

DTSC 5301 PROJECT

9/2/2021

Read in Data from GitHub Repository

```
covid_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/covid...  
  
## Rows: 30447 Columns: 24  
  
## -- Column specification -----  
## Delimiter: ","  
## chr (20): new_case_count, new_death_count, case_count, death_count, vaccine_...  
## dbl (4): year, month, day, statefips  
  
##  
## i Use 'spec()' to retrieve the full column specification for this data.  
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.  
  
move_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/move...  
  
## Rows: 28407 Columns: 11  
  
## -- Column specification -----  
## Delimiter: ","  
## chr (2): gps_parks, gps_transit_stations  
## dbl (9): year, month, day, statefips, gps_retail_and_recreation, gps_grocery...  
  
##  
## i Use 'spec()' to retrieve the full column specification for this data.  
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.  
  
affinity_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/affinity...  
  
## Rows: 29937 Columns: 28  
  
## -- Column specification -----  
## Delimiter: ","  
## chr (23): freq, spend_all, spend_aap, spend_acf, spend_aer, spend_apg, spend...  
## dbl (5): year, month, day, statefips, provisional  
  
##  
## i Use 'spec()' to retrieve the full column specification for this data.  
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

Join the datasets we're interested in into one dataset.

Here we join the datasets of interest based on a shared date and state of measurements.

```
df <- inner_join(covid_daily_df, move_daily_df, by = c("year", "month", "day", "statefips"))

df <- inner_join(df, affinity_daily_df, by = c("year", "month", "day", "statefips"))

# There's definitely a better way to do this, I just don't know what it is.

df <- df %>% mutate(
  spend_all = as.double(spend_all),
  gps_parks = as.double(gps_parks),
  new_case_count = as.double(new_case_count),
  new_death_count = as.double(new_death_count),
  case_count = as.double(case_count),
  death_count = as.double(death_count),
  vaccine_count = as.double(vaccine_count),
  fullvaccine_count = as.double(fullvaccine_count),
  new_vaccine_count = as.double(new_vaccine_count),
  new_fullvaccine_count = as.double(new_fullvaccine_count),
  new_test_count = as.double(new_test_count),
  test_count = as.double(test_count),
  new_case_rate = as.double(new_case_rate),
  case_rate = as.double(case_rate),
  new_death_rate = as.double(new_death_rate),
  death_rate = as.double(death_rate),
  new_test_rate = as.double(new_test_rate),
  test_rate = as.double(test_rate),
  new_vaccine_rate = as.double(new_vaccine_rate),
  vaccine_rate = as.double(vaccine_rate),
  new_fullvaccine_rate = as.double(new_fullvaccine_rate),
  fullvaccine_rate = as.double(fullvaccine_rate),
  gps_transit_stations = as.double(gps_transit_stations),
  spend_aap = as.double(spend_aap),
  spend_acf = as.double(spend_acf),
  spend_aer = as.double(spend_aer),
  spend_apg = as.double(spend_apg),
  spend_durables = as.double(spend_durables),
  spend_nondurables = as.double(spend_nondurables),
  spend_grf = as.double(spend_grf),
  spend_gen = as.double(spend_gen),
  spend_hic = as.double(spend_hic),
  spend_hcs = as.double(spend_hcs),
  spend_inpersonmisc = as.double(spend_inpersonmisc),
  spend_remoteservices = as.double(spend_remoteservices),
  spend_sgh = as.double(spend_sgh),
  spend_tws = as.double(spend_tws),
  spend_retail_w_grocery = as.double(spend_retail_w_grocery),
  spend_retail_no_grocery = as.double(spend_retail_no_grocery),
  spend_all_incmiddle = as.double(spend_all_incmiddle),
  spend_all_q1 = as.double(spend_all_q1),
  spend_all_q2 = as.double(spend_all_q2),
  spend_all_q3 = as.double(spend_all_q3),
```



