Deaths - Top 10 Confirmed Cases Countries") +
    theme(legend.title = element_blank(), plot.title = element_text(size = 9))
```

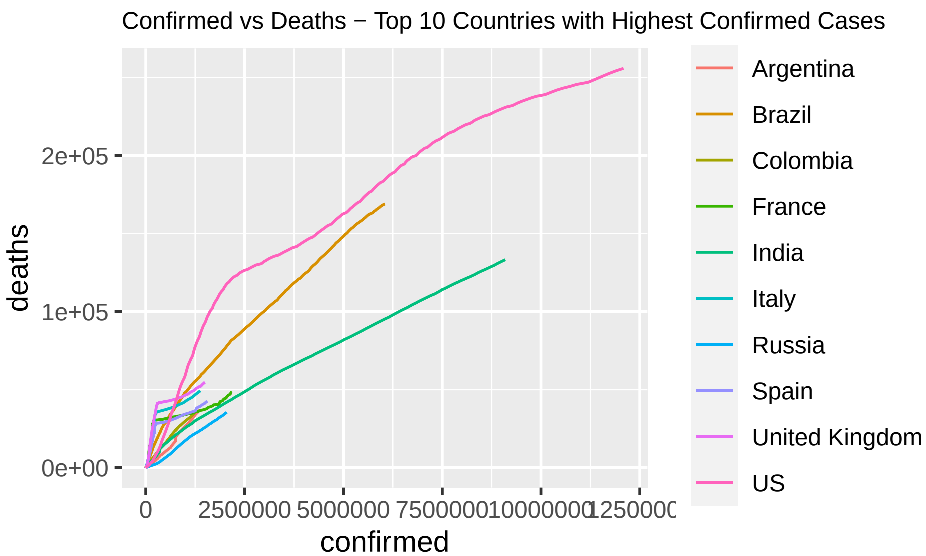



```
## top 10 countries
top_ten_countries <- data_top_ten %>% pull(country) %>% as.character()
```

```r
## data of top 10 countries
top_ten_data <- data %>% filter(country %in% top_ten_countries)

## alter dataset for plot
top_ten_data_without_confirmed_field <- top_ten_data %>% mutate(active = confirmed -
    deaths - recovered) %>% select(-c(confirmed))
top_ten_data_for_graph <- top_ten_data_without_confirmed_field %>%
    gather(key = type, value = count, -c(country, date))

## plot
top_ten_data_for_graph %>% ggplot(aes(x = date, y = count, fill = type)) +
    labs(title = "Cases in Top 10 Countries with Highest Confirmed Cases") +
    geom_area(alpha = 0.5) + scale_fill_manual(values = c("orange",
    "red", "purple")) + facet_wrap(~country, ncol = 3, scales = "free_y") +
    theme(legend.position = "bottom", legend.title = element_blank(),
        axis.title = element_blank(), plot.title = element_text(size = 10))
```



```r
## confirmed vs deaths plot
top_ten_data %>% ggplot(aes(x = confirmed, y = deaths,
    group = country, color = country)) + geom_line() +
    labs(title = "Confirmed vs Deaths - Top 10 Countries with Highest Confirmed Cases") +
    theme(legend.title = element_blank(), plot.title = element_text(size = 9))
```



6.2 Top Ten Countries - Death Rate

