covid_project

Jaehee Lee

11/27/2020

Task1: What days did states have their first case and the rate? Graph the rates in a plot with respect to time. Try different kinds of plots potentially coloring or categorizing by a variable suca as state or region.

```
library(COVID19)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2          v purrr   0.3.4
## v tibble  3.0.4          v dplyr   1.0.2
## v tidyr   1.1.2.9000     v stringr 1.4.0
## v readr   1.4.0          v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
covid19("US", level = 2) %>%
  select(-c("administrative_area_level_1","administrative_area_level_3","latitude","longitude","currency
  relocate("administrative_area_level_2", "date", "confirmed", "tests")-> easy_covid_df
```

```
## We have invested a lot of time and effort in creating COVID-19 Data Hub, please cite the following wh
##
##   Guidotti, E., Ardia, D., (2020), "COVID-19 Data Hub", Journal of Open
##   Source Software 5(51):2376, doi: 10.21105/joss.02376.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
```

```
##     title = {COVID-19 Data Hub},
##     year = {2020},
##     doi = {10.21105/joss.02376},
##     author = {Emanuele Guidotti and David Ardia},
##     journal = {Journal of Open Source Software},
##     volume = {5},
##     number = {51},
##     pages = {2376},
##   }
##
## To retrieve citation and metadata of the data sources see ?covid19cite. To hide this message use 've
```

```
easy_covid_df %>%
  group_by(administrative_area_level_2, confirmed) %>%
  # mutate(first_date = min(date)) %>%
  select(administrative_area_level_2, date, confirmed) %>%
  arrange(date, administrative_area_level_2) %>%
  group_by(administrative_area_level_2) %>%
  filter(confirmed>0) %>%
  mutate(daily_rate = c(0,diff(confirmed))) %>%
  mutate(second_rate= c(0,diff(daily_rate))) %>%
  arrange(administrative_area_level_2, date) -> maindata
```

Get mean of confirmed cases for each States

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

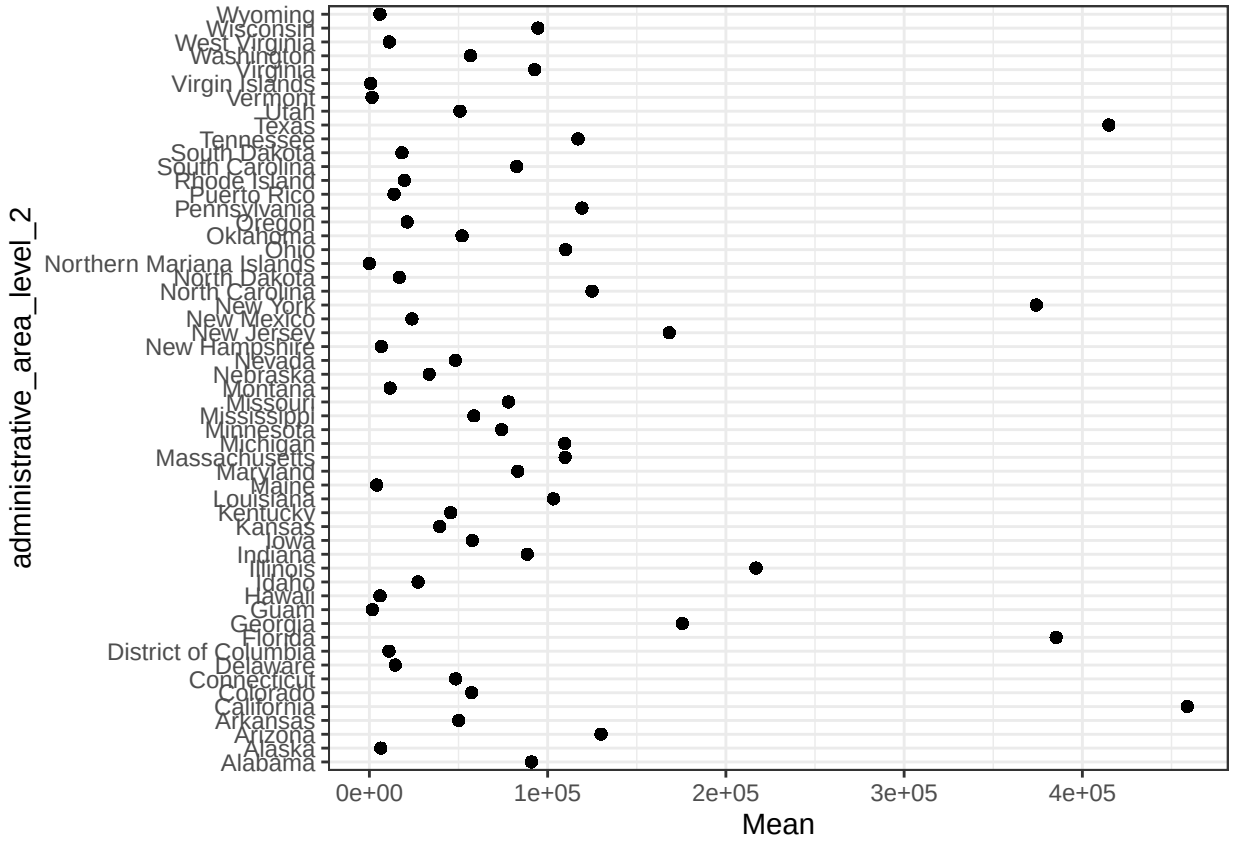
```
meandata <- setDT(maindata)[,list(Mean = as.numeric(mean(confirmed))), by=administrative_area_level_2]
```

