Final Project 2: Reproducible Report on COVID19 Data

Sean Coffey

2024-07-19

Instructions for Final Project 2

1. Import, tidy and analyze the COVID19 dataset from the Johns Hopkins Github site. This is the same dataset I used in class. Feel free to repeat and reuse what I did if you want to.
2. Be sure your project is a reproducible .rmd document which your peers can download and knit.
3. It should contain some visualization and analysis that is unique to your project. You may use the data to do any analysis that is of interest to you.
4. You should include at least two visualizations and one model.
5. Be sure to identify any bias possible in the data and in your analysis.

Summary

This report looks at the COVID data sets, primarily from John Hopkins University. It is deliberately broken into sections to show the data science process of import, tidy and then an iterative visualize, analyze, model cycle.

The report shows a number of charts to represent the evolution of COVID over time and applies a couple of methods to create nuanced views of the trends over time. Namely, comparing cumulative cases and deaths during each period then creating 7-day rolling averages to allow a more granular assessment of changes over time.

The experiences of the states in terms of deaths per million people were analysed and visualised geographically.

The question requests modelling. Given the large variation in outcome performance by state, the analysis tries to answer the question: "Is there a significant linkage between the COVID outcome and the state's political leaning?". A data set for political leaning by US state was imported and used to model the linkage between "political leaning" and "deaths per million" during the COVID period.

Whilst the data does support that there is a correlation and Democratic states appear to perform better, the analysis stops short of claiming causation, for which more evidence would be needed.

Throughout the report, reference is made to potential sources of bias, which include the sourcing of data and potential influence on the presentation of results.

I continue to view these exercises as much about learning the tools of the trade as they are about presenting meaningful analysis and conclusions. As such there is rather more about method than is perhaps necessary and I have kept in charts which I might otherwise have deleted.

Import JHU CSSE COVID-19 Dataset and other data

This project uses the COVID time-series data set from John Hopkins University. Full information can be found here: https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/README.md

In addition global population data is taken from CSSEGI (as per the lectures).

There is a local csv data file which contains the political leaning by state drawn from the website worldpopulationreview.com.

Without this local csv file: the markdown won't knit.

Bias The project question asks that we look at bias. The John Hopkins data set is well respected, has been used by many and their site goes to length to explain the sources and errors in the data. As such, I consider it trustworthy. The most obvious source of bias here is the reporting itself.

1. In the US data, states may have differing policies for how they record cases and how diligently this was performed. (Obviously, more testing is likely to mean more cases)
2. These variations are much harder to interpret in the global data which will certainly have many differing approaches to what is considered a case and what is considered a death caused by COVID.

The numbers of deaths, seems less prone to difficulty than the number of cases. Nonetheless, if this work was being used for other than academic practice, I would want to understand more about the sources and potential problems.

```
# creates four variables containing the csv files from John Hopkins github COVID content
base_url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_co
filenames = c("time_series_covid19_confirmed_US.csv",  "time_series_covid19_confirmed_global.csv", "tim
urls = paste0(base_url, filenames)
us_cases = read_csv(urls[1])
global_cases = read_csv(urls[2])
us_deaths = read_csv(urls[3])
global_deaths = read_csv(urls[4])

# Also load global population data, source as per lecture.
uid_lookup_url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/U
uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat,Long_, Combined_Key, code3, iso2, iso3, Admin2))

# Load political lean data from https://worldpopulationreview.com/state-rankings/most-republican-states
# This is 2024 data, so doesn't fully align with timeline, but for the purposes of the exercise...
political_leaning = read_csv("data/political_lean_by_state.csv")
```

Basic tidying and cleaning of the data sets

Two base data frames ("us" and "global") are created from the imported data, according to the following steps:

1. Use 'pivot_longer' to create separate observations for each date
2. Select a subset of columns for analysis
3. Rename some columns for consistency across data sets
4. Mutate, to create date data from chr data type input (lubridate package)
5. Join deaths and cases into one file for both global data and us data.
6. Add population data for global (from different source as per lecture)

```r
global_cases <- global_cases %>%
  pivot_longer(cols = -c('Province/State','Country/Region','Lat','Long'),
               names_to = "date",
               values_to = "cases") %>%
  select(-c(Lat,Long))

global_deaths <- global_deaths %>%
  pivot_longer(cols = -c('Province/State','Country/Region','Lat','Long'),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat,Long))

# create one file with cumulative deaths and cases
global <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = 'Country/Region',
         Province_State = 'Province/State') %>%
  mutate(date = mdy(date)) %>%
  filter(cases > 0) %>%
  unite(Combined_Key, c(Province_State, Country_Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)
```

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

```r
# add population data to global
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID,FIPS)) %>%
  select(Province_State, Country_Region, date, cases, deaths, Population, Combined_Key)

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
```

```r
us_cases <- us_cases %>%
  pivot_longer(cols = matches(".+/.+/.+"),
               names_to = "date",
               values_to = "cases") %>%
  select(-c(UID, iso2, iso3, code3, Lat, Long_)) %>%
  mutate(date = mdy(date))

us_deaths <- us_deaths %>%
  pivot_longer(cols = matches(".+/.+/.+"),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(UID, iso2, iso3, code3, Lat, Long_)) %>%
  mutate(date = mdy(date))

us <- us_cases %>%
  full_join(us_deaths)
```

```
## Joining with 'by = join_by(FIPS, Admin2, Province_State, Country_Region,
## Combined_Key, date)'
```

```r
summary(us)
```

```
##       FIPS          Admin2          Province_State     Country_Region    
##  Min.   :   60   Length:3819906     Length:3819906     Length:3819906    
##  1st Qu.:19076   Class :character   Class :character   Class :character  
##  Median :31012   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :33043                                                           
##  3rd Qu.:47130                                                           
##  Max.   :99999                                                           
##  NA's   :11430                                                           
##  Combined_Key            date                cases           Population      
##  Length:3819906     Min.   :2020-01-22   Min.   :  -3073   Min.   :       0  
##  Class :character   1st Qu.:2020-11-02   1st Qu.:    330   1st Qu.:    9917  
##  Mode  :character   Median :2021-08-15   Median :   2272   Median :   24892  
##                     Mean   :2021-08-15   Mean   :  14088   Mean   :   99604  
##                     3rd Qu.:2022-05-28   3rd Qu.:   8159   3rd Qu.:   64979  
##                     Max.   :2023-03-09   Max.   :3710586   Max.   :10039107  
##                                                                              
##      deaths       
##  Min.   :  -82.0  
##  1st Qu.:    4.0  
##  Median :   37.0  
##  Mean   :  186.9  
##  3rd Qu.:  122.0  
##  Max.   :35545.0  
## 
```

