COVID-19_Data_analysis

CC

2023-10-09

Import Library

At first, we have to import some useful library of R, which can help to analyze the data in the following sections.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
```

Statement & Interesting

To better understand the impact of COVID-19, especially in the United States, we need to analyze data. We hope that through the insights conveyed by data, we can gather useful information. For example, identifying areas with a higher case rate or regions with a higher mortality rate could provide valuable insights for relevant departments and organizations. This information might lead to further actions, such as investigating whether there's a lack of healthcare resources in certain areas or if the density of healthcare facilities contributes to a higher mortality rate. As we approach the end of 2023, we still aim to gain insights from COVID-19 data to address future challenges, which is our ultimate goal.

Import Data

According to the course instructions, we need to search for COVID-19 data from various sources. Consistent with the course demonstration, after comparison, I chose to use data from Johns Hopkins University because they provide more comprehensive information about the data source. As the pandemic evolved, the website displays the last data date as March 9, 2023, which aligns with the time frame for data import, visualization, and model analysis in our project.

We intend to focus on analyzing COVID-19 data in the United States, so we selected data related to the USA. The specific data source can be found at the following link: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cov
file_names <- c("time_series_covid19_confirmed_US.csv",
                "time_series_covid19_deaths_US.csv")

urls <- str_c(url_in,file_names)
US_cases <- read_csv(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.
```

```
US_deaths <- read_csv(urls[2])
```

```
## 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.
```

After importing the data we intend to analyze, we pause at this step to examine it using RStudio. We want to check if there are any unnecessary variables and understand the specific format of the data. Does it align with our expectations? Through this inspection process, we gain insights into what our next steps should be and how to handle this data effectively.

```
US_cases
```

```
## # A tibble: 3,342 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
##  3 84001005 US    USA     840  1005 Barbour  Alabama        US              31.9
##  4 84001007 US    USA     840  1007 Bibb     Alabama        US              33.0
##  5 84001009 US    USA     840  1009 Blount   Alabama        US              34.0
##  6 84001011 US    USA     840  1011 Bullock  Alabama        US              32.1
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # i 3,332 more rows
## # 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>, ...
```

```
US_deaths
```

```
## # A tibble: 3,342 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
##  3 84001005 US    USA     840  1005 Barbour  Alabama        US              31.9
##  4 84001007 US    USA     840  1007 Bibb     Alabama        US              33.0
##  5 84001009 US    USA     840  1009 Blount   Alabama        US              34.0
##  6 84001011 US    USA     840  1011 Bullock  Alabama        US              32.1
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # i 3,332 more rows
## # 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>, ...
```

Tidy Data At this stage, after examining the imported data format, we identified some columns that are not necessary for our analysis. Here, we proceed with data processing by removing unnecessary variables. We also organize the case count and death count into a format where we have one record per day and per region. This format will facilitate conducting time-series-related analyses, similar to the demonstration provided by the instructor in class.

```
US_cases <- US_cases%>%
  pivot_longer(cols= -c(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= -c(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 with `by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, date)`
```

```
US
```

```
## # A tibble: 3,819,906 x 8
##    Admin2 Province_State Country_Region Combined_Key date       cases Population
##    <chr>  <chr>          <chr>          <chr>        <date>     <dbl>      <dbl>
##  1 Autau~ Alabama        US             Autauga, Al~ 2020-01-22     0      55869
##  2 Autau~ Alabama        US             Autauga, Al~ 2020-01-23     0      55869
##  3 Autau~ Alabama        US             Autauga, Al~ 2020-01-24     0      55869
##  4 Autau~ Alabama        US             Autauga, Al~ 2020-01-25     0      55869
##  5 Autau~ Alabama        US             Autauga, Al~ 2020-01-26     0      55869
##  6 Autau~ Alabama        US             Autauga, Al~ 2020-01-27     0      55869
##  7 Autau~ Alabama        US             Autauga, Al~ 2020-01-28     0      55869
##  8 Autau~ Alabama        US             Autauga, Al~ 2020-01-29     0      55869
##  9 Autau~ Alabama        US             Autauga, Al~ 2020-01-30     0      55869
## 10 Autau~ Alabama        US             Autauga, Al~ 2020-01-31     0      55869
## # i 3,819,896 more rows
## # i 1 more variable: deaths <dbl>
```

