Rmarkdown Practice - COVID Data

R. Dadmun

2024-01-31

Import Libraries

```
library(tidyverse)
library(dplyr)
library(ggplot2)
library(lubridate)
```

Project Purpose

This is an Rmd file analyzing COVID-19 data from the John Hopkins Github site for the final project in DTSA 5301: Data Science as a field. This file will serve as an example that I am able to complete the data science process by constructing a reproducible report.

Questions of Interest

1. Which Tennessee county had the highest case rate per population, and which had the lowest case rate per population?

2. Will a Linear Regression Model be able to predict future COVID case rates and death rates in Tennessee?

Describe and Import the Dataset

Data Description

CSSE COVID-19 Time Series Data The first two data sets are time sereies for US Confirmed COVID-19 Cases and Deaths by county.

The next two data sets are global confirmed COVID-19 Cases and Deaths. Most countrys are reported out at the country level, although some are reported at the province or state level.

Source Link https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

##Import Datasets

```
#All files taken from the URL below:
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cov

#Vector containing our file names:
 file_names <- c("time_series_covid19_confirmed_global.csv",
"time_series_covid19_deaths_global.csv",
"time_series_covid19_confirmed_US.csv",
"time_series_covid19_deaths_US.csv")

#Concatenate our url_in with file names:
urls <- str_c(url_in,file_names)
```

Next we load them into specific data sets for R to read:

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

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
  global_deaths <- read.csv(urls[2])
  US_cases <- read.csv(urls[3])
  US_deaths <- read.csv(urls[4])
```

##Tidy and Transform Global Data

Tidying Goals:
- Each variable should be in an "R friendly syntax" - Each date should have it's own entry - e.g. The dataset is organized such that each date is present on a separate row of the data sheet.
- Field objects should be applicable to the type of data held within the field.
- Filter out unnecessary data.
- Collate and clean missing data.
- Transform the Data:
- Join global cases and global deaths per date.
- Join US cases and US deaths per date.

The final two lines in the tidy code blocks are to ensure that the date column transforms correctly. Although the instructor utilized lubridate, there exist functions within standard R to change the date format. Many of these datasets also had an "X" in front of each date, which was initially a large issue and caused lubridate to fail to parse.

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

```
global_cases <- global_cases %>%
  rename('Country.Region' = 'Country/Region',
    'Province.State' = 'Province/State')

global_cases$date <- as.Date(global_cases$date, "%m/%d/%y")
```

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

global_deaths$date <- sub('.', '', global_deaths$date)
global_deaths$date <- as.Date(global_deaths$date, "%m.%d.%y")
```

```
global <- global_cases %>%
  full_join(global_deaths)
```

```
## Joining with `by = join_by(Province.State, Country.Region, Lat, Long, date)`
```

```
global$date <- as.Date(global$date, "%m/%d/%Y")

#Finally, we pull a summary of the Global Data to ensure everything merged correctly
summary(global)
```

```
##  Province.State     Country.Region          Lat               Long
##  Length:556641      Length:556641      Min.   :-71.950   Min.   :-178.12
##  Class :character   Class :character   1st Qu.:  3.919   1st Qu.: -11.78
##  Mode  :character   Mode  :character   Median : 18.971   Median :  20.94
##                                        Mean   : 19.219   Mean   :  21.67
##                                        3rd Qu.: 40.182   3rd Qu.:  66.92
##                                        Max.   : 71.707   Max.   : 178.06
##                                        NA's   :2286      NA's   :2286
##       date                cases               deaths
##  Min.   :2020-01-22   Min.   :        0   Min.   :      0
##  1st Qu.:2020-11-02   1st Qu.:      680   1st Qu.:      3
##  Median :2021-08-15   Median :    14429   Median :    150
##  Mean   :2021-08-15   Mean   :   959384   Mean   :  13380
##  3rd Qu.:2022-05-28   3rd Qu.:   228517   3rd Qu.:   3032
##  Max.   :2023-03-09   Max.   :103802702   Max.   :1123836
##                       NA's   :226314      NA's   :226314
```

