covid-19

Jamil Garro

4/23/2024

Datasets

The datasets come from the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE). 4 datasets are considerded:

confirmed us cases confirmed global cases confirmed us deaths confirmed global deaths

The datasets registered the official numbers of cases of COVID-19 and the deaths from it in USA and the world from the beginning of the pandemic by country and by county.

objectives

Determine worst states in the USA by number of cases Determine months that have largest occurence of new cases in USA Determine 15 countries that has worst numbers by death cases A model to determine the relationship between new cases and new deaths in Lebanon

Pre-processing

```
knitr::opts_chunk$set(echo = TRUE)
```

Packages

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## 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(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'purrr' was built under R version 4.1.3

## Warning: package 'forcats' was built under R version 4.1.3

## Warning: package 'lubridate' was built under R version 4.1.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## v readr     2.1.4

## -- 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)
```

Data

From https://raw.githubusercontent.com/CSSEGISandData/COVID-19 https://www2.census.gov/programs-surveys/popest/datasets/2010-2020/state/totals/

import and read:

```
url_base <- c('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_
us_cases <- read_csv(paste(url_base,'time_series_covid19_confirmed_US.csv',sep = ""))
```

```
## 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(paste(url_base,'time_series_covid19_deaths_US.csv',sep = ""))
```

```
## 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_cases <- read_csv(paste(url_base,'time_series_covid19_confirmed_global.csv',sep = ""))
```

```
## 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(paste(url_base,'time_series_covid19_deaths_global.csv',sep = ""))
```

```
## 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_pop <- read.csv("https://www2.census.gov/programs-surveys/popest/datasets/2010-2020/state/totals/nst-
uid <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UI
```

```
## 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.
```

Data tidying

Exploring

```
head(us_cases, n=5)
```

```
## # A tibble: 5 x 1,154
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <dbl> <chr> <chr> <dbl> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 2 84001003 US    USA     840  1003 Baldwin Alabama        US              30.7
## 3 84001005 US    USA     840  1005 Barbour Alabama        US                31.9
## 4 84001007 US    USA     840  1007 Bibb    Alabama        US                33.0
## 5 84001009 US    USA     840  1009 Blount  Alabama        US                34.0
## # i 1,145 more variables: Long_ <dbl>, Combined_Key <chr>, ‘1/22/20‘ <dbl>,
## #   ‘1/23/20‘ <dbl>, ‘1/24/20‘ <dbl>, ‘1/25/20‘ <dbl>, ‘1/26/20‘ <dbl>,
## #   ‘1/27/20‘ <dbl>, ‘1/28/20‘ <dbl>, ‘1/29/20‘ <dbl>, ‘1/30/20‘ <dbl>,
## #   ‘1/31/20‘ <dbl>, ‘2/1/20‘ <dbl>, ‘2/2/20‘ <dbl>, ‘2/3/20‘ <dbl>,
## #   ‘2/4/20‘ <dbl>, ‘2/5/20‘ <dbl>, ‘2/6/20‘ <dbl>, ‘2/7/20‘ <dbl>,
## #   ‘2/8/20‘ <dbl>, ‘2/9/20‘ <dbl>, ‘2/10/20‘ <dbl>, ‘2/11/20‘ <dbl>,
## #   ‘2/12/20‘ <dbl>, ‘2/13/20‘ <dbl>, ‘2/14/20‘ <dbl>, ‘2/15/20‘ <dbl>, ...
```

```
head(global_cases, n=5)
```

```
## # A tibble: 5 x 1,147
##   ‘Province/State‘ ‘Country/Region‘   Lat  Long ‘1/22/20‘ ‘1/23/20‘ ‘1/24/20‘
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## # i 1,140 more variables: ‘1/25/20‘ <dbl>, ‘1/26/20‘ <dbl>, ‘1/27/20‘ <dbl>,
## #   ‘1/28/20‘ <dbl>, ‘1/29/20‘ <dbl>, ‘1/30/20‘ <dbl>, ‘1/31/20‘ <dbl>,
## #   ‘2/1/20‘ <dbl>, ‘2/2/20‘ <dbl>, ‘2/3/20‘ <dbl>, ‘2/4/20‘ <dbl>,
## #   ‘2/5/20‘ <dbl>, ‘2/6/20‘ <dbl>, ‘2/7/20‘ <dbl>, ‘2/8/20‘ <dbl>,
## #   ‘2/9/20‘ <dbl>, ‘2/10/20‘ <dbl>, ‘2/11/20‘ <dbl>, ‘2/12/20‘ <dbl>,
## #   ‘2/13/20‘ <dbl>, ‘2/14/20‘ <dbl>, ‘2/15/20‘ <dbl>, ‘2/16/20‘ <dbl>,
## #   ‘2/17/20‘ <dbl>, ‘2/18/20‘ <dbl>, ‘2/19/20‘ <dbl>, ‘2/20/20‘ <dbl>, ...
```

