Johns Hopkins Data Analysis Final Project

2024-05-08

Understanding, identifying, and visualizing trends in a dataset related to a disease such as COVID-19 is a highly important matter in public health. The dataset used in this report contains data from cases across the world and regions of the United States. It publicly sourced from Johns Hopkins University, and can be found here: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

Import Data

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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
```

```
library(lubridate)
library(ggplot2)


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

# File names
file_names <- c(
  "time_series_covid19_confirmed_US.csv",
  "time_series_covid19_confirmed_global.csv",
  "time_series_covid19_deaths_US.csv",
  "time_series_covid19_deaths_global.csv",
  "time_series_covid19_recovered_global.csv"
)

# Create URLs by pasting base URL and file names
urls <- paste(url_in, file_names, sep = "/")

# Read CSV files
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.
```

```
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.
```

```
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.
```

```
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.
```

```
global_recovered <- read_csv(urls[5])
```

```
## Rows: 274 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.
```

Tidy and Transform - US and Global Data Frames

```
US_data <- bind_rows(
  mutate(US_cases, Type = "Confirmed", Country = "US"),
  mutate(US_deaths, Type = "Deaths", Country = "US")
)

Global_data <- bind_rows(
  mutate(global_cases, Type = "Confirmed", Country = coalesce(`Country/Region`, "Unknown")),
  mutate(global_deaths, Type = "Deaths", Country = coalesce(`Country/Region`, "Unknown")),
  mutate(global_recovered, Type = "Recovered", Country = coalesce(`Country/Region`, "Unknown"))
)

US_data <- US_data %>%
  select(-contains("Lat"), -contains("Long"), -Province_State)

Global_data <- Global_data %>%
  select(-contains("Lat"), -contains("Long"), -`Country/Region`)


US_data <- US_data %>%
  gather(key = "Date", value = "Cases", -Country, -Type) %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%y"))

Global_data <- Global_data %>%
  gather(key = "Date", value = "Cases", -Country, -Type) %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%y"))

US_data <- US_data %>%
  group_by(Country, Type, Date) %>%
  summarize(Cases = sum(as.numeric(Cases), na.rm = TRUE))
```

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

```
Global_data <- Global_data %>%
  group_by(Country, Type, Date) %>%
  summarize(Cases = sum(as.numeric(Cases), na.rm = TRUE))
```

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

```
US_tidy_data <- US_data %>%
  spread(key = Type, value = Cases, fill = 0)

Global_tidy_data <- Global_data %>%
  spread(key = Type, value = Cases, fill = 0)


