COVID19

J. Achalapong

2025-07-19

Step1: Import Data

This section initiates the project by importing the necessary packages and reading in four datasets from the Johns Hopkins University CSSE COVID-19 GitHub repository. The datasets include global and US COVID-19 confirmed cases and deaths. The data is loaded into R using read_csv() for further processing.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)

url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_covid_19_da

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

urls <- str_c(url_in,file_names)
urls
```

```
## [1] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_covid_19_data/
## [2] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_covid_19_data/
## [3] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_covid_19_data/
## [4] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_covid_19_data/
```

Let's read in the data and see what we have.

```
global_cases <- read_csv(urls[1])
global_deaths <- read_csv(urls[2])
us_cases <- read_csv(urls[3])
us_deaths <- read_csv(urls[4])
```

Step2: Tidy and Transform Data

The raw datasets are in a wide format with date columns spread across. To prepare for analysis, the data is reshaped into a long format using `pivot_longer()`. Columns such as latitude and longitude are removed, and death and case data are joined together. `Dates` are parsed into Date objects. For US data, the same transformations are applied, and a `Combined_Key` is used to merge data with a population lookup table to facilitate normalization of case and death counts.

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

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

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

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

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:330327      Length:330327      Min.   :2020-01-22   Min.   :        0
##  Class :character   Class :character   1st Qu.:2020-11-02   1st Qu.:      680
##  Mode  :character   Mode  :character   Median :2021-08-15   Median :    14429
##                                        Mean   :2021-08-15   Mean   :   959384
##                                        3rd Qu.:2022-05-28   3rd Qu.:   228517
##                                        Max.   :2023-03-09   Max.   :103802702
##      deaths
##  Min.   :      0
##  1st Qu.:      3
##  Median :    150
##  Mean   :  13380
##  3rd Qu.:   3032
##  Max.   :1123836
```

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

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

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

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

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

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

uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_cov

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

```
## Rows: 4321 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): iso2, iso3, FIPS, Admin2, Province_State, Country_Region, Combined_Key
## dbl (5): UID, code3, Lat, Long_, Population
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

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

global
```

```
## # A tibble: 306,827 x 7
##    Province_State Country_Region date       cases deaths Population Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl> <chr>
##  1 <NA>           Afghanistan    2020-02-24     5      0   38928341 Afghanistan
##  2 <NA>           Afghanistan    2020-02-25     5      0   38928341 Afghanistan
##  3 <NA>           Afghanistan    2020-02-26     5      0   38928341 Afghanistan
##  4 <NA>           Afghanistan    2020-02-27     5      0   38928341 Afghanistan
##  5 <NA>           Afghanistan    2020-02-28     5      0   38928341 Afghanistan
##  6 <NA>           Afghanistan    2020-02-29     5      0   38928341 Afghanistan
##  7 <NA>           Afghanistan    2020-03-01     5      0   38928341 Afghanistan
##  8 <NA>           Afghanistan    2020-03-02     5      0   38928341 Afghanistan
##  9 <NA>           Afghanistan    2020-03-03     5      0   38928341 Afghanistan
## 10 <NA>           Afghanistan    2020-03-04     5      0   38928341 Afghanistan
## # i 306,817 more rows
```

Step3: Visualizing Data

This section uses `ggplot2` to plot the number of cases and deaths over time in the United States, both at the national and state levels (e.g., New York). Logarithmic scaling is applied to the y-axis to handle the wide range of case and death counts. The graphs help reveal trends and highlight spikes in the pandemic timeline.

```
us_by_state <- us %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(death_per_mill = deaths *1000000 / Population) %>%
  select(Province_State, Country_Region, date,
         cases, deaths, death_per_mill, Population) %>%
  ungroup()
```

