Covid_Data_Analysis

Introduction

In this project I conduct a preliminary analysis of Covid-19 data for Confirmed Cases and Deaths in the United States. I more specifically look at how the number of Deaths linearly depends on the number of confirmed cases and the date in the state of Colorado.

Importing and Describing the Dataset

The Covid-19 data set that I chose to use was sourced from Johns Hopkins, and the data itself was collected between the dates of 1/22/20 to 3/10/23. I chose to specifically use the time series data sets for Confirmed Cases and Deaths in the United States. Using the US data sets as opposed to the global data sets allows me to conduct exploratory data analysis on fewer rows, and prototype some models before moving to the larger world wide data sets. It is noted on the Johns Hopkins Github website that for the time series data: 'All data is read in from the daily case report'. So I expect the time series data sets to contain an aggregation of the daily report information.

I first load two libraries that allow for useful functions for data wrangling and manipulation of date and time fields.

```
library( tidyverse )
library( lubridate )
```

In the following code I load two datasets: The US confirmed cases dataset and the US deaths dataset. From looking at the first few rows of each, it appears that each data point is stored in a new column as opposed to a new row, so I use the **piviot_longer()** function to change the format of the data to have new observations in each row.

```
# Load data from the Johns Hopkins Github
conf_us_url <- paste( "https://raw.githubusercontent.com/CSSEGISandData/",
    "COVID-19/refs/heads/master/csse_covid_19_data/csse_covid_19_time_series/",
    "time_series_covid19_confirmed_US.csv", sep = "")
raw_us_confirmed <- read_csv( conf_us_url )

deat_us_url <- paste( "https://raw.githubusercontent.com/CSSEGISandData/",
    "COVID-19/refs/heads/master/csse_covid_19_data/csse_covid_19_time_series/",
    "time_series_covid19_deaths_US.csv", sep = "")
raw_us_deaths <- read_csv( deat_us_url )

# First looks at row and column structure:
head( raw_us_deaths, 2 )
```

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

```r
head( raw_us_confirmed, 2 )
```

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

```r
# Using pivot_longer to change to one row per date as opposed to one column:
raw_us_deaths <- pivot_longer(
    data = raw_us_deaths,
    cols = -c( "UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Province_State",
        "Country_Region", "Lat", "Long_", "Combined_Key", "Population" ),
    names_to = "Date",
    values_to = "Deaths" )
glimpse( raw_us_deaths ) # Verify that 'death' data is in row form ...
```

```
## Rows: 3,819,906
## Columns: 14
## $ UID            <dbl> 84001001, 84001001, 84001001, 84001001, 84001001, 84001~
## $ iso2           <chr> "US", "US", "US", "US", "US", "US", "US", "US", "US", "~
## $ iso3           <chr> "USA", "USA", "USA", "USA", "USA", "USA", "USA", "USA",~
## $ code3          <dbl> 840, 840, 840, 840, 840, 840, 840, 840, 840, 840, 840, ~
## $ FIPS           <dbl> 1001, 1001, 1001, 1001, 1001, 1001, 1001, 1001, 1001, 1~
## $ 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", "~
## $ Lat            <dbl> 32.53953, 32.53953, 32.53953, 32.53953, 32.53953, 32.53~
## $ Long_          <dbl> -86.64408, -86.64408, -86.64408, -86.64408, -86.64408, ~
## $ Combined_Key   <chr> "Autauga, Alabama, US", "Autauga, Alabama, US", "Autaug~
## $ Population     <dbl> 55869, 55869, 55869, 55869, 55869, 55869, 55869, 55869,~
## $ Date           <chr> "1/22/20", "1/23/20", "1/24/20", "1/25/20", "1/26/20", ~
## $ Deaths         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```r
# Using pivot_longer to change to one row per date as opposed to one column:
raw_us_confirmed <- pivot_longer(
    data = raw_us_confirmed,
    cols = -c( "UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Province_State",
        "Country_Region", "Lat", "Long_", "Combined_Key" ),
    names_to = "Date",
    values_to = "Confirmed" )
glimpse( raw_us_confirmed ) # Verify that 'confirmed' data is in row form ...
```

```
## Rows: 3,819,906
## Columns: 13
## $ UID            <dbl> 84001001, 84001001, 84001001, 84001001, 84001001, 84001~
## $ iso2           <chr> "US", "US", "US", "US", "US", "US", "US", "US", "US", "~
## $ iso3           <chr> "USA", "USA", "USA", "USA", "USA", "USA", "USA", "USA",~
## $ code3          <dbl> 840, 840, 840, 840, 840, 840, 840, 840, 840, 840, 840, ~
## $ FIPS           <dbl> 1001, 1001, 1001, 1001, 1001, 1001, 1001, 1001, 1001, 1~
## $ 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", "~
## $ Lat            <dbl> 32.53953, 32.53953, 32.53953, 32.53953, 32.53953, 32.53~
## $ Long_          <dbl> -86.64408, -86.64408, -86.64408, -86.64408, -86.64408, ~
## $ Combined_Key   <chr> "Autauga, Alabama, US", "Autauga, Alabama, US", "Autaug~
## $ Date           <chr> "1/22/20", "1/23/20", "1/24/20", "1/25/20", "1/26/20", ~
## $ Confirmed      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Looking at the data, now stored in a longer format as opposed to the wider format I first loaded, there are 3,819,906 rows or observations, and 14 and 13 columns for the deaths and confirmed data respectively. Many of the columns seem to be identical between the two datasets, so I expect to merge or join these two datasets after tidying each.

Tidying and Transforming the Dataset

I tidied the data in the following steps. The R code and output for each of these steps can be seen just below.

- After verifiing that the 'iso3' and 'iso2' columns duplicate the same information, i.e. that given one value you then know what the other must be, I removed the 'iso2' column from each dataset.
- Because I chose to use the US only versions of the datasets, the 'Country_Region' value is "US" for all data. As that is not useful information for analysis, I removed the 'Country_Region' columns for both datasets.
- The 'code3' column also duplicates the 'iso3' information, similar to how 'iso3' and 'iso2' behaved. After verifying that there is a one-to-one correspondence between 'iso3' and 'code3' I chose to remove the 'code3' column from each dataset.
- The 'UID' columns were removed from both datasets as this information is duplicated in the 'FIPS' columns.
- The 'Lat' and 'Long' columns were removed because I don't intend to do and location or GIS analysis in this report.

