This data contains daily time series summary tables, including confirmed, deaths and recovered of covid 19. All data is read in from the daily case report.

Two time series tables are for the US confirmed cases and deaths, reported at the county level. They are named time_series_covid19_confirmed_US.csv, time_series_covid19_deaths_US.csv, respectively.

Three time series tables are for the global confirmed cases, recovered cases and deaths. Australia, Canada and China are reported at the province/state level. Dependencies of the Netherlands, the UK, France and Denmark are listed under the province/state level. The US and other countries are at the country level. The tables are renamed time_series_covid19_confirmed_global.csv and time_series_covid19_deaths_global.csv, and time_series_covid19_recovered_global.csv, respectively.

The source of COVID-19 data belongs to the Johns Hopkins University website and is downloaded from https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

Step 1: Import data

```
#install.packages("tidyverse")
```

```
# install libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
# create urls for data
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi
file_names <- c("time_series_covid19_confirmed_global.csv", "time_series_covid19_deaths_global.csv", "ti
urls <- str_c(url_in, file_names)
urls
```

```
## [1] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
## [2] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
## [3] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
## [4] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
```

```
# read time series covid19 global cases and view some first rows
global_cases <- read_csv(urls[1])
```

```
## Rows: 280 Columns: 737
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (735): 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.
```

```
head(global_cases)
```

```
## # A tibble: 6 x 737
##   ‘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
## 6 <NA>             Antigua and Barbu~  17.1 -61.8          0         0         0
## # ... with 730 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>, 2/21/20 <dbl>,
## #   2/22/20 <dbl>, 2/23/20 <dbl>, 2/24/20 <dbl>, 2/25/20 <dbl>, ...
```

```
# read time series covid19 global deaths and view some first rows
global_deaths <- read_csv(urls[2])
```

```
## Rows: 280 Columns: 737
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (735): 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.
```

```
head(global_deaths)
```

```
## # A tibble: 6 x 737
##   ‘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
## 6 <NA>            Antigua and Barbu~  17.1 -61.8          0         0         0
## # ... with 730 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>, 2/21/20 <dbl>,
## #   2/22/20 <dbl>, 2/23/20 <dbl>, 2/24/20 <dbl>, 2/25/20 <dbl>, ...
```

```
# read time series covid19 us cases and view some first rows
us_cases <- read_csv(urls[3])
```

```
## Rows: 3342 Columns: 744
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (738): 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.
```

```
head(us_cases)
```

```
## # A tibble: 6 x 744
##        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
## 6 84001011 US    USA     840  1011 Bullock Alabama        US              32.1
## # ... with 735 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>, 2/16/20 <dbl>,
## #   2/17/20 <dbl>, 2/18/20 <dbl>, 2/19/20 <dbl>, 2/20/20 <dbl>, ...
```

```
# read time series covid19 us deaths and view some first rows
us_deaths <- read_csv(urls[4])
```

```
## Rows: 3342 Columns: 745
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (739): 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.
```

```
head(us_deaths)
```

```
## # A tibble: 6 x 745
##        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
## 6 84001011 US    USA     840  1011 Bullock Alabama        US              32.1
## # ... with 736 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>, 2/15/20 <dbl>, 2/16/20 <dbl>, 2/17/20 <dbl>, ...
```

Step 2: Tidy and Transform Data

```
# remove unused columns of the us_cases and convert date from column to row
us_cases <- us_cases %>% pivot_longer(cols = -(UID:Combined_Key), names_to = "date", values_to = "cases
  select(Admin2:cases) %>% mutate(date = mdy(date)) %>% select(-c(Lat, Long_))
head(us_cases)
```

```
## # A tibble: 6 x 6
##   Admin2  Province_State Country_Region Combined_Key          date       cases
##   <chr>   <chr>          <chr>          <chr>                 <date>     <dbl>
## 1 Autauga Alabama        US             Autauga, Alabama, US 2020-01-22     0
## 2 Autauga Alabama        US             Autauga, Alabama, US 2020-01-23     0
## 3 Autauga Alabama        US             Autauga, Alabama, US 2020-01-24     0
## 4 Autauga Alabama        US             Autauga, Alabama, US 2020-01-25     0
## 5 Autauga Alabama        US             Autauga, Alabama, US 2020-01-26     0
## 6 Autauga Alabama        US             Autauga, Alabama, US 2020-01-27     0
```

```
# remove unused columns of the us_deaths and convert date from column to row
us_deaths <- us_deaths %>% pivot_longer(cols = -(UID:Population), names_to = "date", values_to = "death
  select(Admin2:deaths) %>% mutate(date = mdy(date)) %>% select(-c(Lat, Long_))
head(us_deaths)
```

```
## # A tibble: 6 x 7
##   Admin2 Province_State Country_Region Combined_Key Population date       deaths
##   <chr>  <chr>          <chr>          <chr>             <dbl> <date>      <dbl>
## 1 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-22      0
## 2 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-23      0
## 3 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-24      0
## 4 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-25      0
## 5 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-26      0
## 6 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-27      0
```

```
# create us by full join us deaths and us cases
US <- us_cases %>% full_join(us_deaths)
```

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

```
head(US)
```

```
## # A tibble: 6 x 8
##   Admin2  Province_State Country_Region Combined_Key date       cases Population
##   <chr>   <chr>          <chr>          <chr>        <date>     <dbl>      <dbl>
## 1 Autauga Alabama        US             Autauga, Al~ 2020-01-22     0      55869
## 2 Autauga Alabama        US             Autauga, Al~ 2020-01-23     0      55869
## 3 Autauga Alabama        US             Autauga, Al~ 2020-01-24     0      55869
## 4 Autauga Alabama        US             Autauga, Al~ 2020-01-25     0      55869
## 5 Autauga Alabama        US             Autauga, Al~ 2020-01-26     0      55869
## 6 Autauga Alabama        US             Autauga, Al~ 2020-01-27     0      55869
## # ... with 1 more variable: deaths <dbl>
```

```
# summary us data
US <- US %>% filter(cases > 0)
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:2142950     Length:2142950     Length:2142950     Length:2142950
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases           Population           deaths
##  Min.   :2020-01-22   Min.   :      1   Min.   :       0   Min.   :    0.0
##  1st Qu.:2020-09-15   1st Qu.:    272   1st Qu.:   11164   1st Qu.:    3.0
##  Median :2021-02-27   Median :   1365   Median :   26586   Median :   25.0
##  Mean   :2021-02-26   Mean   :   7672   Mean   :  105915   Mean   :  135.4
##  3rd Qu.:2021-08-11   3rd Qu.:   4559   3rd Qu.:   69473   3rd Qu.:   82.0
##  Max.   :2022-01-23   Max.   :2494097   Max.   :10039107   Max.   :28480.0
```

```
# Quick glimpse data also tells us the number of rows (observations), columns (variables) and type of d
glimpse(US)
```

```
## Rows: 2,142,950
## Columns: 8
## $ Admin2         <chr> "Autauga", "Autauga", "Autauga", "Autauga", "Autauga", ~
## $ Province_State <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama", ~
## $ Country_Region <chr> "US", "US", "US", "US", "US", "US", "US", "US", "US", "~
## $ Combined_Key   <chr> "Autauga, Alabama, US", "Autauga, Alabama, US", "Autaug~
## $ date           <date> 2020-03-24, 2020-03-25, 2020-03-26, 2020-03-27, 2020-0~
## $ cases          <dbl> 1, 5, 6, 6, 6, 6, 8, 8, 10, 12, 12, 12, 12, 12, 12, 12,~
## $ Population     <dbl> 55869, 55869, 55869, 55869, 55869, 55869, 55869, 55869,~
## $ deaths         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1~
```

```
# check missing values
sapply(US,function(x) sum(is.na(x)))
```

```
##         Admin2 Province_State Country_Region   Combined_Key           date
##           3490              0              0              0              0
##          cases     Population         deaths
##              0              0              0
```

```
# remove unused columns of the global_cases and convert date from column to row
global_cases <- global_cases %>% pivot_longer(cols = -c('Province/State', 'Country/Region', Lat, Long),
  select(-c(Lat,Long))
```

```
head(global_cases)
```

```
## # A tibble: 6 x 4
##   ‘Province/State‘ ‘Country/Region‘ date    cases
##   <chr>            <chr>            <chr>   <dbl>
## 1 <NA>             Afghanistan      1/22/20     0
## 2 <NA>             Afghanistan      1/23/20     0
## 3 <NA>             Afghanistan      1/24/20     0
## 4 <NA>             Afghanistan      1/25/20     0
## 5 <NA>             Afghanistan      1/26/20     0
## 6 <NA>             Afghanistan      1/27/20     0
```

```
# remove unused columns of the global_deaths and convert date from column to row
global_deaths <- global_deaths %>% pivot_longer(cols = -c('Province/State', 'Country/Region', Lat, Long
  select(-c(Lat,Long))
head(global_deaths)
```

```
## # A tibble: 6 x 4
##   ‘Province/State‘ ‘Country/Region‘ date    deaths
##   <chr>            <chr>            <chr>    <dbl>
## 1 <NA>             Afghanistan      1/22/20      0
## 2 <NA>             Afghanistan      1/23/20      0
## 3 <NA>             Afghanistan      1/24/20      0
## 4 <NA>             Afghanistan      1/25/20      0
## 5 <NA>             Afghanistan      1/26/20      0
## 6 <NA>             Afghanistan      1/27/20      0
```

```
# create global by full join global deaths and global cases
global <- global_cases %>% full_join(global_deaths) %>% rename(Country_Region = 'Country/Region', Provin
```