inner join with maindata and meandata

```
inner_join(meandata, maindata, by = "administrative_area_level_2") -> a
```

Plot mean for every states

```
ggplot(data = a, aes(x = Mean, y = administrative_area_level_2))+
  geom_point()+
  theme_bw()
```



What day did states have their first case?

```
library(dplyr)
maindata %>%
    group_by(administrative_area_level_2) %>%
    arrange(date) %>%
    slice(1L) %>%
    select(administrative_area_level_2, date) -> first_case_day
```

show the states n

```
unique(maindata$administrative_area_level_2)
```

```
##  [1] "Alabama"                  "Alaska"
##  [3] "Arizona"                  "Arkansas"
##  [5] "California"               "Colorado"
##  [7] "Connecticut"              "Delaware"
##  [9] "District of Columbia"     "Florida"
## [11] "Georgia"                  "Guam"
## [13] "Hawaii"                   "Idaho"
## [15] "Illinois"                 "Indiana"
## [17] "Iowa"                     "Kansas"
## [19] "Kentucky"                 "Louisiana"
## [21] "Maine"                    "Maryland"
## [23] "Massachusetts"            "Michigan"
## [25] "Minnesota"                "Mississippi"
## [27] "Missouri"                 "Montana"
## [29] "Nebraska"                 "Nevada"
## [31] "New Hampshire"            "New Jersey"
## [33] "New Mexico"               "New York"
## [35] "North Carolina"           "North Dakota"
## [37] "Northern Mariana Islands" "Ohio"
## [39] "Oklahoma"                 "Oregon"
## [41] "Pennsylvania"             "Puerto Rico"
## [43] "Rhode Island"             "South Carolina"
## [45] "South Dakota"             "Tennessee"
## [47] "Texas"                    "Utah"
## [49] "Vermont"                  "Virgin Islands"
## [51] "Virginia"                 "Washington"
## [53] "West Virginia"            "Wisconsin"
## [55] "Wyoming"
```