Now for US cases

As with the example above, the final two lines in the tidy code blocks are to ensure that the date column transforms correctly. Although the instructor utilized lubridate, there exist functions within standard R to change the date format. Many of these datasets also had an "X" in front of each date, which was initially a large issue and caused lubridate to fail to parse.

```r
US_cases <- US_cases %>%
  pivot_longer(cols = -(UID:Combined_Key),
               names_to = "date",
               values_to = "cases") %>%
  select(Admin2:cases) %>%
  select(-c(Lat,Long_))

US_cases$date <- sub('.', '', US_cases$date)
US_cases$date <- as.Date(US_cases$date, "%m.%d.%y")

summary(US_cases)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:3819906     Length:3819906     Length:3819906     Length:3819906
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases
##  Min.   :2020-01-22   Min.   :  -3073
##  1st Qu.:2020-11-02   1st Qu.:    330
##  Median :2021-08-15   Median :   2272
##  Mean   :2021-08-15   Mean   :  14088
##  3rd Qu.:2022-05-28   3rd Qu.:   8159
##  Max.   :2023-03-09   Max.   :3710586
```

```r
US_deaths <- US_deaths %>%
  pivot_longer(cols = -(UID:Population),
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  select (-c(Lat,Long_))

US_deaths$date <- sub('.', '', US_deaths$date)
US_deaths$date <- as.Date(US_deaths$date, "%m.%d.%y")

summary(US_deaths)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:3819906     Length:3819906     Length:3819906     Length:3819906
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Population           date                deaths
##  Min.   :       0   Min.   :2020-01-22   Min.   :  -82.0
##  1st Qu.:    9917   1st Qu.:2020-11-02   1st Qu.:    4.0
##  Median :   24892   Median :2021-08-15   Median :   37.0
##  Mean   :   99604   Mean   :2021-08-15   Mean   :  186.9
##  3rd Qu.:   64979   3rd Qu.:2022-05-28   3rd Qu.:  122.0
##  Max.   :10039107   Max.   :2023-03-09   Max.   :35545.0
```

```
US_data <- US_cases %>%
  full_join(US_deaths)
```

```
## Joining with `by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, date)`
```

```
#Pull a summary of the US data to ensure it joined and tidied correctly
summary(US_data)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:3819906     Length:3819906     Length:3819906     Length:3819906
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases           Population          deaths
##  Min.   :2020-01-22   Min.   :  -3073   Min.   :       0   Min.   :  -82.0
##  1st Qu.:2020-11-02   1st Qu.:    330   1st Qu.:    9917   1st Qu.:    4.0
##  Median :2021-08-15   Median :   2272   Median :   24892   Median :   37.0
##  Mean   :2021-08-15   Mean   :  14088   Mean   :   99604   Mean   :  186.9
##  3rd Qu.:2022-05-28   3rd Qu.:   8159   3rd Qu.:   64979   3rd Qu.:  122.0
##  Max.   :2023-03-09   Max.   :3710586   Max.   :10039107   Max.   :35545.0
```

Looking through the data to plan our visualization

####Column Descriptions
- **Admin2**: County Name
- **Province_State**: State Name
- **Country_Region**: US - Static field throughout data set
- **Combined_Key**: Concatenate of the county/state
- **date**: Date in ymd format
- **cases**: Total number of COVID-19 cases per county per date
- **Population**: Population per county
- **deaths**: Total number of deaths attributed to COVID-19 per county - Cases_per_population: Cases per population

####Row Descriptions
- Each **row's unique identifier** is the date for the data

##State Cleaning and Drill Down

As my state of origin, I have chosen to drill down on Tennessee data. As such, I will create four dataframes specific to Tennessee.