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

```
head(global)
```

```
## # A tibble: 6 x 5
##   Province_State Country_Region date       cases deaths
##   <chr>          <chr>          <date>     <dbl>  <dbl>
## 1 <NA>           Afghanistan    2020-01-22     0      0
## 2 <NA>           Afghanistan    2020-01-23     0      0
## 3 <NA>           Afghanistan    2020-01-24     0      0
## 4 <NA>           Afghanistan    2020-01-25     0      0
## 5 <NA>           Afghanistan    2020-01-26     0      0
## 6 <NA>           Afghanistan    2020-01-27     0      0
```

```
# add a variable called combined_key that combines Province state and Country region into the global
global <- global %>% unite("Combined_Key", c(Province_State, Country_Region), sep = ", ", na.rm = TRUE,
head(global)
```

```
## # A tibble: 6 x 6
##   Combined_Key Province_State Country_Region date       cases deaths
##   <chr>        <chr>          <chr>          <date>     <dbl>  <dbl>
## 1 Afghanistan  <NA>           Afghanistan    2020-01-22     0      0
## 2 Afghanistan  <NA>           Afghanistan    2020-01-23     0      0
## 3 Afghanistan  <NA>           Afghanistan    2020-01-24     0      0
## 4 Afghanistan  <NA>           Afghanistan    2020-01-25     0      0
## 5 Afghanistan  <NA>           Afghanistan    2020-01-26     0      0
## 6 Afghanistan  <NA>           Afghanistan    2020-01-27     0      0
```

```
# summary global data
global <- global %>% filter(cases > 0)
summary(global)
```

```
##  Combined_Key       Province_State     Country_Region          date           
##  Length:188333      Length:188333      Length:188333      Min.   :2020-01-22  
##  Class :character   Class :character   Class :character   1st Qu.:2020-08-25  
##  Mode  :character   Mode  :character   Mode  :character   Median :2021-02-16  
##                                                           Mean   :2021-02-13  
##                                                           3rd Qu.:2021-08-07  
##                                                           Max.   :2022-01-23  
##      cases               deaths      
##  Min.   :       1   Min.   :     0  
##  1st Qu.:     531   1st Qu.:     4  
##  Median :    6489   Median :    98  
##  Mean   :  435755   Mean   :  9388  
##  3rd Qu.:  108677   3rd Qu.:  1878  
##  Max.   :70700678   Max.   :866540  
```

```
# add a population to the global data by getting information from a csv file of Johns Hopkins website
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: 4215 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.
```

```
# add population to the global data
global <- global %>% left_join(uid, by = c("Province_State", "Country_Region")) %>% select(-c(UID, FIPS
head(global)
```

```
## # A tibble: 6 x 7
##   Province_State Country_Region date       cases deaths Population Combined_Key
##   <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl> <chr>       
## 1 <NA>           Afghanistan    2020-02-24     5      0   38928341 Afghanistan 
## 2 <NA>           Afghanistan    2020-02-25     5      0   38928341 Afghanistan 
## 3 <NA>           Afghanistan    2020-02-26     5      0   38928341 Afghanistan 
## 4 <NA>           Afghanistan    2020-02-27     5      0   38928341 Afghanistan 
## 5 <NA>           Afghanistan    2020-02-28     5      0   38928341 Afghanistan 
## 6 <NA>           Afghanistan    2020-02-29     5      0   38928341 Afghanistan 
```

```
# Quick glimpse data also tells us the number of rows (observations), columns (variables) and type of d
glimpse(global)
```

```
## Rows: 188,333
## Columns: 7
## $ Province_State <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ Country_Region <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanist~
## $ date           <date> 2020-02-24, 2020-02-25, 2020-02-26, 2020-02-27, 2020-0~
## $ cases          <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 8, 8, 8, 8, 11, 11,~
## $ deaths         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ Population     <dbl> 38928341, 38928341, 38928341, 38928341, 38928341, 38928~
## $ Combined_Key   <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanist~
```

```
# check missing values
sapply(global,function(x) sum(is.na(x)))
```

```
## Province_State Country_Region           date           cases          deaths
##         129785              0              0               0               0
##     Population   Combined_Key
##           2753              0
```

Step 3: Add Visualizations and Analysis

- Now, after cleaning up data, I'll analyze and visualize data.

Question 1: How many Cases and Deaths in US by year?

```
# create a table of US by year
us_by_year <- US %>%
  mutate(YEAR = format(as.Date(US$date, format="%Y/%m/%d"),"%Y")) %>%
  group_by(YEAR) %>%
  summarise(CASES = sum(cases), DEATHS = sum(deaths))
us_by_year
```

```
## # A tibble: 3 x 3
##   YEAR        CASES    DEATHS
##   <chr>       <dbl>     <dbl>
## 1 2020   1725975699  46610849
## 2 2021  13263556468 224187736
## 3 2022   1451640334  19438713
```

```
# Number of cases, deaths by year in US
us_number_of_cases_20 = us_by_year[us_by_year$YEAR == "2020", "CASES"]
us_number_of_deaths_20 = us_by_year[us_by_year$YEAR == "2020", "DEATHS"]
us_number_of_cases_21 = us_by_year[us_by_year$YEAR == "2021", "CASES"]
us_number_of_deaths_21 = us_by_year[us_by_year$YEAR == "2021", "DEATHS"]
us_number_of_cases_22 = us_by_year[us_by_year$YEAR == "2022", "CASES"]
us_number_of_deaths_22 = us_by_year[us_by_year$YEAR == "2022", "DEATHS"]
```

```
print(paste("The number of covid19 cases in 2020 was: ",us_number_of_cases_20,"."))
```

```
## [1] "The number of covid19 cases in 2020 was:  1725975699 ."
```

```
print(paste("The number of covid19 deaths in 2020 was: ",us_number_of_deaths_20,"."))
```

```
## [1] "The number of covid19 deaths in 2020 was:  46610849 ."
```

```
print(paste("The number of covid19 cases in 2021 was: ",us_number_of_cases_21,"."))
```

```
## [1] "The number of covid19 cases in 2021 was:  13263556468 ."
```

```
print(paste("The number of covid19 deaths in 2021 was: ",us_number_of_deaths_21,"."))
```

```
## [1] "The number of covid19 deaths in 2021 was:  224187736 ."
```

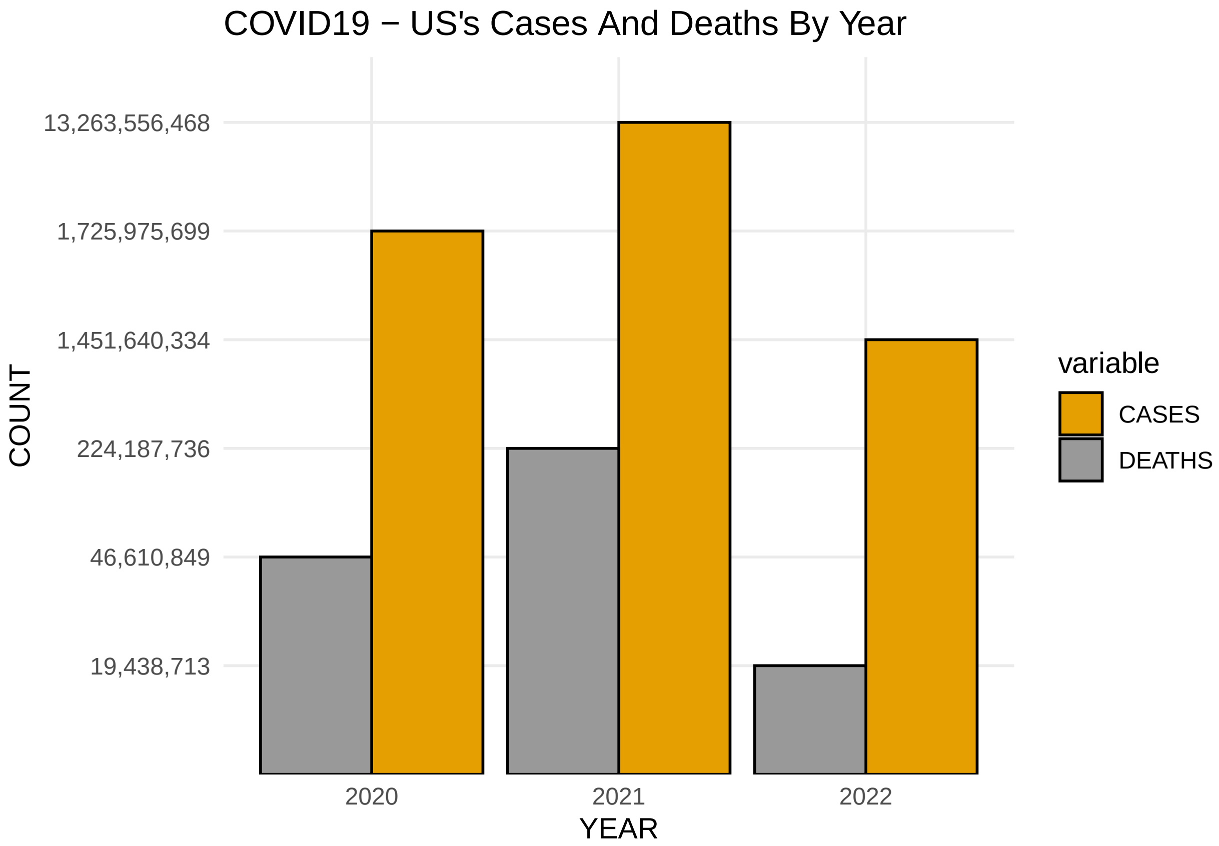
```
print(paste("The number of covid19 cases in 2022 was: ",us_number_of_cases_22,"."))
```

```
## [1] "The number of covid19 cases in 2022 was:  1451640334 ."
```

```
print(paste("The number of covid19 deaths in 2022 was: ",us_number_of_deaths_22,"."))
```

```
## [1] "The number of covid19 deaths in 2022 was:  19438713 ."
```

```
# plot the chart of US cases and deaths by year
ggplot(data=melt(us_by_year, id.vars=c("YEAR")), aes(x=YEAR, y=format(value, scientific = FALSE, big.ma
geom_bar(stat="identity", color="black", position=position_dodge())+ scale_y_discrete(name="COUNT") +
  theme_minimal() + scale_fill_manual(values=c('#E69F00', '#999999')) +
ggtitle("COVID19 - US's Cases And Deaths By Year")
```



