Final Project - Data Wrangling

George Cruz

11/27/2021

Abstract

Background:

At the onset of Covid cases, I started a project for a social tracker. The idea was to allow people to anonymously submit their symptoms (if any) with the "naive" intention of tracking the progress of the epidemic once it got to the US. You can find the live page here: https://covidtrack.app/

As part of this project, I also added data from to publicly available sources. This was mainly to have something to show while the project got traction until we got user provided data to show (which never happened).

These two sources were: 1. **https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties.csv**.This is a repository from the NY Times listing new cases and deaths from us counties. It began collecting data in January 2020.

2. **https://covid-19-coronavirus-statistics.p.rapidapi.com/v1/stats. This is an API that provides similar information by US State.

Proposal

Using the NY Times data repository to track and predict (by linear model) the cases/deaths we should be expecting in any particular state, I'd like to correlate that data (and projections) with different mask mandates in those states.

The idea is to track the cases by state on a weekly basis, generate a linear model of expected cases when mandates go into place, and correlate it with actual cases in subsequent weeks. I found data on mask mandates here: https://statepolicies.com/data/graphs/face-masks/

As an alternative comparison, I could compare the cases data with vaccination rates by state. Again, generating a linear model of expected cases by week and how the vaccination rate affected the actual numbers. We could potentially create an interactive application that would allow us to select a state and date range and get the expected vs actual cases, the difference and if a mask mandate was in place or the vaccination rates affected the outcome.

Proposed visualizations

1. State by State cases timeline
2. State by State projected cases during mask mandate vs actual cases
3. State by State projected cases, expected cases, and vaccination rate.
4. State projection of mask mandate in reduction/increase of cases
5. State projection of vaccination rates in reduction/increase of cases

Getting the Data

```
library(here)
```

```
## Warning: package ’here’ was built under R version 4.0.5
```

```
## here() starts at C:/Users/georg/Documents/George/Data Science MS/Data608Homeworks/final_project
```

```
library(dplyr)
```

```
## Warning: package ’dplyr’ was built under R version 4.0.5
```

```
##
## Attaching package: ’dplyr’
```

```
## The following objects are masked from ’package:stats’:
##
##     filter, lag
```

```
## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
## Warning: package ’lubridate’ was built under R version 4.0.3
```

```
##
## Attaching package: ’lubridate’
```

```
## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

```
data <- read.csv(here(’data’,’raw_data’,’nytimes_us_counties.csv’))
```

```


grouped_data <- data %>%
  select(-fips, -county, -X) %>%
  group_by(date, state) %>%
  summarise(across(everything(), sum))
```

Group the daily data by State

```
## ‘summarise()‘ has grouped output by ’date’. You can override using the ‘.groups‘ argument.
```

```
###Save this new data
write.csv(grouped_data, here(’data’, ’processed_data’,’ny_times_by_state.csv’))
```

Calculate new cases per day and add mask mandate We will calculate the new cases per day by substracting cases from the previous day’s total. We will also add a default mask mandate set as No to begin tallying up that data.

```
new_cases_weekly_data <- grouped_data %>%
  group_by(state) %>%
  mutate(date = as.Date(date)) %>%
    arrange(date, .by_group = TRUE) %>%
  mutate(weekday = weekdays(date)) %>%
  filter(weekday == "Tuesday") %>%
    mutate(new_cases = cases - lag(cases, default = first(cases)),
           new_deaths = deaths - lag(deaths, default = first(deaths)),
           mask_mandate = "no")

### save this data
write.csv(new_cases_weekly_data, here(’data’, ’processed_data’,’ny_times_with_new_weekly_cases.csv’))
```

```


mask_mandate_data <- read.csv(
  here(’data’,
       ’raw_data’,
       ’COVID-19 US state policy database (CUSP) - Face Masks.csv’)
  )

state_info_data <- read.csv(
  here(’data’,
       ’raw_data’,
       ’COVID-19 (CUSP) - State Characteristics.csv’
  )
)
```

Mask Mandate Data And Population Data Notes with the Data:

1. Some states did not set a public face mask mandate but a business face mask mandate. For those states, We will use the business mask mandate as the mandate start date.
2. Other states did not have a mandate at all.

```
names(mask_mandate_data)[names(mask_mandate_data) == ’State’] <- ’state’
names(state_info_data)[names(state_info_data) == ’State’] <- ’state’

###remove columns we will not use and rename the ones we will
mask_mandate_data <- mask_mandate_data %>%
  select(-c(’State.Abbreviation’,
            ’State.FIPS.Code’,
            ’Face.mask.mandate.end.for.fully.vaccinated’,
            ’Face.mask.mandate.resumed.for.fully.vaccinated’,
            ’Face.mask.mandate.in.areas.with.substantial.and.high.COVID.19.transmission.rates’,
            ’Face.mask.mandate.in.schools.for.2021.22.school.year’,
            ’Public.face.mask.mandate.currently.in.place.for.everyone’,
            ’Banned.school.face.mask.mandates’,
            ’Banned.other.local.face.mask.mandates’,
            ’Judicial.decision.blocked.state.from.enforcing.bans.on.mask.mandates.in.schools’,’Notes’))
  mutate(
    mandate_start = as.Date(ifelse(Public.face.mask.mandate.start != 0,
                                   Public.face.mask.mandate.start,
                                   Business.face.mask.mandate.start),
                                                  format=’%m/%d/%Y’),
         mandate_start_2 = as.Date(Public.face.mask.mandate.start.x2,
                                                     format=’%m/%d/%Y’),
         mandate_end = as.Date(Face.mask.mandate.end,
                                         format=’%m/%d/%Y’),
         mandate_end_2 = as.Date(Face.mask.mandate.end.x2,
                                            format=’%m/%d/%Y’)
  ) %>%
  rename(
        fine_enforced = Face.mask.mandate.enforced.by.fines,
        charge_enforced = Face.mask.mandate.enforced.by.criminal.charge.citation,
        not_enforced = No.legal.enforcement.of.face.mask.mandate
  ) %>%
  select(-c(
    Business.face.mask.mandate.start,
    Public.face.mask.mandate.start,
    Public.face.mask.mandate.start.x2,
    Face.mask.mandate.end,
    Face.mask.mandate.end.x2
  )) %>%
  filter(state != "Total")
```

