DTSA5301_Covid19

HA

5/18/2024

Load libraries needed

Research Questions

1 - Is there relationship between number of cases (Covid19 infection) and number of Death? China, Sweden, and Kenya as example 2 - Did Covid19 affected under-developed countries more than developed countries?

```
# link to data source
url0 <- 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid
```

concatenate url0 with filenames

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
library(stringr)
urls <- str_c(url0, filenames)
```

read csv files from urls

```
global_cases <- read_csv(urls[1])
```

```
## 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[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_cases <- read_csv(urls[3])
```

```
## 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[4])
```

```
## 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.
```

Data Exploration Reshapes the `global_cases` dataset from wide to long format by pivoting all columns except 'Province/State', 'Country/Region', 'Lat', and 'Long', converting date columns into a single 'date' column and corresponding values into a 'cases' column, and then removes the 'Lat' and 'Long' columns.

```
global_cases <- global_cases %>%
  pivot_longer(cols = -c('Province/State', 'Country/Region', Lat, Long), names_to = "date", values_to =
  select(-c(Lat,Long))
```

The same thing for global_deaths

```
global_deaths <- global_deaths %>%
  pivot_longer(cols = -c('Province/State', 'Country/Region', Lat, Long), names_to = "date", values_to =
  select(-c(Lat,Long))
```

Merging the global_cases and global_deaths datasets, renames certain columns for consistency, and converts the 'date' column to a date format

```
global <- global_cases %>% full_join(global_deaths) %>%
  rename(Country_Region = 'Country/Region',
         Province_State = 'Province/State') %>%
  mutate(date = mdy(date))
```

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

Filtering positive cases

```
global <- global %>% filter(cases > 0)
```

continue

Create new column 'Combined_Key' by concatenating the 'Province_State' and 'Country_Region' columns with a comma and space separator, while keeping the original columns.

```
global <- global %>%
  unite("Combined_Key", c(Province_State, Country_Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)
```

Add look up url table loading the uid_lookup_url dataset, removes specified columns, and then merges it with the global dataset on the 'Province_State' and 'Country_Region' columns, removing some columns from the merged result and selecting specific columns for the final dataset

```
uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/

uid <- read_csv(uid_lookup_url) %>% select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
```

```
## 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.
```

```
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>% select(-c(UID, FIPS)) %>% select(Provi
#summary(global)
```

grouping the global dataset by 'Province_State', 'Country_Region', and 'date', sums the 'cases', 'deaths', and 'Population' for each group, calculates 'deaths_per_mill', selects specific columns, and then ungroups the dataset

```
global_by_country <- global %>% group_by(Province_State, Country_Region, date) %>% summarize(cases = su
```

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

Data Aggregation

Display the top 5 entries with the minimum deaths_per_thou Display the top 5 entries with the maximum deaths_per_thou

```
global_country_totals <- global_by_country %>%
  group_by(Country_Region) %>%
  summarize(deaths = max(deaths), cases = max(cases), population = max(Population), cases_per_thou = 10
```

```
global_country_totals %>% slice_min(deaths_per_thou, n = 5)
```

```
## # A tibble: 5 x 6
##   Country_Region deaths cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1 Holy See            0    29        809     35.8               0
## 2 Tuvalu              0  2828      11792    240.                0
## 3 Korea, North        6     1   25778815      0.0000388         0.000233
## 4 Burundi            38 53631   11890781      4.51              0.00320
## 5 Chad              194  7679   16425859      0.467             0.0118
```

