COVID 19 Analysis

04/30/2023

Required Packages

Part 1 - Basic Exploration of US Data The New York Times (the Times) has aggregated reported COVID-19 data from state and local governments and health departments since 2020 and provides public access through a repository on GitHub. One of the data sets provided by the Times is county-level data for cumulative cases and deaths each day. This will be your primary data set for the first two parts of your analysis.

County-level COVID data from 2020, 2021, and 2022 has been imported below. Each row of data reports the cumulative number of cases and deaths for a specific county each day. A FIPS code, a standard geographic identifier, is also provided which you will use in Part 2 to construct a map visualization at the county level for a state.

Additionally, county-level population estimates reported by the US Census Bureau has been imported as well. You will use these estimates to calculate statistics per 100,000 people.

```
# Import New York Times COVID-19 data
# Import Population Estimates from US Census Bureau

us_counties_2020 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

```
us_counties_2021 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

```
us_counties_2022 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
```

```
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

```
us_population_estimates <- read_csv("fips_population_estimates.csv")
```

```
## Parsed with column specification:
## cols(
##   STNAME = col_character(),
##   CTYNAME = col_character(),
##   fips = col_double(),
##   STATE = col_double(),
##   COUNTY = col_double(),
##   Year = col_double(),
##   Estimate = col_double()
## )
```

Question 1 Your first task is to combine and tidy the 2020, 2021, and 2022 COVID data sets and find the total deaths and cases for each day since March 15, 2020 (2020-03-15). The data sets provided from the NY Times also includes statistics from Puerto Rico, a US territory. You may remove these observations from the data as they will not be needed for your analysis. Once you have tidied the data, find the total COVID-19 cases and deaths since March 15, 2020. Write a sentence or two after the code block communicating your results. Use inline code to include the `max_date`, `us_total_cases`, and `us_total_deaths` variables. To write inline code use `r`.

```
# Combine and tidy the 2020, 2021, and 2022 COVID data sets.
# Hint: Review the rbind() documentation to combine the three data sets.
#
## YOUR CODE HERE ##
us_counties_total <- rbind(us_counties_2020, us_counties_2021, us_counties_2022)

total <- us_counties_total %>%
  filter(!us_counties_total$state =="Puerto Rico" & !us_counties_total$date < "2020-03-15") %>%
  group_by(date) %>%
  summarise(
    total_cases = sum(cases),
    total_deaths = sum(deaths)
)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
max_date <- tail(total$date, n=1)
us_total_cases <- format(tail(total$total_cases, n = 1), format = "f", big.mark = ",")
us_total_deaths <- format(tail(total$total_deaths, n = 1), format = "f", big.mark = ",")

total
```

```
## # A tibble: 1,022 x 3
##    date       total_cases total_deaths
##    <date>           <dbl>        <dbl>
##  1 2020-03-15        3595           68
##  2 2020-03-16        4502           91
##  3 2020-03-17        5901          117
```

