COVID19 Final Project

Kristin Kernler

2022-04-20

Import Libraries

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

Introduction

Project Purpose

This is the Final Project for the course DTSA 5301: Data Science as a Field. We are demonstrating our ability to complete all steps in the data science process by creating a reproducible report on the COVID19 data set from the John Hopkins GitHub site.

Question of Interest

- Which Wisconsin county has the highest COVID19 mortality rate and which Wisconsin county has the lowest COVID19 mortality rate?
- Can we predict future COVID19 cases and deaths in Wisconsin with a Linear Regression Model?

Project Step 1: Describe and Import the Dataset

Data Description

CSSE COVID19 Time Series Data

Two of the datasets are time series tables for the US confirmed cases and deaths, reported at the county level.

The other two datasets are for the global confirmed 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 data is updated once a day around 23:59 (UTC).

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

Import Datasets

```
# All files begin with this string.
url_in <- ("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_co

# Vector containing four 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")

# String concatenate url_in and each of the file names.
urls <- str_c(url_in, file_names)
```

```
# Store each dataset in a variable.
global_cases <- read_csv(urls[1])
global_deaths <- read_csv(urls[2])
US_cases <- read_csv(urls[3])
US_deaths <- read_csv(urls[4])
```

Step 2: Tidy and Transform Data

- We want our variables to have "R friendly" syntax.
- Each date should be on a separate row.
- Dates should be converted into date objects.
- Remove unnecessary columns.
- Handle any missing data.
- Join global cases and global deaths per date.
- Join US cases and US deaths per date.

Tidy Global Data

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

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

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

```
  rename(Country_Region = 'Country/Region',
         Province_State = 'Province/State') %>%
  mutate(date = mdy(date))
```

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

Summary of Global Data (Descriptive Statistics)

```
summary(global)
```

```
##  Province_State     Country_Region          Lat              Long
##  Length:233732      Length:233732      Min.   :-71.950   Min.   :-178.12
##  Class :character   Class :character   1st Qu.:  4.571   1st Qu.: -23.04
##  Mode  :character   Mode  :character   Median : 21.608   Median :  20.92
##                                        Mean   : 20.106   Mean   :  21.96
##                                        3rd Qu.: 41.113   3rd Qu.:  85.24
##                                        Max.   : 71.707   Max.   : 178.06
##                                        NA's   :1646      NA's   :1646
##       date                cases              deaths
##  Min.   :2020-01-22   Min.   :       0   Min.   :     0
##  1st Qu.:2020-08-14   1st Qu.:     263   1st Qu.:     2
##  Median :2021-03-08   Median :    5867   Median :    75
##  Mean   :2021-03-08   Mean   :  524055   Mean   :  9863
##  3rd Qu.:2021-09-30   3rd Qu.:  113390   3rd Qu.:  1810
##  Max.   :2022-04-23   Max.   :80971930   Max.   :991231
##
```

Tidy US Data

```
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_))
```

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

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

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

Summary of US Data (Descriptive Statistics)

```
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:2750466     Length:2750466     Length:2750466     Length:2750466
##  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.   :      0   Min.   :       0   Min.   :    0.0
##  1st Qu.:2020-08-14   1st Qu.:    108   1st Qu.:    9917   1st Qu.:    1.0
##  Median :2021-03-08   Median :   1233   Median :   24892   Median :   21.0
##  Mean   :2021-03-08   Mean   :   8554   Mean   :   99604   Mean   :  136.3
##  3rd Qu.:2021-09-30   3rd Qu.:   4771   3rd Qu.:   64979   3rd Qu.:   82.0
##  Max.   :2022-04-23   Max.   :2859799   Max.   :10039107   Max.   :31924.0
```

Row Description (US Data)

- Each **row** in the US table is a date.

