COVID-19 Final Report

2023-12-09

How Well Did States Treat COVID?

Introduction

This is for the Data Science as a Field Final Project. All code is left as echoed in order to show work. Note that in cases where the word 'state' is used, US territories are included. Any analysis of US states includes US territories in this report. Thank you.

Question of Interest

Which US states/territories treated COVID cases effectively? Essentially, regardless of case counts, which US states did the worst and best jobs of preventing cases from becoming fatal.

Package Requirements

For your ease, this document will attempt to automatically install required packages, but it may not do so successfully. If it fails, please see the package checks section and code to install needed packages.

Package Checks

```
# Install required packages
options(repos = c(CRAN = "https://cran.rstudio.com/"))

if (!requireNamespace("lubridate", quietly = TRUE))
  install.packages("lubridate")

if (!requireNamespace("dplyr", quietly = TRUE))
  install.packages("dplyr")

if (!requireNamespace("tidyverse", quietly = TRUE))
  install.packages("tidyverse")

if (!requireNamespace("ggplot2", quietly = TRUE))
  install.packages("ggplot2")

if (!requireNamespace("sf", quietly = TRUE))
  install.packages("sf")

if (!requireNamespace("maps", quietly = TRUE))
```

```
  install.packages("maps")

if (!requireNamespace("mapdata", quietly = TRUE))
  install.packages("mapdata")

if (!requireNamespace("forecast", quietly = TRUE))
  install.packages("forecast")
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats 1.0.0     v stringr 1.5.1
## v ggplot2 3.4.4     v tibble  3.2.1
## v purrr   1.0.2     v tidyr   1.3.0
## v readr   2.1.4
```

```
## -- 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(stringr)
library(readr)
library(ggplot2)
library(sf)
```

```
## Linking to GEOS 3.11.2, GDAL 3.7.2, PROJ 9.3.0; sf_use_s2() is TRUE
```

```
library(maps)
```

```
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(mapdata)
library(forecast)
```

Data Description, Importing, and Data Cleaning

The data in this project comes from the John Hopkins COVID-19 cases and deaths data sets. The link is in the source section. It is the one from the course. In this case, we will only be looking at US cases for state comparisons. This section will show how the data was imported and cleaned. This is akin to the examples in the lecture, as instructed to do.

```
# Download data URLs
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cov
file_names <- c(
  "time_series_covid19_confirmed_US.csv",
  "time_series_covid19_deaths_US.csv"
)

urls <- str_c(url_in, file_names)

# Download and read data
us_cases <- read_csv(urls[1])
```

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## 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.
```

```
us_deaths <- read_csv(urls[2])
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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.
```

```
# Get data cleaning, similar to how the lecture suggests.
us_cases <- us_cases %>%
  pivot_longer(cols = -(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 = -(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)`
```

Calculating Fatal Case Rate, Deaths Per Thousand, and Cases Per Thousand Metrics

This report is interested in US analysis only It is unfair to do a raw comparison of population, so this report will use cases and deaths per thousand metrics. Important to the report, a fatal cases (death rate) column based on deaths per thousand and cases per thousand will also be created for new comparison not seen in the lectures.