```

## $ gps_grocery_and_pharmacy <dbl> -7.14e-03, 7.14e-03, 7.14e-03, -2.86e-03, 2.~  

## $ gps_parks <dbl> 0.05570, 0.06290, 0.05430, 0.02860, 0.10600,~  

## $ gps_transit_stations <dbl> 0.06000, 0.03140, 0.04000, 0.00429, 0.00714,~  

## $ gps_workplaces <dbl> 0.01290, 0.01570, 0.00286, 0.01570, 0.01140,~  

## $ gps_residential <dbl> 0.00857, 0.00286, 0.00429, 0.00143, 0.00143,~  

## $ gps_away_from_home <dbl> -0.007980, -0.000578, -0.007510, -0.003010, ~  

## $ freq <chr> "d", "d", "d", "d", "d", "d", "d", "d", "d", ~  

## $ spend_all <dbl> -0.022500, -0.004720, 0.040300, 0.002310, 0.~  

## $ spend_aap <dbl> -0.12800, 0.01110, 0.00411, -0.06720, -0.027~  

## $ spend_acf <dbl> -0.02120, -0.05660, -0.00846, 0.03590, 0.004~  

## $ spend_aer <dbl> -0.11500, 0.10900, -0.00735, -0.14200, 0.021~  

## $ spend_apg <dbl> -0.07890, 0.04630, 0.02820, -0.03640, -0.004~  

## $ spend_durables <dbl> -0.03380, -0.06760, 0.04040, -0.00067, 0.025~  

## $ spend_nondurables <dbl> -0.046700, 0.002350, 0.026000, -0.069200, -0~  

## $ spend_grf <dbl> -0.02350, 0.03680, 0.00547, -0.04650, 0.0026~  

## $ spend_gen <dbl> -0.01510, 0.12300, 0.04620, -0.01880, 0.0078~  

## $ spend_hic <dbl> -0.05860, -0.34600, 0.12700, -0.00115, 0.087~  

## $ spend_hcs <dbl> -0.07260, -0.03000, 0.01640, 0.05210, -0.002~  

## $ spend_inpersonmisc <dbl> 0.00572, 0.23800, 0.09370, 0.06510, 0.01110,~  

## $ spend_remoteservices <dbl> 0.02430, -0.00888, 0.05380, -0.01580, 0.0382~  

## $ spend_sgh <dbl> -0.05870, 0.28000, -0.10200, -0.10700, -0.00~  

## $ spend_tws <dbl> -0.11000, 0.00144, -0.02870, -0.01690, -0.02~  

## $ spend_retail_w_grocery <dbl> -0.038800, -0.051200, 0.035900, -0.052400, 0~  

## $ spend_retail_no_grocery <dbl> -0.04580, -0.09500, 0.04900, -0.05710, 0.013~  

## $ spend_all_incmiddle <dbl> -0.031200, 0.138000, 0.038400, -0.008760, 0.~  

## $ spend_all_q1 <dbl> -0.02020, NA, 0.06270, 0.02320, 0.00381, 0.0~  

## $ spend_all_q2 <dbl> -0.07110, -0.00110, 0.06320, -0.01810, 0.027~  

## $ spend_all_q3 <dbl> 0.032600, 0.188000, 0.012800, 0.009960, 0.01~  

## $ spend_all_q4 <dbl> 9.87e-03, -6.99e-02, 3.11e-02, 1.93e-01, 2.3~  

## $ provisional <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~

```

Combine “month”, “day”, and “year” columns into a “date” column

```
# https://tidyverse.org/reference/unite.html  
df <- df %>% unite("date", day:month:year, remove = FALSE, sep = "-")
```

```
## Warning in x:y: numerical expression has 2 elements: only the first used
```