```r
## highest top 10
hdr_countries <- data_by_country[rev(order(data_by_country$death.rate)),
    ]
highest_death_rate_countries <- head(hdr_countries, 10, row.names = FALSE) %>%
    select(-c(date))
highest_death_rate_countries %>% kable(format = "pipe", align = "c",
    row.names = FALSE, caption = "Top 10 Countries with Highest Death Rates")
```

Table 14: Top 10 Countries with Highest Death Rates

country	confirmed	deaths	recovered	active	death.rate	recovered.rate
United Kingdom	1497135	54721	3310	1439104	94.29615	5.703848
Netherlands	486820	8946	6618	471256	57.47880	42.521203
Yemen	2093	608	1441	44	29.67301	70.326989
France	2178023	48593	156755	1972675	23.66373	76.336268
Spain	1556730	42619	150376	1363735	22.08296	77.917045
Mexico	1025969	100823	770728	154418	11.56823	88.431773
Western Sahara	10	1	8	1	11.11111	88.888889
Sudan	15839	1193	9727	4919	10.92491	89.075092
Syria	7154	372	3043	3739	10.89312	89.106881
Tanzania	509	21	183	305	10.29412	89.705882

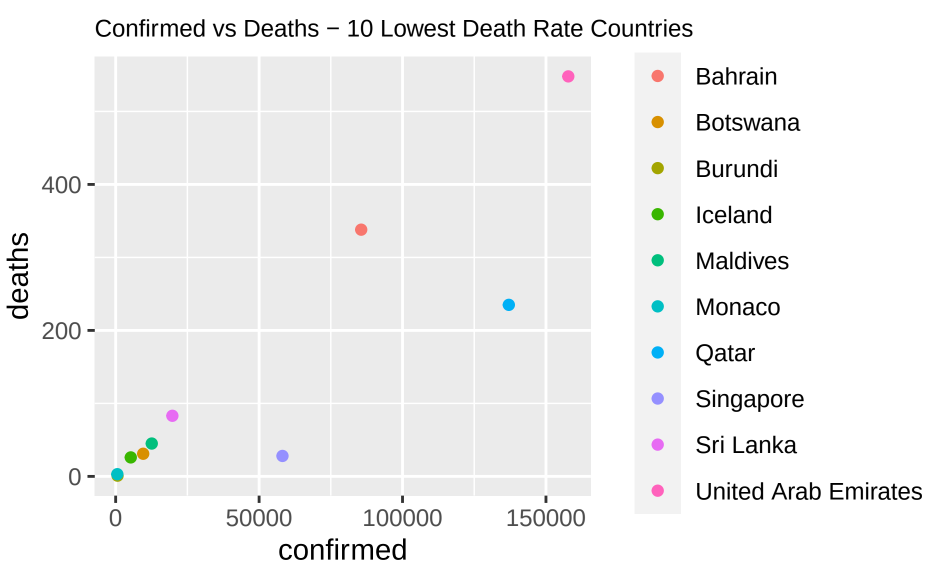
Recovered cases of UK, Netherlands are invalid. With the valid data, we see that Yemen has the highest death rate. But in general, European countries have the highest death rate.

```r
## lowest top 10
ldr_countries <- data_by_country %>% filter(death.rate != 0)
ldr_countries <- ldr_countries[order(ldr_countries$death.rate), ]
lowest_death_rate_countries <- head(ldr_countries, 10, row.names = FALSE) %>%
    select(-c(date))
lowest_death_rate_countries %>% kable(format = "pipe", align = "c",
    row.names = FALSE, caption = "Top 10 Countries with Lowest Death Rates")
```

Table 15: Top 10 Countries with Lowest Death Rates

country	confirmed	deaths	recovered	active	death.rate	recovered.rate
Singapore	58148	28	58064	56	0.0481994	99.95180
Burundi	656	1	575	80	0.1736111	99.82639
Qatar	137062	235	134100	2727	0.1749358	99.82506
United Arab Emirates	157785	548	148080	9157	0.3687058	99.63129
Maldives	12578	45	11559	974	0.3877973	99.61220
Bahrain	85591	338	83617	1636	0.4025966	99.59740
Botswana	9594	31	7296	2267	0.4230927	99.57691
Iceland	5269	26	5019	224	0.5153617	99.48464
Monaco	581	3	507	71	0.5882353	99.41176
Sri Lanka	19771	83	13590	6098	0.6070358	99.39296

```r
## plot confirmed vs death for lowest top 10
lowest_death_rate_countries %>% ggplot(aes(x = confirmed,
    y = deaths, group = country)) + geom_point(aes(color = country)) +
    labs(title = "Confirmed vs Deaths - 10 Lowest Death Rate Countries") +
    theme(legend.title = element_blank(), plot.title = element_text(size = 9))
```



The countries in *Table 15* have maintained very low death rates. Especially countries like Singapore, Qatar and UAE are quite successful considering Qatar and UAE had to deal with more than 100k cases and Singapore with more than 50k cases.

6.3 Top 10 Countries vs Rest of the World

In this section we will have a look at how rest of the world stands with respect to top 10 countries in terms of total confirmed cases.