Simple visualisation and analysis

The subsequent section is pretty close to what was shown in the lectures, which I have kept as revision of key R functions and to view a couple of basic analyses.

The results introduced NaNs and infinite values, I chose not to deal with these, because (a) I doubt it had a major impact on the overall picture, (b) I wanted to improve on my own analyses rather than develop what was provided in the lectures.

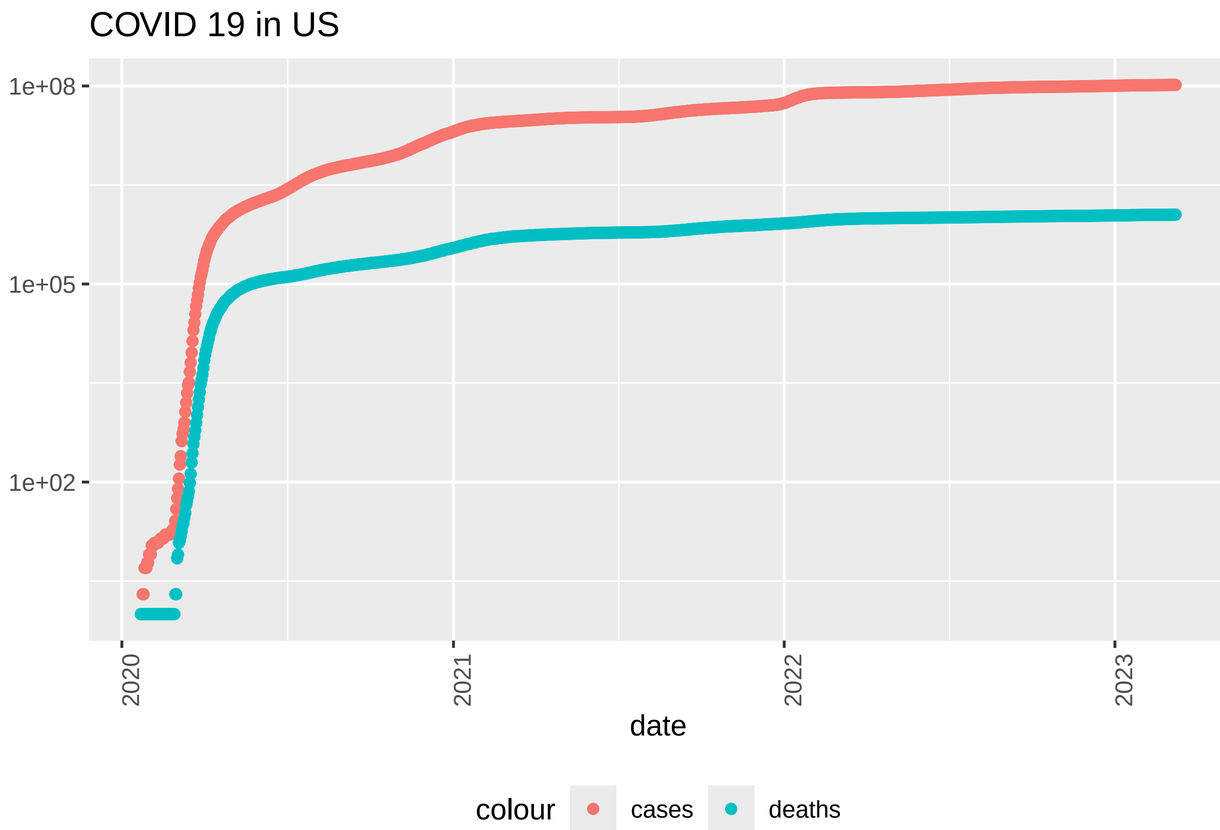
```
us_by_state <- us %>%
  group_by(Province_State, Country_Region, date) %>%
  summarise(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population,
         new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths)) %>%
  select(Province_State, Country_Region, date, cases, deaths, new_cases, new_deaths, deaths_per_mill, Po
  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) %>%
  summarise(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population,
         new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths)) %>%
  select(Country_Region, date, cases, deaths, new_cases, new_deaths, deaths_per_mill, Population) %>%
  ungroup()
```

