Session 5: Contagion and Diffusion

Jennifer Lin

2022-05-04

The goal of this R section is to discuss Contagion and Diffusion models. But beyond these models, our goal will also be to expand comfort and familiarity with R skills that we have discussed earlier in the quarter. These include working with datasets, making plots and wrangling data. We will also discuss new skills, like writing functions. In short, here are our goals:

1. Increase comfort with working with large datasets
2. Become familiar with date objects in R for plots
3. Become more advanced with `ggplot2`, learning different ways to enhance plotting skills

Diffusion Models

In this first section, our data come from a working paper from Lin and Lunz-Trujillo (2022)[1]. In this paper, the authors are interested in studying non-voting political behavior between urban and rural areas in the United States. A part of this analysis looks at the opportunities for protests for people based on where they live. To do this, the authors modified data from the Armed Conflict Location & Event Data Project (ACLED), which is a database that tracks protests and other political violence instances around the world. These data focus on the United States in 2020.

```
load("2020Prot.RData")
```

We use these data in this class to discuss how counties around the country adopt and host political protests on varying issues. Not all counties have protests, and not all counties have protests to a specific issue. Therefore, how do protests for the Black Lives Matter movement spread, geographically, across the country? How do protests related to COVID-19 spread as well?

To tackle this question, we first need to classify protests by their topics:

```
keywords <- c(
  "coronavirus",
  "Black Lives Matter")

m <- sapply(keywords, grepl, protest20$notes)
```

[1] These data are included with the permission from the authors in this class for teaching purposes only. If you are interested in adopting these data for your own work, research or class assignment, please reach out to Jennifer Lin at jenniferlin2025@u.northwestern.edu

```
protest20$keywords <- apply(
  m, 1,
  function(y) paste0(colnames(m)[y], collapse=","))

protest20 <- protest20  %>%
  separate(
    col = "keywords",
    into = c("k1", "k2"),
    sep = ",",
    remove = FALSE
  ) %>%
  filter(event_type == "Protests") %>%
  mutate(
    date = dmy(event_date)
  )
```

The code above basically goes through the notes that ACLED included about each protest to get keywords to figure out, broadly, what each protest was about.

From here, we can explore how counties "get infected" with a political protest. For all protests, we assume that the county is the unit of analysis and that

- **Susceptible**: All counties have the chance to be host to a protest
- **Infected**: Counties are "infected" when they have hosted a protest
- **Recovery**: Post-Protest feelings that may lead to another protest in the same/similar area

Before looking into all of this, let's first take a look at the data, using the data wrangling and plotting skills we learned earlier in the quarter.

First, let's start with the BLM protests. How have BLM protests sprung up and have they been occurring at the same rates throughout the course of 2020? Here are summary statistics to help us figure this out:

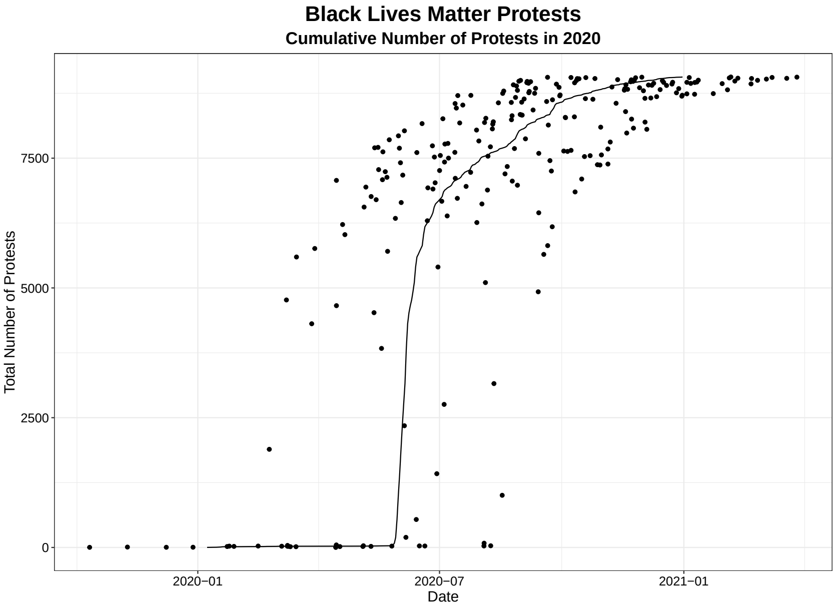
```
blm_prot_time <- protest20 %>%
  filter(k1 == "Black Lives Matter") %>%
  group_by(date) %>%
  summarise(
    n = n()
  ) %>%
  mutate(
    total = cumsum(n),
    days = mdy("01-01-2020") %--% date,
    day_of_year = as.duration(days) / ddays(1)
  )
```

