DTSA-5301 - Final Project - COVID-19

R. Garcia

2024-08-18

Data Download

Download the chosen datasets from the identified URLs. I've added a manual `TRUE/FALSE` flag to allow for local storage of the data if desired by the researcher.

```
if (TRUE) {
    base_path <- 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/cs
    urls <- c("time_series_covid19_confirmed_US.csv",
              "time_series_covid19_confirmed_global.csv",
              "time_series_covid19_deaths_US.csv",
              "time_series_covid19_deaths_global.csv")
    world_population <- 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19
} else {
  base_path = "data/"
    urls <- c("time_series_covid19_confirmed_US.csv",
              "time_series_covid19_confirmed_global.csv",
              "time_series_covid19_deaths_US.csv",
              "time_series_covid19_deaths_global.csv")
    world_population <- 'data/UID_ISO_FIPS_LookUp_Table.csv'
}


US_cases <- read_csv(paste0(base_path, urls[1]))
```

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## 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_cases <- read_csv(paste0(base_path, urls[2]))
```

```
## 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.
```

```
US_deaths <- read_csv(paste0(base_path, urls[3]))
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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(paste0(base_path, urls[4]))
```

```
## 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_population <- read_csv(world_population)
```

```
## Rows: 4321 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): iso2, iso3, FIPS, Admin2, Province_State, Country_Region, Combined_Key
## dbl (5): UID, code3, Lat, Long_, Population
##
## 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.
```

Data cleanup

First, we clean up `global_deaths` as per lecture instructions.

```
# remove unnecessary columns
global_deaths <- global_deaths %>% select(-`Lat`, -`Long`)
# convert the dates to rows
global_deaths <- global_deaths %>%
    pivot_longer(cols = -c('Province/State', 'Country/Region'),
                 names_to = "date",
                 values_to = "deaths")
```

Second, `global_cases`

```
# remove unnecessary columns
global_cases <- global_cases %>% select(-`Lat`, -`Long`)
# convert the dates to rows
global_cases <- global_cases %>%
    pivot_longer(cols = -c('Province/State', 'Country/Region'),
                 names_to = "date",
                 values_to = "cases")
```

Now do the same for US statistics

```
# remove unnecessary columns
US_cases <- US_cases %>% select(-`UID`, -`iso2`, -`iso3`, -`code3`,
                                -`FIPS`, -`Admin2`, -`Lat`, -`Long_`)
# convert the dates to rows
US_cases <- US_cases %>%
    pivot_longer(cols = -c('Province_State', 'Country_Region', 'Combined_Key'),
                 names_to = "date",
                 values_to = "cases")
```

We can do some additional cleanup and remove columns that we are not going to be using later on in the analysis.

```
# remove unnecessary columns
US_deaths <- US_deaths %>% select(-`UID`, -`iso2`, -`iso3`, -`code3`,
                                -`FIPS`, -`Admin2`, -`Lat`, -`Long_`)
# convert the dates to rows
US_deaths <- US_deaths %>%
    pivot_longer(cols = -c('Province_State', 'Country_Region', 'Combined_Key', 'Population'),
                 names_to = "date",
                 values_to = "deaths")
```

Combine case and death datasets

Combine the global cases and death datasets and add columns to match the US datasets

```
# join the cases and deaths tables
global <- global_cases %>%
    full_join(global_deaths) %>%
      rename(Country_Region = `Country/Region`,
             Province_State = `Province/State`) %>%
    mutate (date = mdy(date))
```

```
## Joining with ‘by = join_by(‘Province/State‘, ‘Country/Region‘, date)‘
```

```
# filter out the zeros
global <- global %>% filter(cases > 0)

# create a combined key to match US data columns
global <- global %>%
    unite("Combined_Key",
          c("Province_State", "Country_Region"),
```

```
          sep = ", ",
          na.rm = TRUE,
          remove = FALSE)

global_population <- global_population %>%
    select(-c("Lat", "Long_", "Combined_Key", "code3", "iso2",
              "iso3", "Admin2", "UID", "FIPS"))

global <- global %>%
    left_join(global_population, by=c("Province_State", "Country_Region")) %>%
    select(Province_State, Country_Region, date, cases, deaths, Population, Combined_Key)

# output a summary
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:306827      Length:306827      Min.   :2020-01-22   Min.   :        1
##  Class :character   Class :character   1st Qu.:2020-12-12   1st Qu.:     1316
##  Mode  :character   Mode  :character   Median :2021-09-16   Median :    20365
##                                        Mean   :2021-09-11   Mean   :  1032863
##                                        3rd Qu.:2022-06-15   3rd Qu.:   271281
##                                        Max.   :2023-03-09   Max.   :103802702
##
##      deaths           Population        Combined_Key
##  Min.   :      0   Min.   :6.700e+01   Length:306827
##  1st Qu.:      7   1st Qu.:7.866e+05   Class :character
##  Median :    214   Median :6.948e+06   Mode  :character
##  Mean   :  14405   Mean   :2.890e+07
##  3rd Qu.:   3665   3rd Qu.:2.914e+07
##  Max.   :1123836   Max.   :1.380e+09
##                    NA's   :6729
```

Join the US datasets and filter out the 0 case instances.