```
US_state_table <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(cases_thousand = cases * 1000 / Population) %>%
  mutate(deaths_thousand = deaths * 1000 / Population) %>%
  mutate(death_rate = deaths_thousand / cases_thousand) %>%
  select(Province_State, Country_Region, date, cases, cases_thousand, deaths, deaths_thousand, death_ra
  ungroup()
```

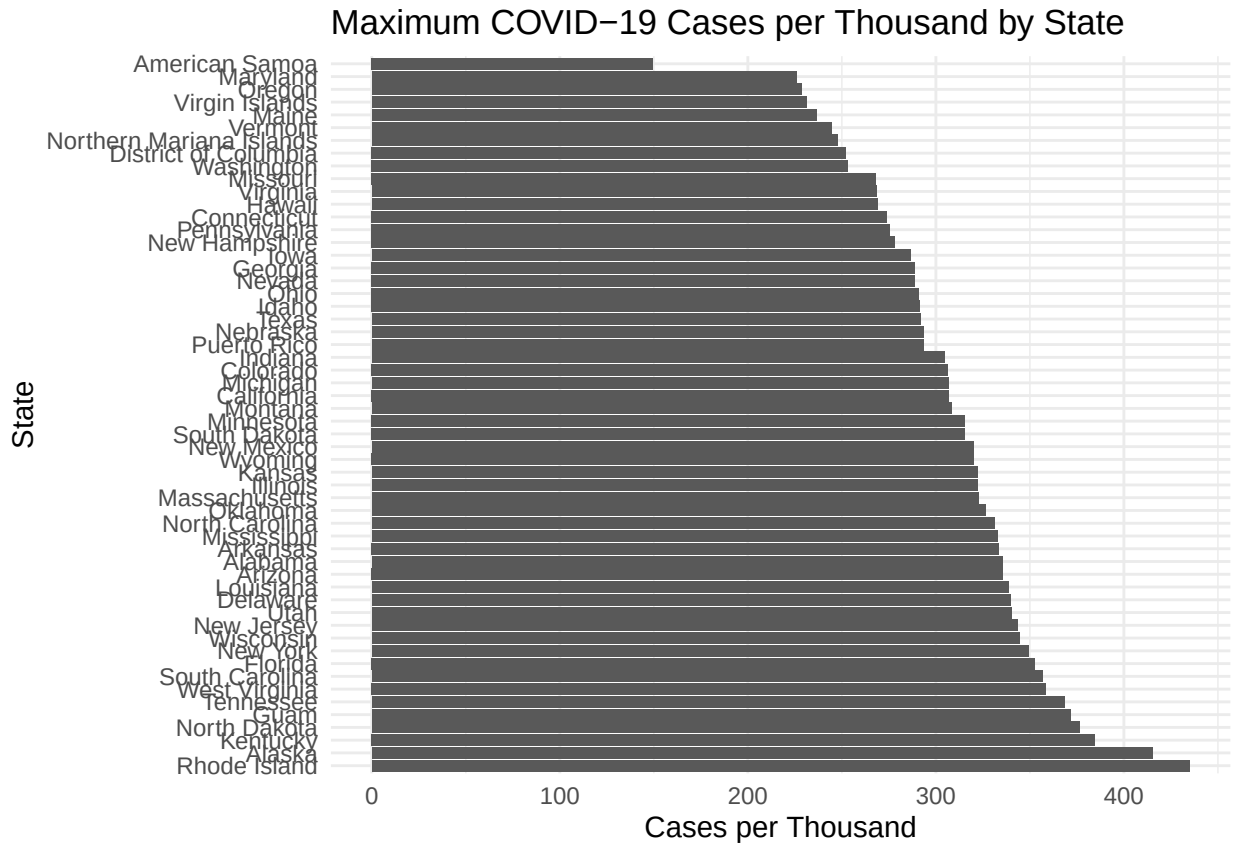
```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can
## override using the `.groups` argument.
```