Now that we have these data, we can generate a plot. However, before doing this, here is a theme function that I have so that I do not need to copy and paste the same theme options each time.

```
theme_contagion <- function() {
  theme_bw()+
  theme(
    plot.title         = element_text(
      hjust = 0.5, size = 20, colour="black", face = "bold"),
    plot.subtitle      = element_text(
      hjust = 0.5, size = 16, colour="black", face = "bold"),
    legend.title       = element_text(
      hjust = 0.5, size = 14, colour="black", face = "bold"),
    plot.caption       = element_text(size = 10, colour="black"),
    axis.title         = element_text(size = 14, colour="black"),
    axis.text.x        = element_text(
      size = 12, colour="black"),
    axis.text.y        = element_text(size = 12, colour="black"),
    legend.position    = 'bottom',
    legend.direction   = "horizontal",
    legend.text        = element_text(size = 12, colour="black")
  )
}
```

Now that we have that cleared, let us look at the number of protests about Black Lives Matter since the start of 2020. Note that this is the number of protests, not counties with protests. This will come later.

```
ggplot(blm_prot_time, aes(x = date, y = total, group=1))+
  geom_line()+
  geom_point(position = position_jitter(100))+
  labs(
    title = "Black Lives Matter Protests",
    subtitle = "Cumulative Number of Protests in 2020",
    caption = "Data: Lin and Lunz-Trujillo, 2022
    Adopted from ACLED
    Author: Jennifer Lin"
  )+
  xlab("Date")+
  ylab("Total Number of Protests")+
  theme_contagion()
```



Data: Lin and Lunz−Trujillo, 2022
Adopted from ACLED
Author: Jennifer Lin

As we will recall, BLM protests did not all spring up at once. Events were held in various counties across time, mostly in the late-May to mid-June time span.

Let's model this using a SIR model – to see the rate of "infection" of BLM protests.

To do this, we first need a grand dataset of counties. All we need to know is the total number of susceptible counties, so let's just use the canned data from `tidycensus`[2]

```
data("fips_codes")
```

One important component about the SIR model is understanding adoption. Let's generate some data about this before applying the model:

```
adopt_blm <- protest20 %>%
  filter(k1 == "Black Lives Matter") %>%
  group_by(county) %>%
  filter(date == min(date)) %>%
  slice(1) %>%
  ungroup() %>%
  arrange(date) %>%
  group_by(date) %>%
  summarise(
    n = n()) %>%
  mutate(
    total = cumsum(n),
    pct_adopt = total/nrow(fips_codes)) %>%
```

[2] If you cannot get `tidycensus` to work, pull the data from Canvas from Week 1.

```
  filter(date >= "2020-05-01" & date <= "2020-08-01") %>%
  mutate(
    days = mdy("05-01-2020") %--% date,
    day_of_year = as.duration(days) / ddays(1))
```

