Reproducible Report on COVID19 Data Final Project 2

Kevin Scroggins

2025-07-05

Introduction

Throughout this project, I will look at reported numbers of the Covid-19 Pandemic from 2020-2023. The data is from John Hopkins University and can be found here https://github.com/CSSEGISandData/COVID-19. The first goal is to create visualizations to help understand the data as whole. After looking at the data, the second goal is to create a model to better understand an aspect of the data.

Setup

The setup imports used libraries.

Importing Data

Note: This section is based on the Professor Wall's Lectures.

The data must be imported, this was done by importing and reading the CSVs. The data itself always a potential source of bias. While it was out of our control what data we were working with, it is worth while to ensure integrity of the data. Potential sources of bias here include where the samples come from, how the data is recorded, and any bias from the observer. The data includes potentially useful variables such as Province/State, Country/Region, date, number of cases, number of deaths, and Population size.

```
file_names <- c("time_series_covid19_confirmed_US.csv",  "time_series_covid19_confirmed_global.csv", "ti

url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi

urls <- str_c(url_in, file_names)

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

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
global_deaths <- read_csv(urls[4])
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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_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[3])
```

```
## 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.
```

```
uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/refs/heads/master/csse_cov
uid <- read_csv(uid_lookup_url)
```

```
## 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.
```

Tidying global data

Note: Note: This is almost exactly the same as done by Professor Wall in Lectures. The goal was to combine the global cases and deaths into one usable table, and focus on interesting variables.

```
global_cases <- global_cases %>%
  pivot_longer(
    cols = matches("^\\d{1,2}/\\d{1,2}/\\d{2}$"),
    names_to = "date",
    values_to = "cases"
  ) %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  select(-c(Lat, Long))

global_deaths <- global_deaths %>%
  pivot_longer(cols = -c(`Province/State`, `Country/Region`, Lat, Long),
               names_to = "date",
               values_to = "death") %>%
  mutate(date = as.Date(date, format = "%m/%d/%y")) %>%
  select(-Lat, -Long)

global <- global_cases %>%
  full_join(global_deaths)
```

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

```
global <- global %>% filter(cases > 0) %>%
  unite("Combined_Key",
        c(`Province/State`, `Country/Region`),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE) %>%
  rename("Province_State" = `Province/State`, "Country_Region" = `Country/Region`)

global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date, cases, death, Population, Combined_Key.y) %>%
  rename("Combined_Key" = Combined_Key.y)

summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:306827      Length:306827      Min.   :2020-01-22   Min.   :        1
##  Class :character   Class :character   1st Qu.:2020-12-12   1st Qu.:     1316
##  Mode  :character   Mode  :character   Median :2021-09-16   Median :    20365
##                                        Mean   :2021-09-11   Mean   :  1032863
##                                        3rd Qu.:2022-06-15   3rd Qu.:   271281
##                                        Max.   :2023-03-09   Max.   :103802702
##
##      death           Population        Combined_Key
##  Min.   :      0   Min.   :6.700e+01   Length:306827
##  1st Qu.:      7   1st Qu.:7.866e+05   Class :character
##  Median :    214   Median :6.948e+06   Mode  :character
##  Mean   :  14405   Mean   :2.890e+07
##  3rd Qu.:   3665   3rd Qu.:2.914e+07
##  Max.   :1123836   Max.   :1.380e+09
##                    NA's   :6729
```

Tidying US data

Note: This is almost exactly the same as done by Professor Wall in Lectures. The goal was to combine the US cases and deaths into one usable table, and focus on interesting variables.

```
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) %>%
  select(Admin2, Province_State, Country_Region, Combined_Key,
         date, cases, Population, deaths) %>%
  rename("County" = Admin2)
```