```
# At this point, lets also pull out the Diamond Princess and Grand Princess, as we are not interested i
US_state_table <- US_state_table %>%
  filter(!(Province_State %in% c("Grand Princess", "Diamond Princess")))
```

State Comparison: Which One Had and Handled COVID-19 Worst?

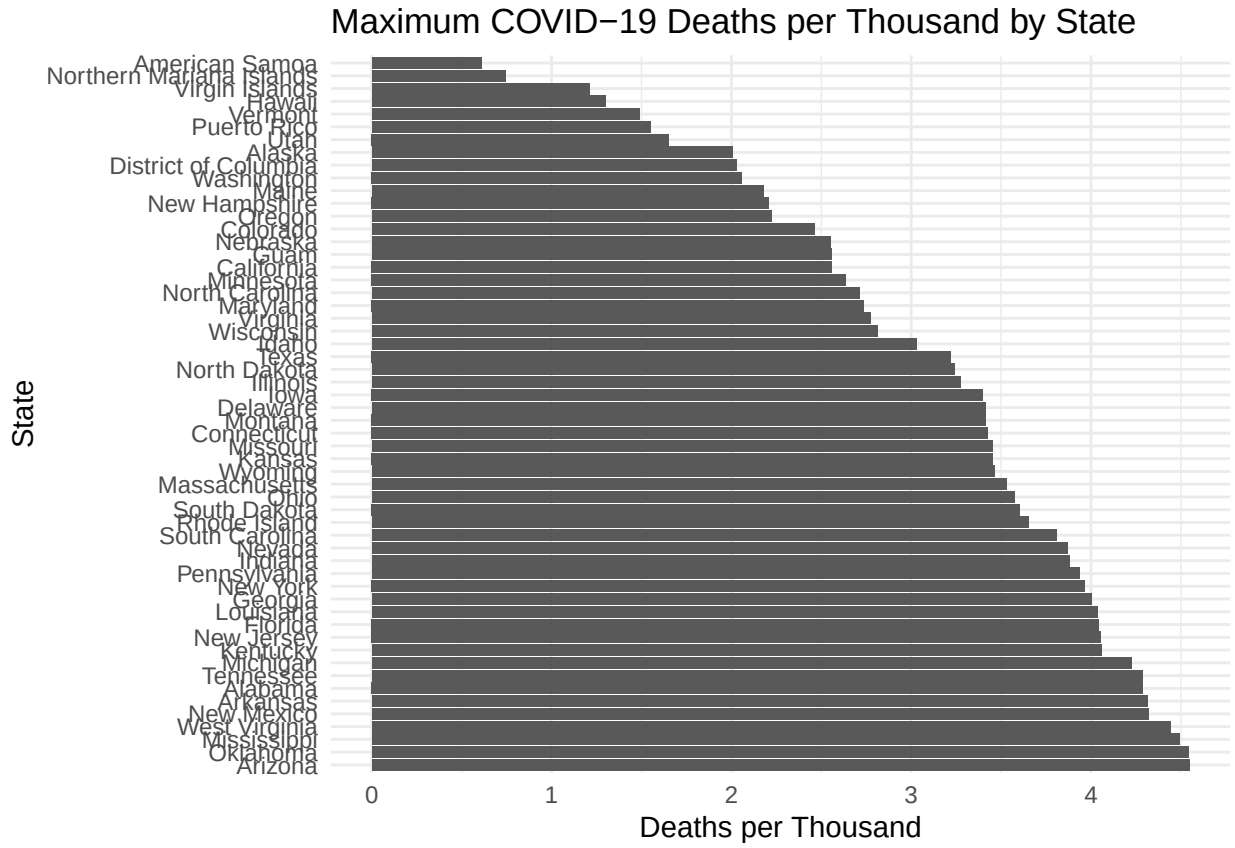
It is evident in the third chart that the death rate in is the worst in Ohio on average. The best is American Samoa (not a state, but a territory). For those interested, the actual US state best at treating COVID cases is Utah. Essentially, the third chart is demonstrating the analysis of how many cases are fatal, relative to how many cases there are per state. The first two charts display max cases and max deaths for reference and scale.

```
# Group by Province_State and calculate maximum cases per thousand
max_cases_per_thousand <- US_state_table %>%
  group_by(Province_State) %>%
  summarize(MaxCasesPerThousand = max(cases_thousand, na.rm = TRUE))

