CU-DTSA-530 - COVID 19 Analysis Report

Rajesh Kutti, Student - University of Colorado, Boulder

2024-02-24

Important Note

- **Libraries Required**: Please check session info in the end and confirm if you installed the libraries needed for this project. Important libraries: **(tidyverse, lubridate, usmap, stringr)**.
- **File Downloads**: If you happen to run into data file download issues due to network issues, please retry, if not please manually download the file and save it to folder: cached_data under your current directory.
- **PDF Document**: A PDF version of this report is available in GitHub.

Executive Summary

- **COVID-19** pandemic challenged the world and changed the lives of many in many sad ways.
- Using the data from CSSE Johns Hopkins University, we will try to gain some insights into the Pandemic. We will follow a step by step and iterative Data Science process to analyze and generate insights.
- **Goal** - Analyze and build a Quantitative model to predict COVID-19 cases in a state or region or country or county.
- **Benefits** - In real life scenario these reports will help the WHO, Governments, Health care professionals and people in general in many ways. For e.g. if cases are rising on weekly basis without signs of slowing down, then there is some underlying problem that Government has too fix immediately - shortage of hospitals, equipment's, Doctors, Nurses, health care services, etc.
- **Objectives** - To achieve our goal, we have to continue to ask questions(as follows) to gain more insights into the problem until all our objectives are met.
  - How was the COVID-19 pandemic managed by countries across the world?
    - Which were the top 10 countries in terms of the cases or deaths?
    - When did the pandemic arrive? How was it managed daily?
    - Can we look at weekly rise in cases and deaths per 100k population and infer valuable info?
  - How was the COVID-19 pandemic managed by all the states in USA?
    - We will ask similar questions that was asked for other Countries.
  - How was the COVID-19 pandemic managed by our favorite city in the world?
  - Were there any Biases?
  - Did we verify our assumptions and Biases through data analysis and Visualizations?
- Finally, Build and test a predictive model to confirm our understanding.
- For those who want a summarized view of this report, please start from **Section 4: Visualizations**. The observations and insights are articulated in that section.

Data Science Process - An iterative approach. Reference:*DTSA-530 by Prof. Jane Wall)*

- We will follow a well-defined process that we learned in DTSA-5301 for doing our analysis. During this process we will document our observations and share insights that will help the team and the key stakeholders.
- **Import data** from CSSE John Hopkins
- **Tidy data** by removing unwanted rows, then formatting the data so that each row is an observation and finally null imputation or null value replacements.
- **Transform data** - Here we change numbers to categorical variables, rename columns or we add new columns as needed.
- **Analyze and Visualize** - We then go through iterative process of analyzing, visualizing, asking questions then going back to prior steps and repeating the process until we have good understanding of the answers to the questions we started with, (until goal is met).
- For analyzing, we will do Univariate analysis by looking at distributions for single variable, Bivariate analysis by looking at correlations between two variables and Multivariate analysis by checking relationship between predictors and our target variables.
- **Model** - We will build a quantitative model for our data and observations. During this phase we will try to check if our assumptions are accurate by running the predictions. For e.g. We may want to predict the number of cases or number of deaths next month in a state or country. Such insights and predictions will help the WHO and health organizations in a Country/region/state to plan and control the pandemic. To build a good model, we want the right predictor variables. We must choose different models, run different experiments to make our models perform better.
- **Communicate** - Finally, we will communicate our results in a reproducible way so that the report can be improved in future or can be leveraged by others to do more analysis. In this report, we will choose a specific date - (2021-12-31) - and do a thorough analysis to gain insights. Eventually, the date can be passed as a parameter to the document to generate insights for a given day/window.

References:

- University of Colorado, Boulder - DTSA-530
- CSSE Johns Hopkins University
- Coursera, CU Boulder

Data Science Process Details:

- This document is split into multiple sections for clarity, readability and re-usability.
- Analyze the Covid 19 Cases and Deaths by importing the data from John Hopkins website:
  - https://www.coursera.org/learn/data-science-as-a-field/supplement/cXrpr/project-files
  - https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_series
  - The data will be cached locally for repeated testing and fast performance.
- There are 4 Vs of the data **Volume, Velocity, Variety, Veracity**, hence the first step is to Tidy the data.
- 70% of the time was spent on Tidying the data, which included:
  - Handle missing data, null values, etc.
  - Replace null cases, deaths with 0, Exclude null countries.
  - Create categorical variables for country/region, state, etc.
- We had 4 datasets, two were global and two were US and one FIPS that contained the population.

  - We will join the US cases and deaths to form one dataset containing cases and deaths.
  - We will join the Global cases and deaths to form one dataset containing cases and deaths.
  - We will join the US and Global datasets with FIPS dataset to get the population for the country/region
  - Then we will spend a good amount of time analyzing and improving the quality of data.

- To compare the cases between two countries we need to compare proportions of cases or deaths and not the actual cases, hence we will add new columns.
  - cases_per_100k : For cases, the formula was (cases/population) * 100000
  - deaths_per_100k : For deaths, the formula was (deaths/population) * 100000

- For detailed analysis, we also need to understand the daily and weekly increase in cases and deaths. The following new columns will help:
  - **delta_cases_7**, delta_cases_7 : Average weekly increase in cases per 100k population.
  - **delta_deaths_7**, delta_deaths_7: Average weekly increase in deaths per 100k population.

Section 1A: Common Functions:

- In this section we will add all common functions used.
- Each function is well-documented.

```r
# ohn Hopkins URL to download data files:
GITHUB_URL_JH <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/"
CACHED_DATA_LOCATION <- "cached_data/csse_covid_19_data/"

# Analyze cases and deaths as on date
ANA_CAD_AS_ON_DATE <- "2021-12-31"
# Analyze Favourite State Name
ANA_FAV_STATE <- "New York"
ANA_FAV_STATE_ABR <- "NY"

# Define a filename list to download
data_files <- list(global_cases = "csse_covid_19_time_series/time_series_covid19_confirmed_global.csv",
  global_deaths = "csse_covid_19_time_series/time_series_covid19_deaths_global.csv",
  us_cases = "csse_covid_19_time_series/time_series_covid19_confirmed_US.csv",
  us_deaths = "csse_covid_19_time_series/time_series_covid19_deaths_US.csv",
  fips = "UID_ISO_FIPS_LookUp_Table.csv")

# We want to cache the data locally to work when internet connection is not available.
# Create local folders - Tested on Windows 11.
createLocalCache <- function() {
  if (!file.exists("cached_data")) {
    dir.create("cached_data")
  }
  if (!file.exists(CACHED_DATA_LOCATION)) {
    dir.create(CACHED_DATA_LOCATION)
  }
  if (!file.exists(str_c(CACHED_DATA_LOCATION, "csse_covid_19_time_series"))) {
    dir.create(str_c(CACHED_DATA_LOCATION, "csse_covid_19_time_series"))
  }
}
```

```
# Download the data files from source
downloadDataFiles <- function() {
  createLocalCache()
  for (file_name in names(data_files)) {
    cached_fie <- str_c(CACHED_DATA_LOCATION, data_files[[file_name]])
    # if local file is
    if (!file.exists(cached_fie)) {
      d_file <- str_c(GITHUB_URL_JH, data_files[[file_name]])
      print(str_c("Dowbloading ... file:", d_file, " to local as: ", cached_fie))
      download.file(d_file, destfile=cached_fie)
    }
  }
}

# function to format large numbers
formatNumber <- function(x) {
  format( x, digits = 2, scientific=FALSE, big.mark = ","
  )
}

# Scale down cases or deaths per 100K population
scaleCasesAndDeathsByPopulation <- function(cases_or_deaths, current_population) {
  round((cases_or_deaths/current_population) * 100000, 4)
}

# Tidy Fips
# 1. Remove unwanted columns
# 2. Rename the columns to be more R friendly
tidyFipsDataset <- function(fips_raw_df) {
  result <- fips_raw_df %>%
  rename(province_state = "Province_State", country_region = "Country_Region") %>% # rename columns
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
  result
}

# Tidy Global cases and deaths
# This function will first rename column name to be more R friendly
# It will filter out all records that have null in country
# Use pivot to move data columns to separate rows using pattern 12/14/21 - each observation should be o
# Drop null values while pivot
# select only date, country, and cases
tidyGlobalCasesAndDeaths <- function(source_df, col_cases_or_deaths) {
  temp_df1 <- source_df %>%
    rename(province_state = "Province/State", country_region = "Country/Region") %>% # rename columns
    # only pick non-null countries
    filter(!is.na(country_region)) %>%
    # convert only date columns with pattern 12/14/21 to observations (cols to rows)
    pivot_longer(
      # Take all date columns for pivotring , except these columns
        cols = !c(province_state,country_region,Lat,Long),
      names_to = c("date"),
        values_to = col_cases_or_deaths,
      names_pattern = "([0-9]{0,2}/[0-9]{0,2}/[0-9]{0,2})",
        values_drop_na = TRUE
  )

  # replace date string with date column in the more
  temp_df2 <- temp_df1 %>%
    # select only date, country and either cases or deaths
    select(date, province_state, country_region, any_of(col_cases_or_deaths)) %>%
    # string 6/23/21 is converted to date
    mutate(
      date = parse_date(date, format = "%m/%d/%y")
    ) %>%
    # convert country to a categoorical variable.
    mutate(
      country_region = factor(country_region)
    )

  # filter and replace NA with 0
  result <- temp_df2 %>% replace_na(list(deaths=0, cases=0))

  result
}

# Tidy the US dataset by moving date columns to rows as observations
# Change date from string to date
# Convert state to factor variable
tidyUSACasesAndDeaths <- function(source_df, col_cases_or_deaths) {
  # cols to rows
  temp_df1 <- source_df %>%
    pivot_longer(
    cols = -(UID:Combined_Key),
    names_to = c("date"),
    values_to = col_cases_or_deaths,
    names_pattern = "([0-9]{0,2}/[0-9]{0,2}/[0-9]{0,2})",
    values_drop_na = TRUE
  )

  # create date data type
  result <- temp_df1 %>%
    rename(state = Province_State, county = Admin2, fips = FIPS ) %>%
    select(date, county, state, fips, any_of(col_cases_or_deaths), any_of("population")) %>%
    mutate(
      date = parse_date(date, format = "%m/%d/%y")
    ) %>%
    replace_na(list(deaths=0, cases=0)) %>%  # replace null with 0
    mutate(state = factor(state)) # create factor/categorical variable for state
  result
}

# Augment 7 day increases in cases_per_100k and deaths_per_100k
# We want to know the average increase in cases in last 7 days
# Avg-7-day = (cases-day-1 + cases-day-2 .... cases-day-7) / 7
augment7DayAvgForCasesAndDeaths <- function(cad_df) {
    # Add the average column using lag function
    cad_df1 <- cad_df %>%
      mutate(
        delta_cases_1 =
          case_when(
            ((cases_per_100k - lag(cases_per_100k, order_by = date)) < 0) ~ 0,
            TRUE ~ (cases_per_100k - lag(cases_per_100k, order_by = date))
          ),
        # if todays deaths is 0 and yesterdays is 10 then the delta becomes negative
        # we cannot have negative deaths on a given day
        delta_deaths_1 =
          case_when(
            ((deaths_per_100k - lag(deaths_per_100k, order_by = date)) < 0) ~ 0,
            TRUE ~ (deaths_per_100k - lag(deaths_per_100k, order_by = date))
          ),

        delta_cases_7 = round((
          delta_cases_1 +
          lag(delta_cases_1, 1, order_by = date) +
          lag(delta_cases_1, 2, order_by = date) +
          lag(delta_cases_1, 3, order_by = date) +
          lag(delta_cases_1, 4, order_by = date) +
          lag(delta_cases_1, 5, order_by = date) +
          lag(delta_cases_1, 6, order_by = date)
        ) / 7, 4), # divide by 7 and round to 4 digits

        delta_deaths_7 = round((
          delta_deaths_1 +
          lag(delta_deaths_1, 1, order_by = date) +
          lag(delta_deaths_1, 2, order_by = date) +
          lag(delta_deaths_1, 3, order_by = date) +
          lag(delta_deaths_1, 4, order_by = date) +
          lag(delta_deaths_1, 5, order_by = date) +
          lag(delta_deaths_1, 6, order_by = date)
        ) / 7, 4)

    ) %>%
    replace_na(list(delta_cases_1=0, delta_deaths_1=0,
                    delta_cases_7=0, delta_deaths_7=0)) # replace NAs with 0

  cad_df1
}

# Function to Plot the USA Counties
# https://www.rdocumentation.org/packages/usmap/versions/0.7.0/topics/plot_usmap
# Parameters
# cases_deaths_df - Dataframe containing cases and deaths
# state_code : The code of the state (NY, CO, ..)
# column_name : The column name to display
# All other columns are colours an label.
plotMapForUSACounties <- function(cases_deaths_df, state_code, column_name,
                          color1, color_low, color_high, plot_name, plot_title) {
  plt <- plot_usmap(regions = "county", include=state_code,data=cases_deaths_df,
             values=column_name, color = color1, labels=TRUE) +
    scale_fill_continuous(low=color_low, high=color_high, name=plot_name) +
    theme(legend.position = "right")
  plt$layers[[2]]$aes_params$size <- 2
  plt <- plt + ggtitle(plot_title)
  plt
}
```

Section 1B: Import the datasets and cache locally:

- Import global COVID-19 statistics aggregated by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University.
- URL : https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/
- We will cache this data so that we don’t have to reload it everytime.

```
downloadDataFiles()
# Load data to Tibbles
csse_global_cases_raw_df <- read_csv(str_c(CACHED_DATA_LOCATION, data_files[["global_cases"]]))
csse_global_deaths_raw_df <- read_csv(str_c(CACHED_DATA_LOCATION, data_files[["global_deaths"]]))
csse_us_cases_raw_df <- read_csv(str_c(CACHED_DATA_LOCATION, data_files[["us_cases"]]))
csse_us_deaths_raw_df <- read_csv(str_c(CACHED_DATA_LOCATION, data_files[["us_deaths"]]))
fips_raw_df <- read_csv(str_c(CACHED_DATA_LOCATION, data_files[["fips"]]))
```