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

```
summary(US)
```

```
##     County          Province_State     Country_Region     Combined_Key
##  Length:3819906     Length:3819906     Length:3819906     Length:3819906
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases           Population          deaths
##  Min.   :2020-01-22   Min.   :  -3073   Min.   :       0   Min.   :  -82.0
##  1st Qu.:2020-11-02   1st Qu.:    330   1st Qu.:    9917   1st Qu.:    4.0
##  Median :2021-08-15   Median :   2272   Median :   24892   Median :   37.0
##  Mean   :2021-08-15   Mean   :  14088   Mean   :   99604   Mean   :  186.9
##  3rd Qu.:2022-05-28   3rd Qu.:   8159   3rd Qu.:   64979   3rd Qu.:  122.0
##  Max.   :2023-03-09   Max.   :3710586   Max.   :10039107   Max.   :35545.0
```

Visualizing Data for US

Note: The data grouping and first visualization were taken from lecture by Professor Wall. The goal was to group cases and deaths into usable groups as well as look at deaths per capita.

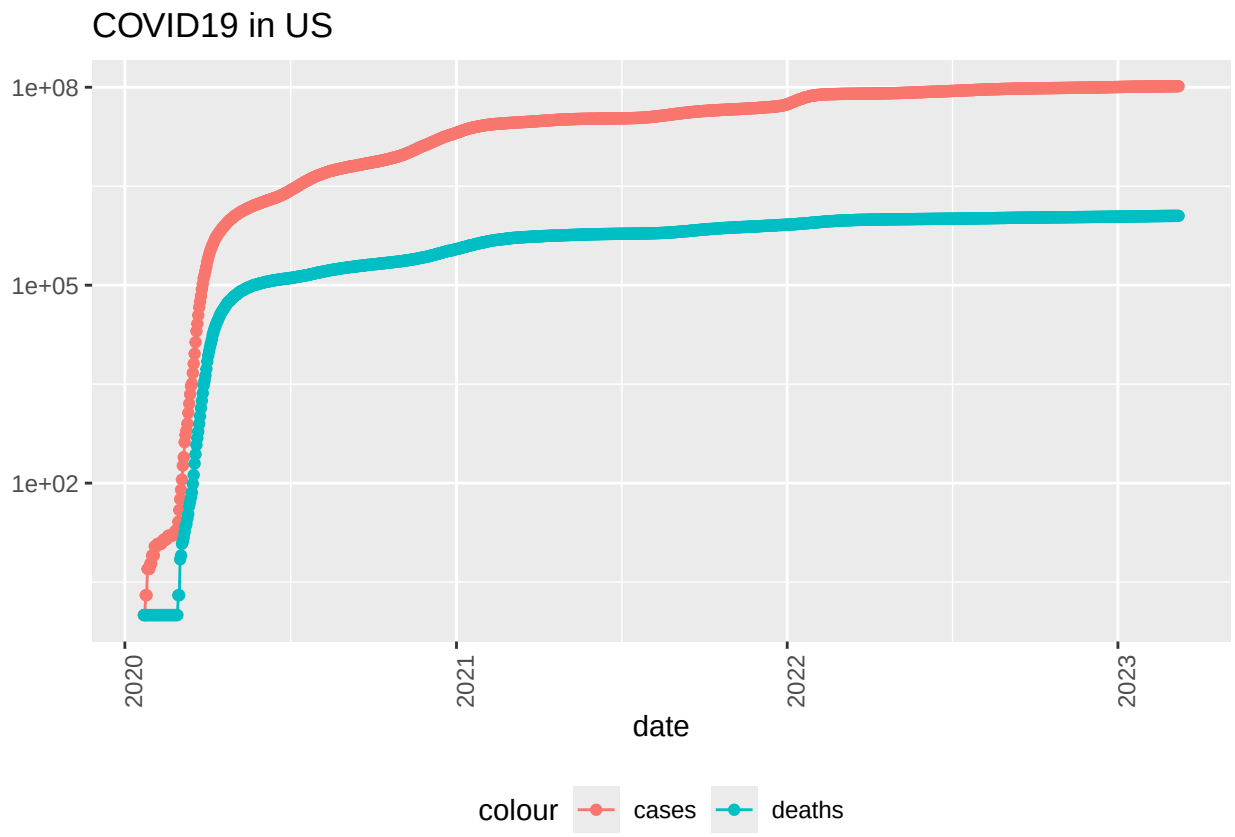
```
US_by_state <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Province_State, Country_Region, date,
         cases,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

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