The above code gives us the first date in which a county has a BLM protest. Now we are ready to fit an SIR Model. Here is a function that can help us calculate SIR model components in R

```
get_sir <- function(beta, gamma, S0, I0, R0, times) {
  # Equation for Calculating SIR
  sir_equations <- function(time, variables, parameters) {
    with(as.list(c(variables, parameters)), {
      N = S+I+R
      lambda = beta*(I/N)
      dS = -lambda*S
      dI = lambda*S-gamma*I
      dR = gamma*I
      return(list(c(dS, dI, dR)))
    })}
  # Parameter Values
  parameters_values <- c(beta = beta, gamma = gamma)
  # Initial Values
  initial_values <- c(S = S0, I = I0, R = R0)
  # Generate the model
  out <- ode(initial_values, times,
             sir_equations, parameters_values)
  # Return
  as.data.frame(out)
}
```

To use this function, we need to supply the parameter values

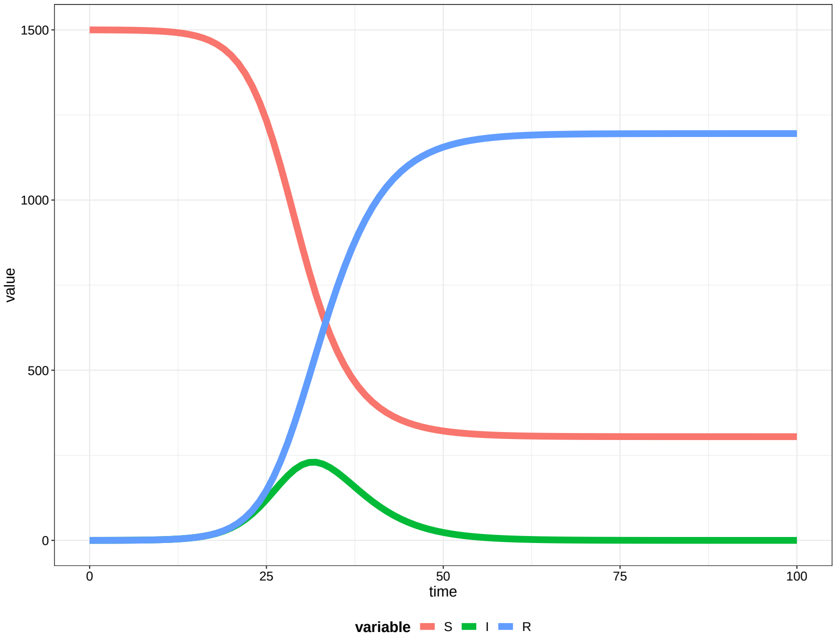
- beta – Infection Rate
- gamma – Recovery Rate
- S0 – Number susceptible at time 0
- I0 – Number infected at time 0
- R0 – Number recovered at time 0
- times – Time interval

We can apply the above function to BLM protests. This is based on rough guessing over a 100 day period.

```
model_BLM <- get_sir(
  beta = 0.6, gamma = 0.3, S0 = 1500, I0 = .1,
  R0 = 0, times = seq(0, 100, 1)) %>%
  reshape2::melt(id = "time")
```

Here is a quick plot for how this model would look like:

```
ggplot(
  model_BLM,
  aes(x = time, y = value,
      group = variable, color = variable))+
  geom_line(size = 3)+
  theme_contagion()
```