Section 1C: Prepare the FIPS dataset

- Tidy the Fips data

```
fips_df <- tidyFipsDataset(fips_raw_df)
```

Section 1D: Prepare the USA dataset

- Tidy the us-cases, us-deaths datasets

```
# Tidy the US cases and deaths
csse_us_cases_df <- tidyUSACasesAndDeaths(csse_us_cases_raw_df, "cases")

# csse_us_deaths_raw_df has the population column, do rename the columns for ease of use
csse_us_deaths_temp_df <- csse_us_deaths_raw_df %>%
    relocate(Population, .before = Combined_Key) %>%
    rename(population = Population)
csse_us_deaths_df <- tidyUSACasesAndDeaths(csse_us_deaths_temp_df, "deaths")

# Todo - To get insights into a small window
#csse_us_cases_df <- csse_us_cases_df %>% filter(date >= "2020-03-15" & date <= "2021-12-31")
#csse_us_deaths_df <- csse_us_deaths_df %>% filter(date >= "2020-03-15" & date <= "2021-12-31")

# Join the US cases and deaths
csse_us_cases_deaths_df1 <- csse_us_cases_df %>%
  full_join(csse_us_deaths_df) %>%
  replace_na(list(cases=0, deaths=0))
```

```r
# We are going to do state level analysis per day.
# Group by date and state and get the total cases and deaths per day for the state
csse_us_cases_deaths_df2 <- csse_us_cases_deaths_df1 %>%
  group_by(date, state) %>%
  summarise(cases = sum(cases), deaths = sum(deaths), population=sum(population)) %>%
  ungroup()

# Create new columns to scale down the cases and deaths using per 100 k approach. Columns are:
# cases_per_100k - Cases per 100 K of population
# deaths_per_100k - Deaths per 100 k of population
csse_us_cases_deaths_df <- csse_us_cases_deaths_df2 %>%
  filter(population > 0) %>%
  mutate(cases_per_100k = scaleCasesAndDeathsByPopulation(cases, population),
         deaths_per_100k = scaleCasesAndDeathsByPopulation(deaths, population)) %>%
  filter(!is.na(cases_per_100k), !is.na(deaths_per_100k)) %>%
  # rank the dataframe based on total cases per 100k
  mutate(c_rank = rank(desc(cases_per_100k), ties.method="first")) %>%
  # rank the dataframe based on total deaths per 100k
  mutate(d_rank = rank(desc(deaths_per_100k), ties.method="first"))

# Augment the data with 7 day increases in cases and deaths
csse_us_cases_deaths_7d_df <- augment7DayAvgForCasesAndDeaths(csse_us_cases_deaths_df)
# display the data
csse_us_cases_deaths_7d_df %>% head(n=2) %>% print(width=Inf)
```

```
## # A tibble: 2 x 13
##   date       state   cases deaths population cases_per_100k deaths_per_100k
##   <date>     <fct>   <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
## 1 2020-01-22 Alabama     0      0    4903185              0               0
## 2 2020-01-22 Alaska      0      0     740995              0               0
##   c_rank d_rank delta_cases_1 delta_deaths_1 delta_cases_7 delta_deaths_7
##    <int>  <int>         <dbl>          <dbl>         <dbl>          <dbl>
## 1  61040  60081             0              0             0              0
## 2  61041  60082             0              0             0              0
```

```r


ANA_FAV_STATE_YEAR <- 2023
## Prepare the data for the Favourite State
# Group the data by date, state and US county
# sum up cases, deaths and population by grouping field.
csse_fav_state_cases_deaths_df1 <- csse_us_cases_deaths_df1 %>%
  filter(year(date) < ANA_FAV_STATE_YEAR) %>%
  filter(state == ANA_FAV_STATE) %>%
  group_by(date, state, county, fips) %>%
  summarise(cases = sum(cases), deaths = sum(deaths), population=sum(population)) %>%
  replace_na(list(cases=0, deaths=0)) %>%
  ungroup()

# Transform - Create columns cases_per_100k and deaths_per_100k for direct comparison between counties.
# replace nulls with 0
# Exlcude all columns with infinity
```

```
csse_fav_state_cases_deaths_df2 <- csse_fav_state_cases_deaths_df1 %>%
  filter(population > 0) %>%
  mutate(cases_per_100k = scaleCasesAndDeathsByPopulation(cases, population),
         deaths_per_100k = scaleCasesAndDeathsByPopulation(deaths, population)) %>%
  filter(!is.na(cases_per_100k), !is.na(deaths_per_100k)) %>%
  select(date,state,county,fips,cases_per_100k, deaths_per_100k) %>%
  replace_na(list(cases_per_100k=0, deaths_per_100k=0))
  #filter(!is.infinite(cases_per_100k), !is.infinite(deaths_per_100k))

# rank the data based on cases per 100K population
csse_fav_state_cases_df <- csse_fav_state_cases_deaths_df2 %>%
  mutate(c_rank = rank(desc(cases_per_100k), ties.method="first")) %>%
  group_by(county, fips) %>%
  filter(c_rank == min(c_rank))

# rank the data based on deaths per 100K population
csse_fav_state_deaths_df <- csse_fav_state_cases_deaths_df2 %>%
  mutate(c_rank = rank(desc(deaths_per_100k), ties.method="first")) %>%
  group_by(county, fips) %>%
  filter(c_rank == min(c_rank))

# display the data
csse_fav_state_cases_df %>% head(n=2) %>% print(width=Inf)
```

Section 1E: Prepare dataset for the Favourite US state

```
## # A tibble: 2 x 7
## # Groups:   county, fips [2]
##   date       state    county  fips cases_per_100k deaths_per_100k c_rank
##   <date>     <fct>    <chr>  <dbl>          <dbl>           <dbl>  <int>
## 1 2022-12-29 New York Orange 36071         37985.            326.    258
## 2 2022-12-30 New York Bronx  36005         37443.            589.    316
```

Section 1F: Prepare the Global dataset

- Tidy the global-cases, global-deaths, datasets
- Global cases and deaths should be aggregated by date, country_region, province_state to get daily observatiions for each country.
- Global cases and deaths datasets should be joined together by date, country_region, province_state to form one global dataset containing following observation for each day (date, cases, deaths, country_region, province_state)

```
# Tidy global cases
csse_global_cases_df <- tidyGlobalCasesAndDeaths(csse_global_cases_raw_df, "cases")

# Tidy global deaths
csse_global_deaths_df <- tidyGlobalCasesAndDeaths(csse_global_deaths_raw_df, "deaths")

# Todo - To get insights into a small window
csse_global_cases_df <- csse_global_cases_df %>% filter(date >= "2020-03-15" & date <= "2021-12-31")
csse_global_deaths_df <- csse_global_deaths_df %>% filter(date >= "2020-03-15" & date <= "2021-12-31")
```

```
# Aggregate cases by date, country, state
csse_global_cases_df <- csse_global_cases_df %>%
  group_by(date, country_region, province_state) %>%
  summarise(cases = sum(cases)) %>%
  ungroup()

# Aggregate deaths by date, country, state
csse_global_deaths_df <- csse_global_deaths_df %>%
  group_by(date, country_region, province_state) %>%
  summarise(deaths = sum(deaths)) %>%
  ungroup()

# join the global cases and deaths
csse_global_cases_and_deaths_df <- csse_global_cases_df %>%
  full_join(csse_global_deaths_df, by = c("date", "country_region", "province_state")) %>%
  filter(cases > 0) # filter cases > 0

# to check US cases only
# csse_global_cases_and_deaths_usa <- csse_global_cases_and_deaths %>% filter(country %in% c("US")

# Join the global dataset with the fips dataset to get population for each row and remove unwanted colu
csse_global_cases_and_deaths_df2 <- csse_global_cases_and_deaths_df %>%
  left_join(fips_df, by = c("province_state", "country_region")) %>%
  select(-c(UID, FIPS))

# replace null values with 0 for doing group-by later
# group by date and country to get daily totals for cases, deaths and population.
csse_global_cases_and_deaths_df3 <- csse_global_cases_and_deaths_df2 %>%
  replace_na(list(cases=0, deaths=0, Population=0)) %>%
  group_by(date, country_region) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),   Population = sum(Population)) %>%
  ungroup()

# Create new columns tp scale down the cases and deaths using per 100 k approach. Columns are:
# cases_per_100k - Cases per 100 K of population
# deaths_per_100k - Deaths per 100 k of population
csse_global_df <- csse_global_cases_and_deaths_df3 %>%
  filter(Population > 0) %>%
  mutate(cases_per_100k = scaleCasesAndDeathsByPopulation(cases, Population),
         deaths_per_100k = scaleCasesAndDeathsByPopulation(deaths, Population)) %>%
  filter(!is.na(cases_per_100k), !is.na(deaths_per_100k)) %>%
  # rank the dataframe based on total cases per 100k
  mutate(c_rank = rank(desc(cases_per_100k), ties.method="first")) %>%
  # rank the dataframe based on total deaths per 100k
  mutate(d_rank = rank(desc(deaths_per_100k), ties.method="first"))

# test
# csse_global %>% filter(country_region == "US" | country_region == "China", date == "2021-01-25")

# Augment the data with 7 day increases in cases and deaths
csse_global_7d_inc_df <- augment7DayAvgForCasesAndDeaths(csse_global_df)

# display the data
```

```
csse_global_7d_inc_df %>% head(n=2) %>% print(width=Inf)
```

```
## # A tibble: 2 x 13
##   date       country_region cases deaths Population cases_per_100k
##   <date>     <chr>          <dbl>  <dbl>      <dbl>          <dbl>
## 1 2020-03-15 Afghanistan       20      0   38928341         0.0514
## 2 2020-03-15 Albania           42      1    2877800         1.46
##   deaths_per_100k c_rank d_rank delta_cases_1 delta_deaths_1 delta_cases_7
##             <dbl>  <int>  <int>         <dbl>          <dbl>         <dbl>
## 1          0      122956 114189          0              0                0
## 2          0.0347 117365 111328          1.41           0.0347           0
##   delta_deaths_7
##            <dbl>
## 1              0
## 2              0
```

Section 1G: Summarize the dataset

- Now our data should look good.
- We have all the columns names in the dataset.
  - cases_per_100k : Cases per 100 K population for each country.
  - deaths_per_100k : Deaths per 100 K population for each counntry.
  - delta_cases_7 : This will help us to review the rise in case per week.
  - delta_cases_7 : This will help us to analyze the rise in deaths per week for each country by their population.
- We will use this cleaned dataset to do detail analysis.

```
# summarize Global dataset
summary(csse_global_7d_inc_df)
```

```
##       date            country_region         cases              deaths
##  Min.   :2020-03-15   Length:123536      Min.   :       1   Min.   :     0
##  1st Qu.:2020-08-31   Class :character   1st Qu.:    3033   1st Qu.:    48
##  Median :2021-02-12   Mode  :character   Median :   29106   Median :   512
##  Mean   :2021-02-10                      Mean   :  605252   Mean   : 13397
##  3rd Qu.:2021-07-24                      3rd Qu.:  243742   3rd Qu.:  4679
##  Max.   :2021-12-31                      Max.   :54907717   Max.   :825468
##    Population        cases_per_100k      deaths_per_100k       c_rank
##  Min.   :8.090e+02   Min.   :    0.001   Min.   :  0.000   Min.   :     1
##  1st Qu.:2.226e+06   1st Qu.:   56.075   1st Qu.:  0.805   1st Qu.: 30885
##  Median :9.449e+06   Median :  409.483   Median :  6.914   Median : 61769
##  Mean   :4.106e+07   Mean   : 2343.461   Mean   : 41.836   Mean   : 61769
##  3rd Qu.:2.983e+07   3rd Qu.: 3121.927   3rd Qu.: 52.819   3rd Qu.: 92652
##  Max.   :1.418e+09   Max.   :30725.426   Max.   :614.737   Max.   :123536
##      d_rank        delta_cases_1       delta_deaths_1   delta_cases_7
##  Min.   :     1   Min.   :    0.000   Min.   :  0.00   Min.   :   0.0
##  1st Qu.: 30885   1st Qu.:    0.000   1st Qu.:  0.00   1st Qu.: 234.8
##  Median : 61769   Median :    0.115   Median :  0.00   Median :1047.0
##  Mean   : 61769   Mean   : 1322.016   Mean   : 27.89   Mean   :1322.0
##  3rd Qu.: 92652   3rd Qu.: 1107.917   3rd Qu.: 19.14   3rd Qu.:2068.2
##  Max.   :123536   Max.   :30227.303   Max.   :431.26   Max.   :7847.0
##  delta_deaths_7
##  Min.   :  0.000
##  1st Qu.:  6.949
##  Median : 20.700
##  Mean   : 27.889
##  3rd Qu.: 42.736
##  Max.   :157.362
```

```
# summarize US dataset
summary(csse_us_cases_deaths_7d_df)
```

```
##       date                           state            cases         
##  Min.   :2020-01-22   Alabama        : 1143   Min.   :       0  
##  1st Qu.:2020-11-02   Alaska         : 1143   1st Qu.:   48429  
##  Median :2021-08-15   American Samoa: 1143   Median :  313342  
##  Mean   :2021-08-15   Arizona        : 1143   Mean   :  840726  
##  3rd Qu.:2022-05-28   Arkansas       : 1143   3rd Qu.:  988434  
##  Max.   :2023-03-09   California     : 1143   Max.   :12129699  
##                       (Other)        :57150                     
##      deaths          population        cases_per_100k  deaths_per_100k
##  Min.   :     0   Min.   :   55144   Min.   :    0   Min.   :  0.0  
##  1st Qu.:   772   1st Qu.: 1355836   1st Qu.: 2250   1st Qu.: 46.9  
##  Median :  4395   Median : 3855955   Median :11211   Median :163.5  
##  Mean   : 11153   Mean   : 5944199   Mean   :13815   Mean   :168.9  
##  3rd Qu.: 14225   3rd Qu.: 6989056   3rd Qu.:24809   3rd Qu.:271.2  
##  Max.   :101159   Max.   :39512223   Max.   :43488   Max.   :454.8  
##                                                                   
##      c_rank          d_rank      delta_cases_1    delta_deaths_1  
##  Min.   :    1   Min.   :    1   Min.   :    0   Min.   :  0.00  
##  1st Qu.:16003   1st Qu.:16003   1st Qu.:    0   1st Qu.:  0.00  
##  Median :32005   Median :32005   Median :    0   Median :  0.00  
##  Mean   :32005   Mean   :32005   Mean   : 1607   Mean   : 37.14  
##  3rd Qu.:48006   3rd Qu.:48006   3rd Qu.: 1865   3rd Qu.: 57.94  
##  Max.   :64008   Max.   :64008   Max.   :27009   Max.   :398.51  
##                                                                   
##  delta_cases_7    delta_deaths_7  
##  Min.   :   0.0   Min.   :  0.00  
##  1st Qu.: 568.6   1st Qu.: 19.33  
##  Median :1455.2   Median : 36.65  
##  Mean   :1607.2   Mean   : 37.14  
##  3rd Qu.:2327.8   3rd Qu.: 53.67  
##  Max.   :5983.9   Max.   :111.42  
## 
```

Section 2A: Analyze the global datasets

- Univariate Analysis
  - Which are the Top 3 countries in terms of cases? Can we compare them?
- Bivariate Analysis

    - is there a correlation between the cases and deaths of top 3 countries?