```
# join the cases and deaths tables
US <- US_cases %>%
    full_join(US_deaths) %>%
    mutate (date = mdy(date))
```

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

```
US <- US %>% filter(cases > 0)

summary(US)
```

```
##  Province_State     Country_Region     Combined_Key           date
##  Length:3474292     Length:3474292     Length:3474292     Min.   :2020-01-22
##  Class :character   Class :character   Class :character   1st Qu.:2020-12-27
##  Mode  :character   Mode  :character   Mode  :character   Median :2021-09-20
##                                                           Mean   :2021-09-19
##                                                           3rd Qu.:2022-06-15
##                                                           Max.   :2023-03-09
```

```
##      cases              Population           deaths
##  Min.   :      1   Min.   :       0   Min.   :    0.0
##  1st Qu.:    687   1st Qu.:   10953   1st Qu.:   10.0
##  Median :   2849   Median :   26248   Median :   47.0
##  Mean   :  15489   Mean   :  104502   Mean   :  205.1
##  3rd Qu.:   9345   3rd Qu.:   68098   3rd Qu.:  137.0
##  Max.   :3710586   Max.   :10039107   Max.   :35545.0
```

Analyze US by State

Instead of simply looking at an individual state, we apply the summary across all states.

```
US_by_state <- US %>%
    group_by(Province_State, Country_Region, date) %>%
    summarize(cases=sum(cases), deaths=sum(deaths),
              Population=sum(Population)) %>%
    mutate(deaths_per_mill = (deaths * 1000000/Population)) %>%
    select(Province_State, "Country_Region", "date", "cases", "deaths",
           "deaths_per_mill", "Population") %>%
    ungroup()
```

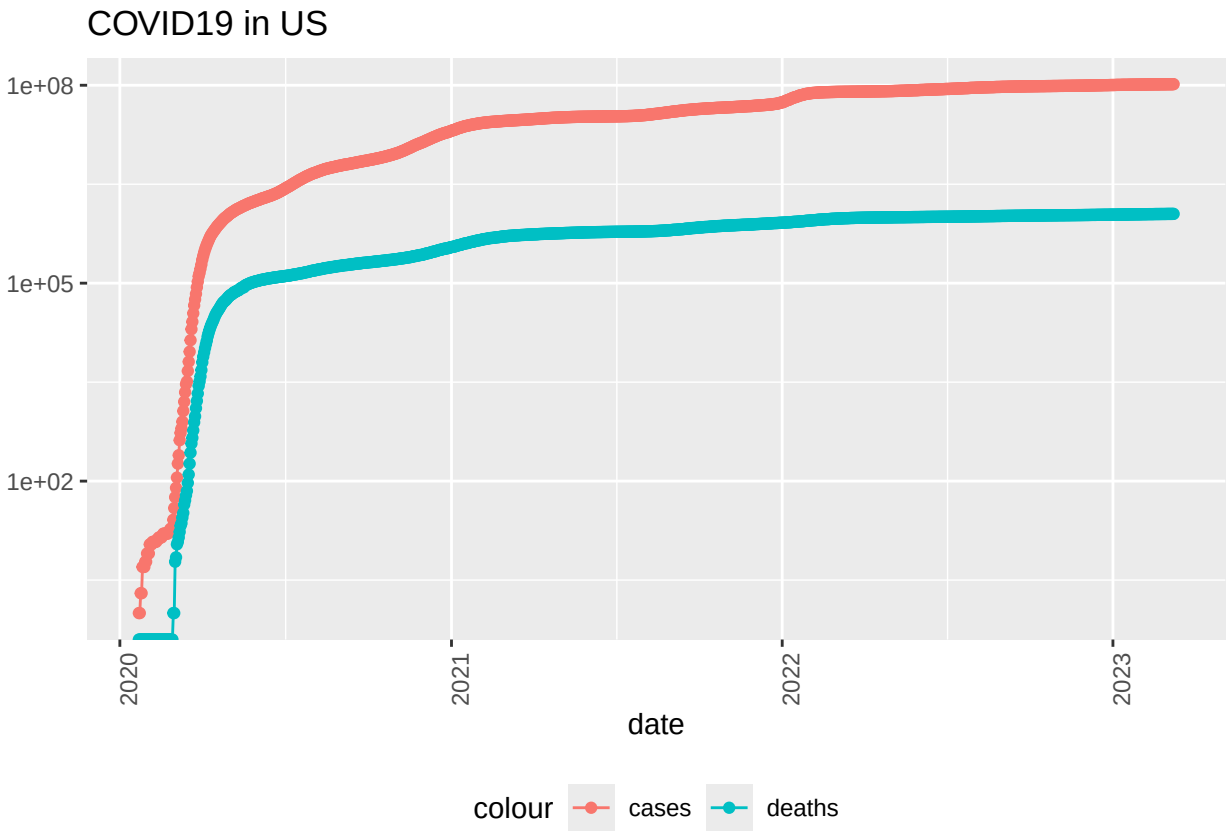
```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can
## override using the `.groups` argument.
```