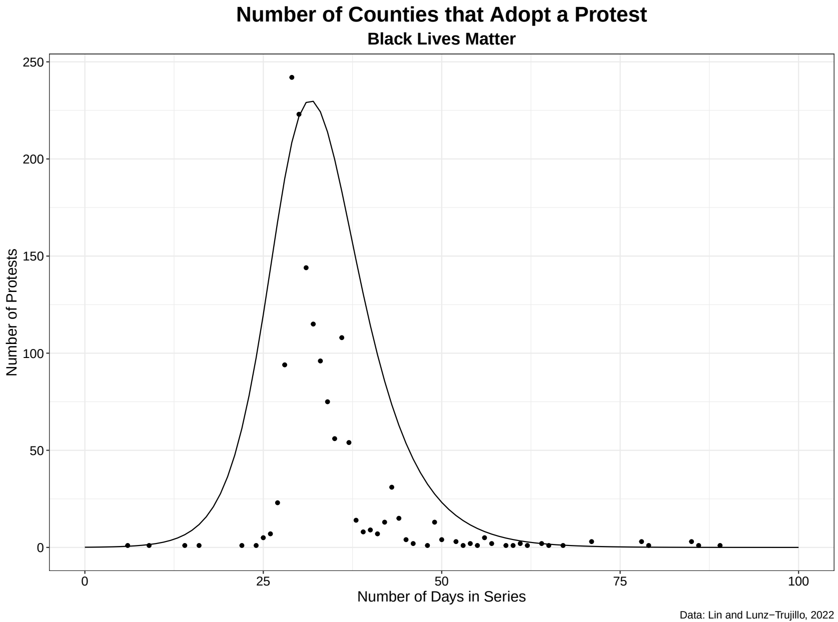
We are interested in the rate of infection of protests. We define “infected” as counties that host a protest. Therefore, our plot for protest adoption concerns the “infected” category. We get those predictions.

Let’s get the curve for infection:

```
I_BLM <- model_BLM %>%
  filter(variable == "I")
```

And let’s see how this curve fits to the data on the number of protests per day over a 100 day period:

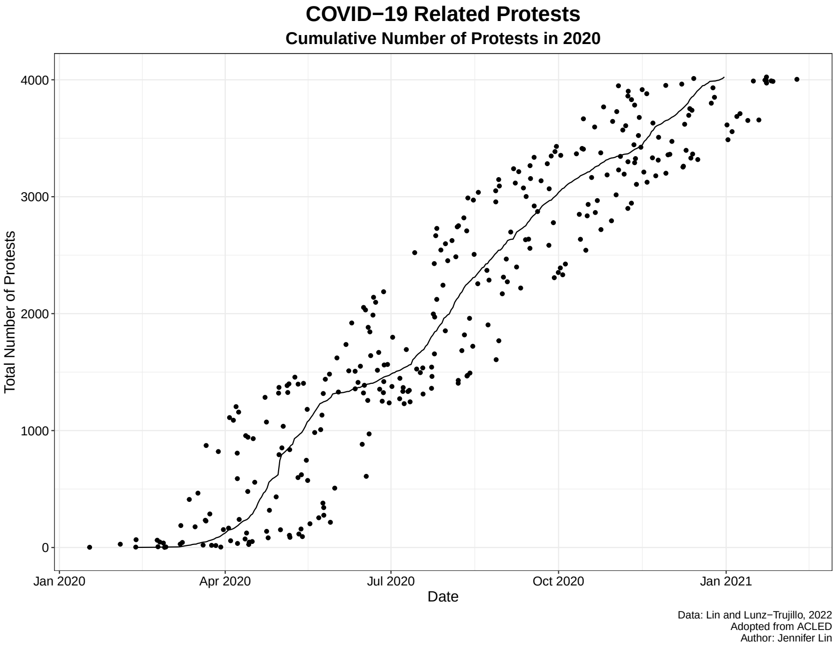
```
ggplot(adopt_blm, aes(x = day_of_year, y = n, group=1))+
  geom_point()+
  geom_line(I_BLM, mapping = aes(x = time, y = value, group = 1))+
  labs(
    title = "Number of Counties that Adopt a Protest",
    subtitle = "Black Lives Matter",
    caption = "Data: Lin and Lunz-Trujillo, 2022
    Adopted from ACLED
    Author: Jennifer Lin"
  )+
  xlab("Number of Days in Series")+
  ylab("Number of Protests")+
  theme_contagion()
```



Data: Lin and Lunz−Trujillo, 2022
Adopted from ACLED
Author: Jennifer Lin

Great! Now, let's repeat this for another protest category – COVID-19. Note that for this demonstration, I am agnostic as to whether the protest was for or against COVID-19 related restrictions. The process is the same as above, which is why I do not include as many comments or code details.