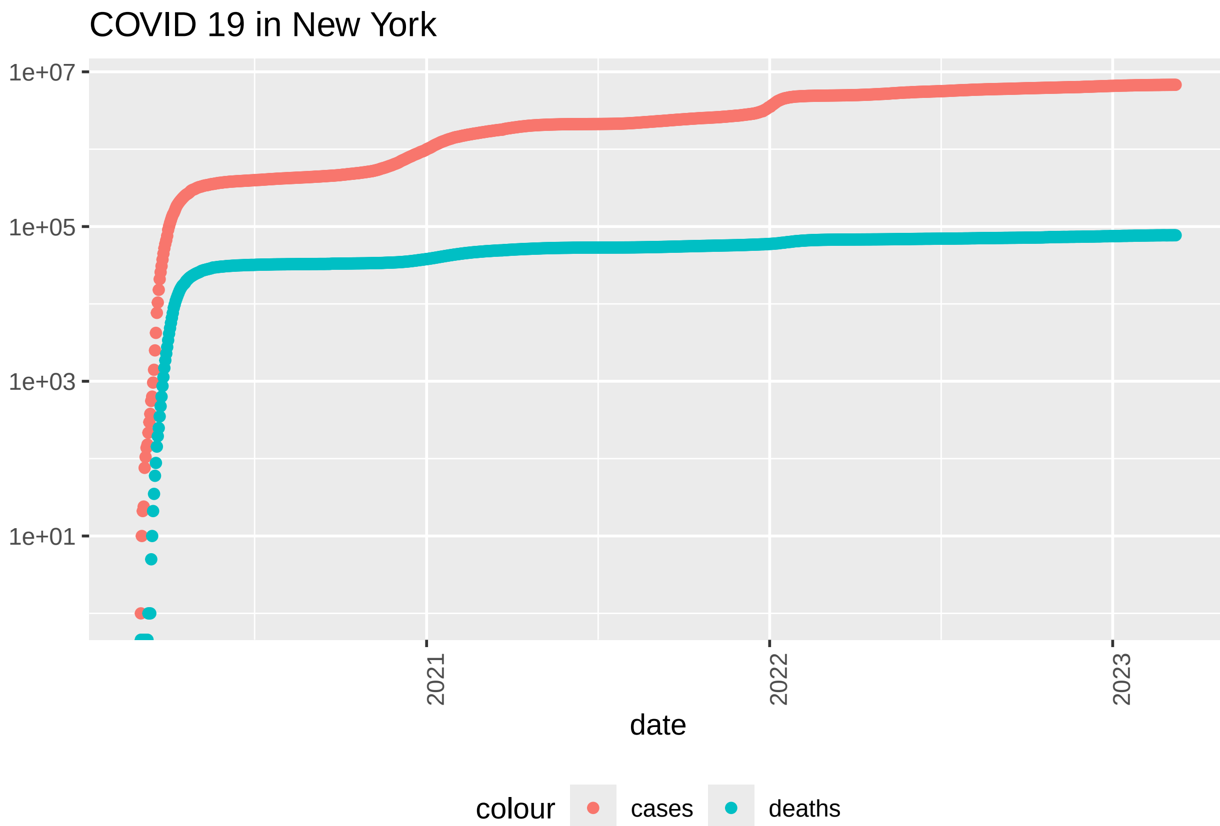
```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

```
us_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date )) +
  geom_point(aes(y = cases, colour = "cases")) +
  geom_point(aes(y = deaths, colour = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID 19 in US", y = NULL)
```




```r
state <- "New York"
us_by_state %>%
  filter(Province_State == state )%>%
  filter(cases > 0) %>%
  ggplot(aes(x = date )) +
  geom_point(aes(y = cases, colour = "cases")) +
  geom_point(aes(y = deaths, colour = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID 19 in ",state), y = NULL)
```

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




