Covid19Data

William Eaton

2025-04-16

0.1 Install packages

Here is a list of all the packages and libraries needed. Un-comment the install line if needed.

```
#install.packages(c("dplyr", "ggplot2", "ggplot2", "tidyr", "stringr", "broom", "tinytex"))
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(ggplot2)
library(tidyr)
library(stringr)
library(broom)
library(tinytex)
```

0.2 Get the data

The data is extracted from the github repository. Acquired from Link Text

```
#Get the data from the URL

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_confirmed_global.csv",
                "time_series_covid19_deaths_US.csv",
                "time_series_covid19_deaths_global.csv")

urls <- str_c(url_in, file_names)

global_cases <- read.csv(urls[2])
```

```
global_deaths <- read.csv(urls[4])

#Look at the data
summary(global_cases)
summary(global_deaths)
head(global_cases)
head(global_deaths)
```

0.3 Cleaning the data

For this project I decided to look specifically into the Canadian data. Therefore, I only used the global cases and global deaths data. I cleaned the data values, removed NA values, and filtered by the country 'Canada'.

```
clean_covid_data <- function(data, region_type = "global") {
  cleaned <- data %>%
    pivot_longer(
      cols = starts_with("X"),
      names_to = "Date",
      values_to = "Count"
    ) %>%
    mutate(
      Date = str_remove(Date, "^X"),
      Date = as.Date(Date, format = "%m.%d.%y")
    ) %>%
    drop_na()  # Remove NA values

  if (region_type == "global") {
    cleaned <- cleaned %>%
      rename(
        Province = `Province.State`,
        Country = `Country.Region`
      ) %>%
      filter(Country == "Canada") %>%
      select(Province, Country, Date, Count)

  } else if (region_type == "us") {
    cleaned <- cleaned %>%
      rename(
        Province = `Province_State`,
        Country = `Country_Region`
      ) %>%
      filter(Country == "Canada") %>%
      select(Province, Country, Date, Count)
  }

  return(cleaned)
}

canadian_cases <- clean_covid_data(global_cases, region_type = "global")
canadian_deaths <- clean_covid_data(global_deaths, region_type = "global")

#Look at the data
```

```
summary(canadian_cases)
summary(canadian_deaths)
head(canadian_cases)
head(canadian_deaths)
```