Northeast

```
# Northeast
# 11 states there are
#Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, Vermont, Delaware, Maryland, New #Jers
maindata %>%
  filter(administrative_area_level_2 == c("Connecticut", "Maine", "Massachusetts", "New Hampshire","Rho
```

```
## Warning in administrative_area_level_2 == c("Connecticut", "Maine",
## "Massachusetts", : longer object length is not a multiple of shorter object
## length
```

```
library(dplyr)
northeast %>%
    group_by(administrative_area_level_2) %>%
    arrange(date) %>%
```

```
    slice(1L) %>%
    select(administrative_area_level_2, date) -> northeast_first_case

# glimpse(northeast_first_case$date)
northeast_first_date <- as.Date(northeast_first_case$date)
# yday is the function to check how many days past after this year.
northeast_convert_days <- yday(northeast_first_date) - 1 # January 1 = day 0
mean_of_northeast <- mean(northeast_convert_days)
mean_NE_start_date <- as.Date(mean_of_northeast, origin = "2020-01-01")
```

Northeast first case date mean

```
mean_NE_start_date
```

```
## [1] "2020-03-10"
```

```
ggplot(data =northeast, mapping = aes(x = date , y = confirmed))+
  stat_smooth(method = loess,color='black') +
  labs(x = "Time", y= "Confirmed Cases")+
   geom_vline(xintercept = mean_NE_start_date, color = "red", linetype = 2)+
  theme_bw()  -> northeast_confirmed

ggplot(data =northeast, mapping = aes(x = date , y = daily_rate))+
  stat_smooth(method = loess,color='black') +
  labs(x = "Time", y = "Daily Rate")+
     geom_vline(xintercept = mean_NE_start_date, color = "red", linetype = 2)+
  theme_bw()  -> northeast_rate

ggplot(data =northeast, mapping = aes(x = date , y = second_rate))+
  stat_smooth(method = loess,color='black') +
  labs(x = "Time", y = "Second Rate")+
       geom_vline(xintercept = mean_NE_start_date, color = "red", linetype = 2)+
  theme_bw()  -> northeast_second_rate
```

Midwest

```
# 12 states
maindata %>%
  filter(administrative_area_level_2== c("Ohio","Michigan","Indiana","Wisconsin","Illinois","Minnesota"
```

```
## Warning in administrative_area_level_2 == c("Ohio", "Michigan", "Indiana", :
## longer object length is not a multiple of shorter object length
```

```
library(dplyr)
midwest %>%
    group_by(administrative_area_level_2) %>%
```

```
    arrange(date) %>%
    slice(1L) %>%
    select(administrative_area_level_2, date) -> midwest_first_case



# glimpse(midwest_first_case$date)
midwest_first_date <- as.Date(midwest_first_case$date)
# yday is the function to check how many days past after this year.
midwest_convert_days <- yday(midwest_first_date) - 1 # January 1 = day 0
mean_of_midwest <- mean(midwest_convert_days)
mean_MW_start_date <- as.Date(mean_of_midwest, origin = "2020-01-01")
```

Midwest first case date mean

```
mean_MW_start_date
```

```
## [1] "2020-03-10"
```

```
ggplot(data =midwest, mapping = aes(x = date , y = confirmed))+
  stat_smooth(method = loess,color='dark blue') +
  labs(x = "Time", y= "Confirmed Cases")+
   geom_vline(xintercept = mean_MW_start_date, color = "red", linetype = 2)+
  theme_bw() -> midwest_confirmed

ggplot(data =midwest, mapping = aes(x = date , y = daily_rate))+
  stat_smooth(method = loess,color='dark blue') +
   labs(x = "Time", y = "Daily Rate")+
   geom_vline(xintercept = mean_MW_start_date, color = "red", linetype = 2)+
  theme_bw() -> midwest_rate

ggplot(data =midwest, mapping = aes(x = date , y = second_rate))+
  stat_smooth(method = loess,color='dark blue') +
   labs(x = "Time", y = "Second Rate")+
   geom_vline(xintercept = mean_MW_start_date, color = "red", linetype = 2)+
  theme_bw() -> midwest_second_rate
```