As the plot above, we can see that, the most US covid19 cases and deaths were in 2021. The number of cases were increase 11,537,580,769 (from 1,725,975,699 to 13,263,556,468). The number of deaths were increase 177,576,887 (from 46,610,849 to 224,187,736). Because now is just the beginning of the year, the number of covid19 cases and deaths in 2022 were smaller than 2021 and 2020.

Question 2: How many cases and deaths in US by state?

```
# create a table of US by state
US_by_state <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Province_State, Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
  ungroup()
```

```
## ‘summarise()‘ has grouped output by ’Province_State’, ’Country_Region’. You can override using the ‘
```

```
head(US_by_state)
```

```
## # A tibble: 6 x 7
##   Province_State Country_Region date       cases deaths deaths_per_mill
##   <chr>          <chr>          <date>     <dbl>  <dbl>           <dbl>
## 1 Alabama        US             2020-03-11     3      0               0
## 2 Alabama        US             2020-03-12     4      0               0
## 3 Alabama        US             2020-03-13     8      0               0
## 4 Alabama        US             2020-03-14    15      0               0
## 5 Alabama        US             2020-03-15    28      0               0
## 6 Alabama        US             2020-03-16    36      0               0
## # ... with 1 more variable: Population <dbl>
```

```
# create a dataframe y with the catagories of US Province_State and the number of cases and deaths of e
y <- US_by_state %>% group_by(Province_State) %>% summarise(CASES = sum(cases), DEATHS = sum(deaths))%>%
mutate(DEATHS_PER_CASES_RATE = round(DEATHS / CASES * 100, 2)) %>%
select(Province_State, CASES, DEATHS, DEATHS_PER_CASES_RATE)
as.data.frame(y)
```

```
##               Province_State      CASES   DEATHS DEATHS_PER_CASES_RATE
## 1                    Alabama  283282881  5234333                  1.85
## 2                     Alaska   39060603   206245                  0.53
## 3             American Samoa        789        0                  0.00
## 4                    Arizona  432838923  8211182                  1.90
## 5                   Arkansas  178058372  2868740                  1.61
## 6                 California 1839596664 27149264                  1.48
## 7                   Colorado  259968431  3343542                  1.29
## 8                Connecticut  158088553  4314072                  2.73
## 9                   Delaware   51593423   849076                  1.65
## 10          Diamond Princess      33216        0                  0.00
## 11      District of Columbia   25268132   578316                  2.29
## 12                   Florida 1229501228 19411487                  1.58
## 13                   Georgia  569298366 10187037                  1.79
## 14            Grand Princess      69001     1979                  2.87
## 15                      Guam    5179095    77538                  1.50
## 16                    Hawaii   24421302   276712                  1.13
## 17                     Idaho   97503876  1108240                  1.14
## 18                  Illinois  665746663 12172977                  1.83
## 19                   Indiana  366127750  6407871                  1.75
## 20                      Iowa  183787370  2758698                  1.50
## 21                    Kansas  158153567  2285740                  1.45
## 22                  Kentucky  235910435  3214297                  1.36
## 23                 Louisiana  262845059  5748483                  2.19
## 24                     Maine   32007610   401659                  1.25
## 25                  Maryland  220305687  4593407                  2.09
## 26             Massachusetts  332112469  9129234                  2.75
## 27                  Michigan  453931040 10012568                  2.21
## 28                 Minnesota  288645998  3605463                  1.25
## 29               Mississippi  171532388  3720209                  2.17
## 30                  Missouri  310799410  4677753                  1.51
## 31                   Montana   56592296   778394                  1.38
## 32                  Nebraska  108758454  1089339                  1.00
## 33                    Nevada  160901056  2717798                  1.69
## 34             New Hampshire   46618016   649666                  1.39
## 35               New Jersey  473701874 14011087                  2.96
## 36               New Mexico  100637143  1937511                  1.93
## 37                 New York  999841117 28645249                  2.86
## 38           North Carolina  496573788  6259097                  1.26
## 39             North Dakota   54727328   707840                  1.29
## 40 Northern Mariana Islands     238070     1929                  0.81
## 41                     Ohio  539574789  9753786                  1.81
## 42                 Oklahoma  223813728  3244661                  1.45
## 43                   Oregon  108440757  1408740                  1.30
## 44             Pennsylvania  560845432 12882390                  2.30
## 45              Puerto Rico   66792521  1154504                  1.73
## 46             Rhode Island   70291283  1331226                  1.89
## 47           South Carolina  297266041  4762878                  1.60
## 48             South Dakota   59651348   889192                  1.49
## 49                Tennessee  435190081  5862287                  1.35
## 50                    Texas 1487676083 24773770                  1.67
## 51                     Utah  199344756  1112670                  0.56
## 52                  Vermont   12554607   128600                  1.02
## 53           Virgin Islands    2209786    22268                  1.01
## 54                 Virginia  326571167  5106821                  1.56
## 55               Washington  229148000  3060435                  1.34
## 56            West Virginia   81751586  1364475                  1.67
## 57                Wisconsin  333851532  3663748                  1.10
## 58                  Wyoming   31941561   370815                  1.16
```

```
max_cases <- max(y$CASES)
min_cases <- min(y$CASES)
max_deaths <- max(y$DEATHS)
min_deaths <- min(y$DEATHS)
```

```
print(paste("The maximum number of covid19 cases was:",max_cases,"in", y$Province_State[y$CASES==max_ca
```

```
## [1] "The maximum number of covid19 cases was: 1839596664 in California ."
```

```
print(paste("The minimum number of covid19 cases was:",min_cases,"in", y$Province_State[y$CASES==min_ca
```

```
## [1] "The minimum number of covid19 cases was: 789 in American Samoa ."
```

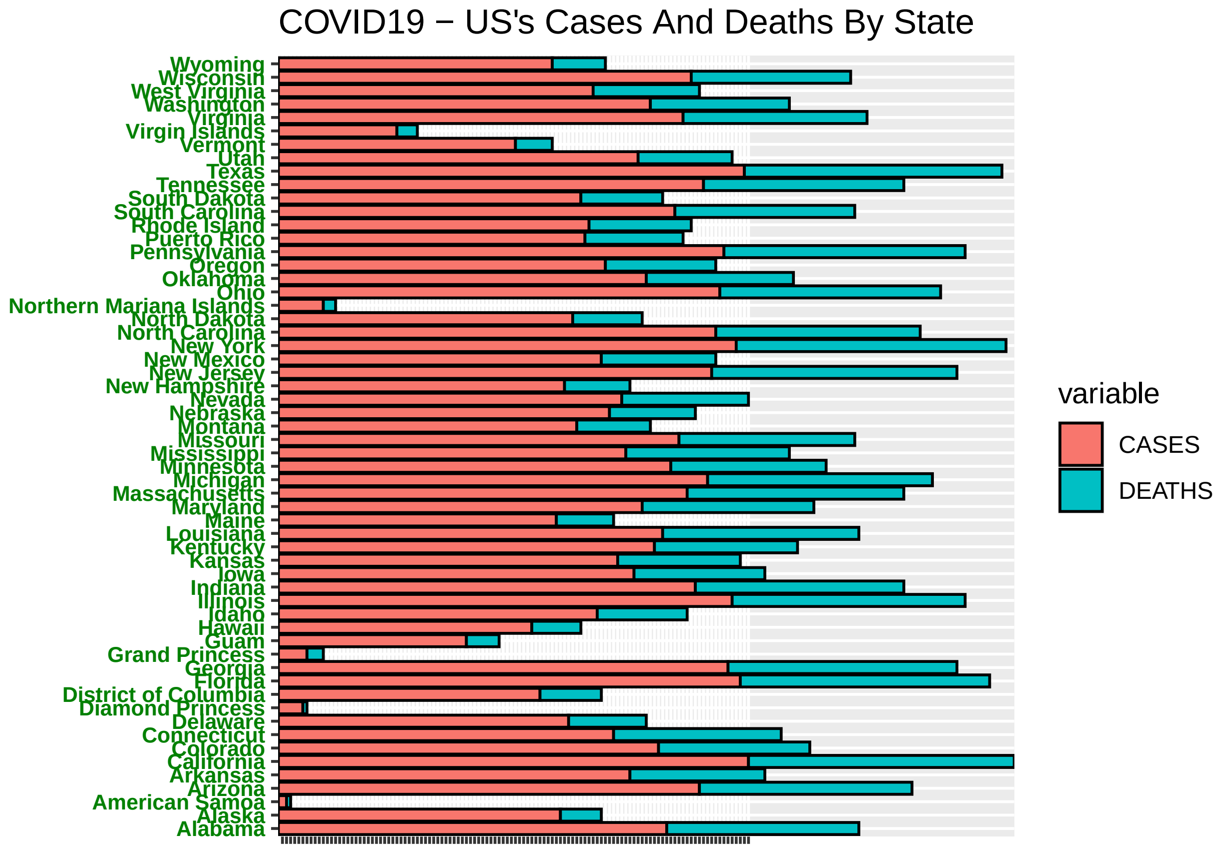
```
print(paste("The maximum number of covid19 deaths was:",max_deaths,"in", y$Province_State[y$DEATHS==max_
```

```
## [1] "The maximum number of covid19 deaths was: 28645249 in New York ."
```

```
print(paste("The minimum number of covid19 deaths was:",min_deaths,"in", y$Province_State[y$DEATHS==min_
```

```
## [1] "The minimum number of covid19 deaths was: 0 in American Samoa ."
## [2] "The minimum number of covid19 deaths was: 0 in Diamond Princess ."
```

```
# plot the US's cases and deaths chart
ggplot(data=melt(y[, 1:3], id.vars=c("Province_State")), aes(x=Province_State, y=format(value, scientifi
     geom_bar(stat="identity", colour="black")+
         coord_flip() + scale_y_discrete(name="") +
      theme(axis.title.x=element_blank(),
      axis.title.y=element_blank(),axis.text.x = element_blank(),
          axis.text.y = element_text(face="bold", color="#008000",
                           size=8, angle=0))+
     ggtitle("COVID19 - US's Cases And Deaths By State")
```