```
global_country_totals %>% slice_max(deaths_per_thou, n = 5)
```

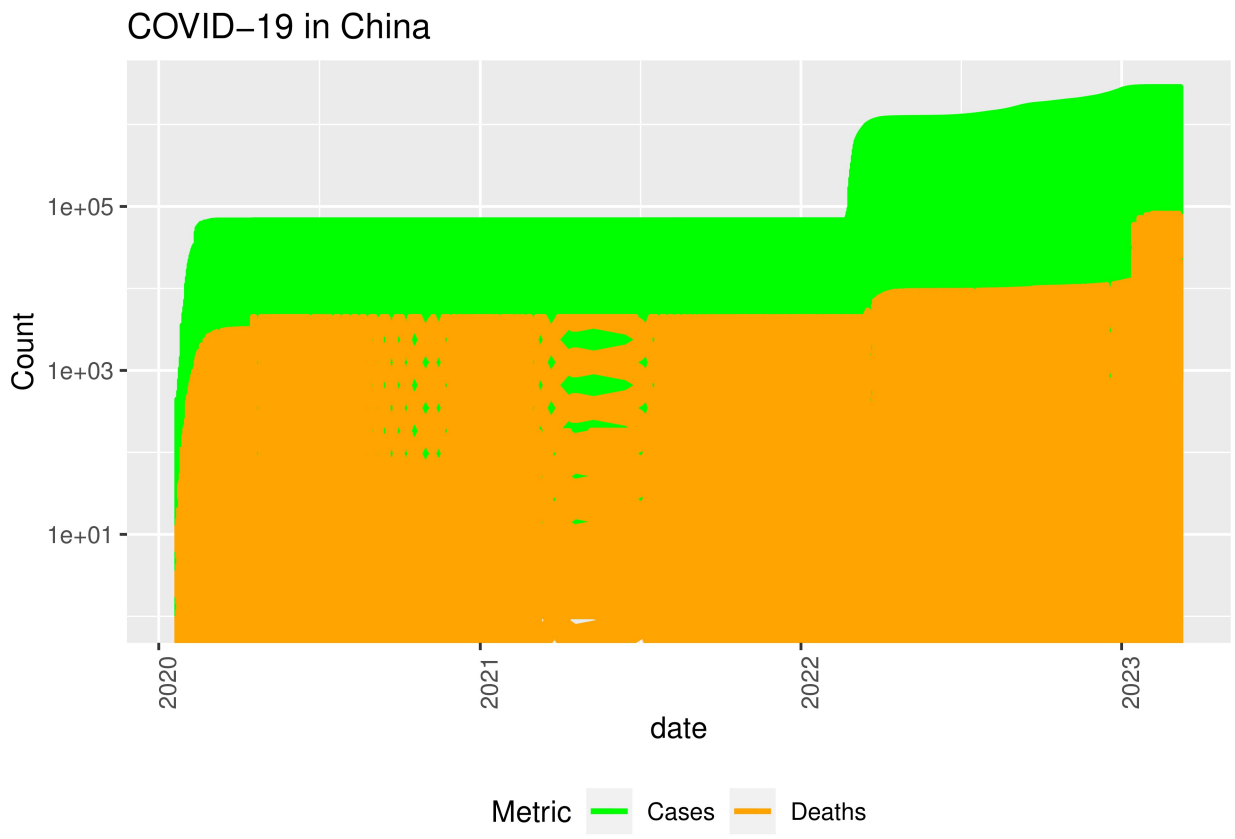
```
## # A tibble: 5 x 6
##   Country_Region         deaths  cases population cases_per_thou deaths_per_thou
##   <chr>                   <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
## 1 Peru                   219539 4.49e6   32971846           136.            