```
US_totals <- US_by_state %>%
    group_by(Country_Region, date) %>%
    summarize(cases=sum(cases), deaths=sum(deaths),
              Population=sum(Population)) %>%
    mutate(deaths_per_mill = deaths*1000000/Population) %>%
    select(Country_Region, date, cases, deaths, deaths_per_mill, Population) %>%
    ungroup()
```

```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

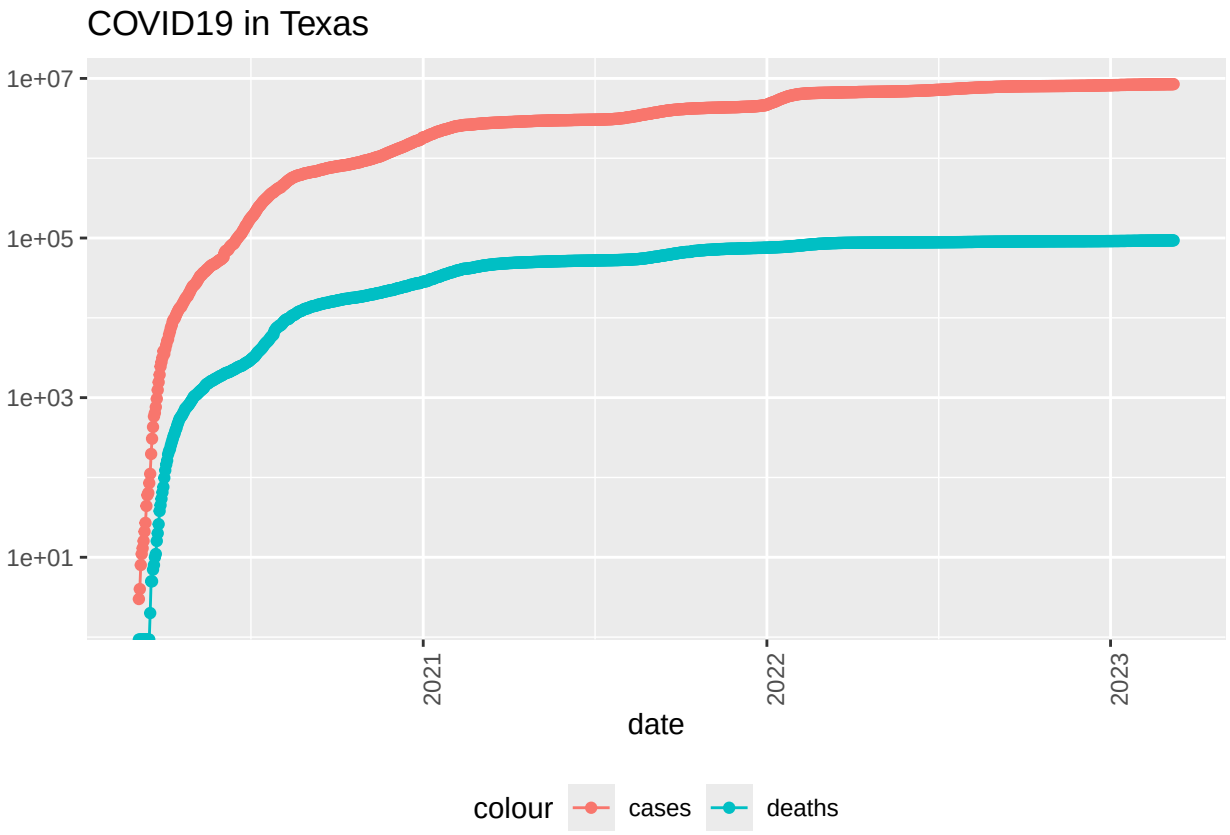
Using the US totals, create a plot for COVID-19 progression