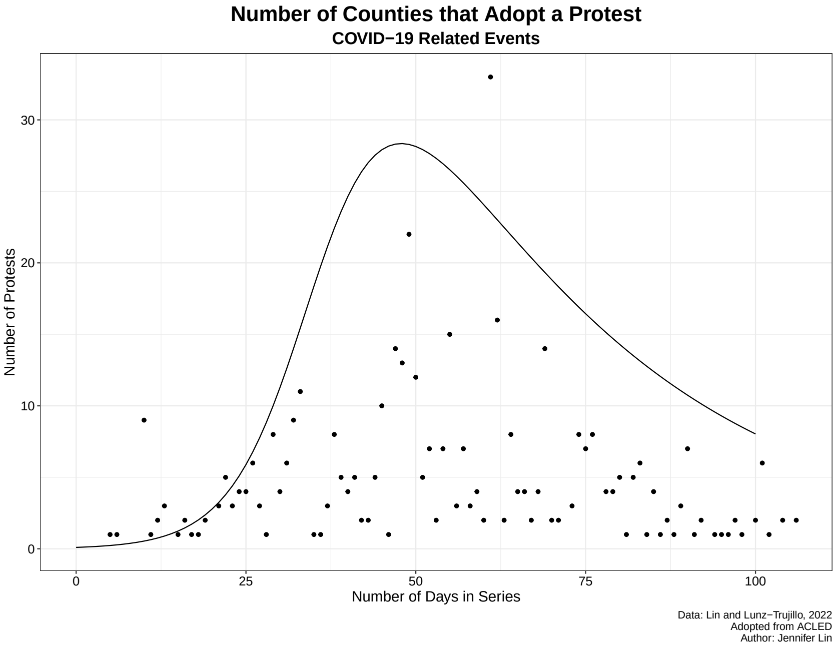
```
covid_prot_time <- protest20 %>%
  filter(k1 == "coronavirus") %>%
  group_by(date) %>%
  summarise(
    n = n()
  ) %>%
  mutate(
    total = cumsum(n)
  )
```



```
adopt_covid <- protest20 %>%
  filter(k1 == "coronavirus") %>%
  group_by(county) %>%
  filter(date == min(date)) %>%
  slice(1) %>%
  ungroup() %>%
  arrange(date) %>%
  group_by(date) %>%
  summarise(
    n = n()) %>%
  mutate(
    total = cumsum(n),
    pct_adopt = total/nrow(fips_codes)) %>%
  filter(date >= "2020-03-01" & date <= "2020-06-15") %>%
  mutate(
    days = mdy("03-01-2020") %--% date,
    day_of_year = as.duration(days) / ddays(1))
```

We can now get the SIR Model and see how it fits.

```
model_covid <- get_sir(
  beta = 0.2, gamma = 0.03, S0 = 50, I0 = .1,
  R0 = 0, times = seq(0, 100, 1)) %>%
  reshape2::melt(id = "time")
```



Contagion Models

Why are some things searched more than others? During COVID times, people are essentially engaging in "doctor googling" their symptoms a lot more, or are they? What are the most searched medical conditions during the pandemic? Did they ebb and flow across time? How do these vary by state?

The data that we are using here covers google searches of medical issues between 2020 and 2021, spanning from January 2020 to December 2021. We will read the data from source:

```
root <- "https://storage.googleapis.com/"
file <- "covid19-open-data/v3/google-search-trends.csv"
trends <- paste0(root, file)
trends_data <- rio::import(trends)
```

This will take a while. If it does not work, use the CSV that is attached with the materials.

