COVID project

2023-02-19

Include the required libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Importing data

The present study entails the acquisition of COVID-19 data from a reliable and reputable source.

```
# Record urls
urls <- c('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cov
uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/

# Import 4 csvs
US_cases <- read.csv(urls[1])
US_deaths <- read.csv(urls[2])
global_cases <- read.csv(urls[3])
global_deaths <- read.csv(urls[4])

# Import population data
uid <- read.csv(uid_lookup_url) %>%
  select(c(Province_State, Country_Region, Population)) %>%
  mutate_all(~na_if(., ''))
```

Tidying and transforming data

Tidy and transform imported data, create new columns for analysis.

```
# Create a data frame for global cases
global_cases <- global_cases %>%
  # Remove some columns
  select(-c(Lat, Long)) %>%
  # show date and cases in 2 columns
  pivot_longer(cols = -c(Province.State,
                         Country.Region),
               names_to = "date",
               values_to = "cases")

# Create a data frame for global deaths
global_deaths <- global_deaths %>%
  select(-c(Lat, Long)) %>%
  pivot_longer(cols = -c(Province.State,
                         Country.Region),
               names_to = "date",
               values_to = "deaths")

# Join global_cases and global_deaths to create final global data frame
global <- global_cases %>%
  full_join(global_deaths) %>%
  # Change date in correct format
  transform(date = gsub("X","",as.character(date))) %>%
  # Transform date from character to date object
  mutate(date = mdy(date)) %>%
  # Change all blank data to NA
  mutate_all(~na_if(., '')) %>%
  # Create a Combined_Key column
  unite("Combined_Key",
        c(Province.State, Country.Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE) %>%
  rename(Country_Region = Country.Region,
         Province_State = Province.State) %>%
  # Join population data to the data frame
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  # Select columns for analysis
  select(Province_State, Country_Region, date,
         cases, deaths, Population, Combined_Key)
```

```
## Joining, by = c("Province.State", "Country.Region", "date")
```

```
# Create a data frame for US cases
US_cases <- US_cases %>%
  pivot_longer(cols = -(UID:Combined_Key),
               names_to = "date",
               values_to = "cases") %>%
  select(Admin2:cases) %>%
```

```
  select(-c(Lat, Long_))

# Create a data frame for US deaths
US_deaths <- US_deaths %>%
  pivot_longer(cols = -(UID:Population),
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  select(-c(Lat, Long_))

# Join US_cases and US_deaths to create final US data frame
US <- US_cases %>%
  full_join(US_deaths) %>%
  transform(date = gsub("X","",as.character(date))) %>%
  mutate(date = mdy(date))
```

```
## Joining, by = c("Admin2", "Province_State", "Country_Region", "Combined_Key",
## "date")
```

```
# Show summary for global data
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:324836      Length:324836      Min.   :2020-01-22   Min.   :        0
##  Class :character   Class :character   1st Qu.:2020-10-28   1st Qu.:      651
##  Mode  :character   Mode  :character   Median :2021-08-05   Median :    13714
##                                        Mean   :2021-08-05   Mean   :   936098
##                                        3rd Qu.:2022-05-13   3rd Qu.:   222900
##                                        Max.   :2023-02-18   Max.   :103123281
##
##      deaths           Population        Combined_Key
##  Min.   :      0   Min.   :6.700e+01   Length:324836
##  1st Qu.:      3   1st Qu.:5.790e+05   Class :character
##  Median :    146   Median :6.574e+06   Mode  :character
##  Mean   :  13204   Mean   :2.769e+07
##  3rd Qu.:   2989   3rd Qu.:2.642e+07
##  Max.   :1117497   Max.   :1.380e+09
##                    NA's   :10116
```

```
# Show summary for US data
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:3756408     Length:3756408     Length:3756408     Length:3756408
##  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.:2020-10-28   1st Qu.:    316   1st Qu.:    9917   1st Qu.:    4.0
```

```
##  Median :2021-08-05   Median :   2214   Median :   24892   Median :   36.0
##  Mean   :2021-08-05   Mean   :  13802   Mean   :   99604   Mean   :  184.4
##  3rd Qu.:2022-05-13   3rd Qu.:   7969   3rd Qu.:   64979   3rd Qu.:  120.0
##  Max.   :2023-02-18   Max.   :3691301   Max.   :10039107   Max.   :35355.0
```

