Johns-Hopkins Covid 19 Project

Ethan Tucker

2/28/2022

Before proceeding to knit, please ensure that you have the following packages installed:

1. tidyverse
2. lubridate
3. scales

Data loading and Description

I chose to use the data in the repository "csse_covid_19_daily_reports_us" from the Johns-Hopkins COVID-19 GitHub page. There are a whole lot of .csv files, each which contain a daily summary of COVID activity (deaths, recovered, etc.) in 59 US states and territories. We will do an in-depth look at variables once the data are loaded. This is a bit of an arduous process, which makes it a good Data Science project! I ended up building a function that starts with an empty data frame, and sequentially calls full_join() from dplyr to add on all the new rows from each new data frame. Before combining data, we first add a new variable called "date" which was not in the original data. Fortunately the nomenclature of each .csv is identical (mm-dd-yyyy), so we can pull the date from the URL of each individual data file. I include comments with this importing process in the code chunk below. Note that the data importing takes quite a while because it joins 687 different files together - it might take a couple minutes. Go get a coffee / tea! If you can think of a more efficient way to perform this task without loading the data into memory, please leave me a note (I'm new to readr).

```r
# Data citation:
# https://www.thelancet.com/journals/laninf/article/PIIS1473-3099(20)30120-1/fulltext
# accessed from GitHub link below

# Create a "Data" directory if it doesn't exist
if(("Data" %in% list.files(getwd())) == FALSE){
  dir.create("Data")
}

# Store the common path to the GitHub repo wherein we pull data
REPO <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid

## Define two functions to help tidy the data loading process.
#The first function adds appends a new record onto the existing data, the second
#function generates the suffixes for the different data because there are too
#many to write by hand.

add_csv <- function(oldData, newURLSuffix){

    colNames <- names(select(oldData, -Date))
    fullURL <- str_c(REPO, newURLSuffix)
    newData <- read_csv(fullURL, col_types = cols())
```

```r
      if(all(colNames == names(newData))){
        colNames <- names(oldData)
        oldData <- newData %>%
                   mutate(Date = str_sub(newURLSuffix, start = 1, end = 10)) %>%
                     full_join(oldData, by = colNames)
      }

      return(oldData)
}

make_suffixes <- function(){

  dates <- seq(as.Date("2020/4/13"), as.Date("2022/2/27"), by = "day")
  suffixes <- rep("", length(dates))

  for(k in seq_along(dates)){
    suffixes[k] <- str_c(str_sub(dates[k], start = 6, end = 10),
                         "-", str_sub(dates[k], start = 1, end = 4),
                         ".csv")
  }

  return(suffixes)
}

# Read an initial file into covidData in order to establish the column names
covidData <- read_csv(str_c(REPO, "04-12-2020.csv"), col_types = cols()) %>%
             mutate(Date = "04-12-2020")

## Apply functions to create covidData tibble!

suffixes <- make_suffixes()

for(s in 1:210){
  covidData <- add_csv(covidData, suffixes[s])
}

## For some reason the column name "Mortality Rate" changes to
##"Case_Fatality_Ratio" and "People_Tested" changes to "Total_Test_Results"
##after 11/08/2020, and so we need to adjust add_csv() to account for this for
##files 11/09 and onward. Fortunately this is the only such variable name change

add_csv2 <- function(oldData, newURLSuffix){

      colNames <- names(select(oldData, -Mortality_Rate, -People_Tested))
      fullURL <- str_c(REPO, newURLSuffix)
      newData <- read_csv(fullURL, col_types = cols()) %>%
                   mutate(Date = str_sub(newURLSuffix, start = 1, end = 10))

      if(all(colNames == names(
        select(newData, -Case_Fatality_Ratio, -Total_Test_Results)))){

        oldData <- newData %>%
                   full_join(oldData,
```

```
                    by = c("Case_Fatality_Ratio" = "Mortality_Rate",
                           "Total_Test_Results" = "People_Tested",
                           colNames)) %>%
                     rename(Mortality_Rate = Case_Fatality_Ratio,
                            People_Tested = Total_Test_Results)
    }

    return(oldData)
}

## Apply updated add_csv() to finish loading data

for(s in 211:length(suffixes)){
  covidData <- add_csv2(covidData, suffixes[s])
}

# Let's take a look at the uncleaned dataframe:
covidData %>%
  glimpse()
```

```
## Rows: 39,864
## Columns: 19
## $ Province_State       <chr> "Alabama", "Alaska", "American Samoa", "Arizona",~
## $ Country_Region       <chr> "US", "US", "US", "US", "US", "US", "US", "US", "~
## $ Last_Update          <dttm> 2022-02-28 04:31:05, 2022-02-28 04:31:05, 2022-0~
## $ Lat                  <dbl> 32.3182, 61.3707, -14.2710, 33.7298, 34.9697, 36.~
## $ Long_                <dbl> -86.9023, -152.4044, -170.1320, -111.4312, -92.37~
## $ Confirmed            <dbl> 1279364, 238683, 84, 1976890, 819426, 8961636, 13~
## $ Deaths               <dbl> 18225, 1163, 0, 27946, 10497, 85043, 11752, 10465~
## $ Recovered            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Active               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ FIPS                 <dbl> 1, 2, 60, 4, 5, 6, 8, 9, 10, 88888, 11, 12, 13, 9~
## $ Incident_Rate        <dbl> 26092.5093, 32627.2478, 150.9678, 27159.8690, 271~
## $ People_Tested        <dbl> 7240417, 3741734, 2140, 18277270, 4955750, 146476~
## $ People_Hospitalized  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Mortality_Rate       <dbl> 1.4245359, 0.4872572, 0.0000000, 1.4136345, 1.281~
## $ UID                  <dbl> 84000001, 84000002, 16, 84000004, 84000005, 84000~
## $ ISO3                 <chr> "USA", "USA", "ASM", "USA", "USA", "USA", "USA", ~
## $ Testing_Rate         <dbl> 147667.628, 511483.777, 3846.085, 251105.655, 164~
## $ Hospitalization_Rate <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Date                 <chr> "02-27-2022", "02-27-2022", "02-27-2022", "02-27-~
```