```
mask_mandate_data <- mask_mandate_data %>%
  select(c(state, mandate_start, mandate_end))
```

```
state_info_data <- state_info_data %>%
  select(state, Population.2018) %>%
  rename(population = Population.2018)%>%
  filter(!is.na(population))

us_population = sum(state_info_data$population)

state_info_data$us_population <- us_population
```

```
joined_data <- merge(x = new_cases_weekly_data,
                     y = mask_mandate_data,
                     by=’state’,
                     all=TRUE)

joined_data <- merge(x=joined_data,
                     y=state_info_data,
                     by=’state’,
                     all=TRUE)
```

```
write.csv(joined_data, here(’data’,
                            ’processed_data’,
                            ’weekly_cases_data_plus_mandates.csv’))
```

Update Mask Mandate data to correctly tag if it was in effect or not

```
joined_data <- read.csv(here(’data’,
                            ’processed_data’,
                            ’weekly_cases_data_plus_mandates.csv’))
joined_data <- joined_data %>%
  mutate(date = as.Date(date),
         mandate_start = as.Date(mandate_start),
         mandate_end = as.Date(mandate_end))
active_mandate <- function(date, start_date, end_date, prev = "no") {

  if (is.na(date) || is.na(start_date) || is.na(end_date)) {
    return(prev)
  }

  return(ifelse(between(date, mandate_start, mandate_end), "yes", prev))
}
```

```
data_with_mandates <- joined_data %>%
                      rowwise() %>%
                      mutate(mask_mandate = ifelse(is.na(mandate_start),
                                                   "no",
                                                   if_else(is.na(mandate_end), "no",
                                                   if_else(date >= mandate_start && date <= mandate_end
                                                           "yes",
                                                           "no"
                                                           )
                                                   )
                                                   )
                             )

write.csv(data_with_mandates, here(’data’,
                            ’processed_data’,
                            ’weekly_cases_data_plus_mandates-2.csv’))
```

```
#Aggregate other totals
data_with_mandates %<>%
  rowwise() %>%
  mutate(prop_cases = cases/population * 1000,
         prop_deaths = deaths/population * 1000,
         deaths_per_cases = deaths/cases * 100)
```

```
# Basic proyection
data_with_mandates %<>%
  mutate(date = as.Date(date)) %>%
  arrange(date) %>%
  group_by(state) %>%
  mutate(weeks = (as.numeric(date - dplyr::first(date)) %/% 7) + 1) %>%
  select(-X)

write.csv(data_with_mandates, here(’data’,
                            ’processed_data’,
                            ’weekly_cases_data_plus_mandates-3.csv’))
```

Now that we have the data, we will clean it up in three parts: -Before mandate -During Mandate -After Mandate

```
data_with_mandates <- read.csv(here(’data’,
                            ’processed_data’,
                            ’weekly_cases_data_plus_mandates-3.csv’))

data_with_mandates %<>%
  mutate(date = as.Date(date)) %>%
  arrange(date) %>%
  group_by(state) %>%
  mutate(cases_proyected = if_else(weeks > 2, sqrt(lag(cases)/lag(cases,2))*lag(cases),0),
         deaths_proyected = if_else(weeks > 2, sqrt(lag(deaths)/lag(deaths,2))*lag(deaths),0),
         new_cases_proyected = cases_proyected - lag(cases_proyected),
         new_deaths_proyected = deaths_proyected - lag(deaths_proyected)
  )

write.csv(data_with_mandates, here(’data’,
                            ’processed_data’,
                            ’weekly_cases_data_plus_mandates-4.csv’))
```

```
clean_data <- data_with_mandates %>%
  filter(new_cases > 0)

before_mandate <- clean_data %>%
  group_by(date, state) %>%
  filter(ifelse(is.na(mandate_start),
                mask_mandate == "no",
                date < mandate_start))

during_mandate <- clean_data %>%
  group_by(date, state) %>%
  filter(ifelse(is.na(mandate_end),
                mask_mandate == "yes", date > mandate_start &&
                  date <= mandate_end))

after_mandate <- clean_data %>%
  group_by(date, state) %>%
  filter(date < mandate_end)

write.csv(before_mandate, here(’data’,
                            ’processed_data’,
                            ’before_mandate.csv’))

write.csv(during_mandate, here(’data’,
                            ’processed_data’,
                            'during_mandate.csv'))

write.csv(after_mandate, here('data',
                            'processed_data',
                            'after_mandate.csv'))
```