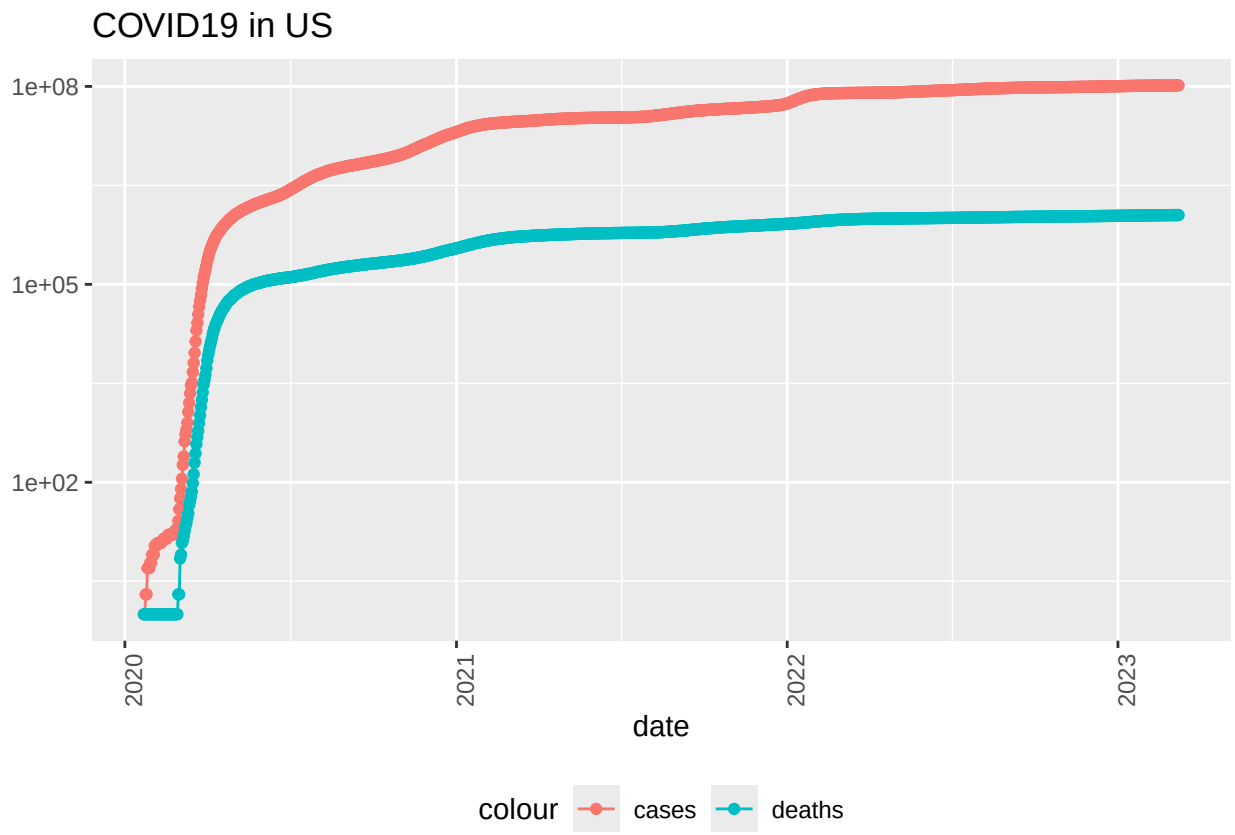
```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can
## override using the `.groups` argument.
```

```
us_total <- us_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(death_per_mill = deaths *1000000 / Population) %>%
  select(Country_Region, date,
         cases, deaths, death_per_mill, Population) %>%
  ungroup()
```