Data Analysis

Canada data analysis

Retrieve Canadian data by extracting it from the global data set.

```
# Extract Canada data from global data
Canada <- global %>%
  filter(Country_Region == "Canada")

# Group Canada data by province
Canada_by_province <- Canada %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  ungroup()
```

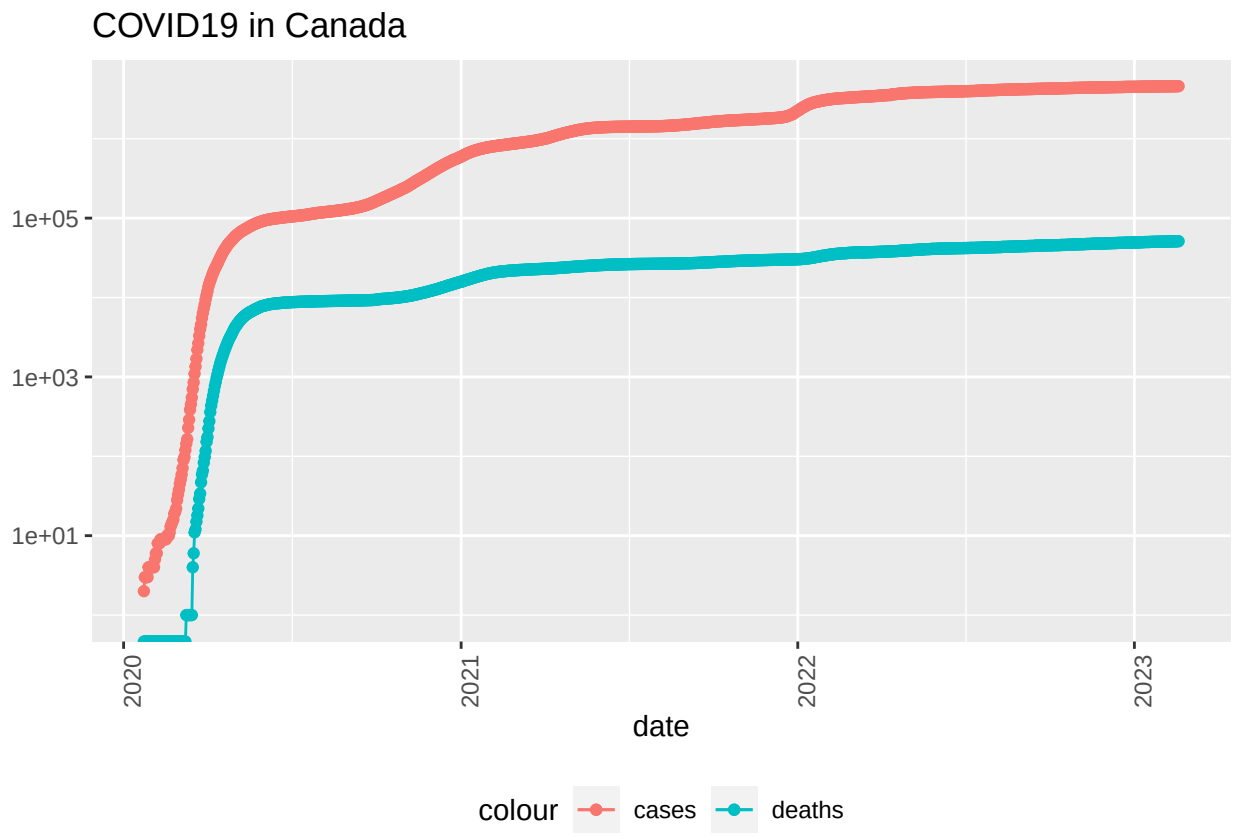
```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can
## override using the `.groups` argument.
```

```
# Calculate Canada's national data
Canada_totals <- Canada_by_province %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population,  na.rm = TRUE)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  ungroup()
```

