Data Science as a Field: COVID 19

null

2025-03-02

Introduction

The COVID-19 pandemic has had a profound impact worldwide, with varying infection and mortality rates across different regions. Understanding the relationship between cases and deaths can provide insights into mortality risk and healthcare system effectiveness. This study examines COVID-19 trends in both U.S. states and globally.

Objective

We hypothesize that an increase in cases will be associated with an increase in deaths, given the expected mortality risk associated with the virus. By using linear regression models, we aim to assess the strength and significance of this relationship and determine whether population size influences mortality rates. This study will help evaluate the effectiveness of basic statistical models in explaining COVID-19 fatality trends and provide insights into potential factors affecting mortality rates.

```
# Install required libraries
# You can using this just copy past into your R studio if you don't have these already installed: insta

#Required libraries
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3

## Warning: package 'ggplot2' was built under R version 4.4.3

## Warning: package 'lubridate' was built under R version 4.4.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(dplyr)
library(lubridate)
library(ggplot2)
library(prophet)
```

```
## Warning: package 'prophet' was built under R version 4.4.3
```

```
## Loading required package: Rcpp
## Loading required package: rlang
##
## Attaching package: 'rlang'
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

```r
# Import the data
url_in = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi
file_names = c("time_series_covid19_confirmed_US.csv",
               "time_series_covid19_deaths_US.csv",
               "time_series_covid19_confirmed_global.csv",
               "time_series_covid19_deaths_global.csv")
urls = str_c(url_in, file_names)
global_conf = read.csv(urls[3])
US_conf = read.csv(urls[1])
global_deaths = read.csv(urls[4])
US_deaths = read.csv(urls[2])
uid_url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UID_ISO_
Pop = read.csv(uid_url)
```

Data Preparation

Reshaping, Conversion, and Renaming of the Global Dataset

```r
# Remove Lat and Long, change the dates to single column with values as cases
global_conf = global_conf %>%
 select(-Lat, -Long) %>%
 pivot_longer(
 cols = starts_with("X"),
 names_to = "Date",
 values_to = "Cases"
 ) %>%
 mutate(
 Date = sub("^X", "", Date),
 Date = as.Date(Date, format = "%m.%d.%y")
 )
```

```r
# Remove Lat and Long, change the dates to single column with values as cases
global_deaths = global_deaths %>%
  select(-Lat, -Long) %>%
  pivot_longer(
    cols = starts_with("X"),
    names_to = "Date",
    values_to = "Cases"
  ) %>%
  mutate(
    Date = sub("^X", "", Date),
    Date = as.Date(Date, format = "%m.%d.%y")
  )
```

```r
# Join the global deaths and confirmed cases
global_combined <- full_join(
  global_conf,
  global_deaths,
  by = c("Province.State", "Country.Region", "Date"),
  suffix = c("_confirmed", "_deaths")
)
```

```r
# Remove any cases that are 0
global_combined = global_combined %>% filter(Cases_confirmed > 0)
```

Reshaping, Conversion, and Renaming of the US Dataset

```r
# Change columns to combined, remove X in front of date, change dates to one column with cases for the
US_conf = US_conf %>%
  pivot_longer(
    cols = -(UID:Combined_Key),
    names_to = "date",
    values_to = "case"
  ) %>%
  mutate(
    date = sub("^X", "", date),
    date = mdy(date)
  ) %>%
  select(Admin2, Province_State, Country_Region, Combined_Key, date, case)
```

```r
# Change columns to combined, remove X in front of date, change dates to one column with cases for the
US_deaths = US_deaths %>%
  pivot_longer(
    cols = -c(UID:Combined_Key, Population),
    names_to = "date",
    values_to = "case"
  ) %>%
  mutate(
    date = sub("^X", "", date),
    date = as.Date(date, format = "%m.%d.%y")
  ) %>%
  select(Admin2, Province_State, Country_Region, Combined_Key, Population, date, case)
```

```r
# Combined US deaths and cases
US_combined = full_join(
  US_conf,
  US_deaths,
  by = c("Admin2", "Province_State", "Country_Region", "Combined_Key", "date"),
  suffix = c("_confirmed", "_deaths")
) %>%
  rename(
    Confirmed = case_confirmed,
    Deaths = case_deaths
  ) %>%
  select(Admin2, Province_State, Country_Region, Combined_Key, date, Population, Confirmed, Deaths)
```

```r
# Create the Combined_Key column and rename the columns
global_combined = global_combined %>%
  unite(Combined_Key, Province.State, Country.Region, sep = "_", remove = FALSE) %>%
  rename(Province_State = Province.State, Country_Region = Country.Region)
```

```r
# Renaming columns to match for the population join
global_combined = global_combined %>%
  rename(Cases = Cases_confirmed, Deaths = Cases_deaths)

US_combined = US_combined %>%
  rename(Cases = Confirmed, Date= date)
```

```r
# Add population to global variable
global_combined = global_combined %>%
  left_join(Pop %>% select(Province_State, Country_Region, Population),
            by = c("Province_State", "Country_Region")) %>%
  select(Province_State, Country_Region, Date, Cases, Deaths, Population, Combined_Key)
```

Data Analysis, Visualization, and Modeling

```r
# US by state
US_by_state = US_combined %>%
  group_by(Province_State, Date) %>%
  summarize(
    Cases = sum(Cases, na.rm = TRUE),
    Deaths = sum(Deaths, na.rm = TRUE),
    Population = sum(Population, na.rm = TRUE),  #
    .groups = "drop"
  )
US_by_state = US_by_state %>%
  mutate(deaths_per_mill = Deaths * 1e6 / Population)