Column Description (US Data)

- **Admin2**: County name
- **Province_State**: State name
- **Country_Region**: US
- **Combined_Key**: Puts together county and state
- **date**: Date in year/month/day format
- **cases**: Total number of COVID19 cases
- **Population**: Population of the County
- **deaths**: Total number of COVID19 related deaths

Step 3: Add Visualizations and Analysis

I am focusing my research on Wisconsin so I will create four new dataframes with only Wisconsin data.

```
# Filter US dataset for only the rows where Province_State is Wisconsin.
wisc <- US %>%
  filter(Province_State == "Wisconsin", cases > 0) %>%
  group_by(date, Admin2)

# Group Wisconsin data by county and add mortality rate column.
wisc_counties <- wisc %>%
  group_by(Admin2, date) %>%
  mutate(mortality_rate = deaths / cases) %>%
  select(Admin2, date, cases, deaths, Population, mortality_rate)

# Sum all Wisconsin county cases, deaths, and populations.
```

```r
wisc_totals <- wisc %>%
  group_by(date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  select(date, cases, deaths, Population) %>%
  ungroup()

# Create a dataframe that contains the most recent statistics for each Wisconsin county.
# Updated April 22, 2022.
current_counties <- wisc_counties %>%
  filter(date == "2022-04-22") %>%
  group_by(Admin2) %>%
  mutate(county_mortality_rate = deaths/cases) %>%
  select(date, Admin2, cases, deaths, Population, county_mortality_rate) %>%
  ungroup()
```

I will now analyze these datasets to find information relevant to my question of interest.

```r
# Total Wisconsin cases to date.
max(wisc_totals$cases)
```

```
## [1] 1601444
```

```r
# Total Wisconsin deaths to date.
max(wisc_totals$deaths)
```

```
## [1] 14403
```

```r
# Wisconsin mortality rate:
max(wisc_totals$deaths) / max(wisc_totals$cases)
```

```
## [1] 0.008993758
```

```r
# Wisconsin county with the highest mortality rate:
current_counties %>% slice_max(county_mortality_rate)
```

```
## # A tibble: 1 x 6
##   date       Admin2 cases deaths Population county_mortality_rate
##   <date>     <chr>  <dbl>  <dbl>      <dbl>                 <dbl>
## 1 2022-04-22 Iron    1470     47       5687                0.0320
```

```r
# Wisconsin county with the lowest mortality rate:
current_counties %>% slice_min(county_mortality_rate)
```

```
## # A tibble: 1 x 6
##   date       Admin2  cases deaths Population county_mortality_rate
##   <date>     <chr>   <dbl>  <dbl>      <dbl>                 <dbl>
## 1 2022-04-22 Buffalo  3531     12      13031                0.00340
```

Wisconsin has had a total of **1,601,444** cases of COVID19.

Wisconsin has had a total of **14,403** COVID19 related deaths.

Wisconsin's mortality rate is **0.009%**.

Iron county has the **highest mortality rate** in Wisconsin at **3.2%**.

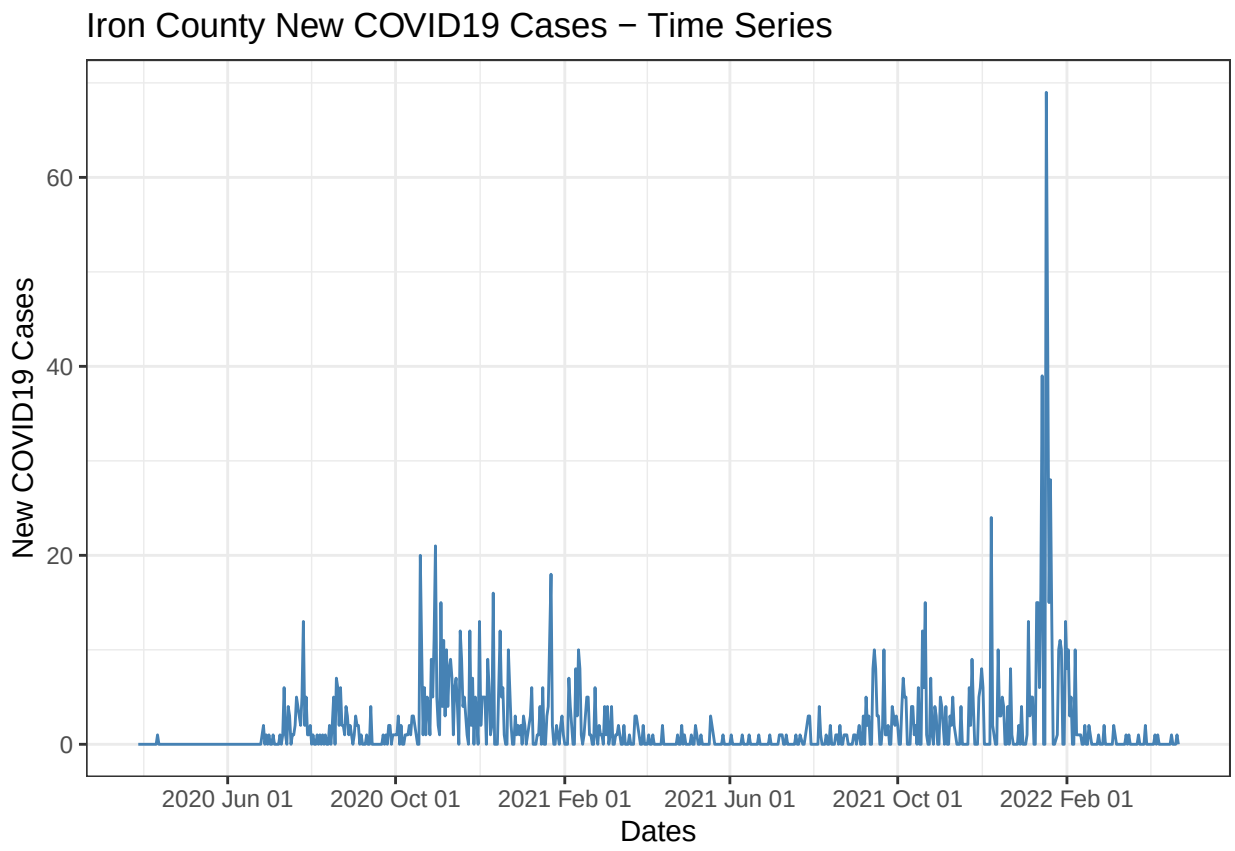
Buffalo county has the **lowest mortality rate** in Wisconsin at **0.0034%**.