```
# https://lubridate.tidyverse.org/reference/ymd.html  
df$date <- dmy(df$date)
```

Plotting spending over time for all states and categories

The dates for the stimulus checks were approximated from this article.

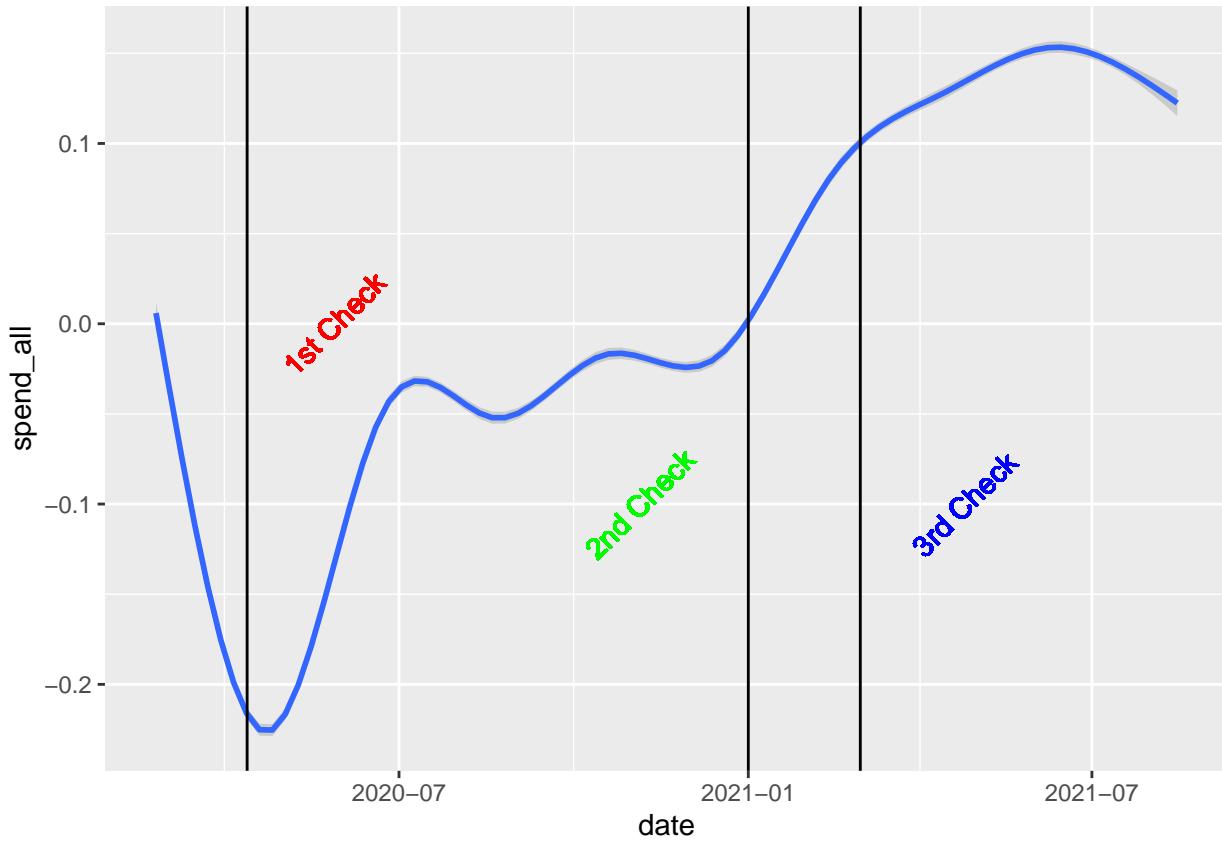
```
# https://stackoverflow.com/questions/38815996/r-adding-geom-vline-labels-to-geom-histogram-labels  
ggplot(df, aes(x = date, y = spend_all)) +  
  geom_smooth() +  
  geom_vline(xintercept = as.Date("2020-04-12")) +
```