South

```
#16states
maindata %>%
  filter(administrative_area_level_2==c("Delaware", "Maryland", "Virginia", "West Virginia", "Kentucky"
```

```
## Warning in administrative_area_level_2 == c("Delaware", "Maryland",
## "Virginia", : longer object length is not a multiple of shorter object length
```

```r
library(dplyr)
south %>%
    group_by(administrative_area_level_2) %>%
    arrange(date) %>%
    slice(1L) %>%
    select(administrative_area_level_2, date) -> south_first_case

# glimpse(south_first_case$date)
south_first_date <- as.Date(south_first_case$date)
# yday is the function to check how many days past after this year.
south_convert_days <- yday(south_first_date) - 1 # January 1 = day 0
mean_of_south <- mean(south_convert_days)
mean_S_start_date <- as.Date(mean_of_south, origin = "2020-01-01")
```

South first case day mean

```r
mean_S_start_date
```

```
## [1] "2020-03-16"
```

```r
ggplot(data =south, mapping = aes(x = date , y = confirmed))+
  stat_smooth(method = loess,color='dark green') +
  labs(x = "Time", y= "Confirmed Cases")+
  geom_vline(xintercept = mean_S_start_date, color = "red", linetype = 2)+
  theme_bw() -> south_confirmed

ggplot(data =south, mapping = aes(x = date , y = daily_rate))+
  stat_smooth(method = loess,color='dark green') +
 # geom_vline(xintercept = mdy("06/03/20"))+
   labs(x = "Time", y = "Daily Rate")+
    geom_vline(xintercept = mean_S_start_date, color = "red", linetype = 2)+
  theme_bw() -> south_rate



ggplot(data =south, mapping = aes(x = date , y = second_rate))+
  stat_smooth(method = loess,color='dark green') +
   labs(x = "Time", y = "Second Rate")+
    geom_vline(xintercept = mean_S_start_date, color = "red", linetype = 2)+
  theme_bw() -> south_second_rate
```

West

```r
#13states
maindata %>%
  filter(administrative_area_level_2 == c("Montana", "Idaho", "Wyoming", "Colorado", "New Mexico", "Ari
```

```
## Warning in administrative_area_level_2 == c("Montana", "Idaho", "Wyoming", :
## longer object length is not a multiple of shorter object length
```

```
library(dplyr)
west %>%
    group_by(administrative_area_level_2) %>%
    arrange(date) %>%
    slice(1L) %>%
    select(administrative_area_level_2, date) -> west_first_case



# glimpse(west_first_case$date)
south_first_date <- as.Date(west_first_case$date)
# yday is the function to check how many days past after this year.
west_convert_days <- yday(south_first_date) - 1 # January 1 = day 0
mean_of_west <- mean(west_convert_days)
mean_W_start_date <- as.Date(mean_of_west, origin = "2020-01-01")
```

West first case date mean

```
mean_W_start_date
```

```
## [1] "2020-03-12"
```

```
ggplot(data =west, mapping = aes(x = date , y = confirmed)) +
  stat_smooth(method = loess,color='orange') +
  labs(x = "Time", y= "Confirmed Cases")+
      geom_vline(xintercept = mean_W_start_date, color = "red", linetype = 2)+
  theme_bw() -> west_confirmed

ggplot(data =west, mapping = aes(x = date , y = daily_rate)) +
  stat_smooth(method = loess,color='orange') +
   labs(x = "Time", y = "Daily Rate")+
      geom_vline(xintercept = mean_W_start_date, color = "red", linetype = 2)+
  theme_bw() -> west_rate

ggplot(data =west, mapping = aes(x = date , y = second_rate)) +
  stat_smooth(method = loess,color='orange') +
   labs(x = "Time", y = "Second Rate")+
      geom_vline(xintercept = mean_W_start_date, color = "red", linetype = 2)+
  theme_bw() -> west_second_rate
```