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

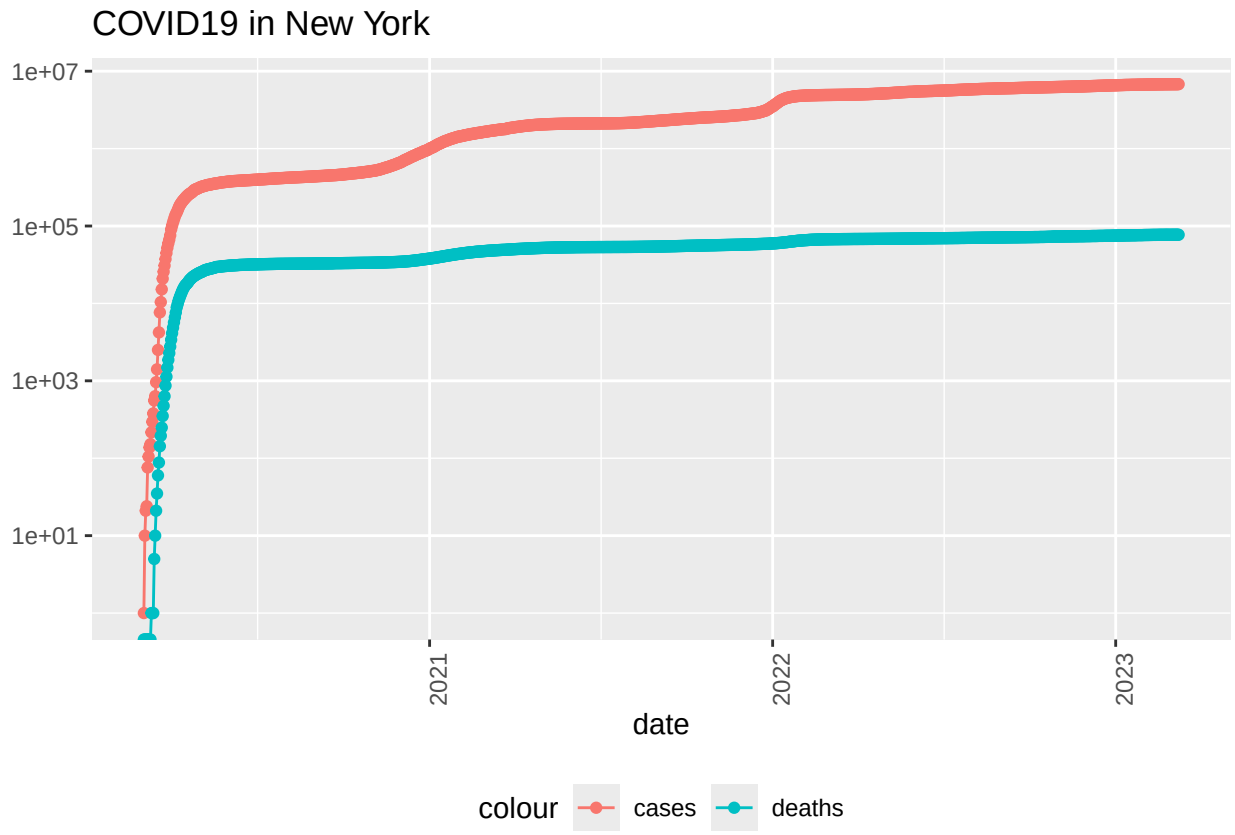
```
us_total %>%
  filter(cases > 0 ) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(colour = "cases")) +
  geom_point(aes(colour = "cases")) +
```

```
  geom_line(aes(y = deaths, colour = "deaths")) +
  geom_point(aes(y = deaths, colour = "deaths")) +
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y = NULL)
```