Central Questions:

Before cleaning the data, I'd like to address my goals for this project. I have a couple related questions that are well-suited to this dataset:

1. How has the lethality of COVID-19 changed over time? I define lethality as number of deaths per day. This is a function of both virus variant and hospital capacity.

2. What is the relationship between mortality rate and date?

Cleaning:

Now that the data are loaded into R, we need to adjust the variable types and deal with missing values. There are a lot of missing values! We will use filter out `NA` values using the !is.na() command during the wrangling and visualization creation steps. We need to be careful here - we cannot make inferences based on data that does not exist. States (particularly right-leaning) ceased to provide Covid hospitalization records at various points throughout the pandemic largely for PR reasons. Though it seems likely to me that Covid cases increased after this change, we cannot infer such a change from the data. If a value is missing, the entire observation is inherently suspicious and a possible source of bias. That said, I don't have sufficient data to answer my questions if I remove every single line with a missing value in it - doing so reduces the total number of observations by an order of magnitude. As such we will take the upcoming visualizations and models with a pessimistic grain of salt.

For context, I include a glimpse() of the data in the code chunk above which shows that there are $39,864$ observations on 19 variables before cleaning. I also include a glimpse() after cleaning for comparison.

```r
#covidData <- data.frame(covidDataCopy)

## Fix variable classes
covidData$Province_State <- as.factor(covidData$Province_State)
covidData$Date <- parse_date(covidData$Date, format = "%m-%d-%Y")

## Drop variables not needed in project
covidData <- covidData %>%
                select(-Country_Region,
                       -UID,
                       -ISO3,
                       -FIPS,
                       -Lat,
                       -Long_,
                       -Last_Update,
                       -Active,
                       -Recovered
                       )

## Create useful variables for upcoming questions before filtration

covidData <- covidData %>%
              arrange(Province_State, Date) %>%
                mutate(New_Deaths = as.double(NA),
                       New_Hospitalizations = as.double(NA))

for(i in 2:nrow(covidData)){
  #Populate New_Deaths as difference in Deaths between previous day and current
  #day
  if(covidData$Province_State[i-1] == covidData$Province_State[i]){

    covidData$New_Deaths[i] <- covidData$Deaths[i] - covidData$Deaths[i-1]
    covidData$New_Hospitalizations[i] <- covidData$People_Hospitalized[i] -
      covidData$People_Hospitalized[i-1]

  } else{
    next()
  }
}
```

```
# Let's take a look at the final cleaned data frame:
covidData %>%
  glimpse()
```

```
## Rows: 39,864
## Columns: 12
## $ Province_State      <fct> Alabama, Alabama, Alabama, Alabama, Alabama, Alab~
## $ Confirmed           <dbl> 3667, 3870, 4041, 4307, 4465, 4557, 4788, 4978, 5~
## $ Deaths              <dbl> 93, 99, 114, 118, 133, 148, 153, 157, 163, 183, 1~
## $ Incident_Rate       <dbl> 74.78812, 78.92829, 82.41582, 87.84086, 91.06326,~
## $ People_Tested       <dbl> 62029, 64636, 67557, 70823, 73878, 82609, 85753, ~
## $ People_Hospitalized <dbl> 437, 457, 493, 525, 553, 594, 620, 641, 641, 699,~
## $ Mortality_Rate      <dbl> 2.536133, 2.558140, 2.821084, 2.739726, 2.978723,~
## $ Testing_Rate        <dbl> 1265.076, 1318.245, 1377.819, 1444.428, 1506.735,~
## $ Hospitalization_Rate <dbl> 12.26495, 12.23889, 12.47154, 12.88344, 12.72727,~
## $ Date                <date> 2020-04-12, 2020-04-13, 2020-04-14, 2020-04-15, ~
## $ New_Deaths          <dbl> NA, 6, 15, 4, 15, 15, 5, 4, 6, 20, 13, 6, 7, 0, 1~
## $ New_Hospitalizations <dbl> NA, 20, 36, 32, 28, 41, 26, 21, 0, 58, 31, 38, 1,~
```

Comments on cleaned dataset   The cleaning process in this section was rather straightforward - all we did was parse some variables into their proper types, create some new variables that will be useful in our visualizations (I know this is technically wrangling, but I needed to do it here), and remove the variables that won't be used in this project. I removed nine variables total, and added two. I've made a concerted effort to make my code as legible and thus reproducible as possible here. In the wrangling and visualization-creation process I call some filter(!is.na(varName)) commands which also qualify as cleaning. I wanted to make note of that here for posterity.

