# Stimulus Checks and Consumer Spending

Aaron Davis, Navya Sonti, Rujula Nadipi, Swapnil Sethi, and Ujas Shah

9/2/2021

## Question of Interest

We want to know whether or not the stimulus checks sent out by the US government have had a positive impact on the economy (using consumer spending as a proxy for how healthy the economy is).

## Why Should YOU Care How Healthy the US Economy Is?

Why should we care about consumer spending? It seems like a big picture idea that won't really effect any of us specifically, right? Wrong. When the economy is healthy, there are more jobs available. Those jobs also pay more. As graduate students, we all want to get good paying jobs as Data Scientists, and that will be more likely to happen more quickly if the economy is healthy.

This analysis will help us understand how stimulus checks effect the US economy, and therefore, indirectly, the analysis will also help us understand whether or not stimulus checks will help us get good-paying jobs quickly after graduation.

There is also a long-standing debate between the liberal and conservative political ideologies in the US over whether or not the increase in consumer spending caused by stimulus checks is actually worth the debt incurred by the government when sending out the checks. While this analysis doesn't cover that, this would be an interesting topic for future research using this dataset.

## Data Source

All of our data was aggregated by Opportunity Insights at https://github.com/OpportunityInsights/EconomicTracker. In this analysis, we use spending data provided by Affinity Solutions, job postings data from Burning Glass Technologies, COVID data from the CDC, GPS mobility reports from Google, unemployment claims from the Department of Labor, and employment levels from Paychex, Intuit, Earnin and Kronos.

**Primary Reference:**

"The Economic Impacts of COVID-19: Evidence from a New Public Database Built Using Private Sector Data", by Raj Chetty, John Friedman, Nathaniel Hendren, Michael Stepner, and the Opportunity Insights Team. November 2020. Available at: https://opportunityinsights.org/wp-content/uploads/2020/05/tracker_paper.pdf

### Read in Data from GitHub Repository

```r
options(width=30)

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

move_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/da

affinity_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/mai

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

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

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

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

# 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. We are joining weekly datasets (job listings and unemployment insurance claims) with the daily datasets. This will leave a bunch of **NA** values. We'll come back to fix that later.

```r
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

```r
# 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
```

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

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

### Data Selection and Cleaning

In this code chunk, we select the columns that we're interested in from the dataframe that we joined a couple of steps ago. We could use all of the data in the dataframe, but we choose not to, since not all of the features will be helpful in answering the question of whether or not the stimulus checks have boosted the economy.

```
df <- df %>%
  select(date, year, month, day, week, statename, stateabbrev, state_pop2019, initclaims_rate_regular, c
  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)
  )


glimpse(df)
```

```
## Rows: 31,161
## Columns: 24
## $ date                       <date> ~
## $ year                       <dbl> ~
## $ month                      <dbl> ~
## $ day                        <dbl> ~
## $ week                       <dbl> ~
## $ statename                  <chr> ~
## $ stateabbrev                <chr> ~
## $ state_pop2019              <dbl> ~
## $ initclaims_rate_regular    <dbl> ~
## $ contclaims_rate_combined   <dbl> ~
## $ bg_posts                   <dbl> ~
## $ emp                        <dbl> ~
## $ spend_all                  <dbl> ~
## $ gps_retail_and_recreation  <dbl> ~
## $ gps_grocery_and_pharmacy   <dbl> ~
## $ gps_parks                  <dbl> ~
## $ gps_transit_stations       <dbl> ~
## $ gps_workplaces             <dbl> ~
## $ gps_residential            <dbl> ~
## $ gps_away_from_home         <dbl> ~
## $ new_case_count             <dbl> ~
## $ new_death_count            <dbl> ~
## $ case_count                 <dbl> ~
## $ death_count                <dbl> ~
```

To fix the **NA** values found in our dataset, we replace them all with **0**. We could have handled them many other ways, like replacing them with the column mean, median, mode, or by training a regression model to fill them based on the rows that were not missing those values. For this dataset, though, we found that

**NA**s are frequently used when the there was no interesting data to report (in other words, the value for the feature was zero). This can be seen particularly in columns like new_death_count and new_case_count from the CDC COVID dataset.