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

```
# Plot a log-scale line graph that displays the cumulative COVID-19 cases and deaths in Canada over tim
Canada_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 Canada", y = NULL)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```



The presented graph illustrates the trend of cumulative COVID-19 cases and related fatalities in Canada. The recorded cases begin from February 2020 and exhibit a sharp upward trajectory before stabilizing at a plateau.

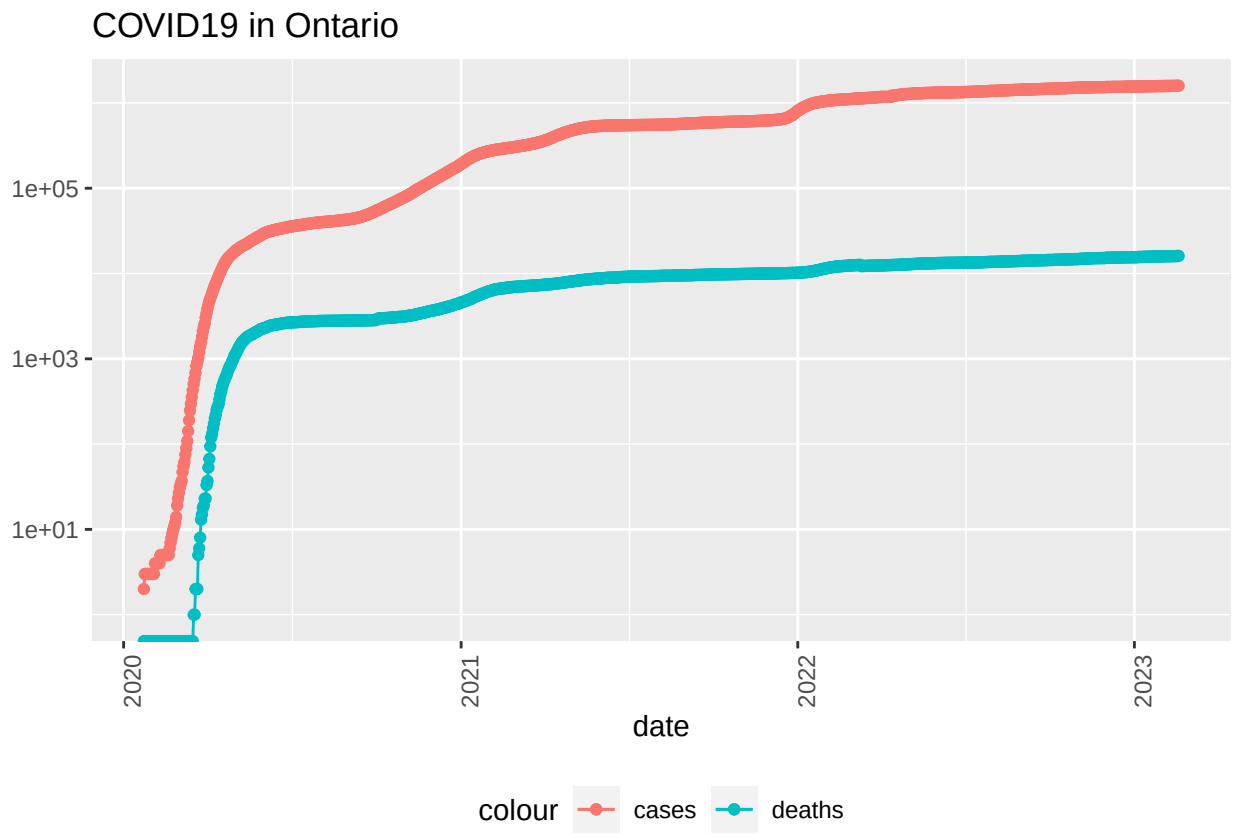
Canadian province analysis

The process of extracting provincial data from the national dataset is undertaken, followed by the development of a function for generating a graph of the number of COVID-19 cases over time for each province.