# Now create the Bar Chart
ggplot(max_cases_per_thousand, aes(x = reorder(Province_State, -MaxCasesPerThousand), y = MaxCasesPerTh
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Maximum COVID-19 Cases per Thousand by State",
       x = "State",
       y = "Cases per Thousand") +
  coord_flip() # Flips the axes for better readability
```

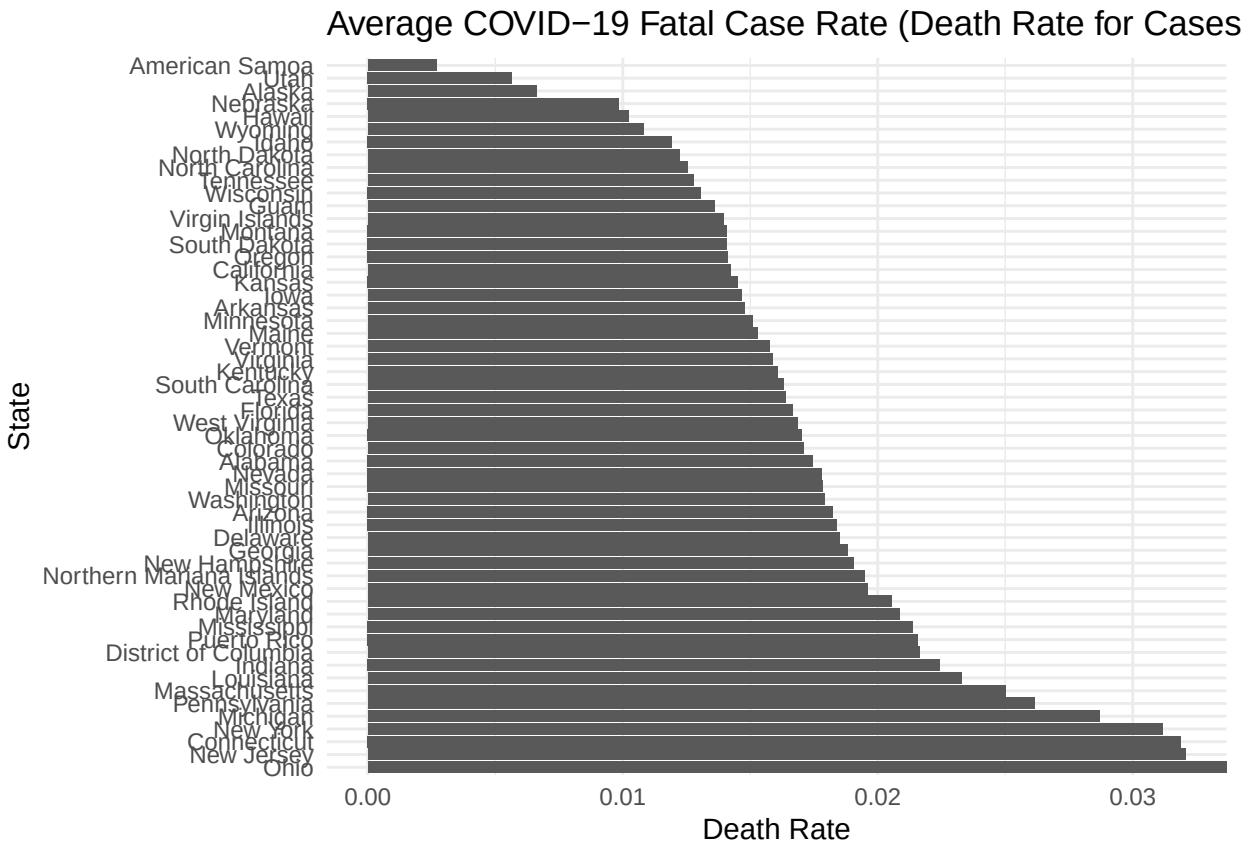



```
# Now again for deaths
max_deaths_per_thousand <- US_state_table %>%
  group_by(Province_State) %>%
```

```
  summarize(MaxDeathsPerThousand = max(deaths_thousand, na.rm = TRUE))

ggplot(max_deaths_per_thousand, aes(x = reorder(Province_State, -MaxDeathsPerThousand), y = MaxDeathsPer
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Maximum COVID-19 Deaths per Thousand by State",
       x = "State",
       y = "Deaths per Thousand") +
  coord_flip()
```




```
# And finally for death rate
max_death_rate_per_thousand <- US_state_table %>%
  group_by(Province_State) %>%
  summarize(MaxDeathRate = mean(death_rate, na.rm = TRUE))