```
US_totals <- US_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Country_Region, date,
         cases, deaths, deaths_per_mill, Population) %>%
  ungroup()
```

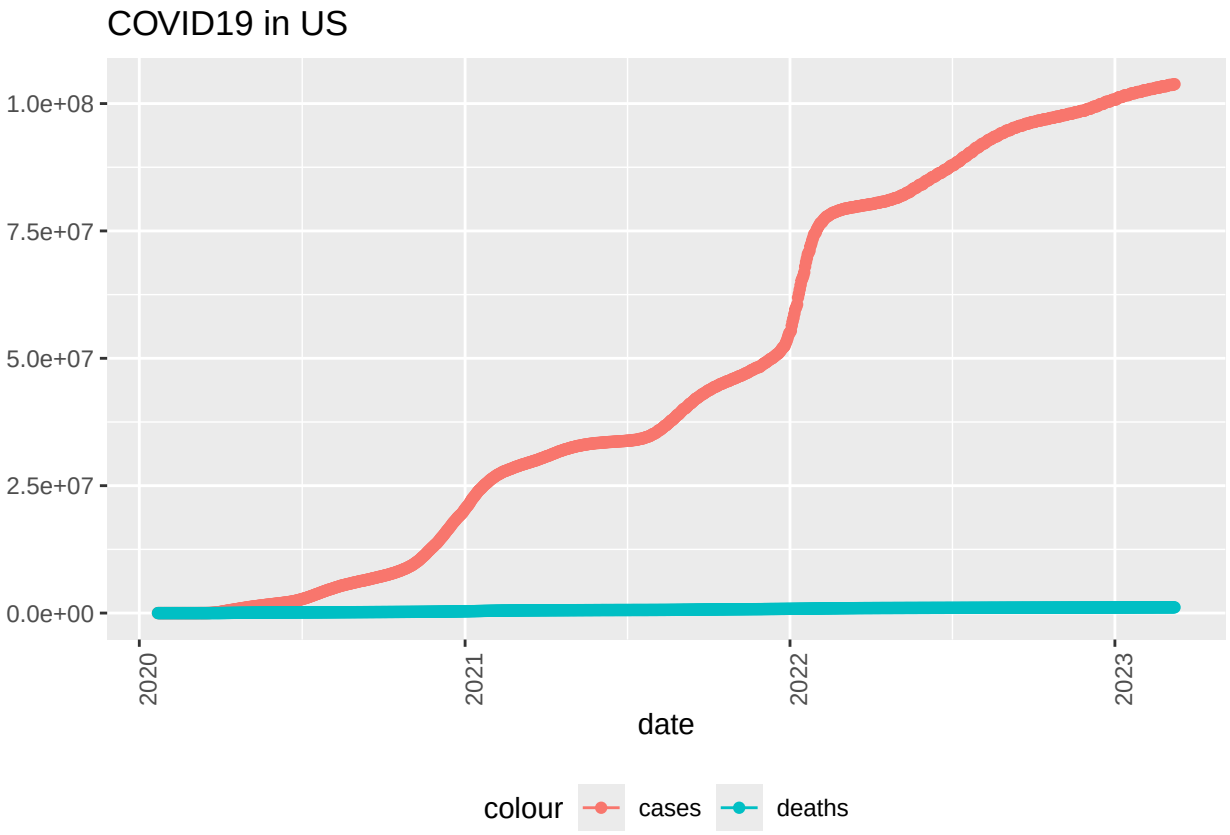
```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

```
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y = NULL)
```



This visualization was shown in the lecture. It uses a logarithmic scale to show the number of cases and deaths in the United States. One source of bias is the logarithmic scale that was used. This scale helps visualize the dramatic rise in cases and deaths early on in the pandemic, and helps put both the deaths and cases in scale. However, it could cause bias in a viewer because it looks flat from 2021.