Four Regions

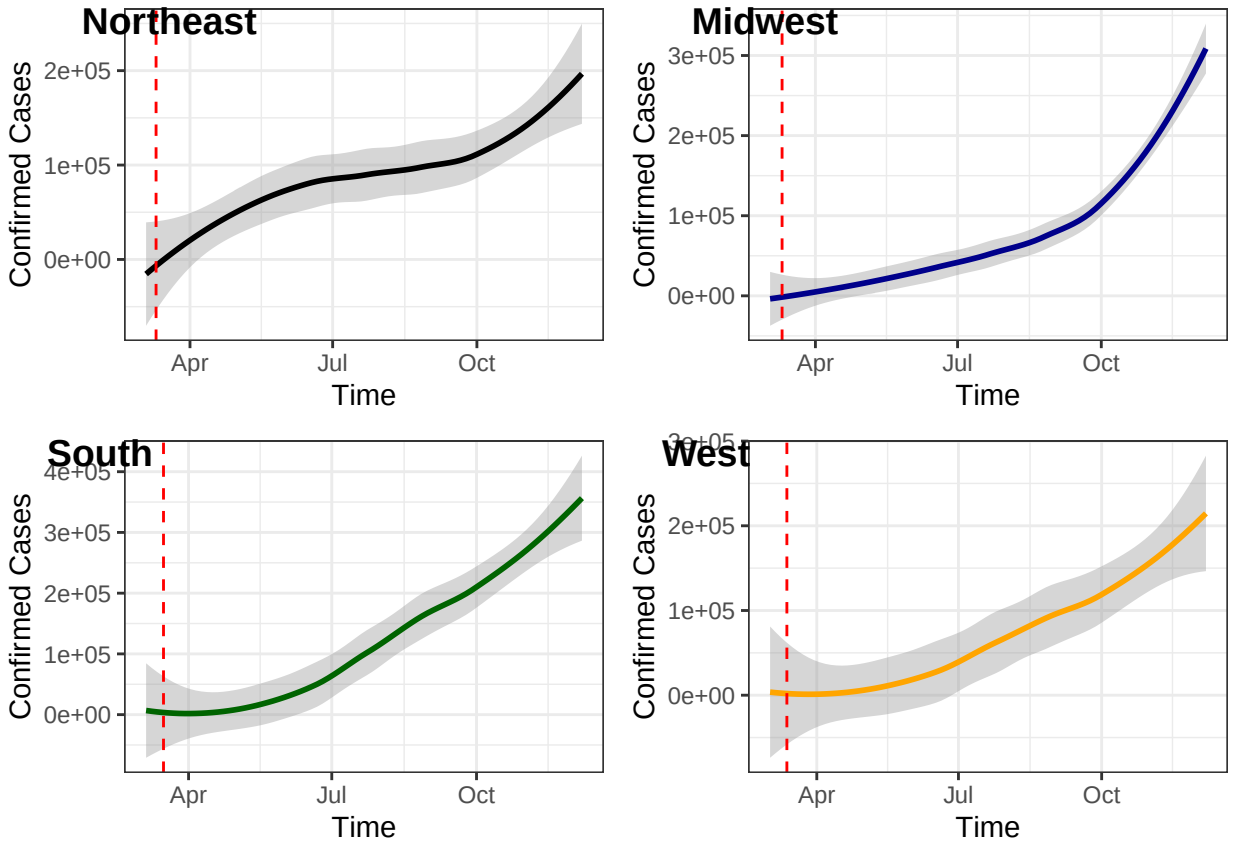
Confirmed vs Time

```
library(ggplot2)
library(ggpubr)
```

```
regions_confirmed <- ggarrange(northeast_confirmed,midwest_confirmed,south_confirmed, west_confirmed,
                    labels = c("Northeast", "Midwest", "South", "West"),
                    ncol = 2, nrow = 2)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