Wisconsin County with the Highest Mortality Rate: Iron County

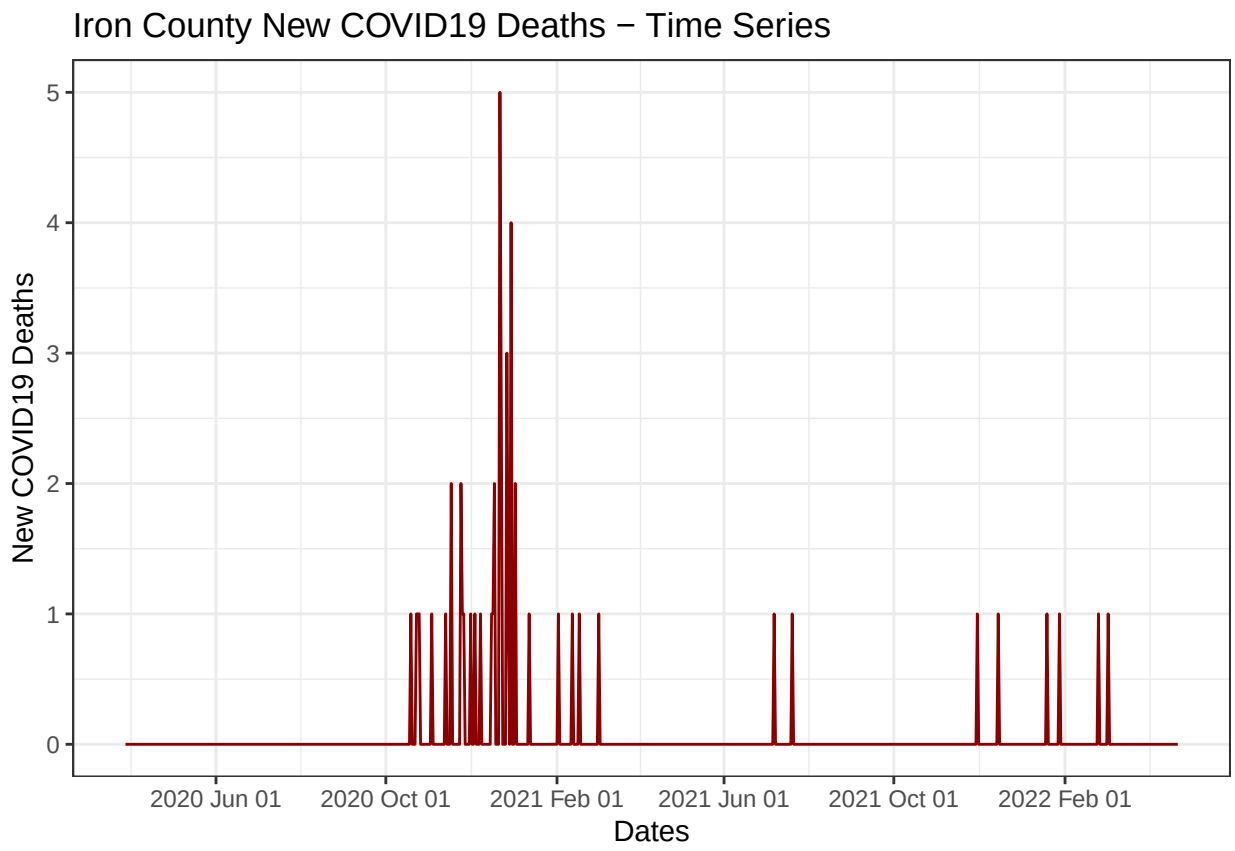
```
# Create a new dataframe for Iron County and add columns for daily new cases and deaths.
iron_county <- wisc_counties %>%
  filter(Admin2 == "Iron") %>%
  group_by(Admin2) %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths)) %>%
  select(date, Admin2, cases, deaths, Population, new_cases, new_deaths)

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

ggplot(iron_county, aes(x=date)) +
  geom_line(aes(y = new_cases), color="steelblue") +
  scale_x_date(date_labels = "%Y %b %d", date_breaks = "4 month") +
  theme_bw() +
  labs(x = "Dates",
       y = "New COVID19 Cases",
       title = "Iron County New COVID19 Cases - Time Series")
```




```
ggplot(iron_county, aes(x=date)) +
  geom_line(aes(y = new_deaths), color = "dark red") +
    scale_x_date(date_labels = "%Y %b %d", date_breaks = "4 month") +
  theme_bw() +
  labs(x = "Dates",
       y = "New COVID19 Deaths",
       title = "Iron County New COVID19 Deaths - Time Series")
```