We create multiple data frames to individually group the data.
- First, we create **tn_df**, a mirror of the total US data for just TN. Drilling down here, we have two counties listed which will throw out errors in our computation - **Out of TN** and **Unassigned**. As such we will drop these rows.
- Next, we create a data frame titled **tn_counties**, which is a cleaner version of **tn_df**, and mutates a new columnn for **cases_per_pop** per county.
- Third, we create **tn_total**, which sums all the county data into state totals. This data frame also freshly calculates the **cases_per_pop** for the whole state.
- Finally, we create **tn_current**, a data frame to hold only the most recent TN COVID-19 statistics.

```r
#Create Tennessee data frame
tn_df <- US_data %>%
  filter(Province_State == "Tennessee", cases > 0, Admin2 != "Out of TN", Admin2 != "Unassigned") %>%
  group_by(date, Admin2)

#Group data frame by county and create two new fields - mortality rate and cases per population
tn_counties <- tn_df %>%
  group_by(Admin2, date) %>%
  mutate(mortality = deaths / cases, cases_per_pop = cases / Population) %>%
  select(Admin2, date, cases, deaths, Population, mortality, cases_per_pop)

#Sum values for cases, deaths and populations
tn_total <- tn_counties %>%
  group_by(date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(cases_per_pop = cases / Population) %>%
  select(date, cases, deaths, Population, cases_per_pop) %>%
  ungroup()

#Create a separate dataframe for the most recent statistics per county
tn_current <- tn_counties %>%
  filter(date == "2023-03-09") %>%
  group_by(Admin2) %>%
  mutate(cases_per_pop = cases/Population) %>%
  select(date, Admin2, cases, deaths, Population, cases_per_pop) %>%
  ungroup()
```

##Data Analysis

```r
#Total Tennessee cases to date
max(tn_total$cases)
```

```
## [1] 2408633
```

```r
#Total Tennessee deaths to date
max(tn_total$deaths)
```

```
## [1] 28720
```

```r
#Total Tennessee County case rates per population
max(tn_total$cases) / max(tn_total$Population)
```

```
## [1] 0.3526976
```

```r
#County with the highest cases per population
tn_current %>% slice_max(cases_per_pop)
```

```
## # A tibble: 1 x 6
##   date       Admin2 cases deaths Population cases_per_pop
##   <date>     <chr>  <int>  <int>      <int>         <dbl>
## 1 2023-03-09 Scott  12228    140      22068         0.554
```

```r
#County with the lowest cases per population
tn_current %>% slice_min(cases_per_pop)
```

```
## # A tibble: 1 x 6
##   date       Admin2  cases deaths Population cases_per_pop
##   <date>     <chr>   <int>  <int>      <int>         <dbl>
## 1 2023-03-09 Stewart  3773     71      13715         0.275
```