0.4 Cases and Deaths Over Time

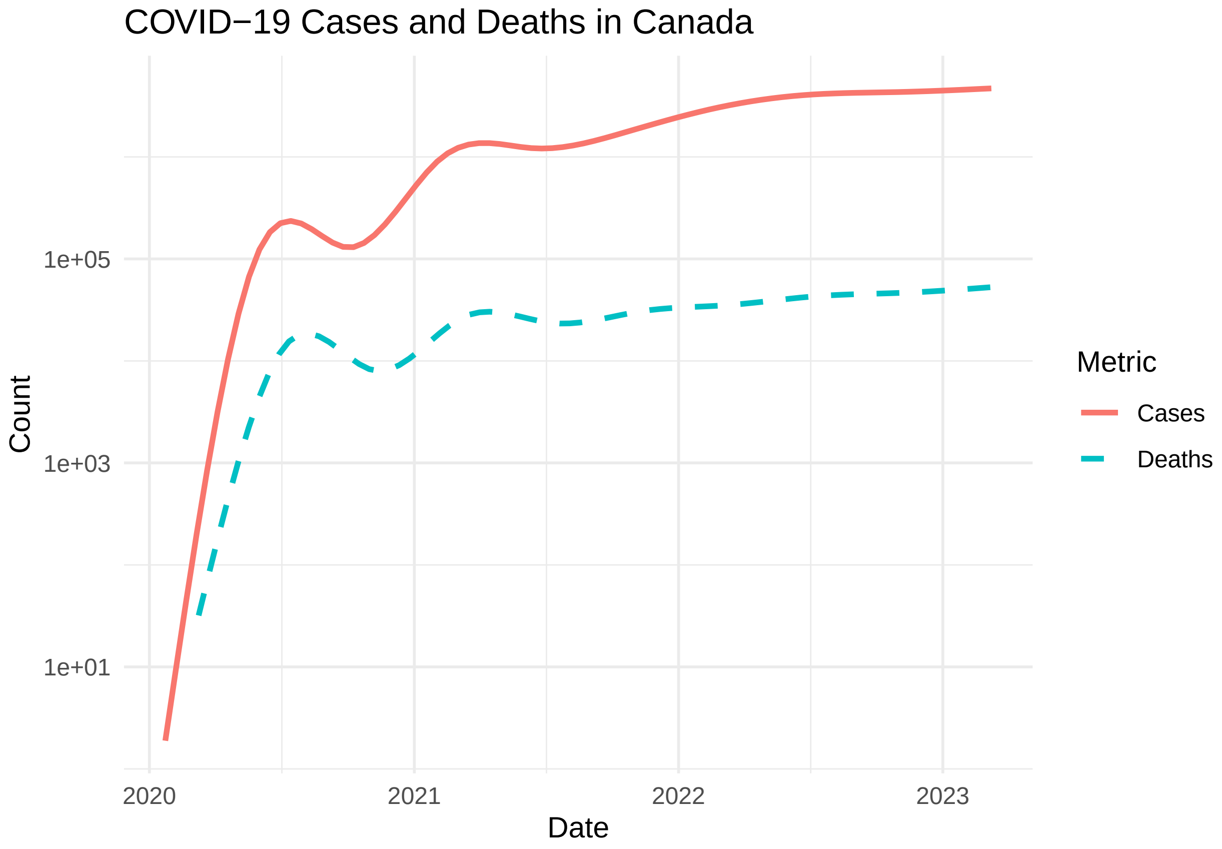
I then plotted a general Cases and Deaths over time plot Using a log scale. I also calculated and plotted the fatality rate: Total Deaths/ Total Cases * 100.

```
cases_summary <- canadian_cases %>%
  group_by(Date) %>%
  summarise(Total_Cases = sum(Count), .groups = "drop")

deaths_summary <- canadian_deaths %>%
  group_by(Date) %>%
  summarise(Total_Deaths = sum(Count), .groups = "drop")

combined <- left_join(cases_summary, deaths_summary, by = "Date") %>%
  filter(Total_Cases > 0)

# Plot 1: Cases and Deaths
ggplot(combined, aes(x = Date)) +
  geom_smooth(aes(y = Total_Cases, color = "Cases"),
              se = FALSE, linetype = "solid", linewidth = 1) +
  geom_smooth(aes(y = Total_Deaths, color = "Deaths"),
              se = FALSE, linetype = "dashed", linewidth = 1) +
  scale_y_log10() +
  labs(
    title = "COVID-19 Cases and Deaths in Canada",
    y = "Count",
    color = "Metric"
  ) +
  theme_minimal()
```