```r
# Unique values of the 'iso2' and 'iso3' columns:
unique( raw_us_confirmed$iso2 )
unique( raw_us_confirmed$iso3 )

# Unique values of the 'iso2' and 'iso3' columns:
unique( raw_us_deaths$iso2 )
unique( raw_us_deaths$iso3 )

# Do all of the entries in ISO2 and ISO3 align?
c(
    all( which(raw_us_confirmed$iso2 == "AS") ==
        which(raw_us_confirmed$iso3 == "ASM") ),
    all( which(raw_us_confirmed$iso2 == "GU") ==
        which(raw_us_confirmed$iso3 == "GUM") ),
    all( which(raw_us_confirmed$iso2 == "MP") ==
        which(raw_us_confirmed$iso3 == "MNP") ),
    all( which(raw_us_confirmed$iso2 == "VI") ==
        which(raw_us_confirmed$iso3 == "VIM") )
)

# Duplicate the align check for 'deaths' data:
c(
    all( which(raw_us_deaths$iso2 == "AS") ==
        which(raw_us_deaths$iso3 == "ASM") ),
    all( which(raw_us_deaths$iso2 == "GU") ==
        which(raw_us_deaths$iso3 == "GUM") ),
    all( which(raw_us_deaths$iso2 == "MP") ==
        which(raw_us_deaths$iso3 == "MNP") ),
    all( which(raw_us_deaths$iso2 == "VI") ==
        which(raw_us_deaths$iso3 == "VIM") )
)
```

```
## [1] "US"  "AS"  "GU"  "MP"  "PR"  "VI"
## [1] "USA" "ASM" "GUM" "MNP" "PRI" "VIR"
## [1] "US"  "AS"  "GU"  "MP"  "PR"  "VI"
## [1] "USA" "ASM" "GUM" "MNP" "PRI" "VIR"
## [1] TRUE TRUE TRUE TRUE
## [1] TRUE TRUE TRUE TRUE
```

```r
# Remove ISO2 as it duplicates ISO3 column for confirmed data:
raw_us_confirmed <- select( .data = raw_us_confirmed, -iso2 )

# Remove ISO2 as it duplicates ISO3 column for 'deaths' data:
raw_us_deaths <- select( .data = raw_us_deaths, -iso2 )
```

```r
# Verify that 'Country_Region' contains only "US" for both data sets.
unique( raw_us_confirmed$Country_Region )
unique( raw_us_deaths$Country_Region )

# Drop the 'Country_Region' column from both data sets as it contains no
# useful information for this analysis:
raw_us_confirmed <- select( .data = raw_us_confirmed, -Country_Region )
raw_us_deaths <- select( .data = raw_us_deaths, -Country_Region )
```