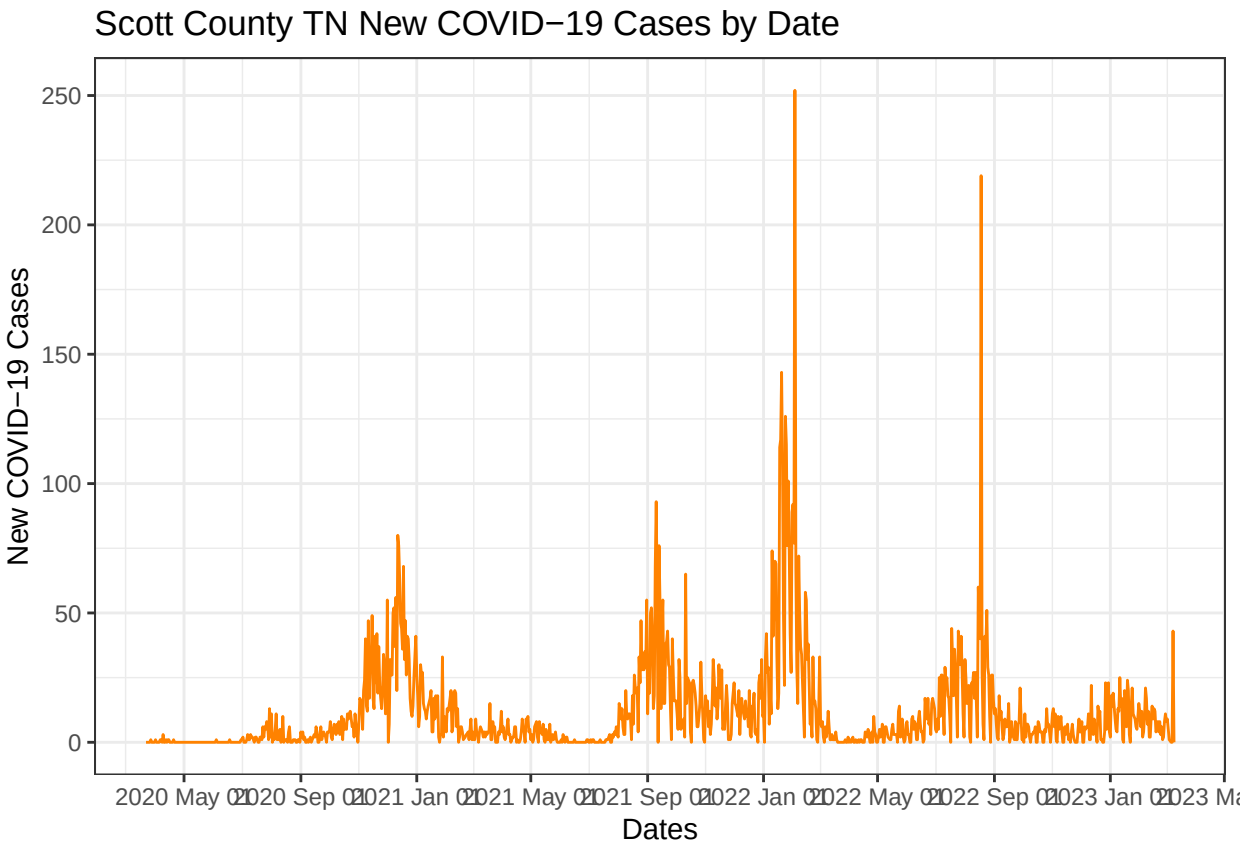
###Findings from Data
- Tennessee has had **2,408,633** cases of COVID so far.
- Tennessee has has a **228,720** of deaths related to COVID so far.
- Tennessee's overall case rate per population is **0.368**.
- **Scott COUNTY** has the **highest COVID case rate per population** in Tennessee at **55%**.
- **Stewart County** has the **lowest COVID case rate per population** in Tennessee at **3%**.

Analysis of Scott County, the Tennessee county with the highest COVID case rate per population