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



Now select a single state to plot

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

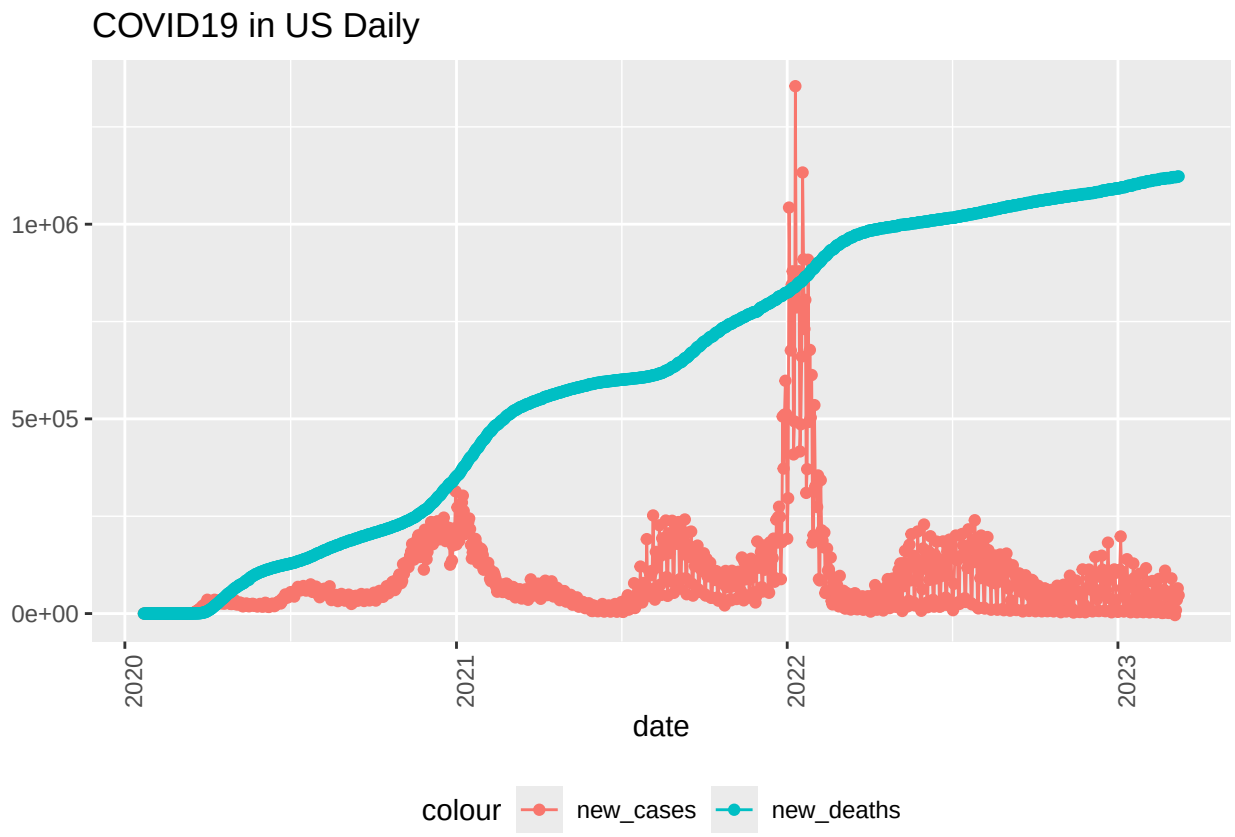



```
## [1] 1122724
```

In this next graph, we look at the data collected and graph out the total deaths and daily cases.

```
## [1] "2023-03-09"
```

```
## [1] 1122724
```