```
head(us_deaths, n=5)
```

```
## # A tibble: 5 x 1,155
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <dbl> <chr> <chr> <dbl> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US                32.5
## 2 84001003 US    USA     840  1003 Baldwin Alabama        US                30.7
## 3 84001005 US    USA     840  1005 Barbour Alabama        US                31.9
## 4 84001007 US    USA     840  1007 Bibb    Alabama        US                33.0
## 5 84001009 US    USA     840  1009 Blount  Alabama        US                34.0
## # i 1,146 more variables: Long_ <dbl>, Combined_Key <chr>, Population <dbl>,
## #   ‘1/22/20‘ <dbl>, ‘1/23/20‘ <dbl>, ‘1/24/20‘ <dbl>, ‘1/25/20‘ <dbl>,
## #   ‘1/26/20‘ <dbl>, ‘1/27/20‘ <dbl>, ‘1/28/20‘ <dbl>, ‘1/29/20‘ <dbl>,
## #   ‘1/30/20‘ <dbl>, ‘1/31/20‘ <dbl>, ‘2/1/20‘ <dbl>, ‘2/2/20‘ <dbl>,
## #   ‘2/3/20‘ <dbl>, ‘2/4/20‘ <dbl>, ‘2/5/20‘ <dbl>, ‘2/6/20‘ <dbl>,
## #   ‘2/7/20‘ <dbl>, ‘2/8/20‘ <dbl>, ‘2/9/20‘ <dbl>, ‘2/10/20‘ <dbl>,
## #   ‘2/11/20‘ <dbl>, ‘2/12/20‘ <dbl>, ‘2/13/20‘ <dbl>, ‘2/14/20‘ <dbl>, ...
```

```
head(global_deaths, n=5)
```

```
## # A tibble: 5 x 1,147
##   ‘Province/State‘ ‘Country/Region‘   Lat  Long ‘1/22/20‘ ‘1/23/20‘ ‘1/24/20‘
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## # i 1,140 more variables: ‘1/25/20‘ <dbl>, ‘1/26/20‘ <dbl>, ‘1/27/20‘ <dbl>,
## #   ‘1/28/20‘ <dbl>, ‘1/29/20‘ <dbl>, ‘1/30/20‘ <dbl>, ‘1/31/20‘ <dbl>,
## #   ‘2/1/20‘ <dbl>, ‘2/2/20‘ <dbl>, ‘2/3/20‘ <dbl>, ‘2/4/20‘ <dbl>,
## #   ‘2/5/20‘ <dbl>, ‘2/6/20‘ <dbl>, ‘2/7/20‘ <dbl>, ‘2/8/20‘ <dbl>,
## #   ‘2/9/20‘ <dbl>, ‘2/10/20‘ <dbl>, ‘2/11/20‘ <dbl>, ‘2/12/20‘ <dbl>,
## #   ‘2/13/20‘ <dbl>, ‘2/14/20‘ <dbl>, ‘2/15/20‘ <dbl>, ‘2/16/20‘ <dbl>,
## #   ‘2/17/20‘ <dbl>, ‘2/18/20‘ <dbl>, ‘2/19/20‘ <dbl>, ‘2/20/20‘ <dbl>, ...
```

tidying

objectives are to convert date columns into date row, formating the date and deleting unwanted columns from the 4 datasets.

```
us_cases <- us_cases %>%
  pivot_longer(-c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Province_State", "Country_Region",
                  "Lat", "Long_", "Combined_Key"),
               names_to = "Date", values_to = "Confirmed") %>%
  mutate(Date = as.Date(Date, format ="%m/%d/%y")) %>%
  select (Admin2:Confirmed) %>% select(-c(Lat, Long_, Combined_Key))
```

```
us_deaths <- us_deaths %>%
  pivot_longer(-c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Province_State", "Country_Region",
                  "Lat", "Long_", "Combined_Key"),
               names_to = "Date", values_to = "Deaths") %>%
  mutate(Date = as.Date(Date, format ="%m/%d/%y")) %>%
  select (Admin2:Deaths) %>% select(-c(Lat, Long_, Combined_Key))
```

```
global_cases <- global_cases %>%
  pivot_longer(-c("Province/State", "Lat",
                  "Long","Country/Region"),
               names_to = "Date", values_to = "Confirmed") %>%   select("Country/Region", "Date","Confi
```

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

Data analysis

States in USA with worst cases per 100,000 population:

In order to do that, us_cases and us_deaths will be join into the same dataset, will be added to them population by state, maximum cases by state will be determine, cases by 100k population.