The final variables in the dataset:

- Province_State: The province or state the record pertains to
- Confirmed: Aggregate confirmed covid cases
- Deaths: Aggregate deaths
- Incident_Rate: A measure of prevalence vs. community vulnerability
- People_Tested: Aggregate tests administered
- Peopl_Hospitalized: Aggregate people admitted to hospitals
- Mortality_Rate: Aggregate liklihood of death with confirmed case
- Testing_Rate: Tests administered per X people (unclear)
- Hospitalization_Rate: Hospitalizations per X people (unclear)
- Date: Variable I made, just the date file corresponds to
- New_Deaths: Variable I made, change in deaths per day
- New_Hospitalizations: Variable I made, change in hospitalizations per day

Question One:

How has the lethality of COVID-19 changed over time?

A picture is worth a thousand words, so I'll jump right into the wrangling/visualizations. As a reminder, I am quantifying lethality as a function of the entire socio-epidemiological event of a pandemic, and thus am using deaths per day as my metric instead of just mortality rate. I created a loop in the cleaning section to do some data mining to calculate new deaths per day and new hospitalizations per day, as those variables were surprisingly not included in the original data set. I will do everything required for any given visualization in as close to a single pipe as possible for readability. The first visualization I have for you dear reader is my favorite of the project: New Deaths per Day against the timeline of the pandemic

```r
## National mortality rate vs. date
covidData %>%
  group_by(Date) %>%
    filter(!is.na(New_Deaths)) %>%
      summarise(Total_New_Deaths = sum(New_Deaths)) %>%
        ggplot(aes(x = Date, y = Total_New_Deaths)) +
        geom_point() +
        scale_x_date(labels = date_format("%m-%Y"), breaks= date_breaks(width = "3 month")) +
        labs(x = "Date",
             y = "New Deaths per Day",
             title = "Aggregate New deaths per Day across all States and Provinces
             \nIsn't this plot so pretty?")
```

Aggregate New deaths per Day across all States and Provinces

Isn't this plot so pretty?
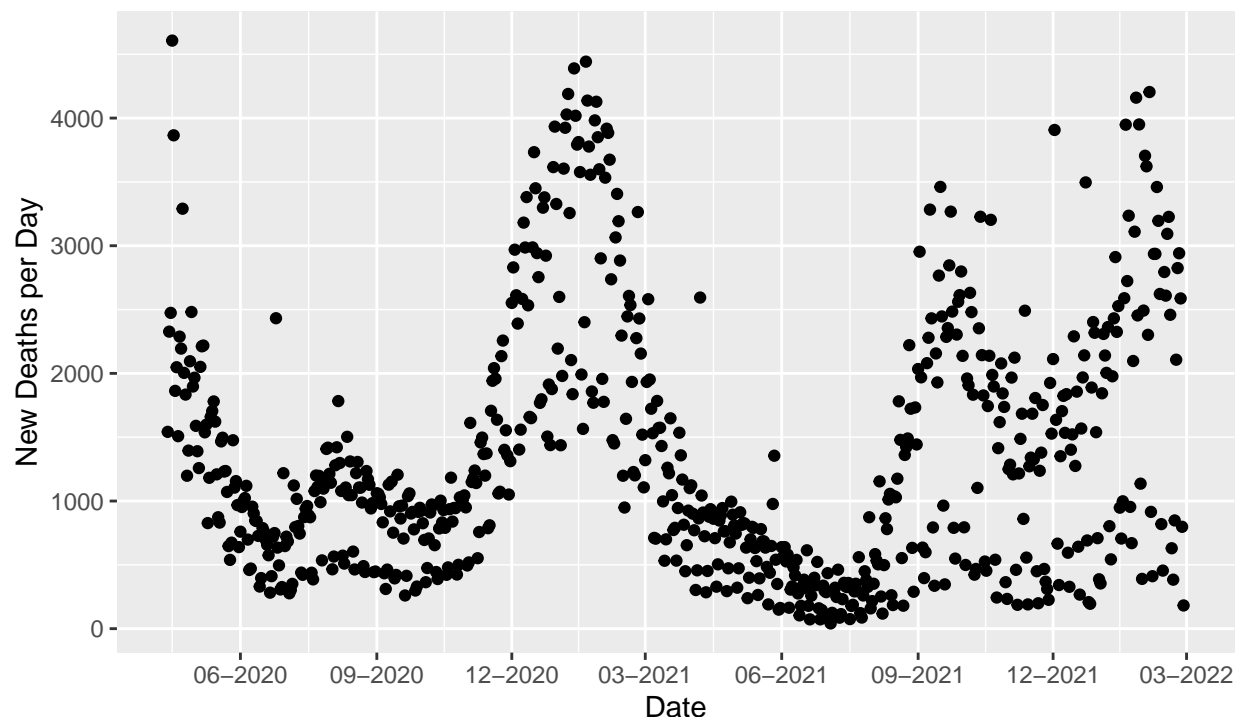
Comments I love how pretty this is. It almost looks like the bifurcation diagram! I had initially fitted a non-linear model to this plot using geom_smooth(method = "gam"), but I think that it ruins the aesthetics. As for analysis of the plot: outside of the bifurcation, there is almost a sinusoidal pattern to New Deaths per day. This may be due to more lethal variants of COVID emerging and vanishing, it may be due to a sinusoidal nature in desire to go outside and have human interaction, and is most likely both. Another interesting observation is that New Deaths per Day are lowest in the summer, and highest in the winter months. This is likely just an illusion caused by low sample size - the data only spans about two full years. Next up we I have isolated New Deaths per Day in four states that have been in the national spotlight for their inept COVID responses. The pattern for each state generally follows the overall pattern above, but are nonetheless interesting to compare and contrast. I fit some non-linear models using geom_smooth() with 95% confidence intervals for this plot.

```r
## New Deaths per day for four interesting states
covidData %>%
```

```
  filter(Province_State %in% c("New York", "Florida", "Texas", "Michigan"),
         !is.na(New_Deaths)) %>%
    ggplot(aes(x = Date, y = New_Deaths, color = Province_State)) +
    scale_x_date(labels = date_format("%m-%Y"), breaks= date_breaks(width = "3 month")) +
    geom_smooth(se = TRUE) +
    labs(x = "Date",
         y = "New Deaths per Day",
         title = "New Deaths Per Day for Four Interesting States",
         color = "State")
```