```r
us_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date )) +
  geom_point(aes(y = new_cases, colour = "new_cases")) +
  geom_point(aes(y = new_deaths, colour = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID 19 in US", y = NULL)
```

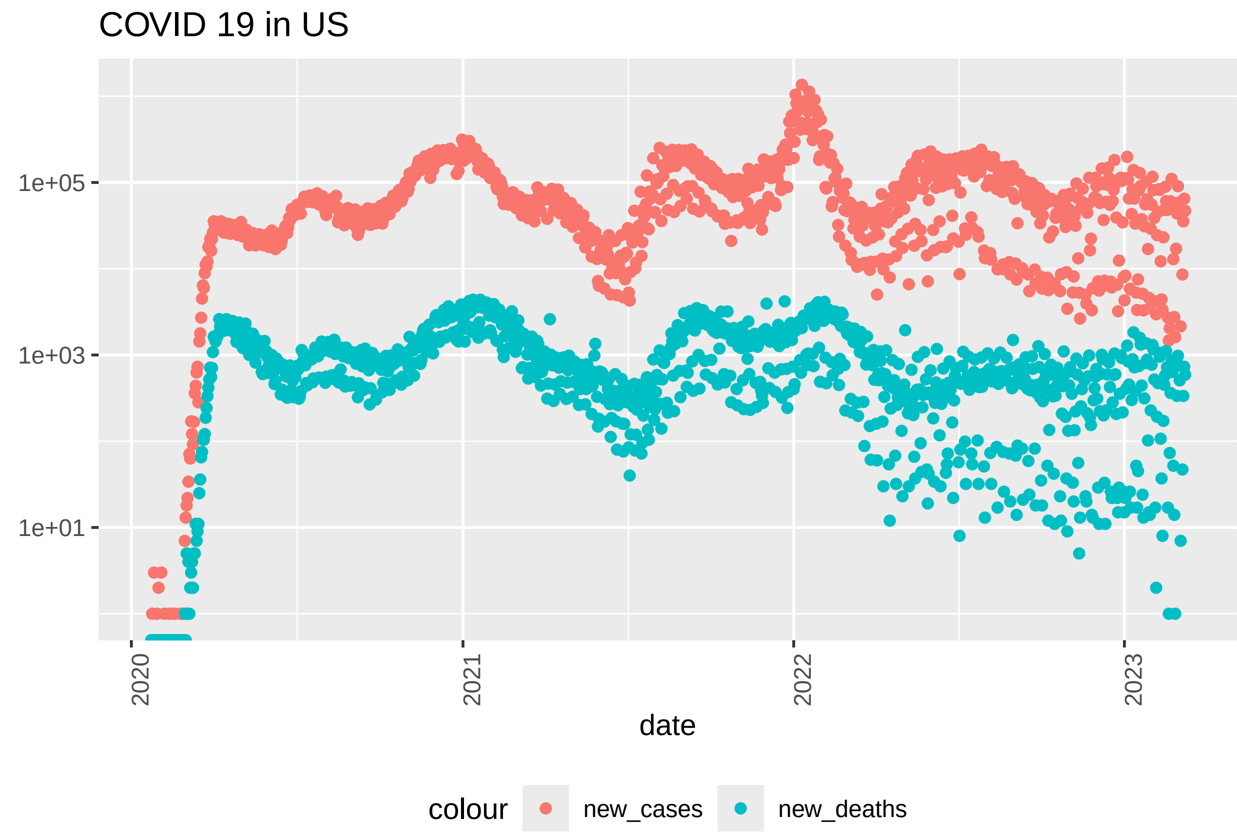
```
## Warning in transformation$transform(x): NaNs produced
## Warning in transformation$transform(x): log-10 transformation introduced
## infinite values.
```

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

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

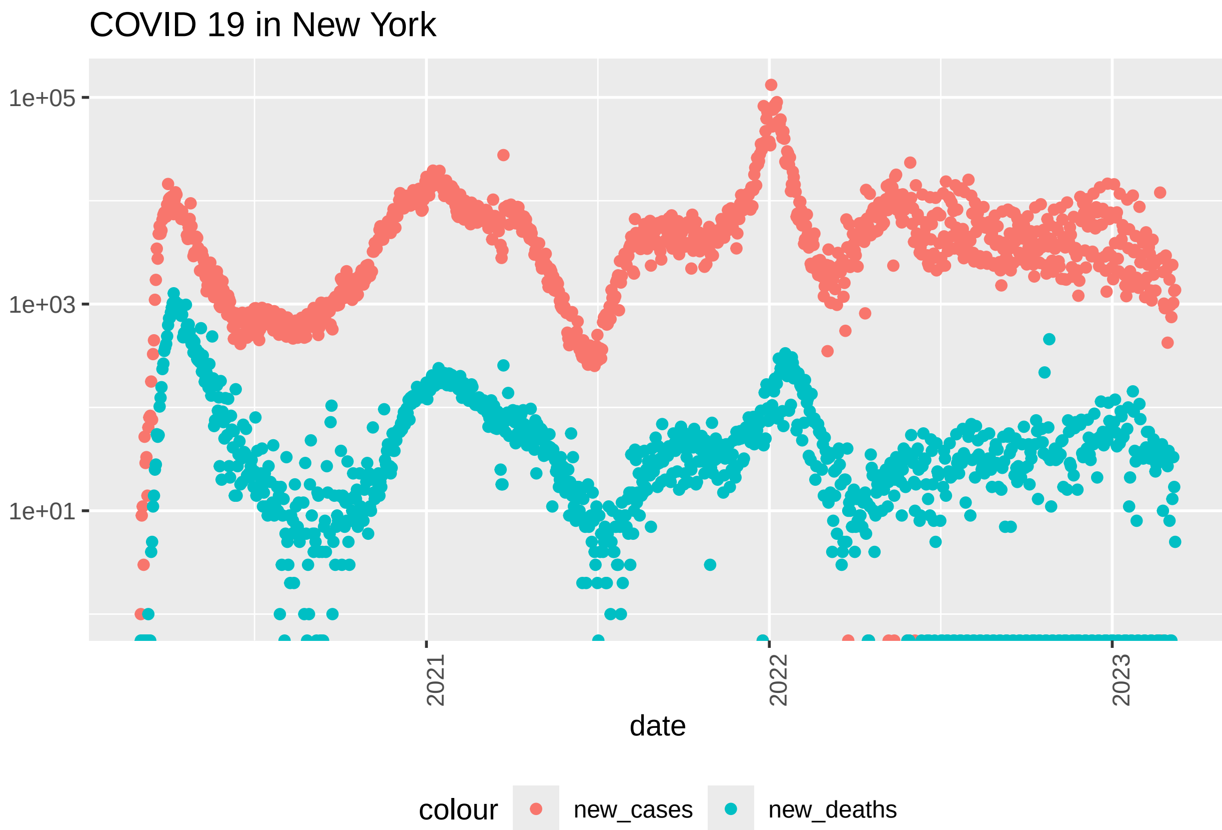
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```r
state <- "New York"
us_by_state %>%
  filter(Province_State == state )%>%
  filter(cases > 0) %>%
  ggplot(aes(x = date )) +
  geom_point(aes(y = new_cases, colour = "new_cases")) +
  geom_point(aes(y = new_deaths, colour = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID 19 in ",state), y = NULL)
```

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

## Warning in transformation$transform(x): NaNs produced

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

## Warning: Removed 8 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Improved Visualisation and Analysis

Drawing on work I did in the earlier Colorado course 'Expressway to Data Science: R Programming and Tidyverse", I used this exercise as an opportunity to remind myself of how to generate these types of visualisation.

The obvious challenge with the two charts that follow is that cases and deaths are shown on the same chart, despite having very different absolute values. I tried to resolve this by using two axes and clearly different colour schemes. In the end, I decided that the benefit of summarising onto one chart allowing a timewise comparison of the trends outweighed the potential confusion of differing axis scales.