Next, we use the 'summary()' function to review the data we've organized to confirm that the data format meets our expectations and doesn't require any specific adjustments.

```
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:3819906     Length:3819906     Length:3819906     Length:3819906
##  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.   :  -3073   Min.   :       0   Min.   :  -82.0
##  1st Qu.:2020-11-02   1st Qu.:    330   1st Qu.:    9917   1st Qu.:    4.0
##  Median :2021-08-15   Median :   2272   Median :   24892   Median :   37.0
##  Mean   :2021-08-15   Mean   :  14088   Mean   :   99604   Mean   :  186.9
##  3rd Qu.:2022-05-28   3rd Qu.:   8159   3rd Qu.:   64979   3rd Qu.:  122.0
##  Max.   :2023-03-09   Max.   :3710586   Max.   :10039107   Max.   :35545.0
```

Visualizing After organizing the data, we follow the steps outlined in the course for visualization. In this step, we plot time on the X-axis and cumulative death counts and case counts on the Y-axis to depict the evolution of COVID-19 in the US over time. The specific results are as follows:

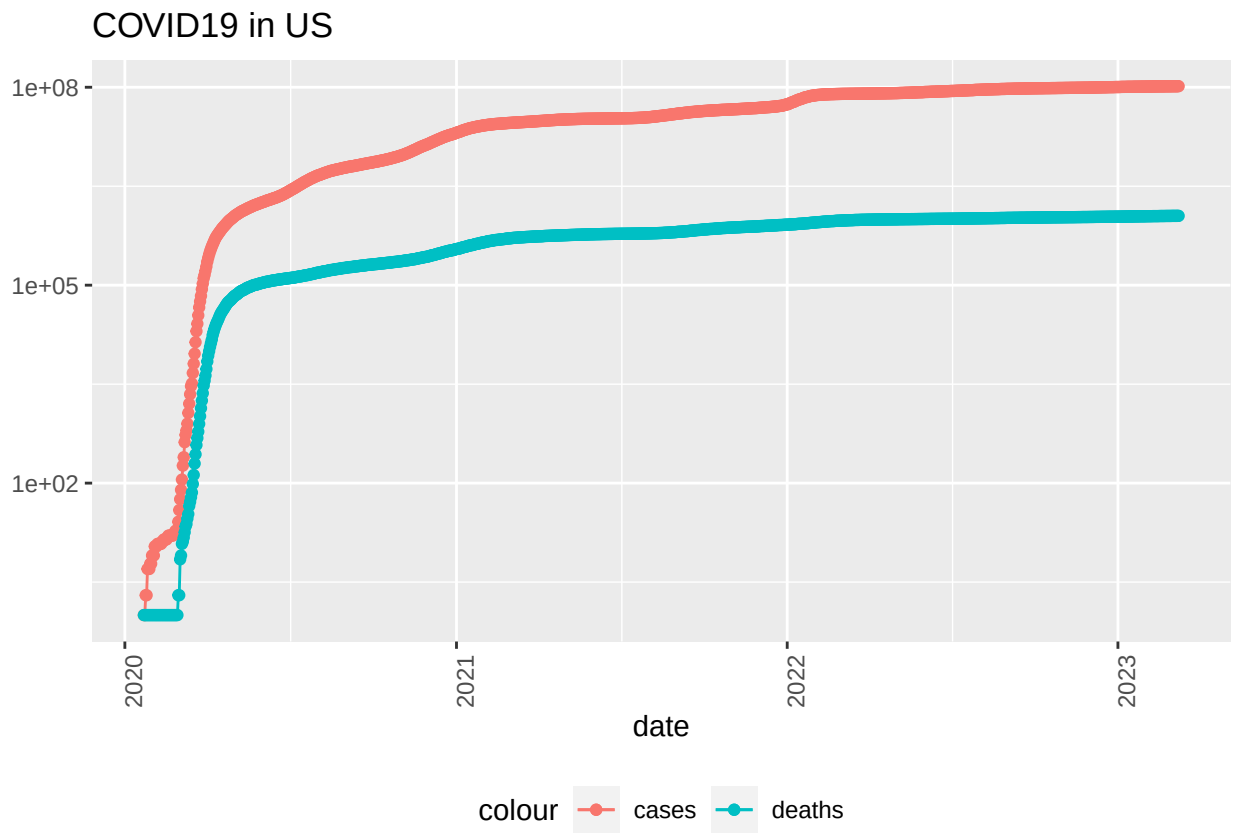
```
#
US_by_state <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000/ Population) %>%
  select(Province_State, Country_Region, date,
         cases,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

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