- Multivariate Analysis
    - How was the top 10 countries managing the pandemic?
        * Prepare a timeseries data to show the 7 day increases in cases and deaths.

```
# Find the Top 10 Country names with most cases as on certain date.
top10_countries_cases_lst <-  csse_global_7d_inc_df %>%
    filter(!is.infinite(cases_per_100k)) %>%
    filter(date == ANA_CAD_AS_ON_DATE) %>%
    slice_max(n = 10, order_by = cases_per_100k) %>%
  pull(country_region)

# slice the dataset
csse_global_top3_countries_lst <- top10_countries_cases_lst[1:3]

# Create the dataset for the Top 10 Countries with most cases for a selected date
# remove infinite observations and create categorical var for country_region
csse_global_top10_countries_cases_df <- csse_global_7d_inc_df %>%
  filter(country_region %in% top10_countries_cases_lst) %>%
  filter(!is.infinite(cases_per_100k)) %>%
    filter(date == ANA_CAD_AS_ON_DATE) %>%
  mutate(country_region = factor(country_region, levels=top10_countries_cases_lst))

# Timeseries data - Create the timeseries dataset for the Top 10 Countries with most cases for all date
# remove infinite observations and create categorical var for country_region
csse_global_top10_countries_cases_ts_df <- csse_global_7d_inc_df %>%
  filter(country_region %in% top10_countries_cases_lst) %>%
  filter(!is.infinite(cases)) %>%
  mutate(country_region = factor(country_region, levels=top10_countries_cases_lst))

# Timeseries data - Create the dataset for the Top 3 Countries with most cases for all dates
csse_global_top3_countries_cases_ts_df <- csse_global_top10_countries_cases_ts_df %>%
  filter(country_region %in% csse_global_top3_countries_lst)

# display the data
csse_global_top3_countries_cases_ts_df %>% head(n=2) %>% print(width=Inf)
```

```
## # A tibble: 2 x 13
##   date       country_region cases deaths Population cases_per_100k
##   <date>     <fct>          <dbl>  <dbl>      <dbl>          <dbl>
## 1 2020-03-15 Andorra            1      0      77265           1.29
## 2 2020-03-15 Slovakia          44      0    5434712           0.810
##   deaths_per_100k c_rank d_rank delta_cases_1 delta_deaths_1 delta_cases_7
##             <dbl>  <int>  <int>         <dbl>          <dbl>         <dbl>
## 1               0 117760 114190          1.18              0           0
## 2               0 119721 114264          0                 0          33.1
##   delta_deaths_7
##            <dbl>
## 1           0
## 2           2.10
```

```
# ---------------------------------------

# Create a list of the names of the Top 10 Countries with the most deaths
top10_countries_deaths_lst <-  csse_global_7d_inc_df %>%
    filter(!is.infinite(deaths_per_100k)) %>%
    filter(date == ANA_CAD_AS_ON_DATE) %>%
    slice_max(n = 10, order_by = deaths_per_100k) %>%
  pull(country_region)

# Slice top3
top3_countries_deaths_lst <- top10_countries_deaths_lst[1:3]

# Create the dataset for the Top 10 Countries with most deaths for selected date
# filter infinite
# create factor var for country_region
csse_global_top10_countries_deaths_df <- csse_global_7d_inc_df %>%
  filter(country_region %in% top10_countries_deaths_lst) %>%
  filter(!is.infinite(deaths_per_100k)) %>%
    filter(date == ANA_CAD_AS_ON_DATE) %>%
  mutate(country_region = factor(country_region, levels=top10_countries_deaths_lst))

# Timeseries data - Create the timeseries dataset for the Top 10 Countries with most deaths for all dat
# filter infinite
# create factor var for country_region
csse_global_top10_countries_deaths_ts_df <- csse_global_7d_inc_df %>%
  filter(country_region %in% top10_countries_deaths_lst) %>%
  filter(!is.infinite(cases)) %>%
  mutate(country_region = factor(country_region, levels=top10_countries_deaths_lst))

# Timeseries data - Create the timeseries dataset for the Top 3 Countries with most deaths for all date
csse_global_top3_countries_deaths_ts_df <- csse_global_top10_countries_deaths_ts_df %>%
  filter(country_region %in% top3_countries_deaths_lst)

# display the data
csse_global_top3_countries_deaths_ts_df %>% head(n=2) %>% print(width=Inf)
```

```
## # A tibble: 2 x 13
##   date       country_region         cases deaths Population cases_per_100k
##   <date>     <fct>                  <dbl>  <dbl>      <dbl>          <dbl>
## 1 2020-03-15 Bosnia and Herzegovina    24      0    3280815          0.732
## 2 2020-03-15 Bulgaria                  51      2    6948445          0.734
##   deaths_per_100k c_rank d_rank delta_cases_1 delta_deaths_1 delta_cases_7
##             <dbl>  <int>  <int>         <dbl>          <dbl>         <dbl>
## 1          0      119887 114198         0.646         0               2.96
## 2          0.0288 119882 111891         0             0.0288          2.77
##   delta_deaths_7
##            <dbl>
## 1         0.005
## 2         0.0091
```

Section 2B: Prepare the graphs for the Global dataset

- Create the graphs to visualize the top 10 countries in terms of cases and report observations

```
# Univariate Analysis - Create pie chart for Cases per 100K for top 10 countries with most cases
# created the labels with country and cases
# reduced the size of labels.
# used reorder to sort the data
csse_global_cases_top10_countries_vz <- csse_global_top10_countries_cases_df %>%
  mutate(top10_cases_per_100k = paste0(country_region, '..', formatNumber(cases_per_100k))) %>%
  mutate(top10_cases_100k = reorder(top10_cases_per_100k, -cases_per_100k)) %>%
  ggplot(aes(x="", y=-cases_per_100k, fill=top10_cases_100k)) +
  geom_bar(stat="identity", width=2) +
  coord_polar("y", start=0) +
  xlab("") + ylab("") +
  ggtitle(str_c("Top 10 Countries based on cases per 100K\n As on: ", ANA_CAD_AS_ON_DATE)) +
  theme(legend.direction="vertical") +
  geom_text(aes(label = country_region ), size=3, position=position_stack(vjust=0.7)) +
  theme_light()

# Univariate Analysis - Create pie chart for Deaths per 100K for top 10 countries with most deaths
# created the labels with country and cases
# reduced the size of labels.
# used reorder to sort the data
csse_global_deaths_top10_countries_vz <- csse_global_top10_countries_deaths_df %>%
  mutate(top10_deaths_per_100k = paste0(country_region, '-', formatNumber(deaths_per_100k))) %>%
  mutate(top10_deaths_100k=reorder(top10_deaths_per_100k, -deaths_per_100k)) %>%
  ggplot(aes(x="", y=-deaths_per_100k, fill=top10_deaths_100k)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  xlab("") + ylab("") + ggtitle(str_c("Top 10 Countries based on deaths per 100K\n As on: ", ANA_CAD_AS_
  theme(legend.direction="vertical") +
  geom_text(aes(label = country_region ), size=3, position=position_stack(vjust=0.7)) +
  theme_light()

#Bivariate Analysis - Is there a correlation between cases and deaths
#geom_point is a good chart for doing timeseries analysis
csse_top3_global_cad_corr_vz <- csse_global_top3_countries_cases_ts_df %>%
  filter(delta_deaths_7>60) %>%
  ggplot() +
  geom_point(aes(x = delta_deaths_7, y = delta_cases_7), size=1, color='coral') +
  theme_light()

# Multivariate Analysis
#geom_point is a good chart for doing timeseries analysis
css_global_cases_7day_increase_vz <- csse_global_top10_countries_cases_ts_df %>%
  ggplot(aes(x=date, y=delta_cases_7, fill=country_region)) +
  geom_point(stat='identity', aes(color=country_region), size=1, alpha=0.8) +
  labs(
    title = "Top 10 Countries - 7 day increase in cases."
  ) +
  theme_light()

css_global_deaths_7day_increase_vz <- csse_global_top10_countries_deaths_ts_df %>%
  ggplot(aes(x=date, y=delta_deaths_7, fill=country_region)) +
  geom_point(stat='identity', aes(color=country_region), size=1, alpha=0.8) +
  labs(
    title = "Top 10 Countries - 7 day increase in deaths."
  ) +
  theme_light()

# filter top 3 for closer look
# Using facet_wrap we can show the data based on categories on different rows
# For spacing between labels we will use breaks
# Used theme to change the text
css_global_cases_top3_7day_increase_vz <- csse_global_top3_countries_cases_ts_df %>%
  ggplot(aes(x=date, y=delta_cases_7, fill=country_region)) +
  geom_bar(stat='identity', aes(color=country_region), alpha=0.8) +
  facet_wrap(~country_region, scales="free_y", nrow=3) +
  scale_x_date(
    date_breaks = "4 months",
    date_labels = "%b %Y"
  ) +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "coral"),
    strip.text = element_text(colour = "white", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    title = "Top 3 Countries - Daily 7 day deltas of Cases (Per 100K) "
  )

# filter top 3 for closer look
# Using facet_wrap we can show the data based on categories on different rows
# For spacing between labels we will use breaks
# Used theme to change the text
css_global_deaths_top3_7day_increase_vz <- csse_global_top3_countries_deaths_ts_df %>%
  ggplot(aes(x=date, y=delta_deaths_7, fill=country_region)) +
  geom_bar(stat='identity', alpha=0.8) +
  facet_wrap(~country_region, scales="free_y", nrow=3) +
  scale_fill_manual(values=c("red" ,"rosybrown3", "rosybrown1")) +
  scale_x_date(
    date_breaks = "4 months",
    date_labels = "%b %Y"
  ) +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "coral"),
    strip.text = element_text(colour = "white", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    title = "Top 3 Countries - Daily 7 day deltas of Deaths  (Per 100K) "
  )
```

Section 3A: Analyze the USA datasets

- Univariate Analysis
  - Which are the Top 3 states in terms of cases?
- Bivariate Analysis
  - is there a correlation between the cases and deaths of top 3 states?
- Multivariate Analysis
  - How was the top 10 states managing the cases?
    * Prepare a timeseries data to show the 7 day increases in cases and deaths.

```
# Find the Top 5 states with most cases
# Filter based on date and get max 10 state names
top10_states_cases_lst <-  csse_us_cases_deaths_7d_df %>%
    filter(!is.infinite(cases_per_100k)) %>%
    filter(date == ANA_CAD_AS_ON_DATE) %>%
    slice_max(n = 10, order_by = cases_per_100k) %>%
  pull(state)

# slice the top 10
top3_states_cases_lst <- top10_states_cases_lst[1:3]

# Create the dataset for the Top 10 states with most cases for selected date
# remove infinie values and fetch records for one date
# create factor for state
csse_top10_us_cases_df <- csse_us_cases_deaths_7d_df %>%
  filter(state %in% top10_states_cases_lst) %>%
  filter(!is.infinite(cases_per_100k)) %>%
    filter(date == ANA_CAD_AS_ON_DATE) %>%
  mutate(state = factor(state, levels=top10_states_cases_lst))

# Timeseries data - Create the timeseries dataset for the Top 10 state with most cases for all dates
# remove infinie values and fetch records for one date
# create factor for state
csse_top10_us_cases_ts_df <- csse_us_cases_deaths_7d_df %>%
  filter(state %in% top10_states_cases_lst) %>%
  filter(!is.infinite(cases)) %>%
  mutate(state = factor(state, levels=top10_states_cases_lst))

# Timeseries data - Create the dataset for the Top 3 states with most cases for all dates
csse_top3_us_cases_ts_df <- csse_top10_us_cases_ts_df %>%
  filter(state %in% top3_states_cases_lst)

# ----------------- Prepare the deaths df

# Find the Top 5 states with most deaths
# select records for one date and select 10 names
top10_states_deaths_lst <-  csse_us_cases_deaths_7d_df %>%
    filter(!is.infinite(deaths_per_100k)) %>%
    filter(date == ANA_CAD_AS_ON_DATE) %>%
    slice_max(n = 10, order_by = deaths_per_100k) %>%
  pull(state)

# slice top 3
top3_states_deaths_lst <- top10_states_deaths_lst[1:3]

# Create the dataset for the Top 10 states with most deaths for selected date
# remove infinie values and fetch records for one date
# create factor for state
csse_top10_us_deaths_df <- csse_us_cases_deaths_7d_df %>%
  filter(state %in% top10_states_deaths_lst) %>%
  filter(!is.infinite(deaths_per_100k)) %>%
    filter(date == ANA_CAD_AS_ON_DATE) %>%
  mutate(state = factor(state, levels=top10_states_deaths_lst))

# Timeseries data - Create the timeseries dataset for the Top 10 states with most deaths for all dates
# remove infinie values and fetch records for one date
# create factor for state
csse_top10_us_deaths_ts_df <- csse_us_cases_deaths_7d_df %>%
  filter(state %in% top10_states_deaths_lst) %>%
  filter(!is.infinite(deaths_per_100k)) %>%
  mutate(state = factor(state, levels=top10_states_deaths_lst))

# Timeseries data - Create the dataset for the Top 3 states with most deaths for all dates
csse_top3_us_deaths_ts_df <- csse_top10_us_deaths_ts_df %>%
  filter(state %in% top3_states_deaths_lst)
```