print(US_tidy_data)
```

```
## # A tibble: 1,144 x 4
```

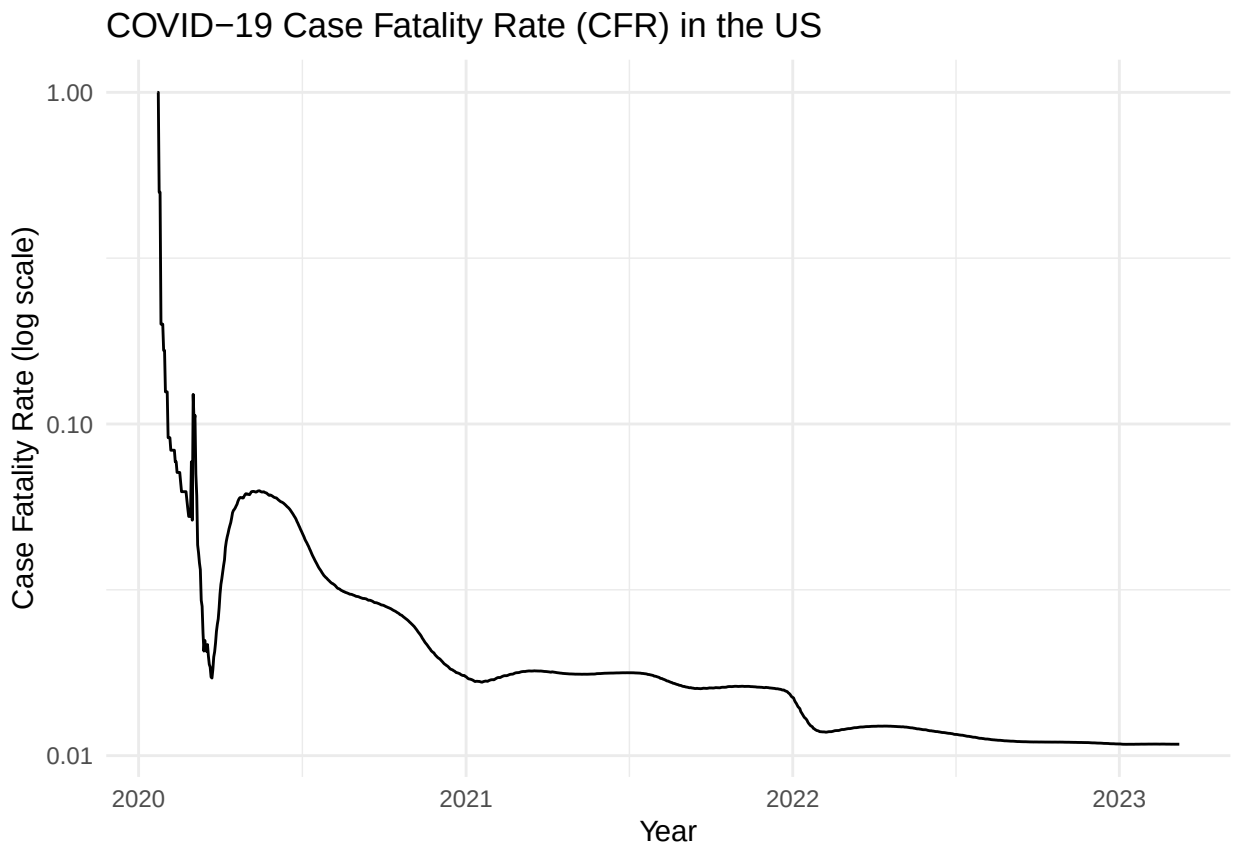
```
## # Groups:   Country [1]
##    Country Date       Confirmed Deaths
##    <chr>   <date>         <dbl>  <dbl>
##  1 US      2020-01-22         1      1
##  2 US      2020-01-23         1      1
##  3 US      2020-01-24         2      1
##  4 US      2020-01-25         2      1
##  5 US      2020-01-26         5      1
##  6 US      2020-01-27         5      1
##  7 US      2020-01-28         5      1
##  8 US      2020-01-29         6      1
##  9 US      2020-01-30         6      1
## 10 US      2020-01-31         8      1
## # i 1,134 more rows
```