```
# Create a function to plot a graph for Province
plot_province <- function(province) {
  Canada_by_province %>%
  filter(Province_State == province) %>%
  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 = str_c("COVID19 in ", province), y = NULL)
}
```

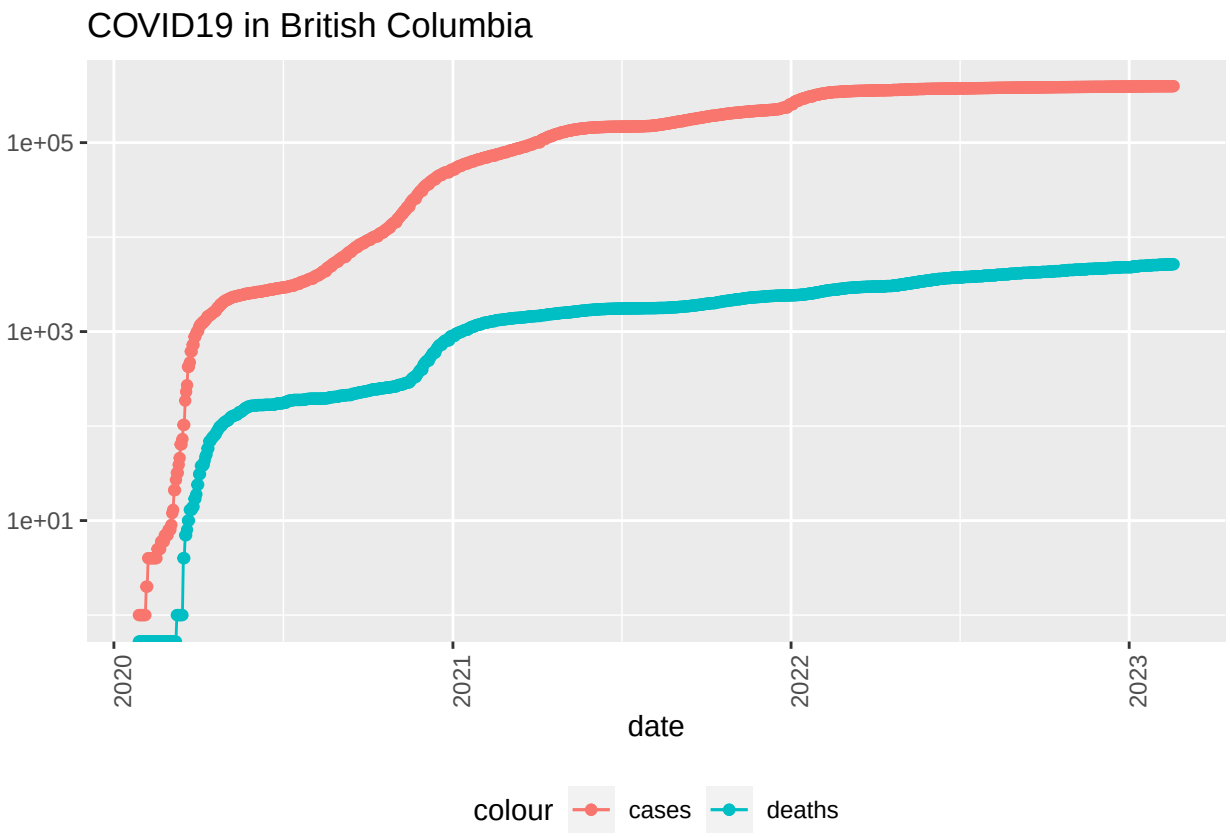
```
# Plot a graph for Ontario
plot_province("Ontario")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```




```
# Plot a graph for British Columbia
plot_province("British Columbia")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```



The initial graph demonstrates the progression of COVID-19 cases in Ontario, while the subsequent graph displays the same for British Columbia. Although both graphs exhibit comparable trends to the national trend, minor variations exist within the first year of data.

New cases analysis

A calculation of new COVID-19 cases was performed for the Canadian data set, with the results tabulated on a monthly basis.