```
US_total <- 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.
```

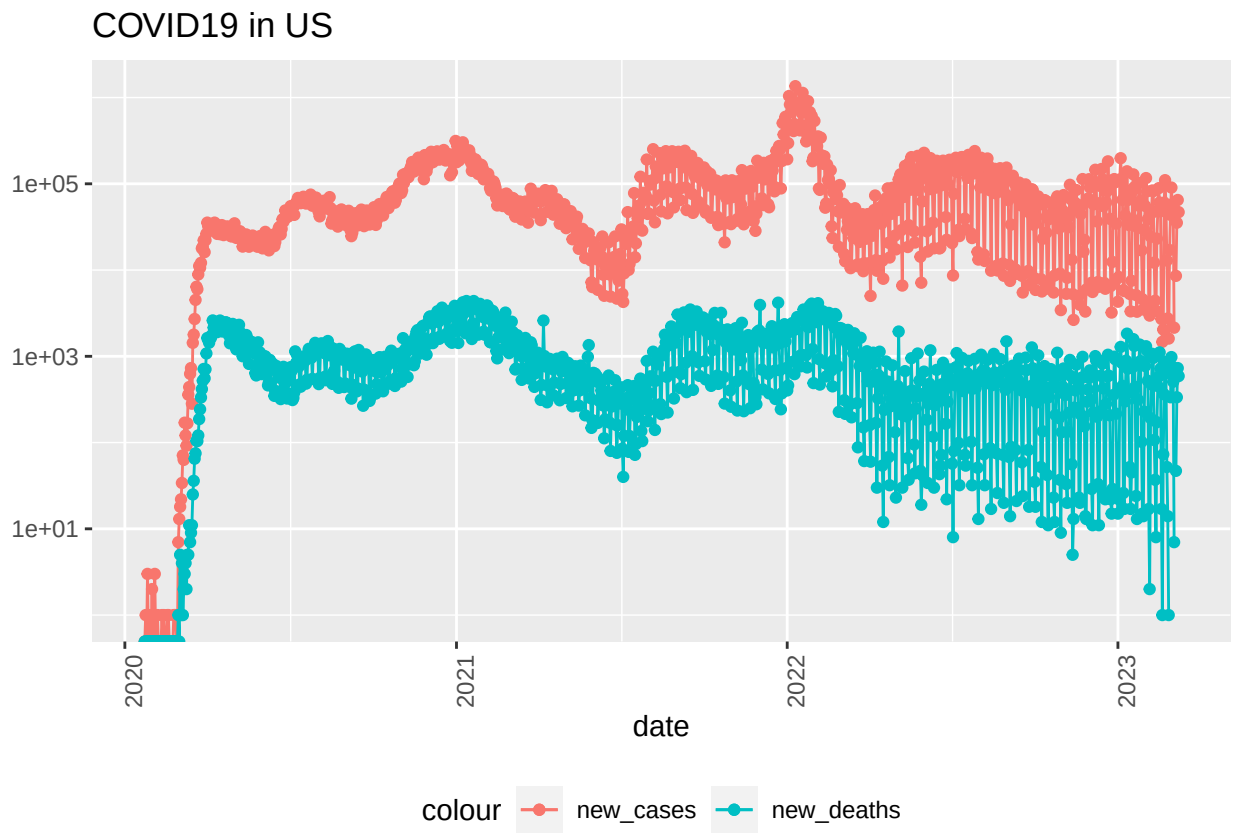
```
US_total %>%
 filter(cases>0) %>%
 ggplot(aes(x=date, y=cases))+
 geom_line(aes(color="cases"))+
 geom_point(aes(color="cases"))+
 geom_line(aes(y=deaths, color = "deaths"))+
 geom_point(aes(y=deaths, color = "deaths"))+
 scale_y_log10()+
 theme(legend.position = "bottom",
       axis.text.x = element_text(angle=90))+
 labs(title = "COVID19 in US", y=NULL)
```



In addition to cumulative numbers, we further incorporated a new variable, which is the daily new counts. This is calculated by subtracting the cumulative counts of one day from the cumulative counts of the previous day. This approach helps to visualize trends.

```
#Add
US_total <- US_total %>%
  mutate(new_cases = cases- lag(cases),
         new_deaths = deaths- lag(deaths),)

US_total %>%
  ggplot(aes(x=date, y=new_cases))+
  geom_line(aes(color="new_cases"))+
  geom_point(aes(color="new_cases"))+
  geom_line(aes(y=new_deaths, color = "new_deaths"))+
  geom_point(aes(y=new_deaths, color = "new_deaths"))+
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle=90))+
  labs(title = "COVID19 in US", y=NULL)
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 2 rows containing missing values (`geom_point()`).

## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



Because COVID-19-related deaths are the least desirable outcome, especially in larger cities where the impact can be more severe, we want to examine cities with a population greater than 5,000,000. We will assess which cities perform the best and worst based on the metric of deaths per thousand people.