```
us_covid <- us_cases %>%
  left_join(us_deaths, by = c("Admin2", "Province_State", "Country_Region","Date"))
us_pop_tidy <- us_pop %>%
  select(c(NAME, POPESTIMATE2020)) %>%
  rename(Province_State = NAME, Population = POPESTIMATE2020)
us_covid_pop <- us_covid %>%
  left_join(us_pop_tidy, by = c("Province_State"))
cases_long_max <- us_cases %>%
  group_by(Province_State) %>%
  summarise(max_cases = max(Confirmed))
us_covid_max <- cases_long_max %>%
  left_join(us_pop_tidy, by = c("Province_State"))
cases_long_max <- us_cases %>%
  group_by(Province_State) %>%
  summarise(max_cases = max(Confirmed))
us_covid_max_per100k <- us_covid_max %>%
  mutate(cases_per_100k = max_cases/Population * 100000) %>% arrange(desc(cases_per_100k)) %>%
  slice_max(cases_per_100k, n=25)
```

```
us_covid_max_per100k
```

```
## # A tibble: 25 x 4
##    Province_State       max_cases Population cases_per_100k
##    <chr>                    <dbl>      <int>          <dbl>
##  1 Rhode Island            278748    1057125         26368.
##  2 District of Columbia    177945     712816         24964.
##  3 Nevada                  671243    3138259         21389.
##  4 Arizona                1530296    7421401         20620.
##  5 Delaware                186969     986809         18947.
##  6 Hawaii                  264197    1407006         18777.
##  7 Alaska                  124334     731158         17005.
##  8 Utah                    410508    3249879         12631.
##  9 Illinois               1533935   12587530         12186.
## 10 California             3710586   39368078          9425.
## # i 15 more rows
```

months which had worst Covid-19 cases

US cases data set will be used as base and data will be aggregated based on month of the year

```
us_covid_pop_new_cases <- us_covid_pop  %>%
  mutate(new_cases = Confirmed - lag(Confirmed)) %>%
  mutate(Date = as.Date(Date,"%m/%d/%y"))
us_covid_pop_new_cases_worst <- us_covid_pop_new_cases %>%
  mutate(yearmon = floor_date(Date, unit = "month")) %>%
  group_by(yearmon) %>%
  summarise(sum = sum(new_cases, default=0)) %>%
  filter(sum > 0)
```

countries with worst deaths per 100K population

Both global cases and deaths will be combined to get a complete dataset, will also be joined by global population dataset, will be aggregated at country level and maximum number of deaths per 100K population.

```
global_covid <- global_cases %>% left_join(global_deaths, by = c("Country/Region","Date"))
```

```
## Warning in left_join(., global_deaths, by = c("Country/Region", "Date")):
## Detected an unexpected many-to-many relationship between ‘x‘ and ‘y‘.
```

```
global_population_tidy <- uid %>%
  filter(is.na(Province_State)) %>%
  select("Country_Region","Population") %>%
  rename("Country/Region" = Country_Region)
global_covid_pop <- global_covid %>%
  left_join(global_population_tidy, by =c("Country/Region"))
global_covid_pop_maxdeaths <- global_covid_pop %>%
  rename(Country=  "Country/Region") %>%
  group_by(Country) %>%
  summarise(max_deaths = max(Deaths)) %>%
  rename(`Country/Region`= Country) %>%
  left_join(global_population_tidy, by = "Country/Region") %>%
  mutate(deaths_per_100k = max_deaths/Population * 100000) %>%
  slice_max(deaths_per_100k, n=25)
```

```
global_covid_pop_maxdeaths
```

```
## # A tibble: 25 x 4
##    ‘Country/Region‘       max_deaths Population deaths_per_100k
##    <chr>                       <dbl>      <dbl>           <dbl>
##  1 Peru                       219539   32971846            666.
##  2 Bulgaria                    38228    6948445            550.
##  3 Hungary                     48762    9660350            505.
##  4 Bosnia and Herzegovina      16280    3280815            496.
##  5 North Macedonia              9662    2083380            464.
##  6 Montenegro                   2808     628062            447.
##  7 Croatia                     17987    4105268            438.
##  8 Georgia                     16971    3989175            425.
##  9 Czechia                     42491   10708982            397.
## 10 Slovakia                    21035    5434712            387.
## # i 15 more rows
```

relationship between Lebanon’s new death rate and new cases

New cases and deaths for Lebanon will be determined on daily basis from global dataset