Section 3B: Visualize: Prepare the graphs for the US dataset

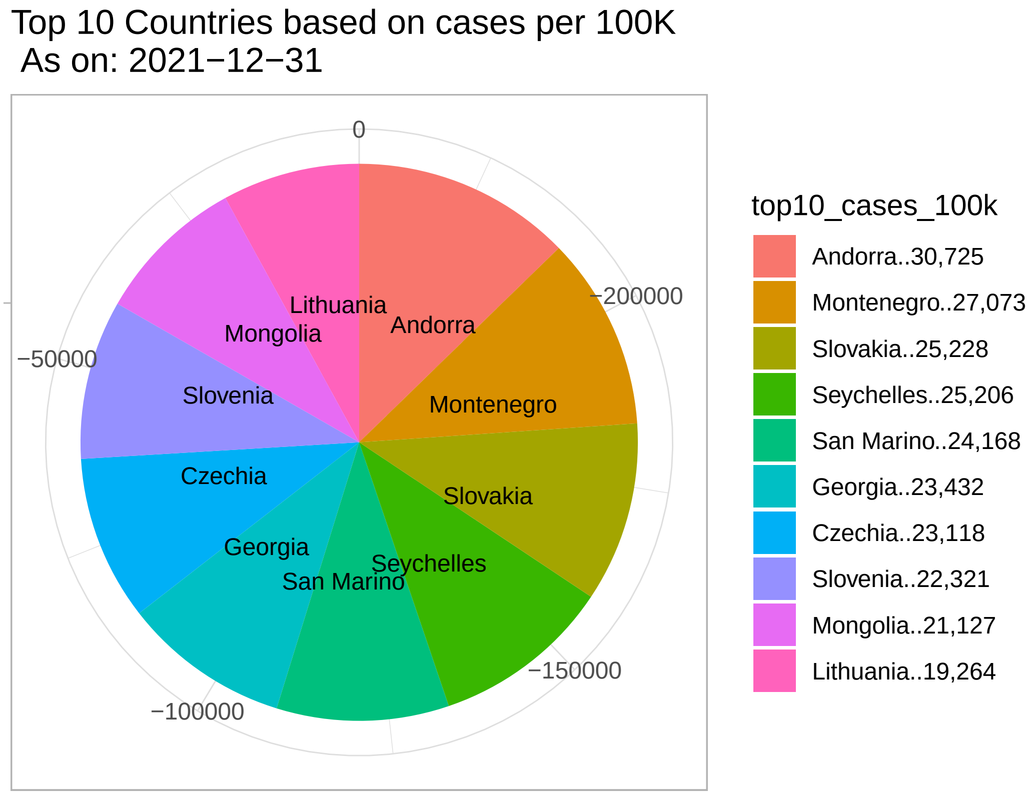
- Create the graphs to visualize the top 10 countries in terms of cases and report observations

```
# Prepare Visualizations
# Univariate Analysis - Create pie chart for Cases per 100K for top 10 states with most cases
# created the labels with country and cases
# reduced the size of labels using geom_text and used a light theme.
# used reorder to sort the data
csse_top10_us_cases_vz <- csse_top10_us_cases_df %>%
  # append casses to the country in the legend
  mutate(top10_cases_per_100k = paste0(state, '..', formatNumber(cases_per_100k))) %>%
  mutate(top10_cases_100k = reorder(top10_cases_per_100k, -cases_per_100k)) %>%
  ggplot(aes(x="", y=-cases_per_100k, fill=top10_cases_100k)) +
  geom_bar(stat="identity", width=2) +
  coord_polar("y", start=0) +
  xlab("") + ylab("") + ggtitle(str_c("Top 10 states based on cases per 100K\n As on: ", ANA_CAD_AS_ON_
  theme(legend.direction="vertical") +
  geom_text(aes(label = state ), size=3, position=position_stack(vjust=0.7)) +
  theme_light()

# Univariate Analysis - Create pie chart for Deaths per 100K for top 10 countries with most cases
# created the labels with country and cases
# reduced the size of labels using geom_text and used a light theme.
csse_top10_us_deaths_vz <- csse_top10_us_deaths_df %>%
  mutate(top10_deaths_per_100k = paste0(state, '..', formatNumber(deaths_per_100k))) %>%
  mutate(top10_deaths_100k = reorder(top10_deaths_per_100k, -deaths_per_100k)) %>%
  ggplot(aes(x="", y=-deaths_per_100k, fill=top10_deaths_100k)) +
  geom_bar(stat="identity", width=2) +
  coord_polar("y", start=0) +
  xlab("") + ylab("") + ggtitle(str_c("Top 10 states based on deaths per 100K\n As on: ", ANA_CAD_AS_ON_
  theme(legend.direction="vertical") +
  geom_text(aes(label = state ), size=3, position=position_stack(vjust=0.7)) +
  theme_light()

# Bivariate Analysis - Is there a correlation between cases and deaths
# geom_point is a good chart for doing timeseries analysis
csse_top3_us_cad_corr_vz <- csse_top3_us_cases_ts_df %>% filter(delta_deaths_7>60) %>%
  ggplot() +
  geom_point(aes(x = delta_deaths_7, y = delta_cases_7), size=1, color='coral') +
  theme_light()

# Multivariate Analysis
# 7 day increase in cases per 100K for top10 US states with most cases
# geom_point is a good chart for doing timeseries analysis
csse_top10_us_cases_7day_inc_vz <- csse_top10_us_cases_ts_df %>%
  ggplot(aes(x=date, y=delta_cases_7, fill=state)) +
  geom_point(stat='identity', aes(color=state), size=1, alpha=0.8) +
  labs(
    title = "Top 10 states - 7 day increase in cases."
  ) +
  theme_light()

# 7 day increase in cases per 100K for top10 US states with most cases
# geom_point is a good chart for doing timeseries analysis
csse_top10_us_deaths_7day_inc_vz <- csse_top10_us_deaths_ts_df %>%
  ggplot(aes(x=date, y=delta_deaths_7, fill=state)) +
  geom_point(stat='identity', aes(color=state), size=1, alpha=0.8) +
  labs(
    title = "Top 10 states - 7 day increase in deaths"
  ) +
  theme_light()

# filter top 3 cases for closer look
# geom_point is a good chart for doing timeseries analysis
# Using facet_wrap we can show the data based on categories on different rows
# Used theme to change the text
csse_top3_us_cases_7day_inc_vz <- csse_top3_us_cases_ts_df %>%
  ggplot(aes(x=date, y=delta_cases_7, fill=state)) +
  geom_bar(stat='identity', aes(color=state), alpha=0.8) +
  facet_wrap(~state, scales="free_y", nrow=3) +
  scale_x_date(
    date_breaks = "4 months",
    date_labels = "%b %Y"
  ) +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "coral"),
    strip.text = element_text(colour = "white", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    title = "Top 3 States - Daily 7 day deltas of Cases (Per 100K) "
  )

# filter top 3 deaths for closer look
# Using facet_wrap we can show the data based on categories on different rows
# Used theme to change the text
csse_top3_us_deaths_7day_inc_vz <- csse_top3_us_deaths_ts_df %>%
  ggplot(aes(x=date, y=delta_deaths_7, fill=state)) +
  geom_bar(stat='identity', alpha=0.8) +
  facet_wrap(~state, scales="free_y", nrow=3) +
  scale_fill_manual(values=c("red" ,"rosybrown3", "rosybrown1")) +
  scale_x_date(
    date_breaks = "4 months",
    date_labels = "%b %Y"
  ) +
  theme_light() +
  theme(
    #strip.background = element_rect(fill = "coral"),
    strip.text = element_text(colour = "white", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    title = "Top 3 States - Daily 7 day deltas of Deaths (Per 100K) "
  )

# Plot a heat map of NYC counties to show the cases per 100K population
csse_fav_state_cases_vz <- plotMapForUSACounties(
  csse_fav_state_cases_df, ANA_FAV_STATE_ABR, "cases_per_100k",
  "coral", "yellow", "coral3", "Cases per 100k",
  "New York - Cases per 100K population")

# Plot a heat map of NYC counties to show the deaths per 100K population
csse_fav_state_deaths_vz <- plotMapForUSACounties(
  csse_fav_state_deaths_df, ANA_FAV_STATE_ABR, "deaths_per_100k",
  "red", "peachpuff1", "red", "Deaths per 100k",
  "New York - Deaths per 100K population")

# -- Add white space for moving next section to new page.
#
#
#
#
#
#
#
```

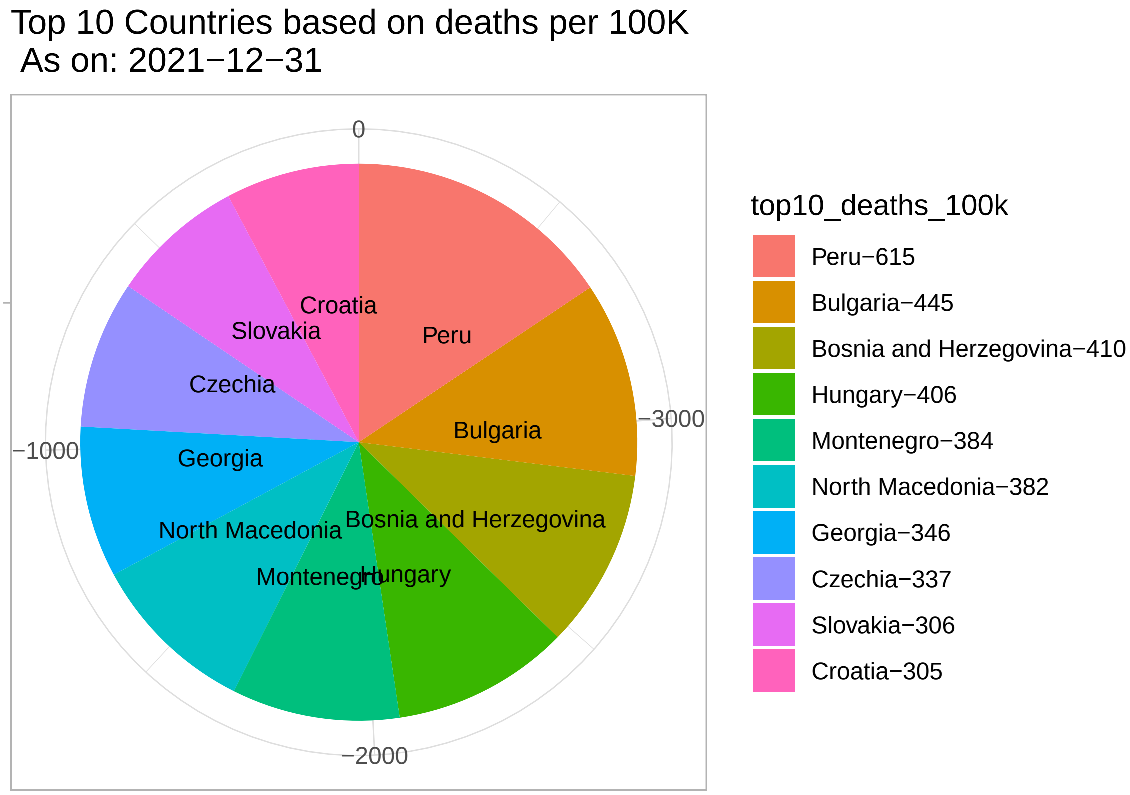
Section 4: Visualizations

Section 4A: Univariate Analysis - Cases and Deaths per 100k population of Top 10 countries as on 2021-12-31

- **Which are the top 10 countries in terms of cases and deaths per 100k population?**
  - The size of the pie clearly shows the relative sizes of the Top 10 countries in terms of cases per 100k population.
  - The cases/100k-of-population approach clearly helps us to identify the top 10 counties.
  - Even though this is chart is not a histogram of distribution, we can see the share of cases each country has in the top 10 bucket.

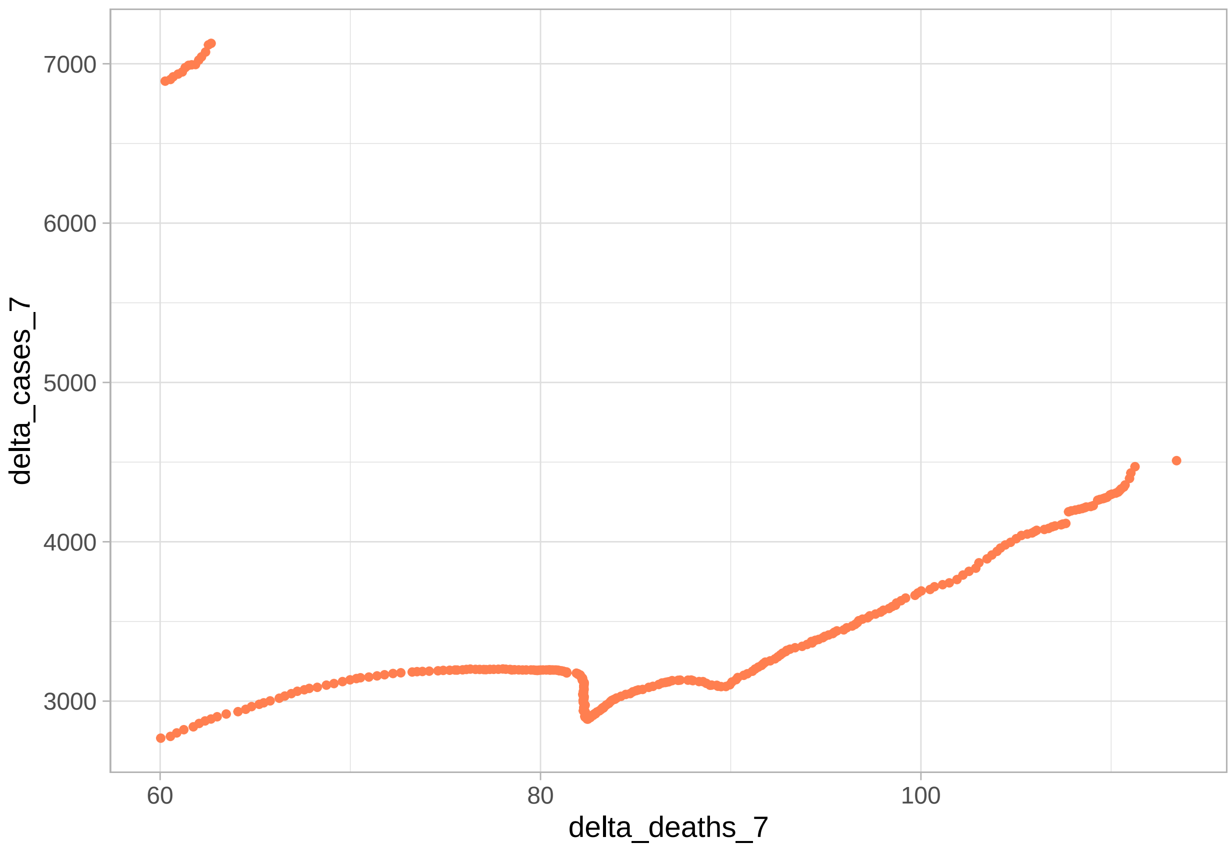


- Here we observed that the top 10 countries with more deaths were not the same as top 10 countries with cases. This is because the deaths may be related to not detecting the cases earlier.
- There could be other reasons, such as shortage of health care facilities. etc.