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

New Deaths Per Day for Four Interesting States
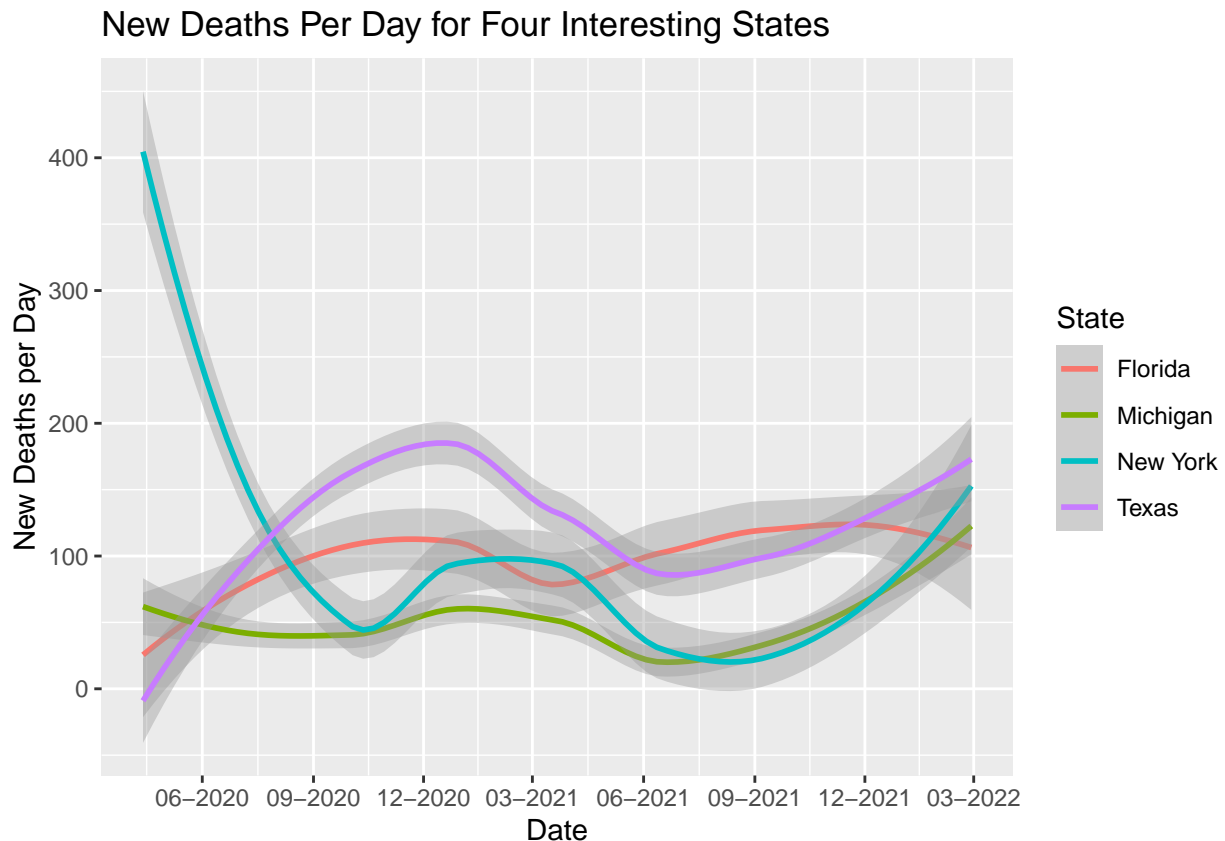
Comments While new deaths per day gives an idea regarding the overall lethality of Coronavirus from a socio-epidemiological perspective, it does not give a sense of how sick covid actually made people. For that we need to look at hospitalization rate. In my cleaning chunk I created a new variable called New_Hospitalizations, which tracked the difference in hospitalizations per day. Again, it was very surprising to me that the hospitalizations variable did not track this, and instead tracked cumulative hospitalizations from the beginning of the data. Weird. Both my following visualization and linear model show that daily hospitalizations were a nigh constant throughout 2020. This makes sense in context, because the total number of hospital beds is a finite number which quickly got capped out in the early days of the pandemic. As such it would be impossible for total hospitalizations per day to exceed the sum number of beds in American hospitals. The data unfortunately stopped recording total hospitalizations in October of 2020, and so I cannot make any inferences or conclusions regarding the impact of the vaccine (which was released after that).

```
## Hospitalization rate vs. date. I have filtered out 2020/06/04 because there
## is a weird singularity that may have to do with bad data entry? Unclear.
```

```
covidData %>%
  filter(!is.na(New_Hospitalizations)) %>%
    group_by(Date) %>%
      summarise(Total_New_Hospitalizations = sum(New_Hospitalizations)) %>%
        filter(Date != as.Date("2020-06-04", format = "%Y-%m-%d")) %>%
          ggplot(aes(x = Date, y = Total_New_Hospitalizations)) +
          geom_point() +
          scale_x_date(labels = date_format("%m-%Y"), breaks= date_breaks(width = "1 month")) +
          geom_smooth(method = "lm", color = "blue") +
          labs(x = "Date",
               y = "Total New Hospitalizations",
               title = "Aggregate New Hospitalizations per Day over all States and Provinces")
```

