covid-19-analysis

Rujvi Doppalapudi

2023-07-18

R Markdown

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

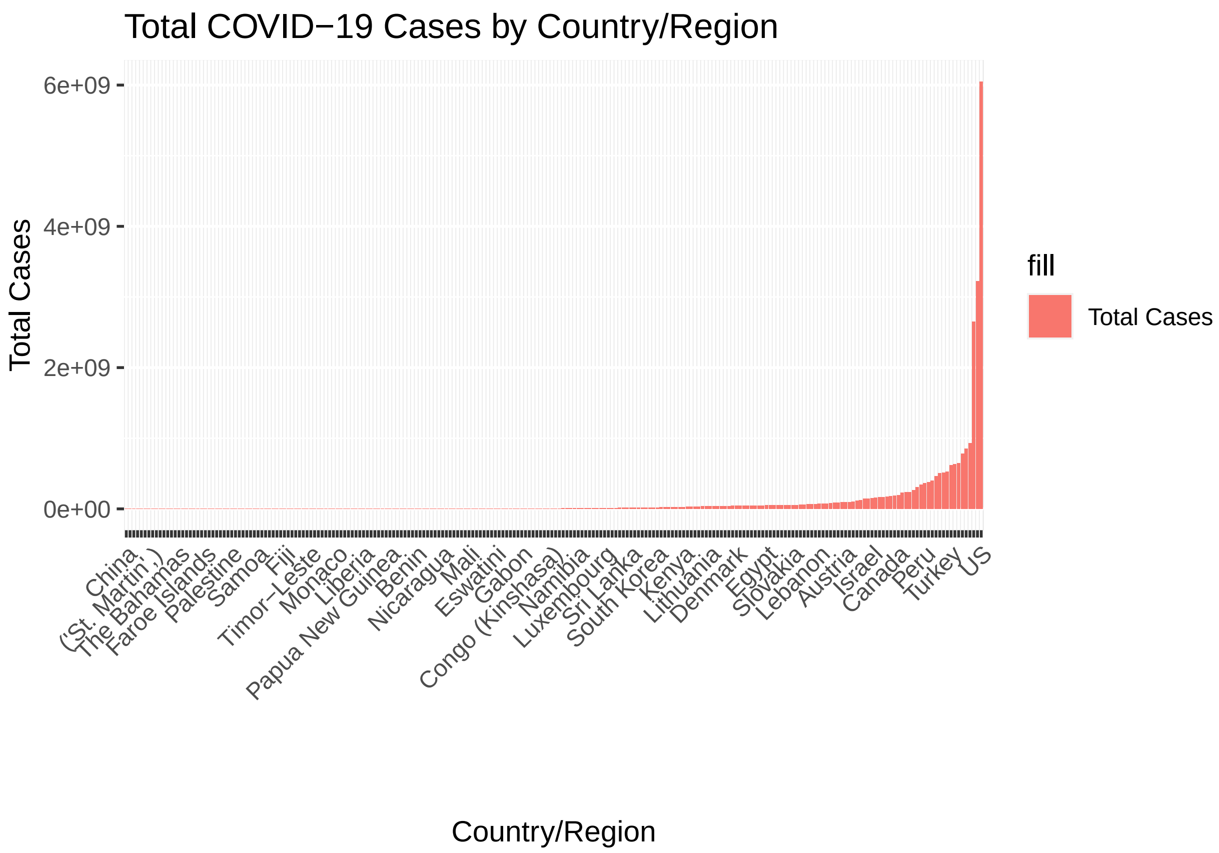
```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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 error
```