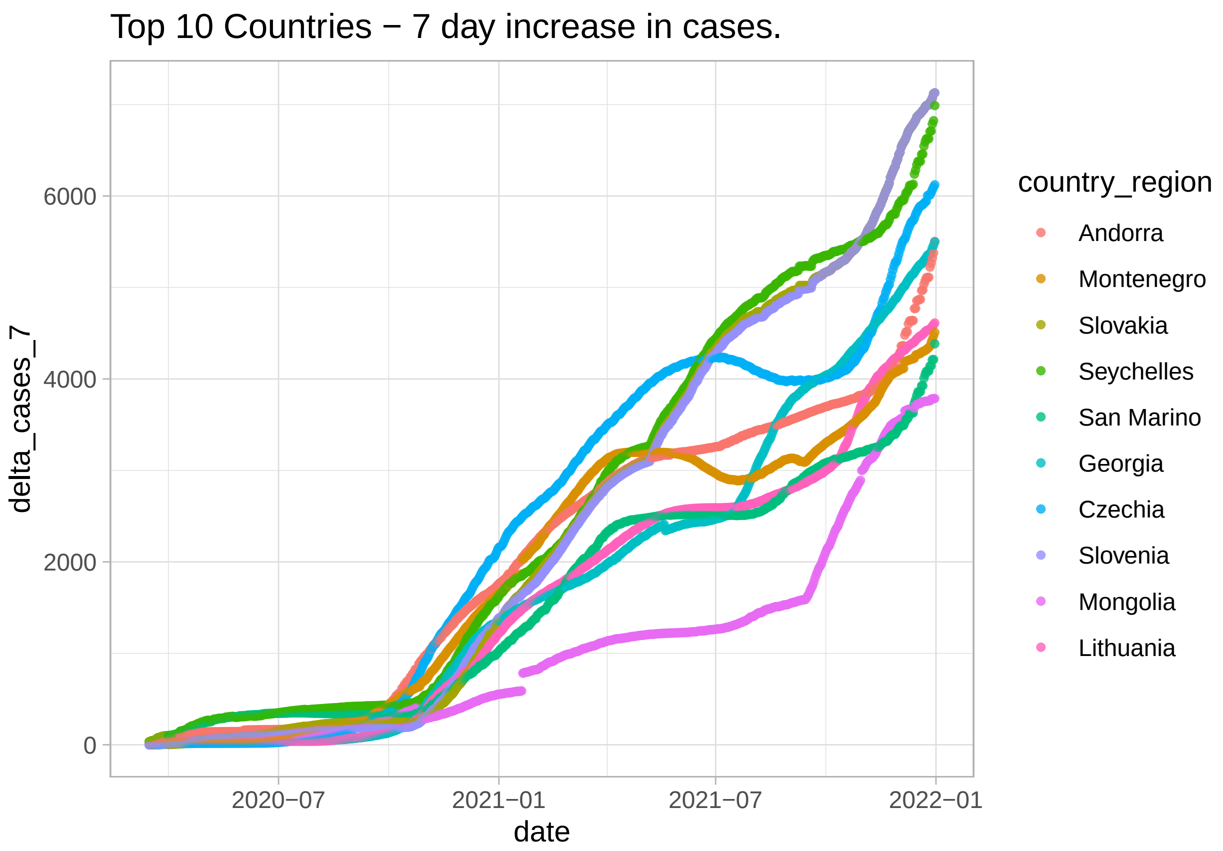
Section 4B: Bivariate Analysis - Cases and Deaths per 100k population for Top 3 countries as on 2021-12-31

- **Is there a correlation between cases and deaths for the Top 3 countries battling the pandemic?**
  - The question helps us to understand if all the cases reported will result in deaths.
  - We observed that there was no one-to-one correlation between cases and deaths.
  - All deaths were from the cases that were reported, however, all cases may not lead to death since people get cured too.
  - If we can spot that the cases are decreasing and the deaths are few, then we can conclude that the country is managing the pandemic very well. We should share the approach with other countries so that other countries benefit.

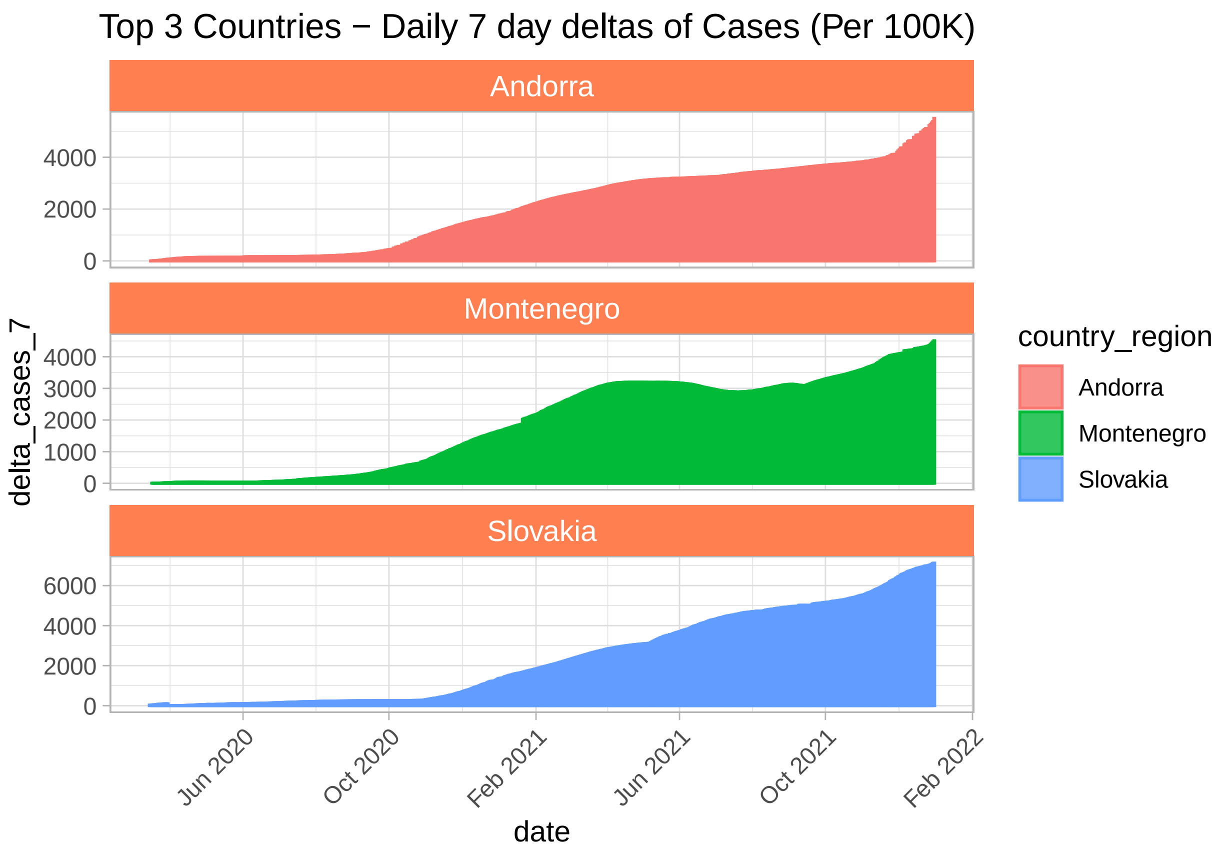


Section 4C: Multivariate Analysis Understanding the trends in Cases and Deaths per 100k population of Top 10 countries as on 2021-12-31

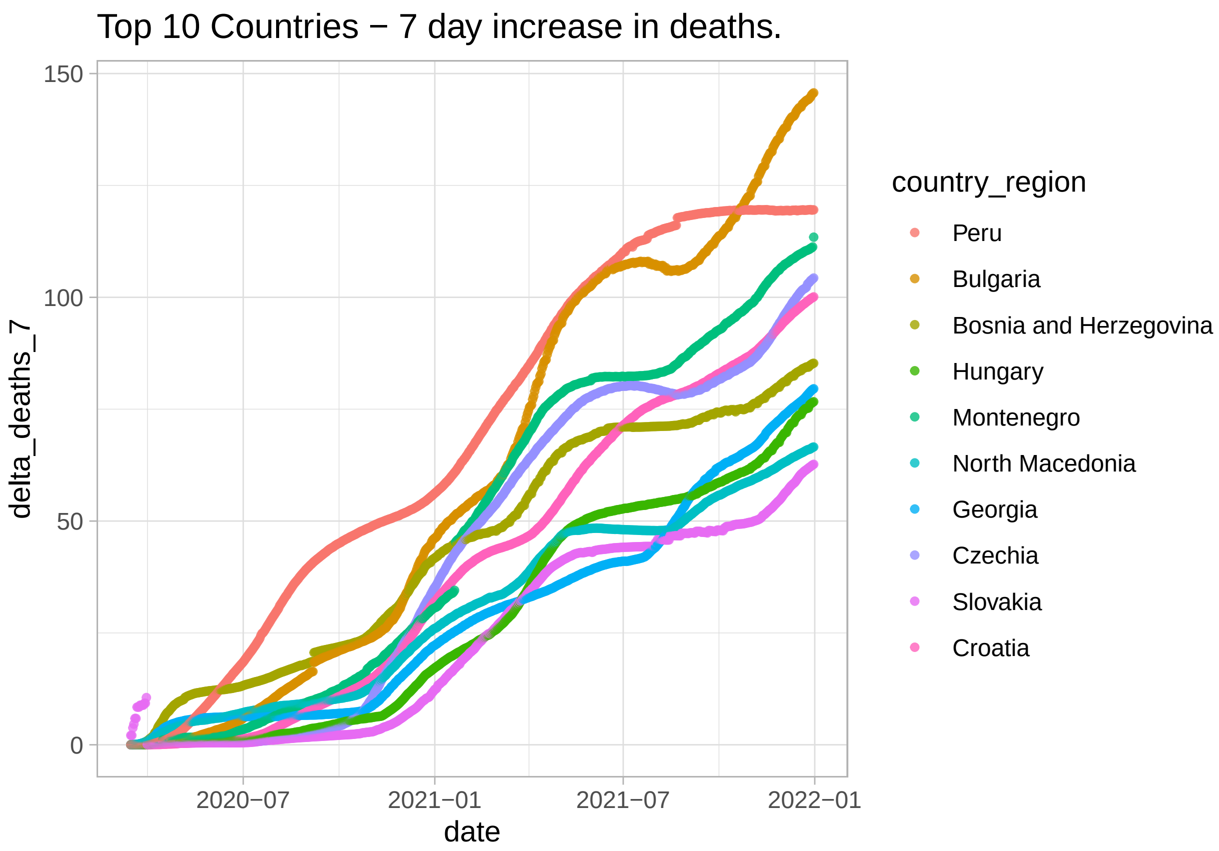
- **On a weekly basis, do we know if the cases are increasing or decreasing?**
- if the cases are on the rise, that means there need to be more restrictions put in place by local government and Health organizations.
  - Looking at the chart for top 10 countries in terms of cases, we can see that the cases are out of control.
  - Between 2021 June and end of 2021, we see that on weekly basis few countries on the top are reporting 4000+ cases/100k population.
  - This may indicate that the people in these countries are getting tested or they are getting admitted.



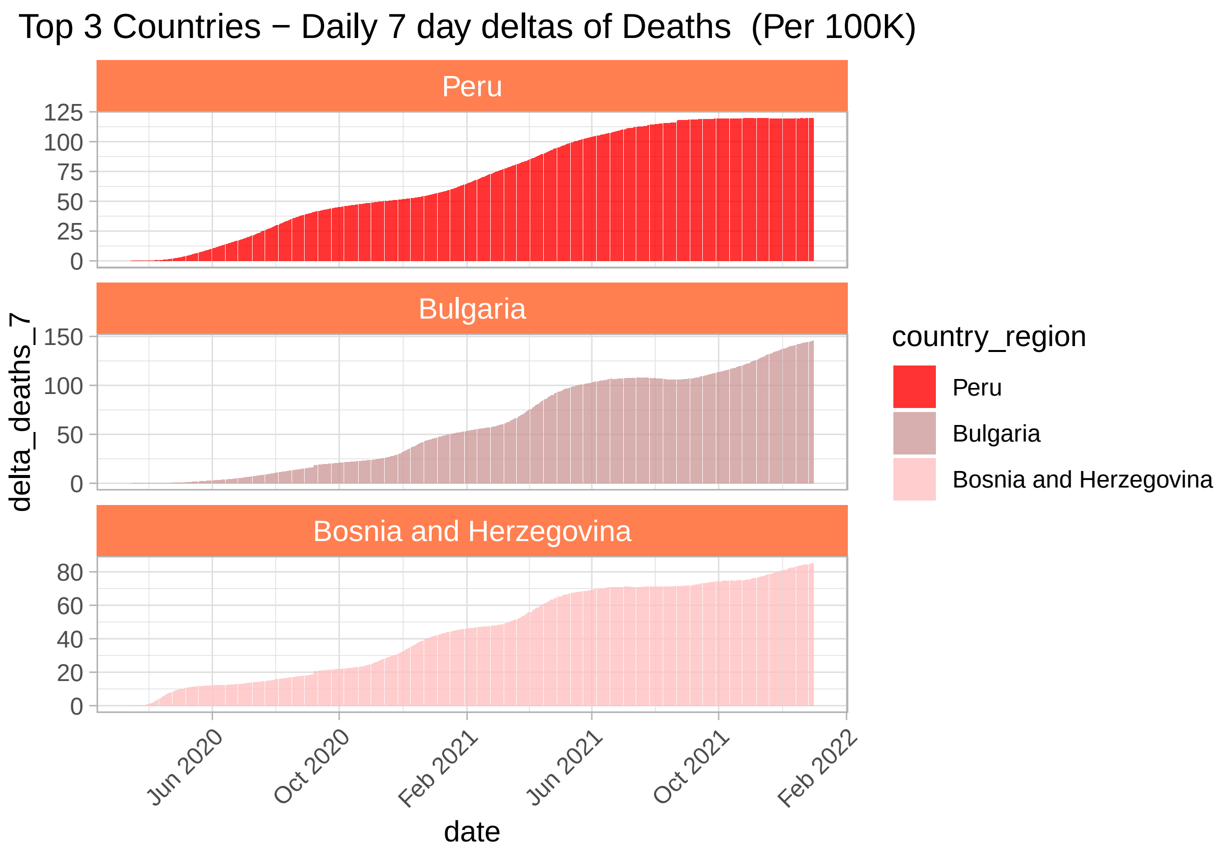
- When we just look at the top 3 countries with most cases, we can get the true picture.
- Certainly there are more than 4000 cases per 100k population.
- The Government needs to step in and impose certain rules that can help contain the virus.



- When we look at top 10 countries having most deaths, we notice that the pandemic is out of control here.
- The reasons could be that there are not many hospitals or health care providers or other reasons such as hygiene.
- The WHO organization, Red cross, etc. need to do something to help these countries.