```
regions_confirmed
```

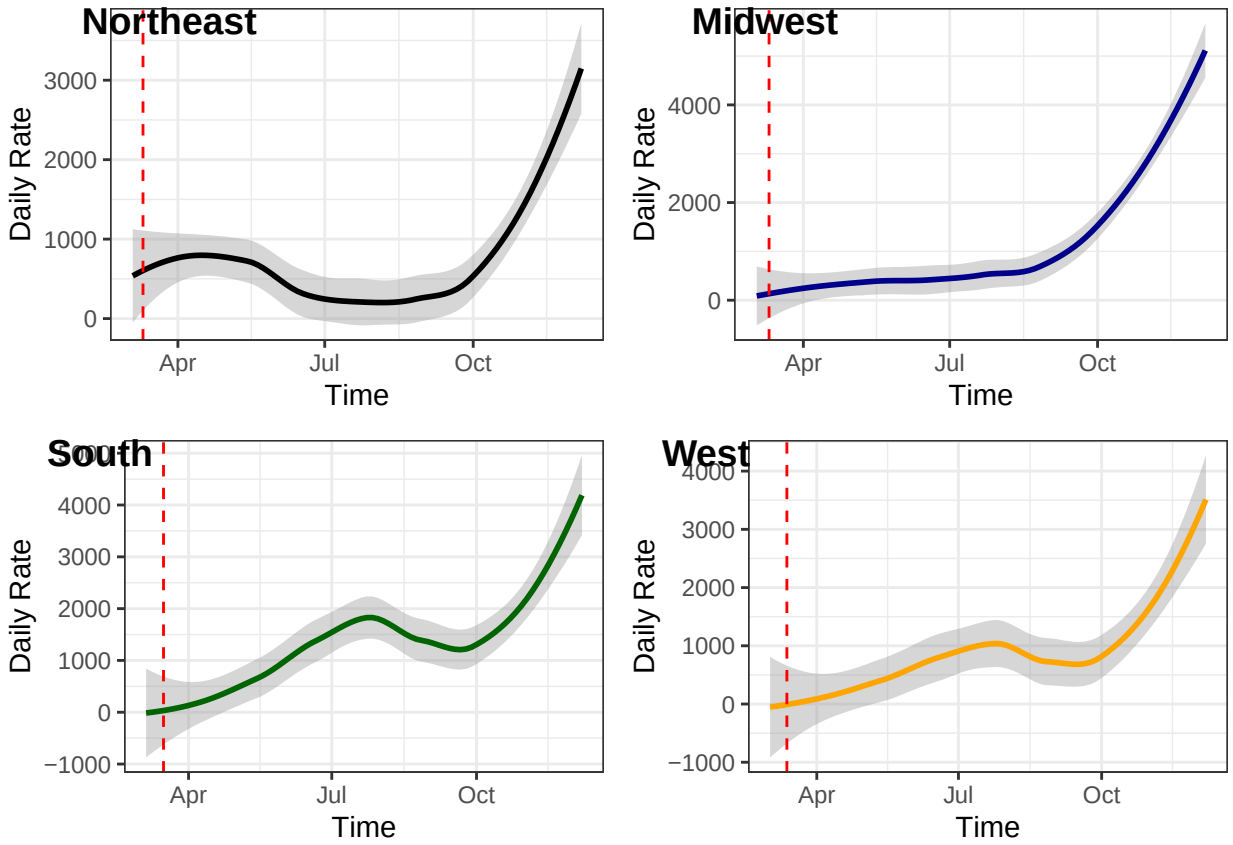


Four Regions # Daily_rate vs Time

```
regions_rate <- ggarrange(northeast_rate, midwest_rate,south_rate, west_rate,
                    labels = c("Northeast", "Midwest", "South", "West"),
                    ncol = 2, nrow = 2)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

```
regions_rate
```