```r
## total confirmed cases in top 10 countries
sum(data_top_ten$confirmed)
```

```
## [1] 38503659
```

```r
## data of rest of the countries
data_rest_countries <- data_by_country %>% filter(!(country %in% top_ten_countries))

## total confirmed cases in rest of the world
sum(data_rest_countries$confirmed)
```

```
## [1] 19639463
```

We see that top 10 countries contain ~66.5% of the total confirmed cases in the world where as rest of the world contains ~33.5% of the total cases. In other words, top 10 countries contain twice the confirmed cases as rest of the world.

7 China

China, the country where coronavirus broke out for the first time, has yet never appeared in any of the analysis carried out in this report. This could imply that the virus is contained in the country. Let us have a quick look at the current situation of China with respect to corona virus and check whether our implications stand true.

```
## total cases in China
data_by_country %>% filter(country == "China") %>% select(-c(date,
    country)) %>% kable(format = "pipe", caption = "Current Situation - China",
    row.names = FALSE, align = "c")
```

Table 16: Current Situation - China

confirmed	deaths	recovered	active	death.rate	recovered.rate
92037	4742	86769	526	5.181891	94.81811

There are still 559 active cases in China with ~5% death rate overall. Lets see the status for last 15 days.

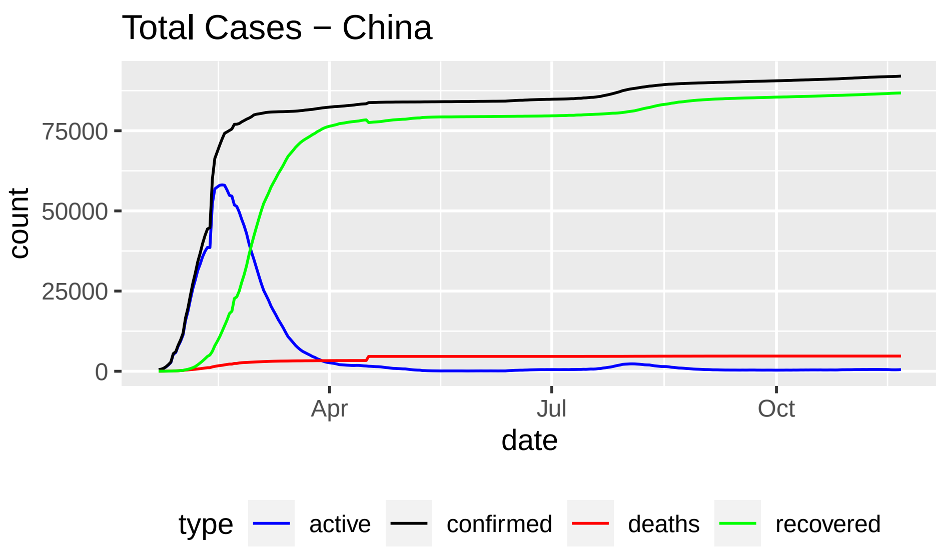
```
## filter data of China
china_data <- data %>% filter(country == "China") %>% mutate(active = confirmed -
    deaths - recovered) %>% select(-c(country))

## last 15 days data of China
china_data_recent <- tail(china_data, 15)
china_data_recent %>% kable(format = "pipe", align = "c", row.names = FALSE,
    caption = "Last 15 Days - China")
```

Table 17: Last 15 Days - China

date	confirmed	deaths	recovered	active
2020-11-07	91622	4741	86344	537
2020-11-08	91665	4741	86369	555
2020-11-09	91693	4741	86393	559
2020-11-10	91719	4742	86421	556
2020-11-11	91752	4742	86451	559
2020-11-12	91783	4742	86488	553
2020-11-13	91807	4742	86518	547
2020-11-14	91828	4742	86543	543
2020-11-15	91850	4742	86572	536
2020-11-16	91872	4742	86616	514
2020-11-17	91885	4742	86667	476
2020-11-18	91906	4742	86700	464
2020-11-19	91935	4742	86727	466
2020-11-20	91977	4742	86751	484
2020-11-21	92037	4742	86769	526

The virus is not completely contained in China but the increase in new cases are only in tens for the past 15 days. Since the past 15 days of data for China is not giving us the clear picture, let us have a look at the plot for the cases since breakout and try to analyze the situation.