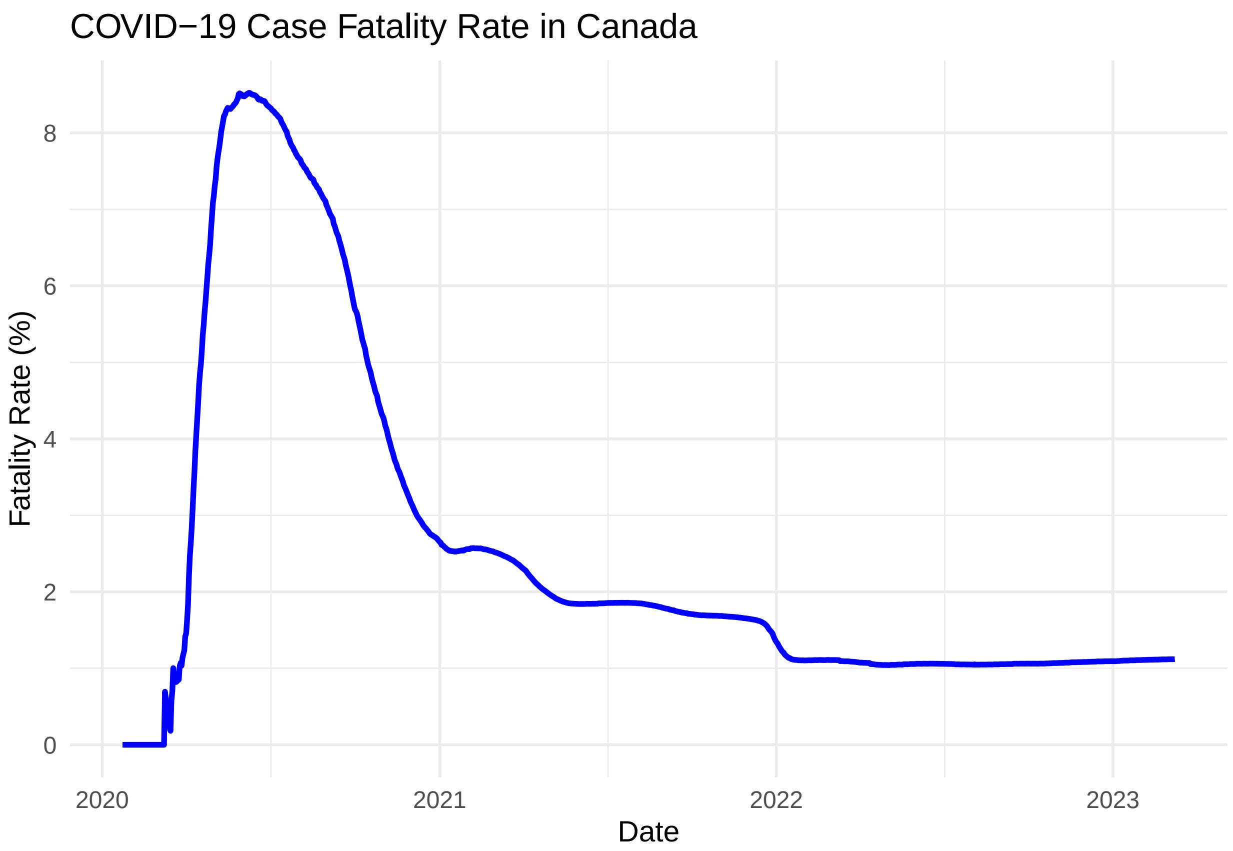
```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
## ‘geom_smooth()‘ using method = ’gam’ and formula = ’y ~ s(x, bs = "cs")’
## ‘geom_smooth()‘ using method = ’gam’ and formula = ’y ~ s(x, bs = "cs")’
```

```
## Warning: Removed 46 rows containing non-finite outside the scale range
## (‘stat_smooth()‘).
```



```r
# Plot 2: Fatality Rate
fatality_data <- combined %>%
  mutate(Fatality_Rate = Total_Deaths / Total_Cases * 100)

ggplot(fatality_data, aes(x = Date, y = Fatality_Rate)) +
  geom_line(color = "blue", linewidth = 1) +
  labs(
    title = "COVID-19 Case Fatality Rate in Canada",
    x = "Date",
    y = "Fatality Rate (%)"
  ) +
  theme_minimal()
```