summary(US_by_state)
```

```
##  Province_State          Date                 Cases              Deaths
##  Length:66294       Min.   :2020-01-22   Min.   :       0   Min.   :     0
##  Class :character   1st Qu.:2020-11-02   1st Qu.:   31115   1st Qu.:   555
```

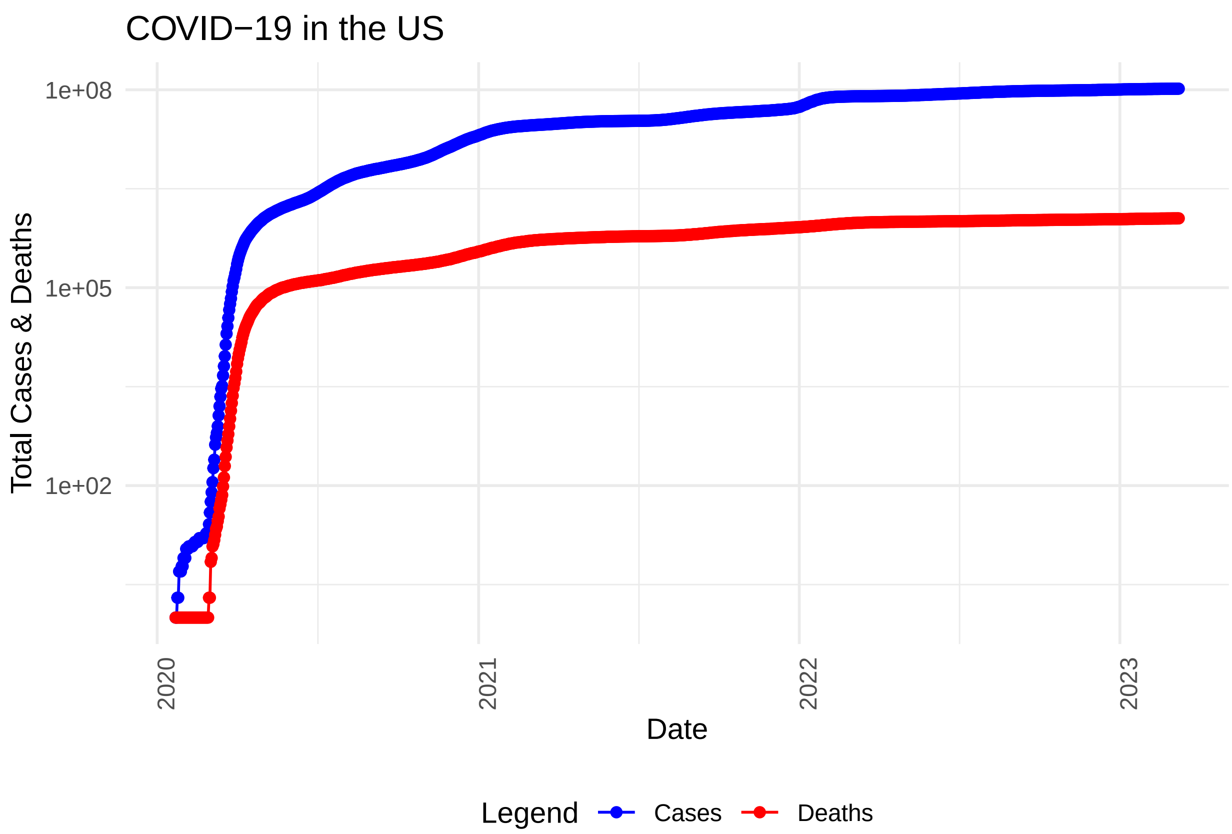
```
##  Mode  :character   Median :2021-08-15   Median :  293146   Median :  3849
##                     Mean   :2021-08-15   Mean   :  811738   Mean   : 10768
##                     3rd Qu.:2022-05-28   3rd Qu.:  953450   3rd Qu.: 13695
##                     Max.   :2023-03-09   Max.   :12129699   Max.   :101159
##
##    Population       deaths_per_mill
##  Min.   :       0   Min.   :   0.0
##  1st Qu.: 1068778   1st Qu.: 490.2
##  Median : 3660113   Median :1665.9
##  Mean   : 5739226   Mean   :   Inf
##  3rd Qu.: 6892503   3rd Qu.:2794.0
##  Max.   :39512223   Max.   :   Inf
##                     NA's   :1211
```

```r
# Creating US totals
US_totals = US_by_state %>%
  group_by(Date) %>%
  summarize(
    Cases = sum(Cases, na.rm = TRUE),
    Deaths = sum(Deaths, na.rm = TRUE),
    Population = sum(Population, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(deaths_per_mill = Deaths * 1e6 / Population) %>%
  select(Date, Cases, Deaths, Population, deaths_per_mill)
summary(US_totals)
```

```
##       Date                Cases                Deaths           Population
##  Min.   :2020-01-22   Min.   :        1   Min.   :      1   Min.   :332875137
##  1st Qu.:2020-11-02   1st Qu.:  9401880   1st Qu.: 232564   1st Qu.:332875137
##  Median :2021-08-15   Median : 36845902   Median : 618029   Median :332875137
##  Mean   :2021-08-15   Mean   : 47080794   Mean   : 624563   Mean   :332875137
##  3rd Qu.:2022-05-27   3rd Qu.: 84083678   3rd Qu.:1006626   3rd Qu.:332875137
##  Max.   :2023-03-09   Max.   :103802702   Max.   :1123836   Max.   :332875137
##  deaths_per_mill
##  Min.   :   0.003
##  1st Qu.: 698.652
##  Median :1856.639
##  Mean   :1876.267
##  3rd Qu.:3024.033
##  Max.   :3376.149
```