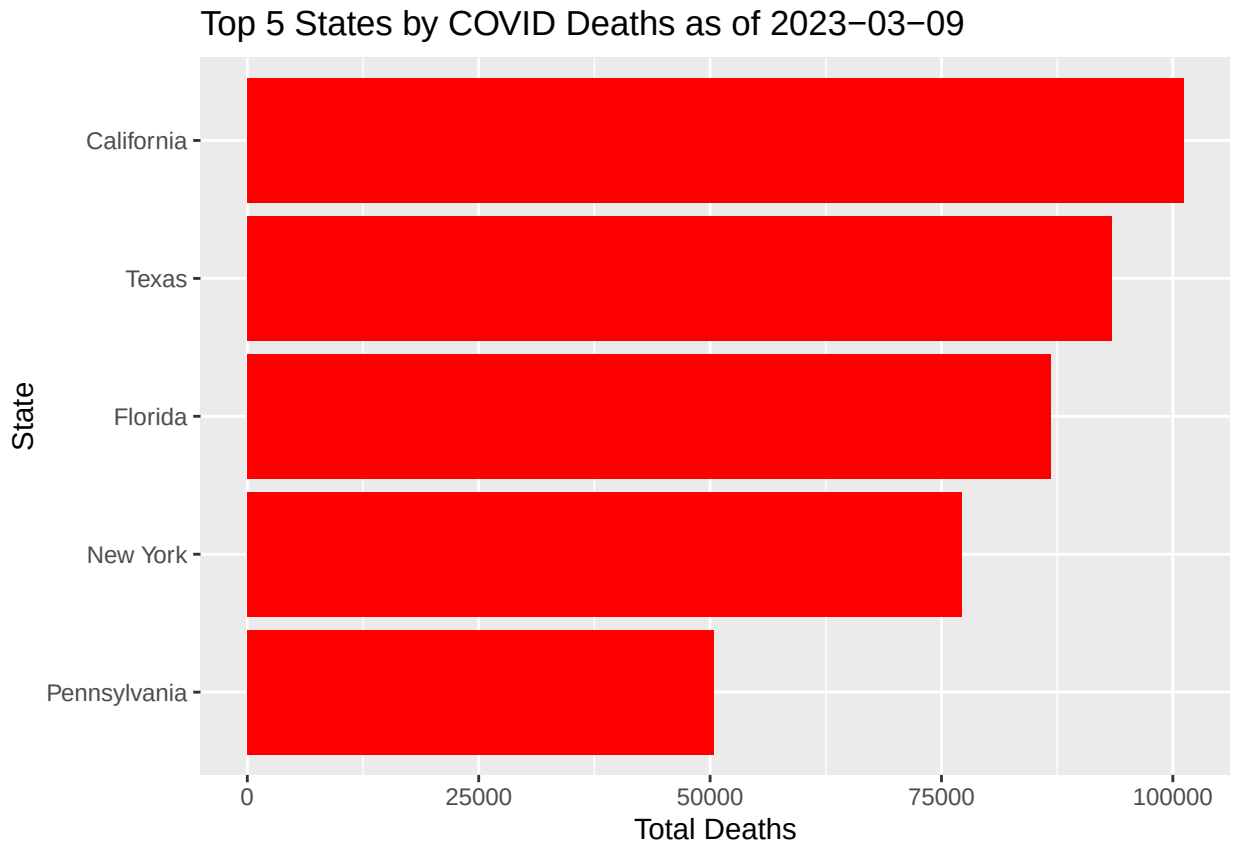
```
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y = NULL)
```