This plot tells us that the maximum number of covid 19 cases was in California and the minimum of covid 19 cases was in American Samoa. Diamond Princess and American Samoa were the places have no death cases. Moreover, Texas, New York and Florida were the states have the large number of covid 19 cases and deaths. And Grand Princess and Northern were the states have the small number of covid 19 cases and deaths.

Question 3: What is the rate of deaths per cases in US by state?

```
max_rate <- max(y$DEATHS_PER_CASES_RATE)
min_rate <- min(y$DEATHS_PER_CASES_RATE)
```

```
print(paste("The highest rate of covid19 deaths per cases was:",max_rate,"in", y$Province_State[y$DEATHS
```

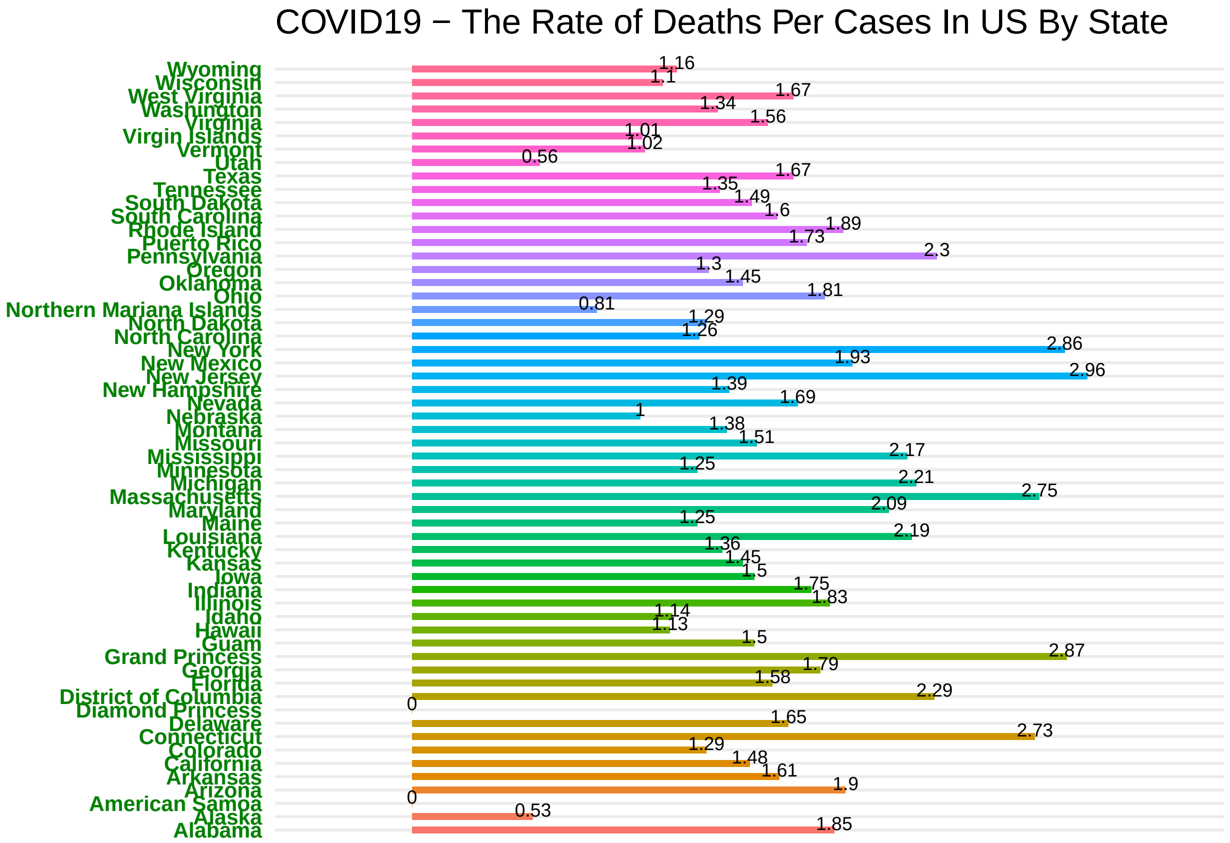
```
## [1] "The highest rate of covid19 deaths per cases was: 2.96 in New Jersey ."
```

```
print(paste("The lowest rate of covid19 deaths per cases was:",min_rate,"in", y$Province_State[y$DEATHS_
```

```
## [1] "The lowest rate of covid19 deaths per cases was: 0 in American Samoa ."
## [2] "The lowest rate of covid19 deaths per cases was: 0 in Diamond Princess ."
```

```
# plot the chart of US deaths by state
ggplot(data=y, aes(x=Province_State, y=DEATHS_PER_CASES_RATE, fill=Province_State)) +
geom_bar(stat="identity", width=0.5)+ theme_minimal() +
         coord_flip()+
geom_text(aes(label=DEATHS_PER_CASES_RATE), vjust=0, color="black",
            position = position_dodge(2), size=2.5)+
scale_y_discrete(name="DEATHS")+ theme(axis.title.x=element_blank(),
      axis.title.y=element_blank(),axis.text.x = element_blank(),
          axis.text.y = element_text(face="bold", color="#008000",
                                    size=8, angle=0),legend.position="none")+
ggtitle("COVID19 - The Rate of Deaths Per Cases In US By State")
```

```
## Warning: position_dodge requires non-overlapping x intervals
```



As we can see, all but two states have the death cases. Moreover, the two states are Alaska and Utah have the low rates of deaths per cases (0.53% and 0.56% respectively). The highest rate of covid 19 deaths per cases was in New Jersey (2.97%).

Question 4: How were the trend of new cases and new deaths in US?

```
# Create the data for the chart
US_totals <- US_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
  ungroup()
```