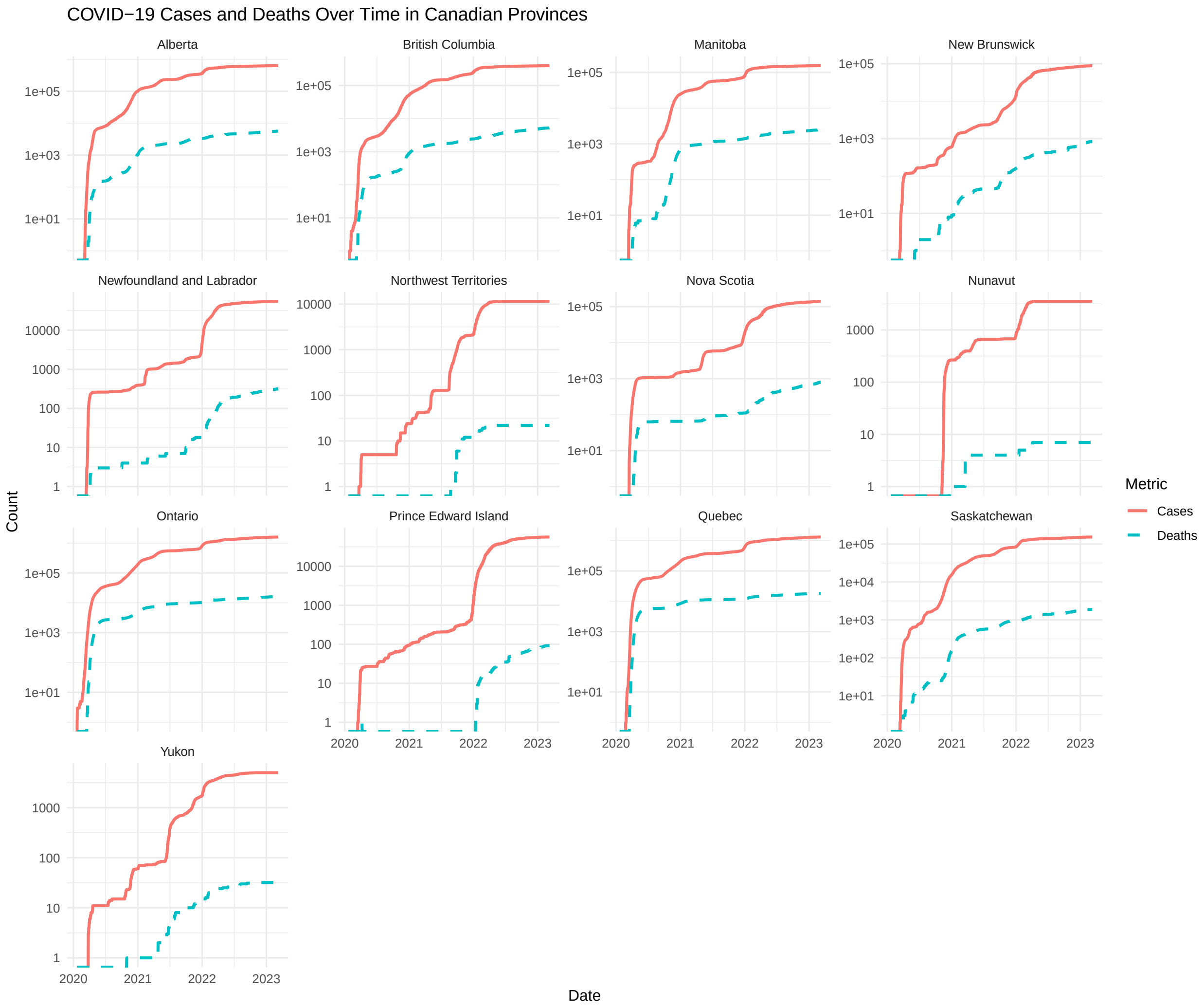


## Cases and Deaths Over Time For each province I then did the same thing, but using the Province Data rather than the whol country.

```
## Canadian Provinces

exclude_provinces <- c("Diamond Princess", "Grand Princess", "Repatriated Travellers")

canadian_cases <- canadian_cases %>%
  filter(!Province %in% exclude_provinces)

canadian_deaths <- canadian_deaths %>%
  filter(!Province %in% exclude_provinces)

cases_by_province <- canadian_cases %>%
  group_by(Province, Date) %>%
  summarise(Total_Cases = sum(Count), .groups = "drop")

deaths_by_province <- canadian_deaths %>%
  group_by(Province, Date) %>%
  summarise(Total_Deaths = sum(Count), .groups = "drop")

canada_combined <- left_join(cases_by_province, deaths_by_province,
                             by = c("Province", "Date"))

ggplot(canada_combined, aes(x = Date)) +
  geom_line(aes(y = Total_Cases, color = "Cases"), linewidth = 1) +
  geom_line(aes(y = Total_Deaths, color = "Deaths"), linetype = "dashed", size = 1) +
  scale_y_log10() +
  facet_wrap(~ Province, scales = "free_y") +
  labs(
    title = "COVID-19 Cases and Deaths Over Time in Canadian Provinces",
    y = "Count",
    color = "Metric"
  ) +
  theme_minimal()
```

