Wastewater Analysis Using R

Introduction

In this module, we will analyze a wastewater dataset from the CDC to investigate how COVID-19 viral levels change through time.

We will learn ways to manipulate dates in R using the library **lubridate** (part of the Tidyverse package).

Recap: Dates and Times in R

There are two most popular R classes used when working with dates and times:

- `Date` class representing a calendar date
- `POSIXct` class representing a calendar date with hours, minutes, seconds

A date-time object in R is a point on the timeline stored as the number of seconds since *January 1st, 1970, 00:00 UTC*. This date is also called the **Unix epoch** and serves as a reference by computer systems to track and operate with time.

The `lubridate` Package

The lubridate package allows us to work with dates and times in a simple and efficient way.

In this module, we will learn to:

- Input dates in different formats
- Perform basic operations with dates
- Extract components of dates

`lubridate` First Needs to Understand the Date Format

You tell lubridate how to “read” your character object.

```
# YYYY-M-DD
a <- ymd("2024-9-20")

# DD/MM/YYYY
a <- dmy("20/09/2024")

# MM-DD-YY
a <- mdy("09-20-24")
```

Extract the Components of a Date with `lubridate`

Year, month, and day of the month (`mday`):

```
a <- ymd_hms("2024-09-20 3:59:01")
year(a)
```

```
[1] 2024
```

```
month(a)
```

```
[1] 9
```

```
mday(a)
```

```
[1] 20
```

Extract the Components of a Date (Continued)

Day of the year (`yday`), day of the week (`wday`), hour, minute, and seconds:

```
yday(a)
```

```
[1] 264
```

```
wday(a)
```

```
[1] 6
```

```
hour(a)
```

```
[1] 3
```

```
minute(a)
```

```
[1] 59
```

```
second(a)
```

```
[1] 1
```

Perform Operations with Dates

```
a <- ymd("2024-09-20")
b <- ymd("2025-07-15")
b - a
```

```
Time difference of 298 days
```

```
a - ymd("2024-08-20")
```

```
Time difference of 31 days
```

```
year(a) - 1989
```

```
[1] 35
```

Gut Check

What does the package `lubridate` allow us to do?

- Work with dates in R
- Produce time series data
- Plot time series data

Data Used

We will use data reporting the level of SARS-CoV-2 (the virus responsible for COVID-19) viral particles in wastewater.

This data reports the level of SARS-CoV-2 at the state, regional, and national level through time. According to this website, national, regional, and state/territory data represent the median values across all wastewater treatment plants in the respective area.

You can read about the data and how it was collected here: https://www.cdc.gov/nwss/rv/COVID19-statetrend.html

Download the Data

We will use the data located here:

```
dat <- read_csv("http://daseh.org/data/wastewater_COVID-19_State_and_Territory_Trends.csv")
```

Clean Up the Data a Little Bit

Some column names contain `/` and spaces, which are not standard form. We will change them so it is easier to work with them.

To do so, we use the function `clean_names` in the `janitor` package.

```
library(janitor)
dat1 <- dat |> clean_names()
```

Clean Up the Data a Little Bit

Look at the column names of the old dataset (called `dat`) and the column names of the new dataset (called `dat1`). The new names are all in lower case with underscores between words.

```
colnames(dat)
```