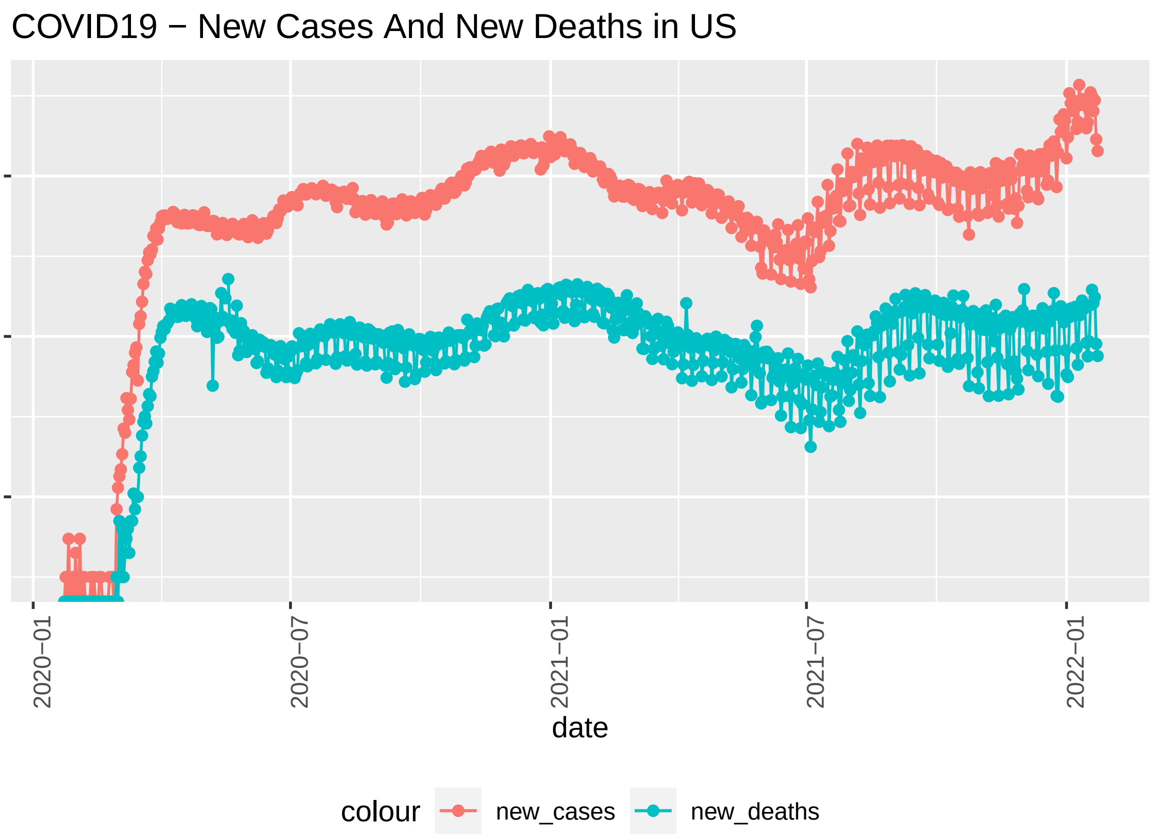
```
## `summarise()` has grouped output by 'Country_Region'. You can override using the `.groups` argument.
```

```
US_totals <- US_totals %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths))
tail(US_totals)
```

```
## # A tibble: 6 x 8
##   Country_Region date          cases deaths deaths_per_mill Population new_cases
##   <chr>          <date>        <dbl>  <dbl>           <dbl>      <dbl>     <dbl>
## 1 US             2022-01-18 67693339 854442           2574.  331944132   1103191
## 2 US             2022-01-19 68684431 858257           2586.  331944132    991092
## 3 US             2022-01-20 69329860 860845           2593.  331944132    645429
## 4 US             2022-01-21 70209840 863924           2603.  331944132    879980
## 5 US             2022-01-22 70495874 864732           2605.  331944132    286034
## 6 US             2022-01-23 70700678 865302           2607.  331944132    204804
## # ... with 1 more variable: new_deaths <dbl>
```

```
# visualize the chart of trend of US cases and deaths
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 - New Cases And New Deaths in US", y = NULL)
```

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

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

## Warning in self$trans$transform(x): NaNs produced

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

## Warning in self$trans$transform(x): NaNs produced

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

## Warning: Removed 1 row(s) containing missing values (geom_path).

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

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 3 rows containing missing values (geom_point).
```



This plot tells us that the number of new cases and new deaths increased most in March 2020. After that, there was a decrease of new cases and new deaths in July 2021 but the new cases increased again from September 2021 to now. And there were still a lot of new deaths until now.

Question 5: How many cases and deaths globally by year?

```
global_totals <- global %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
  ungroup()
```

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

```
head(global_totals)
```

```
## # A tibble: 6 x 6
##   Country_Region date       cases deaths deaths_per_mill Population
##   <chr>          <date>     <dbl>  <dbl>           <dbl>      <dbl>
## 1 Afghanistan    2020-02-24     5      0               0   38928341
## 2 Afghanistan    2020-02-25     5      0               0   38928341
## 3 Afghanistan    2020-02-26     5      0               0   38928341
## 4 Afghanistan    2020-02-27     5      0               0   38928341
## 5 Afghanistan    2020-02-28     5      0               0   38928341
## 6 Afghanistan    2020-02-29     5      0               0   38928341
```

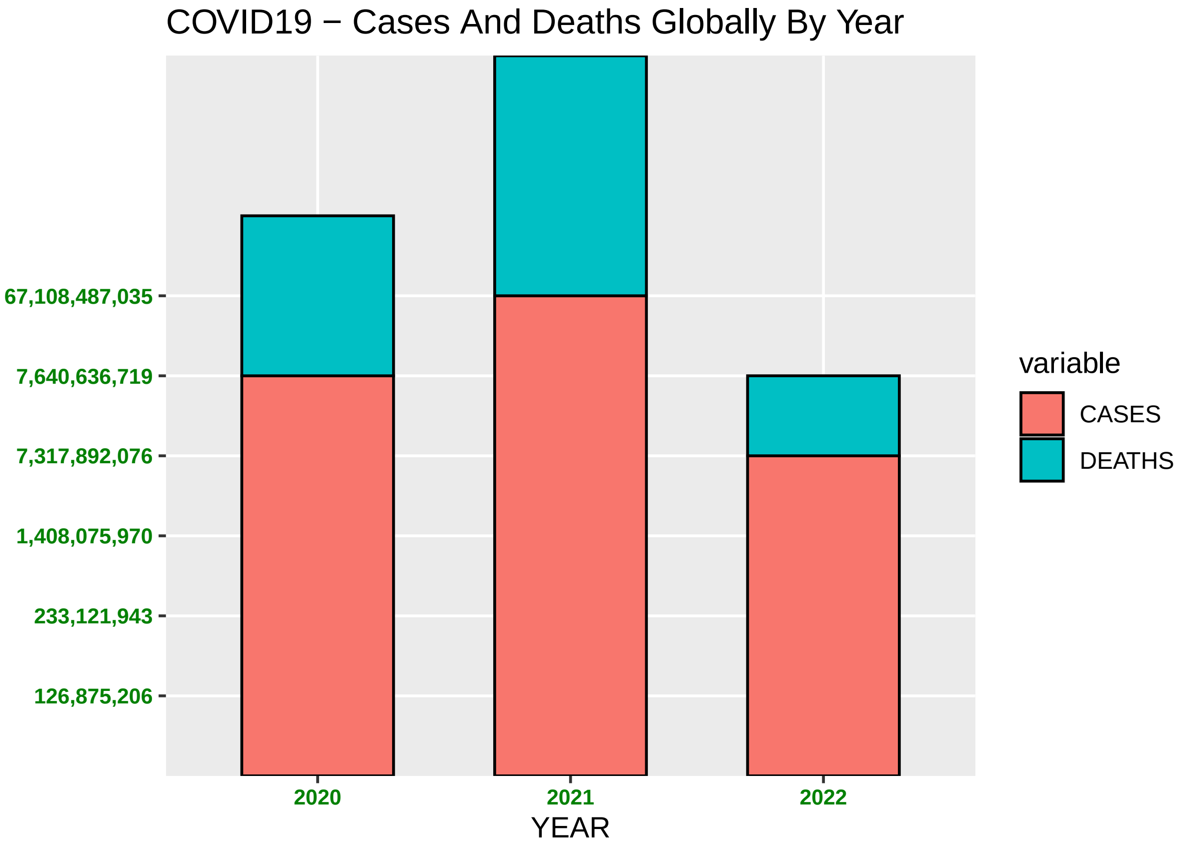
```r
global_totals <- global_totals %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths))
tail(global_totals)
```

```
## # A tibble: 6 x 8
##   Country_Region date         cases deaths deaths_per_mill Population new_cases
##   <chr>          <date>       <dbl>  <dbl>           <dbl>      <dbl>     <dbl>
## 1 Zimbabwe       2022-01-18 226460   5258            354.   14862927         0
## 2 Zimbabwe       2022-01-19 226887   5266            354.   14862927       427
## 3 Zimbabwe       2022-01-20 227552   5276            355.   14862927       665
## 4 Zimbabwe       2022-01-21 227961   5288            356.   14862927       409
## 5 Zimbabwe       2022-01-22 228179   5292            356.   14862927       218
## 6 Zimbabwe       2022-01-23 228254   5294            356.   14862927        75
## # ... with 1 more variable: new_deaths <dbl>
```

```r
# Create the data for the chart
n <- global_totals %>% filter(cases > 0) %>%
mutate(YEAR = format(as.Date(global_totals$date, format="%Y/%m/%d"),"%Y")) %>%
  group_by(YEAR) %>%
  summarise(CASES = sum(cases), DEATHS = sum(deaths))
head(n)
```

```
## # A tibble: 3 x 3
##   YEAR        CASES     DEATHS
##   <chr>       <dbl>      <dbl>
## 1 2020   7640636719  233121943
## 2 2021  67108487035 1408075970
## 3 2022   7317892076  126875206
```

```r
# Visualize the number of cases and deaths globally by year
ggplot(data=melt(n, id.vars=c("YEAR")), aes(x=YEAR, y=format(value, scientific = FALSE, big.mark = ',')
     geom_bar(width = 0.6, stat="identity", colour="black")+
         scale_y_discrete(name="") +
      theme(axis.text.x = element_text(face="bold", color="#008000",
                           size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                           size=8, angle=0))+
     ggtitle("COVID19 - Cases And Deaths Globally By Year")
```



As histogram above, until the beginning of 2022, the largest covid 19 cases globally was 67,108,487,035 and the largest covid 19 deaths globally was 1,408,075,970 in 2021.

Question 6: How were the trend of covid 19 cases and deaths globally by season?