```r
## gather data of China to plot graph
china_data_for_graph <- china_data %>% gather(key = type, value = count,
    -c(date))

## create plot
china_graph <- china_data_for_graph %>% ggplot(aes(x = date, y = count,
    color = type)) + geom_line() + labs(title = "Total Cases - China") +
    scale_color_manual(values = c("blue", "black", "red", "green")) +
    theme(legend.position = "bottom")

## display plot
china_graph
```



From the plot above, we notice that the cases in China has stabilized and the increase is negligible. The active cases are very low and China has successfully maintained this state from April.

8 Conclusions

Almost every country on the globe is impacted by the virus and as on the day of report generation, there are 191 countries with confirmed cases. Among these 191 countries, only 2 countries do not have active cases as of now. The death rate in the world because of the virus has almost stabilized between 3% to 4% and it could reduce going forward with time as there are improvements in medical care. China, where the corona virus broke out for the first time looks to have contained the virus. USA is still facing the heat with daily cases still over 100k. European countries are amidst the so called second wave but the death rates are very low compared to their first wave. The cases have started to decrease in India with close to 40k new cases daily, but it is still far away from being contained. The world is looking forward for the first vaccine to eradicate the corona virus, fingers crossed!

Though the data available is mostly accurate, there are some issues with the data. There were noticeable discrepancies especially in the recovered data either because of inaccurate data or missing entries. Few countries which have inaccurate data include United Kingdom, Netherlands, Sweden, Serbia. But it is to be noted that few other countries like Marshall Islands, Vanuatu actually do have 0 recovered cases as they are recently affected by the virus and the patients are yet to be recovered.

Since the recovered data of many countries are invalid, few of the analysis done in this report could have gone for a toss. For instance, in section 3.5, recovery rate could be still better and death rate could decrease if we had valid recovered data. Total active cases could also decrease drastically from 17.2M in the same section with valid data.

The data of all the countries with confirmed, active, deaths and recovered cases, recovery and death rates are listed below for the reference.