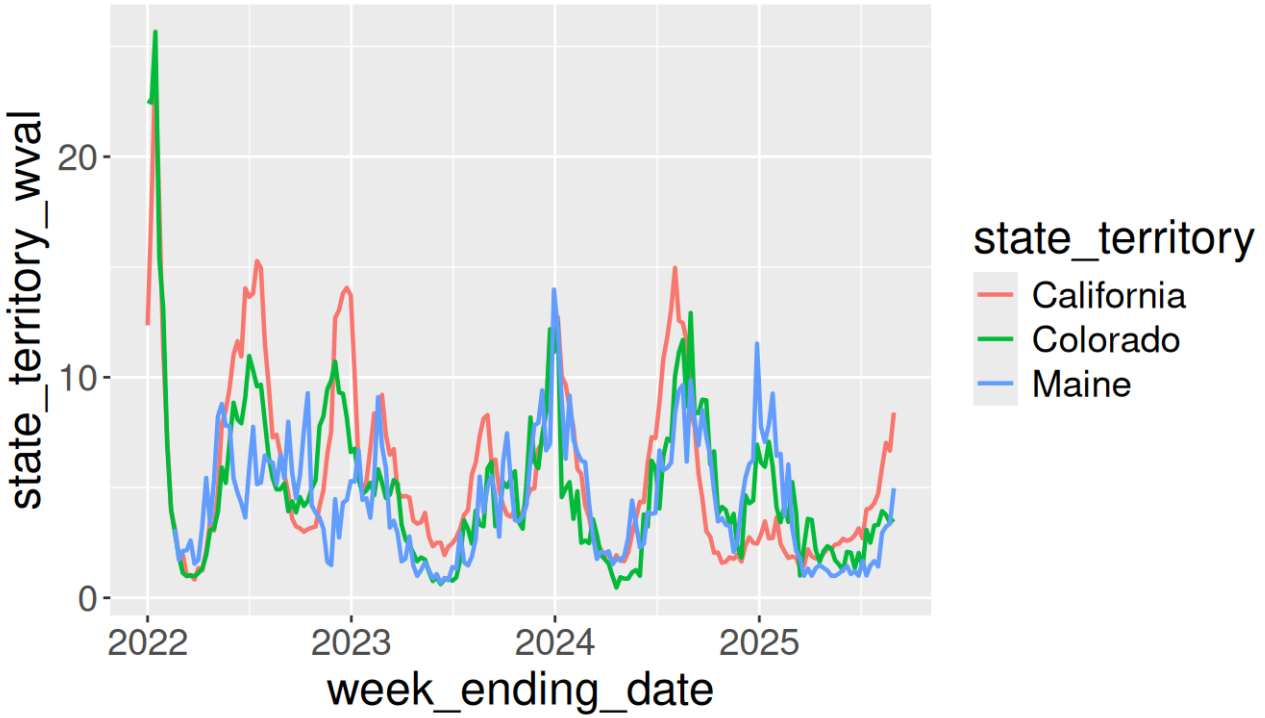
```
[1] "State/Territory"        "Week_Ending_Date"       "Data_Collection_Period"
[4] "State/Territory_WVAL"   "National_WVAL"          "Regional_WVAL"
[7] "WVAL_Category"          "Coverage"               "date_updated"
```

```
colnames(dat1)
```

```
[1] "state_territory"        "week_ending_date"       "data_collection_period"
[4] "state_territory_wval"   "national_wval"          "regional_wval"
[7] "wval_category"          "coverage"               "date_updated"
```

Mission: Understand the Trends of COVID-19 Disease Through Time in Different Parts of the United States

Let's imagine we want to understand how SARS-CoV-2 infections (and hence COVID-19 disease) change through time in three different parts of the US: the East coast, the West coast, and the Midwest. We want to obtain plots that should look like this:



Clean Up the Data (Continued)

We will not be using the columns `Coverage` and `date_updated`, so we will remove them from our data frame. We will conduct our analysis using all the results available, so we will remove the rows with results at 1 year or every 6 months.

```
dat2 <- dat1 |> select(!c(coverage, date_updated)) |> filter(data_collection_period == "All Results")
head(dat2)
```

```
# A tibble: 6 × 7
  state_territory week_ending_date data_collection_period state_territory_wval
  <chr>           <date>           <chr>                                  <dbl>
1 Tennessee       2022-04-02       All Results                            0.987
2 Tennessee       2022-02-12       All Results                            2.45
3 Tennessee       2022-01-01       All Results                            3.37
4 Tennessee       2022-01-08       All Results                            1.31
5 Wyoming         2023-09-23       All Results                            3.94
6 Vermont         2022-05-07       All Results                           13.9
# ℹ 3 more variables: national_wval <dbl>, regional_wval <dbl>,
#   wval_category <chr>
```

Select 3 States to Look At

We will select three states in the US in three different parts (East, Center, West) to look at their wastewater data: South Carolina, Nebraska, and Washington.