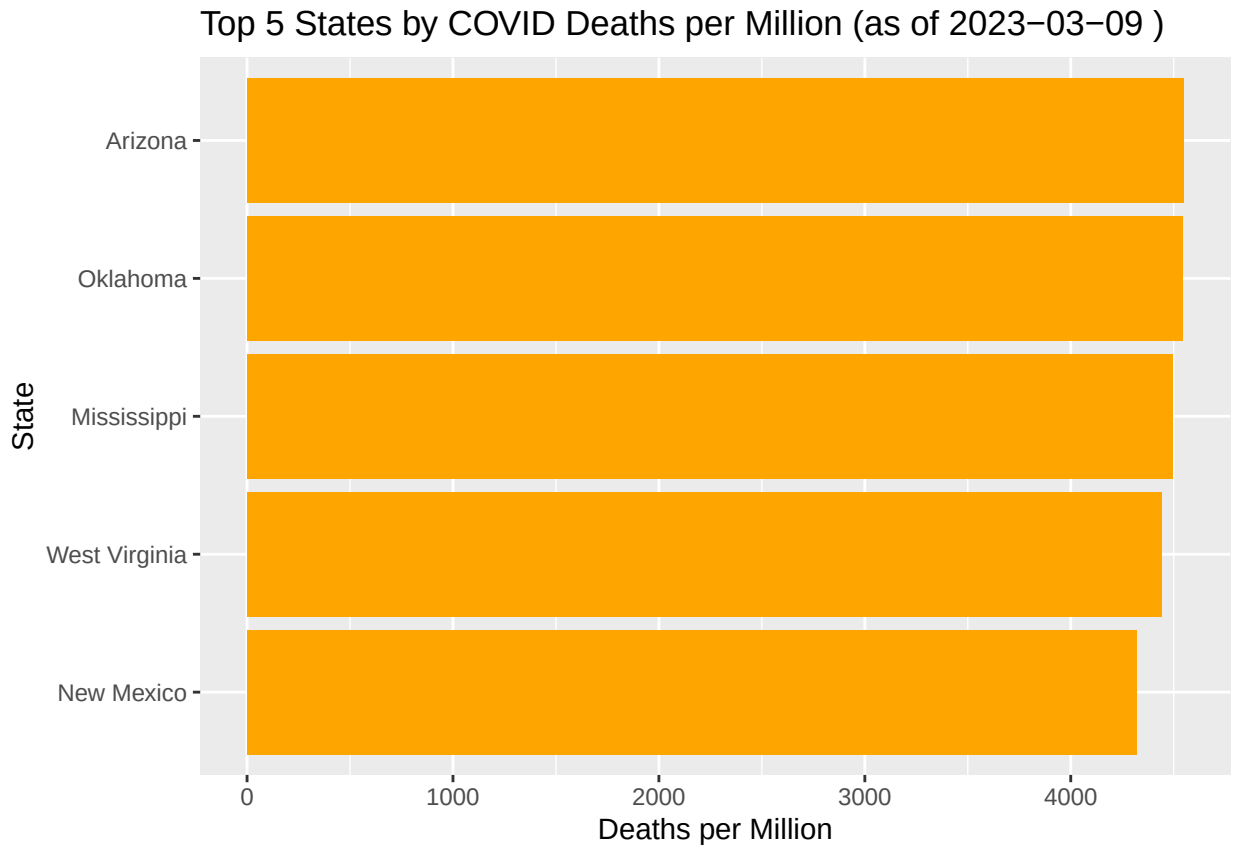
This is the linear scale version of the same plot. This would remove the potential bias of a flat growth of the cases. We can see that the cases continue to rise well past 2021. However, this removes the sense of scale for the deaths. It could cause and observer to think that the number of deaths was much smaller than it really was.

```
latest_date <- max(US_by_state$date)

top_5_states <- US_by_state %>%
  filter(date == latest_date) %>%
  arrange(desc(deaths)) %>%
  slice_head(n = 5)
ggplot(top_5_states, aes(x = reorder(Province_State, deaths), y = deaths)) +
  geom_col(fill = "red") +
  coord_flip() +
  labs(
    title = paste("Top 5 States by COVID Deaths as of", latest_date),
    x = "State",
    y = "Total Deaths"
  )
```




```
US_by_state %>%
  filter(date == latest_date) %>%
  arrange(desc(deaths_per_mill)) %>%
  slice(2:6) %>%
  ggplot(aes(x = reorder(Province_State, deaths_per_mill), y = deaths_per_mill)) +
  geom_col(fill = "orange") +
  coord_flip() +
  labs(
    title = paste("Top 5 States by COVID Deaths per Million (as of", latest_date, ")"),
    x = "State",
    y = "Deaths per Million"
  )
```