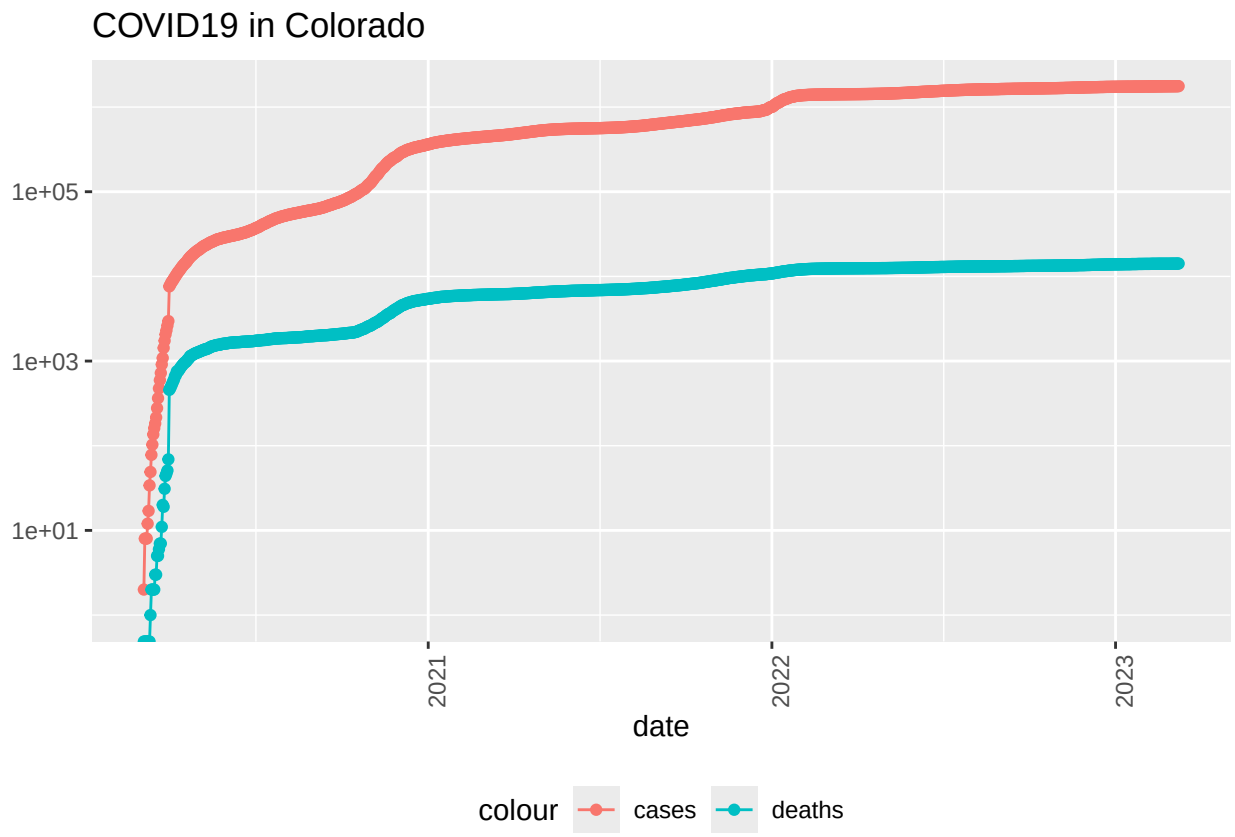

```
state <- "New York"
us_by_state %>%
  filter(Province_State == state) %>%
  filter(cases > 0 ) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(colour = "cases")) +
  geom_point(aes(colour = "cases")) +
  geom_line(aes(y = deaths, colour = "deaths")) +
  geom_point(aes(y = deaths, colour = "deaths")) +
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", state), y = NULL)
```