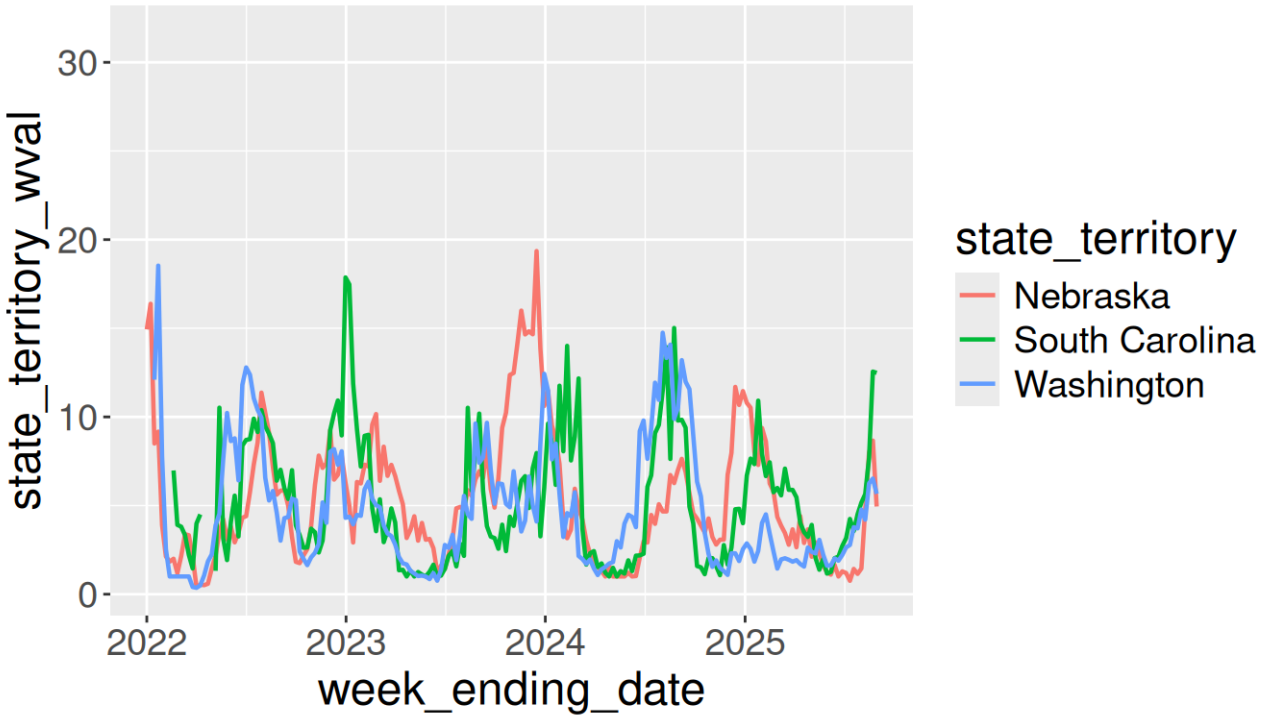
```
df <- dat2 |> filter(state_territory %in% c('South Carolina', 'Nebraska', 'Washington'))
head(df)
```

```
# A tibble: 6 × 7
  state_territory week_ending_date data_collection_period state_territory_wval
  <chr>           <date>           <chr>                                 <dbl>
1 South Carolina  2024-02-24       All Results                            8.89
2 South Carolina  2024-03-16       All Results                            1.67
3 South Carolina  2025-01-25       All Results                           10.9
4 South Carolina  2023-12-30       All Results                            5.99
5 South Carolina  2024-04-06       All Results                            1.53
6 South Carolina  2024-06-01       All Results                            1.91
# ℹ 3 more variables: national_wval <dbl>, regional_wval <dbl>,
#   wval_category <chr>
```