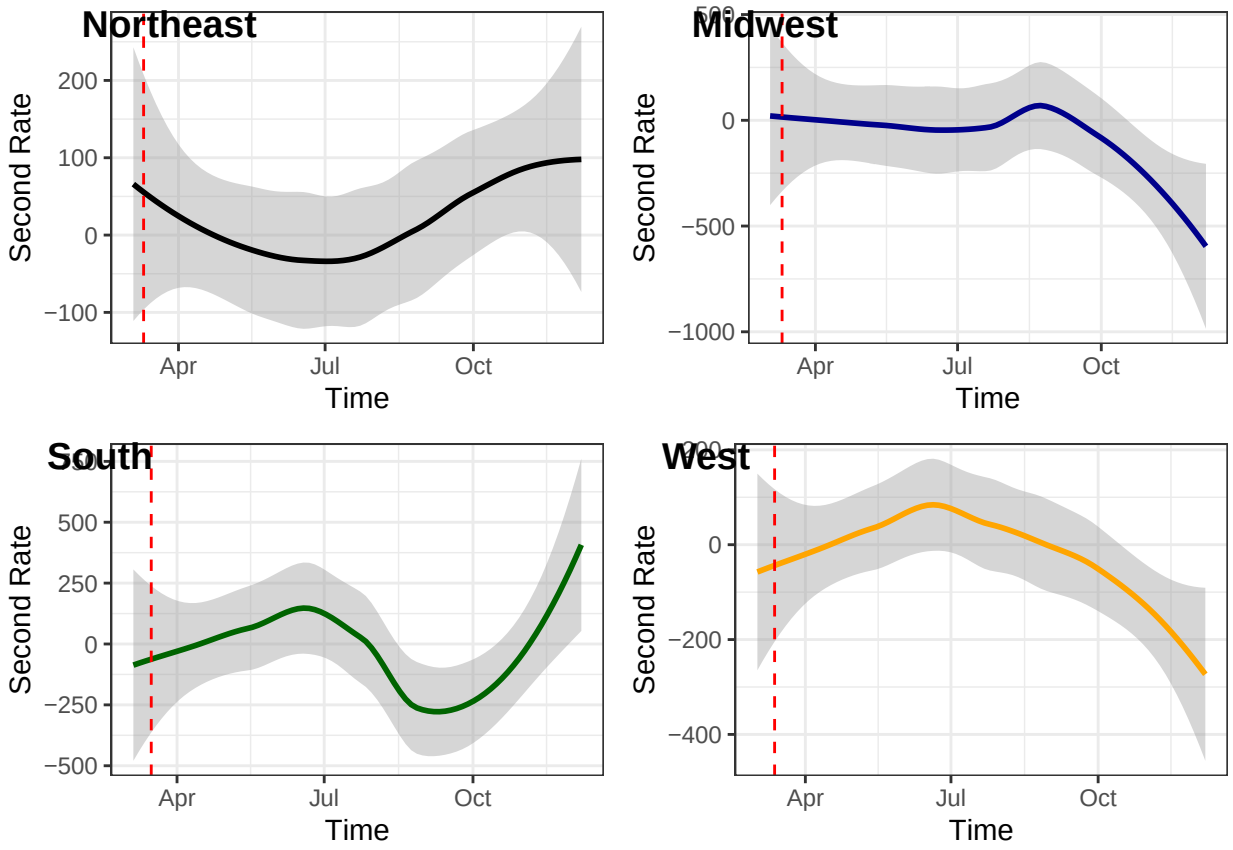


Four Regions # Second_rate vs Time

```
regions_second_rate <- ggarrange(northeast_second_rate,midwest_second_rate,south_second_rate, west_seco
                    labels = c("Northeast", "Midwest", "South", "West"),
                    ncol = 2, nrow = 2)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

```
regions_second_rate
```


Northeast
Second Rate
200
100
0
−100
Apr
Jul
Oct
Time
Midwest
Second Rate
0
−500
−1000
Apr
Jul
Oct
Time
South
Second Rate
500
250
0
−250
−500
Apr
Jul
Oct
Time
West
Second Rate
0
−200
−400
Apr
Jul
Oct
Time