```
## [1] "US"
## [1] "US"
```

```r
# The code3 column seems to duplicate the iso3 information. Checking:
unique( raw_us_confirmed$code3 )
```

```
## [1] 840  16 316 580 630 850
```

```r
unique( raw_us_confirmed$iso3 )
```

```
## [1] "USA" "ASM" "GUM" "MNP" "PRI" "VIR"
```

```r
# Indices where each 'code3' value exists:
idx_16 <- which( raw_us_confirmed$code3 == 16)
idx_316 <- which( raw_us_confirmed$code3 == 316)
idx_580 <- which( raw_us_confirmed$code3 == 580)
idx_630 <- which( raw_us_confirmed$code3 == 630)
idx_850 <- which( raw_us_confirmed$code3 == 850)

# Finding 'iso3' values, and checking that entries all have the same value:
head( raw_us_confirmed$iso3[idx_16] )
```

```
## [1] "ASM" "ASM" "ASM" "ASM" "ASM" "ASM"
```

```r
all( raw_us_confirmed$iso3[idx_16] == "ASM" )
```

```
## [1] TRUE
```

```r
head( raw_us_confirmed$iso3[idx_316] )
```

```
## [1] "GUM" "GUM" "GUM" "GUM" "GUM" "GUM"
```

```r
all( raw_us_confirmed$iso3[idx_316] == "GUM" )
```

```
## [1] TRUE
```

```r
head( raw_us_confirmed$iso3[idx_580] )
```

```
## [1] "MNP" "MNP" "MNP" "MNP" "MNP" "MNP"
```

```r
all( raw_us_confirmed$iso3[idx_580] == "MNP" )
```

```
## [1] TRUE
```

```r
head( raw_us_confirmed$iso3[idx_630] )
```

```
## [1] "PRI" "PRI" "PRI" "PRI" "PRI" "PRI"
```

```r
all( raw_us_confirmed$iso3[idx_630] == "PRI" )
```

```
## [1] TRUE
```

```r
head( raw_us_confirmed$iso3[idx_850] )
```

```
## [1] "VIR" "VIR" "VIR" "VIR" "VIR" "VIR"
```

```r
all( raw_us_confirmed$iso3[idx_850] == "VIR" )
```

```
## [1] TRUE
```

```r
# Duplicate this analysis for the 'deaths' data:
unique( raw_us_deaths$code3 )
```

```
## [1] 840  16 316 580 630 850
```

```r
all( raw_us_confirmed$iso3[ which( raw_us_deaths$code3 == 16 ) ] == "ASM" )
```

```
## [1] TRUE
```

```r
all( raw_us_confirmed$iso3[ which( raw_us_deaths$code3 == 316 ) ] == "GUM" )
```

```
## [1] TRUE
```

```r
all( raw_us_confirmed$iso3[ which( raw_us_deaths$code3 == 580 ) ] == "MNP" )
```

```
## [1] TRUE
```

```r
all( raw_us_confirmed$iso3[ which( raw_us_deaths$code3 == 630 ) ] == "PRI" )
```

```
## [1] TRUE
```

```r
all( raw_us_confirmed$iso3[ which( raw_us_deaths$code3 == 850 ) ] == "VIR" )
```

```
## [1] TRUE
```

```r
# Remove all of the index vectors created: cleaning work space.
rm( idx_16, idx_316, idx_580, idx_630, idx_850  )
```

```r
# Remove the 'code3' columns as they duplicate the 'iso3' columns:
raw_us_confirmed <- select( .data = raw_us_confirmed, -code3 )
raw_us_deaths <- select( .data = raw_us_deaths, -code3 )

# Removing UID columns as they are duplicated in the 'FIPS' information:
raw_us_confirmed <- select( .data = raw_us_confirmed, -UID )
raw_us_deaths <- select( .data = raw_us_deaths, -UID )