Next, we will create a linear model to determine whether it has predictive power at a statistically significant confidence level. This analysis aims to provide insights and information for decision-making.

```
#State
US_state_totals <- US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            Population = max(Population),
            cases_per_thou = 1000* cases / Population,
            deaths_per_thou = 1000* deaths / Population) %>%
  filter(cases >0, Population >5000000)

US_state_totals %>%
  slice_min(deaths_per_thou, n = 10) %>%
select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State deaths    cases Population
##              <dbl>          <dbl> <chr>           <dbl>    <dbl>      <dbl>
##  1            2.06           253. Washington      15683  1928913    7614893
##  2            2.46           306. Colorado        14181  1764401    5758736
##  3            2.56           307. California     101159 12129699   39512223
```

```
##  4           2.64          315. Minnesota       14870  1778866    5639632
##  5           2.71          331. North Carolina  28432  3472644   10488084
##  6           2.74          226. Maryland        16544  1365297    6045680
##  7           2.77          269. Virginia        23666  2291951    8535519
##  8           2.81          345. Wisconsin       16375  2006582    5822434
##  9           3.22          292. Texas           93390  8466220   28995881
## 10           3.27          322. Illinois        41496  4083292   12671821
```

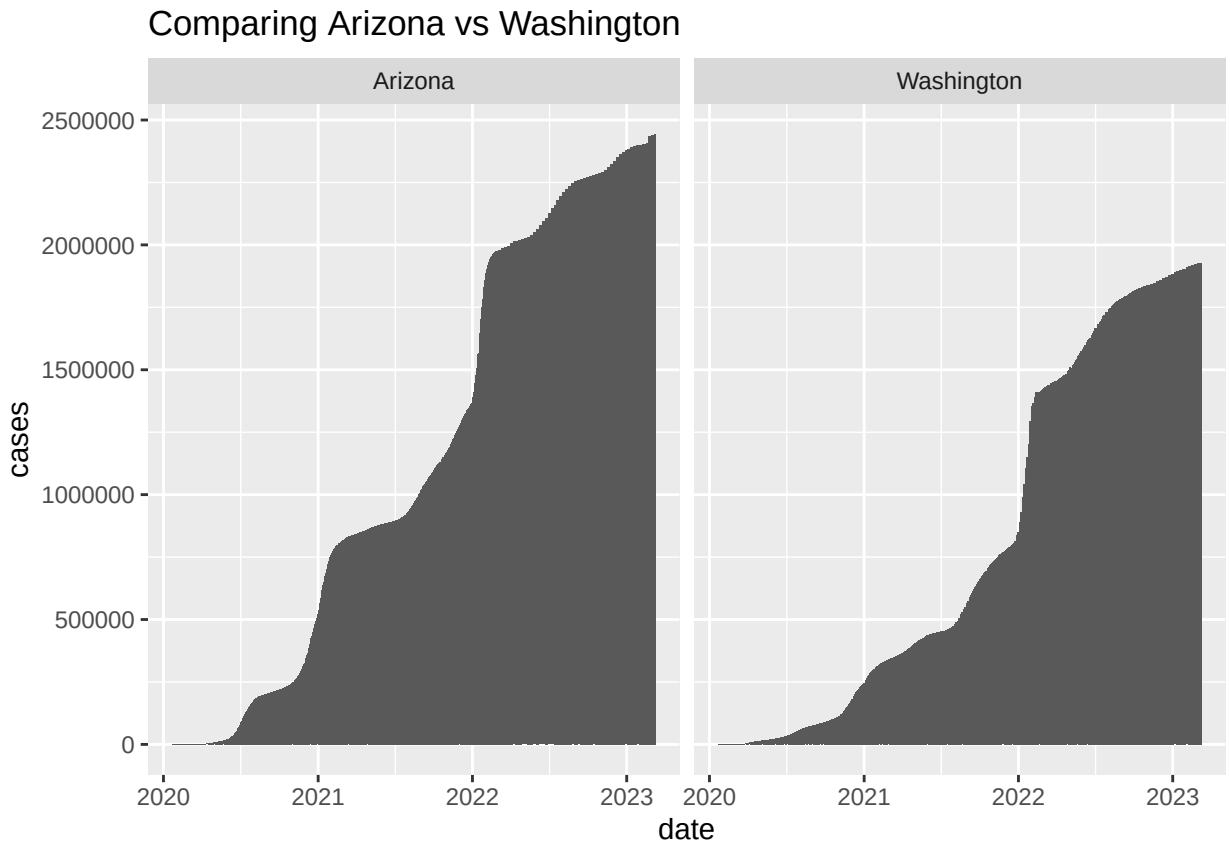
```
US_state_totals %>%
  slice_max(deaths_per_thou, n = 10) %>%
select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    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            4.28           368. Tennessee       29263 2515130    6829174
##  3            4.23           307. Michigan        42205 3064125    9986857
##  4            4.05           343. New Jersey      36015 3048984    8882190
##  5            4.04           353. Florida         86850 7574590   21477737
##  6            4.00           289. Georgia         42489 3068208   10617423
##  7            3.97           349. New York        77157 6794738   19453561
##  8            3.94           276. Pennsylvania    50398 3527854   12801989
##  9            3.88           305. Indiana         26115 2051104    6732219
## 10            3.81           357. South Carolina  19600 1836568    5148714
```