```
global_month <- global_totals %>% filter(cases>0) %>%
  mutate(month = month(as.POSIXlt(date, format="%d/%m/%Y")) %>% as.integer() ) %>%
  mutate(year = year(as.POSIXlt(date, format="%d/%m/%Y"))) %>%
  select(year,month, cases, deaths)
head(global_month)
```

```
## # A tibble: 6 x 4
##    year month cases deaths
##   <dbl> <int> <dbl>  <dbl>
## 1  2020     2     5      0
## 2  2020     2     5      0
## 3  2020     2     5      0
## 4  2020     2     5      0
## 5  2020     2     5      0
## 6  2020     2     5      0
```

```
global_month <- global_month %>%
  group_by(year, month) %>%
  summarize(cases = sum(cases), deaths = sum(deaths))
```

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

```
global_month
```

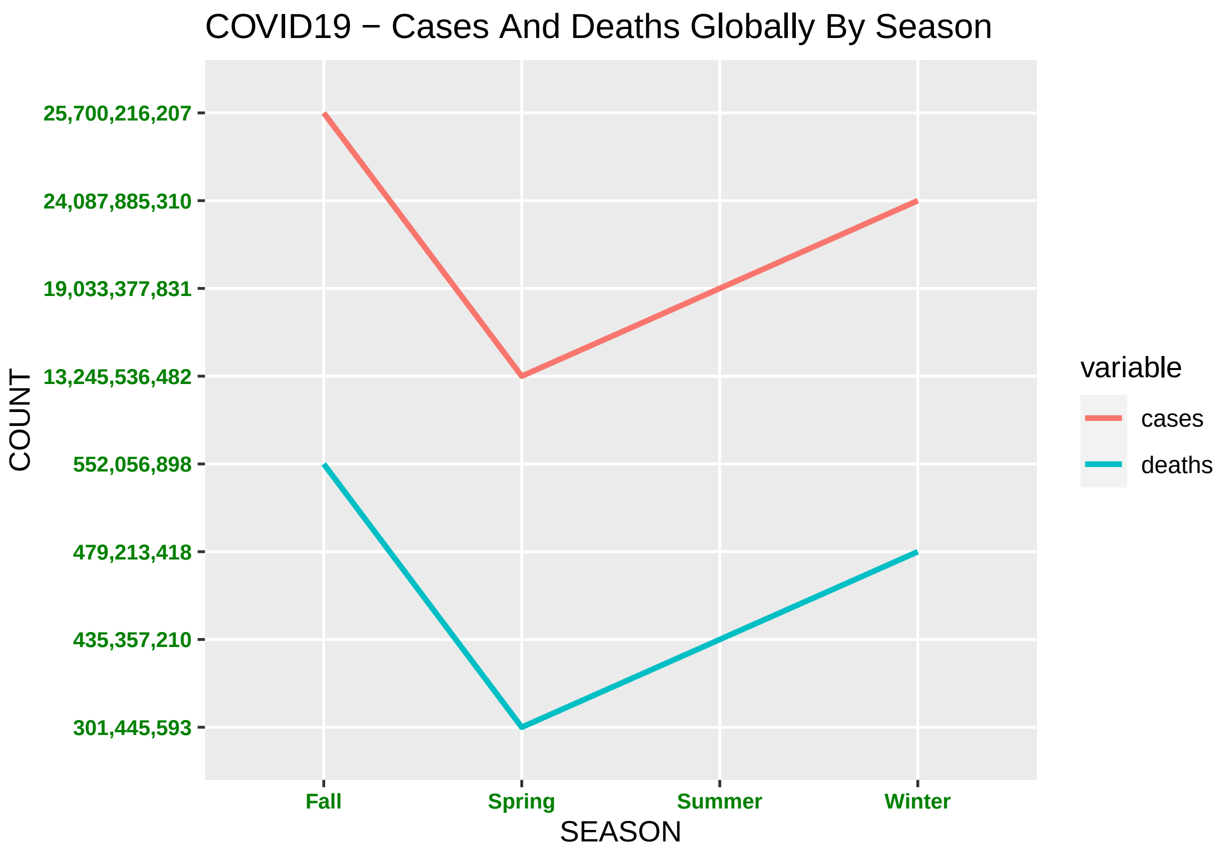
```
## # A tibble: 25 x 4
## # Groups:   year [3]
##     year month      cases   deaths
##    <dbl> <int>      <dbl>    <dbl>
##  1  2020     1      38539      889
##  2  2020     2    1672070    46911
##  3  2020     3    9064473   400553
##  4  2020     4   63486110  4400342
##  5  2020     5  145026161  9986371
##  6  2020     6  246717664 13975764
##  7  2020     7  431716854 19312594
##  8  2020     8  672119975 25081958
##  9  2020     9  895467437 29550847
## 10  2020    10 1229346730 35870023
## # ... with 15 more rows
```

```
global_month <- global_month %>%
  mutate(
    season = case_when(
      month %in%  9:11 ~ "Fall",
      month %in%  c(12, 1, 2)  ~ "Winter",
      month %in%  3:5  ~ "Spring",
      TRUE ~ "Summer"))
global_season <- global_month %>% group_by(season) %>%
     summarize(cases = sum(cases), deaths = sum(deaths)) %>%
     mutate(freq_cases = round(cases / sum(cases)*100, 2))%>%
     mutate(freq_deaths = round(deaths / sum(deaths)*100, 2))%>%
     select(season, cases, deaths, freq_cases, freq_deaths)
global_season
```

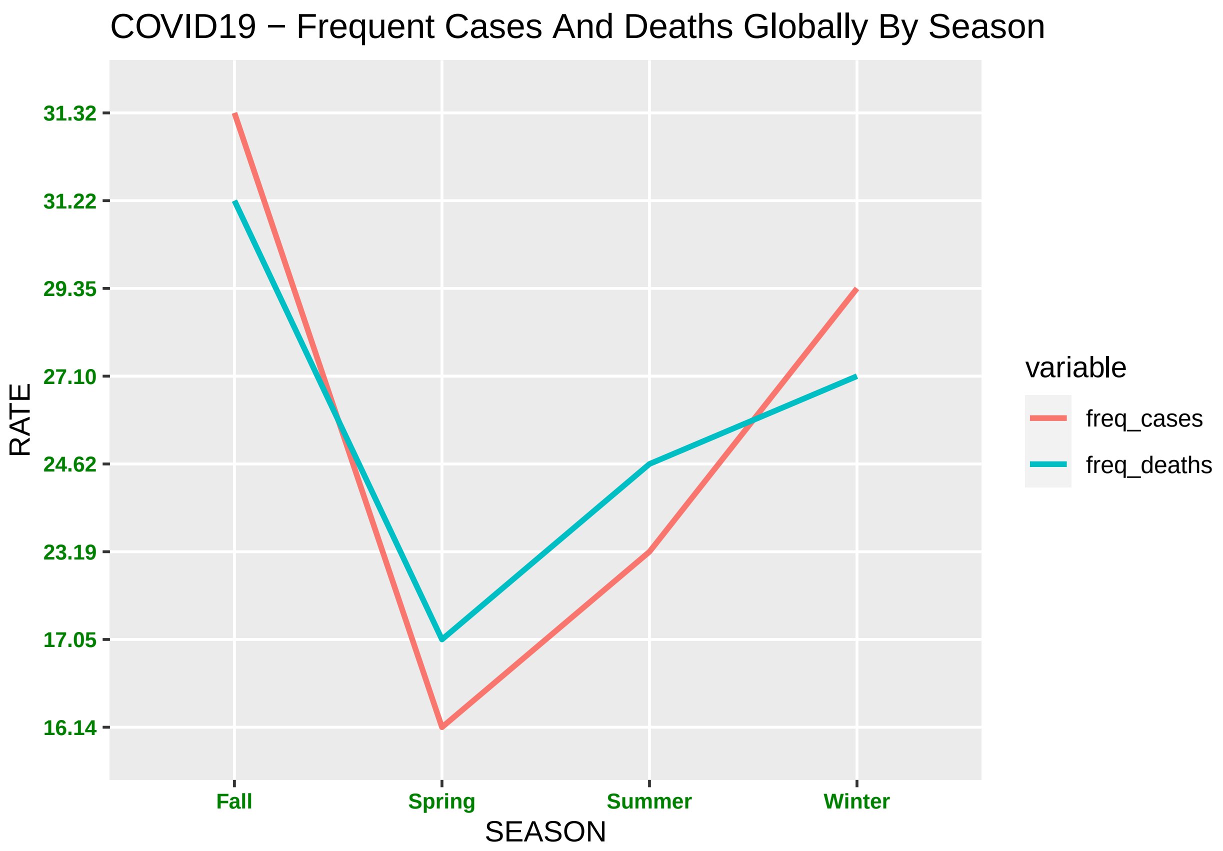
```
## # A tibble: 4 x 5
##   season       cases    deaths freq_cases freq_deaths
##   <chr>        <dbl>     <dbl>      <dbl>       <dbl>
## 1 Fall   25700216207 552056898       31.3        31.2
## 2 Spring 13245536482 301445593       16.1        17.0
## 3 Summer 19033377831 435357210       23.2        24.6
## 4 Winter 24087885310 479213418       29.4        27.1
```

```
data1 <- melt(global_season[,1:3], id.vars=c("season"))
data2 <- melt(global_season %>% select(season, freq_cases, freq_deaths), id.vars=c("season"))
```