`geom_smooth()` using formula 'y ~ x'

Aggregate New Hospitalizations per Day over all States and Provinces
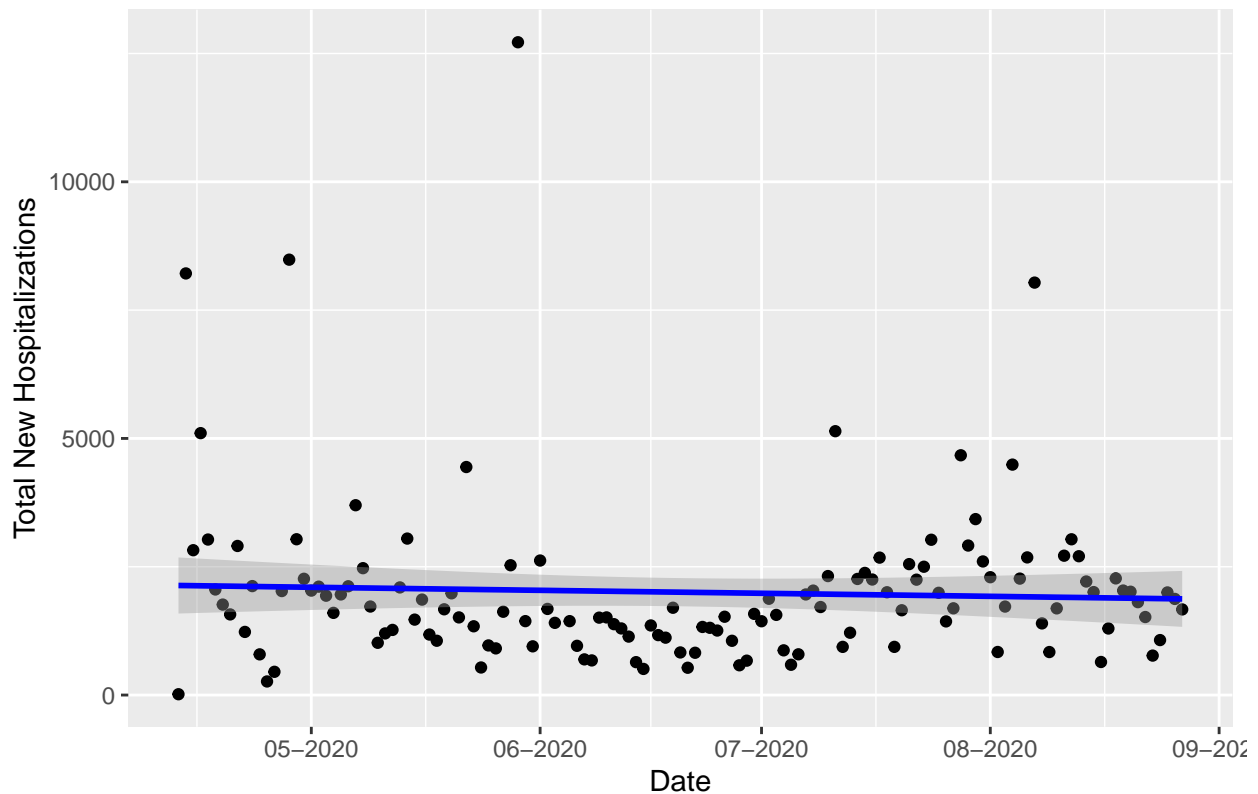
```
#Linear Model
```

```
summary(lm(Total_New_Hospitalizations ~ Date ,
    data = covidData %>%
  filter(!is.na(New_Hospitalizations)) %>%
    group_by(Date) %>%
      summarise(Total_New_Hospitalizations = sum(New_Hospitalizations)) %>%
        filter(Date != as.Date("2020-06-04", format = "%Y-%m-%d")))
)
```