```r
# Plot of US total cases and deaths
US_totals %>%
  filter(Cases > 0) %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = Cases, color = "Cases")) +
  geom_point(aes(y = Cases, color = "Cases")) +
  geom_line(aes(y = Deaths, color = "Deaths")) +
  geom_point(aes(y = Deaths, color = "Deaths")) +
  scale_y_log10() +
  scale_color_manual(values = c("Cases" = "blue", "Deaths" = "red")) +
  theme_minimal() +
```

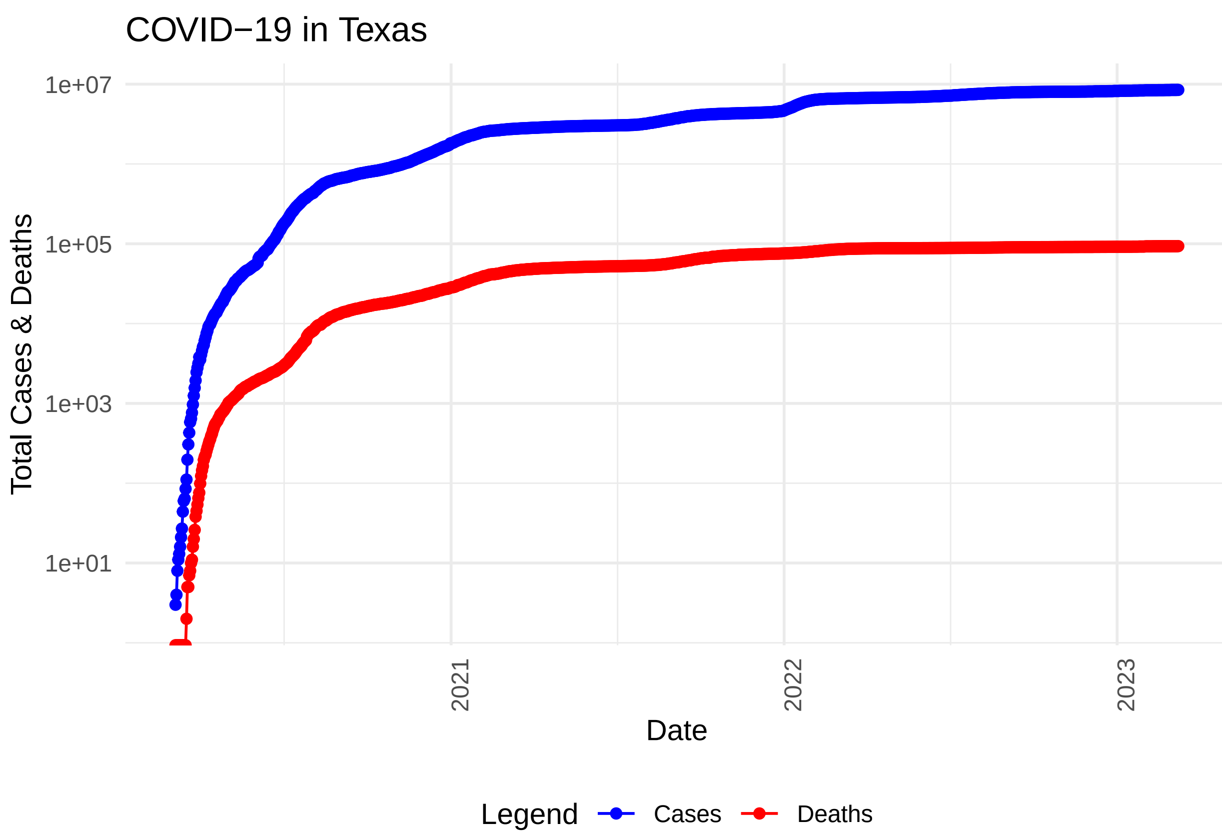
```
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 90, hjust = 1)
  ) +
  labs(
    title = "COVID-19 in the US",
    y = "Total Cases & Deaths",
    x = "Date",
    color = "Legend"
  )
```




```
# Texas cases and deaths
US_by_state %>%
  filter(Province_State == "Texas") %>%
  filter(Cases > 0) %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = Cases, color = "Cases")) +
  geom_point(aes(y = Cases, color = "Cases")) +
  geom_line(aes(y = Deaths, color = "Deaths")) +
  geom_point(aes(y = Deaths, color = "Deaths")) +
  scale_y_log10() +
  scale_color_manual(values = c("Cases" = "blue", "Deaths" = "red")) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
```

```r
    axis.text.x = element_text(angle = 90, hjust = 1)
  ) +
  labs(
    title = "COVID-19 in Texas",
    y = "Total Cases & Deaths",
    x = "Date",
    color = "Legend"
  )
```

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




```r
# max by date
cat("Max Date:", format(max(US_totals$Date), "%Y-%m-%d"), "\n")
```

```
## Max Date: 2023-03-09
```

```r
# max deaths
cat("Max Deaths:", (max(US_totals$Deaths)), "\n")
```

```
## Max Deaths: 1123836
```

```r
# evaluate if the deaths are plateauing

US_by_state <- US_by_state %>%
  arrange(Province_State, Date) %>%
  mutate(
    new_cases = Cases - lag(Cases, default = 0),
    new_deaths = Deaths - lag(Deaths, default = 0)
  )

US_totals <- US_totals %>%
  mutate(
    new_cases = Cases - lag(Cases, default = 0),
    new_deaths = Deaths - lag(Deaths, default = 0)
  )