ggplot(max_death_rate_per_thousand, aes(x = reorder(Province_State, -MaxDeathRate), y = MaxDeathRate)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average COVID-19 Fatal Case Rate (Death Rate for Cases)",
       x = "State",
       y = "Death Rate") +
  coord_flip()
```

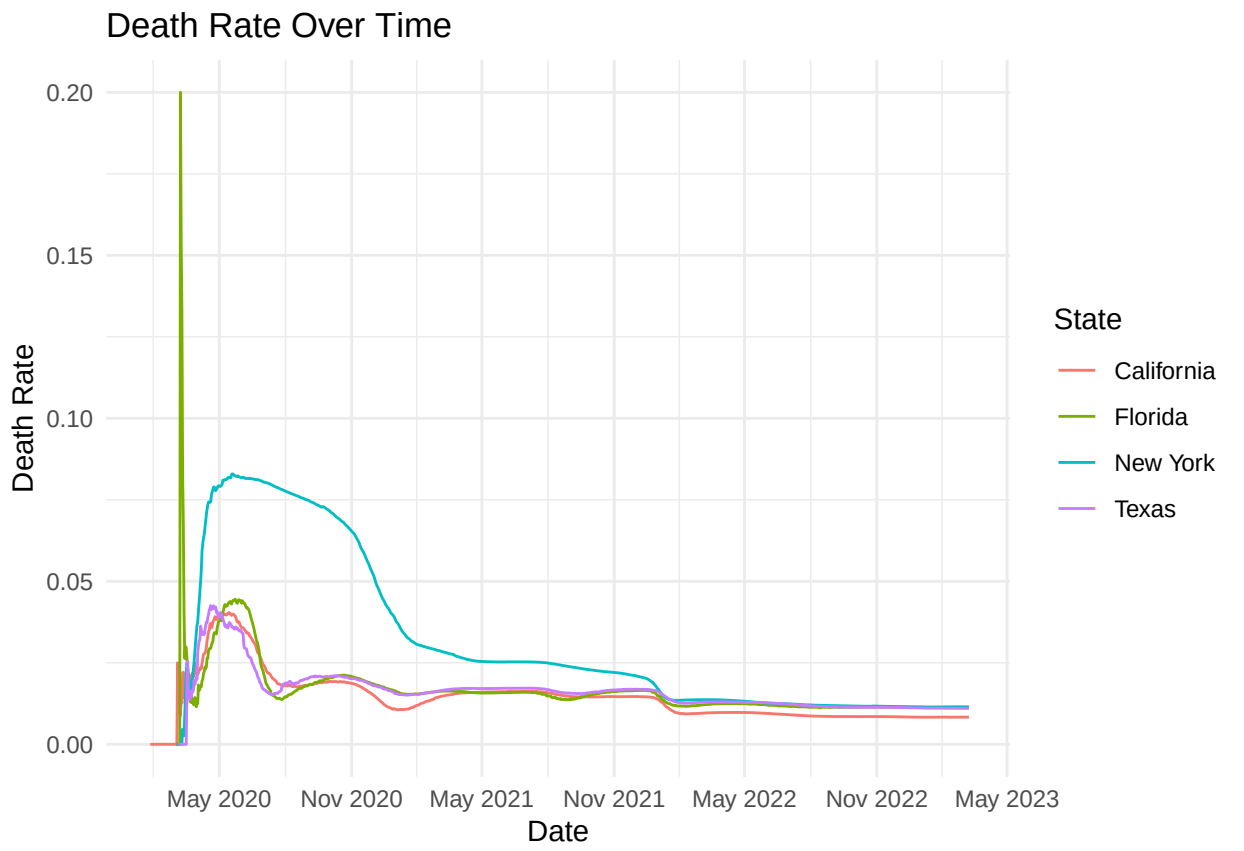


State Comparison: Four Major States Ability to Treat Cases

Looking at four of the largest states, it appears that New York performed worst when treating COVID. This is due to it's consistently high fatal case rate. Notably, Florida was at one point, doing extremely worse than others. This spike is so high, that it could be an issue with the data, but Florida was also notorious for handling COVID poorly, so the spike was potentially related to this.

```
filtered_data <- US_state_table %>%
  filter(Province_State %in% c("Texas", "Florida", "New York", "California"))

# Plotting the data
ggplot(filtered_data, aes(x = date, y = death_rate, color = Province_State)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Death Rate Over Time",
       x = "Date",
       y = "Death Rate",
       color = "State") +
  scale_x_date(date_breaks = "6 month", date_labels = "%b %Y")
```