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.
```




```
state <- "Colorado"
us_by_state %>%
  filter(Province_State == state) %>%
  filter(cases > 0 ) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(colour = "cases")) +
  geom_point(aes(colour = "cases")) +
  geom_line(aes(y = deaths, colour = "deaths")) +
  geom_point(aes(y = deaths, colour = "deaths")) +
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", state), y = NULL)
```

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.
```

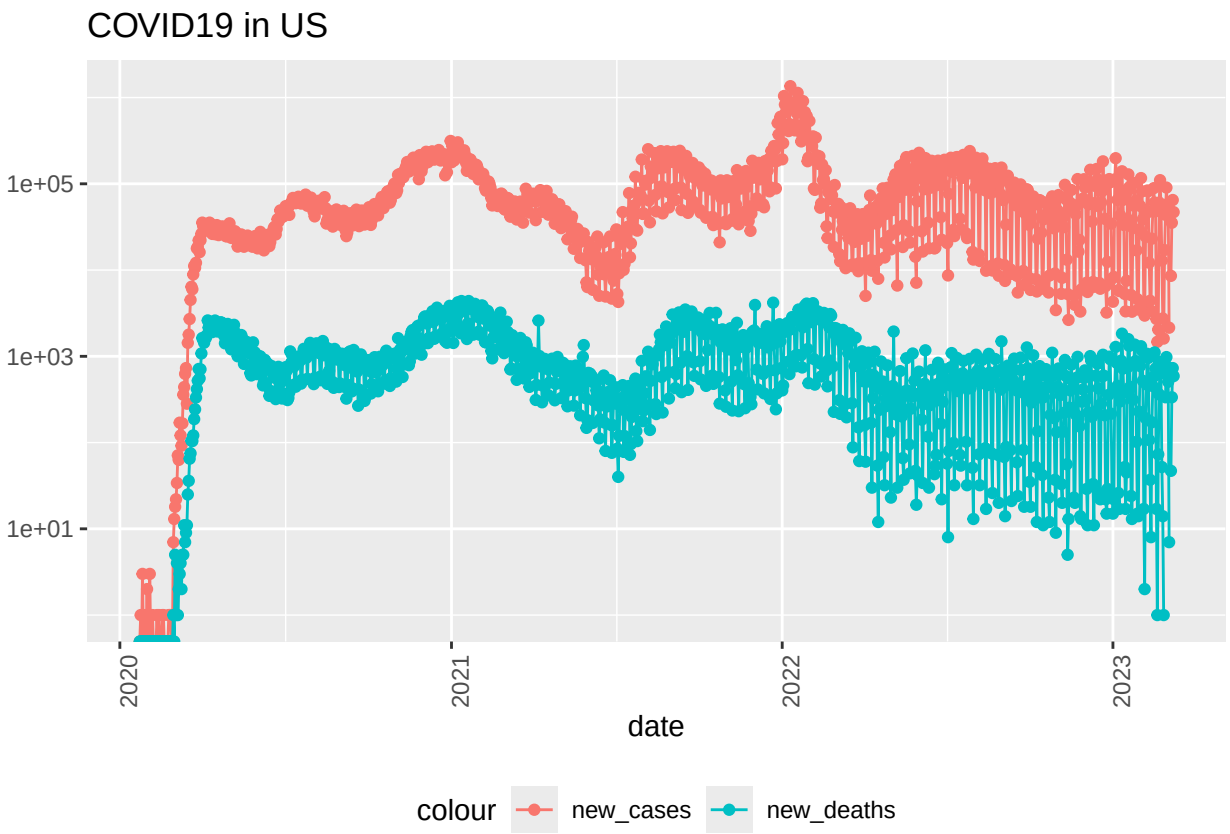


Step4: Analyzing Data

New daily case and death counts are computed using the `lag()` function. These derived variables (`new_cases`, `new_deaths`) are plotted to visualize the progression and waves of the pandemic. Warnings indicate missing or infinite values due to log-scaling and differences involving zeros.

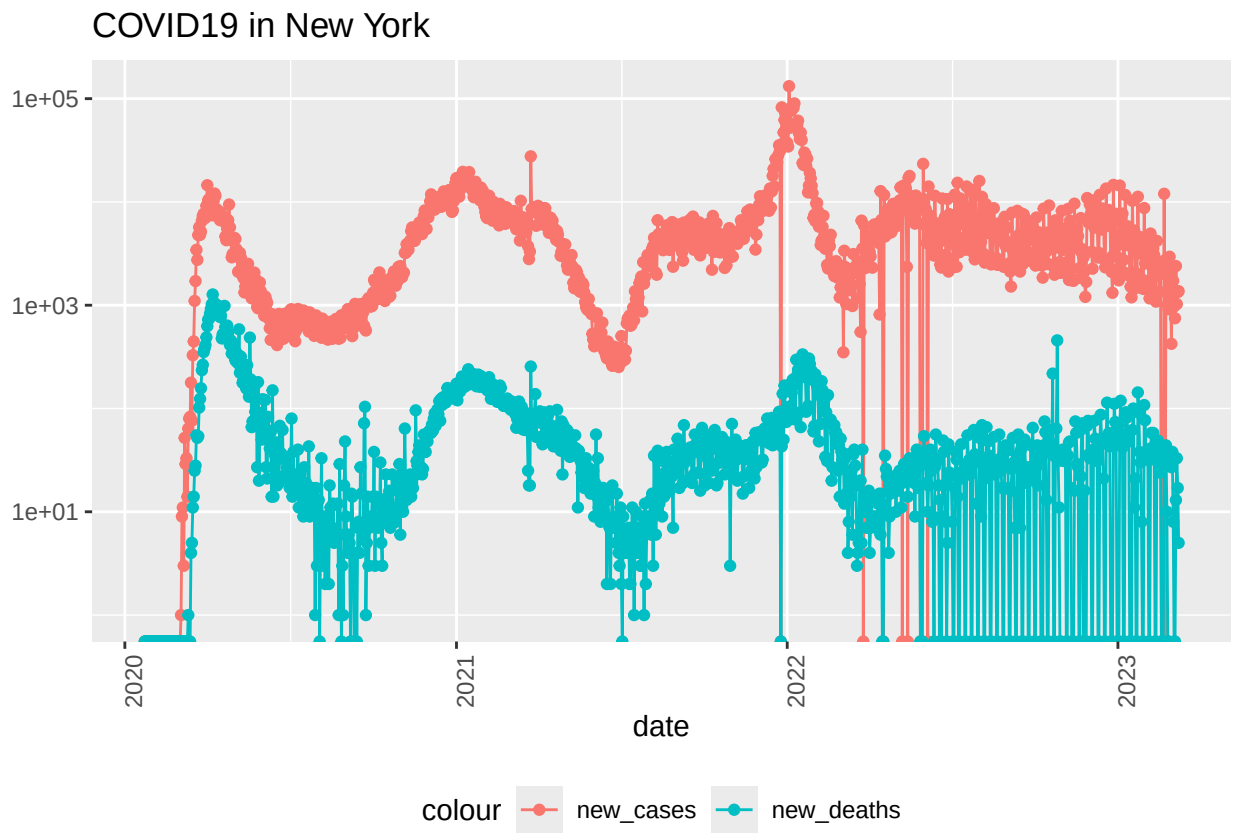
```
us_by_state <- us_by_state %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))

us_total <- us_total %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))

us_total %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(colour = "new_cases")) +
  geom_point(aes(colour = "new_cases")) +
  geom_line(aes(y = new_deaths, colour = "new_deaths")) +
  geom_point(aes(y = new_deaths, colour = "new_deaths")) +
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y = NULL)
```

```
## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

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

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).

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

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
state <- "New York"
us_by_state %>%
  filter(Province_State == state) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(colour = "new_cases")) +
  geom_point(aes(colour = "new_cases")) +
  geom_line(aes(y = new_deaths, colour = "new_deaths")) +
  geom_point(aes(y = new_deaths, colour = "new_deaths")) +
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", state), y = NULL)
```