tail(US_totals %>% select(new_cases, new_deaths, everything()))
```

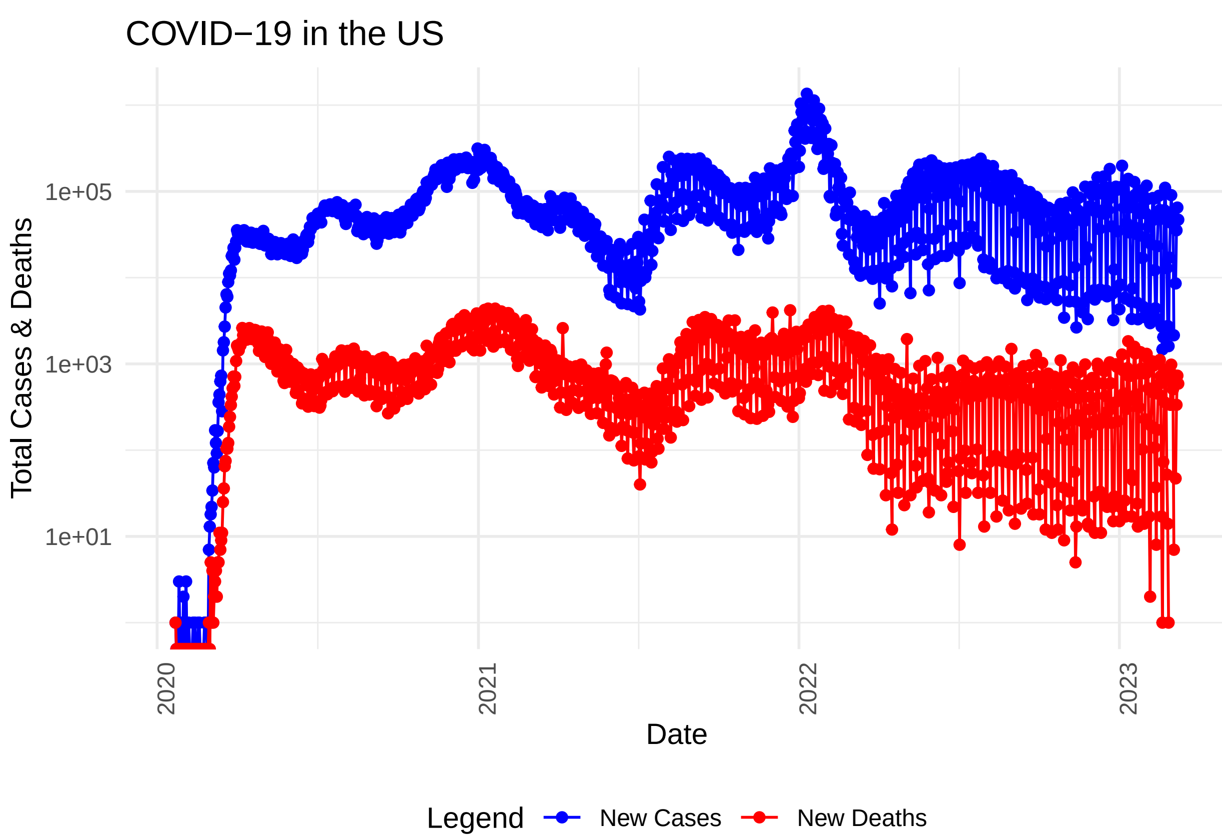
```
## # A tibble: 6 x 7
##   new_cases new_deaths Date           Cases  Deaths Population deaths_per_mill
##       <int>      <int> <date>         <int>   <int>      <int>           <dbl>
## 1      2147          7 2023-03-04 103650837 1122172  332875137           3371.
## 2     -3862        -38 2023-03-05 103646975 1122134  332875137           3371.
## 3      8564         47 2023-03-06 103655539 1122181  332875137           3371.
## 4     35371        335 2023-03-07 103690910 1122516  332875137           3372.
## 5     64861        730 2023-03-08 103755771 1123246  332875137           3374.
## 6     46931        590 2023-03-09 103802702 1123836  332875137           3376.
```

```r
# Plot the new cases and new deaths

US_totals %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = new_cases, color = "New Cases")) +
  geom_point(aes(y = new_cases, color = "New Cases")) +
  geom_line(aes(y = new_deaths, color = "New Deaths")) +
  geom_point(aes(y = new_deaths, color = "New Deaths")) +
  scale_y_log10() +
  scale_color_manual(values = c("New Cases" = "blue", "New Deaths" = "red")) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 90, hjust = 1)
  ) +
  labs(
    title = "COVID-19 in the US",
    y = "Total Cases & Deaths",
    x = "Date",
    color = "Legend"
  )
```

```
## 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_point()`).

## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Texas graph of new deaths and cases

state = "Texas"

US_by_state %>%
```

```r
  filter(Province_State == state) %>%
  filter(new_cases > 0) %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = new_cases, color = "New Cases")) +
  geom_point(aes(y = new_cases, color = "New Cases")) +
  geom_line(aes(y = new_deaths, color = "New Deaths")) +
  geom_point(aes(y = new_deaths, color = "New Deaths")) +
  scale_y_log10() +
  scale_color_manual(values = c("New Cases" = "blue", "New Deaths" = "red")) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 90, hjust = 1)
  ) +
  labs(
    title = "COVID-19 in Texas",
    y = "New Cases & Deaths",
    x = "Date",
    color = "Legend"
  )
```

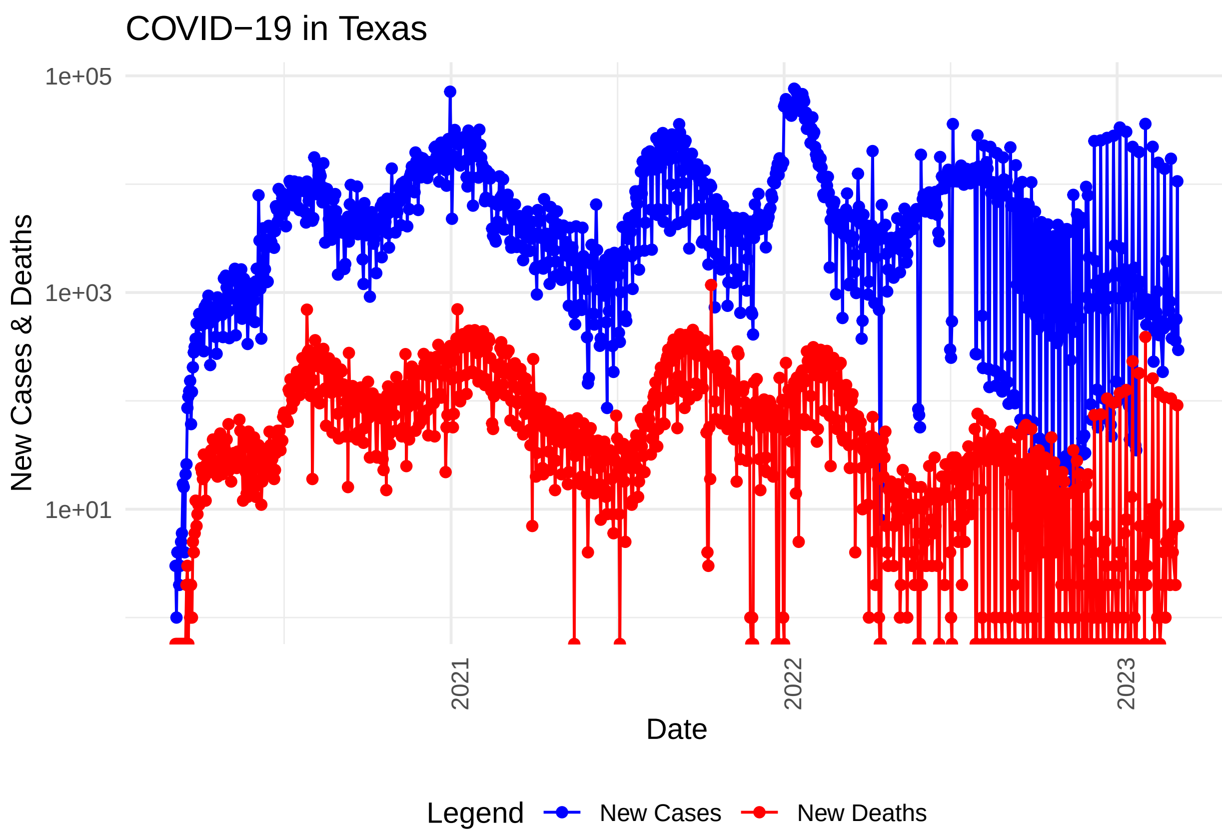
```
## Warning in transformation$transform(x): NaNs produced
```