# Removing the 'Lat' and 'Long_' columns as I will not be doing any GIS analysis
# for this particular project:
raw_us_confirmed <- select( .data = raw_us_confirmed, -Lat )
raw_us_deaths <- select( .data = raw_us_deaths, -Lat )
raw_us_confirmed <- select( .data = raw_us_confirmed, -Long_ )
raw_us_deaths <- select( .data = raw_us_deaths, -Long_ )
```

After removing the duplicate data, I transformed the data to make analysis easier in the following steps:

- I first join the two data sets, effectively adding the 'Population' and 'Deaths' columns to the Confirmed data set.
- I transform date column using the lubridate library.
- I remove any rows/observations that have zero confirmed cases. This removes approximately 345,000 observations.

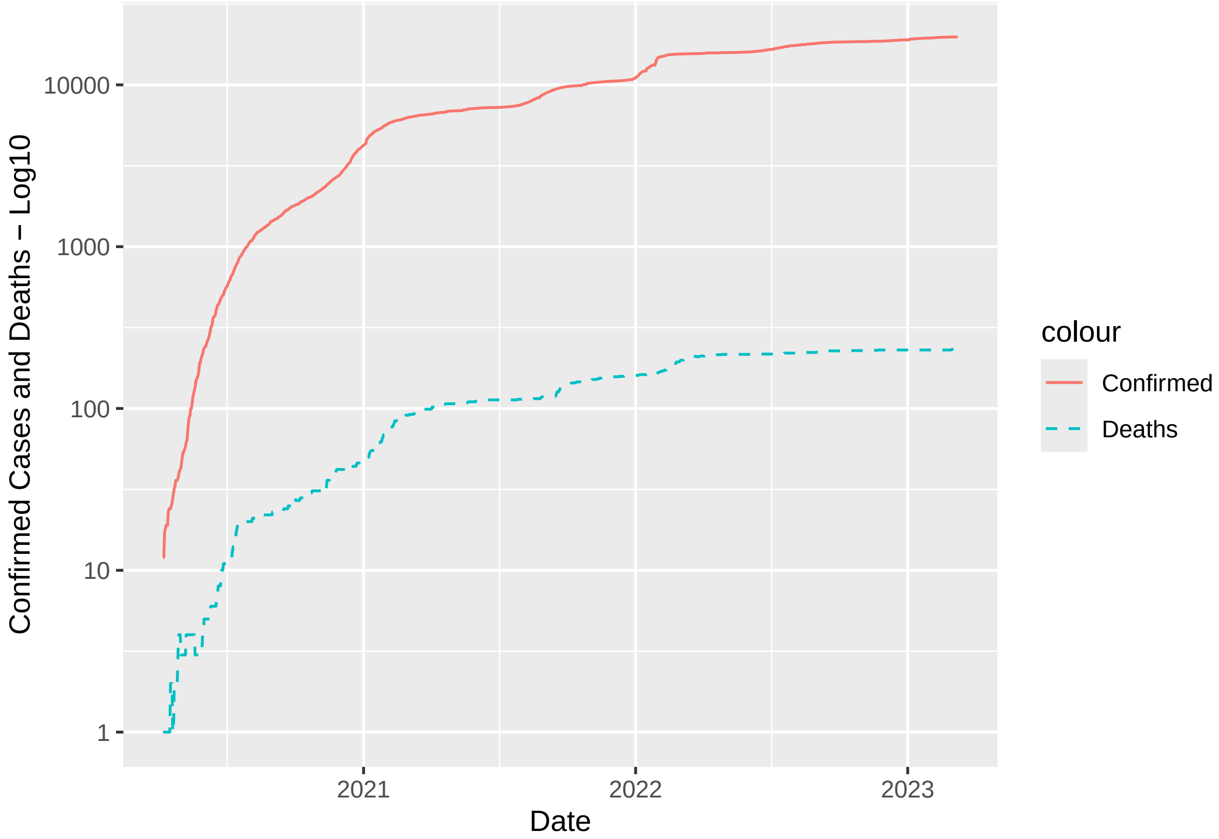
After these operations, I believe that the dataset is ready for analysis:

```r
# Create clean data that combines both confirmed cases and deaths information:
clean_data <- left_join( raw_us_confirmed, raw_us_deaths,
    by = c("iso3", "FIPS", "Admin2", "Province_State","Combined_Key", "Date") )

# Transform 'Date' using lubridate
clean_data <- mutate( .data = clean_data, Date_Lub = mdy(Date),
    .keep = "unused" )

# Filter data to remove any rows when there are zero confirmed cases:
clean_data <- filter( .data = clean_data, Confirmed > 0 )