```
##  4 2020-03-18        8345          162
##  5 2020-03-19       12387          212
##  6 2020-03-20       17998          277
##  7 2020-03-21       24507          359
##  8 2020-03-22       33050          457
##  9 2020-03-23       43474          577
## 10 2020-03-24       53899          783
## # ... with 1,012 more rows
```

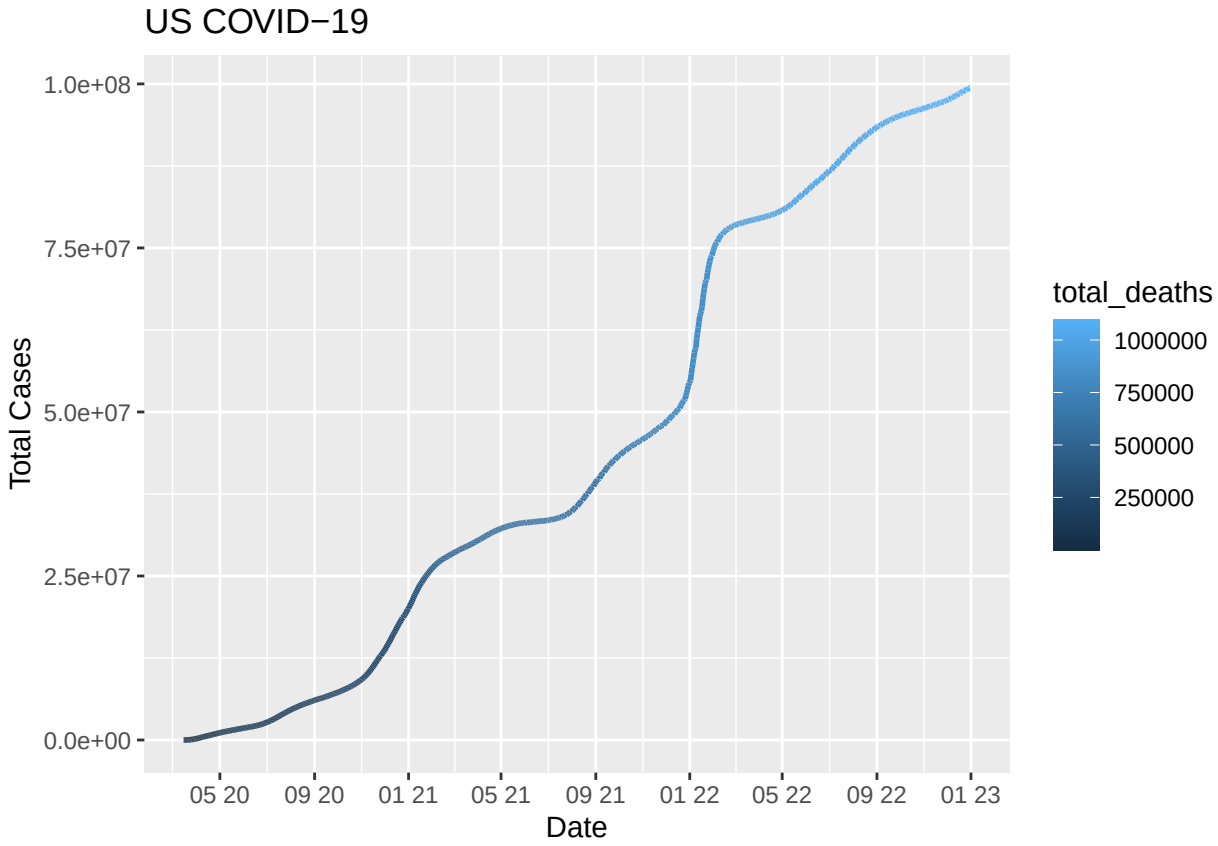
```
# Your output should look similar to the following tibble:
#
#   A tibble: 657 x 3
#       date          total_deaths   total_cases
#      <date>            <dbl>         <dbl>
#   1 2020-03-15           68          3595
#   2 2020-03-16           91          4502
#   3 2020-03-17          117          5901
#   4 2020-03-18          162          8345
#   5 2020-03-19          212         12387
#   6 2020-03-20          277         17998
#   7 2020-03-21          359         24507
#   8 2020-03-22          457         33050
#   9 2020-03-23          577         43474
#  10 2020-03-24          783         53899
# ... with 647 more rows
#
```

– Communicate your methodology, results, and interpretation here –

As of December 31, 2022, `us_total_cases` and `us_total_deaths` occurred at this time. I aggregated the data by date and excluded the states of Puerto Rico and dates lower than "2020-03-15". After that, I summarized the columns total_cases and total _Deaths.

Question 2 Create a visualization for the total number of deaths and cases in the US since March 15, 2020. Before you create your visualization, review the types of plots you can create using the ggplot2 library and think about which plots would be effective in communicating your results. After you have created your visualization, write a few sentences describing your visualization. How could the plot be interpreted? Could it be misleading?

```
# Create a visualization for the total number of US cases and deaths since March 15, 2020.
#
us_total_cases <- format(total$total_cases,format = "f", big.mark = ",")
total %>%
  ggplot(aes(date, total_cases, color=total_deaths)) +
  scale_x_date(date_break = "4 months", date_labels = "%m %y") +
  geom_line(size=1)+
  ggtitle("US COVID-19") +
  ylab(label = "Total Cases") +
  xlab(label = "Date")
```



Looking at the graph, we see that over time, the number of cases increased, but the number of fatalities decreased.

Question 3 While it is important to know the total deaths and cases throughout the COVID-19 pandemic, it is also important for local and state health officials to know the the number of new cases and deaths each day to understand how rapidly the virus is spreading. Using the table you created in Question 1, calculate the number of new deaths and cases each day and a seven-day average of new deaths and cases. Once you have organized your data, find the days that saw the largest number of new cases and deaths. Write a sentence or two after the code block communicating your results.