We also drop rows that have 0 spending data because this value is not realistic. Also, we need the spend_all column to be clean because we will be using it for plotting and training a regression model later on.

```
# 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)
```

### Combing Weekly and Daily Data

In this code chunk, we aggregate daily data into weekly data. For most columns, the appropriate aggregate function is **mean()**, with the exception of the *new_case_count*, *new_death_count*, *case_count*, *death_count*, and *date columns*.

We also add in additional data about the states, at this point (like state name, abbreviation, and population).

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

### Adding Stimulus Check Data

Here we add in the data for the COVID stimulus checks. We do this by creating a feature encoding for each check, where the value for that check is **0** before the check is sent out, and **1** after the check is sent out. This process created three new features, which we creatively named *first_check*, *second_check*, and *third_check*

```
df_weekly <- df_weekly %>% mutate(
  first_check = (if (date < ymd("2020-04-15")) {
    0
  } else {
    1
  }),
  second_check = (if (date < ymd("2021-01-04")) {
    0
  } else {
    1
  }),
  third_check = (if (date < ymd("2021-03-18")) {
    0
  } else {
```

```
    1
  })
)
```

### Double Check that Data is Clean

We already took care of **NA** values a few steps ago. Now we need to ensure that there are no infinite values, as well.

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

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

As you can see from the results above, the number of infinite values in our dataset currently is 0.

### Training a Linear Model on Our Data

In this code chunk, we fit a linear model to the *gps_retail_and_recreation*, *emp*, *first_check*, *second_check*, and *third_check* features, with the goal of predicting the *spend_all* variable. The working assumption here is that *spend_all* is a dependent variable, and the others are independent variables.

The reason we chose these five input variables is that *first_check*, *second_check*, and *third_check* are directly relevant to our analysis. We are trying to identify if these features had a positive or negative effect on the economy. These two features, *gps_retail_and_recreation* and *emp*, were added to help account for change in spending that could not necessarily be accounted for by the *first_check*, *second_check*, and *third_check*. We chose not to include any other input features because we believe that *gps_retail_and_recreation* and *emp* account for much of the possible variance without over-complicating our linear model.

```
mod <- lm(spend_all ~ gps_retail_and_recreation + emp + first_check +
  second_check + third_check, data = df_weekly)
```