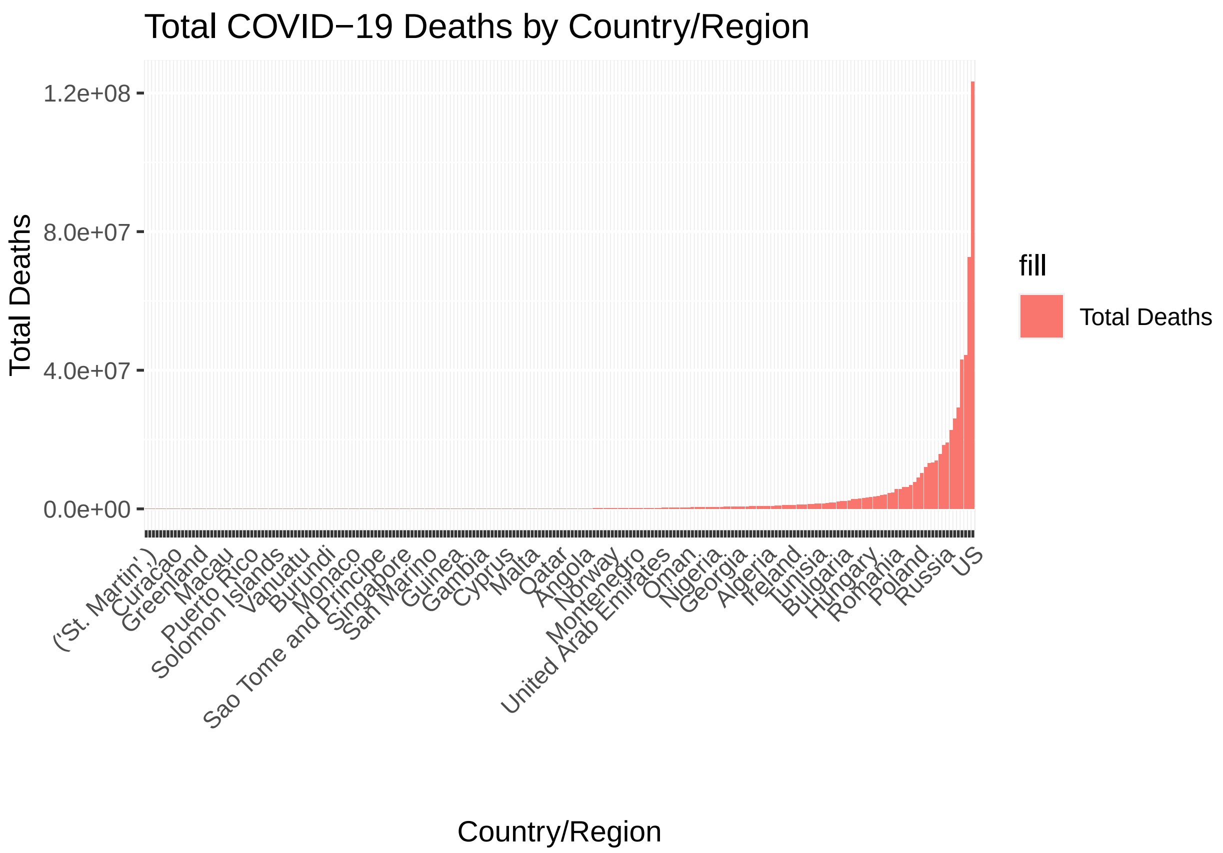
```
# Replace "path/to/covid_19_data.csv" with the actual path to the CSV file
covid_data <- read_csv("covid_19_data.csv")
```

```
## Rows: 306429 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): ObservationDate, Province/State, Country/Region, Last Update
## dbl (4): SNo, Confirmed, Deaths, Recovered
##
## 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.
```