```

geom_vline(xintercept = as.Date("2021-01-01")) +
geom_vline(xintercept = as.Date("2021-03-01")) +
geom_text(aes(x = as.Date("2020-05-28"), label = "1st Check"), color = "red", angle = 45, y = 0) +
geom_text(aes(x = as.Date("2020-11-05"), label = "2nd Check"), color = "green", angle = 45, y = -.1) +
geom_text(aes(x = as.Date("2021-04-25"), label = "3rd Check"), color = "blue", angle = 45, y = -.1)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

```



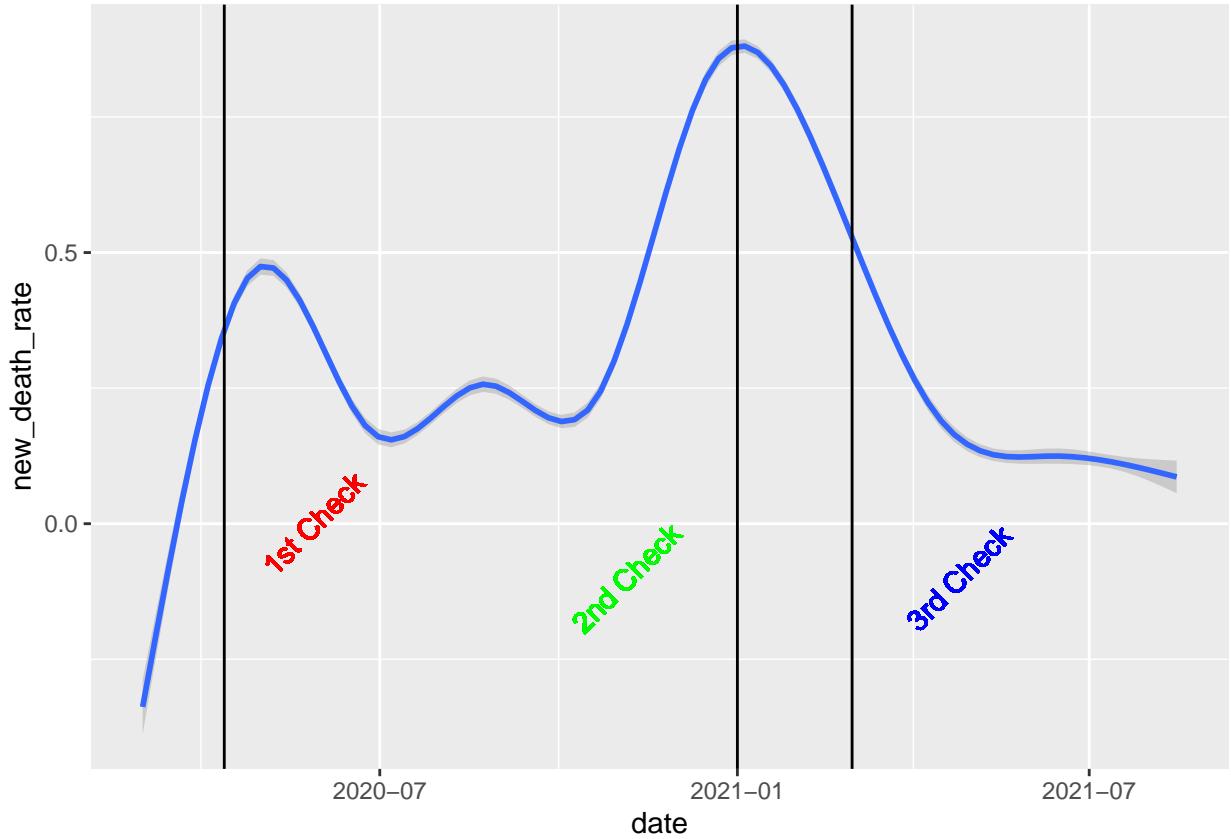
```

ggplot(df, aes(x = date, y = new_death_rate)) +
  geom_smooth() +