I created two visualizations to show the states that had the most Covid deaths, as well as the states with the highest deaths per million. This is an example of how data analysis must take care to analyze and plot data effectively. If only the total deaths were looked at, we get a much different answer than by looking at the per capita deaths of states. This could cause bias when looking at which states handled the pandemic effectively. While both of these visualizations are truthful, they can be used to tell different stories.

```
global_by_day <- global %>%
  group_by(date) %>%
  summarize(
    total_cases = sum(cases, na.rm = TRUE),
    total_deaths = sum(death, na.rm = TRUE),
    total_population = sum(Population, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(
    cases_per_100k = total_cases / total_population * 100000,
    deaths_per_100k = total_deaths / total_population * 100000
  )

mod <- lm(deaths_per_100k ~ cases_per_100k, data = global_by_day)
summary(mod)
```

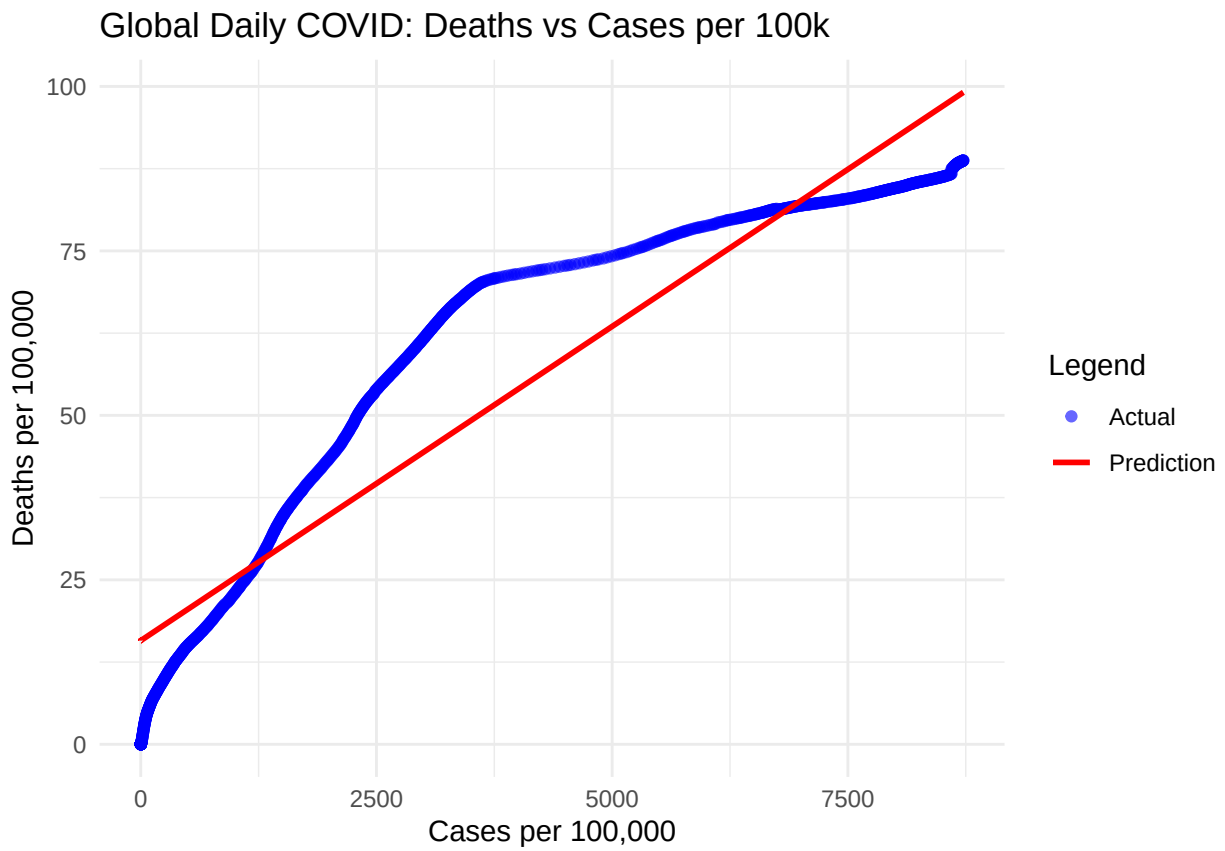
```
##
## Call:
## lm(formula = deaths_per_100k ~ cases_per_100k, data = global_by_day)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.759  -9.092  -2.375   8.741  19.957
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.576e+01  4.882e-01   32.28   <2e-16 ***
## cases_per_100k 9.552e-03  1.032e-04   92.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.79 on 1141 degrees of freedom
## Multiple R-squared:  0.8825, Adjusted R-squared:  0.8824
## F-statistic:  8567 on 1 and 1141 DF,  p-value: < 2.2e-16
```

```
global_by_day <- global_by_day %>%
  mutate(pred = predict(mod, newdata = global_by_day))


ggplot(global_by_day, aes(x = cases_per_100k)) +
  geom_point(aes(y = deaths_per_100k, color = "Actual"), alpha = 0.6) +
  geom_line(aes(y = pred, color = "Prediction"), size = 1) +
  labs(
    title = "Global Daily COVID: Deaths vs Cases per 100k",
    x = "Cases per 100,000",
    y = "Deaths per 100,000",
    color = "Legend"
  ) +
  scale_color_manual(values = c("Actual" = "blue", "Prediction" = "red")) +
  theme_minimal()
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



Similar to the lecture, I made a linear regression to compare the cases per capita vs deaths per capita. I used the global statistics for this comparison, so the data was per 100,000 people. It is looking at a singular global point across the plot. The very small p-value means that this regression is statistically significant. The relatively high R-squared of 0.88 means that the trend is relavent and deaths do follow cases well.

We can see that the actual deaths vary from the cases throughout the time period. It would be interesting to look at how other factors affect this, such as hospital capacity and vaccination rates, and see if the correlates with either the higher or lower than expected death rates.

Conclusion

This markdown file shows an initial look into the Covid 19 dataset from John Hopkins. Throughout this analysis, I discussed potential sources of bias. Bias could come from the data itself, or be caused by interpretations of the visualizations. While the goal is to make visualizations as clear as possible, it can be difficult to remove all bias. I showed how data can be plotted on different scales, each way with its own strengths and weaknesses. I also showed that using the right analysis is key, as the same data can tell different stories.

The model I used could be the start of future research. A potential area for future research would be to see what factors influenced the Covid death rates as opposed to the expectation from the number of cases.

My Session Info