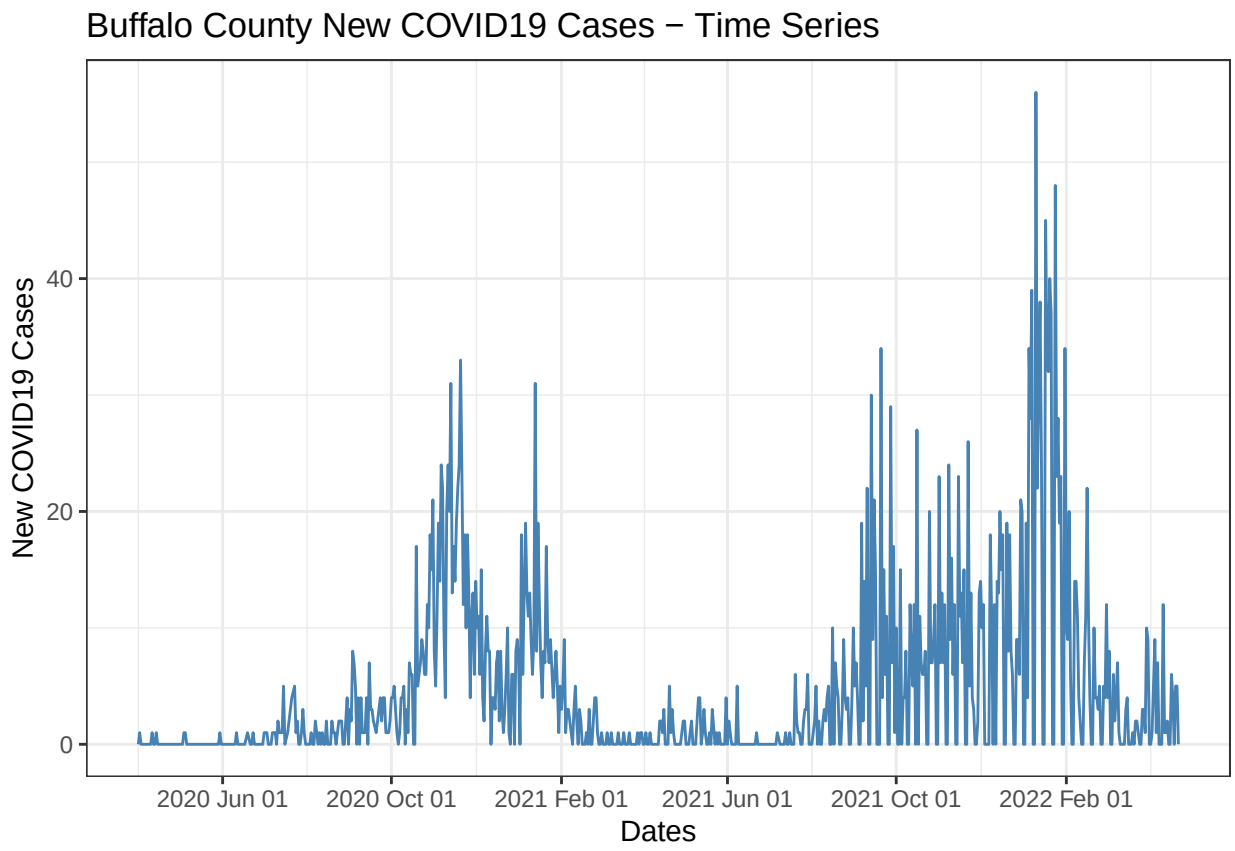


Wisconsin County with the Lowest Mortality Rate: Buffalo County

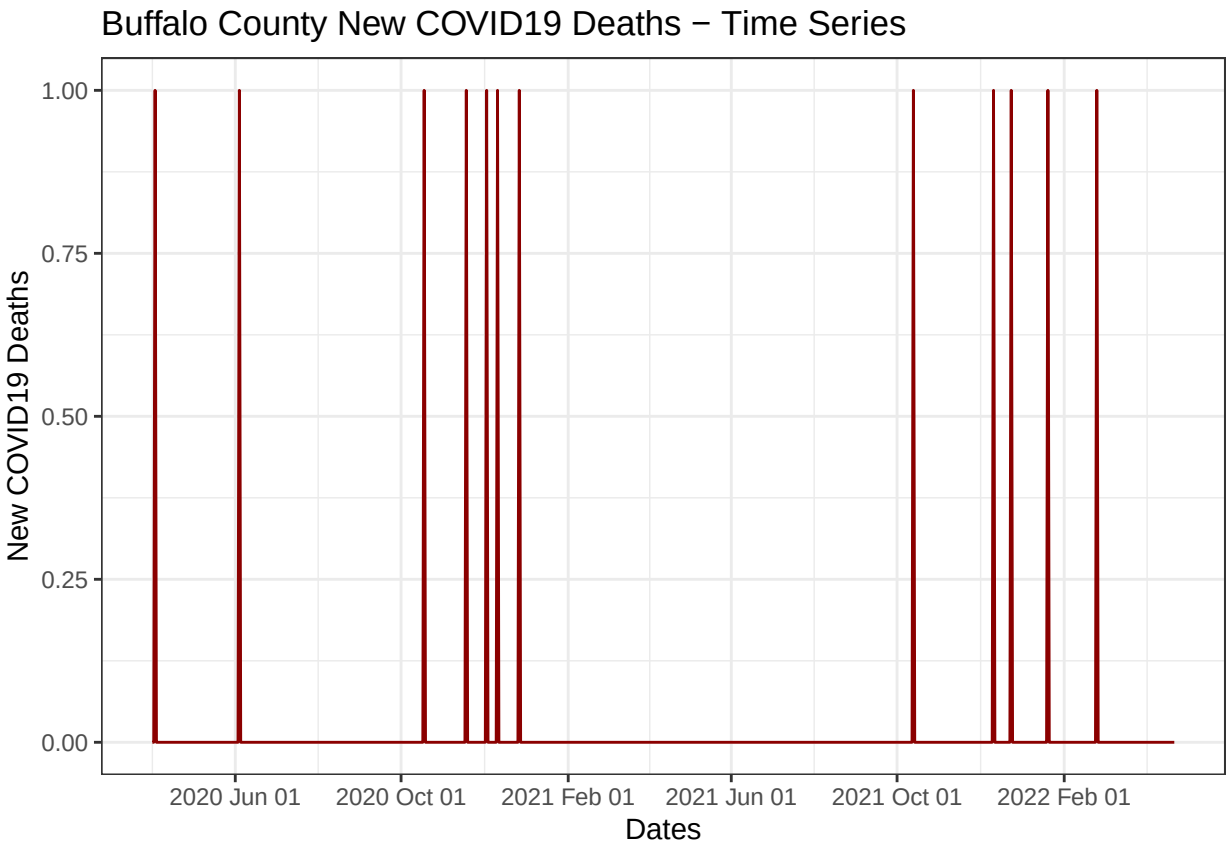
```
# Create a new dataframe for Iron County and add columns for daily new cases and deaths.
buffalo_county <- wisc_counties %>%
  filter(Admin2 == "Buffalo") %>%
  group_by(Admin2) %>%
  mutate(new_cases = cases - lag(cases), new_deaths = deaths - lag(deaths)) %>%
  select(date, Admin2, cases, deaths, Population, new_cases, new_deaths)

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

ggplot(buffalo_county, aes(x=date)) +
  geom_line(aes(y = new_cases), color="steelblue") +
  scale_x_date(date_labels = "%Y %b %d", date_breaks = "4 month") +
  theme_bw() +
  labs(x = "Dates",
       y = "New COVID19 Cases",
       title = "Buffalo County New COVID19 Cases - Time Series")
```




```
ggplot(buffalo_county, aes(x=date)) +
  geom_line(aes(y = new_deaths), color = "dark red") +
    scale_x_date(date_labels = "%Y %b %d", date_breaks = "4 month") +
  theme_bw() +
  labs(x = "Dates",
       y = "New COVID19 Deaths",
       title = "Buffalo County New COVID19 Deaths - Time Series")
```