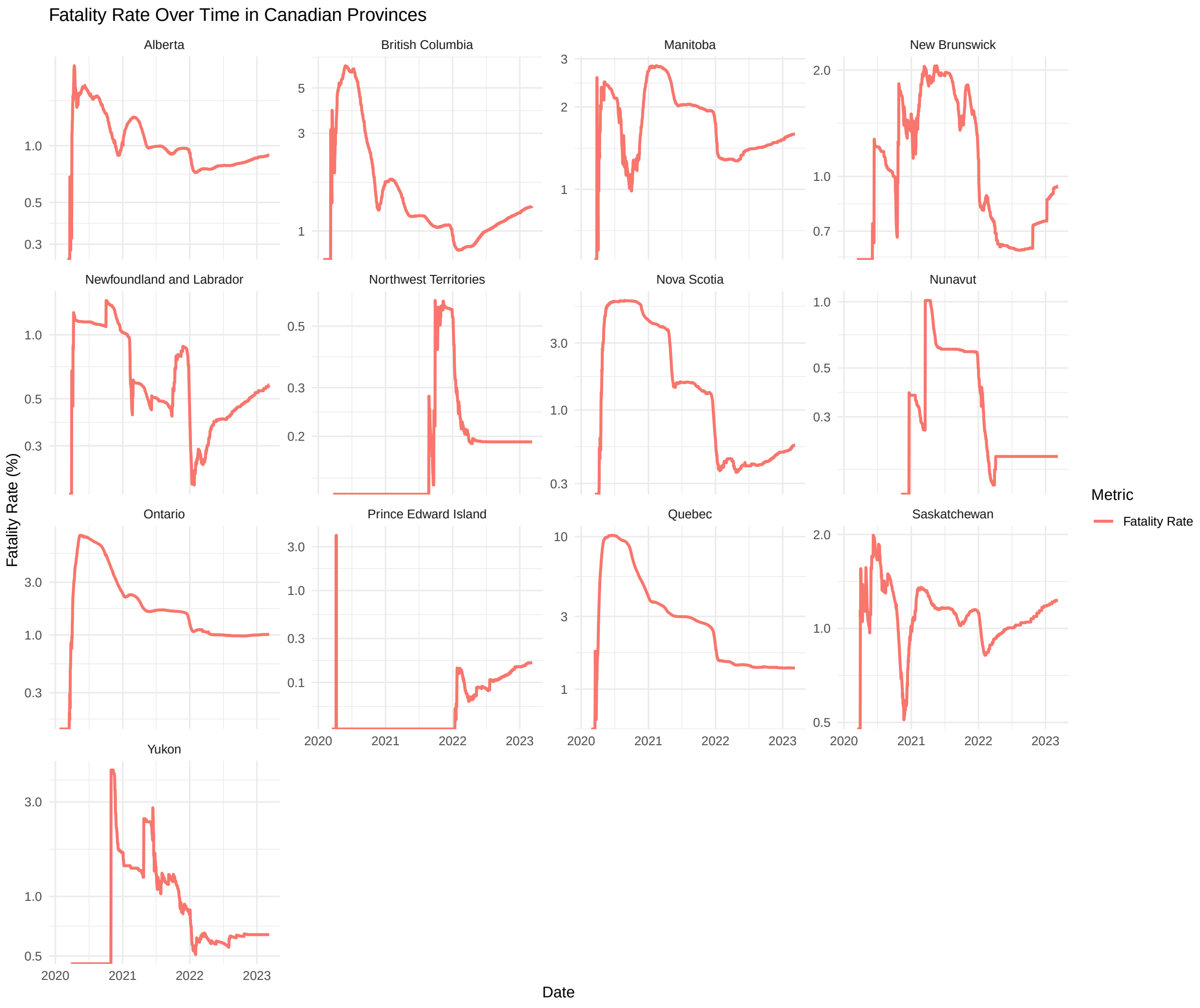
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