```
# Create a new table, based on the table from Question 1, and calculate the number of new deaths and ca
#
# Hint: Look at the documentation for lag() when computing the number of new deaths and cases and the s
#
#
## YOUR CODE HERE ##

us_counties_deaths <- total %>%
  mutate(cases_1 = total_cases - (lag(total_cases, 1)),
         deaths_1 = total_deaths - (lag(total_deaths, 1)),
         cases_7 = (total_cases - (lag(total_cases, 7)))/7,
         deaths_7 = (total_deaths - (lag(total_deaths, 7)))/7) %>%
  select(date, total_deaths, total_cases, deaths_1, cases_1,
         deaths_7, cases_7)

us_counties_deaths
```

```
## # A tibble: 1,022 x 7
```

```
##    date       total_deaths total_cases deaths_1 cases_1 deaths_7 cases_7
##    <date>            <dbl>       <dbl>    <dbl>   <dbl>    <dbl>   <dbl>
##  1 2020-03-15           68        3595       NA      NA     NA       NA
##  2 2020-03-16           91        4502       23     907     NA       NA
##  3 2020-03-17          117        5901       26    1399     NA       NA
##  4 2020-03-18          162        8345       45    2444     NA       NA
##  5 2020-03-19          212       12387       50    4042     NA       NA
##  6 2020-03-20          277       17998       65    5611     NA       NA
##  7 2020-03-21          359       24507       82    6509     NA       NA
##  8 2020-03-22          457       33050       98    8543     55.6   4208.
##  9 2020-03-23          577       43474      120   10424     69.4   5567.
## 10 2020-03-24          783       53899      206   10425     95.1   6857.
## # ... with 1,012 more rows
```

```
max_new_cases_date <- us_counties_deaths %>% filter(cases_1 == max(cases_1, na.rm = TRUE)) %>% select(da

max_new_deaths_date <- us_counties_deaths %>% filter(deaths_1 == max(deaths_1, na.rm = TRUE)) %>% selec

max_new_cases_date
```

```
## # A tibble: 1 x 1
##   date
##   <date>
## 1 2022-01-10
```

```
max_new_deaths_date
```

```
## # A tibble: 1 x 1
##   date
##   <date>
## 1 2022-11-11
```

```
#  Your output should look similar to the following tibble:
#
#  date
#  total_deaths     >  the cumulative number of deaths up to and including the associated date
#  total_cases      >  the cumulative number of cases up to and including the associated date
#  delta_deaths_1   >  the number of new deaths since the previous day
#  delta_cases_1    >  the number of new cases since the previous day
#  delta_deaths_7   >  the average number of deaths in a seven-day period
#  delta_cases_7    >  the average number of cases in a seven-day period
#==
#  A tibble: 813 x 7
#    date          total_deaths   total_cases   delta_deaths_1    delta_cases_1  delta_deaths_7  delta
#    <date>           <dbl>          <dbl>         <dbl>             <dbl>        <dbl>           <
#  1 2020-03-15          68          3600           0                    0         NA
#  2 2020-03-16          91          4507          23                  907         NA
#  3 2020-03-17         117          5906          26                 1399         NA
#  4 2020-03-18         162          8350          45                 2444         NA
#  5 2020-03-19         212         12393          50                 4043         NA
#  6 2020-03-20         277         18012          65                 5619         NA
#  7 2020-03-21         360         24528          83                 6516         NA
#  8 2020-03-22         458         33073          98                 8545       55.7              4
#  9 2020-03-23         579         43505         121                10432       69.7              5
# 10 2020-03-24         785         53938         206                10433       95.4              6
# ... with 803 more rows
```

To solve it I added the data of cases 1 and deaths 1 with difference of one week (lag (total_cases, 1). Then I did the same thing with the difference of 7 days. And I selected only the columns that mattered.

```
# Create a new table, based on the table from Question 3, and calculate the number of new deaths and ca

# Hint: To calculate per 100,000 people, first tidy the population estimates data and calculate the US
#
# Hint: look at the help documentation for grepl() and case_when() to divide the averages by the US pop
# For example, take the simple tibble, t_new:
#
#     x     y
#   <int> <chr>
#     1     a
#     2     b
#     3     a
#     4     b
#     5     a
#     6     b
#
#
# To add a column, z, that is dependent on the value in y, you could:
#
# t_new %>%
#   mutate(z = case_when(grepl("a", y) ~ "not b",
#                        grepl("b", y) ~ "not a"))
#