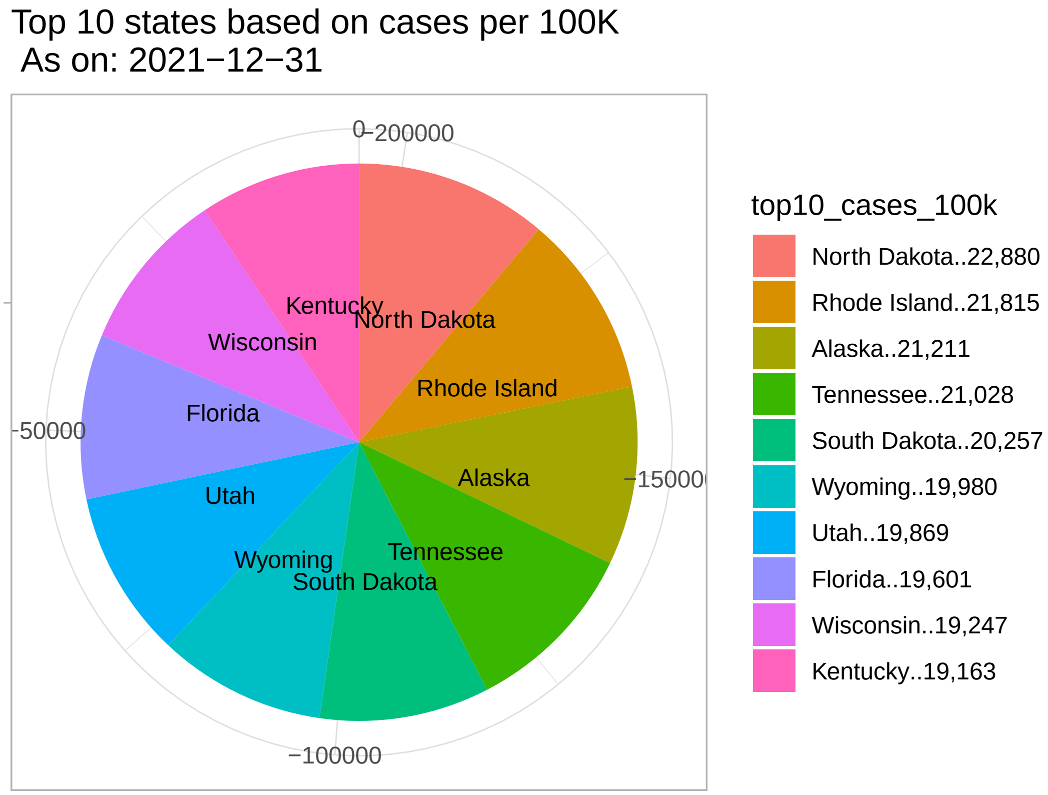


- **Is the average deaths per week in the Top3 Countries improving?**
  - A closer look at these Top 3 countries clearly shows that deaths might have reached a plateau.
  - The cases were rising for months.
  - What could be the reasons? Are there older population dying or is it general?
  - This insights should help WHO, Health Organizations and Medical professionals.

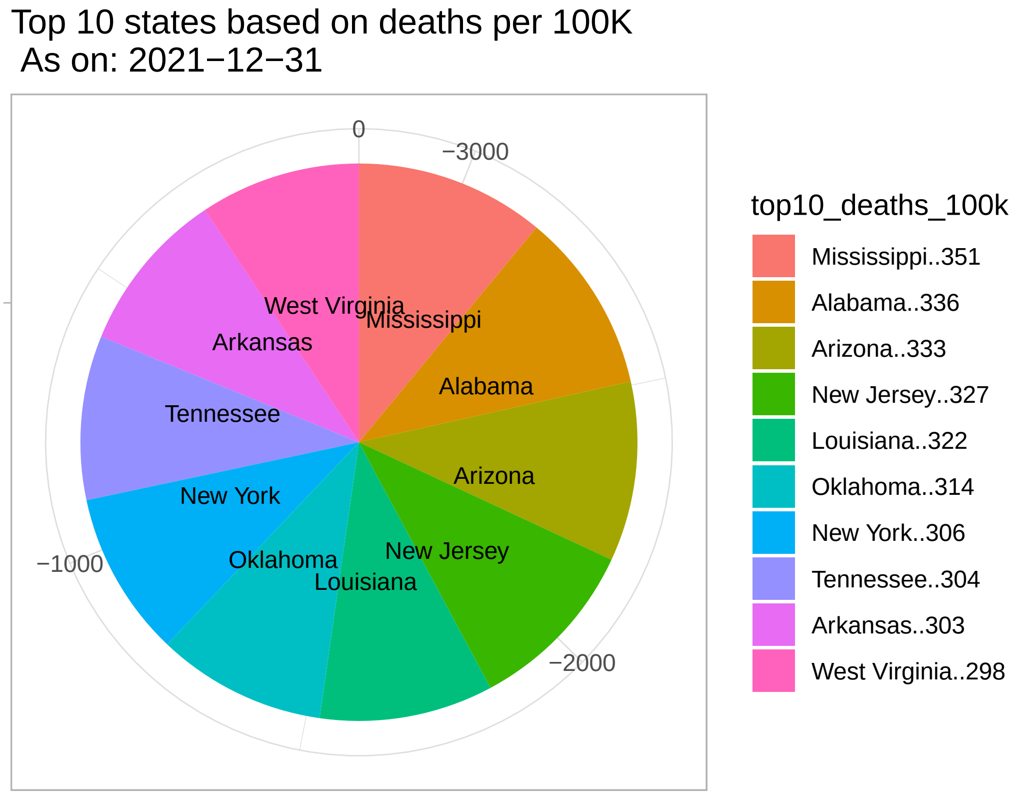


Section 4D: Uni-variate Analysis - Cases and Deaths per 100k population in USA as on 2021-12-31

- **How are the cases managed in USA?**
  - The Top 10 states which had the most cases per 100k population were:-

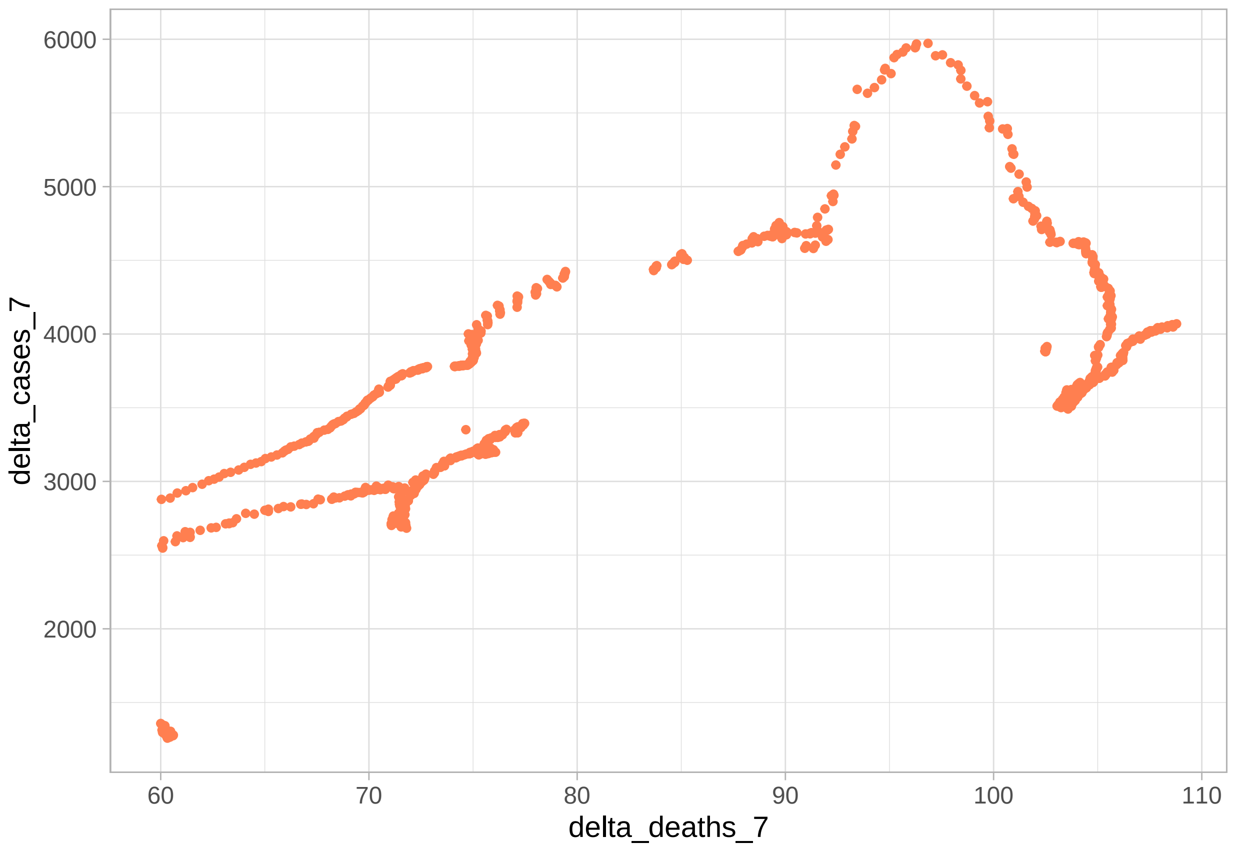


- The Top 10 states which had the most deaths per 100k population were:-



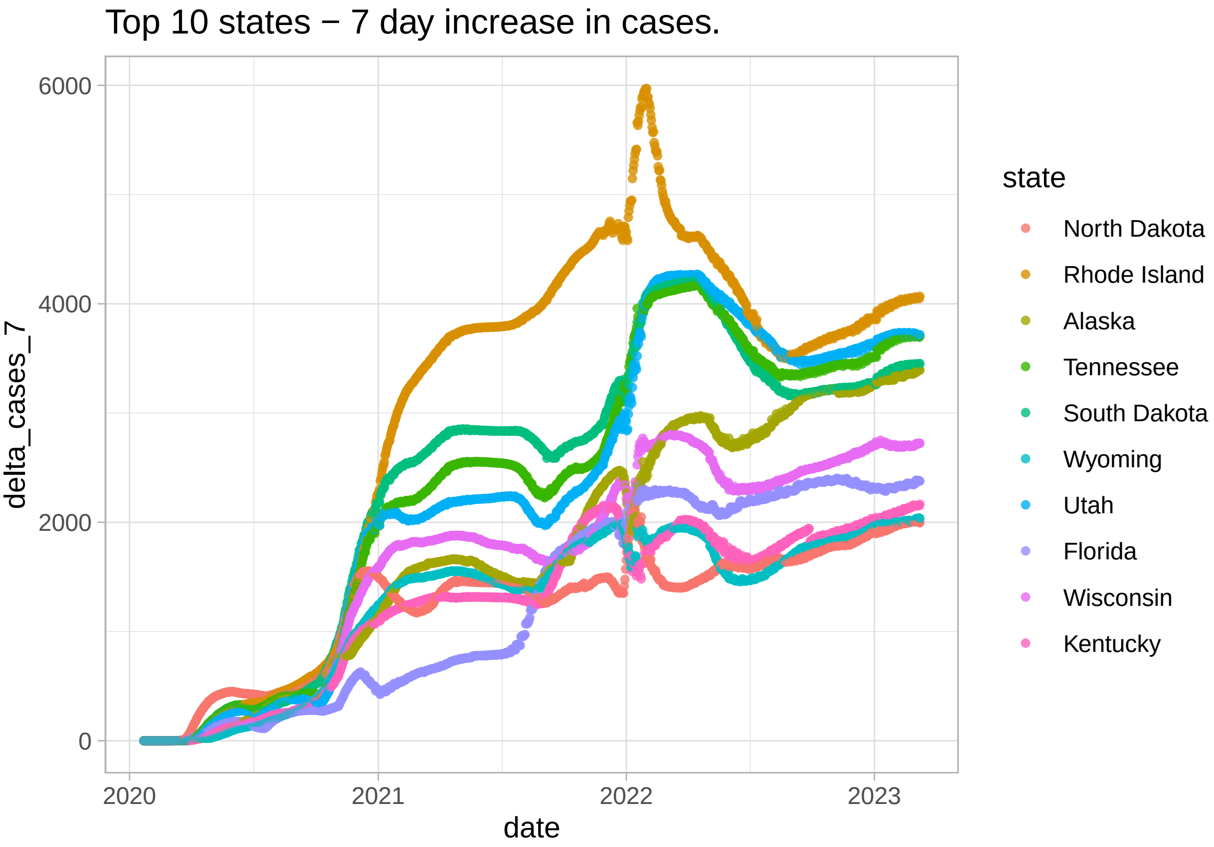
Section 4E: Bivariate Analysis - Cases and Deaths per 100k population for Top 3 states as on 2021-12-31

- **Is there a correlation between cases and deaths for the Top 3 states battling the pandemic?**
  - There is no strong correlation, however we can notice that as cases rises so do the deaths.

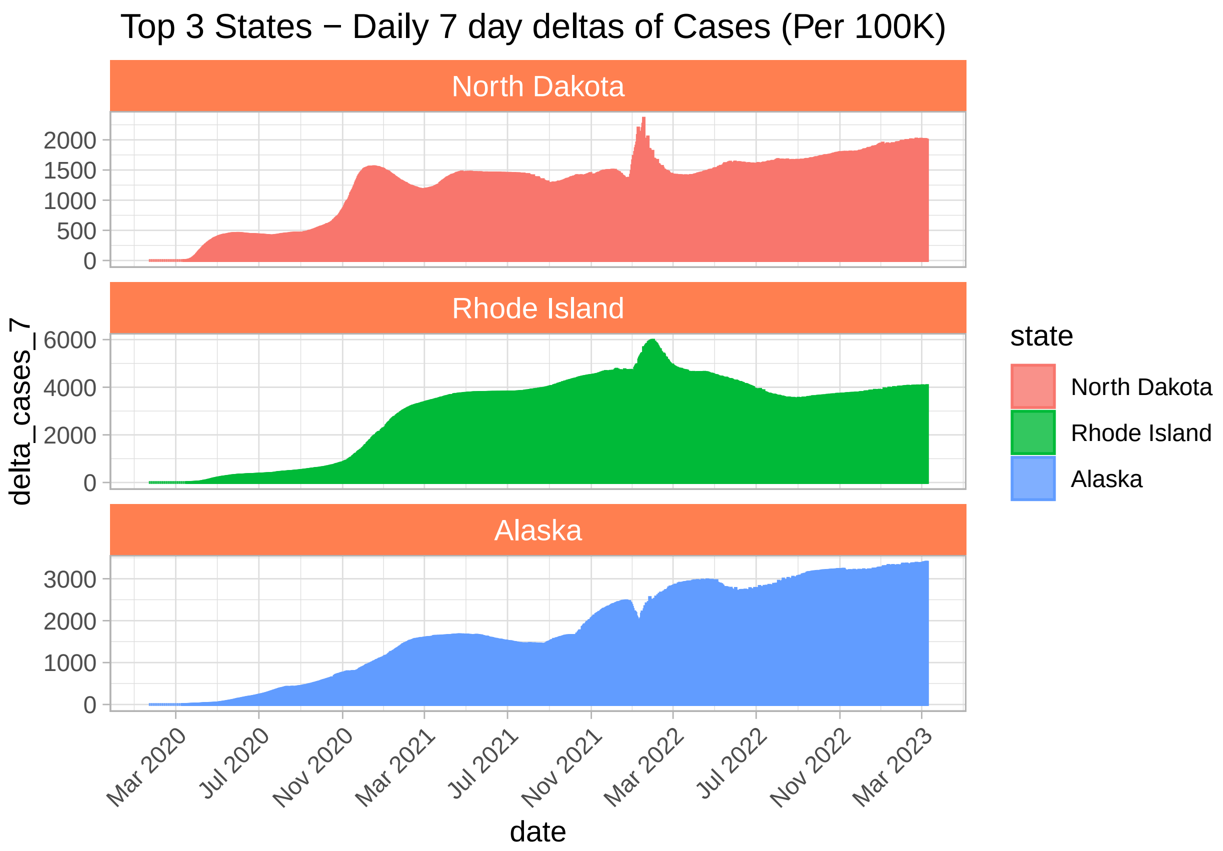


Section 4F: Multivariate Analysis - Understanding the trends in Cases and Deaths per 100k population as on 2021-12-31