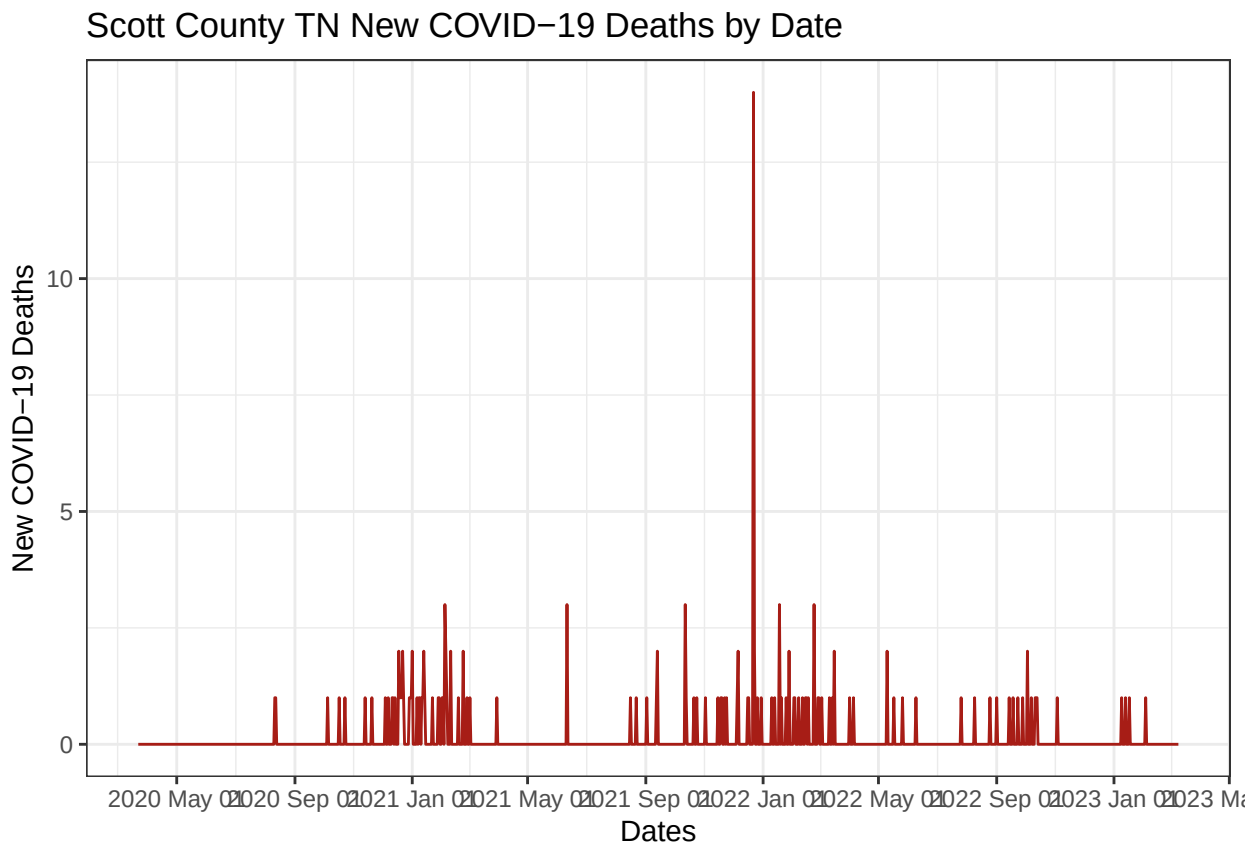
```r
# Create a new data frame for Scott County and add columns for daily new cases and deaths.
scott_county <- tn_counties %>%
  filter(Admin2 == "Scott") %>%
  group_by(Admin2) %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths)) %>%
  select(date, Admin2, cases, deaths, Population, new_cases, new_deaths)

scott_county <- scott_county %>%
  filter(new_cases >= 0, new_deaths >=0)

ggplot(scott_county, aes(x=date)) +
  geom_line(aes(y = new_cases), color="#FF8200") +
  scale_x_date(date_labels = "%Y %b %d", date_breaks = "4 month") +
  theme_bw() +
  labs(x = "Dates",
       y = "New COVID-19 Cases",
       title = "Scott County TN New COVID-19 Cases by Date")
```




```
ggplot(scott_county, aes(x=date)) +
  geom_line(aes(y = new_deaths), color = "#a71d16") +
    scale_x_date(date_labels = "%Y %b %d", date_breaks = "4 month") +
  theme_bw() +
  labs(x = "Dates",
       y = "New COVID-19 Deaths",
       title = "Scott County TN New COVID-19 Deaths by Date")
```

