DTSC 5301 PROJECT

9/2/2021

Read in Data from GitHub Repository

```
covid_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/
```

```
## 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/da
```

```
## Rows: 28611 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/ma
```

```
## Rows: 31008 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.
```

```
job_listings_weekly_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracke
```

```
## Rows: 4488 Columns: 17

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (17): year, month, day_endofweek, statefips, bg_posts, bg_posts_ss30, bg...

##
## 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.
```

```
employment_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/m
```

```
## Rows: 27081 Columns: 16

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): emp, emp_incq1, emp_incq2, emp_incq3, emp_incq4, emp_incmiddle, em...
## dbl  (6): year, month, day, statefips, emp_incbelowmed, emp_ss70

##
## 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.
```

```
ui_claims_weekly_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/m
```

```
## Rows: 4437 Columns: 18

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): contclaims_count_peuc, contclaims_count_pua, contclaims_count_comb...
## dbl (12): year, month, day_endofweek, statefips, initclaims_count_regular, c...

##
## 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.
```

```
state_id <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/Ge
```

```
## Rows: 51 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): statename, stateabbrev
## dbl (2): statefips, state_pop2019

##
## 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.
```

```
#stimulus <- read_csv("")
```

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 <- left_join(affinity_daily_df, move_daily_df, by = c("year", "month", "day", "statefips"))

df <- left_join(df, covid_daily_df, by = c("year", "month", "day", "statefips"))

df <- left_join(df, employment_daily_df, by = c("year", "month", "day", "statefips"))
```

```
df <- full_join(df, job_listings_weekly_df, by = c("year", "month", "day" = "day_endofweek", "statefips

df <- full_join(df, ui_claims_weekly_df, by = c("year", "month", "day" = "day_endofweek", "statefips"))

df <- left_join(df, state_id, by = c("statefips"))
```

Combine "month", "day", and "year" columns into a "date" column

```
# https://tidyr.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)

df <- df %>% mutate(week = week(date))
```

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

df <- df %>%
  select(date, year, month, day, week, statename, stateabbrev, state_pop2019, initclaims_rate_regular, 
  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),
    gps_transit_stations = as.double(gps_transit_stations),
    emp = as.double(emp),
    contclaims_rate_combined = as.double(contclaims_rate_combined)
  )
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
glimpse(df)
```

```
## Rows: 31,110
## Columns: 24
## $ date                       <date> 2020-01-01, 2020-01-01, 2020-01-01, 2020-01~
## $ year                       <dbl> 2020, 2020, 2020, 2020, 2020, 2020, 2020, 20~
## $ month                      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ day                        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ week                       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ statename                  <chr> "Alabama", "Alaska", "Arizona", "Arkansas", ~
## $ stateabbrev                <chr> "AL", "AK", "AZ", "AR", "CA", "CO", "CT", "D~
## $ state_pop2019              <dbl> 4903185, 731545, 7278717, 3017804, 39512223,~
## $ initclaims_rate_regular    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ contclaims_rate_combined   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ bg_posts                   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ emp                        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ spend_all                  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ gps_retail_and_recreation  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ gps_grocery_and_pharmacy   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ gps_parks                  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ gps_transit_stations       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ gps_workplaces             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ gps_residential            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ gps_away_from_home         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ new_case_count             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ new_death_count            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ case_count                 <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ death_count                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

```
# https://stackoverflow.com/questions/45576805/how-to-replace-all-na-in-a-dataframe-using-tidyrreplace-

#length(df$date)

#colSums(is.na(df))

df <- df %>% replace(is.na(.), 0)
df <- df %>% filter(spend_all != 0)
```

```
df_weekly <- df %>%
  group_by(year, week, stateabbrev) %>%
  summarize(spend_all = mean(spend_all), contclaims_rate_combined = mean(contclaims_rate_combined), bg_
```

```
## `summarise()` has grouped output by 'year', 'week'. You can override using the `.groups` argument.
```