```
## data of all the countries
data_by_country %>% select(-c(date)) %>% kable(format = "latex", align = "c",
    booktabs = T, longtable = T, row.names = FALSE, caption = "Countrywise Data") %>%
    kable_styling(font_size = 7)
```

Table 18: Countrywise Data

country	confirmed	deaths	recovered	active	death.rate	recovered.rate
Afghanistan	44503	1675	35422	7406	4.5151899	95.484810
Albania	32196	685	15469	16042	4.2404358	95.759564
Algeria	73774	2255	48183	23336	4.4708355	95.529164
Andorra	6207	76	5290	841	1.4163250	98.583675
Angola	14413	336	7273	6804	4.4158234	95.584177
Antigua and Barbuda	139	4	128	7	3.0303030	96.969697
Argentina	1366182	36902	1187053	142227	3.0149801	96.985020
Armenia	124839	1931	92829	30079	2.0377797	97.962220
Australia	27821	907	25522	1392	3.4318362	96.568164
Austria	241962	2328	162751	76883	1.4102339	98.589766
Azerbaijan	89898	1107	62243	26548	1.7474349	98.252565
Bahamas	7395	163	5628	1604	2.8147125	97.185288
Bahrain	85591	338	83617	1636	0.4025966	99.597403
Bangladesh	445281	6350	360352	78579	1.7316513	98.268349
Barbados	255	7	241	7	2.8225806	97.177419
Belarus	122435	1089	102113	19233	1.0552121	98.944788
Belgium	556904	15522	0	541382	0.0000000	NA
Belize	5110	112	2800	2198	3.8461538	96.153846
Benin	2916	43	2579	294	1.6399695	98.360030
Bhutan	379	0	360	19	0.0000000	100.000000
Bolivia	143922	8904	119180	15838	6.9516880	93.048312
Bosnia and Herzegovina	79309	2246	43793	33270	4.8784726	95.121527
Botswana	9594	31	7296	2267	0.4230927	99.576907
Brazil	6052786	168989	5437189	446608	3.0143353	96.985665
Brunei	148	3	145	0	2.0270270	97.972973

Bulgaria	120697	2820	35752	82125	7.3110028	92.688997
Burkina Faso	2703	68	2521	114	2.6264967	97.373503
Burma	77848	1722	57679	18447	2.8989411	97.101059
Burundi	656	1	575	80	0.1736111	99.826389
Cabo Verde	10234	104	9649	481	1.0663386	98.933661
Cambodia	306	0	295	11	0.0000000	100.000000
Cameroon	23528	435	22177	916	1.9237573	98.076243
Canada	329084	11455	265568	52061	4.1350357	95.864964
Central African Republic	4911	63	1924	2924	3.1706090	96.829391
Chad	1633	101	1469	63	6.4331210	93.566879
Chile	539143	15030	514584	9529	2.8379159	97.162084
China	92037	4742	86769	526	5.1818907	94.818109
Colombia	1240493	35104	1144923	60466	2.9748472	97.025153
Comoros	596	7	572	17	1.2089810	98.791019
Congo (Brazzaville)	5632	93	3887	1652	2.3366834	97.663317
Congo (Kinshasa)	12180	327	11300	553	2.8124194	97.187581
Costa Rica	129418	1608	79372	48438	1.9856755	98.014325
Cote d'Ivoire	21126	129	20777	220	0.6170477	99.382952
Croatia	100410	1304	80027	19079	1.6033247	98.396675
Cuba	7798	132	7307	359	1.7744320	98.225568
Cyprus	8456	43	2055	6358	2.0495710	97.950429
Czechia	490750	7095	394830	88825	1.7652547	98.234745
Denmark	70152	781	54949	14422	1.4013996	98.598600
Diamond Princess	712	13	659	40	1.9345238	98.065476
Djibouti	5661	61	5552	48	1.0867629	98.913237
Dominica	72	0	55	17	0.0000000	100.000000
Dominican Republic	137770	2308	112090	23372	2.0175178	97.982482
Ecuador	184876	13139	160639	11098	7.5607960	92.439204
Egypt	112676	6535	101783	4358	6.0331616	93.966838
El Salvador	37250	1070	33984	2196	3.0524334	96.947567
Equatorial Guinea	5130	85	4975	70	1.6798419	98.320158
Eritrea	551	0	473	78	0.0000000	100.000000
Estonia	9375	87	5661	3627	1.5135699	98.486430
Eswatini	6205	119	5850	236	1.9936338	98.006366