```
sessionInfo()
```

```
## R version 4.4.2 (2024-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.4 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
##  [5] purrr_1.0.4     readr_2.1.5     tidyr_1.3.1     tibble_3.2.1
##  [9] ggplot2_3.5.1   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] bit_4.5.0.1       gtable_0.3.6      crayon_1.5.3      compiler_4.4.2
##  [5] tidyselect_1.2.1  parallel_4.4.2    scales_1.3.0      yaml_2.3.10
##  [9] fastmap_1.2.0     R6_2.5.1          labeling_0.4.3    generics_0.1.3
## [13] curl_6.2.0        knitr_1.49        munsell_0.5.1     pillar_1.10.1
## [17] tzdb_0.4.0        rlang_1.1.5       stringi_1.8.4     xfun_0.50
## [21] bit64_4.6.0-1     timechange_0.3.0  cli_3.6.3         withr_3.0.2
## [25] magrittr_2.0.3    digest_0.6.37     grid_4.4.2        vroom_1.6.5
## [29] rstudioapi_0.17.1 hms_1.1.3         lifecycle_1.0.4   vctrs_0.6.5
## [33] evaluate_1.0.3    glue_1.8.0        farver_2.1.2      colorspace_2.1-1
## [37] rmarkdown_2.29    tools_4.4.2       pkgconfig_2.0.3   htmltools_0.5.8.1
```