```
## 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 3 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```




```r
# Summarize state level totals
US_state_totals = US_by_state %>%
  group_by(Province_State) %>%
  summarize(
    deaths = max(Deaths, na.rm = TRUE),
    cases = max(Cases, na.rm = TRUE),
    population = max(Population, na.rm = TRUE),
    cases_per_thou = 1000 * cases / population,
    deaths_per_thou = 1000 * deaths / population,
    .groups = "drop"
  ) %>%
  filter(cases > 0, population > 0)

# Get the 10 states with the lowest deaths per thousand
US_state_totals %>%
  slice_min(deaths_per_thou, n = 10) %>%
  select(Province_State, deaths_per_thou, cases_per_thou, cases, deaths, population)
```

```
## # A tibble: 10 x 6
##    Province_State        deaths_per_thou cases_per_thou  cases deaths population
##    <chr>                           <dbl>          <dbl>  <int>  <int>      <int>
##  1 American Samoa                  0.611           150. 8.32e3     34      55641
##  2 Northern Mariana Isl~           0.744           248. 1.37e4     41      55144
##  3 Virgin Islands                  1.21            231. 2.48e4    130     107268
##  4 Hawaii                          1.30            269. 3.81e5   1841    1415872
##  5 Vermont                         1.49            245. 1.53e5    929     623989
```

```
##  6 Puerto Rico                          1.55            293. 1.10e6   5823    3754939
##  7 Utah                                 1.65            340. 1.09e6   5298    3205958
##  8 Alaska                               2.01            415. 3.08e5   1486     740995
##  9 District of Columbia                 2.03            252. 1.78e5   1432     705749
## 10 Washington                           2.06            253. 1.93e6  15683    7614893
```

```
# Top 10 worse states
US_state_totals %>%
  slice_max(deaths_per_thou, n = 10) %>%
  select(Province_State, deaths_per_thou, cases_per_thou, cases, deaths, population)
```

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

```
# Linear model predicting deaths per 1000 based on cases per 1000 for the US dataset
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
```

```
# Deaths per 1000 by population for the US dataset
mod2 = lm(deaths_per_thou ~ cases_per_thou + population, data = US_state_totals)
summary(mod2)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou + population, data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18875 -0.57670  0.08483  0.63530  1.25999
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.428e-01  7.184e-01  -0.616    0.540
## cases_per_thou  1.113e-02  2.297e-03   4.843 1.15e-05 ***
## population      2.404e-08  1.594e-08   1.508    0.137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8515 on 53 degrees of freedom
## Multiple R-squared:  0.3347, Adjusted R-squared:  0.3095
## F-statistic: 13.33 on 2 and 53 DF,  p-value: 2.045e-05
```

```r
# Analysis of variance deaths vs cases per thousand and population for the US dataset
anova(mod, mod2)
```

A simple linear regression model predicting deaths per thousand based on cases per thousand in the U.S. dataset showed a statistically significant relationship ($p < 0.001$), with an R-squared value of 0.31, indicating that 30.6% of the variance in deaths per thousand can be explained by cases per thousand. The regression coefficient suggests that for every additional case per thousand, the death rate increases by 0.0113 per thousand. Adding population as an additional predictor in the second model slightly improved the R-squared value to 0.33, but the population variable was not statistically significant ($p = 0.137$).

```
## Analysis of Variance Table
##
## Model 1: deaths_per_thou ~ cases_per_thou
## Model 2: deaths_per_thou ~ cases_per_thou + population
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     54 40.080
## 2     53 38.431  1     1.649 2.2741 0.1375
```

```r
# Add deaths and cases per thousand for the global dataset
global_combined = global_combined %>%
  mutate(
    deaths_per_thou = (Deaths / Population) * 1000,
    cases_per_thou = (Cases / Population) * 1000
  )
```

```r
# Linear model for deaths and cases per 1000 for the global dataset
mod_global = lm(deaths_per_thou ~ cases_per_thou, data = global_combined)
summary(mod_global)
```

An ANOVA comparison between the two models revealed no significant improvement (p = 0.1375), suggesting that population does not substantially contribute to explaining deaths per thousand.

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = global_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7452 -0.2890 -0.2577  0.0712  5.7881
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.890e-01  1.623e-03   178.0   <2e-16 ***
## cases_per_thou 4.271e-03  1.107e-05   385.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7754 on 300096 degrees of freedom
##   (6729 observations deleted due to missingness)
## Multiple R-squared:  0.3316, Adjusted R-squared:  0.3316
## F-statistic: 1.489e+05 on 1 and 300096 DF,  p-value: < 2.2e-16
```

```r
mod2_global = lm(deaths_per_thou ~ cases_per_thou + Population, data = global_combined)
summary(mod2_global)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou + Population, data = global_combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7475 -0.2867 -0.2563  0.0740  5.7864
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.819e-01  1.700e-03  165.82   <2e-16 ***
## cases_per_thou 4.284e-03  1.111e-05  385.77   <2e-16 ***
## Population     2.132e-10  1.516e-11   14.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7751 on 300095 degrees of freedom
##   (6729 observations deleted due to missingness)
## Multiple R-squared:  0.332,  Adjusted R-squared:  0.332
## F-statistic: 7.459e+04 on 2 and 300095 DF,  p-value: < 2.2e-16
```

```r
# Analysis of variance deaths vs cases per thousand and population for the global dataset
anova(mod_global, mod2_global)
```

In the global dataset, the relationship between deaths per thousand and cases per thousand is much stronger, with an R-squared value of 0.33. The p-value for cases per thousand is extremely low ($p < 2.2e-16$), confirming a strong and highly significant association. The estimated coefficient indicates that each additional case per thousand results in a 0.0043 increase in deaths per thousand. When population was added as a predictor, the model slightly improved, with R-squared increasing to 0.332. Unlike the U.S. dataset, population was statistically significant ($p < 2.2e-16$), indicating that it does contribute to explaining deaths per thousand globally.