6.66
## 2 Bulgaria                38228 1.30e6    6948445           187.            5.50
## 3 Hungary                 48762 2.20e6    9660350           227.            5.05
## 4 Bosnia and Herzegovina  16280 4.02e5    3280815           122.            4.96
## 5 North Macedonia          9662 3.47e5    2083380           166.            4.64
```

###Visualization

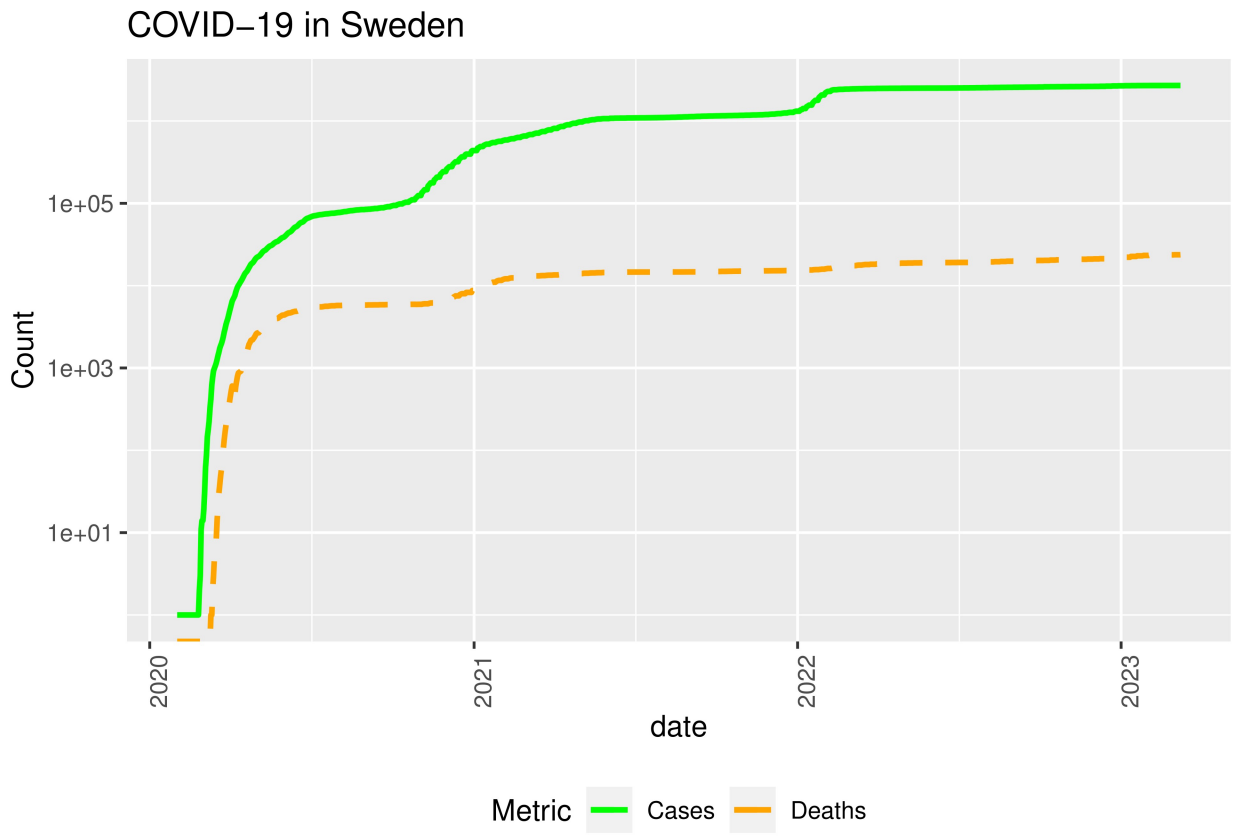
Create Plot for 'China','Sweden', and 'Kenya'