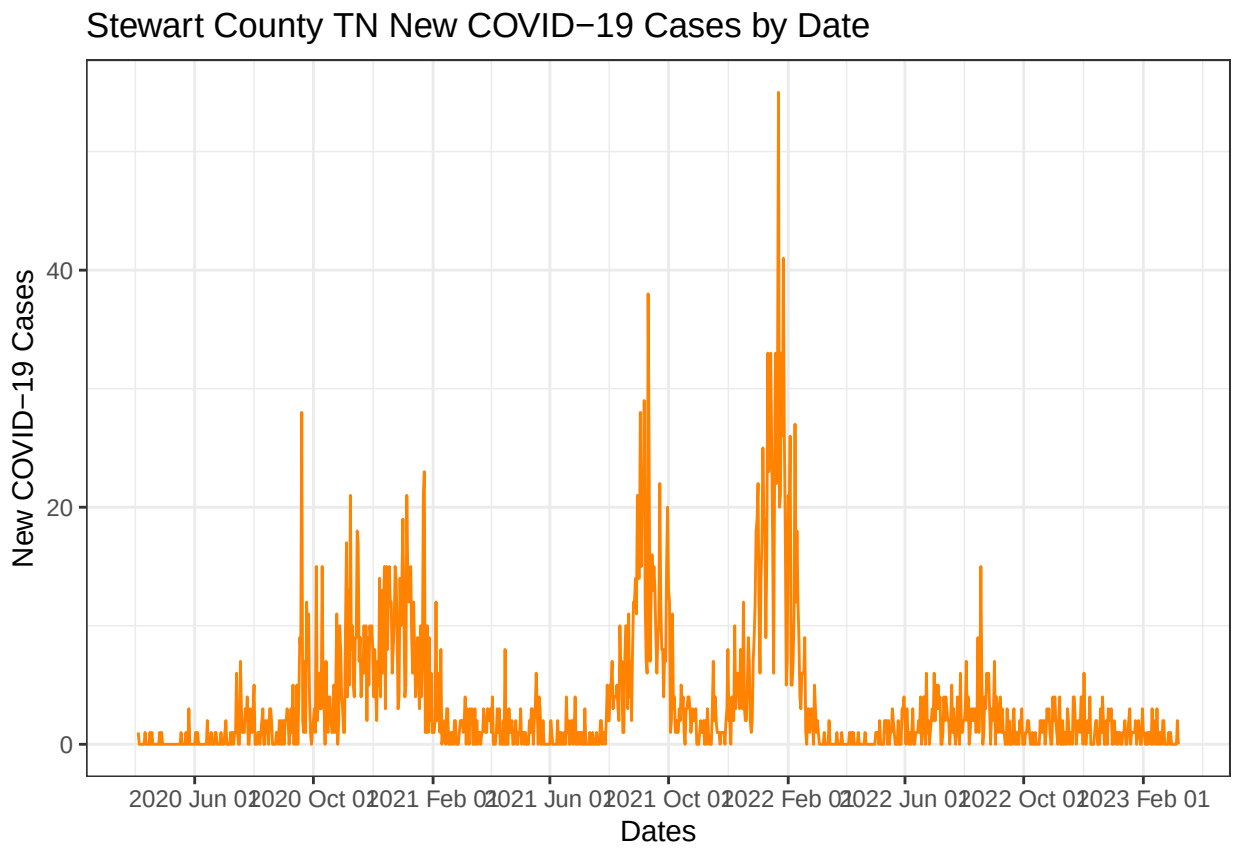


Analysis of Stewart County, the Tennessee county with the lowest COVID case rate per population