```
print(Global_tidy_data)
```

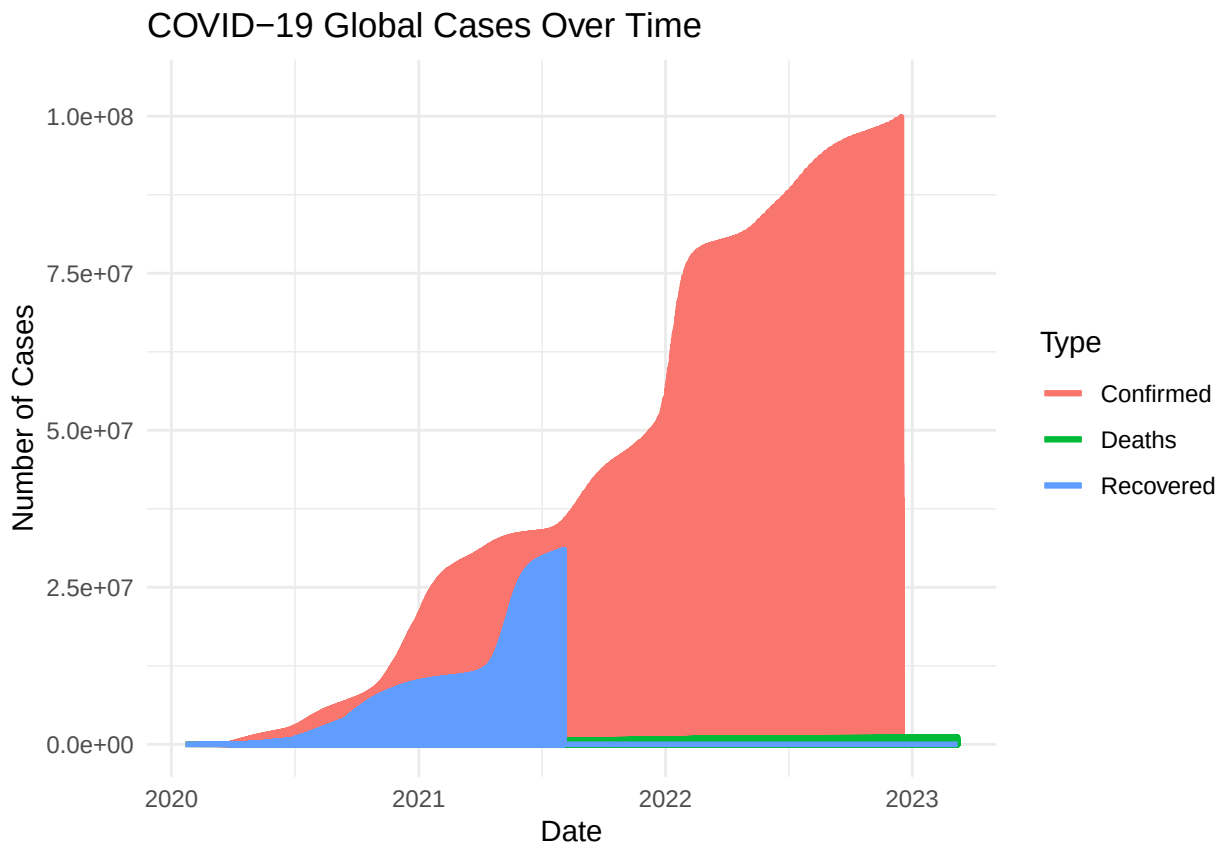
```
## # A tibble: 229,944 x 5
## # Groups:   Country [201]
##    Country     Date       Confirmed Deaths Recovered
##    <chr>       <date>         <dbl>  <dbl>     <dbl>
##  1 Afghanistan 2020-01-22         0      0         0
##  2 Afghanistan 2020-01-23         0      0         0
##  3 Afghanistan 2020-01-24         0      0         0
##  4 Afghanistan 2020-01-25         0      0         0
##  5 Afghanistan 2020-01-26         0      0         0
##  6 Afghanistan 2020-01-27         0      0         0
##  7 Afghanistan 2020-01-28         0      0         0
##  8 Afghanistan 2020-01-29         0      0         0
##  9 Afghanistan 2020-01-30         0      0         0
## 10 Afghanistan 2020-01-31         0      0         0
## # i 229,934 more rows
```

Data Visualization

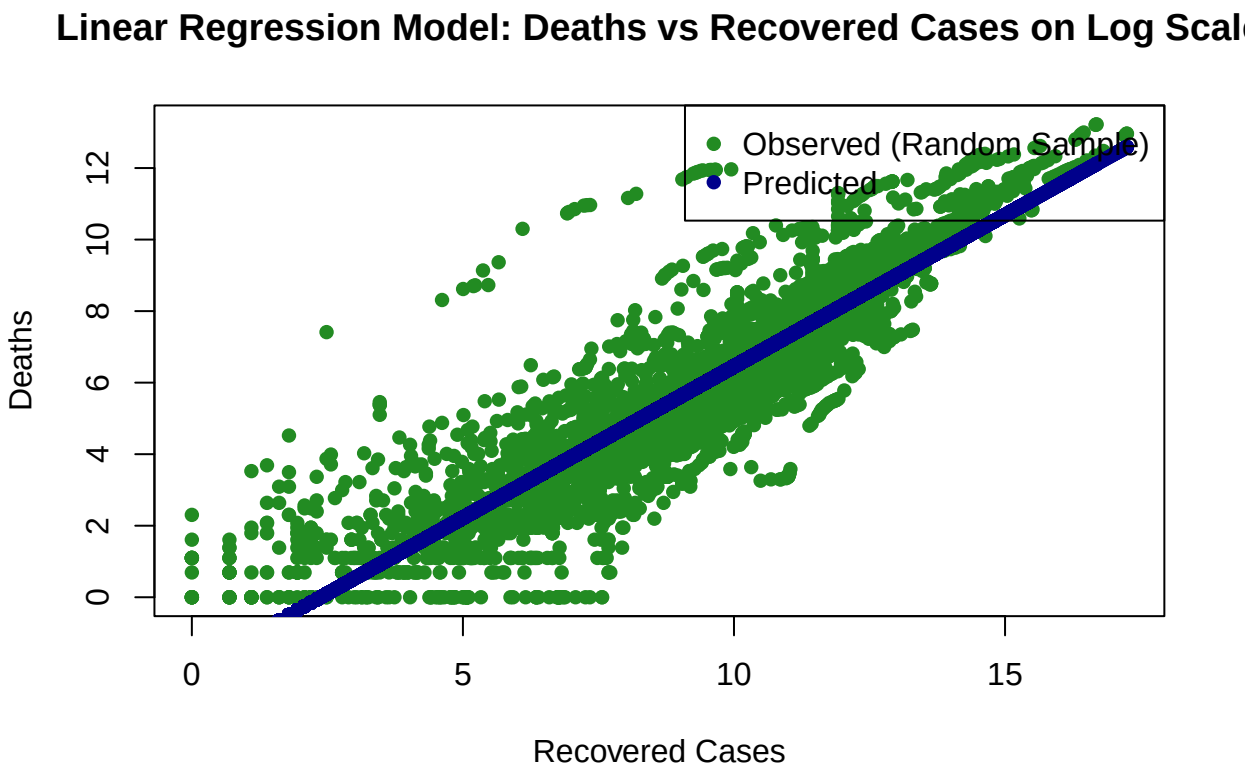
```
#this shows the case fatality rate (so deaths/confirmed cases) over time! note that the Y-axis is a log