  geom_vline(xintercept = as.Date("2020-04-12")) +
  geom_vline(xintercept = as.Date("2021-01-01")) +
  geom_vline(xintercept = as.Date("2021-03-01")) +
  geom_text(aes(x = as.Date("2020-05-28"), label = "1st Check"), color = "red", angle = 45, y = 0) +
  geom_text(aes(x = as.Date("2020-11-05"), label = "2nd Check"), color = "green", angle = 45, y = -.1) +
  geom_text(aes(x = as.Date("2021-04-25"), label = "3rd Check"), color = "blue", angle = 45, y = -.1)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 1301 rows containing non-finite values (stat_smooth).

```



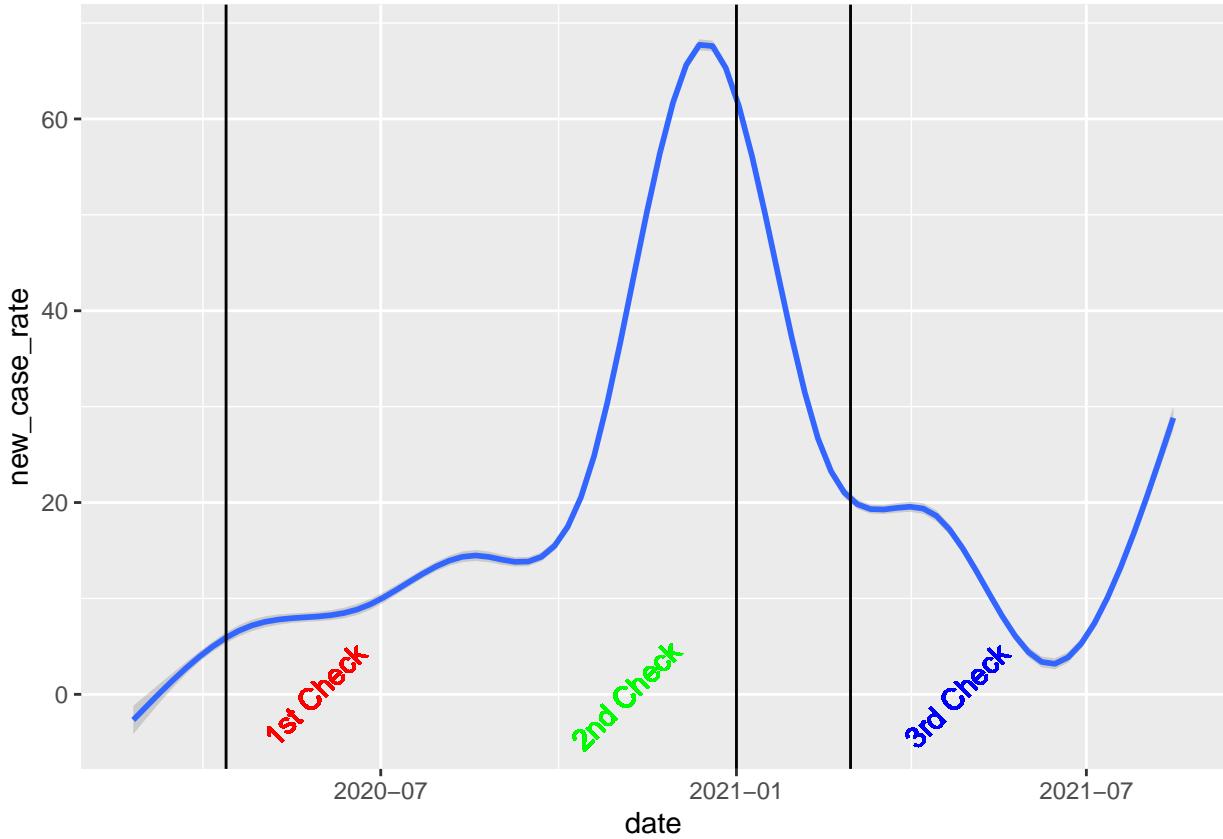
```