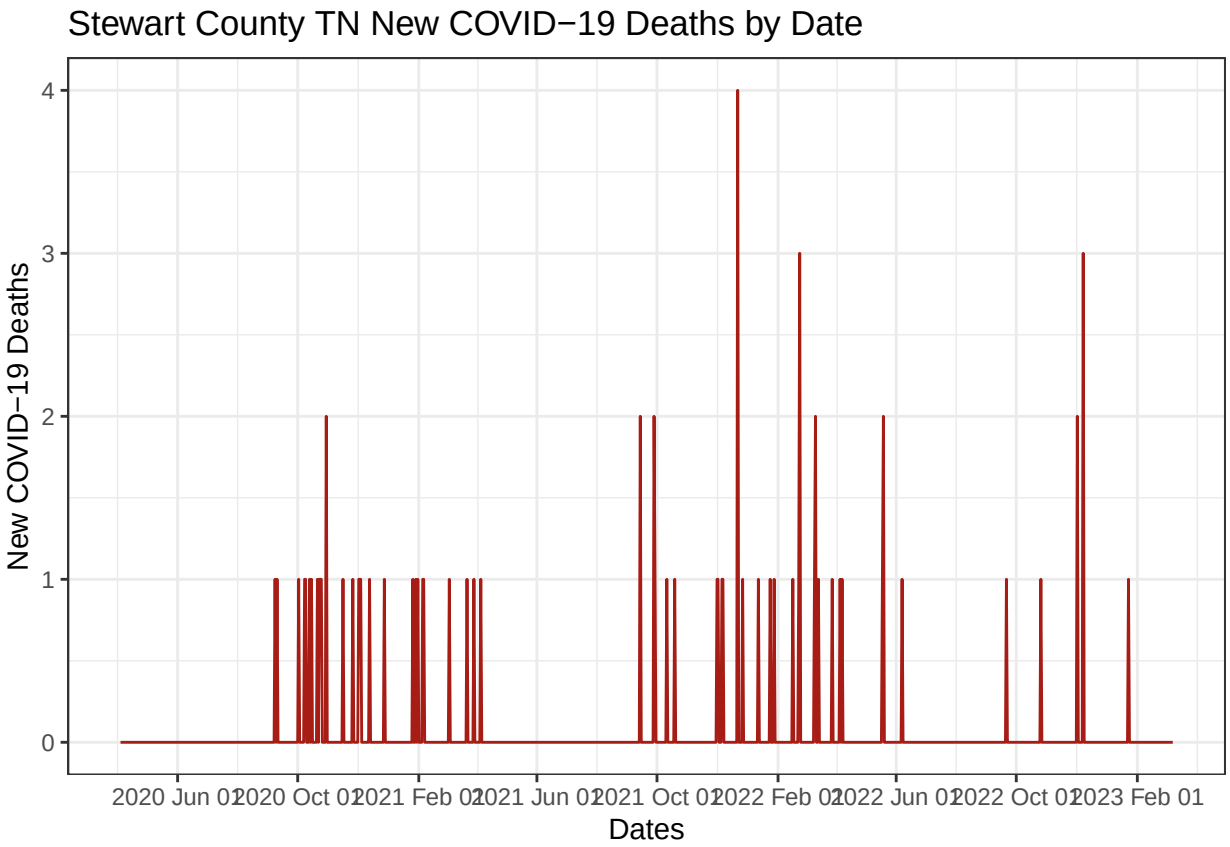
```
# Create a new data frame for Stewart County and add columns for daily new cases and deaths.
stewart_county <- tn_counties %>%
  filter(Admin2 == "Stewart") %>%
  group_by(Admin2) %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths)) %>%
  select(date, Admin2, cases, deaths, Population, new_cases, new_deaths)

stewart_county <- stewart_county %>%
  filter(new_cases >= 0, new_deaths >=0)

ggplot(stewart_county, aes(x=date)) +
  geom_line(aes(y = new_cases), color="#FF8200") +
  scale_x_date(date_labels = "%Y %b %d", date_breaks = "4 month") +
  theme_bw() +
  labs(x = "Dates",
       y = "New COVID-19 Cases",
       title = "Stewart County TN New COVID-19 Cases by Date")
```




```
ggplot(stewart_county, aes(x=date)) +
  geom_line(aes(y = new_deaths), color = "#a71d16") +
    scale_x_date(date_labels = "%Y %b %d", date_breaks = "4 month") +
  theme_bw() +
  labs(x = "Dates",
       y = "New COVID-19 Deaths",
       title = "Stewart County TN New COVID-19 Deaths by Date")
```



Modeling with Linear Regression

Now let's explore whether a Linear Regression Model can predict future Tennessee COVID19 cases and deaths.