Look at the Data

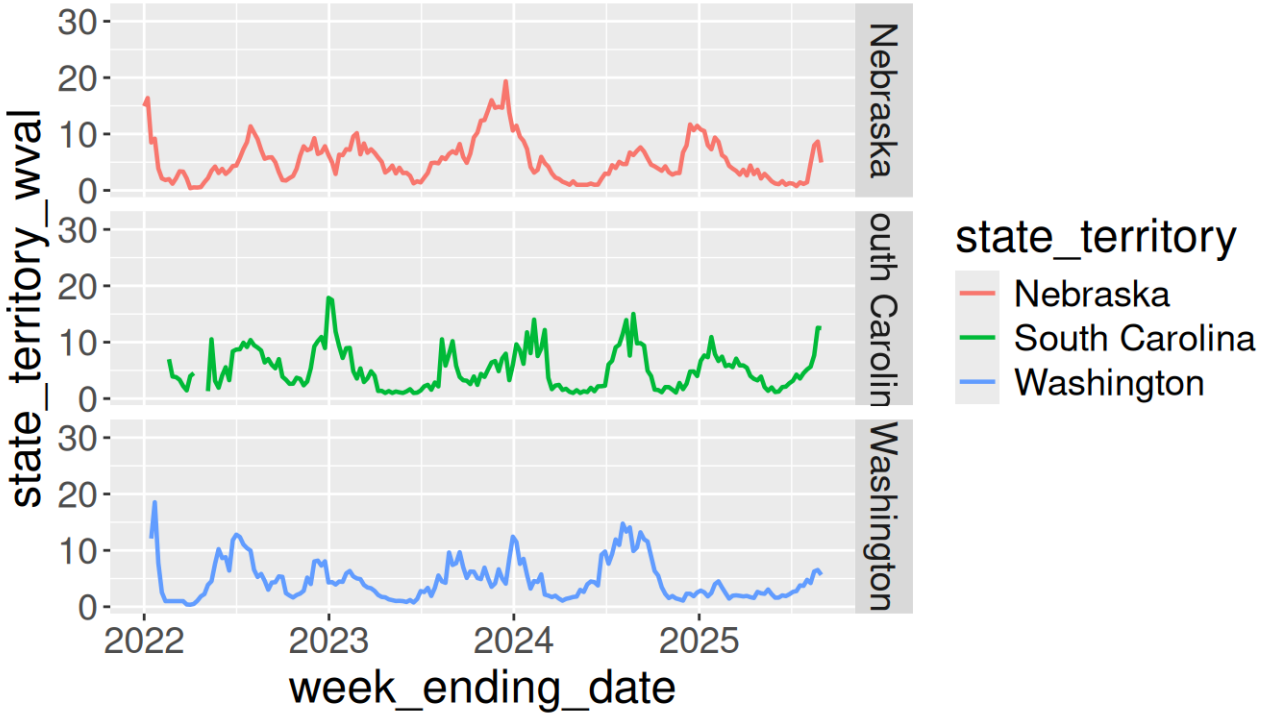
We will plot the data using ggplot:

```
ggplot(data = df,
       aes(x = week_ending_date, y = state_territory_wval, color = state_territory)) +
  geom_line(linewidth = 0.8) +
  theme(text = element_text(size = 18))
```



Split the Plots to See the Patterns More Clearly

```
ggplot(data = df,
       aes(x = week_ending_date, y = state_territory_wval, color = state_territory)) +
  geom_line(linewidth = 0.8) + theme(text = element_text(size = 18)) + facet_grid(state_territory ~ .)
```



What Can We Say About This Data?

- There is a seasonality pattern for each state
- The peaks of the 2023-2024 winter season occurred at slightly different times in each state
- Most of the big waves happen in winter time, but all three states had a summer outbreak in 2024

Zoom In With the Data

Now, we want to investigate the size of the summer outbreak in each state in 2024.

We use `lubridate` to select only the data from year 2024:

Quick Quiz

Do you remember how to extract the year from a date called `mydate`?

1. `year(mydate)`
2. `extract(year, mydate)`
3. `mydate.year()`

Extract the Data for Year 2024

```
df2024 <- df |> filter(year(week_ending_date) == 2024)
head(df2024)
```

```
# A tibble: 6 × 7
  state_territory week_ending_date data_collection_period state_territory_wval
  <chr>           <date>           <chr>                                  <dbl>
1 South Carolina  2024-02-24       All Results                             8.89
2 South Carolina  2024-03-16       All Results                             1.67
3 South Carolina  2024-04-06       All Results                             1.53
4 South Carolina  2024-06-01       All Results                             1.91
5 South Carolina  2024-12-28       All Results                             4.01
6 South Carolina  2024-04-27       All Results                             1
# ℹ 3 more variables: national_wval <dbl>, regional_wval <dbl>,
#   wval_category <chr>
```

Obtain the Peak Viral Level in Each State

First, we need to group the data by state. We use the function `group_by` from `dplyr` to do it:

```
maxRows <- df2024  |> group_by(state_territory) |>
```

Then retrieve the rows with the peak level:

```
maxRows <- df2024  |> group_by(state_territory) |> ???
```

How do we do this?