# Remove the raw datasets, freeing up memory:
rm( raw_us_confirmed, raw_us_deaths )
```

Analysis and Visualization of Data:

The question that I am initially interested in is if there is a linear relationship between the number of confirmed cases and deaths, i.e. if Covid has some relatively fixed amount of deaths for each 100 cases of Covid. I am also curious if the number of deaths depended on other variables: population, time, location, etc. To study these questions I started with a single location (FIPS = 1001, which corresponds to Autauga county Alabama) and plotted both Confirmed cases and Deaths vs. Date. The y axis is formatted in a log10 scale so that details can be seen on both lines but the data itself is not log transformed.

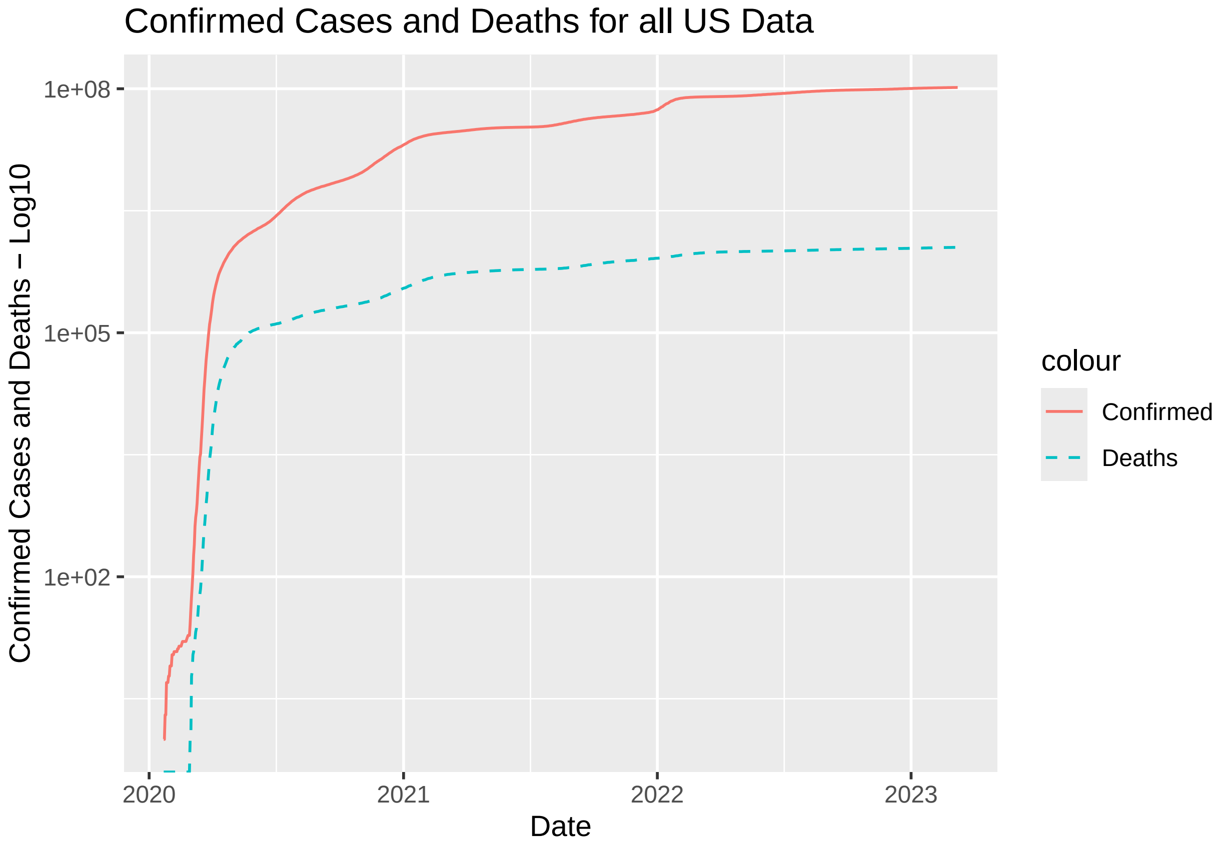
```
# Plotting Confirmed and Deaths vs. Date
data_subset <- filter( .data = clean_data, FIPS == 1001, Deaths > 0)
# dev.new() # Useful for plotting outside of the .rmd
ggplot( data = data_subset) +
    geom_line( mapping = aes(x = Date_Lub, y = Confirmed, color = "Confirmed") ) +
    geom_line( mapping = aes(x = Date_Lub, y = Deaths, color = "Deaths"),
        linetype = "dashed" ) +
    scale_y_log10() +
    labs( x = "Date", y = "Confirmed Cases and Deaths - Log10" )
```



From this plot, it looks like for this location there is a relationship between confirmed cases and deaths. To expand this idea to a larger piece of the data, we create a simple linear model where we aggregate all of the confirmed cases and deaths for each date, ignoring location, and fitting a linear model. I also plot the aggregated confirmed cases and deaths before showing the model output. This model's summary and diagnostic plots are shown just below. From the summary, the number of confirmed cases is very significant, with a p-value of less than 2e-16. The R squared value is surprisingly large as well at 0.95, this indicates an almost perfect linear relationship between confirmed cases and deaths. The diagnostic plots however show the model does not fit the basic assumption of a linear model well at all. We would expect the residuals to lack a structure/pattern and the QQ plot to show normal data for typical assumptions of the linear model. Our data does not display those trends; so this simple model needs modification.