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



```r
canada_Fatality_Rate <- canada_combined %>%
  mutate(CFR = case_when(
    Total_Cases > 0 & Total_Deaths > 0 ~ Total_Deaths / Total_Cases * 100,
    Total_Cases > 0 & Total_Deaths == 0 ~ 0,
    TRUE ~ NA_real_
  )) %>%
  filter(!is.na(CFR), is.finite(CFR))

ggplot(canada_Fatality_Rate, aes(x = Date)) +
  geom_line(aes(y = CFR, color = "Fatality Rate"), linewidth = 1) +
  scale_y_log10() +
  facet_wrap(~ Province, scales = "free_y") +
  labs(
    title = "Fatality Rate Over Time in Canadian Provinces",
    y = "Fatality Rate (%)",
    color = "Metric"
  ) +
  theme_minimal()
```

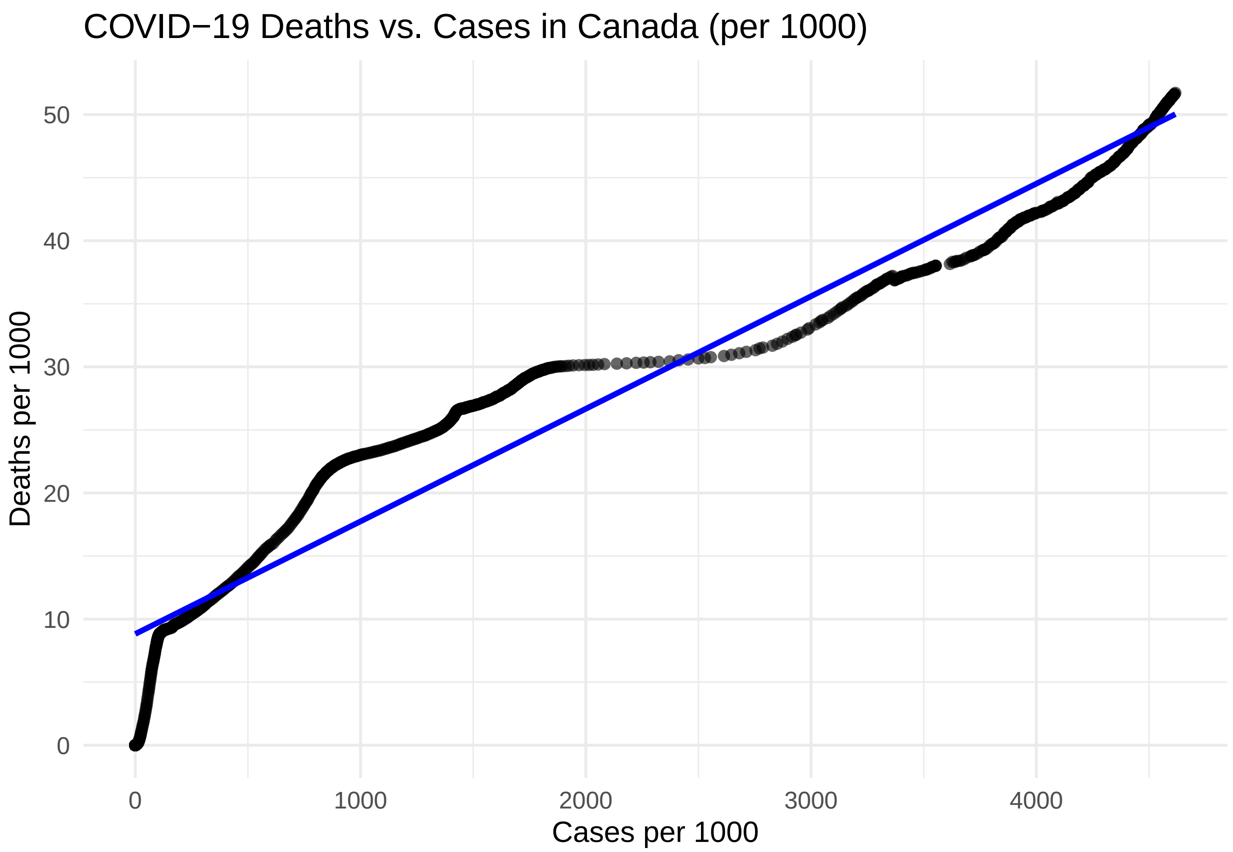
```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```



0.5 Cases and Deaths per thousand.

Finally, I plotted the cases and Deaths per thousands and modeled a linear fit to the data.