## YOUR CODE HERE ##
us_pop_estimate <- us_population_estimates %>%
  group_by(Year) %>%
  summarize(pop_est = sum(Estimate))
```

Question 4

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# Pop_Estimate 2020
pop_est_2020 <- us_pop_estimate %>%
  filter(Year == "2020") %>%
  select(pop_est)
pop_2020 <- pop_est_2020[[1]]

# Pop_Estimate 2021
pop_est_2021 <- us_pop_estimate %>%
  filter(Year == "2021") %>%
  select(pop_est)

pop_2021 <- pop_est_2021[[1]]

per_100tous <- us_counties_deaths %>%
  mutate(total_deaths_1 = case_when(grepl("2020", date) ~ (total_deaths / pop_2020) * 100000,
                                    grepl("2021", date) ~ (total_deaths / pop_2021) * 100000,
                                    grepl("2022", date) ~ (total_deaths / pop_2021) * 100000),
         total_cases_1 = case_when(grepl("2020", date) ~ (total_cases / pop_2020) * 100000,
```

```
                                grepl("2021", date) ~ (total_cases / pop_2021) * 100000,
                                grepl("2022", date) ~ (total_cases / pop_2021) * 100000),
         delta_deaths_1 = case_when(grepl("2020", date) ~ (deaths_1 / pop_2020) * 100000,
                                    grepl("2021", date) ~ (deaths_1 / pop_2021) * 100000,
                                    grepl("2022", date) ~ (deaths_1 / pop_2021) * 100000),
         delta_cases_1 = case_when(grepl("2020", date) ~ (cases_1 / pop_2020) * 100000,
                                   grepl("2021", date) ~ (cases_1 / pop_2021) * 100000,
                                   grepl("2022", date) ~ (cases_1 / pop_2021) * 100000),
         delta_cases_7 = case_when(grepl("2020", date) ~ (cases_7 / pop_2020) * 100000,
                                   grepl("2021", date) ~ (cases_7 / pop_2021) * 100000,
                                   grepl("2022", date) ~ (cases_7 / pop_2021) * 100000),
         delta_deaths_7 = case_when(grepl("2020", date) ~ (deaths_7 / pop_2020) * 100000,
                                    grepl("2021", date) ~ (deaths_7 / pop_2021) * 100000,
                                    grepl("2022", date) ~ (deaths_7 / pop_2021) * 100000)) %>%

  select(date, total_deaths_1, total_cases_1, delta_deaths_1, delta_cases_1, delta_deaths_7, delta_cases


per_100tous
```

```
## # A tibble: 1,022 x 7
##    date       total_deaths_1 total_cases_1 delta_deaths_1 delta_cases_1
##    <date>              <dbl>         <dbl>          <dbl>         <dbl>
##  1 2020-03-15         0.0205          1.08       NA              NA
##  2 2020-03-16         0.0275          1.36        0.00694         0.274
##  3 2020-03-17         0.0353          1.78        0.00784         0.422
##  4 2020-03-18         0.0489          2.52        0.0136          0.737
##  5 2020-03-19         0.0640          3.74        0.0151          1.22
##  6 2020-03-20         0.0836          5.43        0.0196          1.69
##  7 2020-03-21         0.108           7.39        0.0247          1.96
##  8 2020-03-22         0.138           9.97        0.0296          2.58
##  9 2020-03-23         0.174          13.1         0.0362          3.14
## 10 2020-03-24         0.236          16.3         0.0621          3.14
## # ... with 1,012 more rows, and 2 more variables: delta_deaths_7 <dbl>,
## #   delta_cases_7 <dbl>
```

```
#  Your output should look similar to the following tibble:
#
#  date
#  total_deaths     >  the cumulative number of deaths up to and including the associated date
#  total_cases      >  the cumulative number of cases up to and including the associated date
#  delta_deaths_1   >  the number of new deaths since the previous day
#  delta_cases_1    >  the number of new cases since the previous day
#  delta_deaths_7   >  the average number of deaths in a seven-day period
#  delta_cases_7    >  the average number of cases in a seven-day period
#==
#  A tibble: 657 x 7
#       date         total_deaths   total_cases   delta_deaths_1   delta_cases_1 delta_deaths_7 delta_c
#      <date>              <dbl>         <dbl>          <dbl>           <dbl>           <dbl>         <db
#   1 2020-03-15         0.0205        1.08                0               0              NA           N
#   2 2020-03-16         0.0275        1.36          0.00694         0.274                NA           N
#   3 2020-03-17         0.0353        1.78          0.00784         0.422                NA           N
#   4 2020-03-18         0.0489        2.52           0.0136         0.737                NA           N
#   5 2020-03-19         0.0640        3.74           0.0151          1.22                NA           N
```