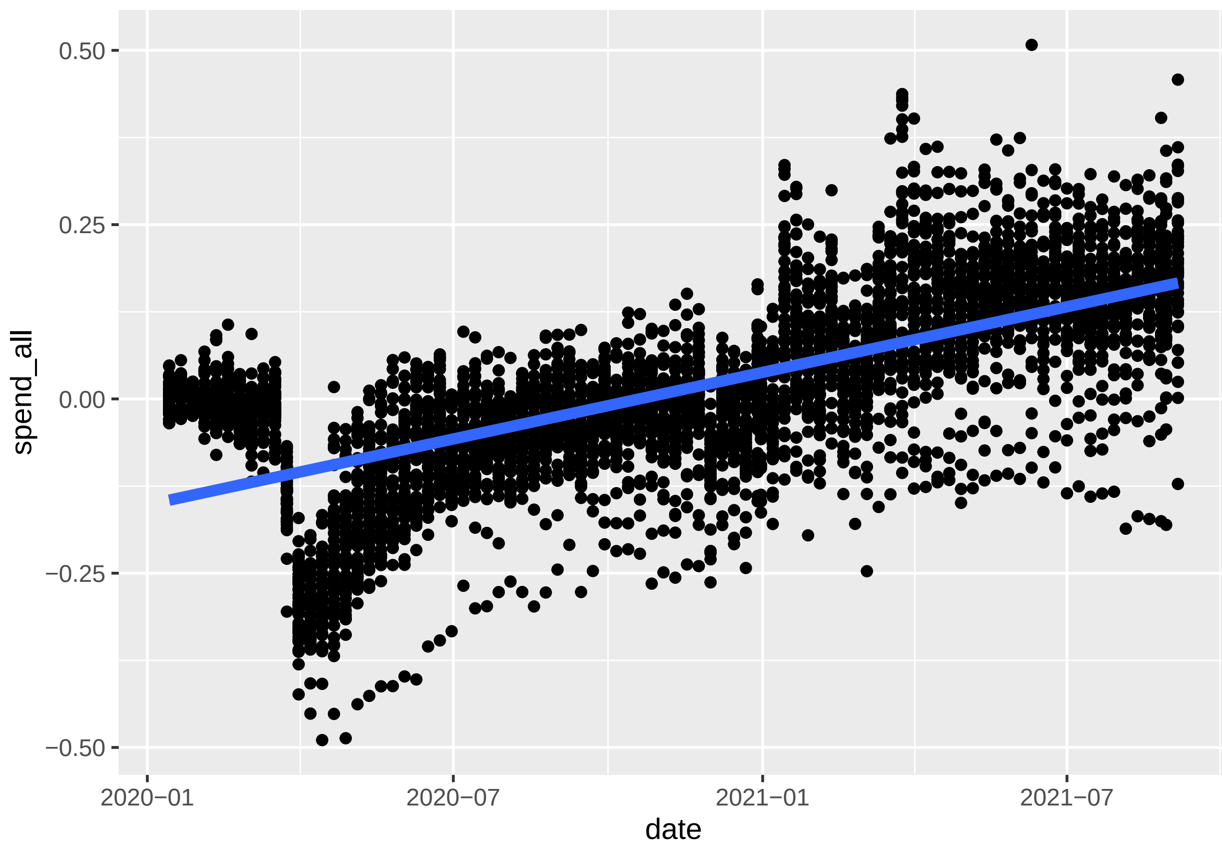
```
summary(mod)
```

```
##
## Call:
## lm(formula = spend_all ~ gps_retail_and_recreation + emp + first_check +
##     second_check + third_check, data = df_weekly)
##
## Residuals:
##      Min       1Q   Median
## -0.34163 -0.04494  0.00693
##       3Q      Max
##  0.04757  0.41405
##
## Coefficients:
##                            Estimate
## (Intercept)               -0.041870
## gps_retail_and_recreation  0.308678
## emp                        0.185988
## first_check                0.044965
## second_check               0.133771
```

```
## third_check                 0.013451
##                            Std. Error
## (Intercept)                  0.003174
## gps_retail_and_recreation   0.012020
## emp                          0.025534
## first_check                  0.003733
## second_check                 0.004405
## third_check                  0.004700
##                            t value
## (Intercept)                -13.190
## gps_retail_and_recreation  25.681
## emp                          7.284
## first_check                 12.045
## second_check                30.367
## third_check                  2.862
##                            Pr(>|t|)
## (Intercept)                 < 2e-16
## gps_retail_and_recreation  < 2e-16
## emp                        3.81e-13
## first_check                 < 2e-16
## second_check                < 2e-16
## third_check                 0.00423
##
## (Intercept)                ***
## gps_retail_and_recreation ***
## emp                        ***
## first_check                ***
## second_check               ***
## third_check                **
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01
##   '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08172 on 4482 degrees of freedom
## Multiple R-squared:  0.6325, Adjusted R-squared:  0.6321
## F-statistic:  1543 on 5 and 4482 DF,  p-value: < 2.2e-16
```

```r
ggplot(df_weekly, aes(x = date, y = spend_all)) + geom_point() + geom_smooth(method = "lm", size = 2)
```

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




### Summarize the Model

Based on the information displayed above, all of the variables we are regressing on are statistically significant for predicting the overall spending.

Our model is fairly limited in how accurately it predicts overall consumer spending because it is a linear model, and because we've limited ourselves to regressing on five variables for the sake of model simplicity, rather than the 30+ that we could've regressed on.