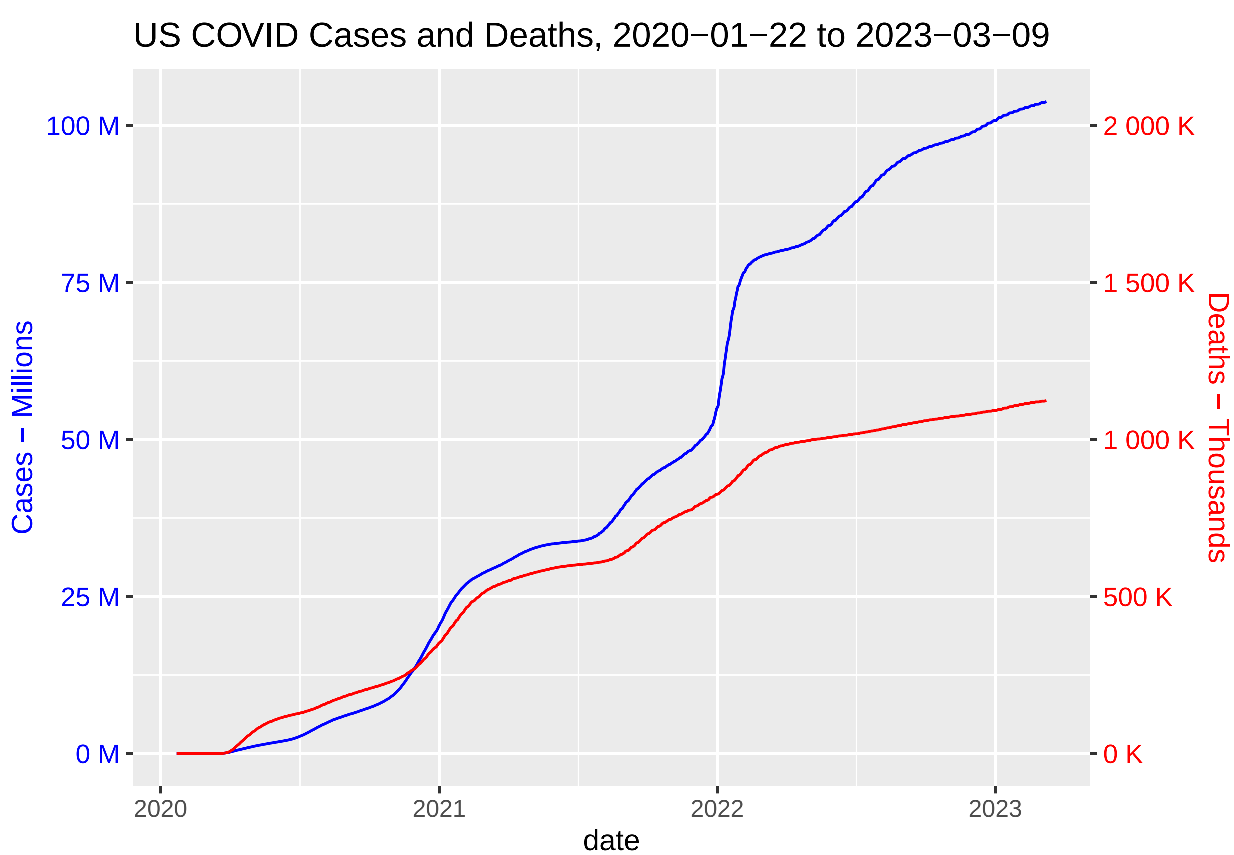
The chart shows the positive change in 2022 as case numbers continued to rise sharply, but this was not matched by an equivalent rise in deaths. Further analysis, for which I don't have time, would be needed to establish causation.

However, as becomes clear in the next section, the cumulative data is hiding something. Whilst the cumulative chart appears to show that the ratio of deaths to cases halved at the start of 2022 (a significant positive improvement), actually this change is largely explained by a major peak in cases during January 2022, so the improvement is overstated.

How did the number of cases and deaths in the US evolve over time?

```
# scalar used to plot secondary axis
scalar = 50

# local variables for date range
```

```
max_date = max(us_totals$date)
min_date = min(us_totals$date)

ggplot(data = us_totals, mapping = aes(x = date)) +
  geom_line(mapping = aes(y = cases), colour = "blue") +
  geom_line(mapping = aes(y = deaths * scalar), colour = "red") +
  # format with two axes for ggplot, the lines are in reality plotted
  # for the same axis, requiring a scaling factor to be used.
  scale_y_continuous(
    labels = label_number(suffix = " M", scale = 1e-6),
    name = "Cases - Millions",
    sec.axis = sec_axis(
      ~./scalar,
      name = "Deaths - Thousands",
      labels = label_number(suffix = " K", scale = 1e-3)
      )
  ) +
  # Amend the colours of the axes for readability
  theme(
    axis.title.y = element_text(colour = "blue", size=11),
    axis.title.y.right = element_text(colour = "red", size=11),
    axis.text.y = element_text(color = "blue", size = 10),
    axis.text.y.right = element_text(color = "red", size = 10)
  ) +

  ggtitle(str_c("US COVID Cases and Deaths, ", min_date, " to ",max_date,sep = ""))
```