ggplot(df, aes(x = date, y = new_death_rate)) +
  geom_smooth() +
  geom_vline(xintercept = as.Date("2020-04-12")) +
  geom_vline(xintercept = as.Date("2021-01-01")) +
  geom_vline(xintercept = as.Date("2021-03-01")) +
  geom_text(aes(x = as.Date("2020-05-28"), label = "1st Check"), color = "red", angle = 45, y = 0) +
  geom_text(aes(x = as.Date("2020-11-05"), label = "2nd Check"), color = "green", angle = 45, y = -.1) +
  geom_text(aes(x = as.Date("2021-04-25"), label = "3rd Check"), color = "blue", angle = 45, y = -.1)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 564 rows containing non-finite values (stat_smooth).

```



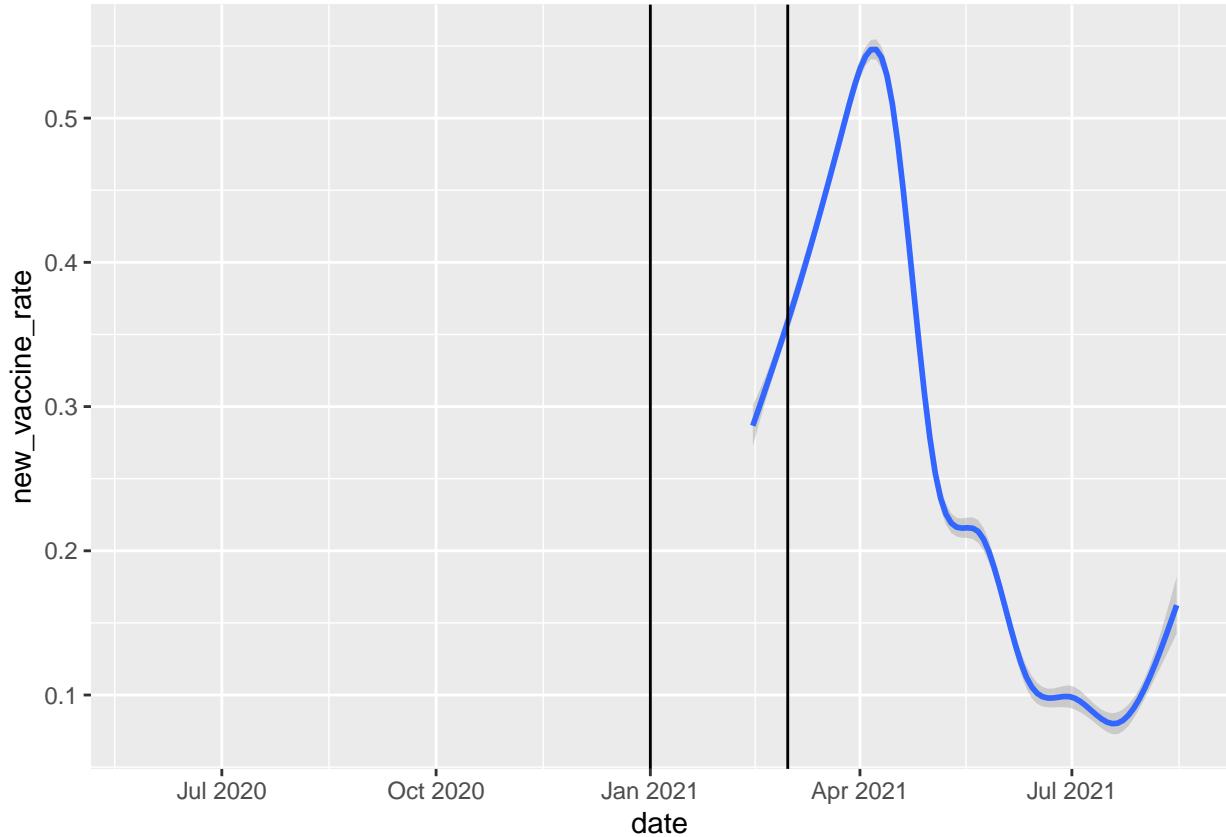
```