```
## Analysis of Variance Table
##
## Model 1: deaths_per_thou ~ cases_per_thou
## Model 2: deaths_per_thou ~ cases_per_thou + Population
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1 300096 180414
## 2 300095 180295  1    118.84 197.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# State with the minimum cases per 1000
US_state_totals %>%
  slice_min(cases_per_thou, n = 1)
```

An ANOVA test confirmed that adding population resulted in a statistically significant improvement ($p < 2.2e-16$).

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

```r
# State with the maximum cases per 1000
US_state_totals %>%
  slice_max(cases_per_thou, n = 1)
```

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

```r
# Create a sequence from 1 to 151
x_grid = seq(1, 151)

# Create a new tibble for cases_per_thou
```

```r
new_df = tibble(cases_per_thou = x_grid)

# Add predicted values from the regression model
US_state_totals = US_state_totals %>%
  mutate(pred = predict(mod, newdata = US_state_totals))

head(US_state_totals)
```

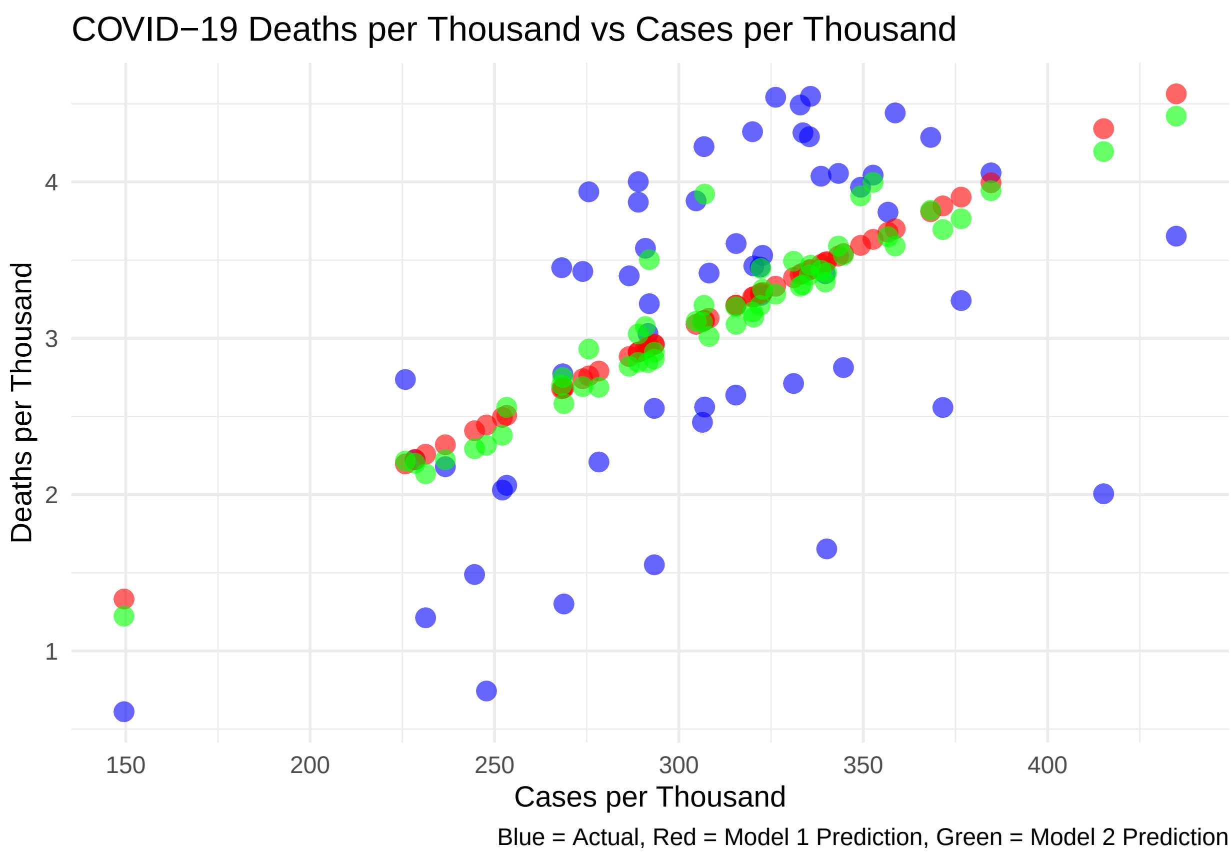
```
## # A tibble: 6 x 7
##   Province_State deaths    cases population cases_per_thou deaths_per_thou  pred
##   <chr>           <int>    <int>      <int>          <dbl>           <dbl> <dbl>
## 1 Alabama         21032  1644533    4903185           335.           4.29   3.44
## 2 Alaska           1486   307655     740995           415.           2.01   4.34
## 3 American Samoa     34     8320      55641           150.           0.611  1.33
## 4 Arizona         33102  2443514    7278717           336.           4.55   3.44
## 5 Arkansas        13020  1006883    3017804           334.           4.31   3.42
## 6 California     101159 12129699   39512223           307.           2.56   3.12
```

```r
library(dplyr)

US_total_w_pred = US_state_totals %>%
  mutate(
    pred = predict(mod, newdata = US_state_totals),
    pred2 = predict(mod2, newdata = US_state_totals),
    std_ratio = ((deaths_per_thou / cases_per_thou) -
                (mean(deaths_per_thou, na.rm = TRUE) / mean(cases_per_thou, na.rm = TRUE))) /
                sd(deaths_per_thou / cases_per_thou, na.rm = TRUE)
  )