```
data1 <- melt(global_season[,1:3], id.vars=c("season"))
data2 <- melt(global_season %>% select(season, freq_cases, freq_deaths), id.vars=c("season"))

par(mfrow = c(1, 2))
ggplot(data1, aes(x = factor(season), y = format(value, scientific = FALSE, big.mark = ','), colour = va
  geom_line(stat="identity", size = 1)+
         scale_y_discrete(name="COUNT") + scale_x_discrete(name="SEASON") +
      theme(axis.text.x = element_text(face="bold", color="#008000",
                          size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                          size=8, angle=0))+
     ggtitle("COVID19 - Cases And Deaths Globally By Season")
```



```
ggplot(data2, aes(x = factor(season), y = format(value, scientific = FALSE, big.mark = ','), colour = va
  geom_line(stat="identity", size = 1)+
         scale_y_discrete(name="RATE") + scale_x_discrete(name="SEASON") +
      theme(axis.text.x = element_text(face="bold", color="#008000",
                           size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                           size=8, angle=0))+
     ggtitle("COVID19 - Frequent Cases And Deaths Globally By Season")
```



The two plots above tell us that the most globally covid 19 cases were 25,700,216,207 and the most globally covid 19 deaths were 552,056,898 in fall. The least globally covid 19 cases were 13,245,536,482 and the least globally covid 19 deaths were 301,445,593 in spring. The most frequent cases were 31.45% in fall, the least frequent cases were 16.21% in spring. The most frequent deaths were 31.32% in fall and the least frequent deaths were 17.1% in spring.

Build model and visualization

```
# create the data to build the model
US_month <- US_totals %>% filter(cases>0) %>%
  mutate(month = month(as.POSIXlt(date, format="%d/%m/%Y")) %>% as.integer() ) %>%
  group_by(Country_Region, month) %>%
  summarize(cases = sum(cases), deaths = sum(deaths)) %>%
  select(Country_Region, month, cases, deaths)
```

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

```
US_month
```

```
## # A tibble: 12 x 4
## # Groups:   Country_Region [1]
##    Country_Region month      cases   deaths
##    <chr>          <int>      <dbl>    <dbl>
##  1 US                 1 2185592053 31843495
##  2 US                 2  777043858 13586118
##  3 US                 3  920991055 16615865
##  4 US                 4  967680213 17889475
##  5 US                 5 1068895978 20811320
##  6 US                 6 1072874889 21536607
##  7 US                 7 1172636851 23150354
##  8 US                 8 1317442749 24589101
##  9 US                 9 1449719181 25934806
## 10 US                10 1644183895 29241530
## 11 US                11 1758047310 30441563
## 12 US                12 2106064469 34597064
```

```
# create US covid19 with cases, deaths, frequent cases and deaths by month
US_month <- US_month %>%
   mutate(freq_cases = round(cases / sum(cases)*100, 2))%>%
   mutate(freq_deaths = round(deaths / sum(deaths)*100, 2))
US_month
```

```
## # A tibble: 12 x 6
## # Groups:   Country_Region [1]
##    Country_Region month      cases   deaths freq_cases freq_deaths
##    <chr>          <int>      <dbl>    <dbl>      <dbl>       <dbl>
##  1 US                 1 2185592053 31843495      13.3        11.0
##  2 US                 2  777043858 13586118       4.73        4.68
##  3 US                 3  920991055 16615865       5.6         5.72
##  4 US                 4  967680213 17889475       5.89        6.16
##  5 US                 5 1068895978 20811320       6.5         7.17
##  6 US                 6 1072874889 21536607       6.53        7.42
##  7 US                 7 1172636851 23150354       7.13        7.98
##  8 US                 8 1317442749 24589101       8.01        8.47
##  9 US                 9 1449719181 25934806       8.82        8.94
## 10 US                10 1644183895 29241530      10          10.1
## 11 US                11 1758047310 30441563      10.7        10.5
## 12 US                12 2106064469 34597064      12.8        11.9
```

```
# Use the lm() function to perform a polinomial regression with frequent cases as the response
# and month as the predictor.
# Use the summary() function to print the results
mod1 <- lm(freq_cases ~ poly(month, 2, raw=TRUE), data = US_month)
summary(mod1)
```

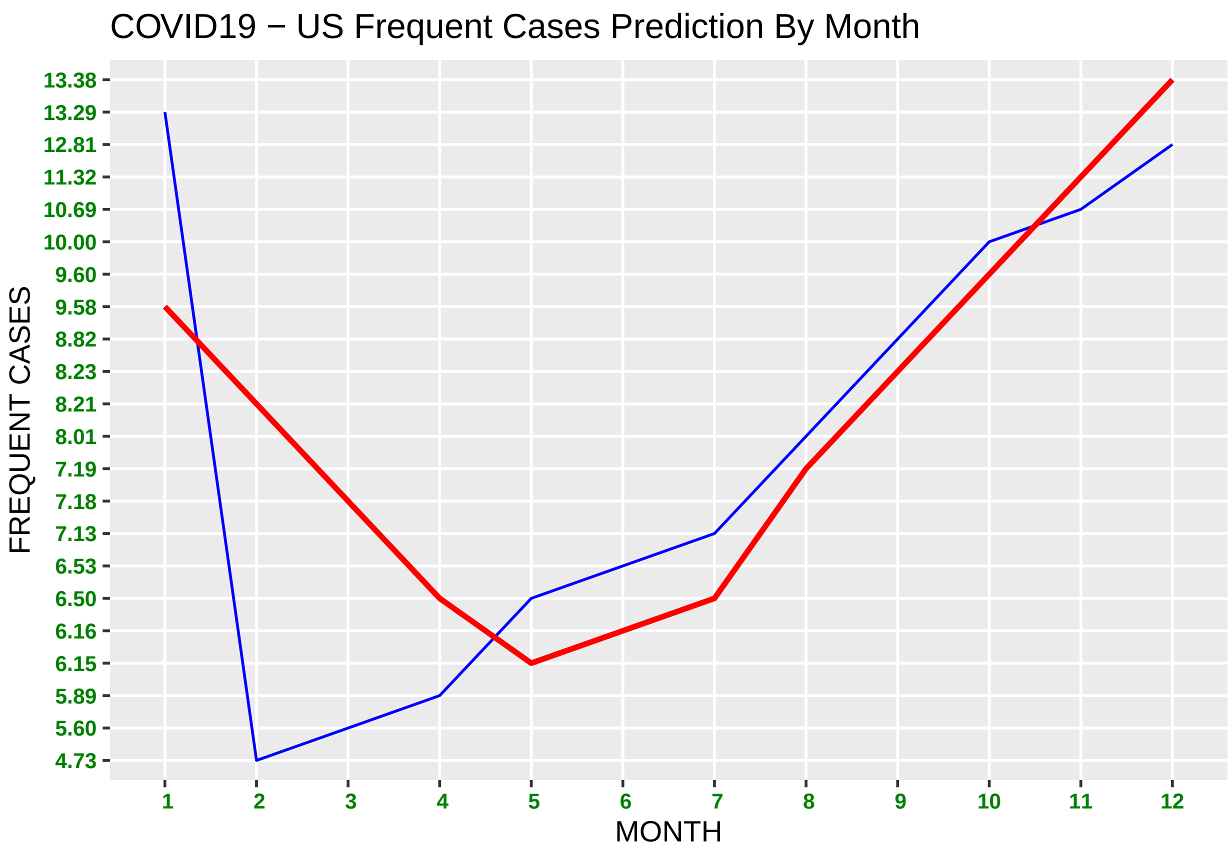
```
##
## Call:
## lm(formula = freq_cases ~ poly(month, 2, raw = TRUE), data = US_month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4777 -0.6118  0.3595  0.6031  3.7108
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 11.29409    1.92709   5.861 0.000241 ***
## poly(month, 2, raw = TRUE)1 -1.88665    0.68157  -2.768 0.021821 *
## poly(month, 2, raw = TRUE)2  0.17174    0.05104   3.365 0.008324 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.865 on 9 degrees of freedom
## Multiple R-squared:  0.6435, Adjusted R-squared:  0.5643
## F-statistic: 8.123 on 2 and 9 DF,  p-value: 0.009645
```

Looking at the summary of this model, we can see that our p-value is very small, this means that the predictor (month) were statistically significant in determining the response (frequent cases). And the frequent cases is 11.02477 - 1.80340 X month + 0.16674 X $month^2$.

```
# create new data with the predict of the frequent cases by the month
US_month_w_pred <- US_month %>% mutate(PREDICT = round(predict(mod1), 2))
US_month_w_pred
```

```
## # A tibble: 12 x 7
## # Groups:   Country_Region [1]
##    Country_Region month      cases   deaths freq_cases freq_deaths PREDICT
##    <chr>          <int>      <dbl>    <dbl>      <dbl>       <dbl>   <dbl>
##  1 US                 1 2185592053 31843495      13.3        11.0     9.58
##  2 US                 2  777043858 13586118       4.73        4.68    8.21
##  3 US                 3  920991055 16615865       5.6         5.72    7.18
##  4 US                 4  967680213 17889475       5.89        6.16    6.5
##  5 US                 5 1068895978 20811320       6.5         7.17    6.15
##  6 US                 6 1072874889 21536607       6.53        7.42    6.16
##  7 US                 7 1172636851 23150354       7.13        7.98    6.5
##  8 US                 8 1317442749 24589101       8.01        8.47    7.19
##  9 US                 9 1449719181 25934806       8.82        8.94    8.23
## 10 US                10 1644183895 29241530      10          10.1     9.6
## 11 US                11 1758047310 30441563      10.7        10.5    11.3
## 12 US                12 2106064469 34597064      12.8        11.9    13.4
```

```
# plot the new data
US_month_w_pred %>% ggplot() + geom_line(aes(x = format(month, scientific = FALSE, big.mark = ','), y =
  geom_line(aes(x = format(month, scientific = FALSE, big.mark = ','), y = format(PREDICT, scientific =
         scale_y_discrete(name="FREQUENT CASES") + scale_x_discrete(name="MONTH") +
      theme(axis.text.x = element_text(face="bold", color="#008000",
                          size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                          size=8, angle=0))+
     ggtitle("COVID19 - US Frequent Cases Prediction By Month")
```



In the plot above, our predictions are in red and our actuals are in blue. So we can see the model does a reasonably good job of predicting from month 7 to 12.