Ethiopia	105352	1636	65534	38182	2.4356111	97.564389
Fiji	35	2	33	0	5.7142857	94.285714
Finland	21216	375	15300	5541	2.3923445	97.607656
France	2178023	48593	156755	1972675	23.6637318	76.336268
Gabon	9131	59	8976	96	0.6530160	99.346984
Gambia	3726	123	3582	21	3.3198381	96.680162
Georgia	100684	927	81783	17974	1.1207835	98.879216
Germany	927990	14061	600991	312938	2.2861482	97.713852
Ghana	50717	323	49281	1113	0.6511572	99.348843
Greece	90121	1527	23074	65520	6.2070648	93.792935
Grenada	41	0	30	11	0.0000000	100.000000
Guatemala	118417	4074	107241	7102	3.6598841	96.340116
Guinea	12798	75	11821	902	0.6304640	99.369536
Guinea-Bissau	2421	43	2286	92	1.8462860	98.153714
Guyana	5093	143	4018	932	3.4366739	96.563326
Haiti	9214	232	7854	1128	2.8691566	97.130843
Holy See	27	0	15	12	0.0000000	100.000000
Honduras	104435	2857	46208	55370	5.8228880	94.177112
Hungary	170298	3689	40820	125789	8.2882114	91.711789
Iceland	5269	26	5019	224	0.5153617	99.484638
India	9095806	133227	8521617	440962	1.5393345	98.460666
Indonesia	493308	15774	413955	63579	3.6706855	96.329314
Iran	841308	44327	596136	200845	6.9210868	93.078913
Iraq	533555	11925	463040	58590	2.5107113	97.489289
Ireland	70143	2022	23364	44757	7.9650201	92.034980
Israel	328397	2757	317171	8469	0.8617564	99.138244
Italy	1380531	49261	539524	791746	8.3665515	91.633449
Jamaica	10240	237	5481	4522	4.1448059	95.855194
Japan	130871	1932	108981	19958	1.7419058	98.258094
Jordan	178161	2172	108353	67636	1.9651663	98.034834
Kazakhstan	125466	1945	112292	11229	1.7026007	98.297399

Kenya	76404	1366	51352	23686	2.5911453	97.408855
Korea, South	30733	505	26466	3762	1.8723814	98.127619
Kosovo	33588	904	20493	12191	4.2248913	95.775109
Kuwait	139734	863	131560	7311	0.6516995	99.348300
Kyrgyzstan	69149	1227	60294	7628	1.9944409	98.005559
Laos	25	0	23	2	0.0000000	100.000000
Latvia	12744	153	1557	11034	8.9473684	91.052632
Lebanon	115283	894	66135	48254	1.3337511	98.666249
Lesotho	2085	44	1276	765	3.3333333	96.666667
Liberia	1551	82	1331	138	5.8032555	94.196744
Libya	76808	1068	47587	28153	2.1950468	97.804953
Liechtenstein	1109	8	883	218	0.8978676	99.102132
Lithuania	44740	374	10869	33497	3.3265143	96.673486
Luxembourg	29762	260	20014	9488	1.2824307	98.717569
Madagascar	17310	250	16592	468	1.4843843	98.515616
Malawi	6003	185	5441	377	3.2883043	96.711696
Malaysia	53679	332	40493	12854	0.8132272	99.186773
Maldives	12578	45	11559	974	0.3877973	99.612203
Mali	4206	143	3012	1051	4.5324881	95.467512
Malta	8822	111	6565	2146	1.6626723	98.337328
Marshall Islands	4	0	1	3	0.0000000	100.000000
Mauritania	8096	169	7573	354	2.1828985	97.817102
Mauritius	494	10	433	51	2.2573363	97.742664
Mexico	1025969	100823	770728	154418	11.5682272	88.431773
Moldova	96689	2130	78112	16447	2.6544702	97.345530
Monaco	581	3	507	71	0.5882353	99.411765
Mongolia	608	0	337	271	0.0000000	100.000000
Montenegro	30653	434	19129	11090	2.2184736	97.781526
Morocco	320962	5256	266711	48995	1.9325874	98.067413
Mozambique	14981	123	13151	1707	0.9266235	99.073376
MS Zaandam	9	2	0	7	0.0000000	NA
Namibia	13811	145	13209	457	1.0858170	98.914183
Nepal	218639	1305	193325	24009	0.6705030	99.329497
Netherlands	486820	8946	6618	471256	57.4787972	42.521203
New Zealand	2028	25	1953	50	1.2639029	98.736097
Nicaragua	5725	159	4225	1341	3.6268248	96.373175