```
plot_covid19_data <- function(country, data) {
  data %>%
    filter(Country_Region == country) %>%
    filter(cases > 0) %>%
    ggplot(aes(x = date)) +
    geom_line(aes(y = cases, color = "Cases"), size = 1) +
    geom_line(aes(y = deaths, color = "Deaths"), linetype = "dashed", size = 1) +
    scale_y_log10() +
    theme(
      legend.position = "bottom",
      axis.text.x = element_text(angle = 90)
    ) +
    labs(
      title = str_c("COVID-19 in ", country),
      y = "Count",
      color = "Metric"
    ) +
```

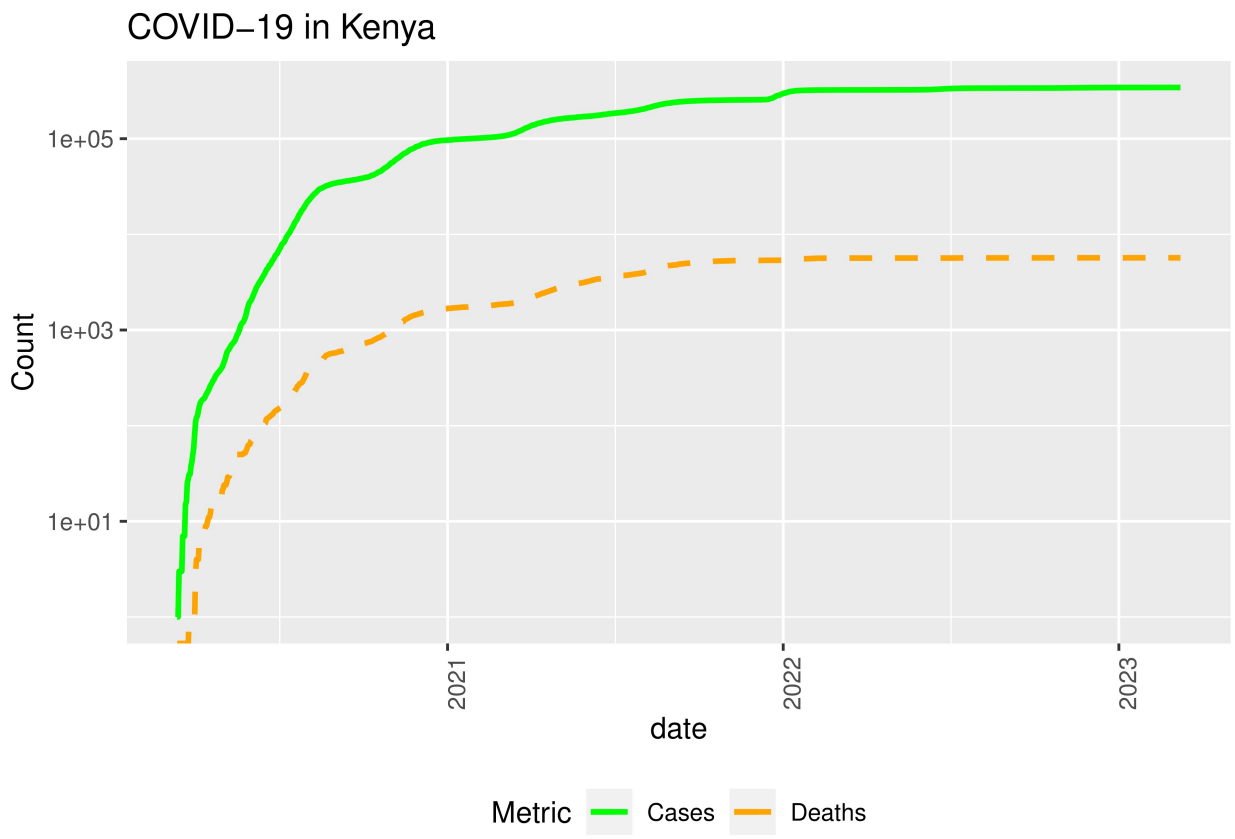
```
    scale_color_manual(
      values = c("Cases" = "Green", "Deaths" = "orange")
    )
}