Quick Quiz

Do you remember what function to use to obtain the peak level?

1. `max(dat)`
2. `min(dat)`
3. `unique(dat)`

Obtain the Peak Viral Level in Each State

To do this, we need to group the data by state and retrieve the rows with the maximum level:

```
maxRows <- df2024  |> group_by(state_territory) |>
          filter(state_territory_wval == max(state_territory_wval, na.rm=TRUE))
maxRows
```

```
# A tibble: 3 × 7
# Groups:   state_territory [3]
  state_territory week_ending_date data_collection_period state_territory_wval
  <chr>           <date>           <chr>                                 <dbl>
1 Washington      2024-08-03       All Results                            14.7
2 South Carolina  2024-08-24       All Results                            15.0
3 Nebraska        2024-12-14       All Results                            11.7
# ℹ 3 more variables: national_wval <dbl>, regional_wval <dbl>,
#   wval_category <chr>
```

Extract the Peak Wastewater Levels for Each State

We will compute the date of the summer peak in each state:

```
Nebraska_peak <- maxRows %>% filter(state_territory=="Nebraska") %>%
                             pull(week_ending_date)
South_Carolina_peak <- maxRows %>% filter(state_territory=="South Carolina") %>%
                                   pull(week_ending_date)
Washington_peak <- maxRows %>% filter(state_territory=="Washington") %>%
                               pull(week_ending_date)
Nebraska_peak
```

```
[1] "2024-12-14"
```

```
South_Carolina_peak
```

```
[1] "2024-08-24"
```

```
Washington_peak
```

```
[1] "2024-08-03"
```

Days Between Peaks

Compute the Number of Days Between the Peak Wastewater Levels for Each State During the Summer Wave:

Difference between South Carolina and Nebraska:

```
Nebraska_peak - South_Carolina_peak
```

```
Time difference of 112 days
```

Difference between South Carolina and Washington:

```
South_Carolina_peak - Washington_peak
```

```
Time difference of 21 days
```

Difference between Nebraska and Washington:

```
Nebraska_peak - Washington_peak
```

```
Time difference of 133 days
```