```
lebanon_cases <- global_covid_pop %>%
  filter(str_detect(`Country/Region`, 'Lebanon'))
lebanon_new_cases_deaths <- lebanon_cases %>%
  mutate(new_cases = Confirmed - lag(Confirmed)) %>%
  mutate(new_deaths = Deaths - lag(Deaths)) %>% drop_na(new_cases) %>% drop_na(new_deaths) %>%
  mutate(Date = as.Date(Date,"%m/%d/%y"))
lebanon_model <- lm(new_deaths ~ new_cases, data= lebanon_new_cases_deaths)
lebanon_new_cases_deaths_pred <- lebanon_new_cases_deaths %>% mutate(pred = predict(lebanon_model))
```

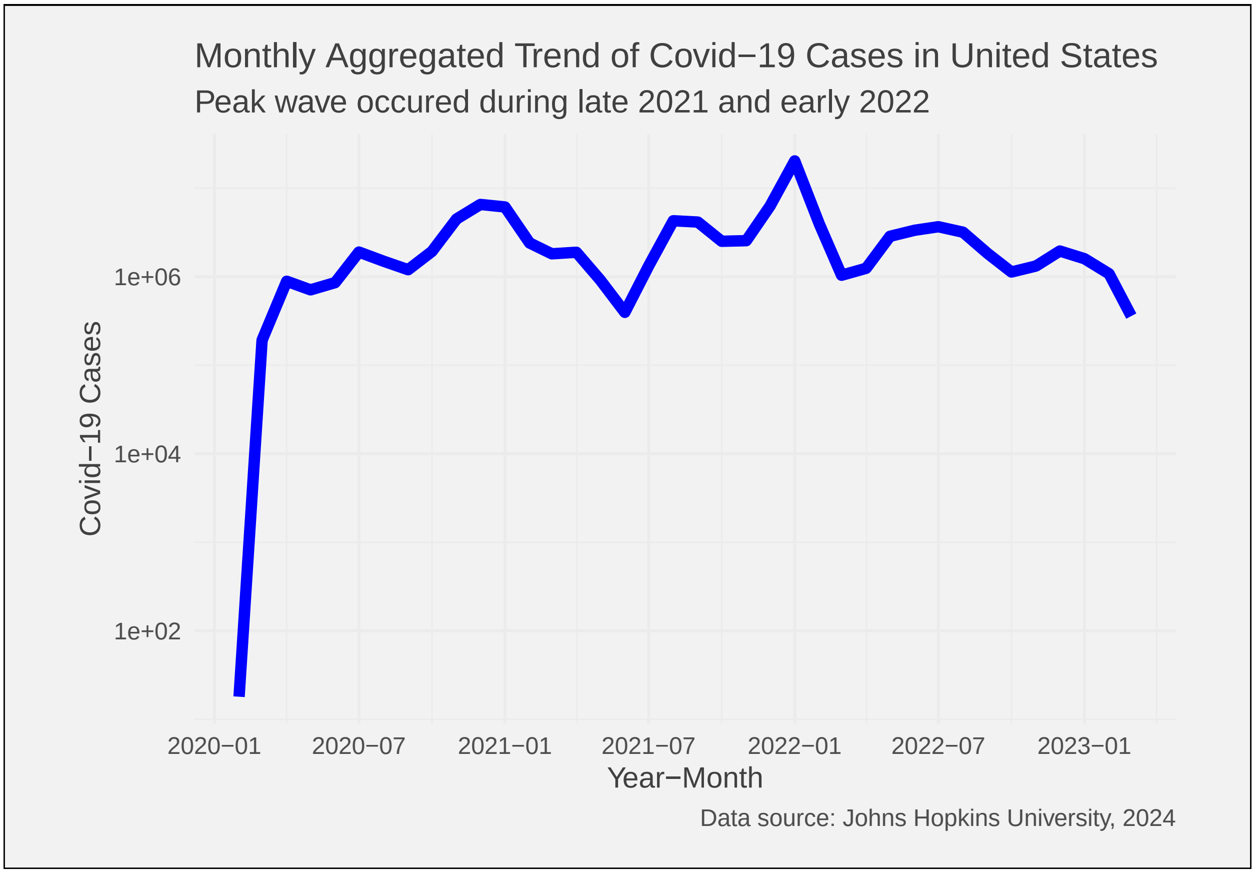
Data visualization

the theme