Let's take a look at some of the most searched medical conditions. Here is a function to help us get there:

```
get_top_search <- function(key, position){
  loc = as.character(key)
  pos = as.numeric(position)

  most_searched <- trends_data %>%
    filter(location_key == loc) %>%
    gather(key, value, 4:425) %>%
    group_by(date) %>%
    arrange(desc(value)) %>%
    filter(key != "location_key") %>%
    slice(pos)

  return(most_searched)
}
```

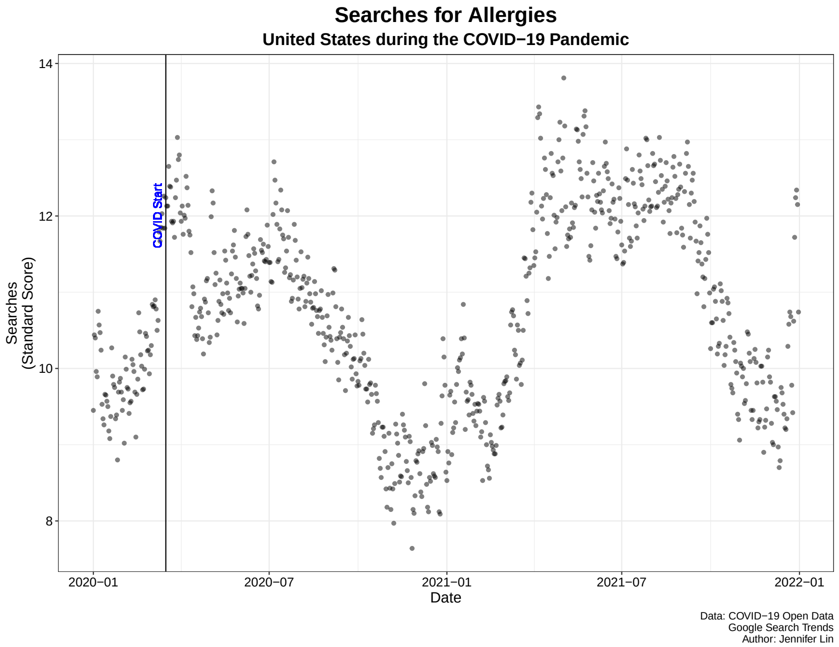
It seems that the top searches in the US include:

- search_trends_infection
- search_trends_pain
- search_trends_allergy

```
most_searched <- get_top_search("US", 1)
```

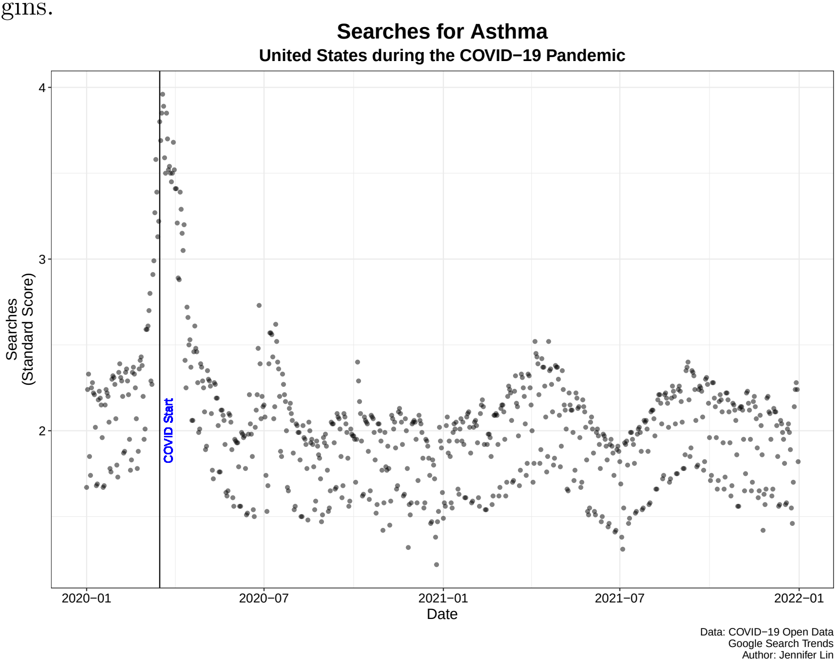
So let's make a plot to reflect the trends in searches before and after the onset of COVID, starting with allergies. Can we noticeably detect allergy seasons?

```
trends_data %>%
  filter(location_key == "US") %>%
  ggplot()+
  geom_point(
    aes(x = date, y = search_trends_allergy), alpha = 0.5)+
  geom_vline(xintercept = ymd("2020-03-16"))+
  geom_text(
    aes(
      x=ymd("2020-03-16"),
      label="COVID Start\n", y = 12),
    colour="blue", angle=90) +
  labs(
    title = "Searches for Allergies",
    subtitle = "United States during the COVID-19 Pandemic",
    caption = "Data: COVID-19 Open Data
    Google Search Trends
    Author: Jennifer Lin")+
  xlab("Date")+
  ylab("Searches\n(Standard Score)")+
  theme_contagion()
```