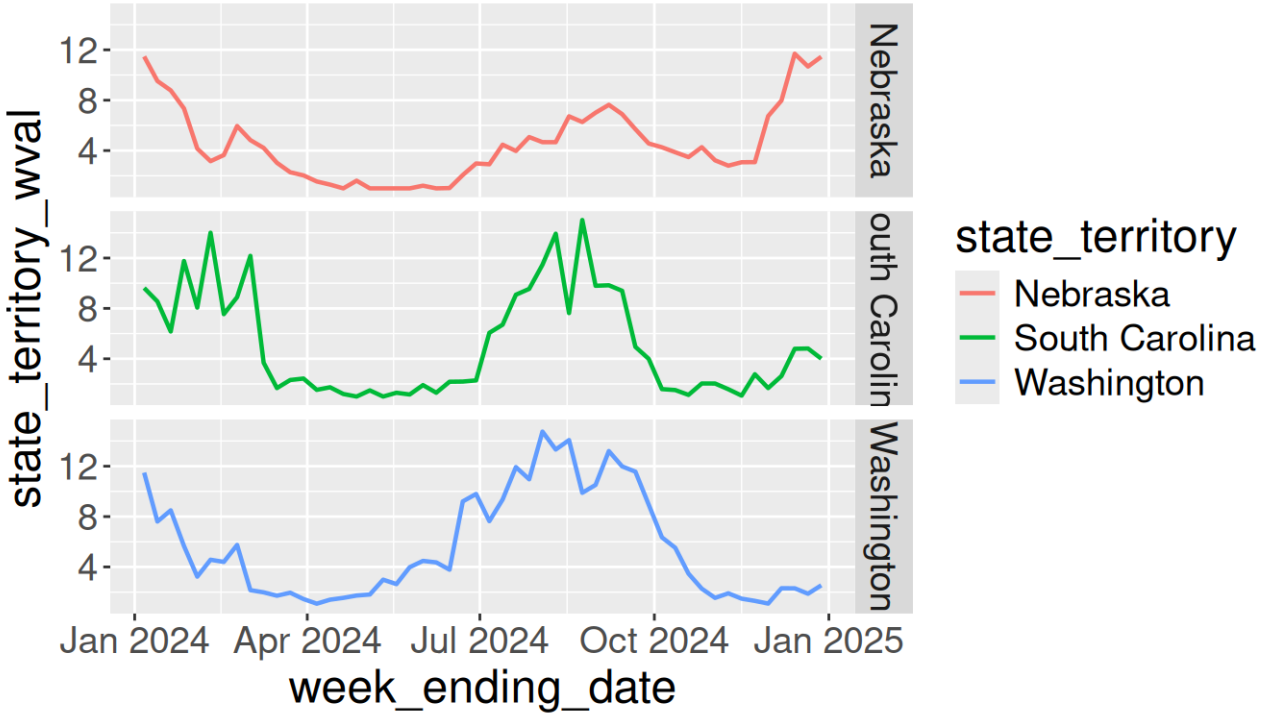
Summer Wave

From this data, we can see that South Carolina and Washington had a summer wave that peaked 21 days apart.

However, the maximum calculated for Nebraska corresponds to the winter wave, not to the summer wave.

Let's Look Again at the Plot, but This Time Restricted to the Year 2024

```
ggplot(data = df2024,
       aes(x = week_ending_date, y = state_territory_wval, color = state_territory)) +
  geom_line(linewidth = 0.8) + theme(text = element_text(size = 18)) + facet_grid(state_territory ~ .)
```



Summer Wave (Continued)

How can we obtain the day that the wastewater measurement reached the maximum level for Nebraska in the summer?

Summer Wave (Continued)

We will further zoom our attention to the summer months and modify the code to look at the data for the months of June, July, and August 2024:

```
maxRows <- df2024 |> filter(month(week_ending_date) %in% c(06, 07, 08)) |>
                     group_by(state_territory) |>
                     filter(state_territory_wval == max(state_territory_wval, na.rm=TRUE))

maxRows
```

```
# A tibble: 3 × 7
# Groups:   state_territory [3]
  state_territory week_ending_date data_collection_period state_territory_wval
  <chr>           <date>           <chr>                                 <dbl>
1 Washington      2024-08-03       All Results                           14.7
2 South Carolina  2024-08-24       All Results                           15.0
3 Nebraska        2024-08-31       All Results                            7.01
# ℹ 3 more variables: national_wval <dbl>, regional_wval <dbl>,
#   wval_category <chr>
```

Re-compute the Peaks in the Summer

Now we can re-compute the number of days between the peak wastewater levels for each state during the summer wave:

```
Nebraska_peak <- maxRows %>% filter(state_territory=="Nebraska") %>%
                             pull(week_ending_date)
South_Carolina_peak <- maxRows %>% filter(state_territory=="South Carolina") %>%
                                   pull(week_ending_date)
Washington_peak <- maxRows %>% filter(state_territory=="Washington") %>%
                               pull(week_ending_date)
Nebraska_peak
```

```
[1] "2024-08-31"
```

```
South_Carolina_peak
```

```
[1] "2024-08-24"
```

```
Washington_peak
```

```
[1] "2024-08-03"
```

Number of Days Between Summer Peaks

Difference between South Carolina and Nebraska:

```
Nebraska_peak - South_Carolina_peak
```

```
Time difference of 7 days
```

Difference between South Carolina and Washington:

```
South_Carolina_peak - Washington_peak
```

```
Time difference of 21 days
```

Difference between Nebraska and Washington:

```
Nebraska_peak - Washington_peak
```

```
Time difference of 28 days
```

Conclusion

- South Carolina and Washington had a summer wave that peaked 21 days apart
- Nebraska and Washington had a summer wave that peaked 28 days apart
- Nebraska and South Carolina had a summer wave that peaked one week apart