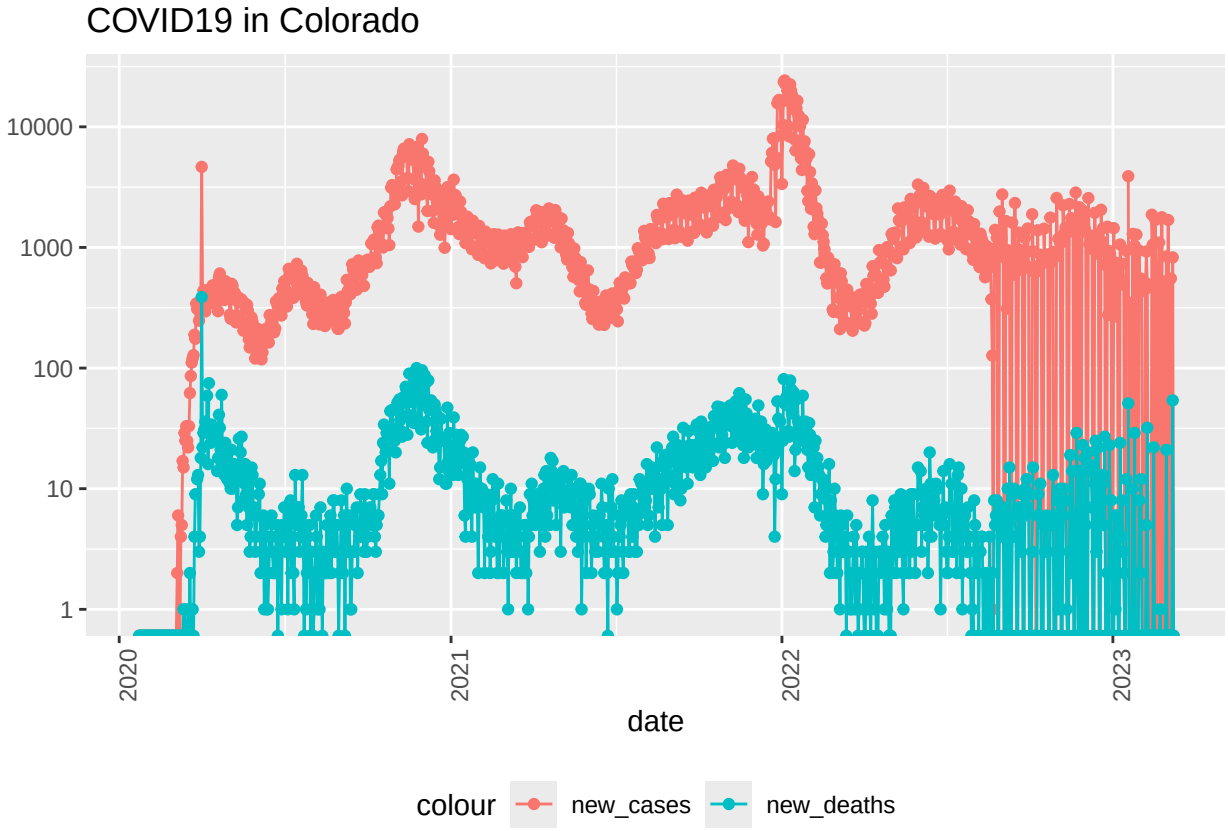
```
## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

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

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

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

## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
state <- "Colorado"
us_by_state %>%
  filter(Province_State == state) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(colour = "new_cases")) +
  geom_point(aes(colour = "new_cases")) +
  geom_line(aes(y = new_deaths, colour = "new_deaths")) +
  geom_point(aes(y = new_deaths, colour = "new_deaths")) +
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", state), y = NULL)
```

```
## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

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

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).

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

## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
us_state_totals <- us_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            Population = max(Population),
            cases_per_thou = 1000* cases / Population,
            deaths_per_thou = 1000* deaths / Population) %>%
  filter(cases > 0, Population > 0)
```

```
us_state_totals %>%
  slice_min(deaths_per_thou, n=10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State        deaths  cases Population
##              <dbl>          <dbl> <chr>                 <dbl>  <dbl>      <dbl>
##  1           0.611           150. American Samoa           34 8.32e3      55641
##  2           0.744           248. Northern Mariana Isl~    41 1.37e4      55144
##  3           1.21            231. Virgin Islands          130 2.48e4     107268
##  4           1.30            269. Hawaii                 1841 3.81e5    1415872
##  5           1.49            245. Vermont                 929 1.53e5     623989
##  6           1.55            293. Puerto Rico            5823 1.10e6    3754939
##  7           1.65            340. Utah                   5298 1.09e6    3205958
##  8           2.01            415. Alaska                 1486 3.08e5     740995
##  9           2.03            252. District of Columbia   1432 1.78e5     705749
## 10           2.06            253. Washington            15683 1.93e6    7614893
```

```
us_state_totals %>%
  slice_max(deaths_per_thou, n=10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State deaths   cases Population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            4.55           336. Arizona         33102 2443514    7278717
##  2            4.54           326. Oklahoma        17972 1290929    3956971
##  3            4.49           333. Mississippi     13370  990756    2976149
##  4            4.44           359. West Virginia    7960  642760    1792147
##  5            4.32           320. New Mexico       9061  670929    2096829
##  6            4.31           334. Arkansas        13020 1006883    3017804
##  7            4.29           335. Alabama         21032 1644533    4903185
##  8            4.28           368. Tennessee       29263 2515130    6829174
##  9            4.23           307. Michigan        42205 3064125    9986857
## 10            4.06           385. Kentucky        18130 1718471    4467673
```