```
df_weekly <- df_weekly %>% mutate(lagged_spend_one = spend_all - lag(spend_all, order_by = date),
                                  lagged_spend_two = spend_all - lag(spend_all, n = 2, order_by = date)
                                  lagged_spend_three = spend_all - lag(spend_all, n = 3, order_by = dat

df_weekly <- left_join(df_weekly, state_id, by = c("stateabbrev"))
```

```
df_weekly <- df_weekly %>% mutate(first_check = (if (date < ymd("2020-04-15") | date >= ymd("2020-07-15
                                                 else 1),
                                  second_check = (if (date < ymd("2021-01-04") | date >= ymd("2021-03-1
                                                  else 1),
                                  third_check = (if (date < ymd("2021-03-18") | date >= ymd("2021-06-18
                                                 else 1))
```

```
sum(sapply(df_weekly, is.infinite))
```

```
## [1] 0
```

```
df_weekly <- df_weekly %>% filter(!is.infinite(third_check))
sum(sapply(df_weekly, is.infinite))
```

```
## [1] 0
```

```
lm <- lm(spend_all ~ 0 + gps_retail_and_recreation + emp + first_check +
           second_check + third_check, df_weekly)
summary(lm)
```

```
##
## Call:
## lm(formula = spend_all ~ 0 + gps_retail_and_recreation + emp +
##     first_check + second_check + third_check, data = df_weekly)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33879 -0.03691  0.01039  0.06654  0.53603
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## gps_retail_and_recreation  3.646e-01  1.410e-02  25.854   <2e-16 ***
## emp                       -7.603e-05  3.102e-02  -0.002    0.998
## first_check               -4.370e-02  4.703e-03  -9.291   <2e-16 ***
## second_check               1.410e-01  4.910e-03  28.714   <2e-16 ***
## third_check                1.546e-01  3.691e-03  41.881   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.09816 on 4483 degrees of freedom
## Multiple R-squared:  0.4737, Adjusted R-squared:  0.4731
## F-statistic: 806.8 on 5 and 4483 DF,  p-value: < 2.2e-16
```

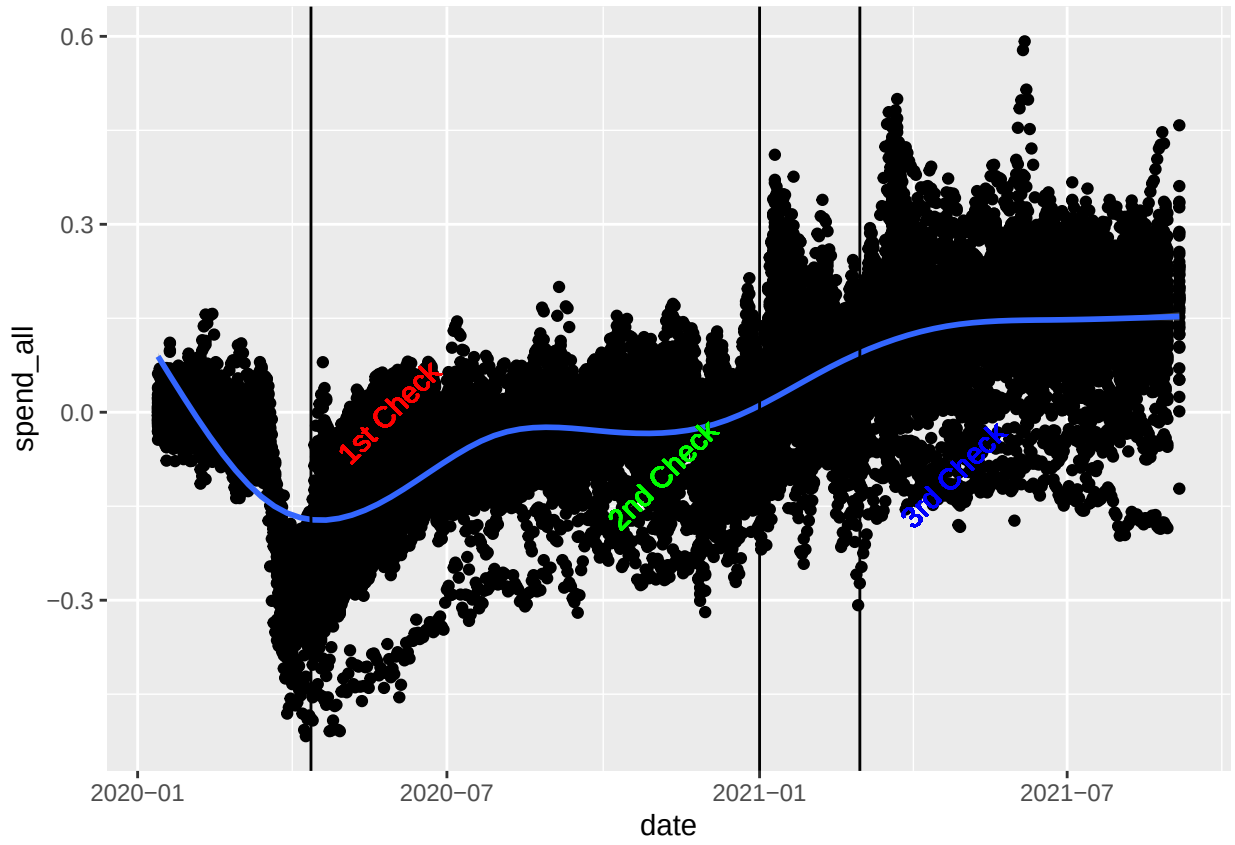
```
view(df_weekly)
```

Plotting spending over time for all states and categories

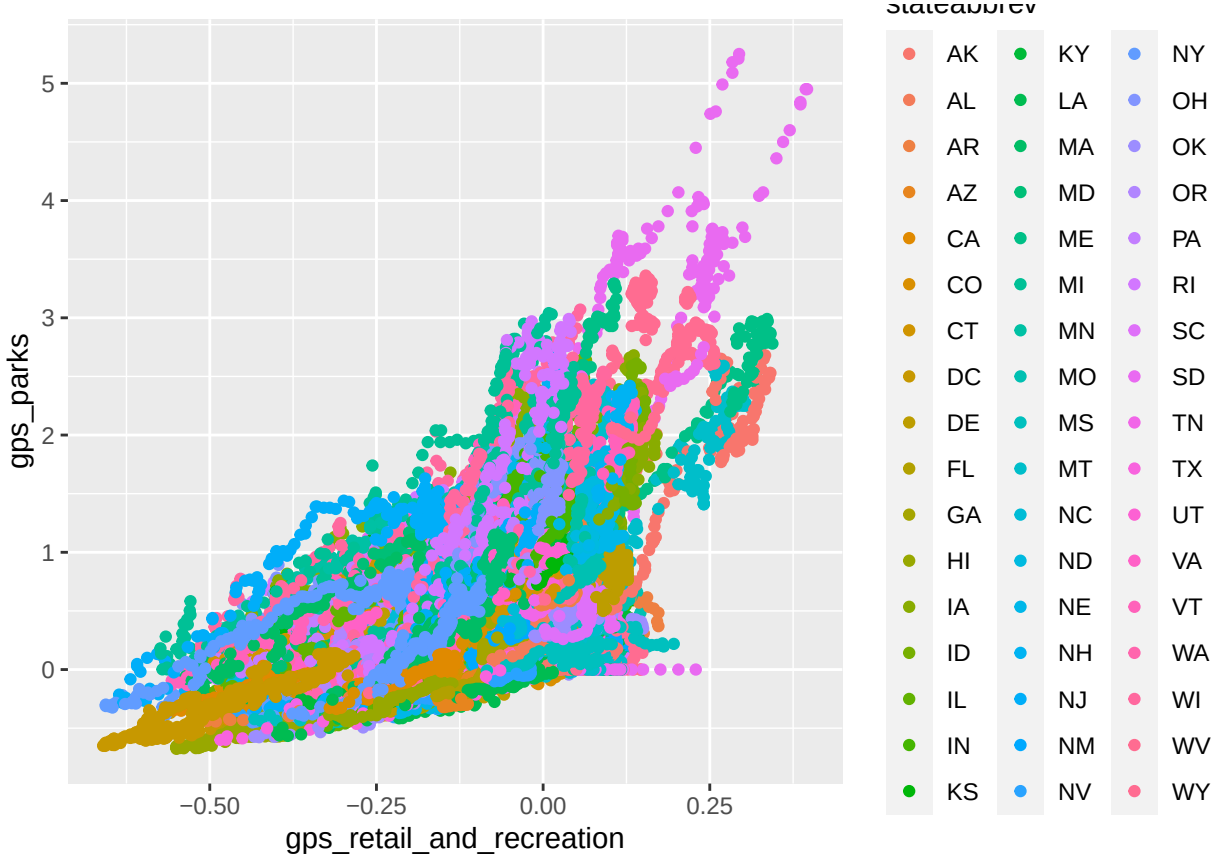
The dates for the stimulus checks were approximated from this article.