```
theme_shooting <- function() {
    theme_minimal() +
theme(
    text = element_text(color = "gray25"),
    plot.subtitle = element_text(size = 12),
    plot.caption = element_text(color = "gray30"),
    plot.background = element_rect(fill = "gray95"),
    plot.margin = unit(c(5, 10, 5, 10), units = "mm")
  )
}
```

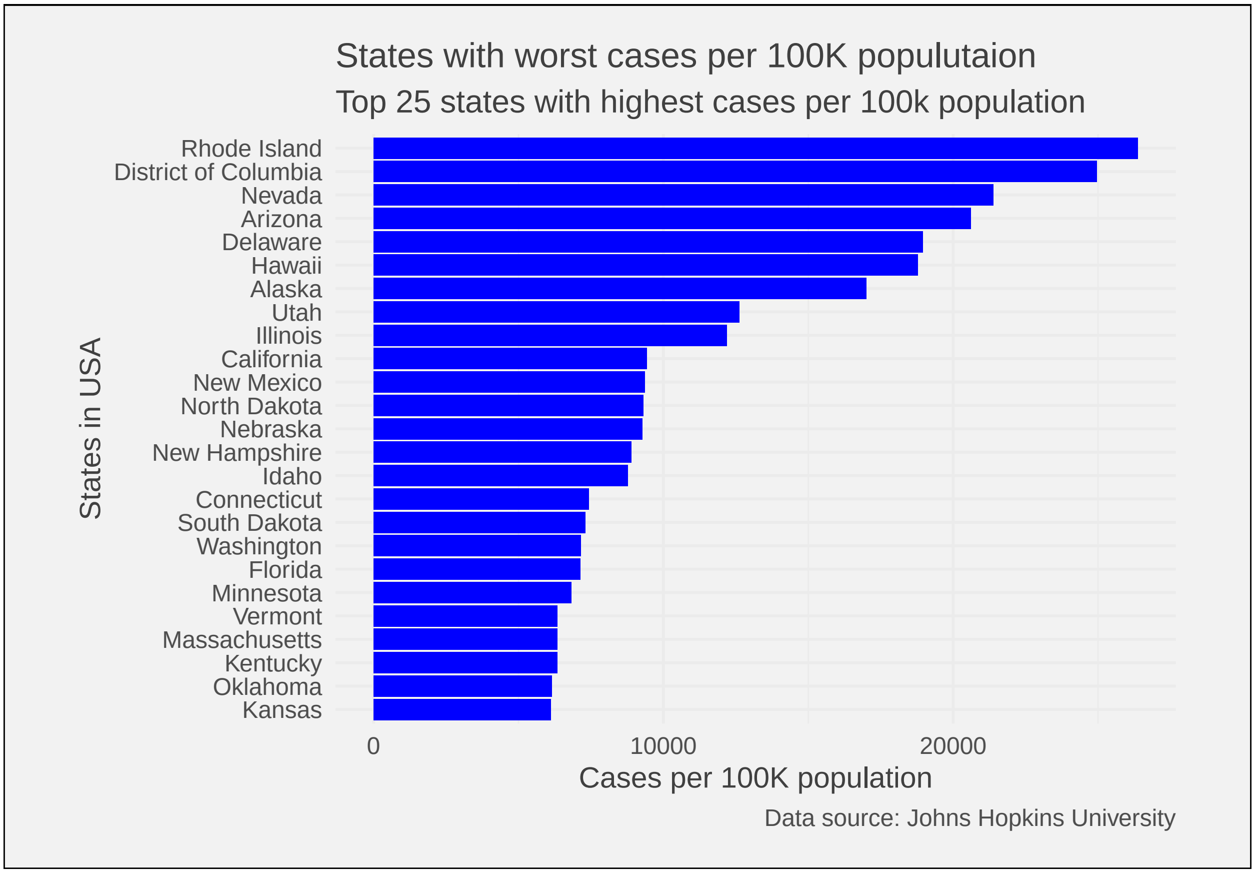
Year-Month in USA with worst cases per 100,000 population