US_total_w_pred
```

```
## # A tibble: 56 x 9
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <int>  <int>      <int>          <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
## # i 2 more variables: pred2 <dbl>, std_ratio <dbl>
```

```r
# US total with the 2 prediction models
US_total_w_pred %>%
  ggplot(aes(x = cases_per_thou)) +
  geom_point(aes(y = deaths_per_thou), color = "blue", alpha = 0.6, size = 3) +
  geom_point(aes(y = pred), color = "red", alpha = 0.6, size = 3) +
```

```r
  geom_point(aes(y = pred2), color = "green", alpha = 0.6, size = 3) +
  labs(
    title = "COVID-19 Deaths per Thousand vs Cases per Thousand",
    x = "Cases per Thousand",
    y = "Deaths per Thousand",
    caption = "Blue = Actual, Red = Model 1 Prediction, Green = Model 2 Prediction"
  ) +
  theme_minimal()
```




```r
# Clear memory, sometimes needed for older operating systems
gc()
```

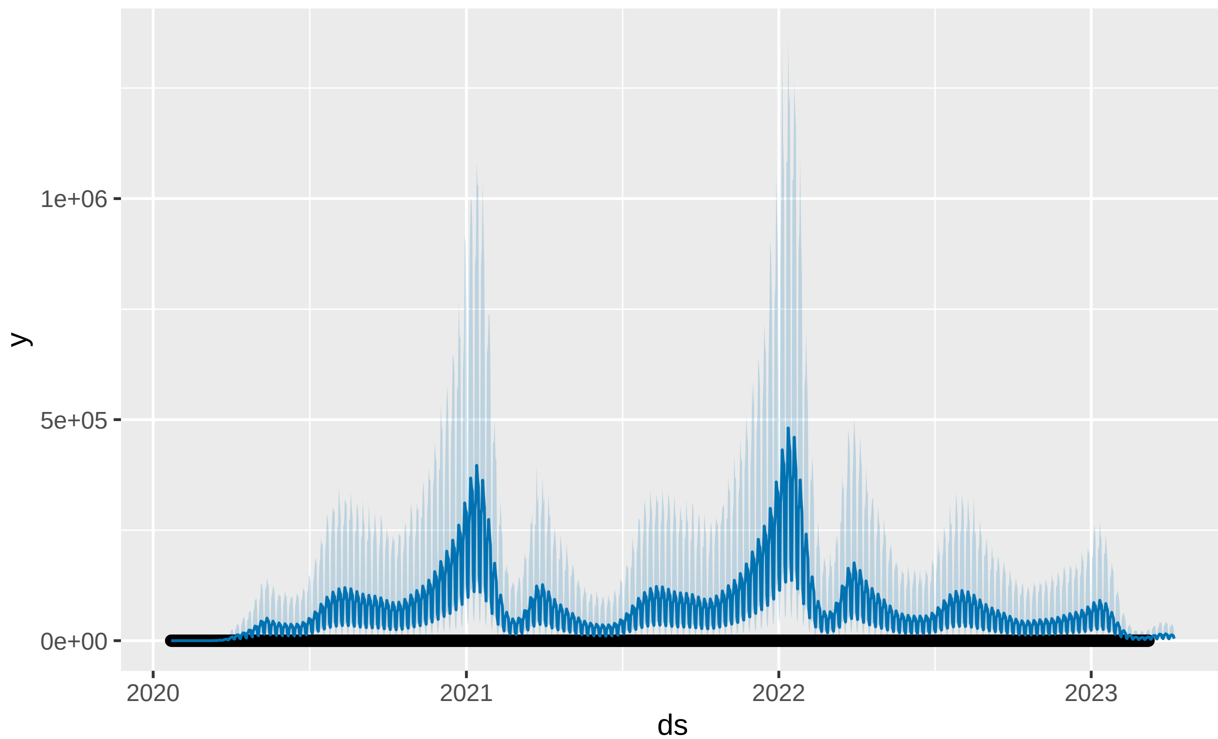
```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  1664602  88.9    3078367  164.5   2467941  131.9
## Vcells 82922772 632.7  144122046 1099.6 143230413 1092.8
```