```
# https://stackoverflow.com/questions/38815996/r-adding-geom-vline-labels-to-geom-histogram-labels

ggplot(df, aes(x = date, y = spend_all)) +
  geom_point() +
  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")'
```

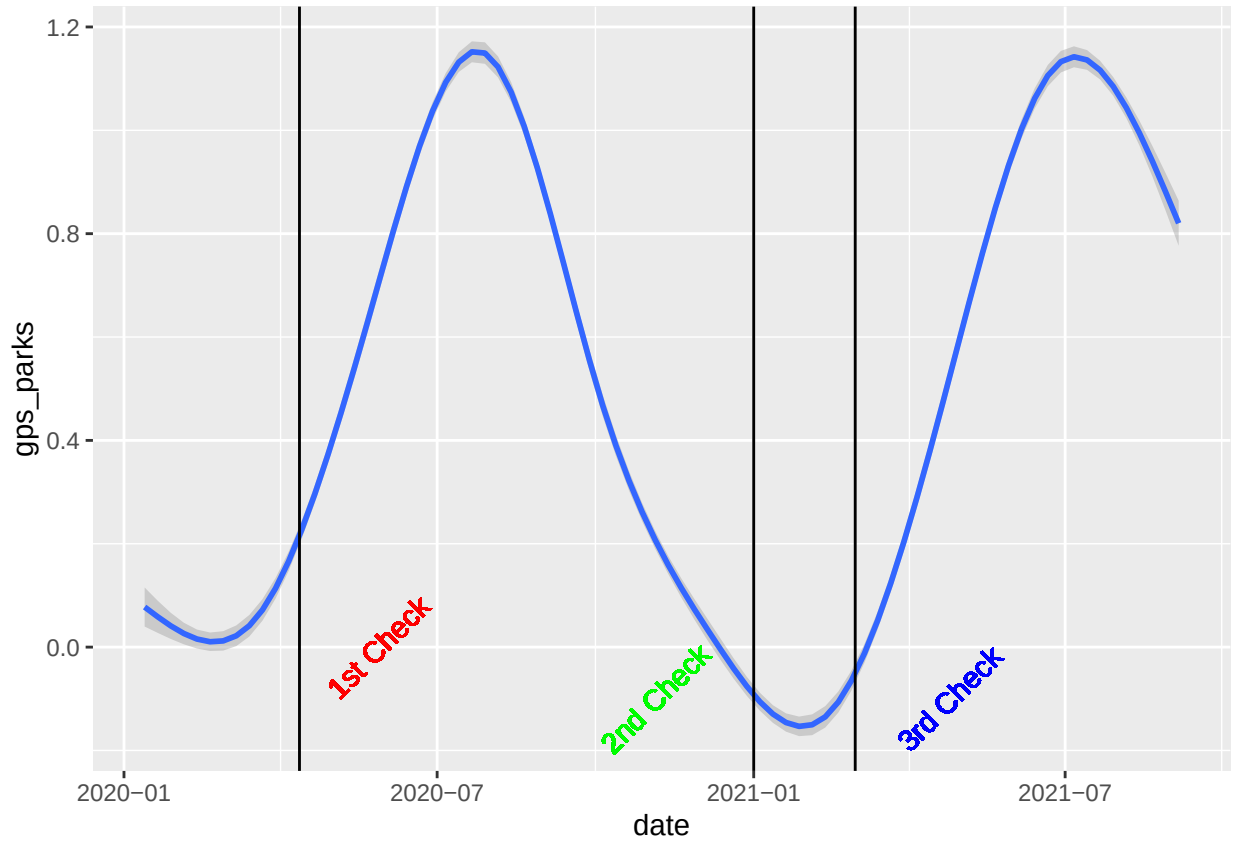