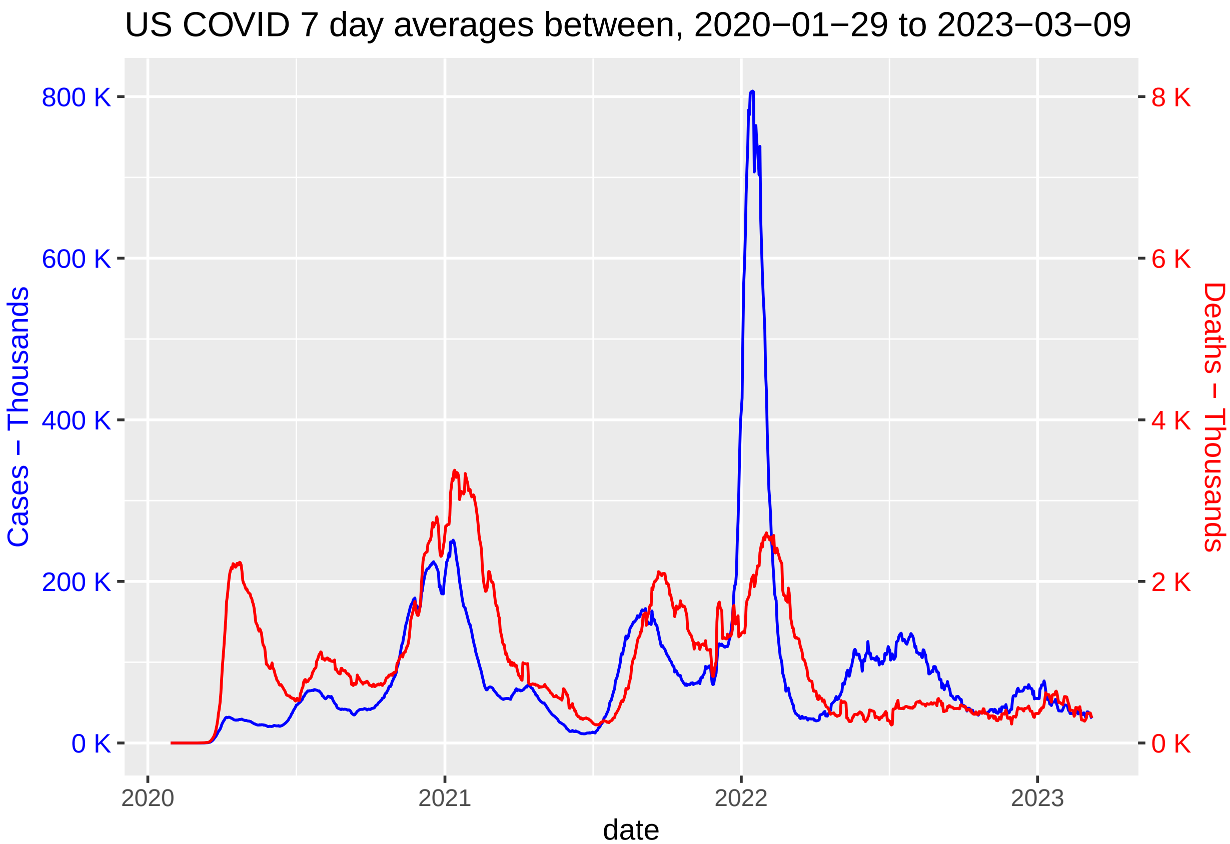


Seven day rolling averages

As the cumulative data could be hiding detail and the daily data (above) shows so much variance day-to-day that it becomes hard to see short term trends, I decided to include an assessment of the 7 day rolling averages for cases and deaths. I remember this was one of the key stats that the majority of news channels showed at the time. It also seems a useful R technique to keep to hand for timeseries analysis.

Rolling seven day averages were calculated using **rollmean()**, selected from the package **"zoos"**.

```
# Add seven day rolling averages to data
us_weeks <- us_totals %>%
  mutate(
    deaths_7 = rollmean(new_deaths, k = 7, align = "right", fill = NA),
    cases_7 = rollmean(new_cases, k = 7, align = "right", fill = NA)
  )

# Extract some other key stats
max_cases_7 <- max(us_weeks$new_cases, na.rm = TRUE)
max_deaths_7 <- max(us_weeks$new_deaths, na.rm = TRUE)
max_cases_7_date <- max(us_weeks$date[us_weeks$new_cases == max_cases_7][2])
max_deaths_7_date <- max(us_weeks$date[us_weeks$new_deaths == max_deaths_7][2])


# scalar used to plot secondary axis
scalar = 100

us_weeks %>%
  # shift start date to allow for rolling average
  filter(between(date, min_date+7, max_date)) %>%

  ggplot(mapping = aes(x = date)) +
    geom_line(mapping = aes(y = cases_7), colour = "blue") +
    geom_line(mapping = aes(y = deaths_7 * scalar), colour = "red") +
    # format with two axes for ggplot, the lines are in reality plotted
    # for the same axis, requiring a scaling factor to be used.
    scale_y_continuous(
      name = "Cases - Thousands",
      labels = label_number(suffix = " K", scale = 1e-3),
      sec.axis = sec_axis(
        ~./scalar,
        name = "Deaths - Thousands",
        labels = label_number(suffix = " K", scale = 1e-3),
        )
    ) +
    # Amend the colours of the axes for readability
  theme(
    axis.title.y = element_text(colour = "blue", size=11),
    axis.title.y.right = element_text(colour = "red", size=11),
    axis.text.y = element_text(color = "blue", size = 10),
    axis.text.y.right = element_text(color = "red", size = 10)
  ) +

  ggtitle(str_c("US COVID 7 day averages between, ", min_date +7, " to ",max_date,sep = ""))
```



From the data, a maximum of 1,354,508 new cases in one week was recorded on January 10, 2022 and a maximum of 4,375 new deaths were recorded in one week on January 12, 2021.

I have not been able to determine the reason for the large peak in cases during January 2022. This would be interesting, if for no other reason than to rule out a data quality issue.

It is also interesting to note that the ratio of cases to deaths seems to move in the wrong direction at the end of the period. Is this the result of new strains of the virus? Or, some other reason. More work would be needed.

How did the performance compare by US state during COVID?

As we also have the state data, it was interesting to look at how the various states compared in terms of their performance in handling COVID.

There are multiple metrics that could have been used here (absolute cases and deaths, ratio of cases to deaths,…) but I selected **deaths per million people** which seemed to have least bias:

1. Not impacted by the number of cases reported (which could differ according to policy rather than reality)
2. Not impacted by the size of the state
3. Objectively, the most important metric to improve. (i.e. reduce deaths)

```
state_performance <- us_by_state %>%
  filter(Province_State != "Grand Princess" & Province_State != "Diamond Princess") %>% #removes this o
  group_by(Province_State) %>%
  summarise(cases = max(cases), deaths = max(deaths), Population = max(Population)) %>%
  mutate(deaths_per_million = deaths * 1000000 / Population) %>%
```