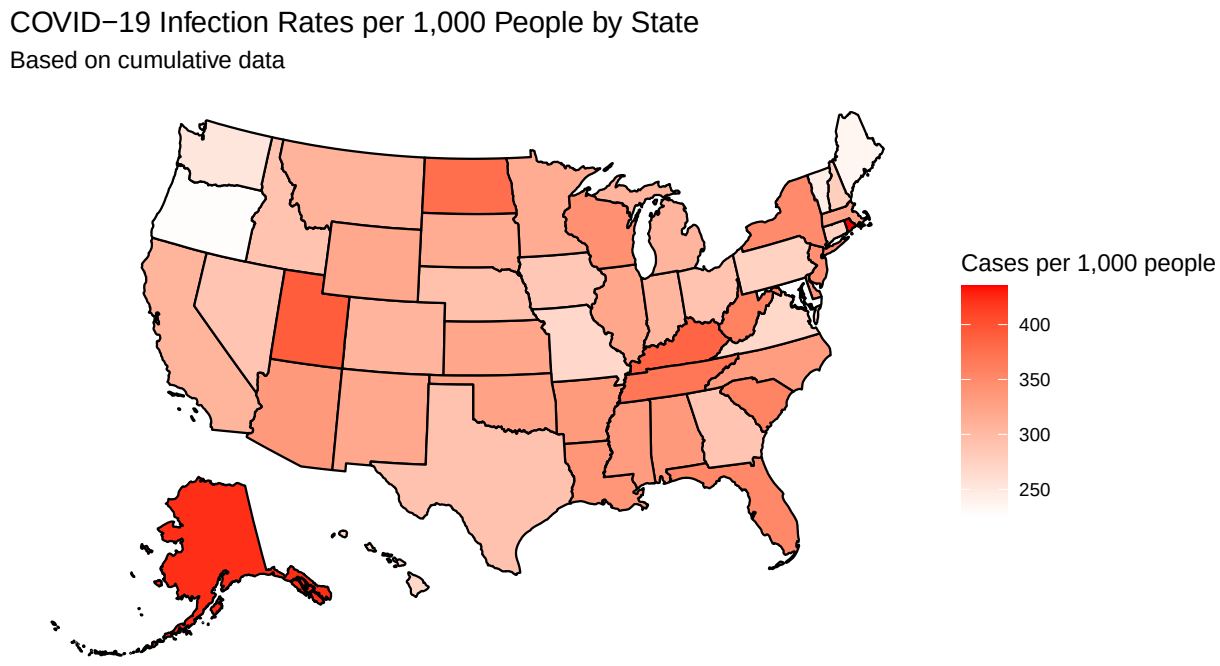
Next, we want to do some state by state visualizations for infections, deaths, and death rate for confirmed infections. Prep the data first.

```
## # A tibble: 10 x 7
##    deaths_per_thou cases_per_thou Province_State        deaths  cases population
##              <dbl>          <dbl> <chr>                 <dbl>  <dbl>      <dbl>
##  1           0.611           150. American Samoa           34 8.32e3      55641
##  2           0.744           248. Northern Mariana Isl~    41 1.37e4      55144
##  3           1.21            231. Virgin Islands          130 2.48e4     107268
##  4           1.30            269. Hawaii                 1841 3.81e5    1415872
##  5           1.49            245. Vermont                 929 1.53e5     623989
##  6           1.55            293. Puerto Rico            5823 1.10e6    3754939
##  7           1.90            391. Utah                   5298 1.09e6    2785478
##  8           2.03            252. District of Columbia   1432 1.78e5     705749
##  9           2.04            422. Alaska                 1486 3.08e5     728809
## 10           2.06            253. Washington            15683 1.93e6    7614893
## # i 1 more variable: deaths_per_case <dbl>
```

```
## # A tibble: 10 x 7
##    deaths_per_thou cases_per_thou Province_State deaths   cases population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            4.55           336. Arizona         33102 2443514    7278717
##  2            4.54           326. Oklahoma        17972 1290929    3956971
##  3            4.49           333. Mississippi     13370  990756    2976149
##  4            4.44           359. West Virginia    7960  642760    1792147
##  5            4.32           320. New Mexico       9061  670929    2096829
```

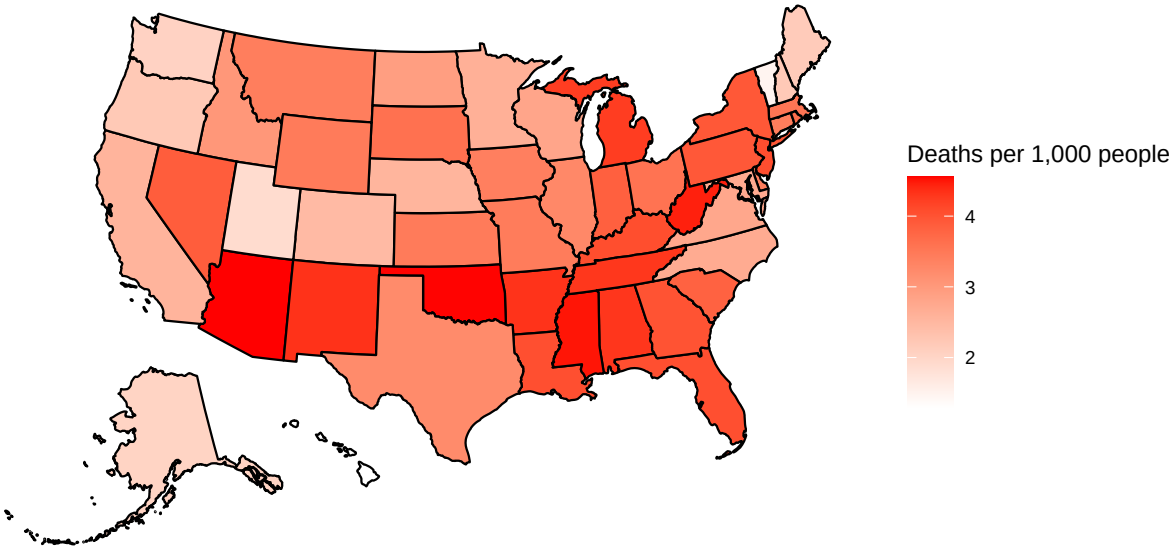
```
##  6            4.31           334. Arkansas        13020 1006883    3017804
##  7            4.29           335. Alabama         21032 1644533    4903185
##  8            4.28           368. Tennessee       29263 2515130    6829174
##  9            4.23           307. Michigan        42205 3064125    9986857
## 10            4.06           385. Kentucky        18130 1718471    4467673
## # i 1 more variable: deaths_per_case <dbl>
```