Linear regression is a statistical model that is used to predict the value of Y based on an input X. We want to establish a linear relationship between the predictor variable (X) and the outcome variable (Y). A linear relationship is a straight line plotted on a graph.

We test a linear regression model by determining our **bull_hypothesis** and our **alternate hypothesis**. For this experiment:
Null Hypothesis - The cases per population and the deaths related to COVID-19 are not correlated.
Alternate Hypothesis - The cases per population and the deaths related to COVID-19 are correlated.

```
# Prepare the data
tn_county_totals <- tn_counties %>%
  group_by(Admin2) %>%
  summarize(deaths = max(deaths), cases = max(cases), Population = max(Population)) %>%
  mutate(cases_per_pop = cases / Population) %>%
  select(Admin2, cases, deaths, Population, cases_per_pop)

# Build the linear regression model.
lr_model <- lm(deaths ~ cases_per_pop, data = tn_county_totals)

# Display summmary for model analysis.
summary(lr_model)
```

```
##
```

```
## Call:
## lm(formula = deaths ~ cases_per_pop, data = tn_county_totals)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -525.8 -206.4  -83.2   54.6 3195.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1413.0      436.6   3.237  0.00168 **
## cases_per_pop  -2966.8     1159.7  -2.558  0.01214 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 448.2 on 93 degrees of freedom
## Multiple R-squared:  0.06574,    Adjusted R-squared:  0.0557
## F-statistic: 6.544 on 1 and 93 DF,  p-value: 0.01214
```

####Is this model mathematically significant?

- The p-value of the individual predictor variable is 0.012, which is less than 0.05, the standard value for which we would accept our Null Hypothesis. Therefore, our model is statistically significant to ignore our Null Hypothesis, and we should instead turn to our Alternate Hypothesis.

- The model p-value is equal to our individual predictor p-value, as we ran this linear regression with only one variable, which has already been calculated from two separate variables.

- this model is statistically significant, as the p-values are less than 0.05.

####Does this raise additional questions that you should investigate?

- What are the additional factors which could contribute to a high rate of COVID-19 cases per population in Scott County TN? Was there significant vaccine hesitancy in the populace?

- What is the landscape of health centers in Scott County, and how does this compare to Stewart County TN? Is there a major hospital in Stewart County when there are fewer healthcare facilities in surrounding counties,n therefore mandating that individuals in need of medical assistance travel to Scott County?
- Are the spikes in this data tracking with US total cases, or are we witnessing some errors in reporting?

Step 4: Report Conclusion and Sources of Bias

Conclusion I found that Scott County has the highest COVID-19 case rate per population in Tennessee and that Stewart County has the lowest mortality rate in Tennessee. I was able to create a linear regression model which is statistically significant to predict future Tennessee COVID-19 deaths based on the cases per population per county. If I were to continue this analysis, I would likely need a data set of healthcare facility types and staffing numbers to ensure counties with elevated rates of COVID-29 have sufficient healthcare delivery opportunities. If not, then the analysis becomes more difficult as we determine the likelihood of travel to adjacent counties for the purpose of seeking healthcare.

Sources of Bias

COVID-19 has become a politically heated topic, and is therefore prone to bias. I believe COVID-19 was a terrifying disease that swept through the US in 2020, for which we are still feeling impacts today. I mitigated this bias by remaining objective and ensuring the data told the story it could, without my inference. There can also be a bias in the way data is collected. This particular data set had a lot of documentation regarding how it was collected and by which organizations; but lacked sufficient elements to fully deliver the context of the data. There may be some confusion in how COVID-19 cases were reported, but Ito be honest that could be said about most data.