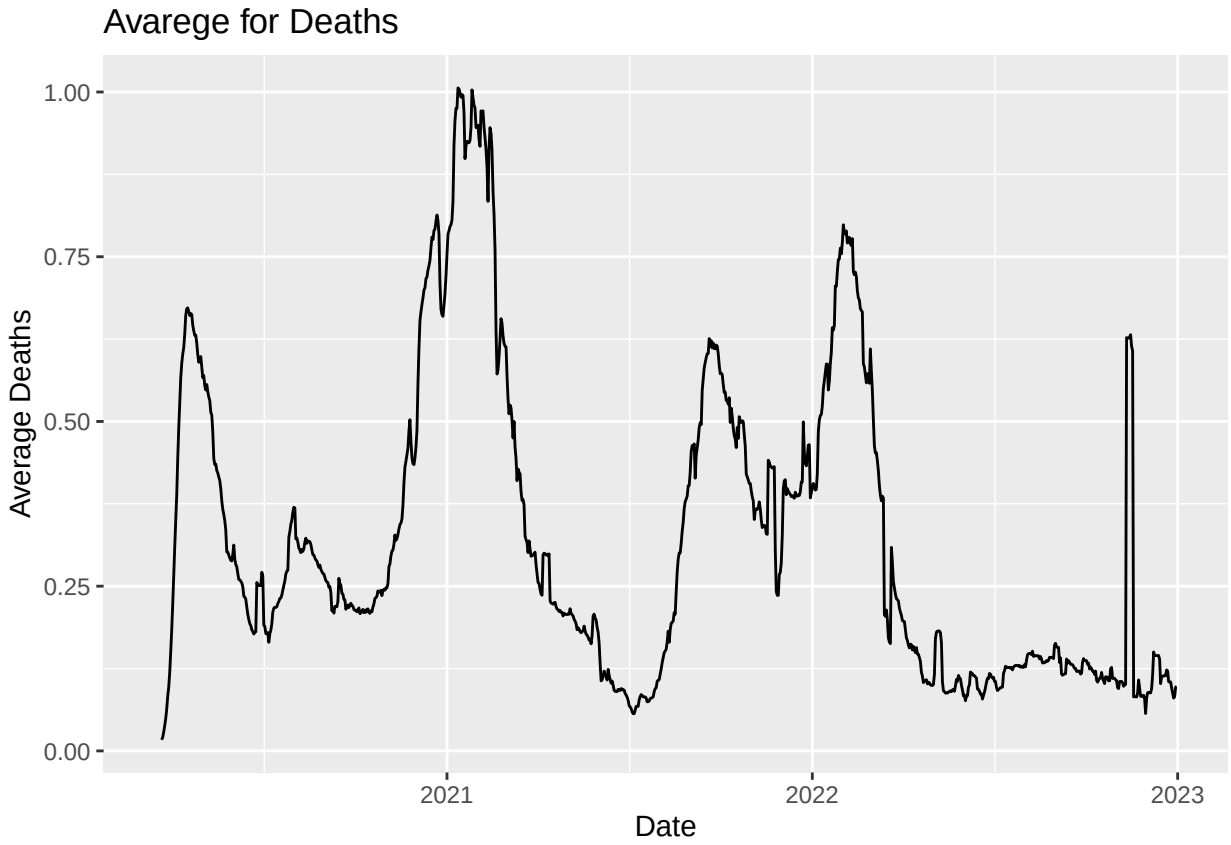
```
#   6 2020-03-20          0.0836          5.43          0.0196          1.69            NA            N
#   7 2020-03-21           0.108          7.39          0.0247          1.96            NA            N
#   8 2020-03-22           0.138          9.97          0.0296          2.58        0.0168         1.2
#   9 2020-03-23           0.174          13.1          0.0362          3.14        0.0209         1.6
#  10 2020-03-24           0.236          16.3          0.0621          3.14        0.0287         2.0
```

This gave rise to a table showing the total number of deaths and cases, new deaths and new cases, and the seven-day average of new deaths and new cases per 100,000 persons. Because they are calculated on a per capita basis, the figures in this table are much smaller than the ones used. These numbers help us interpret these data in the context of the broader U.S. population and provide a broader view of the effects of COVID.

```
# Create a visualization to compare the seven-day average cases and deaths per 100,000 people.
ggplot(per_100tous, aes(date, delta_deaths_7)) + geom_line(color = "black") +
  labs(x= "Date",
       y= "Average Deaths") +
  ggtitle("Avarege for Deaths")
```

Question 5

```
## Warning: Removed 7 row(s) containing missing values (geom_path).
```



Due to the mission population estimates for 2022, this visualization only covers 2020 and 2021. I put that observation in the title. I used an unsized y-axis for this visualization. This is a much clearer representation than the previous question, in which I used a logarithmic transformation on the y-axis. There is a low mortality rate, masking the absolute magnitude, with over 800,000 deaths occurring during the period. This is shown in the chart.