```
# All data aggregated
test_data <- clean_data %>%
    group_by( Date_Lub ) %>%
    summarize( Sum_Conf = sum( Confirmed ), Sum_Deaths = sum( Deaths ) )

ggplot( data = test_data ) +
    geom_line( mapping = aes(x = Date_Lub, y = Sum_Conf, color = "Confirmed") ) +
    geom_line( mapping = aes(x = Date_Lub, y = Sum_Deaths, color = "Deaths"),
        linetype = "dashed" ) +
    scale_y_log10() +
    labs( x = "Date", y = "Confirmed Cases and Deaths - Log10",
        title = "Confirmed Cases and Deaths for all US Data" )
```

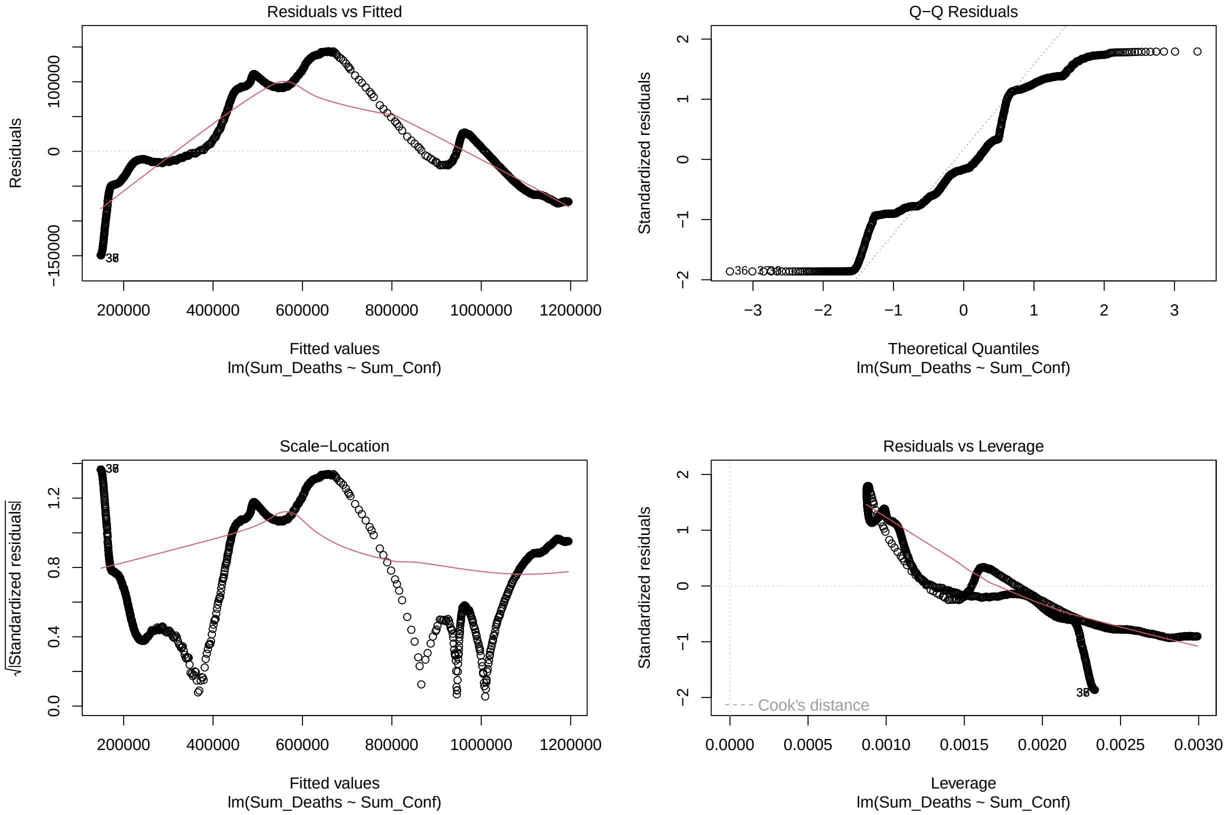
```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```



```
test_model <- lm( data = test_data, Sum_Deaths ~ Sum_Conf )
summary( test_model )
```

```
##
## Call:
## lm(formula = Sum_Deaths ~ Sum_Conf, data = test_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -149147  -62306  -13075   89692  143800
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.491e+05  3.874e+03    38.5   <2e-16 ***
## Sum_Conf    1.008e-02  6.505e-05   154.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80190 on 1141 degrees of freedom
## Multiple R-squared:  0.9546, Adjusted R-squared:  0.9546
## F-statistic: 2.399e+04 on 1 and 1141 DF,  p-value: < 2.2e-16
```

```r
plot(test_model)
```