```
Canada_totals %>%
  # Extract year and month part from date
  mutate(year_month = format_ISO8601(date, precision = "ym")) %>%
  # Calculate new cases and deaths
  mutate(new_cases = cases - lag(cases),
      new_deaths = deaths - lag(deaths)) %>%
  # Group the data by month
  group_by(year_month) %>%
  summarize(new_cases = sum(new_cases), new_deaths = sum(new_deaths)) %>%
  ungroup() %>%
  # Plot a line graph with log-scale y-axis.
  ggplot(aes(x = year_month, y = new_cases)) +
  geom_line(aes(color = "new_cases", group = 1)) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths", group = 1)) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
```

```
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "New COVID19 cases in Canada", y = NULL)
```

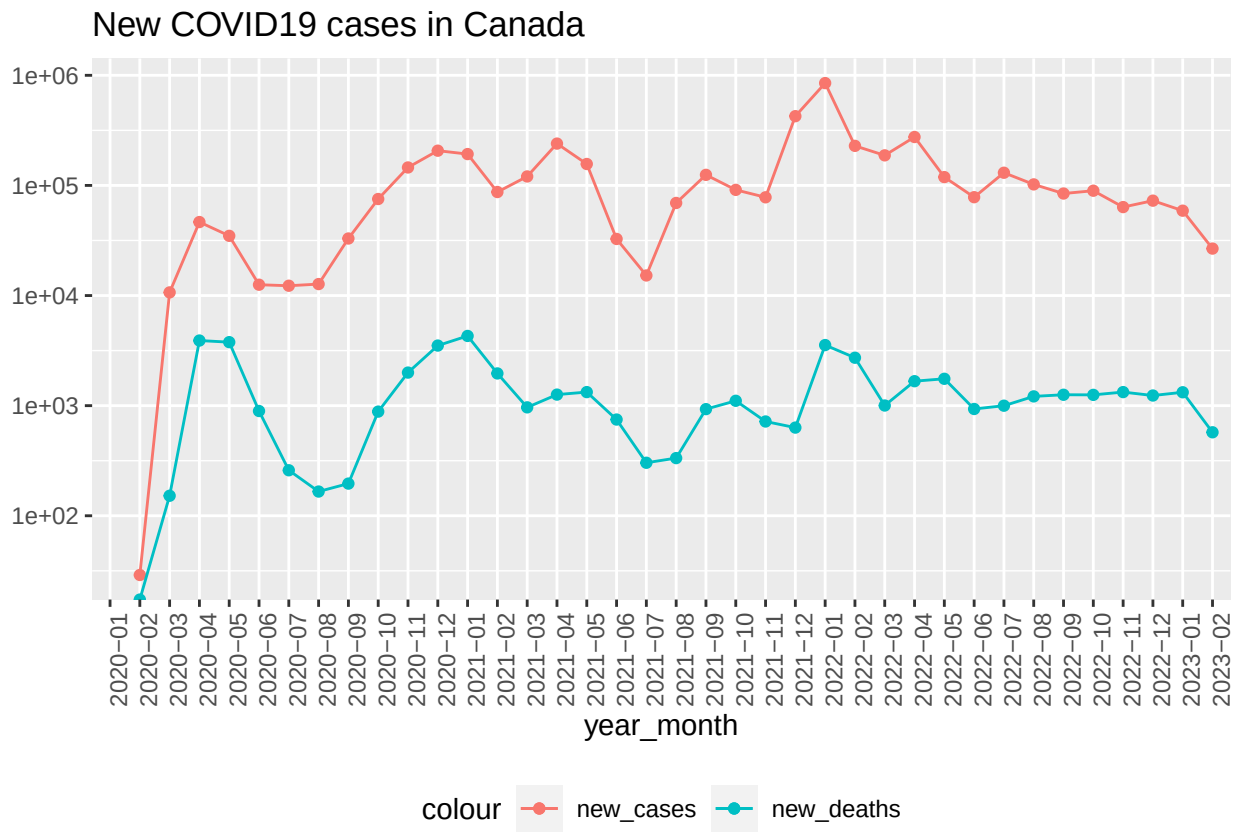
```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



The graph displays the monthly new COVID-19 cases starting from February 2020, revealing a pronounced surge during the winter months and a decline during the summer months. The trend stabilized towards the end of 2022 and started to decrease gradually.