Step5: Modeling Data

A simple linear regression is conducted to model the relationship between cases per thousand (`cases_per_thou`) and deaths per thousand (`deaths_per_thou`) using `lm()`. The model shows a statistically significant positive relationship, suggesting that states with more cases per capita tend to have more deaths per capita. The model's predictions are visualized alongside the actual data points.

```
mod <- lm(deaths_per_thou ~ cases_per_thou, data = us_state_totals)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = us_state_totals)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.3352 -0.5978  0.1491  0.6535  1.2086
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.36167    0.72480  -0.499     0.62
## cases_per_thou  0.01133    0.00232   4.881 9.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8615 on 54 degrees of freedom
## Multiple R-squared:  0.3061, Adjusted R-squared:  0.2933
## F-statistic: 23.82 on 1 and 54 DF,  p-value: 9.763e-06
```

```
us_state_totals %>% slice_min(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths cases Population cases_per_thou deaths_per_thou
##   <chr>           <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1 American Samoa     34  8320      55641           150.           0.611
```

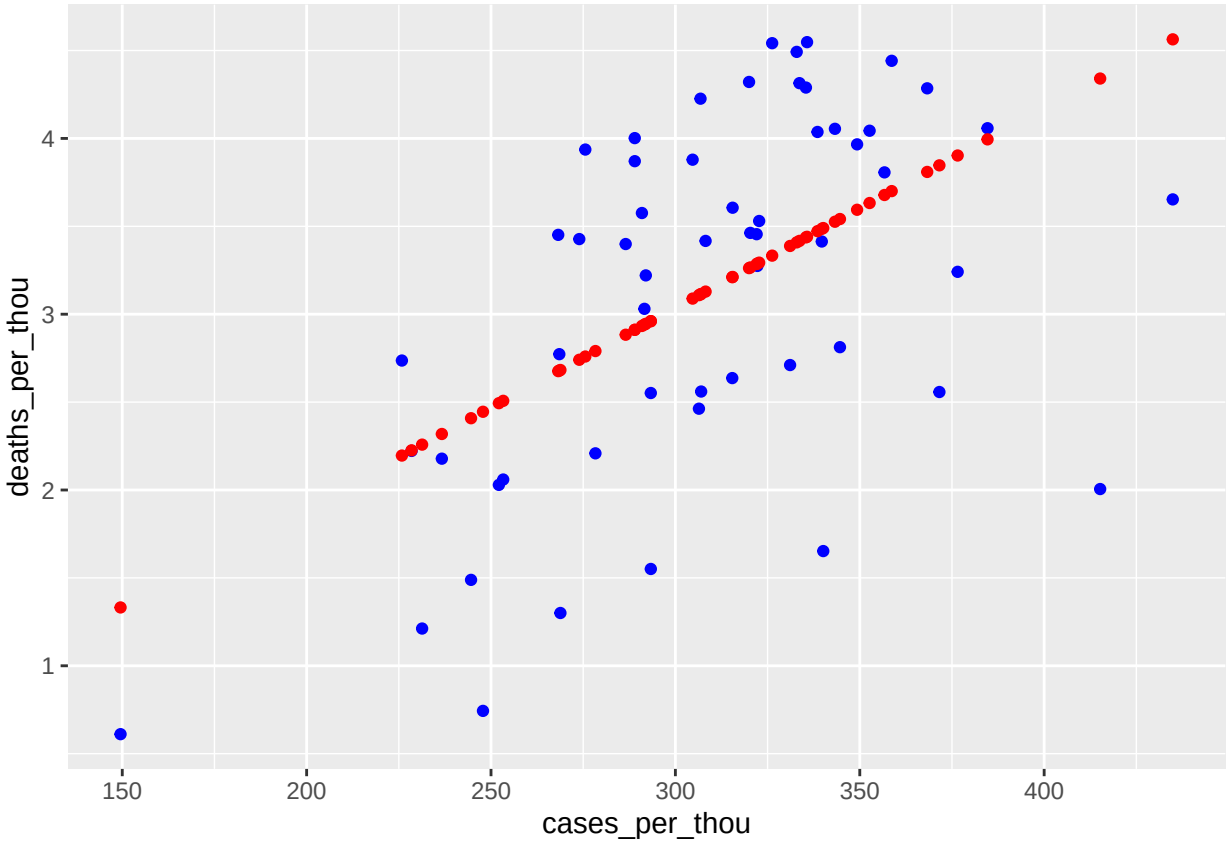
```
us_state_totals %>% slice_max(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths  cases Population cases_per_thou deaths_per_thou
##   <chr>           <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
## 1 Rhode Island     3870 460697    1059361           435.            3.65
```

```
x_grid <- seq(1,451)
new_df <- tibble(cases_per_thou = x_grid)
us_state_totals %>% mutate(pred = predict(mod))
```

```
## # A tibble: 56 x 7
##    Province_State  deaths  cases Population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          21032 1.64e6    4903185           335.            4.29   3.44
##  2 Alaska            1486 3.08e5     740995           415.            2.01   4.34
##  3 American Samoa      34 8.32e3      55641           150.            0.611  1.33
##  4 Arizona          33102 2.44e6    7278717           336.            4.55   3.44
##  5 Arkansas         13020 1.01e6    3017804           334.            4.31   3.42
##  6 California      101159 1.21e7   39512223           307.            2.56   3.12
##  7 Colorado         14181 1.76e6    5758736           306.            2.46   3.11
##  8 Connecticut      12220 9.77e5    3565287           274.            3.43   2.74
##  9 Delaware          3324 3.31e5     973764           340.            3.41   3.49
## 10 District of Co~   1432 1.78e5     705749           252.            2.03   2.49
## # i 46 more rows
```

```
us_tot_w_pred <- us_state_totals %>% mutate(pred = predict(mod))
us_tot_w_pred %>% ggplot() +
  geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue") +
  geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```



Step 6: Report Conclusion and Sources of Bias

This analysis provides a high-level view of the COVID-19 pandemic in the United States and globally, offering insights into trends, state-wise impacts, and the relationship between infection and mortality rates.

Key conclusions

- The progression of cases and deaths followed a pattern of waves.
- Some states experienced much higher case and death rates per capita.
- There is a significant linear relationship between case incidence and death incidence.

Potential sources of bias and limitations

- Underreporting or inconsistent testing: COVID-19 case counts depend heavily on testing availability and public health reporting systems.
- Population data limitations: Mismatches between case/death data and population data could introduce bias in per-capita calculations.
- Data lag and reporting delays: These affect the accuracy of daily case and death counts, especially visible in the spikes and drops.

- Policy differences: Differences in public health measures, reporting standards, and healthcare infrastructure across states may confound the observed relationships.
- Simplistic modeling: The linear regression used does not account for other influential factors such as age demographics, vaccination rates, or comorbidities.