```
ggplot(df, aes(x = gps_retail_and_recreation, y = gps_parks)) +
  geom_point(aes(color = stateabbrev))
```

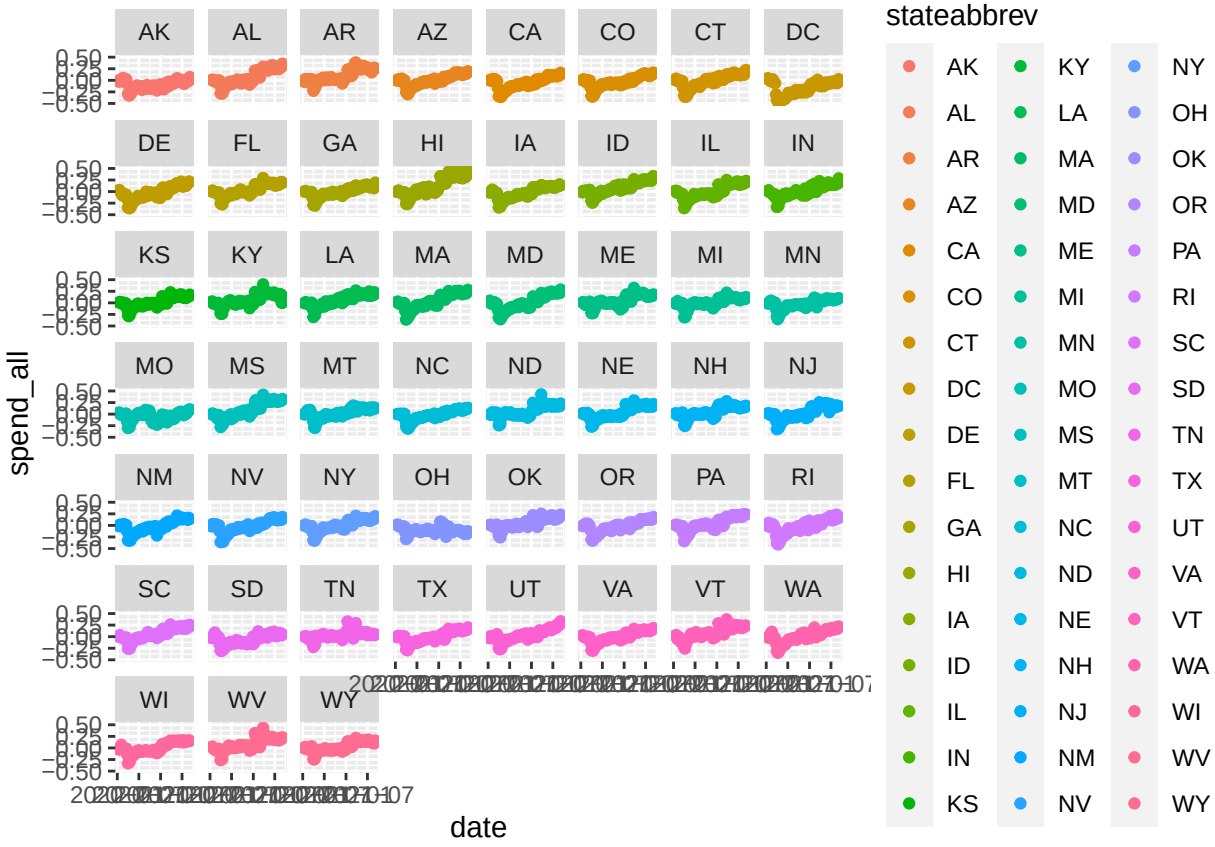



```
# https://stackoverflow.com/questions/38815996/r-adding-geom-vline-labels-to-geom-histogram-labels

ggplot(df, aes(x = date, y = gps_parks)) +
  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")'
```




```
# https://stackoverflow.com/questions/38815996/r-adding-geom-vline-labels-to-geom-histogram-labels

ggplot(df_weekly, aes(x = date, y = spend_all, color = stateabbrev)) +
  geom_point() +
  facet_wrap(. ~ stateabbrev)
```


AK
AL
AR
AZ
CA
CO
CT
DC
DE
FL
GA
HI
IA
ID
IL
IN
KS
KY
LA
MA
MD
ME
MI
MN
MO
MS
MT
NC
ND
NE
NH
NJ
NM
NV
NY
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
VA
VT
WA
WI
WV
WY
spend_all
date
stateabbrev
AK
AL
AR
AZ
CA
CO
CT
DC
DE
FL
GA
HI
IA
ID
IL
IN
KS
KY
LA
MA
MD
ME
MI
MN
MO
MS
MT
NC
ND
NE
NH
NJ
NM
NV
NY
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
VA
VT
WA
WI
WV
WY