```
ggplot(data = us_covid_pop_new_cases_worst) +
  geom_line(aes(x = yearmon, y = sum), size = 2, color = "Blue") +
  scale_y_log10() +
  scale_x_date(date_labels = "%Y-%m",
               breaks = seq(from = as.Date("2020-01-01"),
                            to = as.Date("2023-02-01"), by = "6 months")) +
  labs(
    x = "Year-Month",
    y = "Covid-19 Cases",
    title = "Monthly Aggregated Trend of Covid-19 Cases in United States",
    subtitle = "Peak wave occured during late 2021 and early 2022",
    caption = "Data source: Johns Hopkins University, 2024"
  ) + theme_shooting()
```



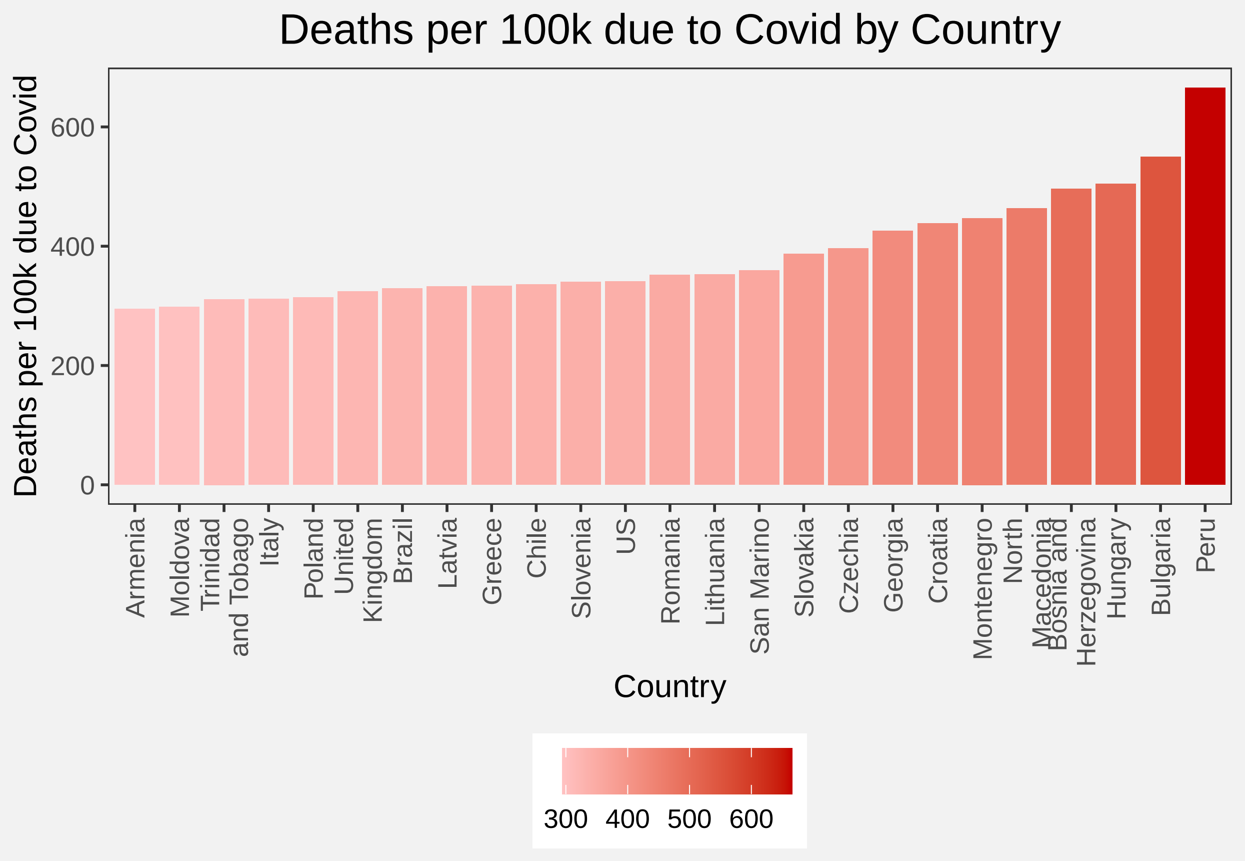
States with worst cases per 100K population