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

```
## lm(formula = Total_New_Hospitalizations ~ Date, data = covidData %>%
##     filter(!is.na(New_Hospitalizations)) %>% group_by(Date) %>%
##     summarise(Total_New_Hospitalizations = sum(New_Hospitalizations)) %>%
##     filter(Date != as.Date("2020-06-04", format = "%Y-%m-%d")))
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -2118.6  -878.8  -332.3   323.2 10672.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37377.163  64561.102   0.579    0.564
## Date           -1.919      3.502  -0.548    0.585
##
## Residual standard error: 1620 on 134 degrees of freedom
## Multiple R-squared:  0.002235,   Adjusted R-squared:  -0.005211
## F-statistic: 0.3002 on 1 and 134 DF,  p-value: 0.5847
```

The simple linear model where response is Total New Hospitalizations and predictor is Date given by the formula:

$$\widehat{y_i} = 37377.163 - 1.919d$$

yields a model F-statistic with p-value 0.5847. For any reasonable selection of α we fail to reject the null hypothesis that the model is insufficient at explaining variability in the response. Clearly we need to add more predictors to get a better hang on the response. The lesson here is that there is not sufficient evidence to suggest that date is a useful predictor for total number of people hospitalized in the US for Covid. Thats not that surprising - the total hospitalization records cut off in October 2020, which is before the vaccine was available to the public. We need more and better data to answer this question decisively.
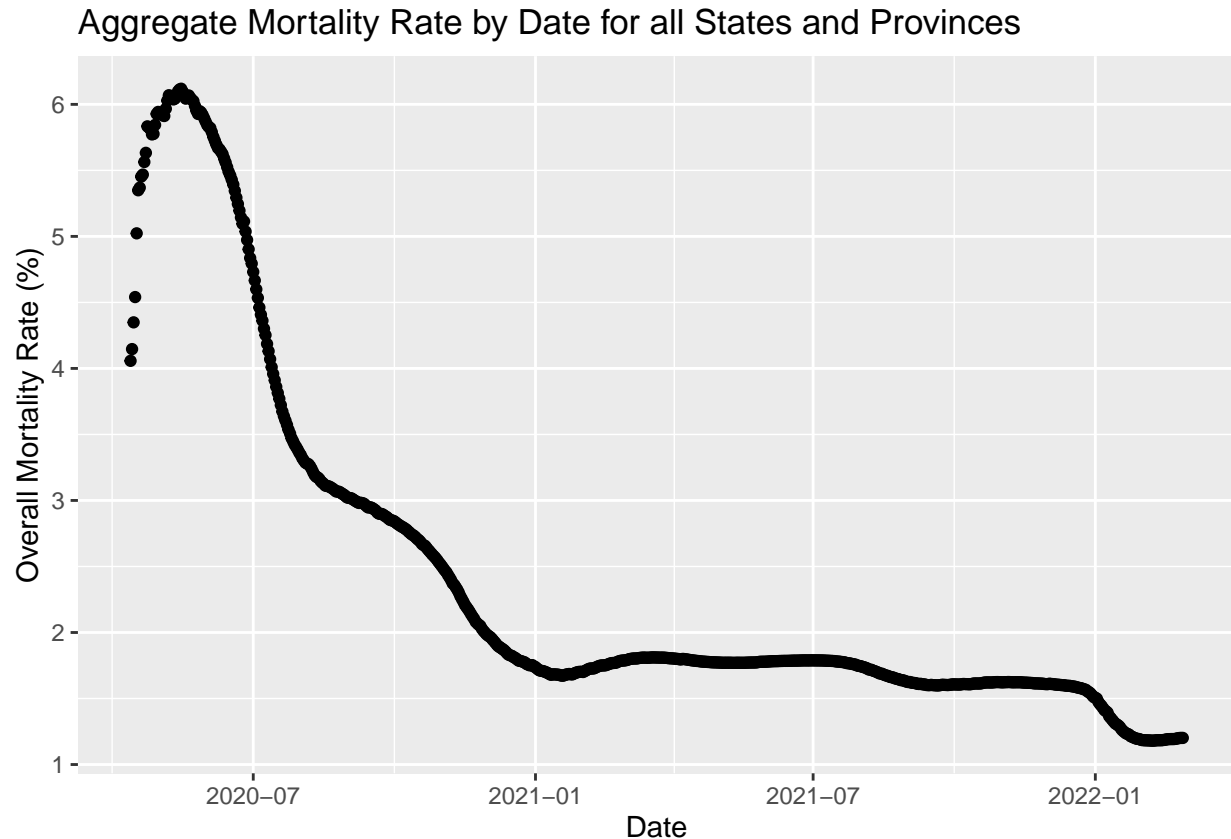
Question Two:

What is the relationship between mortality rate and date?