ggplot(df, aes(x = date, y = new_vaccine_rate)) +
  geom_smooth() +
  geom_vline(xintercept = as.Date("2020-04-12")) +
  geom_vline(xintercept = as.Date("2021-01-01")) +
  geom_vline(xintercept = as.Date("2021-03-01")) +
  geom_text(aes(x = as.Date("2020-05-28"), label = "1st Check"), color = "red", angle = 45, y = 0) +
  geom_text(aes(x = as.Date("2020-11-05"), label = "2nd Check"), color = "green", angle = 45, y = -.1) +
  geom_text(aes(x = as.Date("2021-04-25"), label = "3rd Check"), color = "blue", angle = 45, y = -.1)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

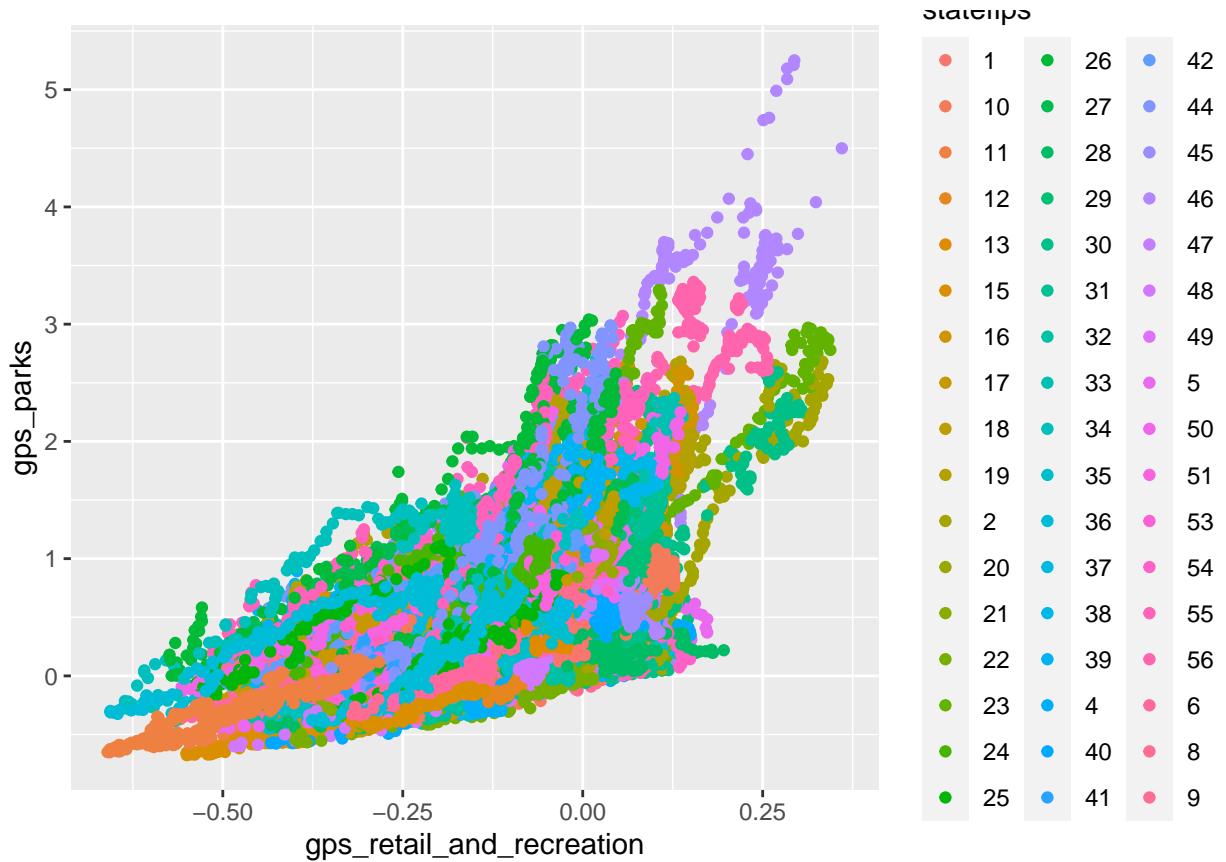
## Warning: Removed 18162 rows containing non-finite values (stat_smooth).

```



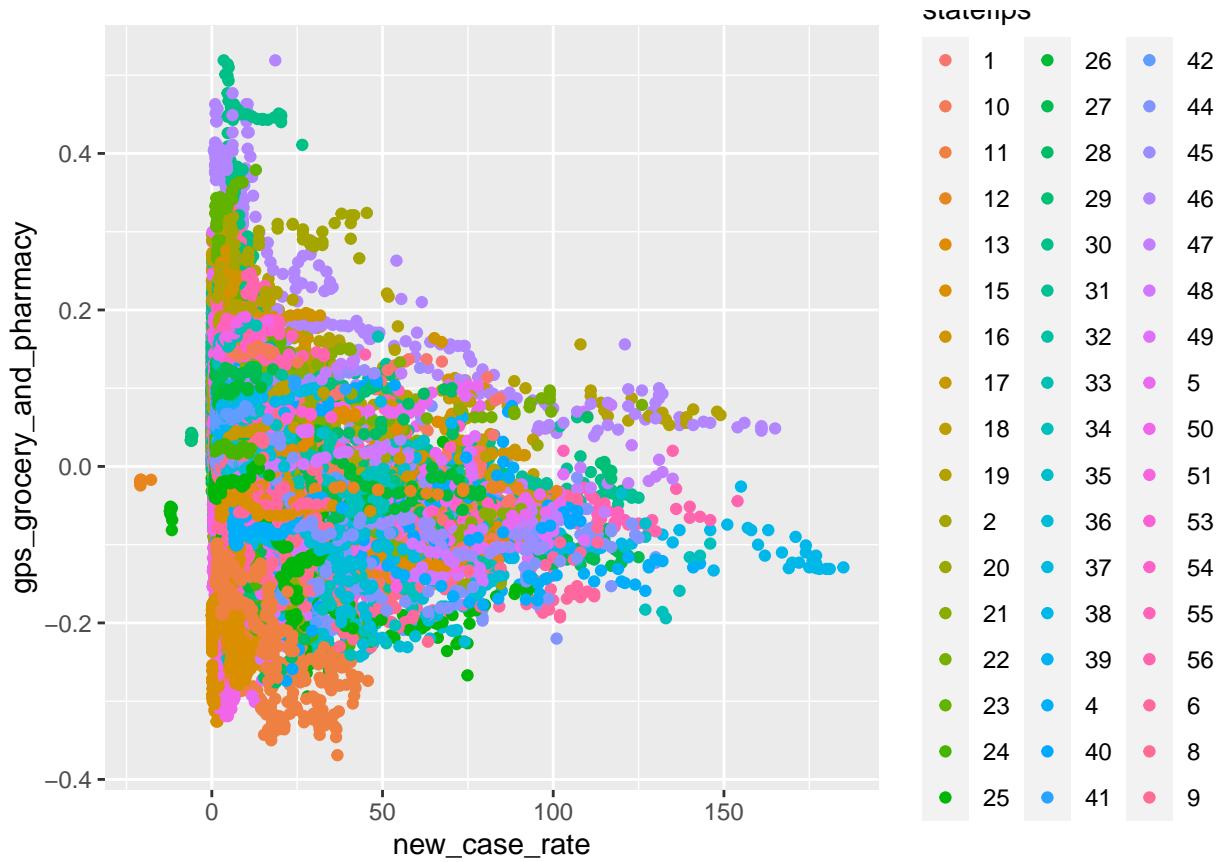
```
ggplot(df, aes(x = gps_retail_and_recreation, y = gps_parks)) +  
  geom_point(aes(color=statefips))
```