# Plot for China
plot_covid19_data("China", global_by_country)
```




```
# Plot for Sweden
plot_covid19_data("Sweden", global_by_country)
```




```
# Plot for Kenya
plot_covid19_data("Kenya", global_by_country)
```



Build a Poisson Model

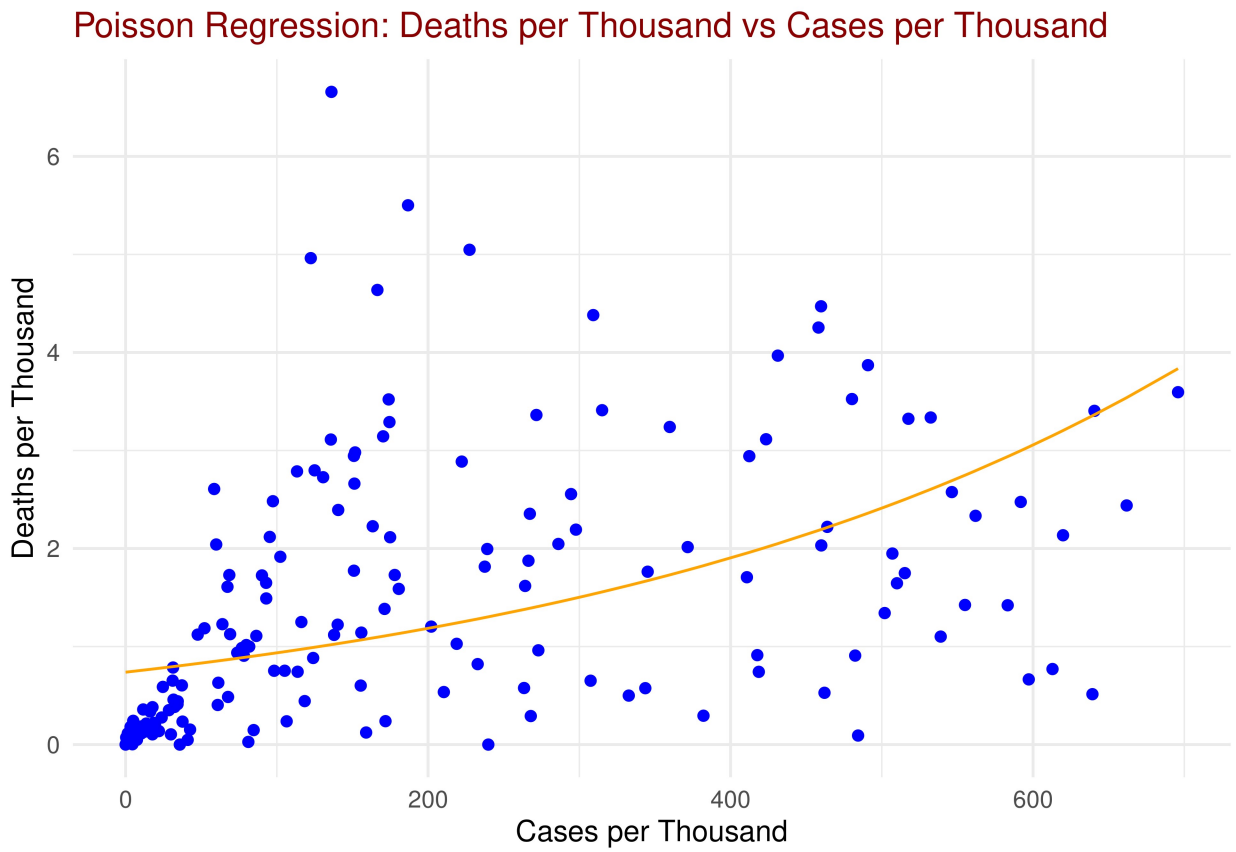
Fit Poisson model and predict values Plot the model

```
mod_poisson <- glm(deaths_per_thou ~ cases_per_thou, data = global_country_totals, family = poisson())

# Summary of the model
summary(mod_poisson)
```

```
##
## Call:
## glm(formula = deaths_per_thou ~ cases_per_thou, family = poisson(),
##     data = global_country_totals)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.3028759  0.1021868  -2.964  0.00304 **
## cases_per_thou  0.0023671  0.0002899   8.164 3.24e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 264.92  on 193  degrees of freedom
## Residual deviance: 203.35  on 192  degrees of freedom
```

```
## AIC: Inf
##
## Number of Fisher Scoring iterations: 5
```

```
global_country_totals <- global_country_totals %>%
  mutate(predicted_deaths_per_thou = predict(mod_poisson, type = "response"))


ggplot(global_country_totals, aes(x = cases_per_thou, y = deaths_per_thou)) +
  geom_point(color = "blue") +  # Change dots to blue
  geom_line(aes(y = predicted_deaths_per_thou), color = "orange") +
  labs(title = "Poisson Regression: Deaths per Thousand vs Cases per Thousand",
       x = "Cases per Thousand",
       y = "Deaths per Thousand") +
  theme_minimal() +
  theme(plot.title = element_text(color = "darkred"))  # Change title color
```



Interpretation For each additional case per thousand, the expected number of deaths per thousand to go up by approximately 0.24%. The very small p-value for the coefficient of cases_per_thou indicates a strong association between the number of cases and the number of deaths.

Conclusion: Strong relationship between cases per thousand and deaths per thousand.

1 - The Poisson regression model shows a statistically significant relationship between the number of cases per thousand (cases_per_thou) and the number of deaths per thousand (deaths_per_thou). 2 - There is no evidence that Covid19 affected under-developed countries more than developed countries.

Bias

1 - We cannot compare China to any other country because China was the epicenter. 2 - Sweden is a small wealthy country 3 - Kenya is a poor country with very weak healthcare system. Kenya did not have the resources to assess its population. 4 - Too many missing values make data skewed.