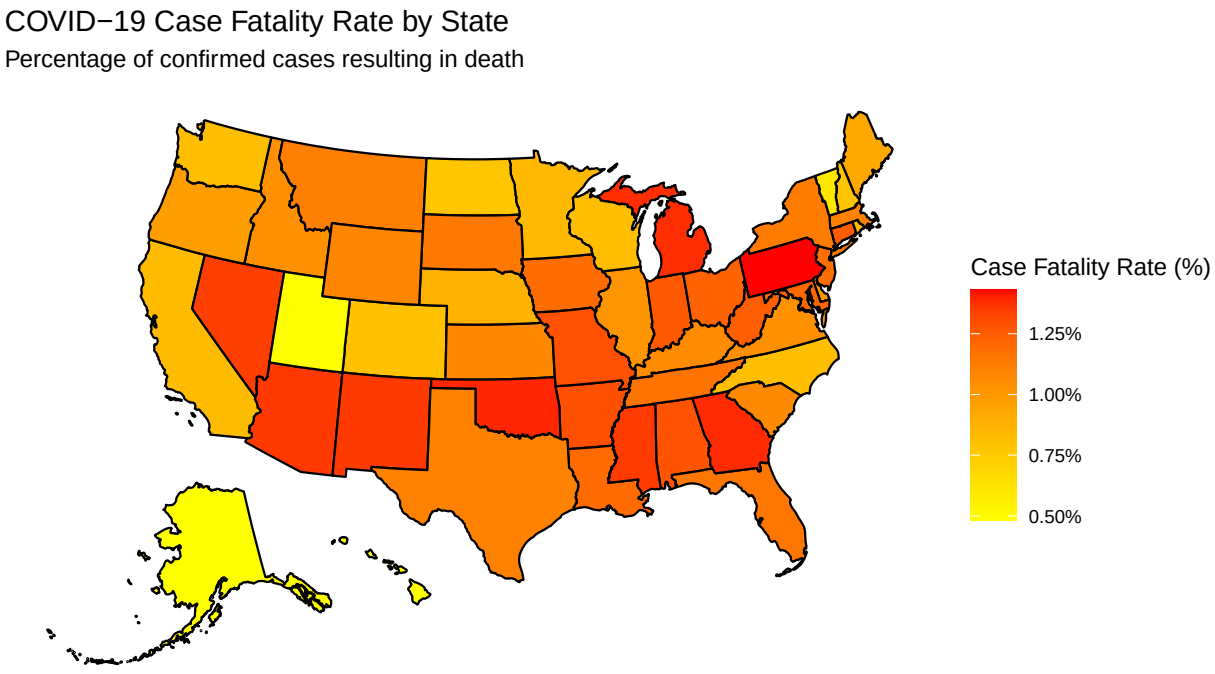
These plots require the `usmap` library. Please install if you do not have it already.