Niger	1351	70	1158	123	5.7003257	94.299674
Nigeria	66228	1166	61884	3178	1.8493259	98.150674
North Macedonia	53631	1487	31659	20485	4.4862125	95.513788
Norway	32352	306	17998	14048	1.6717657	98.328234
Oman	121360	1365	112406	7589	1.1997785	98.800222
Pakistan	374173	7662	329828	36683	2.2702895	97.729710
Panama	153577	2946	134360	16271	2.1455727	97.854427
Papua New Guinea	604	7	586	11	1.1804384	98.819562
Paraguay	75857	1652	53988	20217	2.9690870	97.030913
Peru	948081	35549	873970	38562	3.9085495	96.091450
Philippines	416852	8080	375548	33224	2.1062071	97.893793
Poland	843475	13288	407075	423112	3.1610774	96.838923
Portugal	255970	3824	169379	82767	2.2078140	97.792186
Qatar	137062	235	134100	2727	0.1749358	99.825064
Romania	412808	9916	287310	115582	3.3361819	96.663818
Russia	2047563	35442	1564897	447224	2.2146558	97.785344
Rwanda	5620	46	5163	411	0.8830870	99.116913
Saint Kitts and Nevis	19	0	19	0	0.0000000	100.000000
Saint Lucia	204	2	60	142	3.2258065	96.774193
Saint Vincent and the Grenadines	84	0	78	6	0.0000000	100.000000
San Marino	1395	43	1082	270	3.8222222	96.177778
Sao Tome and Principe	979	17	920	42	1.8143010	98.185699
Saudi Arabia	355034	5761	342882	6391	1.6524066	98.347593
Senegal	15865	330	15465	70	2.0892688	97.910731
Serbia	116125	1168	0	114957	0.0000000	NA
Seychelles	163	0	159	4	0.0000000	100.000000
Sierra Leone	2405	74	1827	504	3.8926881	96.107312
Singapore	58148	28	58064	56	0.0481994	99.951801
Slovakia	95257	644	43188	51425	1.4692462	98.530754

Slovenia	64284	1026	43272	19986	2.3161317	97.683868
Solomon Islands	16	0	5	11	0.0000000	100.000000
Somalia	4382	108	3384	890	3.0927835	96.907217
South Africa	765409	20845	707784	36780	2.8608524	97.139148
South Sudan	3047	60	1290	1697	4.4444444	95.555556
Spain	1556730	42619	150376	1363735	22.0829555	77.917045
Sri Lanka	19771	83	13590	6098	0.6070358	99.392964
Sudan	15839	1193	9727	4919	10.9249084	89.075092
Suriname	5295	116	5166	13	2.1961378	97.803862
Sweden	208295	6406	0	201889	0.0000000	NA
Switzerland	290601	4031	180700	105870	2.1820918	97.817908
Syria	7154	372	3043	3739	10.8931186	89.106881
Taiwan*	611	7	546	58	1.2658228	98.734177
Tajikistan	11854	86	11229	539	0.7600530	99.239947
Tanzania	509	21	183	305	10.2941176	89.705882
Thailand	3913	60	3761	92	1.5702696	98.429730
Timor-Leste	30	0	30	0	0.0000000	100.000000
Togo	2829	64	2209	556	2.8156621	97.184338
Trinidad and Tobago	6324	115	5559	650	2.0267889	97.973211
Tunisia	87471	2752	61402	23317	4.2896780	95.710322
Turkey	440805	12219	370825	57761	3.1899730	96.810027
Uganda	17667	168	8611	8888	1.9136576	98.086342
Ukraine	629850	11149	294851	323850	3.6434641	96.356536
United Arab Emirates	157785	548	148080	9157	0.3687058	99.631294
United Kingdom	1497135	54721	3310	1439104	94.2961521	5.703848
Uruguay	4564	69	3621	874	1.8699187	98.130081
US	12088410	255861	4529700	7302849	5.3465205	94.653480
Uzbekistan	71431	603	68659	2169	0.8706073	99.129393
Vanuatu	1	0	0	1	0.0000000	NA
Venezuela	99435	869	94355	4211	0.9125851	99.087415
Vietnam	1306	35	1142	129	2.9736619	97.026338
West Bank and Gaza	70254	620	58383	11251	1.0507940	98.949206
Western Sahara	10	1	8	1	11.1111111	88.888889
Yemen	2093	608	1441	44	29.6730112	70.326989
Zambia	17394	356	16659	379	2.0922715	97.907729
Zimbabwe	9172	265	8235	672	3.1176471	96.882353