```r
# Timeseries Forcasting


# Load and prepare data
US_combined = US_combined %>%
    mutate(Date = as.Date(Date))

# Aggregate cases for all states (sum across states for each date)
```

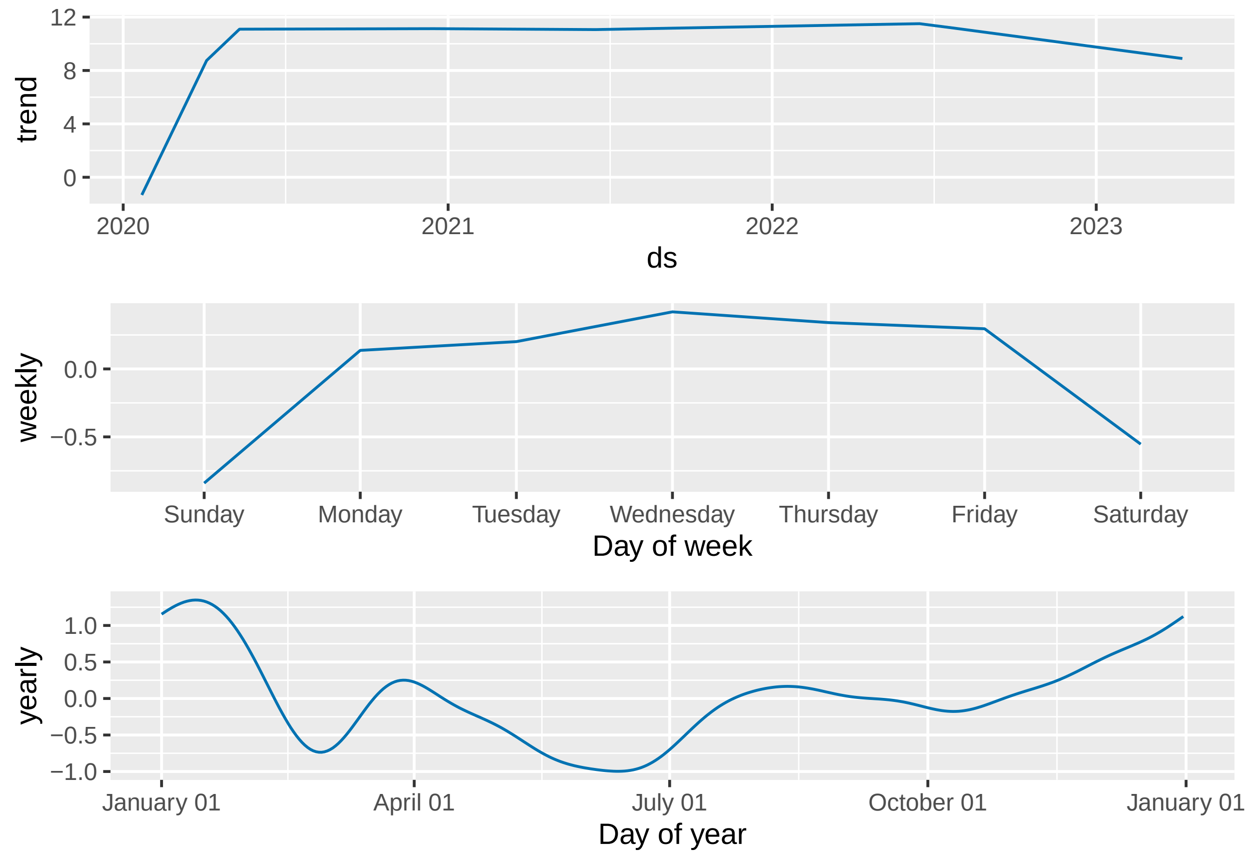
```r
us_aggregated = US_combined %>%
    group_by(Date) %>%
    summarise(y = sum(Cases, na.rm = TRUE)) %>%
    ungroup()

# Rename columns for Prophet
colnames(us_aggregated) = c("ds", "y")

# Convert cumulative cases to daily new cases
us_aggregated = us_aggregated %>%
    arrange(ds) %>%
    mutate(y = y - lag(y, default = first(y))) %>%
    mutate(y = ifelse(y < 0, 0, y))

# Apply log transformation
us_aggregated = us_aggregated %>%
    mutate(y = log1p(y))

# Train Prophet model
model = prophet(us_aggregated,
                weekly.seasonality = TRUE,
                yearly.seasonality = TRUE)
```

```
## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.
```

```r
# Generate future predictions
future_dates = make_future_dataframe(model, periods = 30)
forecast = predict(model, future_dates)

# Convert back from log scale
forecast = forecast %>%
    mutate(yhat = expm1(yhat),
           yhat_lower = expm1(yhat_lower),
           yhat_upper = expm1(yhat_upper))

# Plot the forecast
plot(model, forecast)
```




```r
# Plot seasonality components (trend, weekly, yearly)
prophet_plot_components(model, forecast)
```



First Image (Seasonality Components). Second Image (Actual vs Predicted Cases): Black Dots: Actual COVID-19 daily cases.Blue Line: Forecasted values with confidence intervals. ds = date stamp. y = responsive variable (daily covid cases).

Conclusions, Biases

COVID-19 was considered a serious global threat, and understanding its spread is crucial for epotential future infectious issues. This analysis examines how population size and case rates per thousand influence deaths per thousand across U.S. states. Using linear regression models, I generated state-level death rate predictions. However, the models could be significantly improved with additional predictors such as lockdown, masking, testing rates, vaccinations, and temperature.

In the U.S. dataset, the latest recorded date is March 9, 2023, with a maximum total deaths of 1,123,836. Among the 10 worst-affected states, Arizona has the highest death rate per 1,000 population (4.55 deaths per 1,000), followed closely by Oklahoma (4.54) and Mississippi (4.49). The list also includes West Virginia, New Mexico, Arkansas, Alabama, Tennessee, Michigan, and Kentucky, all exhibiting death rates above 4 deaths per 1,000 people. This data highlights regional disparities in COVID-19 mortality, potentially influenced by healthcare infrastructure, vaccination rates, underlying health conditions, and public health policies. States with higher death rates may have faced greater challenges in pandemic response and higher vulnerability among their populations.

The COVID-19 trend in the U.S. shows a sharp initial rise in both cases and deaths in early 2020, reflecting the rapid spread of the virus and its severe impact. While the cumulative number of cases is significantly higher than deaths, both follow a similar pattern over time. The growth rate slowed after 2021, with cases and deaths stabilizing by late 2022. The use of a logarithmic scale emphasizes the early surges but also highlights how the rate of increase became more gradual after the initial waves. Similarly, the COVID-19 trend in Texas follows a trajectory comparable to the national trend, with an early surge in cases and deaths. However, Texas' total case count is much lower than the national total, which aligns with its smaller population. The proportion of deaths to total cases appears consistent with the national average, suggesting a similar mortality rate. By late 2022, both cases and deaths plateaued, mirroring the overall U.S. trend.

The scatter plot shows a positive correlation between COVID-19 deaths per thousand and cases per thousand, indicating that as cases increase, deaths also tend to rise. The data points generally follow a clear trend, though some deviations and outliers are present, particularly at lower death rates where actual values diverge more significantly. This suggests that the relationship between cases and deaths is strong.

Timeseries forecasting for the US states: This demonstrates a rise in cases in early 2020, peaks in 2021–2022, and then stabilizes. Cases increase early in the week (Monday–Wednesday) and drop on weekends (Saturday). Peaks appear around April and December, which might align with holiday surges or seasonal COVID-19 waves. Actual vs Predicted Cases-> This model captures major COVID-19 waves (2021 & 2022 spikes).

A key source of bias in the data stems from inconsistencies in state-level reporting of cases and deaths, influenced by infrastructure limitations and reporting delays. Testing rates also introduce bias, as there are likely underreported. There is no socioeconimic status such as access or quality of healthcare, testing rates, and reporting discrepancies (such as reporting a COVID death when it was assocaited but not causal). The choice of an alpha level of 0.05 could also impact statistical conclusions.