COVID−19 Death Rates per 1,000 People by State

Based on cumulative data

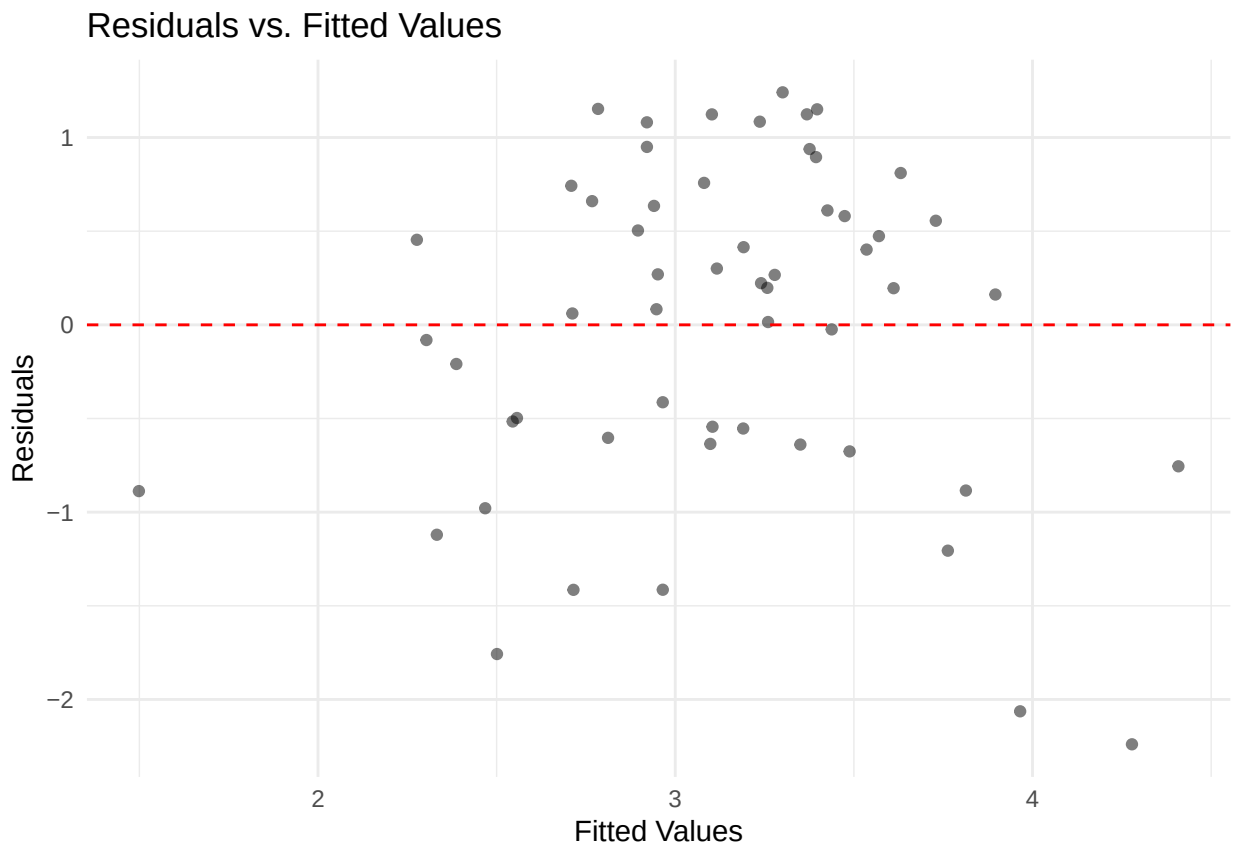


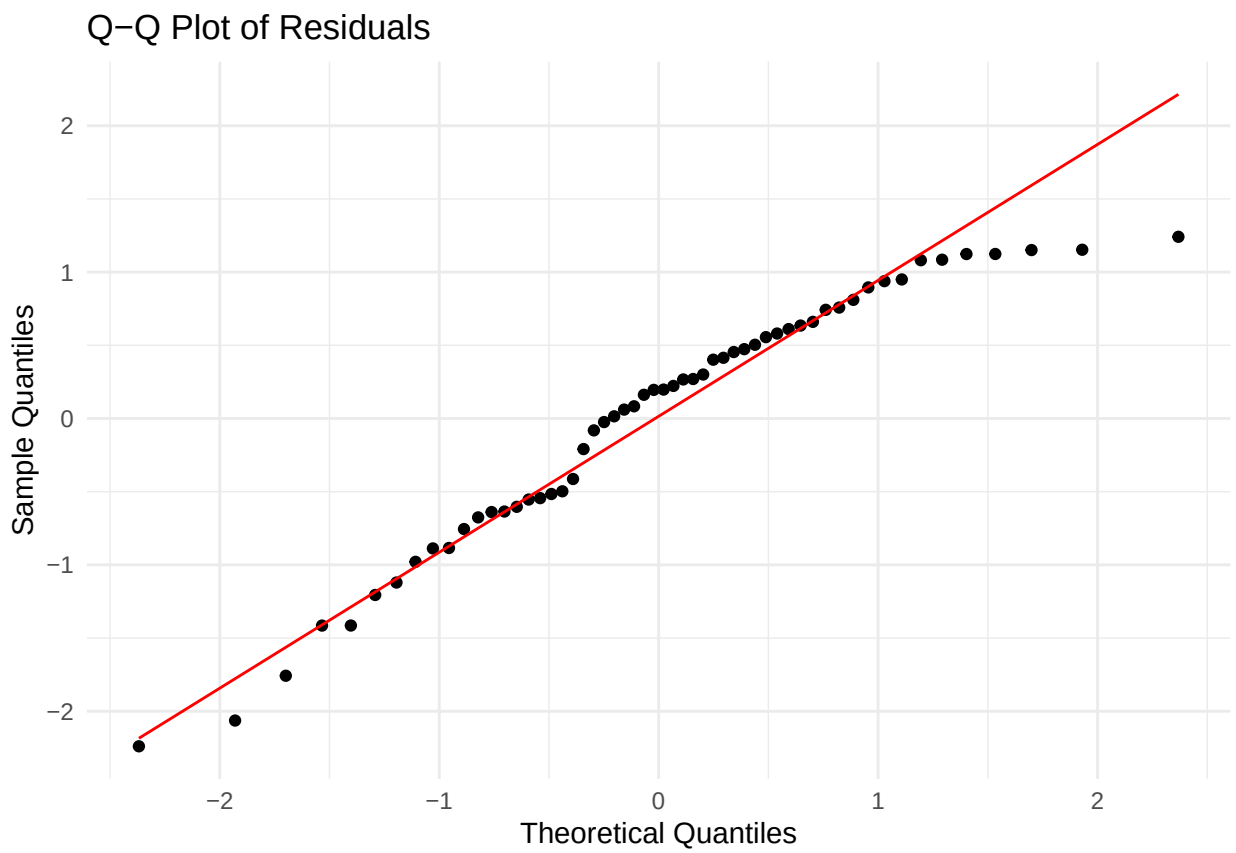


Creating a Model for the data

Here, we continue looking at the death rate per infections by creating a linear regression model. This can help indicate of the effectiveness of a state's healthcare systems for treatment of COVID-19 given the infection rates.

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2394 -0.6114  0.1965  0.6413  1.2413
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.02599    0.72442  -0.036    0.972
## cases_per_thou  0.01020    0.00231   4.414 4.89e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8803 on 54 degrees of freedom
## Multiple R-squared:  0.2652, Adjusted R-squared:  0.2516
## F-statistic: 19.49 on 1 and 54 DF,  p-value: 4.894e-05
```


Residuals vs. Fitted Values
Residuals
Fitted Values
1
0
−1
−2
2
3
4



So here, we create a linear regression model based on the cases and deaths, and the expected relation between those two values. Although we see that our p-value is low and it indicates a meaningful correlation, summaries of derived statistics can be difficult to interpret, so we also create a couple of visualizations for the model. First, we do a residuals plot. This plot shows the model's residuals (differences between observed and predicted values) against the fitted values. This helps check for constant variance and linearity. Ideally, you want to see a random scatter of points with no clear patterns, which is what we generally see here, although we can see some slight clustering near the middle, and no visible outliers on the high y side. The second plot is a Q-Q plot, which compares the distribution of the model's residuals to a theoretical normal distribution. Points following the diagonal line suggest normally distributed residuals, which is an assumption of linear regression. In this case, we see that our values stay near the normal line up to a standard deviation in either direction, but begin to deviate outside of that. This makes intuitive sense, as it would indicate that areas with either very high or very low case loads have results that reflect how much strain their health systems were subjected to.

Sources of Bias