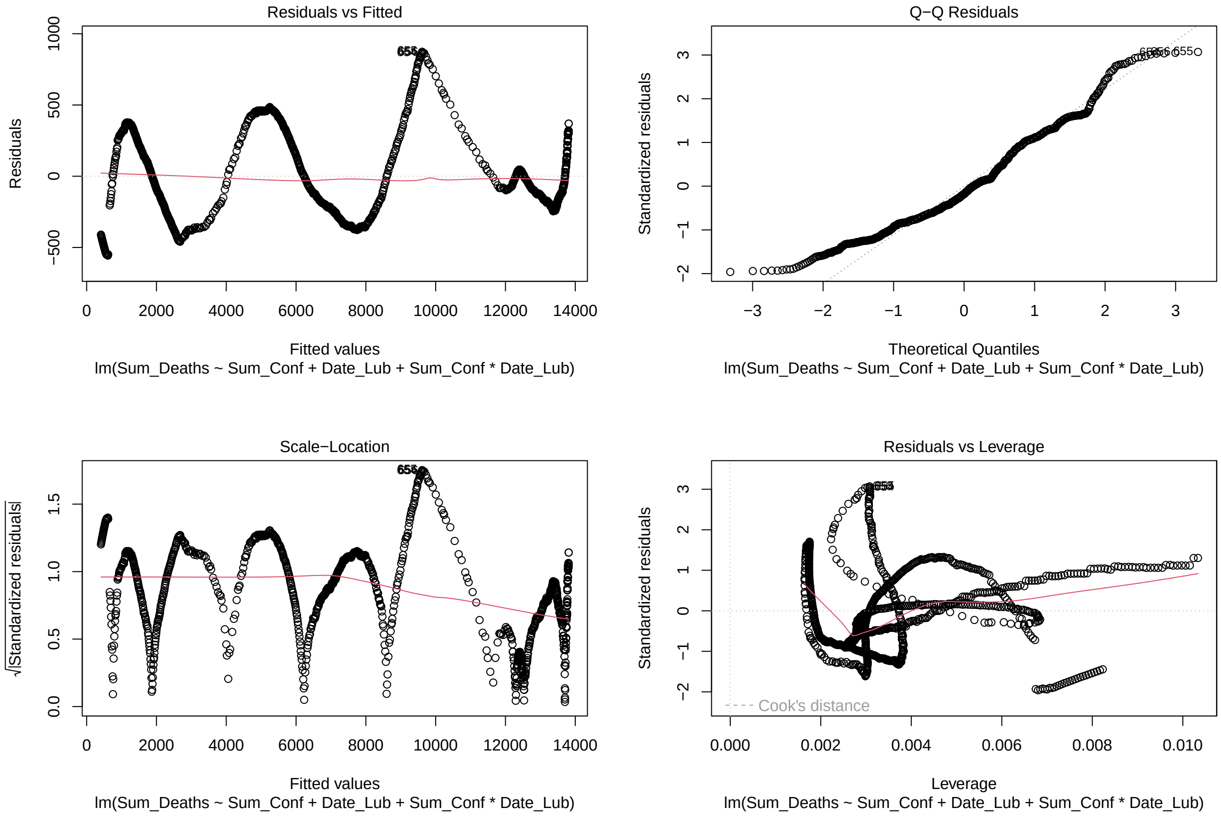


I form a second model by restricting the data to a single state (Colorado), and allowing date to be a predictor variable. I also include the interaction term of date and confirmed as a predictor variable because it would make sense that the number of confirmed cases depends on the date as well. The diagnostic plots from this second linear model show that this model is better, with respect to the QQ plot showing more normality than the the first linear model did. However, there still structure left in the residuals, so there are some underlying trends that I am not accounting for. Nonetheless, with an R squared value 0.99, the chosen predictor variables have accounted for the vast majority of randomness in the data. I think this second simple model answers my curiosity of how confirmed cases and deaths are linearly related. From the coefficients of the model, In the state of Colorado, we expect 8.854 deaths for each additional 100 confirmed cases, if the date was held constant.

```r
# All data aggregated
test_data <- clean_data %>%
    filter( Province_State == "Colorado" ) %>%
    group_by( Date_Lub ) %>%
    summarize( Sum_Conf = sum( Confirmed ), Sum_Deaths = sum( Deaths ) )

test_model <- lm( data = test_data, Sum_Deaths ~ Sum_Conf + Date_Lub +
        Sum_Conf * Date_Lub )
summary( test_model )
```

```
##
## Call:
## lm(formula = Sum_Deaths ~ Sum_Conf + Date_Lub + Sum_Conf * Date_Lub,
##     data = test_data)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -556.9 -213.9  -50.9  211.8  873.5
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.296e+05  2.518e+03  -51.45   <2e-16 ***
## Sum_Conf            8.854e-02  1.134e-03   78.05   <2e-16 ***
## Date_Lub            7.093e+00  1.364e-01   52.01   <2e-16 ***
## Sum_Conf:Date_Lub -4.395e-06  5.866e-08  -74.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 285 on 1096 degrees of freedom
## Multiple R-squared:  0.9962, Adjusted R-squared:  0.9961
## F-statistic: 9.477e+04 on 3 and 1096 DF,  p-value: < 2.2e-16
```

```
plot(test_model) # Diagnostic plots for linear model
```



Further modifications and expansions of this model are certainly possible. One expansion that seem interesting is to fit a similar linear model for each state and compare the coefficients for each model. I would also be curious to do similar models for each country, using the world wide data sets, and compare the resulting coefficients. In the interest of a shorter and more readable presentation, I will hold off on those expansion for a later project.

Consideration of Bias:

There are quite a few possible sources of Bias in the collection of confirmed cases of Covid as well as deaths from this disease. During the pandemic, the Covid vaccination became politically charged, and I suspect the political discord lead to some people not engaging in Covid testing or treatment. Regardless of political leanings, there were certainly some un-reported cases of Covid, but perhaps more un-reported cases amongst different groups of people.

For deaths related to Covid, there are more variables that could lead to bias in that data. Different cites/states/counties could code deaths differently, i.e. one city may code a death as heart attack, while a second may code it a Covid if the patient had Covid and a heart attack too. Having multiple medical conditions makes coding a death as Covid related or not complicated. Thus there may be some locations reporting quite different amounts of Covid deaths even if the situations in both locations are similar.

Lastly, my analysis has my own bias. My last linear model focused on Covid deaths in Colorado. I certainly chose to look locally for answers before looking more globally. I am unfamiliar with many global locations, and any analysis I conducted on data from those locations may be flawed do to my unfamiliarity.

Appendix - Session Information

To wrap up this document, I include session info for the R code, i.e. calling out the current versions of packages for reproducibility.

```
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 22631)
## 
## Matrix products: default
## 
## 
## locale:
## [1] LC_COLLATE=English_United States.utf8 
## [2] LC_CTYPE=English_United States.utf8   
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.utf8    
## 
## time zone: America/Denver
## tzcode source: internal
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] lubridate_1.9.3 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4    
##  [5] purrr_1.0.2     readr_2.1.5     tidyr_1.3.1     tibble_3.2.1   
##  [9] ggplot2_3.5.1   tidyverse_2.0.0
## 
## loaded via a namespace (and not attached):
##  [1] bit_4.0.5         gtable_0.3.5      highr_0.11        crayon_1.5.3     
##  [5] compiler_4.4.1    tidyselect_1.2.1  parallel_4.4.1    scales_1.3.0     
##  [9] yaml_2.3.10       fastmap_1.2.0     R6_2.5.1          generics_0.1.3   
## [13] curl_5.2.2        knitr_1.48        munsell_0.5.1     pillar_1.9.0     
## [17] tzdb_0.4.0        rlang_1.1.4       utf8_1.2.4        stringi_1.8.4    
## [21] xfun_0.47         bit64_4.0.5       timechange_0.3.0  cli_3.6.3        
## [25] withr_3.0.1       magrittr_2.0.3    digest_0.6.37     grid_4.4.1       
## [29] vroom_1.6.5       rstudioapi_0.16.0 hms_1.1.3         lifecycle_1.0.4  
## [33] vctrs_0.6.5       evaluate_1.0.0    glue_1.7.0        farver_2.1.2     
## [37] fansi_1.0.6       colorspace_2.1-1  rmarkdown_2.28    tools_4.4.1      
## [41] pkgconfig_2.0.3   htmltools_0.5.8.1
```