```r
ggplot(data = us_covid_max_per100k,
       aes(x = reorder(Province_State, cases_per_100k),
           y = cases_per_100k)) +
  geom_bar(stat = "identity", fill = "blue") + labs(
    x = "States in USA",
    y = "Cases per 100K population",
    title = "States with worst cases per 100K populutaion",
    subtitle = "Top 25 states with highest cases per 100k population",
    caption = "Data source: Johns Hopkins University"
  ) + coord_flip() + theme_shooting()
```



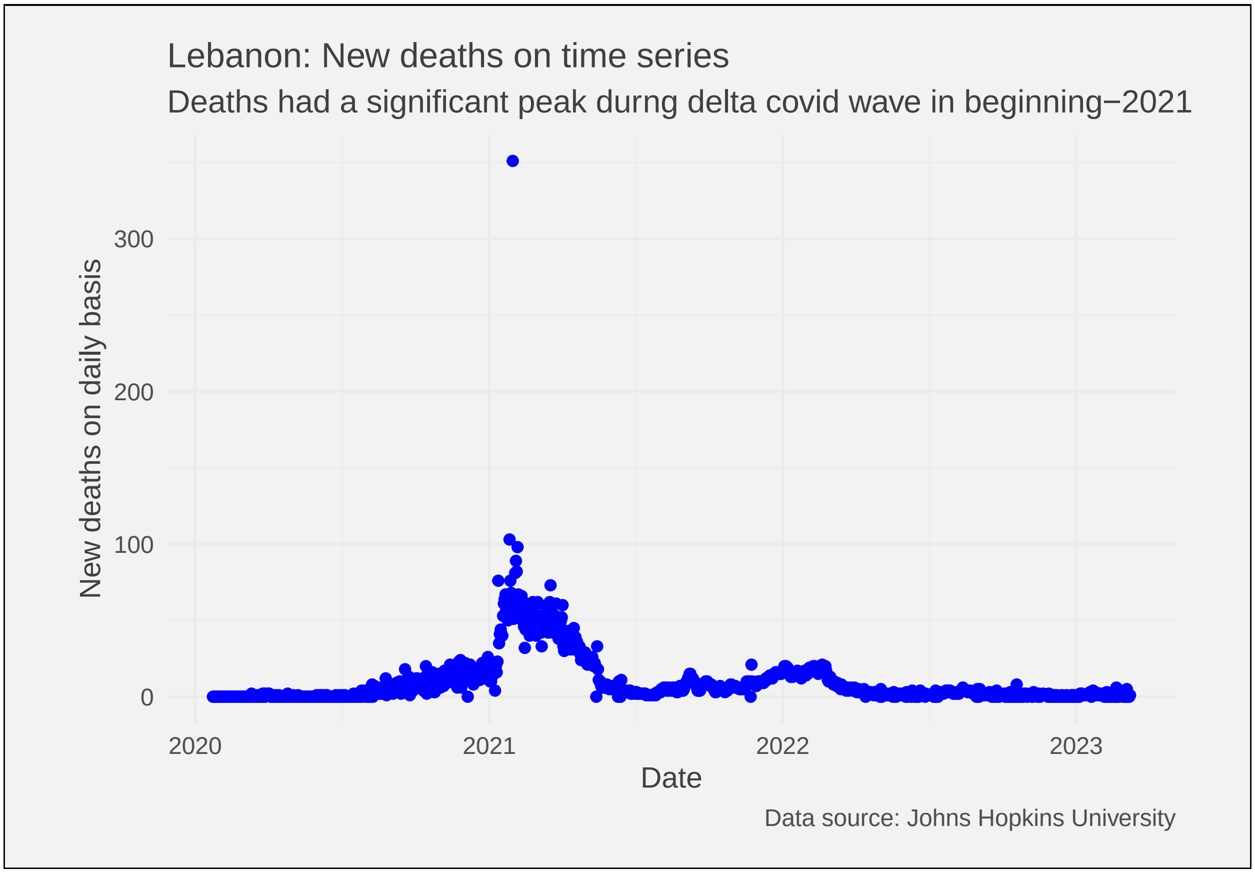
Countries with worst s per 100K population

```
library(stringr)
ggplot(global_covid_pop_maxdeaths, aes(x = reorder(`Country/Region`, deaths_per_100k) ,
                                       y = deaths_per_100k, fill = deaths_per_100k)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Country", y = "Deaths per 100k due to Covid",
       title = "Deaths per 100k due to Covid by Country") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5, size = 16),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text.x = element_text(size = 10, angle = 90, hjust = 1, vjust = 0.5),
        axis.text.y = element_text(size = 10),
        legend.title = element_blank(),
        legend.position = "bottom",
        legend.text = element_text(size = 10),
        panel.background = element_rect(fill = "grey95"),
        panel.grid = element_blank(),
        plot.background = element_rect(fill = "gray95")) +
  scale_fill_gradient(low = "#FFC2C2", high = "#C40000") +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10))
```



Lebanon's new deaths on time series

```
ggplot(data = lebanon_new_cases_deaths ) +
  geom_point(aes(x = Date, y = new_deaths), color="Blue") + labs(
    x = "Date",
    y = "New deaths on daily basis",
    title = "Lebanon: New deaths on time series",
    subtitle = "Deaths had a significant peak durng delta covid wave in beginning-2021",
    caption = "Data source: Johns Hopkins University"
  )  + theme_shooting()
```



Analyzing the relationship between new deaths and new cases in Lebanon

```
lebanon_model <- lm(new_deaths ~ new_cases, data= lebanon_new_cases_deaths)
```

```
lebanon_model
```

```
##
## Call:
## lm(formula = new_deaths ~ new_cases, data = lebanon_new_cases_deaths)
##
## Coefficients:
## (Intercept)    new_cases
##    3.656267     0.005407
```