The last graph left me pretty unsatisfied, and I'm sure you too. To make up for it, I have an alternate means to grasp a similar idea. How has the mortality rate - probability of dying once being confirmed with the disease **AND REPORTING IT** - changed over time? I made heavy use of the group_by() and summarise() methods to create the following graphic. While not as pretty as the first in this project, it provides a much clearer insight into the true behavior of its corresponding statistic. Mortality started out VERY high back with the alpha and beta variants, took a sharp downtick in August of 2020 which is notably when Donald Trump started wearing a mask, and continued to fall as the vaccine became widely available. Another factor that may have influenced the decrease in mortality rate is the changing mortality rates of COVID variants. From the data it is impossible to say *what* caused the effects on these data, this is just an EDA. I'll try to refrain from excessive speculation.

```
## Overall Mortality Rate by Date - calculated by
covidData %>%
  filter(!is.na(Deaths), !is.na(Confirmed)) %>%
    group_by(Date) %>%
      summarise(Overall_Mortality_Rate = sum(Deaths)/sum(Confirmed) * 100) %>%
        ggplot(aes(x = Date, y = Overall_Mortality_Rate)) +
        geom_point() +
        labs(x = "Date",
```
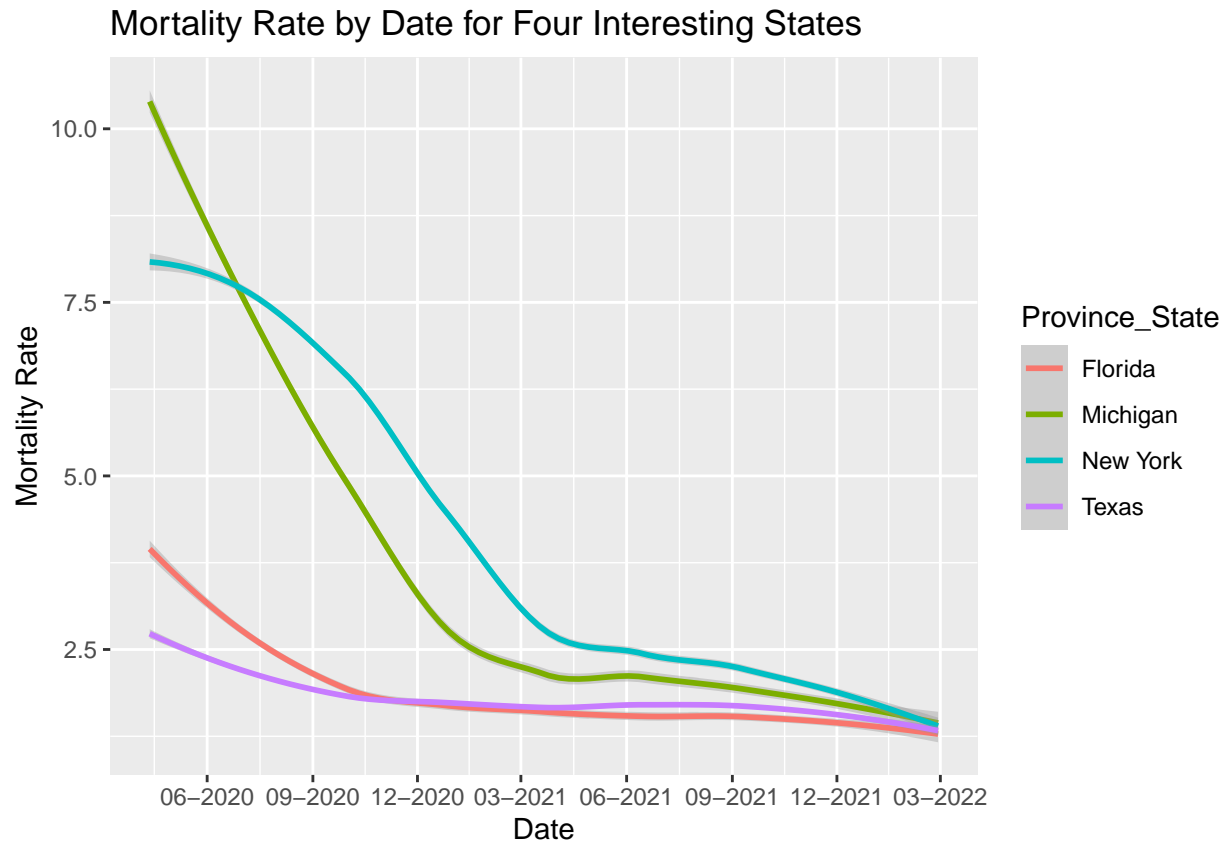
```
            y = "Overall Mortality Rate (%)",
            title = "Aggregate Mortality Rate by Date for all States and Provinces")
```

Aggregate Mortality Rate by Date for all States and Provinces

Comments Just for fun I also ran a quick graph comparing mortality rate by date for the same four states I highlighted in question (1). The output is below:

```
## Mortality Rate by Date for the same four interesting states as in Question 1
covidData %>%
  group_by(Date) %>%
    filter(Province_State %in% c("New York", "Florida", "Texas", "Michigan"),
        !is.na(Mortality_Rate)) %>%
          ggplot(aes(x = Date, y = Mortality_Rate, color = Province_State)) +
          geom_smooth() +
          scale_x_date(labels = date_format("%m-%Y"),
                      breaks= date_breaks(width = "3 month")) +
          labs(x = "Date",
              y = "Mortality Rate",
              title = "Mortality Rate by Date for Four Interesting States")
```