```r
deaths_summary <- canadian_deaths %>%
  group_by(Date) %>%
  summarise(Total_Deaths = sum(Count), .groups = "drop")

scatter_data <- left_join(cases_summary, deaths_summary, by = "Date") %>%
  filter(Total_Cases > 0) %>%
  mutate(
    Cases_per_1000  = Total_Cases  / 1000,
    Deaths_per_1000 = Total_Deaths / 1000
  )

scatter_data <- scatter_data %>%
  select(Cases_per_1000, Deaths_per_1000)

ggplot(scatter_data, aes(x = Cases_per_1000, y = Deaths_per_1000)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(
    title = "COVID-19 Deaths vs. Cases in Canada (per 1000)",
    x = "Cases per 1000",
    y = "Deaths per 1000"
  ) +
  theme_minimal()
```



```
lm_fit <- lm(Deaths_per_1000 ~ Cases_per_1000, data = scatter_data)

cat("=== Base R summary ===\n")
```

```
## === Base R summary ===
```

```
print(summary(lm_fit))
```

```
##
## Call:
## lm(formula = Deaths_per_1000 ~ Cases_per_1000, data = scatter_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8266 -2.2174 -0.8196  4.3834  5.4868
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.823e+00  1.764e-01    50.0   <2e-16 ***
## Cases_per_1000 8.925e-03  6.797e-05   131.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.855 on 1140 degrees of freedom
## Multiple R-squared:  0.938,  Adjusted R-squared:  0.9379
## F-statistic: 1.724e+04 on 1 and 1140 DF,  p-value: < 2.2e-16
```

```
cat("\n=== Coefficients (tidy) ===\n")
```

```
##
## === Coefficients (tidy) ===
```

```
print(tidy(lm_fit))
```

```
## # A tibble: 2 x 5
##   term           estimate std.error statistic   p.value
##   <chr>             <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)     8.82    0.176          50.0 1.11e-289
## 2 Cases_per_1000  0.00893 0.0000680     131.  0
```

```
cat("\n=== Model Fit Statistics (glance) ===\n")
```

```
##
## === Model Fit Statistics (glance) ===
```

```
print(glance(lm_fit))
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.938         0.938  3.86    17244.       0     1 -3160. 6327. 6342.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

0.6 Analysis.

Compared to the American data we saw in class, we see that Canada had a similar trend in terms of the cases and deaths over time. We see that both cases and deaths sharpely incresed in the early dates but then gradually relaxed and approached a constant amount. We are also able to see that the fatality rate was much higher during earlier times but also settled down closer to zero later on.

When looking at the Province Data, we see that larger provinces with larger populations (Quebec, Ontario, etc…) had similar trends to the Country wide Canada data. However, Provinces with smaller populations had both smaller cases and death counts. Their fatality rates were also significantely smaller.