US_tidy_data <- US_tidy_data %>%
  arrange(Country, Date)

US_tidy_data <- US_tidy_data %>%
  mutate(CFR = Deaths / Confirmed)

ggplot(US_tidy_data, aes(x = Date, y = CFR)) +
  geom_line() +
  labs(title = "COVID-19 Case Fatality Rate (CFR) in the US",
       y = "Case Fatality Rate (log scale)",
       x = "Year") +
  scale_y_log10() +  # Adding logarithmic scale to y-axis
  theme_minimal()
```




```
#this shows the different types of covid cases worldwide over time

Global_tidy_data <- Global_tidy_data %>%
  arrange(Country, Date)

ggplot(Global_tidy_data, aes(x = Date)) +
  geom_line(aes(y = Confirmed, color = "Confirmed"), size = 1) +
  geom_line(aes(y = Deaths, color = "Deaths"), size = 1) +
  geom_line(aes(y = Recovered, color = "Recovered"), size = 1) +
  labs(title = "COVID-19 Global Cases Over Time",
       y = "Number of Cases",
       x = "Date",
       color = "Type") +
  theme_minimal()
```




```r
#This shows a model of the (logs) of a random sample of deaths from the global dataset and the recovere

data <- Global_tidy_data[, c("Date", "Deaths", "Recovered")]