```
summary(lebanon_model)
```

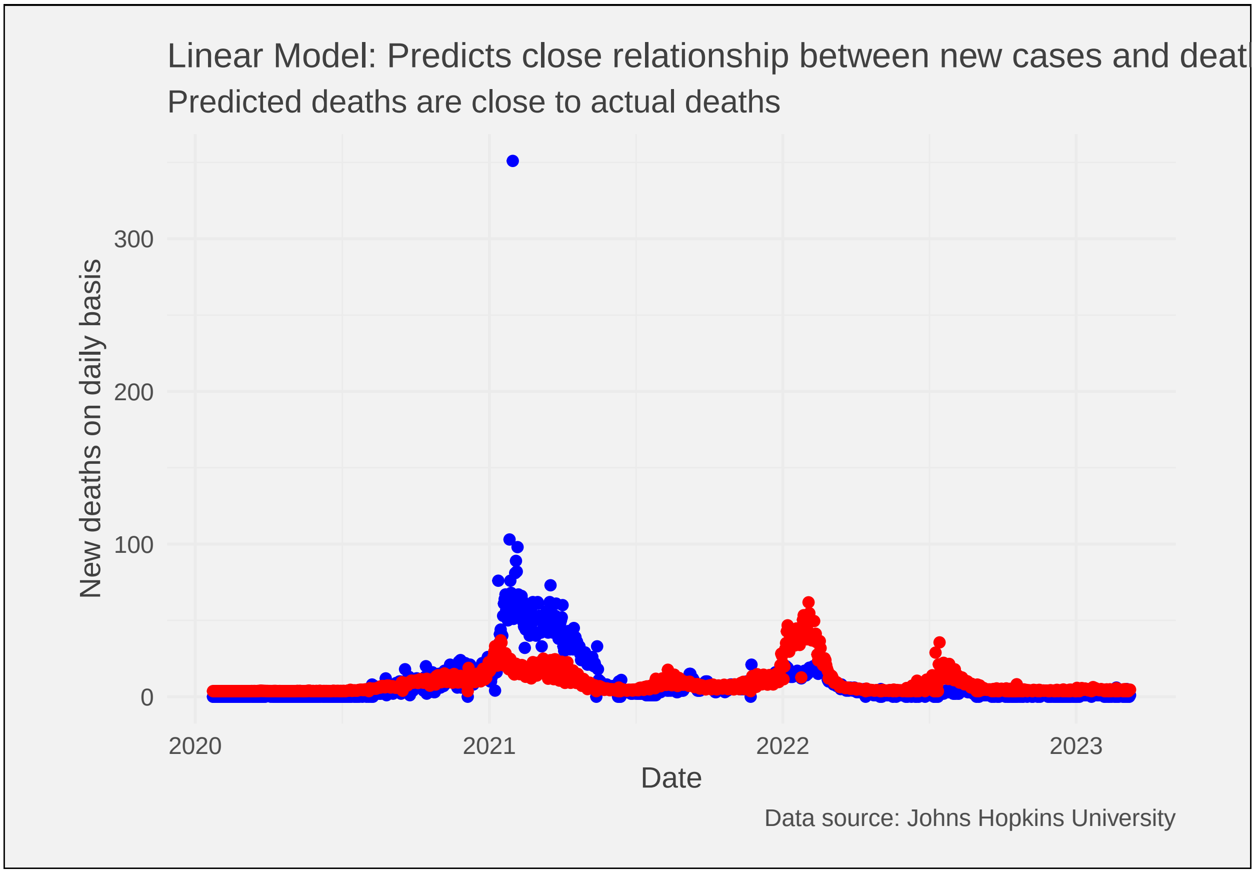
```
##
## Call:
## lm(formula = new_deaths ~ new_cases, data = lebanon_new_cases_deaths)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -44.83  -3.76  -3.05  -0.28 333.12
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.6562665  0.5652215   6.469 1.46e-10 ***
## new_cases   0.0054067  0.0002956  18.289  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 15.76 on 1140 degrees of freedom
## Multiple R-squared:  0.2268, Adjusted R-squared:  0.2262
## F-statistic: 334.5 on 1 and 1140 DF,  p-value: < 2.2e-16
```

The p-value of the model is significantly less than .05 indicating that model should be able to accurately predict the count of new deaths based on new cases. Based on above, there is a significant relationship between new deaths and new cases in Lebanon.

```
lebanon_new_cases_deaths_pred <- lebanon_new_cases_deaths %>% mutate(pred = predict(lebanon_model))
```

Plotting the model performance

```
ggplot(data = lebanon_new_cases_deaths_pred ) +
  geom_point(aes(x = Date, y = new_deaths),
             color="Blue") +
  geom_point(aes(x = Date, y = pred), color="Red") +
  labs(
    x = "Date",
    y = "New deaths on daily basis",
    title = "Linear Model: Predicts close relationship between new cases and deaths",
    subtitle = "Predicted deaths are close to actual deaths",
    caption = "Data source: Johns Hopkins University"
  )  + theme_shooting()
```



Data bias

Data could be biased at the level of collection. Data is collected from covid-19 centers, not all occurrences may be registered and registration could differ from country to country. So under or over reporting may occur depending on the process of registration. Inconsistencies and errors may occur.

Conclusion

The model that we've tested show that a high correlation coefficient between new deaths and new cases in Lebanon exists suggesting that the two variables are closely related. The two variables shows similar patterns over time.

USA witnessed two peaks in COVID cases between December 2020 to January 2021 and December 2021 to January 2022 which suggests that the pandemic was at its peak during that time.

These conclusions goes with a lot of patterns related to Covid-19 such as increase in lock-down policies by the states and shortages of medical supplies.