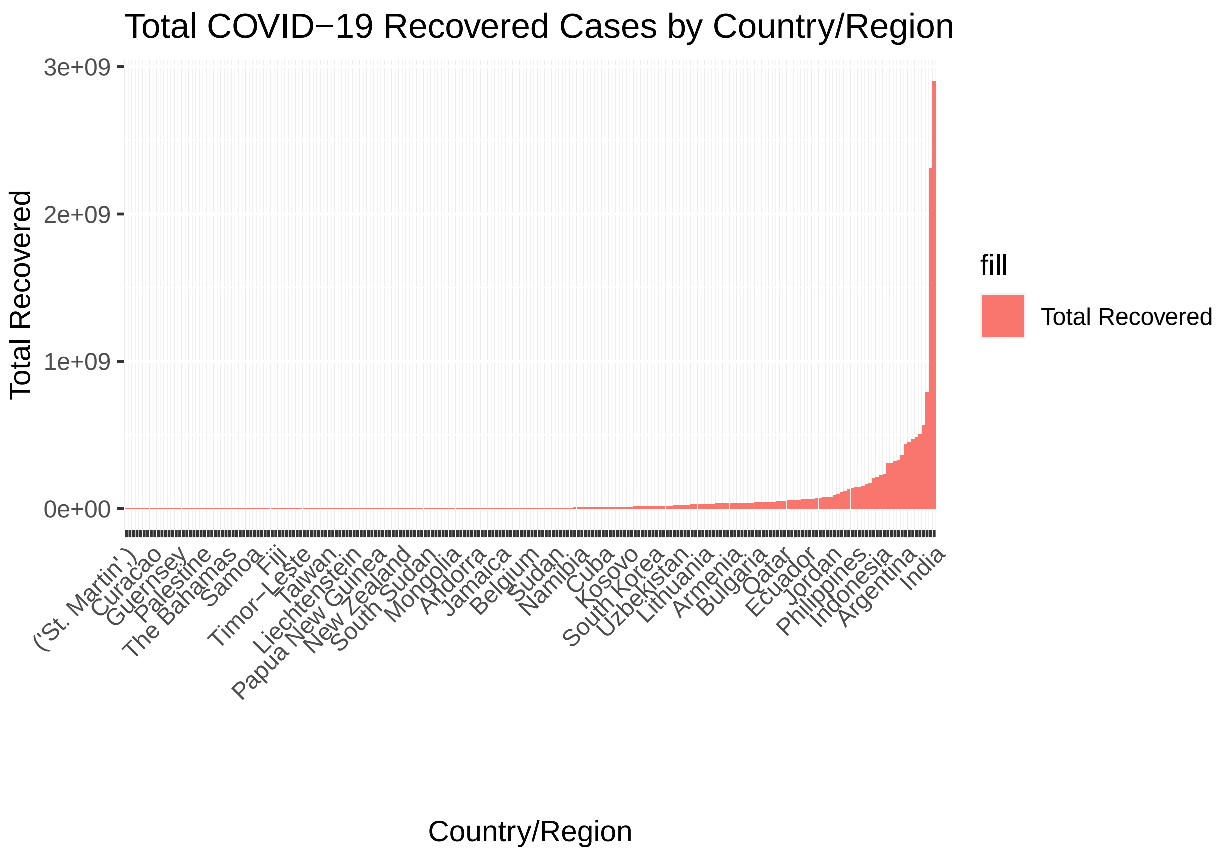
```r
# Convert the ObservationDate column to a date object
covid_data$ObservationDate <- as.Date(covid_data$ObservationDate, format = "%m/%d/%Y")


# Summarize the data by country/region and calculate the total cases, deaths, and recovered
country_data <- covid_data %>%
  group_by(`Country/Region`) %>%
  summarise(TotalCases = sum(Confirmed),
            TotalDeaths = sum(Deaths),
            TotalRecovered = sum(Recovered)) %>%
  arrange(desc(TotalCases))

# Bar plot for total cases, deaths, and recovered
ggplot(country_data, aes(x = reorder(`Country/Region`, TotalCases), y = TotalCases, fill = "Total Cases
  geom_bar(stat = "identity") + scale_x_discrete(guide = guide_axis(check.overlap = TRUE)) +
  labs(title = "Total COVID-19 Cases by Country/Region",
       x = "Country/Region",
       y = "Total Cases")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
ggplot(country_data, aes(x = reorder(`Country/Region`, TotalDeaths), y = TotalDeaths, fill = "Total Dea
  geom_bar(stat = "identity") +
  scale_x_discrete(guide = guide_axis(check.overlap = TRUE)) +
  labs(title = "Total COVID-19 Deaths by Country/Region",
       x = "Country/Region",
       y = "Total Deaths") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
ggplot(country_data, aes(x = reorder(`Country/Region`, TotalRecovered), y = TotalRecovered, fill = "Tota
  geom_bar(stat = "identity") +
  scale_x_discrete(guide = guide_axis(check.overlap = TRUE)) +
  labs(title = "Total COVID-19 Recovered Cases by Country/Region",
       x = "Country/Region",
       y = "Total Recovered") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Total COVID−19 Recovered Cases by Country/Region
Total Recovered
3e+09
2e+09
1e+09
0e+00
fill
Total Recovered
('St. Martin',)
Curacao
Guernsey
Palestine
The Bahamas
Samoa
Fiji
Timor−Leste
Taiwan.
Liechtenstein
Papua New Guinea
New Zealand
South Sudan
Mongolia
Andorra
Jamaica
Belgium
Sudan.
Namibia
Cuba
Kosovo
South Korea
Uzbekistan.
Lithuania
Armenia.
Bulgaria
Qatar
Ecuador
Jordan
Philippines.
Indonesia
Argentina
India
Country/Region