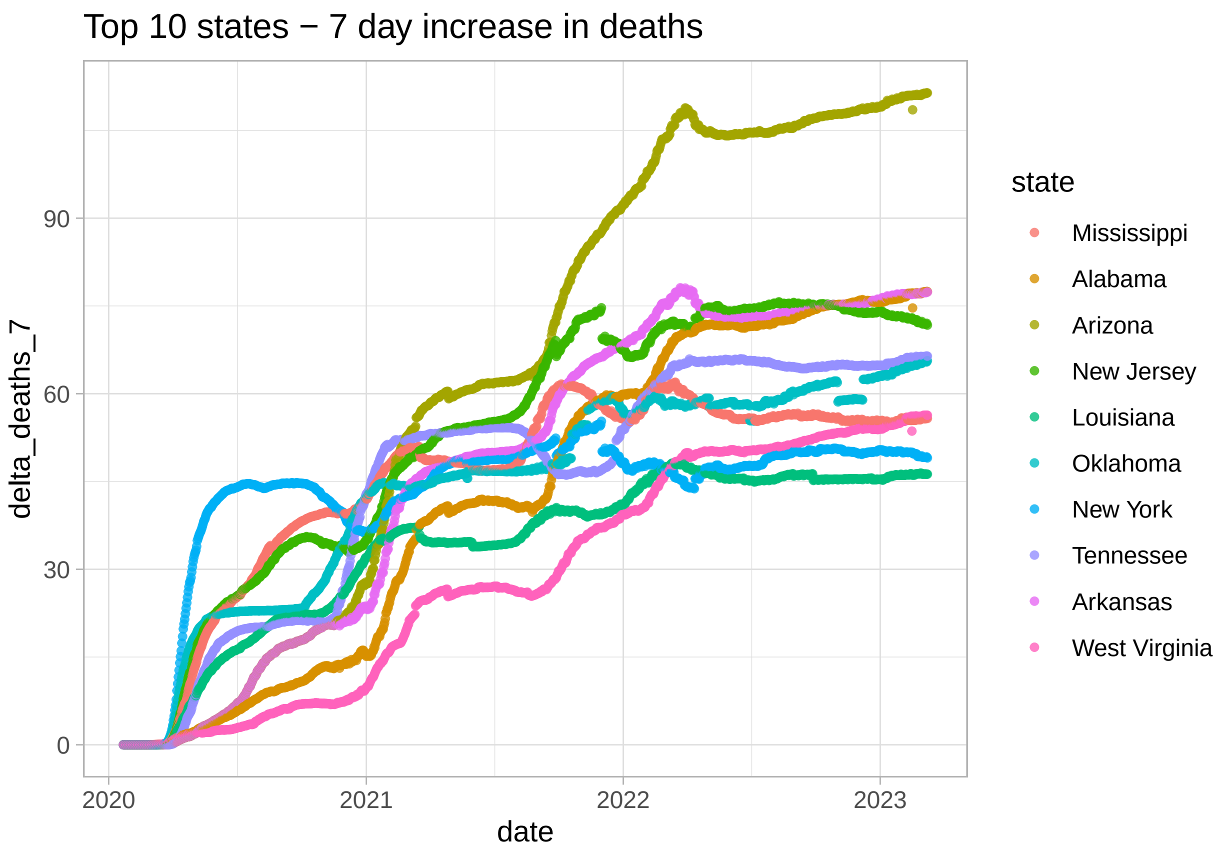
- **On a weekly basis, do we know if the cases are increasing or decreasing in the states of USA?**
  - The 7 day increase in cases for top 10 states reveals that the cases are on the rise.
  - The reason was shortage of Health care professional, shortage of health care facilities in USA
  - This insight will help the Government to control the situation.



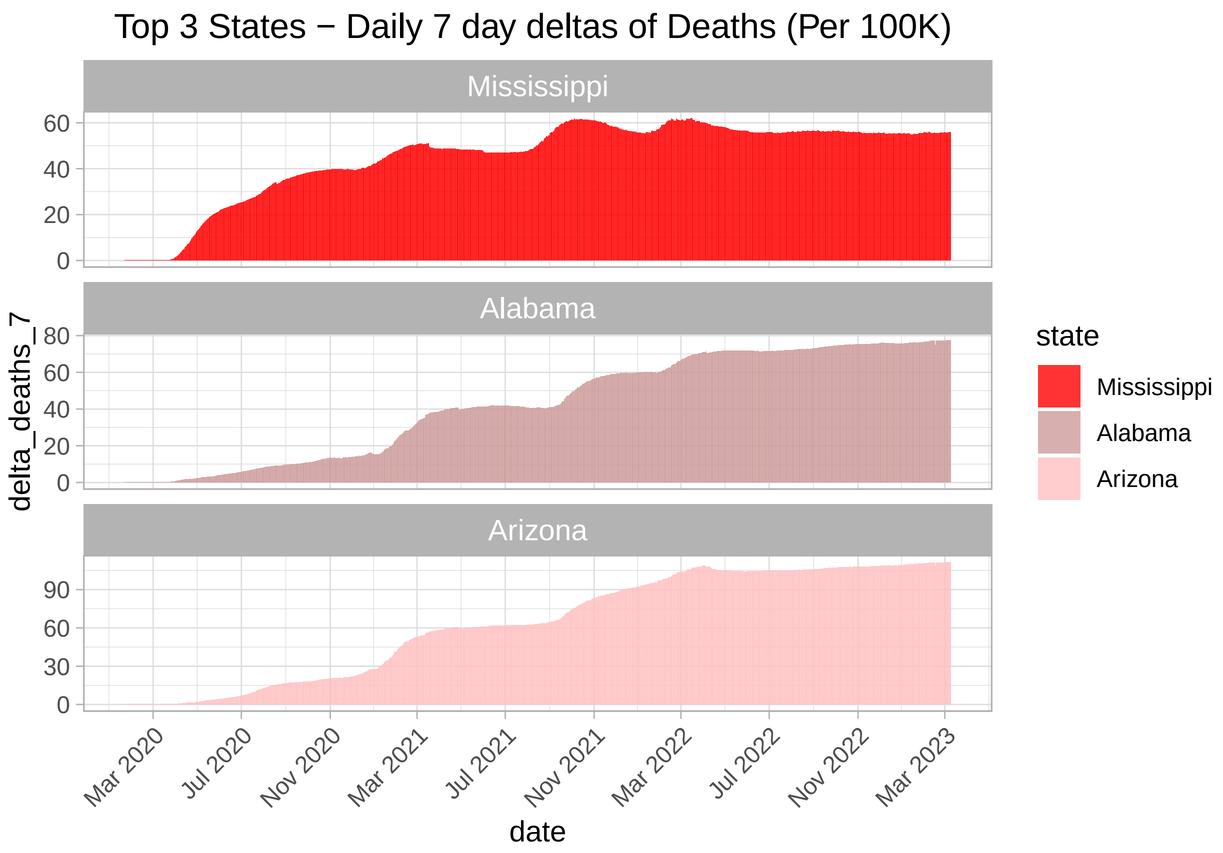
- A Closer look at the Top3 states with most cases clearly reveals that the states are not in control.
- There seems to be a plateau however peak point has not yet reached which might worry many stakeholders.



- **On a weekly basis, do we know if the deaths are increasing or decreasing in the states of USA?**
  - With the increase in cases, we also see a increase in deaths on a weekly basis.
  - The reasons could be that the health facilities are getting flooded with the influx in cases. The people who needed urgent care are not able to get to the doctors on time.
  - There might be shortage of medical staff
  - The patients who are dying may be immune compromised or senior citizens.
  - More data annd more analysis is needed to find details.

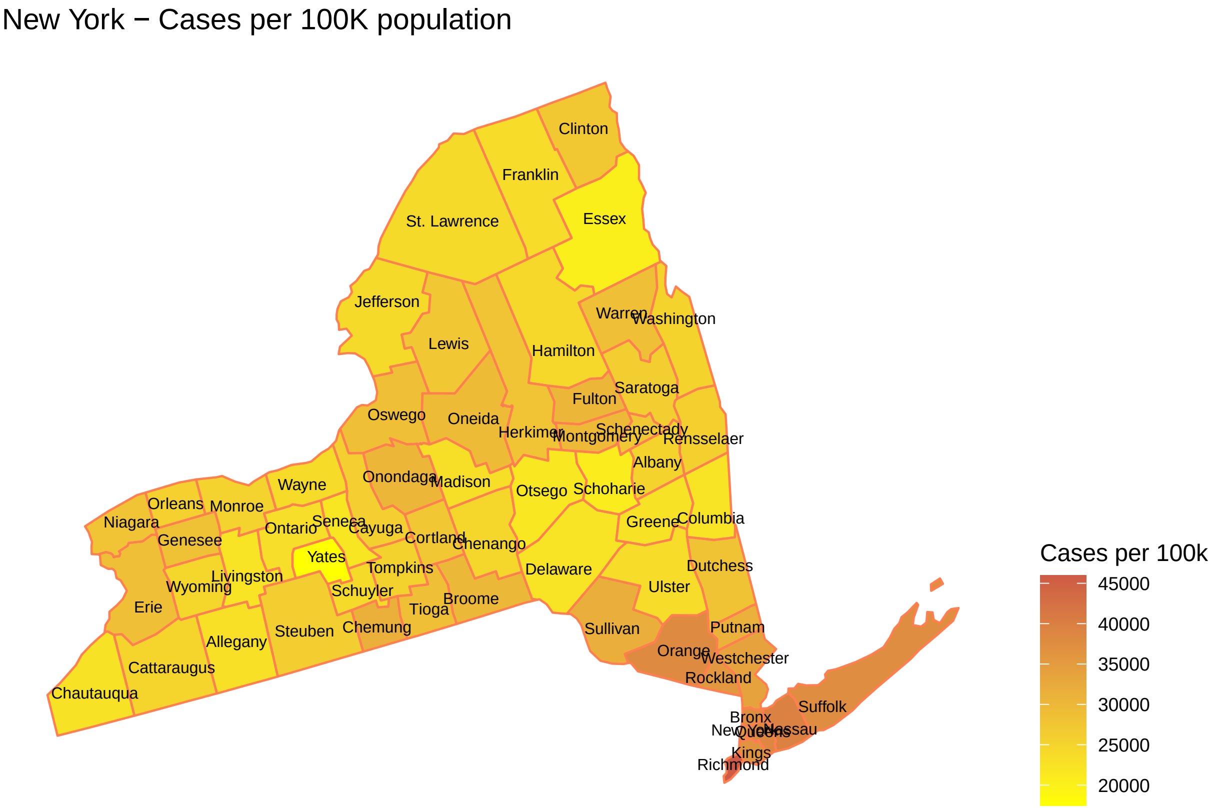


- Clearly, the top 3 states with most cases are not able to contain the 7 day increase in deaths.
- These states will need help from Government.



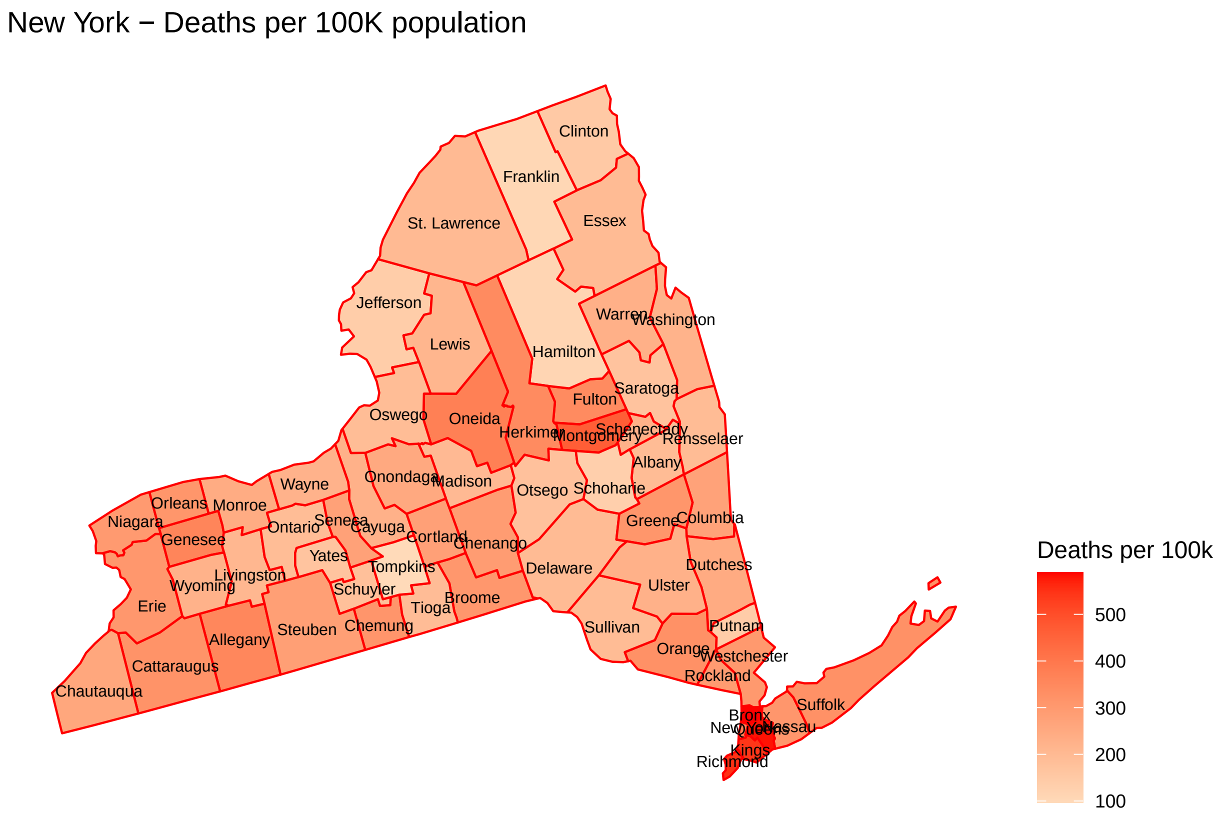
Section 4G: Visualize Favorite State - Heat Maps of the favorite state to see Cases and Deaths per 100k population as on 2021-12-31

- **How are the counties in New York managing the pandemic and which counties needs more help?**
  - We notice that the counties with more than 40K are darker and need more help.
  - This info helps media, Health care officials and Government to focus on specific counties.



New York − Cases per 100K population

- **How are the counties in New York containing the rise in deaths and Which counties need more help?**
  - The counties that are dark red may need more help with controlling deaths.
  - Queens, Bronx and Richmond areas had more deaths.
  - We need to look at demographics in those locations to get further insights.