# adjust data for analysis (further tidying)
data <- data[complete.cases(data$Recovered) & data$Recovered > 0 & complete.cases(data$Deaths) & data$D

# random sampling of the data - allows for a more visually appealing model
set.seed(123)  # Set seed for reproducibility
sample_size <- 5000  # Adjusted to 5000
sample_indices <- sample(1:nrow(data), size = sample_size)
sample_data <- data[sample_indices, ]

linear_model <- lm(log(Deaths) ~ log(Recovered), data = sample_data)

predictions <- exp(predict(linear_model, newdata = data.frame(Recovered = data$Recovered)))


plot(log(sample_data$Recovered), log(sample_data$Deaths), col = "forestgreen", pch = 16, xlab = "Recove
points(log(data$Recovered), log(predictions), col = "darkblue", pch = 16)
legend("topright", legend = c("Observed (Random Sample)", "Predicted"), col = c("forestgreen", "darkblu
```



Bias and Conclusion

There is potential for bias in both the data manipulation as well as the collection portions of working with this dataset. COVID-19 data has been collected for a relatively short time period, but allows for different types of close analyses to understand as much as possible about the disease for public health purposes. One potential bias is the lack of standardized metrics in reporting data from one country or region to another, or different levels in assuming a patient is "recovered" or not. Reinfections can also cause further complexity when working with disease data, and lack of resources and underreporting can further bias the data. In working with the dataset for this project, it is important to note that there was an additional column added for "recovered" cases in the global dataframe, but there was not one for the US dataframe. Dealing with missing data also is a part of mitigating bias, but was handled in the tidying and data transformation process. In addition to this, it is also worth noting that the somewhat sharp cutoff of the "Recovered" cases in the COVID-19 Global Cases OVer Time graph was due to either an error in the dataset/manipulation or even the data reporting during that time period. For instance, a news source (linked below) created a report on how people grew "tired" of COVID-19 in mid-2021, calling it 'Covid Fatigue.' Understanding biases in a broader context is necessary to understand the complete situation. Overall, conducting data on an ever-changing, highly relevant data set such as COVID-19 is very useful for understanding and predicting future challenges with the disease.

Sources

- "It doesn't end. We just stop caring': How a pandemic fades into the background"- Adam Cohen, The Oklahoman (https://www.oklahoman.com/story/opinion/2021/11/07/cohen-how-pandemic-fades-into-background/6267859001/)
- "Time series summary (csse_covid_19_time_series)" Johns Hopkins University, Github Repository (https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series)

```r
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## 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_1.9.3 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
##  [5] purrr_1.0.2     readr_2.1.4     tidyr_1.3.0     tibble_3.2.1
##  [9] ggplot2_3.4.4   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] bit_4.0.5         gtable_0.3.4      highr_0.10        crayon_1.5.2
##  [5] compiler_4.3.2    tidyselect_1.2.0  parallel_4.3.2    scales_1.2.1
##  [9] yaml_2.3.7        fastmap_1.1.1     R6_2.5.1          labeling_0.4.3
## [13] generics_0.1.3    curl_5.1.0        knitr_1.45        munsell_0.5.0
## [17] pillar_1.9.0      tzdb_0.4.0        rlang_1.1.1       utf8_1.2.3
## [21] stringi_1.8.2     xfun_0.41         bit64_4.0.5       timechange_0.2.0
## [25] cli_3.6.1         withr_2.5.2       magrittr_2.0.3    digest_0.6.33
## [29] grid_4.3.2        vroom_1.6.4       rstudioapi_0.15.0 hms_1.1.3
## [33] lifecycle_1.0.3   vctrs_0.6.4       evaluate_0.23     glue_1.6.2
## [37] farver_2.1.1      fansi_1.0.4       colorspace_2.1-0  rmarkdown_2.25
## [41] tools_4.3.2       pkgconfig_2.0.3   htmltools_0.5.7
```