Global trend

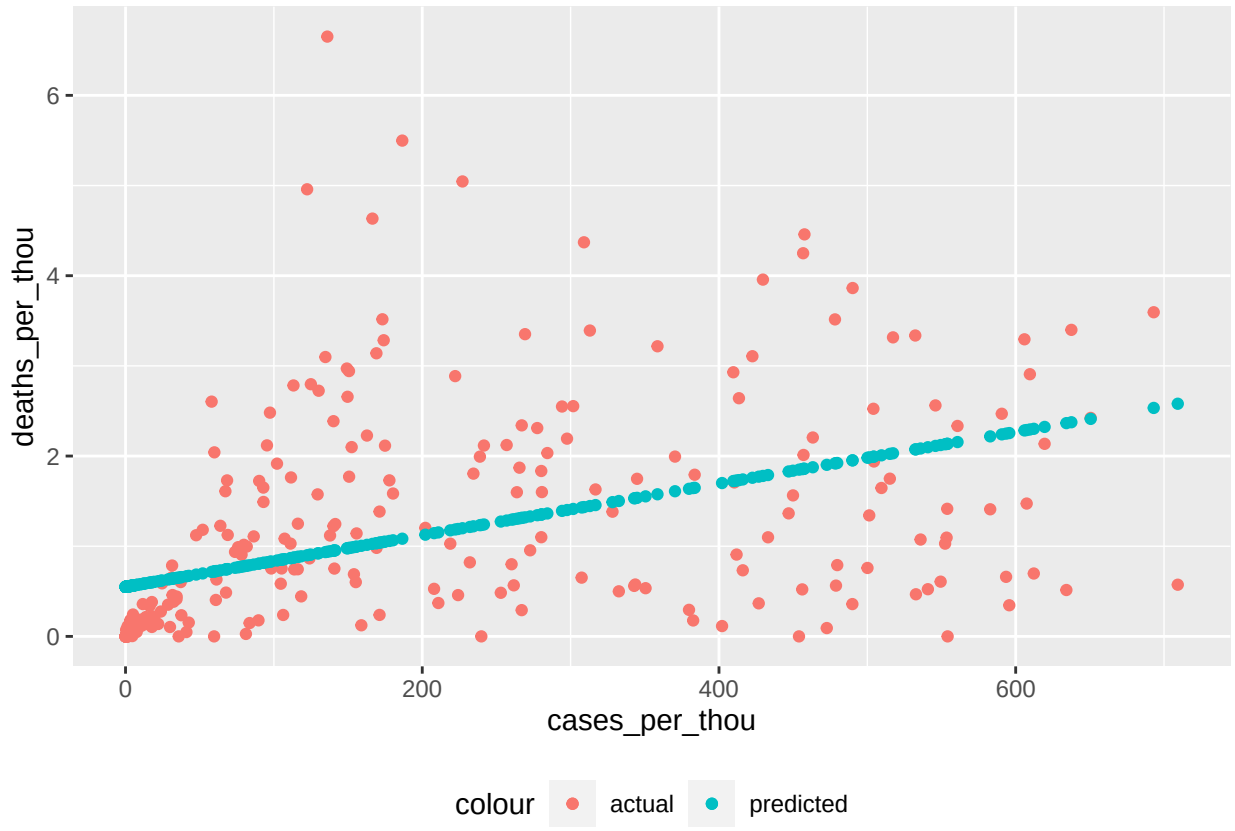
To predict a linear model for deaths per thousand and cases per thousand, we calculated these values for each region in the data.

```r
# Create data for every region in global data
global_total <- global %>%
  group_by(Combined_Key) %>%
  # Calculate deaths per thousand and cases per thousand
  summarize(deaths = max(deaths), cases = max(cases),
            Population = max(Population),
            cases_per_thou = 1000 * cases / Population,
            deaths_per_thou = 1000 * deaths / Population) %>%
  filter(cases > 0, Population > 0)

# Create a linear model
mod <- lm(deaths_per_thou ~ cases_per_thou, data = global_total)

# Show summary of the model
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = global_total)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1358 -0.5496 -0.3903  0.4101  5.7130
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.5493125  0.0890224   6.171 2.41e-09 ***
## cases_per_thou 0.0028631  0.0003306   8.660 3.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.089 on 277 degrees of freedom
## Multiple R-squared:  0.2131, Adjusted R-squared:  0.2102
## F-statistic:    75 on 1 and 277 DF,  p-value: 3.959e-16
```