The results indicate that Arizona has the highest deaths per thousand, while Washington has the lowest. In the next step, we will use data from these two cities for our model analysis.

#Compared the Arizona and Washington

```
ComparedAZWA <- US_by_state %>%
                filter(Province_State=="Arizona" | Province_State=="Washington")


ggplot(data=ComparedAZWA,aes(x=date,y=cases)) +
  geom_bar(stat="identity") +
  facet_wrap("Province_State") +
  labs(title="Comparing Arizona vs Washington")
```




```
US_by_state
```

```
## # A tibble: 66,294 x 7
##    Province_State Country_Region date       cases deaths deaths_per_mill
##    <chr>          <chr>          <date>     <dbl>  <dbl>           <dbl>
##  1 Alabama        US             2020-01-22     0      0               0
##  2 Alabama        US             2020-01-23     0      0               0
##  3 Alabama        US             2020-01-24     0      0               0
##  4 Alabama        US             2020-01-25     0      0               0
##  5 Alabama        US             2020-01-26     0      0               0
##  6 Alabama        US             2020-01-27     0      0               0
##  7 Alabama        US             2020-01-28     0      0               0
##  8 Alabama        US             2020-01-29     0      0               0
##  9 Alabama        US             2020-01-30     0      0               0
## 10 Alabama        US             2020-01-31     0      0               0
## # i 66,284 more rows
## # i 1 more variable: Population <dbl>
```

#Modeling

```
ArizonaData <- ComparedAZWA %>% filter(Province_State=="Arizona") %>% mutate(indicator = 1000* cases / P
WashingtonData <- ComparedAZWA %>% filter(Province_State=="Washington")  %>% mutate(indicator = 1000* ca

ComparedAZWA2 <- merge(ArizonaData, WashingtonData, by = "date")
mod <- lm(indicator.y ~ indicator.x, data = ComparedAZWA2)
```

```r
summary(mod)
```

```
##
## Call:
## lm(formula = indicator.y ~ indicator.x, data = ComparedAZWA2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.903 -14.867   6.626  11.662  19.562
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.485513   0.718317  -25.73   <2e-16 ***
## indicator.x   0.780137   0.003614  215.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.42 on 1141 degrees of freedom
## Multiple R-squared:  0.9761, Adjusted R-squared:  0.9761
## F-statistic: 4.66e+04 on 1 and 1141 DF,  p-value: < 2.2e-16
```

According to the linear model, our analysis involving Arizona and Washington resulted in the establishment of a predictive model. At a 0.01 confidence level, the variables demonstrate statistical significance. Furthermore, with an r^2 value of 0.976, indicating very high explanatory power, it suggests a strong linear relationship between the two. Based on the data we currently have, it appears that y-variable can be predicted from x-variable.

Bias Identification and conclusion

In this final project, we conducted various analyses, including presenting distributions using visual charts, filtering out cities with the highest and lowest deaths per thousand people, and then building linear models and data predictions for them. We found statistically significant correlations, which could potentially be used for decision-making. For example, if we noticed an increase in confirmed cases in Arizona, we could anticipate a similar trend in Washington, allowing us to prepare medical resources in advance.

However, our data analysis has biases, primarily because our data is very limited. We had only one variable (x) to use for prediction, which could easily lead to model over-fitting. This resulted in a very high r-squared value. Therefore, we should maintain a conservative approach. The conclusions drawn are based on the data we currently have. Furthermore, we should strive to collect more comprehensive data, validate relationships using data from multiple perspectives, and identify more critical variables. This is what data scientists should do—remain objective, avoid preconceived notions, and gather as much complete data as possible to provide higher-quality decision recommendations to help solve problems. This is my biggest takeaway from this course. Thank you for your time.