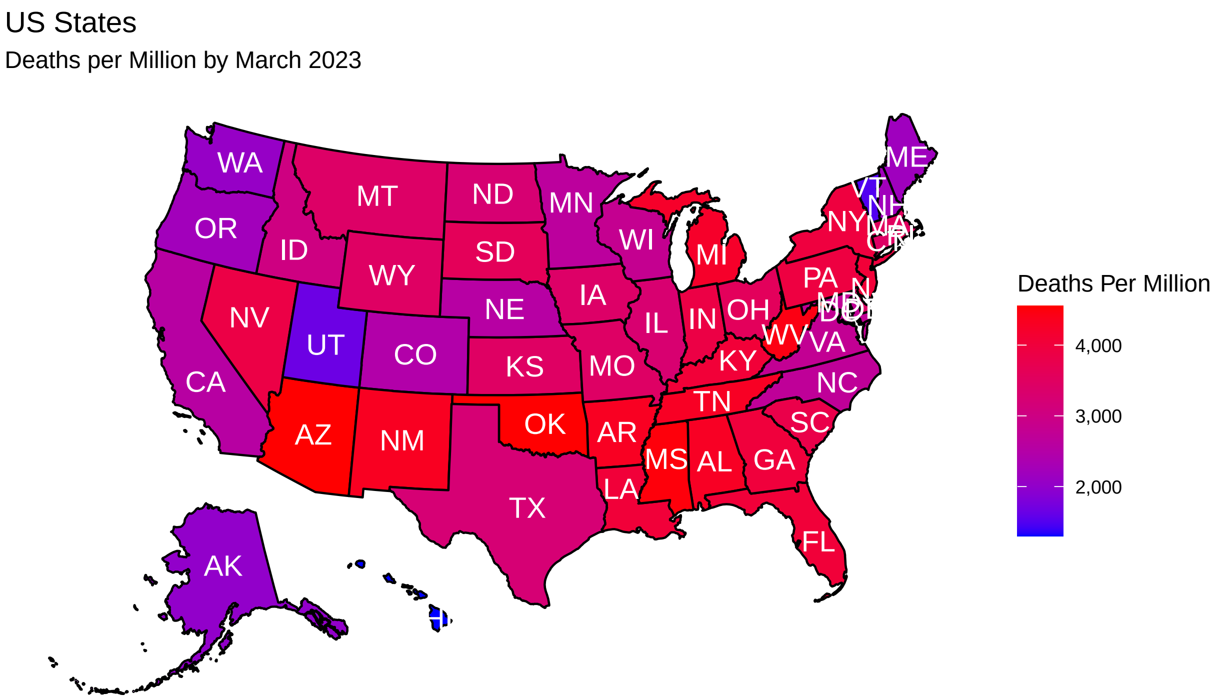
```r
  arrange(desc(deaths_per_million)) %>%
  inner_join(statepop, by = join_by(Province_State == full)) %>%
  select(fips, Province_State, cases, deaths, Population, deaths_per_million)

head(state_performance, n=10 )
```

```
## # A tibble: 10 x 6
##    fips  Province_State   cases deaths Population deaths_per_million
##    <chr> <chr>            <dbl>  <dbl>      <dbl>              <dbl>
##  1 04    Arizona        2443514  33102    7278717              4548.
##  2 40    Oklahoma       1290929  17972    3956971              4542.
##  3 28    Mississippi     990756  13370    2976149              4492.
##  4 54    West Virginia   642760   7960    1792147              4442.
##  5 35    New Mexico      670929   9061    2096829              4321.
##  6 05    Arkansas       1006883  13020    3017804              4314.
##  7 01    Alabama        1644533  21032    4903185              4289.
##  8 47    Tennessee      2515130  29263    6829174              4285.
##  9 26    Michigan       3064125  42205    9986857              4226.
## 10 21    Kentucky       1718471  18130    4467673              4058.
```

```r
tail(state_performance, n = 10)
```

```
## # A tibble: 10 x 6
##    fips  Province_State         cases deaths Population deaths_per_million
##    <chr> <chr>                  <dbl>  <dbl>      <dbl>              <dbl>
##  1 41    Oregon                963564   9373    4217737              2222.
##  2 33    New Hampshire         378428   3003    1359711              2209.
##  3 23    Maine                 318130   2928    1344212              2178.
##  4 53    Washington           1928913  15683    7614893              2060.
##  5 11    District of Columbia  177945   1432     705749              2029.
##  6 02    Alaska                307655   1486     740995              2005.
##  7 49    Utah                 1090346   5298    3205958              1653.
##  8 72    Puerto Rico          1101469   5823    3754939              1551.
##  9 50    Vermont               152618    929     623989              1489.
## 10 15    Hawaii                380608   1841    1415872              1300.
```

```r
plot_usmap(data = state_performance, regions = "states", values = "deaths_per_million", labels = TRUE, 
  scale_fill_continuous(low = "blue", high = "red", name = "Deaths Per Million", label = scales::comma)
  labs(title = "US States", subtitle = "Deaths per Million by March 2023") +
  theme(legend.position = "right")
```



The chart shows that there is significant variation in deaths per million across the states. Arizon experiencing 4547 deaths per million, whilst Vermont experienced 32% of that at 1489 deaths per million.

This variation triggered me to look a little deeper at one of the possible explanations.

Modelling COVID outcome vs Political Leaning - Is there a correlation between politics and COVID deaths?

A quick inspection of the geographic plot, appeared to show a linkage between politics and COVID outcome. To validate if this was supported by the data, a data set was imported from [https://worldpopulationreview.com/state-rankings/most-republican-states] which shows the 2024 political leanings of each state in the union. With more time, a political landscape for each year of the COVID period would be preferable, but for the sake of this academic exercise, the 2024 data provides an approximation. Then, a quick correlation analysis between the degree of Republican political advantage and the deaths per million experienced in that state was performed. This was supplemented with a t-test to reject or confirm the hypothesis that mean performance (in terms of deaths per million) is significantly different between Democratic and Republican states.