```
## Warning: Removed 440 rows containing missing values (geom_point).
```



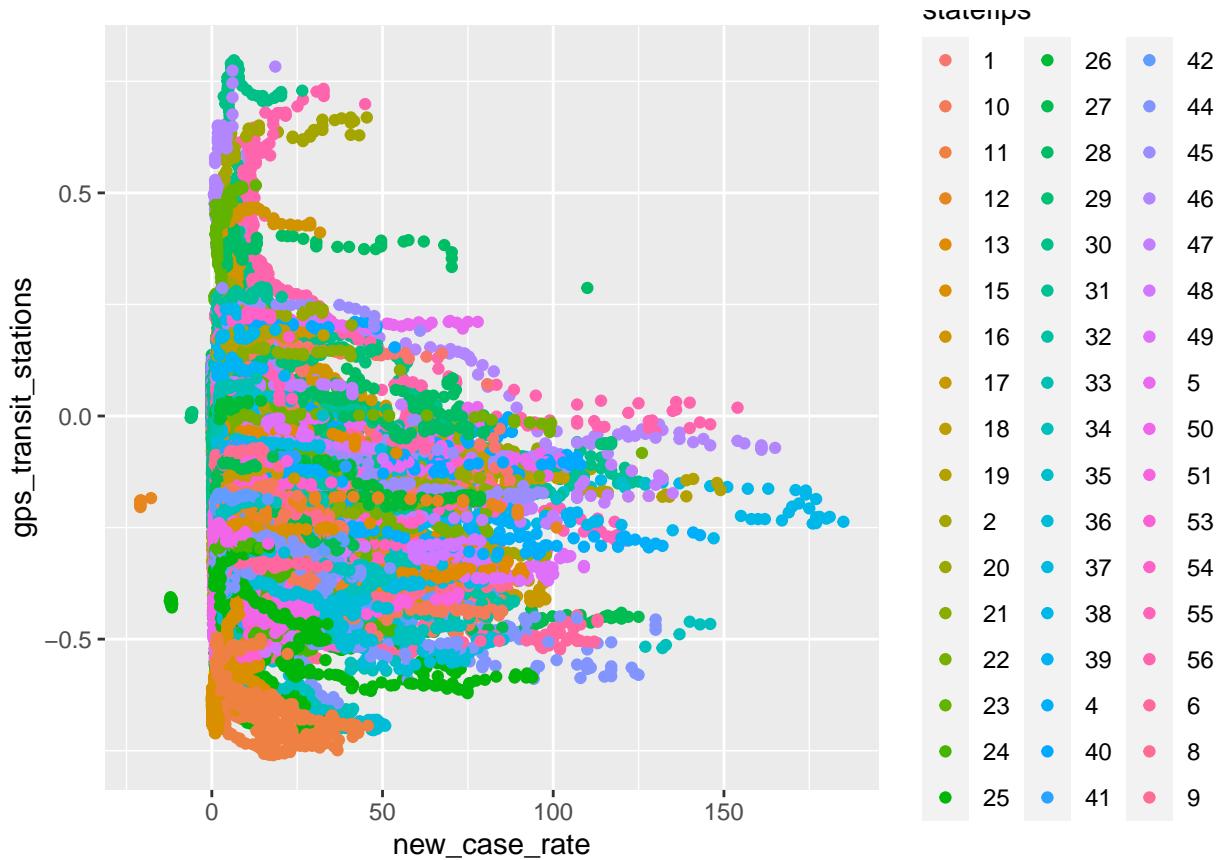
```
ggplot(df, aes(x = new_case_rate, y = gps_grocery_and_pharmacy)) +
  geom_point(aes(color=statefips))
```

```
## Warning: Removed 564 rows containing missing values (geom_point).
```



```
ggplot(df, aes(x = new_case_rate, y = gps_transit_stations)) +
  geom_point(aes(color=statefips))
```

```
## Warning: Removed 717 rows containing missing values (geom_point).
```



```
ggplot(df, aes(x = new_case_rate, y = gps_residential)) +
  geom_point(aes(color=statefips))
```

```
## Warning: Removed 564 rows containing missing values (geom_point).
```