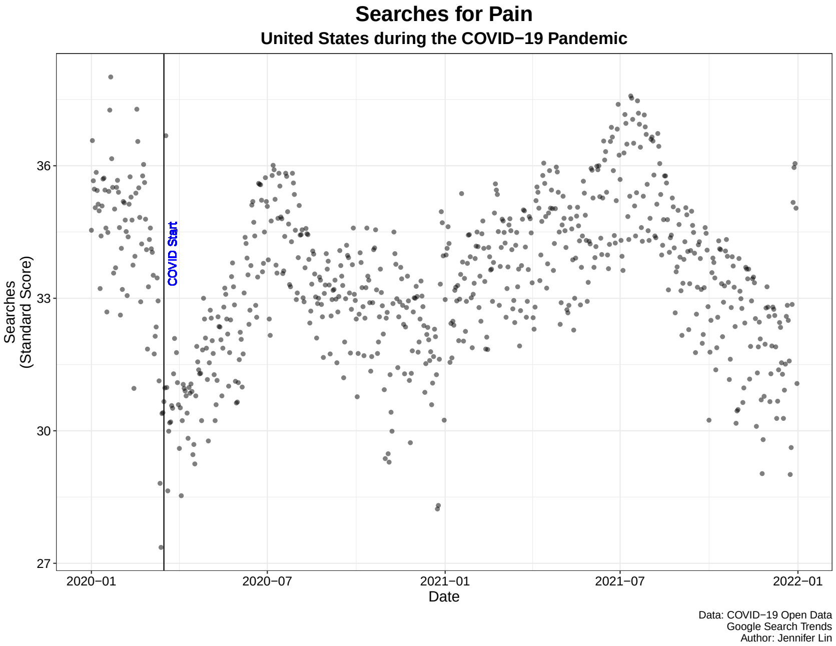

Searches for Allergies
United States during the COVID−19 Pandemic

Data: COVID−19 Open Data
Google Search Trends
Author: Jennifer Lin

Now, let's look at some other medical conditions. You can get a full list of the conditions included in the data by using `names(trends_data)`. The code is the same, with the exception of replacing the names and the point with which we include the label on the line for when COVID begins.


Searches for Asthma
United States during the COVID−19 Pandemic

Data: COVID−19 Open Data
Google Search Trends
Author: Jennifer Lin



Exercise – Pick One

1. Using the Protests Data, rerun the fuzzy match code and generate a new SIR model for protests related to "election", "school board", "Iran", "fracking", "abortion", "gun violence", or anything else that might be interesting.
2. Find other medical symptoms that people googled in the US, Australia, New Zealand or the UK, and plot the tends. You can use the entire country, a state/province, or the the US, a county (cross reference with FIPS codes data for the county identifier) Try to see if you can identify key dates that might explain the ebbs and flows of the trends.