```
# Use the lm() function to perform a regression with frequent deaths as the response
# and frequent cases as the predictor.
# Use the summary() function to print the results
mod2 <- lm(freq_deaths ~ freq_cases, data = US_month)
summary(mod2)
```

```
##
## Call:
## lm(formula = freq_deaths ~ freq_cases, data = US_month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1580 -0.3573  0.2371  0.4050  0.5679
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.95360    0.56704   3.445  0.00628 **
## freq_cases   0.76557    0.06472  11.828 3.34e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6063 on 10 degrees of freedom
## Multiple R-squared:  0.9333, Adjusted R-squared:  0.9266
## F-statistic: 139.9 on 1 and 10 DF,  p-value: 3.345e-07
```

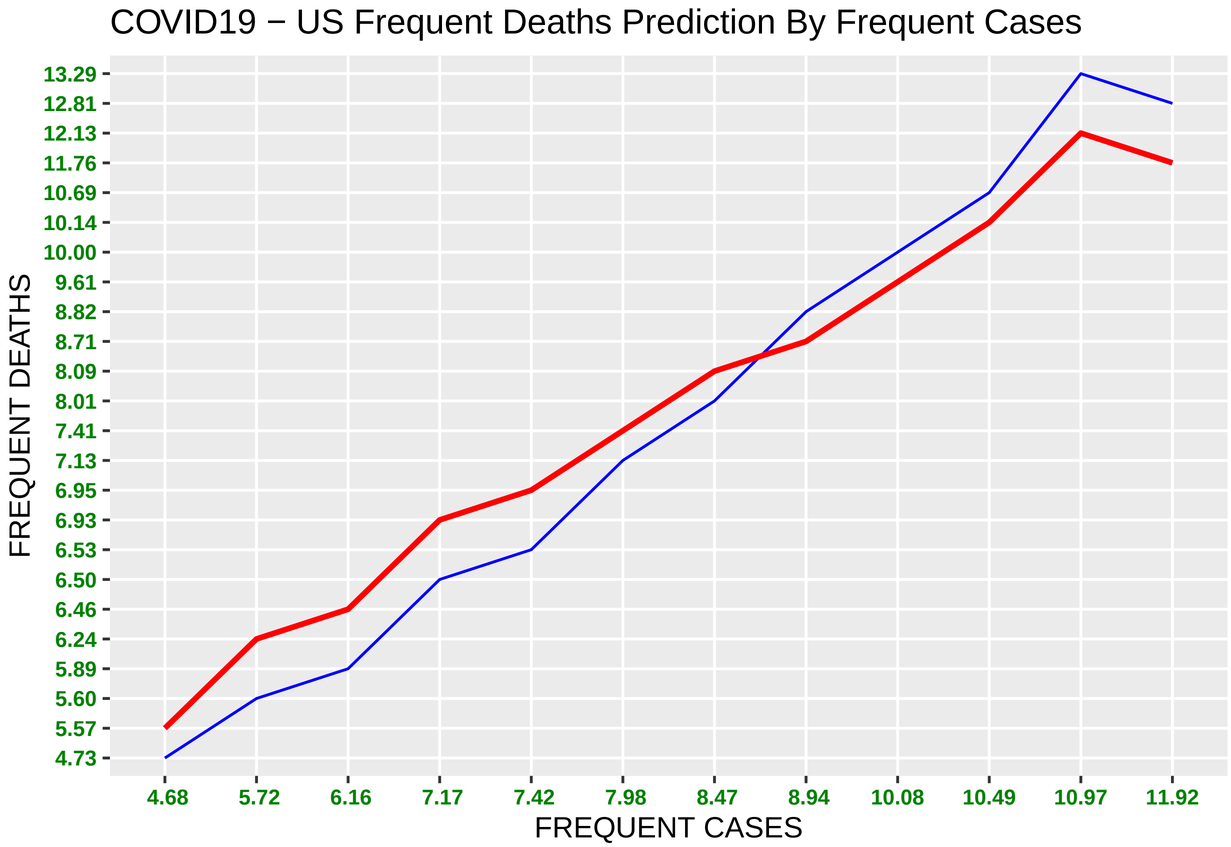
- Looking at the summary of this model, we can see that our p-value is very small, this means that the predictor (frequent cases) were statistically significant in determining the response (frequent deaths). And the frequent deaths is 1.90183 + 0.77180 X frequent cases.

- The regression coefficient for frequent cases is: 0.77180, this means an increase of frequent cases is associated with an increase of frequent deaths by 0.77180%, keeping all else constant.

```
# create new data with predict the monthly frequent deaths by the monthly frequent cases
US_month_w_d_pred <- US_month %>% mutate(PRED = round(predict(mod2), 2))
US_month_w_d_pred
```

```
## # A tibble: 12 x 7
## # Groups:   Country_Region [1]
##    Country_Region month      cases   deaths freq_cases freq_deaths  PRED
##    <chr>          <int>      <dbl>    <dbl>      <dbl>       <dbl> <dbl>
##  1 US                 1 2185592053 31843495      13.3        11.0  12.1
##  2 US                 2  777043858 13586118       4.73        4.68  5.57
##  3 US                 3  920991055 16615865       5.6         5.72  6.24
##  4 US                 4  967680213 17889475       5.89        6.16  6.46
##  5 US                 5 1068895978 20811320       6.5         7.17  6.93
##  6 US                 6 1072874889 21536607       6.53        7.42  6.95
##  7 US                 7 1172636851 23150354       7.13        7.98  7.41
##  8 US                 8 1317442749 24589101       8.01        8.47  8.09
##  9 US                 9 1449719181 25934806       8.82        8.94  8.71
## 10 US                10 1644183895 29241530      10          10.1   9.61
## 11 US                11 1758047310 30441563      10.7        10.5  10.1
## 12 US                12 2106064469 34597064      12.8        11.9  11.8
```

```
# plot the new data
US_month_w_d_pred %>% ggplot() + geom_line(aes(x = format(freq_deaths, scientific = FALSE, big.mark = '
  geom_line(aes(x = format(freq_deaths, scientific = FALSE, big.mark = ','), y = format(PRED, scientifi
         scale_y_discrete(name="FREQUENT DEATHS") + scale_x_discrete(name="FREQUENT CASES") +
      theme(axis.text.x = element_text(face="bold", color="#008000",
                          size=8, angle=0),
          axis.text.y = element_text(face="bold", color="#008000",
                          size=8, angle=0))+
     ggtitle("COVID19 - US Frequent Deaths Prediction By Frequent Cases")
```



In the plot above, our predictions are in red and our actuals are in blue. So we can see the model does a reasonably good job in predicting frequent deaths by frequent cases.

Step 4: Conclusion and add bias identification

In conclusion, base on US covid 19 and Global covid 19 data from the Johns Hopkins University:

- First, while cleaning up the data, I recognized that there are a lot of missing values about Province_State and Population in global data, US covid 19 data has missing values of admin2 as well. Missing data can be a major cause of information bias, where certain groups of people are more likely to have missing data. Since this is a huge number, deleting the instances with missing observations can result in biased parameters and estimates and reduce the statistical power of the analysis.
- Next, by plotting the COVID 19 - US Cases And Deaths By Year, we can see that the most of covid 19 cases and deaths were in 2021.
- As “COVID19 - US’s Cases And Deaths By State” histogram, we see that the maximum number of covid 19 cases was in California and the minimum of covid 19 cases was in American Samoa. Diamond Princess and American Samoa were the places have no death cases. Moreover, Texas, New York and Florida were the states have the large number of covid 19 cases and deaths. And Grand Princess and Northern were the states have the small number of covid 19 cases and deaths.
- COVID19 - The Rate of Deaths Per Cases In US By State histogram tells us that all but two states have the death cases. Moreover, the two states are Alaska and Utah have the low rates of deaths per cases. The highest rate of covid 19 deaths per cases was in New Jersey.
- COVID19 - New Cases And New Deaths in US chart shows that the number of new cases and new deaths increased most in March 2020. After that, there was a decrease of new cases and new deaths

in July 2021 but the new cases increased again from September 2021 to now. And there were still a lot of new deaths until now.

- Base on COVID19 - Cases And Deaths Globally By Year plot,until the beginning of 2022, the largest covid 19 cases globally was 67,108,487,035 and the largest covid 19 deaths globally was 1,408,075,970 in 2021.
- The most globally covid 19 cases and deaths were in fall. The least globally covid 19 cases and deaths were in spring. The most frequent cases were 31.45% in fall, the least frequent cases were 16.21% in spring. The most frequent deaths were 31.32% in fall and the least frequent deaths were 17.1% in spring.