Sources of bias in COVID-19 data have been generally identified among the following areas:

- Testing availability
- Population density (urban/rural)
- Demographic differences
- Healthcare system capacity

My personal bias which I noticed while going thru the problem was that I used my own state as the one which I initially wanted to look at more indepth. I mitigated this by extending the same analysis on other

states and doing per-capita comparisons. More broadly, this could be mitigated by masking state names so that analyses can be developed and run blindly across various states.

```
## R version 4.4.1 (2024-06-14)
## Platform: aarch64-apple-darwin23.4.0
## Running under: macOS Sonoma 14.6.1
##
## Matrix products: default
## BLAS:   /opt/homebrew/Cellar/openblas/0.3.28/lib/libopenblasp-r0.3.28.dylib
## LAPACK: /opt/homebrew/Cellar/r/4.4.1/lib/R/lib/libRlapack.dylib;  LAPACK version 3.12.0
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/POSIX/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Chicago
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] gridExtra_2.3   usmap_0.7.1     lubridate_1.9.3 forcats_1.0.0
##  [5] stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2     readr_2.1.5
##  [9] tidyr_1.3.1     tibble_3.2.1    ggplot2_3.5.1   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4         generics_0.1.3     class_7.3-22       KernSmooth_2.23-24
##  [5] stringi_1.8.4      hms_1.1.3          digest_0.6.36      magrittr_2.0.3
##  [9] evaluate_0.24.0    grid_4.4.1         timechange_0.3.0   fastmap_1.2.0
## [13] e1071_1.7-14       DBI_1.2.3          fansi_1.0.6        scales_1.3.0
## [17] cli_3.6.3          rlang_1.1.4        crayon_1.5.3       units_0.8-5
## [21] bit64_4.0.5        munsell_0.5.1      withr_3.0.1        yaml_2.3.10
## [25] tools_4.4.1        parallel_4.4.1     tzdb_0.4.0         usmapdata_0.3.0
## [29] colorspace_2.1-1   curl_5.2.1         vctrs_0.6.5        R6_2.5.1
## [33] proxy_0.4-27       classInt_0.4-10    lifecycle_1.0.4    bit_4.0.5
## [37] vroom_1.6.5        pkgconfig_2.0.3    pillar_1.9.0       gtable_0.3.5
## [41] Rcpp_1.0.13        glue_1.7.0         sf_1.0-16          xfun_0.46
## [45] tidyselect_1.2.1   highr_0.11         rstudioapi_0.16.0  knitr_1.48
## [49] farver_2.1.2       htmltools_0.5.8.1  labeling_0.4.3     rmarkdown_2.27
## [53] compiler_4.4.1
```