Modeling with Linear Regression

My objective is to determine whether I can predict future Wisconsin COVID19 cases and deaths with a Linear Regression Model.

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. If a variable's exponent is not equal to 1, there will be a curve.

```
# Prepare the data set for modeling.
wisc_county_totals <- wisc_counties %>%
  group_by(Admin2) %>%
  summarize(deaths = max(deaths), cases = max(cases), Population = max(Population)) %>%
  mutate(cases_per_hundred = 100 * cases / Population, deaths_per_hundred = 100 * deaths / Population ) %>%
  select(Admin2, cases, deaths, Population, cases_per_hundred, deaths_per_hundred)

# Build the linear regression model.
lr_model <- lm(deaths_per_hundred ~ cases_per_hundred, data = wisc_county_totals)

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

```
##
## Call:
```

```
## lm(formula = deaths_per_hundred ~ cases_per_hundred, data = wisc_county_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19393 -0.07111 -0.01378  0.06454  0.54117
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.2698968  0.1187278   2.273   0.0261 *
## cases_per_hundred 0.0005949  0.0043994   0.135   0.8928
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1161 on 70 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0002611,  Adjusted R-squared:  -0.01402
## F-statistic: 0.01828 on 1 and 70 DF,  p-value: 0.8928
```

Is this model mathematically significant?

- The p-value of the individual predictor variable is 0.0261. This is less than 0.05, which means that it could be significant.
- The model p-value is 0.8928, which is larger than 0.05.
- Both p-values need to be less than 0.05 for a linear model to be statistically significant.

Model Observations

- This linear model is **not mathematically significant**.
- There are other variables we should consider when predicting future COVID19 trends in Wisconsin.
- A different model may be more suitable for this analysis.

Does this raise additional questions that you should investigate?

- What are the demographics of Iron County? Does this contribute to the high mortality rate?
- How is the data being reported in Iron County? Are we only seeing deaths for the county's citizens or are we also seeing deaths for the hospital patients in the county? It is possible that the surrounding counties are sending COVID19 positive patients to Iron County hospitals, especially if the nearby counties do not have space in their hospitals.
- What is happening during the spikes in cases and deaths? Did the hospital do a report dump during the large spikes or is this the actual trend?

Step 4: Report Conclusion and Sources of Bias

Conclusion

I found that Iron County has the highest mortality rate in Wisconsin and that Buffalo County has the lowest mortality rate in Wisconsin. I was not able to predict future Wisconsin COVID19 cases and deaths with a Linear Regression Model. I would choose a different dataset with more environmental factors if I wanted to continue this investigation.

Sources of Bias

COVID19 has become a politically heated topic. A strong opinion on this debate could become a source of bias. I mitigated this bias by remaining objective and avoiding assumptions. I need to focus on the data, not the political climate around the pandemic. There can also be a bias in the way data is collected. This particular dataset had a lot of documentation regarding how it was collected and by which organizations. This makes me want to use this data because it is more trustworthy. There may be some confusion in how COVID19 cases were reported, but I think this is a concern with any data involving an infectious disease. It is to be expected, and we just have to work with the available data the best we can.