There are potential issues with this assessment:

1. As mentioned, the data source shows 2024 political alignment, which has almost certainly changed since the period under analysis.
2. The correlation is only moderate and might be confused by other factors that are also impacting the outcomes from COVID. For example, rural areas vs city areas could be more important than the politics.
3. Personal bias that the democrat policies for dealing with COVID seemed to make more sense, which could influence my thinking. (Though I have tried to avoid this.)
4. Correlation does not necessarily mean causation.

```
state_covid_politics <- state_performance %>%
  inner_join(political_leaning, by = join_by(Province_State == state))
```

```r
state_covid_politics
```

```
## # A tibble: 50 x 9
##    fips  Province_State   cases deaths Population deaths_per_million PVI
##    <chr> <chr>            <dbl>  <dbl>      <dbl>              <dbl> <chr>
##  1 04    Arizona        2443514  33102    7278717              4548. R+2
##  2 40    Oklahoma       1290929  17972    3956971              4542. R+20
##  3 28    Mississippi     990756  13370    2976149              4492. R+11
##  4 54    West Virginia   642760   7960    1792147              4442. R+22
##  5 35    New Mexico      670929   9061    2096829              4321. D+3
##  6 05    Arkansas       1006883  13020    3017804              4314. R+16
##  7 01    Alabama        1644533  21032    4903185              4289. R+15
##  8 47    Tennessee      2515130  29263    6829174              4285. R+14
##  9 26    Michigan       3064125  42205    9986857              4226. R+1
## 10 21    Kentucky       1718471  18130    4467673              4058. R+16
## # i 40 more rows
## # i 2 more variables: republican_adv <dbl>, lean <chr>
```

```r
political_correlation = round(cor(state_covid_politics$deaths_per_million, state_covid_politics$republi
# just to see if there is any linkage with the most populous states (seems not)
population_correlation = round(cor(state_covid_politics$deaths_per_million, state_covid_politics$Popula


# A statistical test for the difference in performance between Republican and Democratic states
rep_dem <- state_covid_politics %>%
  group_by(lean) %>%
  summarise(mean_perf = mean(deaths_per_million), var_perf = var(deaths_per_million), n = n())

rep_dem
```

```
## # A tibble: 2 x 4
##   lean  mean_perf var_perf     n
##   <chr>     <dbl>    <dbl> <int>
## 1 D         2856.  735296.    19
## 2 R         3625.  525410.    31
```

```r
# comparing variances
SD_sq = sum(rep_dem[1,3])
SR_sq = sum(rep_dem[2,3])
meanD = sum(rep_dem[1,2])
meanR = sum(rep_dem[2,2])
nD = sum(rep_dem[1,4])
nR = sum(rep_dem[2,4])
alpha = .05


# Use an F-Statistic to determine if variances are similar for both populations
crit_val = SD_sq/SR_sq
F_stat = qf(p = 1-alpha/2, df1 = nD, df2 = nR)

if(crit_val > F_stat) {
  similar_variance = FALSE
```

```r
} else {
  similar_variance = TRUE
}
str_c("A critical value of ", round(crit_val, 3), " compared to an F-statistic of ", round(F_stat, 3),
      " means the null hypothesis of similar variance is likely ", similar_variance,
      " with a ", 1-alpha, " confidence level.")
```

```
## [1] "A critical value of 1.399 compared to an F-statistic of 2.197 means the null hypothesis of simil
```

```r
# calculate pooled variance
SP_sq = ((nD-1)*SD_sq + (nR-1)*SR_sq)/(nD + nR - 2)

# calculate crit value for t-test and
cee = qt(p = 1-alpha/2, df = (nD + nR -2)) * sqrt((1/nD + 1/nR)*SP_sq)
if((meanR-meanD) > cee) {
  H1 = TRUE
  msg = str_c("We can reject the null hypothesis at the ", 1-alpha,
              " confidence level and conclude that Democratic states do perform better than Republican s
} else {
  H1 = FALSE
  msg = str_c("We cannot reject the null hypothesis at the ", 1-alpha,
              " confidence level, there is no significant difference in performance between Democratic a
}
msg
```

```
## [1] "We can reject the null hypothesis at the 0.95 confidence level and conclude that Democratic stat
```

Conclusion

Whilst noting some potential biases (see above), a correlation of 0.41 indicates a **moderate positive correlation** between Republican leaning politics and poor COVID outcomes, assessed by deaths per million in the state during the period 2020-01-22 to 2023-03-09. This correlation figure aligns with a visual interpretation of the plot and the tabular data associated. Nine of the top 10 states ranked by death per million are Republican leaning, whereas 7 of the 10 best performing states lean towards Democrat. A t-test at 95% confidence level confirmed the hypotheses that Democratic states perform statistically significantly better than Republican leaning states in terms of deaths per million during the COVID period. Issues with bias and confounding variables (e.g. rural vs city) are noted meaning it is hard to assert any kind of causal relationship between the two.

sessionInfo

```r
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14)
## Platform: aarch64-apple-darwin20
## Running under: macOS Sonoma 14.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
```

```
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LAPACK ve
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/Lisbon
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] usmap_0.7.1     zoo_1.8-12      scales_1.3.0    lubridate_1.9.3
##  [5] forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2
##  [9] readr_2.1.5     tidyr_1.3.1     tibble_3.2.1    ggplot2_3.5.1
## [13] 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      lattice_0.22-6     hms_1.1.3          digest_0.6.36
##  [9] magrittr_2.0.3     evaluate_0.24.0    grid_4.4.1         timechange_0.3.0
## [13] fastmap_1.2.0      e1071_1.7-14       DBI_1.2.3          tinytex_0.51
## [17] fansi_1.0.6        cli_3.6.3          rlang_1.1.4        crayon_1.5.3
## [21] units_0.8-5        bit64_4.0.5        munsell_0.5.1      withr_3.0.0
## [25] yaml_2.3.8         tools_4.4.1        parallel_4.4.1     tzdb_0.4.0
## [29] usmapdata_0.3.0    colorspace_2.1-0   curl_5.2.1         vctrs_0.6.5
## [33] R6_2.5.1           proxy_0.4-27       classInt_0.4-10    lifecycle_1.0.4
## [37] bit_4.0.5          vroom_1.6.5        pkgconfig_2.0.3    pillar_1.9.0
## [41] gtable_0.3.5       Rcpp_1.0.13        glue_1.7.0         sf_1.0-16
## [45] highr_0.11         xfun_0.45          tidyselect_1.2.1   rstudioapi_0.16.0
## [49] knitr_1.47         farver_2.1.2       htmltools_0.5.8.1  labeling_0.4.3
## [53] rmarkdown_2.27     compiler_4.4.1
```