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

Mortality Rate by Date for Four Interesting States

Comments My ggplot "Aggregate Mortality Rate by Date for all States and Provinces" doesn't demonstrate a direct linear relationship, but does seem to exhibit an approximate exponential decay. From linear modelling class I know that we can transform an exponential system into a linear model as follows.

$$\widehat{y_i} = \hat{A}e^{\hat{\beta}_1 x_i}$$

$$\log \widehat{y_i} = \log \hat{A} + \hat{\beta}_1 x_i$$

If we let $\hat{\beta}_1 = \log \hat{A}$, we have arrived at a linear model. I implement this using Aggregate Mortality Rate as response and Date as predictor below.

```
summary(lm( log(Overall_Mortality_Rate) ~ Date,
    data = covidData %>%
  filter(!is.na(Deaths), !is.na(Confirmed)) %>%
    group_by(Date) %>%
      summarise(Overall_Mortality_Rate = sum(Deaths)/sum(Confirmed) * 100)))
```

```
##
## Call:
## lm(formula = log(Overall_Mortality_Rate) ~ Date, data = covidData %>%
##     filter(!is.na(Deaths), !is.na(Confirmed)) %>% group_by(Date) %>%
##     summarise(Overall_Mortality_Rate = sum(Deaths)/sum(Confirmed) *
##         100))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -0.3876 -0.1314  0.0133  0.1238  0.4332
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.752e+01  7.439e-01   50.44   <2e-16 ***
## Date        -1.965e-03  3.976e-05  -49.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2067 on 685 degrees of freedom
## Multiple R-squared:  0.7809, Adjusted R-squared:  0.7806
## F-statistic:  2441 on 1 and 685 DF,  p-value: < 2.2e-16
```

Comments In start contrast to the linear model in question (1), this model is really good! The p value for the F statistic is a rounding error from zero, implying that date certainly has a linear effect on the log of mortality rate. In addition, the adjusted R-squared is just 0.7806, which implies we haven't over-fit our data. Yay!

Conclusion and Bias Analysis

How has the lethality of COVID-19 changed over time? It's complicated. Deaths/day have had a sinusoidal bifurcated manifestation over the course of the pandemic. While it is possible to fit a non-linear model to the extant data, it is unclear how useful such a model would be at explaining, or even predicting, variance in the response. How has mortality rate of COVID changed as the pandemic continued? This we get a pretty satisfying answer to: as time increases linearly, the mortality rate decreases logarithmically.

There are a lot of potential sources of bias in this project. First and foremost, the data is reliant on hospitals and states self-reporting COVID statistics. As has been all too clear over the course of the pandemic, many crucial statistics are unreported or even faked in the case of Florida's governor DeSantis. It is most likely that bad outcomes are under reported in this data. Moreover, the mortality and death statistics are only for those people who were diagnosed with COVID at the time of death. I can almost guarantee that this does not accurately reflect the true death toll of the pandemic. My own personal biases come in the interpretations and analyses of results. I am by nature quite cynical, and am unlikely to believe data even when it is genuine. My speculations that are riddled throughout this project (while in my opinion reasonable) might easily be debunked upon further research into a causal analysis.

This project was a lot work, especially the data import and cleaning phases. I learned the most regarding file management and reading .csv files from GitHub. I built custom functions to assist me in pulling the data, which while inefficient did a pretty good job for my first try. Thanks so much for reading my project, and have a good rest of your day :)

-Ethan Tucker

Data Citation:

- Lancet Article

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
```

```
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] scales_1.1.1    lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0
##  [5] dplyr_1.0.7     purrr_0.3.4     readr_2.1.1     tidyr_1.1.4
##  [9] tibble_3.1.6    ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7       lattice_0.20-45   assertthat_0.2.1 digest_0.6.27
##  [5] utf8_1.2.2       R6_2.5.1          cellranger_1.1.0 backports_1.3.0
##  [9] reprex_2.0.1     evaluate_0.14     httr_1.4.2       highr_0.9
## [13] pillar_1.6.4     rlang_0.4.11      curl_4.3.2       readxl_1.3.1
## [17] rstudioapi_0.13  Matrix_1.3-4      rmarkdown_2.11   splines_4.1.2
## [21] labeling_0.4.2   bit_4.0.4        munsell_0.5.0     broom_0.7.10
## [25] compiler_4.1.2   modelr_0.1.8     xfun_0.24         pkgconfig_2.0.3
## [29] mgcv_1.8-38      htmltools_0.5.1.1 tidyselect_1.1.1 fansi_0.5.0
## [33] crayon_1.4.2     tzdb_0.2.0        dbplyr_2.1.1     withr_2.4.3
## [37] grid_4.1.2       nlme_3.1-153      jsonlite_1.7.2   gtable_0.3.0
## [41] lifecycle_1.0.1  DBI_1.1.1         magrittr_2.0.1   cli_3.1.0
## [45] stringi_1.7.3    vroom_1.5.7       farver_2.1.0     fs_1.5.0
## [49] xml2_1.3.3       ellipsis_0.3.2    generics_0.1.1   vctrs_0.3.8
## [53] tools_4.1.2      bit64_4.0.5       glue_1.4.2       hms_1.1.1
## [57] parallel_4.1.2   yaml_2.2.1        colorspace_2.0-2 rvest_1.0.2
## [61] knitr_1.36       haven_2.4.3
```