```
## Warning: Removed 128 rows containing missing values (‘geom_line()‘).
```

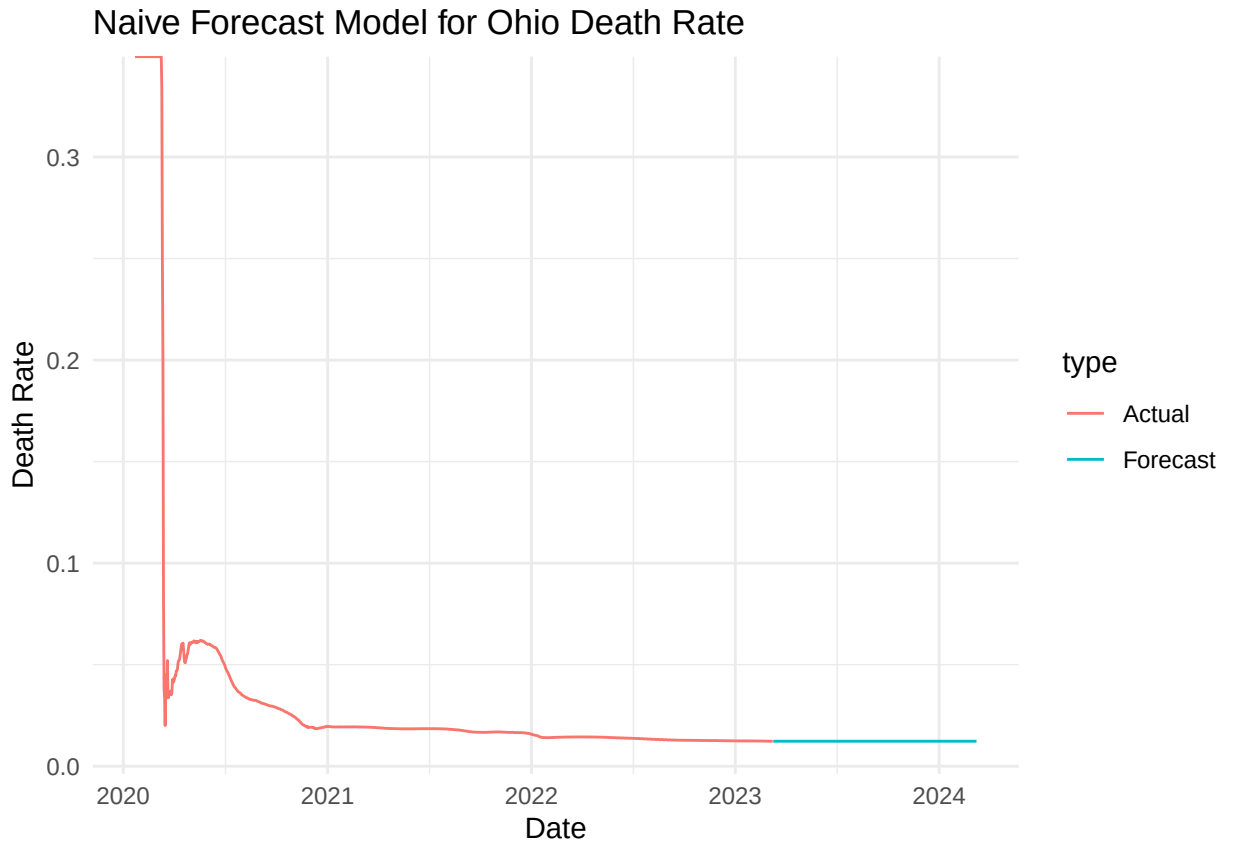


Modeling and Predicting Ohio's Treatement Abilities Over Time

This model projects that Ohio will likely have a small fatal case rate of COVID for the next year (after the data set ends). This could be due to some COVID variants being less deadly, or a better treatment for COVID.

```
# Filter Ohio data and remove rows with NA in death_rate
ohio_data <- US_state_table %>%
  filter(Province_State == "Ohio" & !is.na(death_rate)) %>%
  arrange(date)  # Ensure data is sorted by date

# Convert to a time series object
ts_data <- ts(ohio_data$death_rate, start = c(year(min(ohio_data$date)), yday(min(ohio_data$date))), fr

# Now create a forecast with the naive method
naive_forecast <- naive(ts_data, h=365)

# This part is just for plotting
ts_data_df <- data.frame(date = seq(as.Date(min(ohio_data$date)),
                                    by = "day",
                                    length.out = length(ts_data)),
                         value = as.numeric(ts_data),
                         type = "Actual")

forecast_df <- data.frame(date = seq(as.Date(max(ts_data_df$date)),
```

```
                                    by = "day",
                                    length.out = length(naive_forecast$mean) + 1)[-1], # Exclude the f
                          value = as.numeric(naive_forecast$mean),
                          type = "Forecast")

# Combine actual and forecast data for plotting
combined_df <- rbind(ts_data_df, forecast_df)

ggplot(combined_df, aes(x = date, y = value, color = type)) +
  geom_line() +
  labs(title = "Naive Forecast Model for Ohio Death Rate", x = "Date", y = "Death Rate") +
  theme_minimal()
```



Conclusion

This report explored a variety of statistics including average fatal cases. From this it was determined that Ohio was the worst state at medically treating COVID cases (again, not preventing cases from happening, but preventing cases from turning fatal). Utah was the best state by this metric, and American Samoa was the best out of any US state or territory.

Potential Sources of Bias

Due to public sentiment, I expected right-leaning states to have performed worse with COVID. It turns out that this was not necessarily the case when it came to treatment of COVID.

I handled this bias by basing my conclusions off numeric results, and ensuring that any analysis included a fair number of left and right leaning states.

The data could also be biased in some ways. Sources may have over or under reported cases and deaths. Data could have also been mistakenly biased due to the nature of information on this scale being challenging to record.

Additional Questions

- Would a different model yield different predictions?
- Could other metrics contest the idea that Ohio was worst at treating COVID?

Source

All data comes from this repository: https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/