Utilize the linear model generated to forecast deaths per thousand based on cases per thousand, subsequently comparing the results with the actual values.

```r
# Predict data using model
global_total_w_pred <- global_total %>% mutate(pred = predict(mod))

# Plot the graph
global_total_w_pred %>% ggplot() +
  geom_point(aes(x = cases_per_thou, y = deaths_per_thou, color = "actual")) +
  geom_point(aes(x = cases_per_thou, y = pred, color = "predicted")) +
  theme(legend.position = "bottom")
```



The graph illustrates a correlation between deaths per thousand and cases per thousand, exhibiting a near-linear pattern.

Summary

The report presents an analysis of COVID-19 data for Canada, including trends in cumulative cases and fatalities for the country and its provinces. The analysis includes a calculation of new cases on a monthly basis, demonstrating a surge during the winter months and a decline during the summer months. Additionally, the report examines the global trend and predicts a linear model for deaths per thousand based on cases per thousand, finding a correlation between the two.

Bias Identification

Using a single data source in the report may limit its ability to capture a comprehensive picture of the COVID-19 situation and result in potential bias. The report also lacks mention of any limitations or sources of error in the data or analysis; and fails to thoroughly examine the geographic location and demographic variables, which may impact the pandemic's patterns and trends. Addressing these shortcomings is necessary for ensuring the report's validity and credibility.

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
```

```
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.utf8  LC_CTYPE=English_Canada.utf8
## [3] LC_MONETARY=English_Canada.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.0  timechange_0.1.1 forcats_0.5.2    stringr_1.5.0
##  [5] dplyr_1.0.10     purrr_1.0.0      readr_2.1.3      tidyr_1.2.1
##  [9] tibble_3.1.8     ggplot2_3.4.0    tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0    xfun_0.36           haven_2.5.1
##  [4] gargle_1.2.1        colorspace_2.0-3    vctrs_0.5.1
##  [7] generics_0.1.3      htmltools_0.5.4     yaml_2.3.6
## [10] utf8_1.2.2          rlang_1.0.6         pillar_1.8.1
## [13] withr_2.5.0         glue_1.6.2          DBI_1.1.3
## [16] dbplyr_2.2.1        modelr_0.1.10       readxl_1.4.1
## [19] lifecycle_1.0.3     munsell_0.5.0       gtable_0.3.1
## [22] cellranger_1.1.0    rvest_1.0.3         evaluate_0.19
## [25] labeling_0.4.2      knitr_1.41          tzdb_0.3.0
## [28] fastmap_1.1.0       fansi_1.0.3         highr_0.10
## [31] broom_1.0.2         scales_1.2.1        backports_1.4.1
## [34] googlesheets4_1.0.1 jsonlite_1.8.4      farver_2.1.1
## [37] fs_1.5.2            hms_1.1.2           digest_0.6.31
## [40] stringi_1.7.8       grid_4.2.2          cli_3.4.1
## [43] tools_4.2.2         magrittr_2.0.3      crayon_1.5.2
## [46] pkgconfig_2.0.3     ellipsis_0.3.2      xml2_1.3.3
## [49] reprex_2.0.2        googledrive_2.0.0   assertthat_0.2.1
## [52] rmarkdown_2.19      httr_1.4.4          rstudioapi_0.14
## [55] R6_2.5.1            compiler_4.2.2
```