**Section 5: Build & Test Models to predict the no of cases and deaths in USA.

Section 5A: Prepare the model dataset for US.

- We will create csse_us_cases_deaths_mdl_df by
  - grouping by state, year and month and making these columns as factors
  - Aggregating the cases_per_100k, deaths_per_100k and population
  - filtering to remove null
- We will also create a very basic model df to compare with first mmodel.

```
csse_us_cases_deaths_mdl_df <- csse_us_cases_deaths_7d_df %>%
  mutate(yr = year(date), mn=month(date)) %>%
  mutate(yr = factor(yr), mn=factor(mn), state=factor(state)) %>%
  group_by(state, yr, mn) %>%
  summarize(max_deaths_per_100k = max(deaths_per_100k),
    max_cases_per_100k = max(cases_per_100k),
    max_population = max(population)) %>%
  ungroup() %>%
  filter(max_cases_per_100k > 0, max_population > 0, max_deaths_per_100k > 0)
```

```r
# Create a very basic model
csse_us_cases_deaths_basic_mdl_df <- csse_us_cases_deaths_7d_df %>%
  group_by(state) %>%
  summarize(max_deaths_per_100k = max(deaths_per_100k),
    max_cases_per_100k = max(cases_per_100k),
    max_population = max(population)) %>%
  ungroup() %>%
  filter(max_cases_per_100k > 0, max_population > 0, max_deaths_per_100k > 0)
```

Section 5B: Model Training - Train the model using US datasets

- Train the linear model for champion use case
- Train the linear model for basic use case

```r
csse_us_cases_deaths_mdl <- lm(max_deaths_per_100k ~
                               state + max_cases_per_100k + yr + mn,
                               data = csse_us_cases_deaths_mdl_df)

# train the basic model
csse_us_cases_deaths_basic_mdl <- lm(max_deaths_per_100k ~ max_cases_per_100k,
                               data = csse_us_cases_deaths_basic_mdl_df)
```

- ** Observations:**
  - The Linear regression formula used is listed below,
  - Since we have multiple predictor variables, we have mutliple intercepts.
  - Each intercept will have Estimate, Std. Error.
  - The lower the error the more better our predictions will be.
  - Hence by minimizing the loss function, we can improve the predictions.

```r
# skip summary of basic model
#summary(csse_us_cases_deaths_basic_mdl)

# Formula is
summary(csse_us_cases_deaths_mdl)[1]
```

```
## $call
## lm(formula = max_deaths_per_100k ~ state + max_cases_per_100k +
##     yr + mn, data = csse_us_cases_deaths_mdl_df)
```

```r
# summarize the champion model
summary(csse_us_cases_deaths_mdl$coefficients)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -198.379  -92.145  -36.658  -42.833    5.614   83.341
```

Section 5C: Model Prediction - Do the predictions for US states and visualize the predictions.

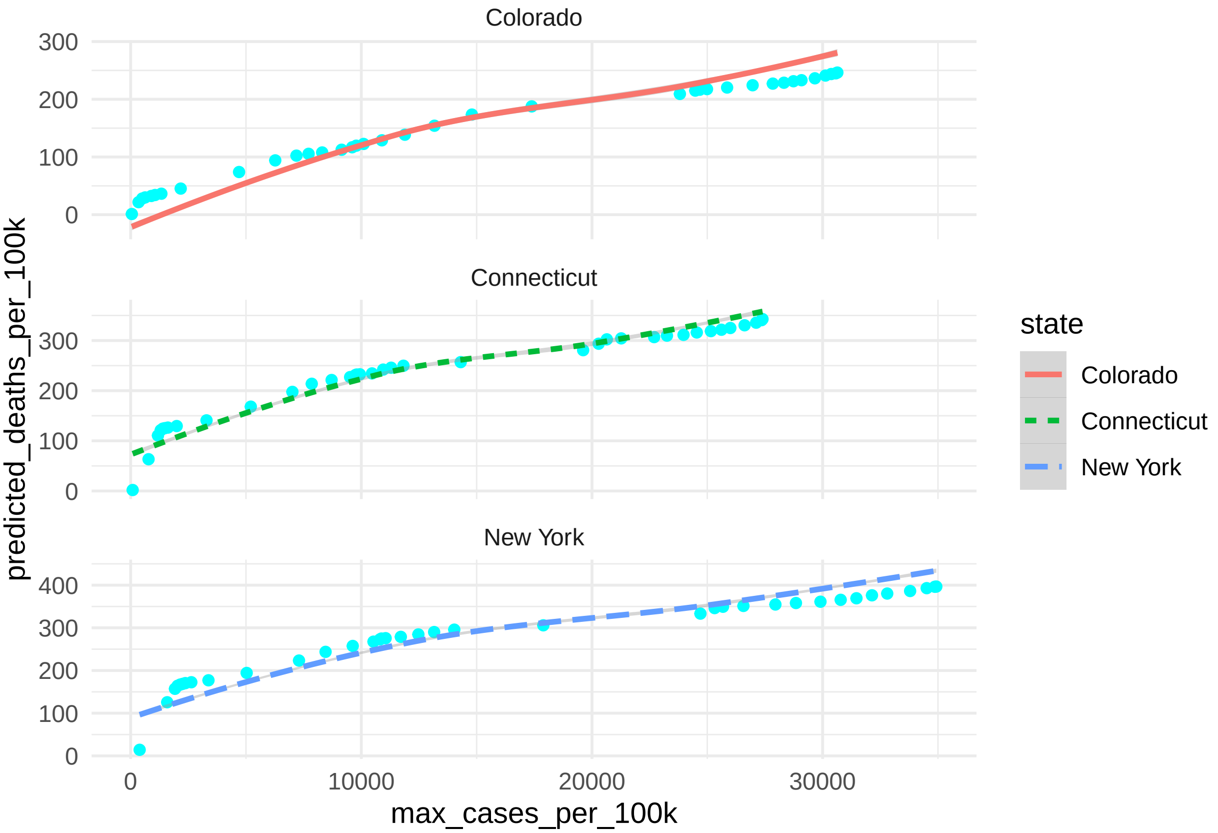
- Invoke the predict function using our training dataset
- Prepare Visualization for actual vs predicted values.

```r
# run predictions on champion model
csse_us_cases_deaths_pred_df <- csse_us_cases_deaths_mdl_df %>%
  mutate(predicted_deaths_per_100k = predict(csse_us_cases_deaths_mdl))

#run predictions on basic model
csse_us_cases_deaths_basic_pred_df <- csse_us_cases_deaths_basic_mdl_df %>%
  mutate(predicted_deaths_per_100k = predict(csse_us_cases_deaths_basic_mdl))

#Prepare Visualization for champion model
csse_us_cases_deaths_pred_vz <- csse_us_cases_deaths_pred_df %>%
  filter(state %in% c("New York", "Colorado", "Connecticut")) %>%
  ggplot(aes(x = max_cases_per_100k, y = predicted_deaths_per_100k, color=state)) +
  facet_wrap(~state, scales="free_y", nrow=3) +
  geom_point(aes(x = max_cases_per_100k, y = max_deaths_per_100k), color = "cyan") +
  geom_smooth(aes(linetype=state))  +
  theme_minimal()

#Prepare Visualization for basic model
csse_us_cases_deaths_basic_pred_vz <- csse_us_cases_deaths_basic_pred_df %>%
  ggplot() +
  geom_point(aes(x = max_cases_per_100k, y = predicted_deaths_per_100k), colour="blue") +
  geom_point(aes(x = max_cases_per_100k, y = max_deaths_per_100k), color = "coral") +
  theme_minimal()
```

Section 5D: Champion Model - Visualize the Predictions

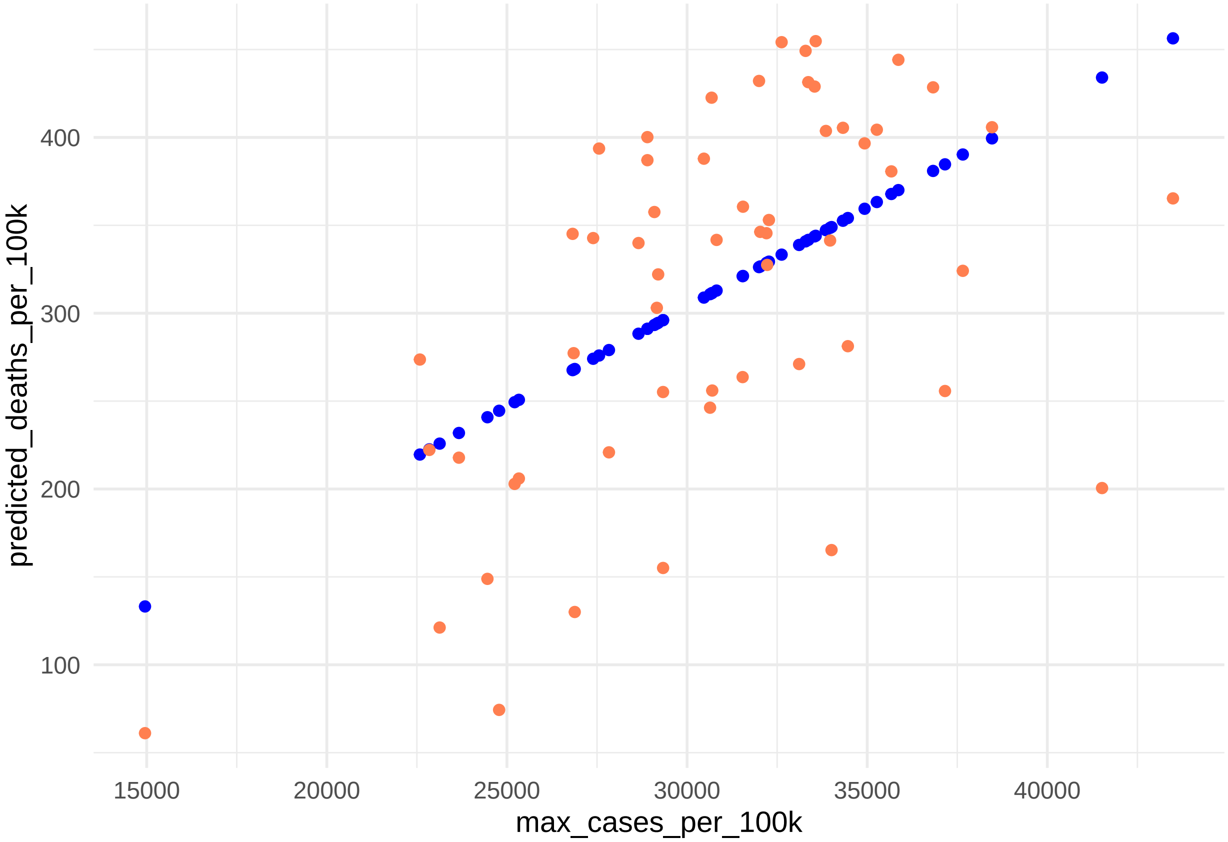
- Visualize the Predictions for the champion model
  - The actual values are in cyan
  - The predicted values for each state is a colored line.
  - From this report we can see how close our predictions are. This is because we used state, year and month as our predictor variables.

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Section 5E: Basic Model - Visualize the Predictions

- Visualize the predictions for the Basic model
  - Note there is a big gap between actual and predicted values.
  - This is because we chose very few predictor variables in the basic model



Section 6: Biases

- Professor Jane Wall rightly mentioned that Biases are natural. Everyone falls into their trap.
- **Emotional Biases**: COVID-19 itself was emotionally intense. There were many families across the world who lost a loved one or a friend. We want to be very careful in presenting our analysis. Our objective is to prepare a report that will help the right agencies to control the situation, hence we should not express our emotions in our reporting.
- **Data is gold**: We want to always trust data and use data to make inferences. Doing so we will earn trust.
- **Confirmation biases**: We are familiar with COVID019. Thus we can make mistakes if we become overconfident. Our analysis should be purely based in facts in the data.
- **Selection biases**: Quality of information - Who and how the data was collected? Several countries were blamed for not reporting the information on time, several Government officials were fired since they failed to report important info. From all these historic info, we can be certain that Selection bias existed during the data collection process. We received this data from trusted source, but the origin had some selection bias.
- **Anchoring biases**: We react to the first piece of information.

Section 7A: Conclusion

- In this project, we followed the Iterative steps in the Data Science process (DTSA-530) and analyzed the different countries and reported observations for the top 3 countries.
- For the countries, we narrowed our analysis to a small window (2020-03-15 and 2021-12-31) to understand the details during the peak pandemic period.
- We looked at US states and reported observations for the top 3 states followed by sharing the cases and deaths in one of the favorite states **(New York)**. For the top 3 USA states, from March 2022, we noticed a plateau in the graph for deaths per 100k.
- This report clearly indicated that the different countries in the World and the different states in USA were not able to control the rise in cases and deaths during the **peak COVID-19** pandemic period.
- We built two simple linear models - Using the predictor variables as state, max-cases-per-100k-population, year and month, we were able to train a linear model and predict the target variable max-deaths-per-100k-population with good accuracy.
- These Quantitative models will help WHO, Health Organizations and Government to control cases and deaths.
- Furthermore, if we had the demographic data, the facilities data, and other local data, we could have done much better analysis to generate insights that could help key stakeholders.

Section 7B: Session Information

- This will help the reader to understand the packages used.

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22621)
## 
## 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/New_York
## tzcode source: internal
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] usmap_0.6.2     lubridate_1.9.2 forcats_1.0.0   stringr_1.5.0  
##  [5] dplyr_1.1.2     purrr_1.0.2     readr_2.1.4     tidyr_1.3.0    
##  [9] tibble_3.2.1    ggplot2_3.4.3   tidyverse_2.0.0
## 
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.3        generics_0.1.3    lattice_0.21-8    stringi_1.7.12   
##  [5] hms_1.1.3         digest_0.6.33     magrittr_2.0.3    evaluate_0.21    
##  [9] grid_4.3.1        timechange_0.2.0  fastmap_1.1.1     Matrix_1.5-4.1   
## [13] mgcv_1.8-42       fansi_1.0.4       scales_1.2.1      cli_3.6.1        
## [17] rlang_1.1.1       crayon_1.5.2      splines_4.3.1     bit64_4.0.5
## [21] munsell_0.5.0     withr_2.5.0       yaml_2.3.7        tools_4.3.1
## [25] parallel_4.3.1    tzdb_0.4.0        usmapdata_0.1.0   colorspace_2.1-0
## [29] vctrs_0.6.3       R6_2.5.1          lifecycle_1.0.3   bit_4.0.5
## [33] vroom_1.6.3       pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.3
## [37] glue_1.6.2        xfun_0.40         tidyselect_1.2.0  highr_0.10
## [41] rstudioapi_0.15.0 knitr_1.43        farver_2.1.1      nlme_3.1-162
## [45] htmltools_0.5.6   rmarkdown_2.24    labeling_0.4.2    compiler_4.3.1
```