Future analysis could be done using neural networks, gradient boosted decision trees, or recurrent neural networks. We believe that all of these would be able to learn the nuances of our dataset better than a linear model could.

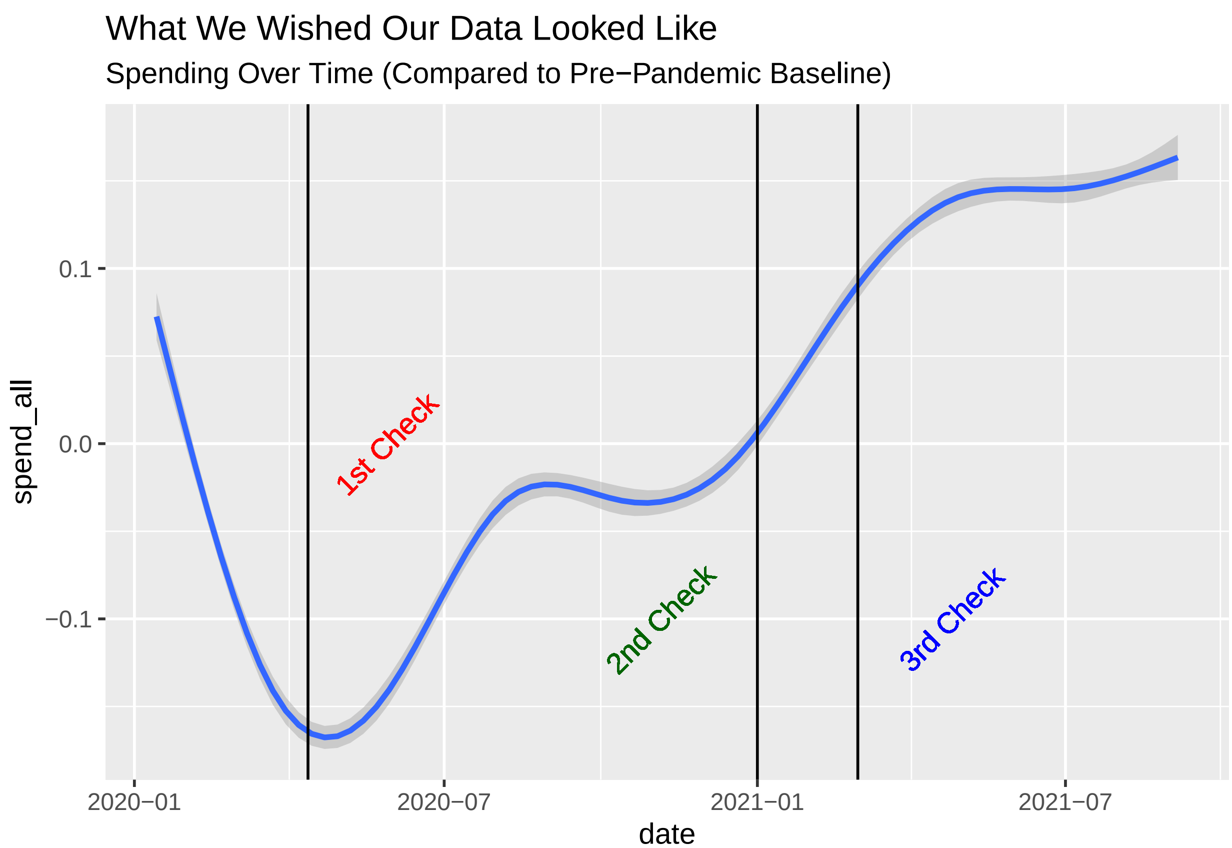
### Plotting spending over time for all states and categories

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

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

ggplot(df_weekly, 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 = "dark green", angle = 45, y = -.1
  ) +
  geom_text(aes(x = as.Date("2021-04-25"), label = "3rd Check"),
    color = "blue",
    angle = 45, y = -.1
  ) +
  labs(
    title = "What We Wished Our Data Looked Like",
    subtitle = "Spending Over Time (Compared to Pre-Pandemic Baseline)"
  )
```

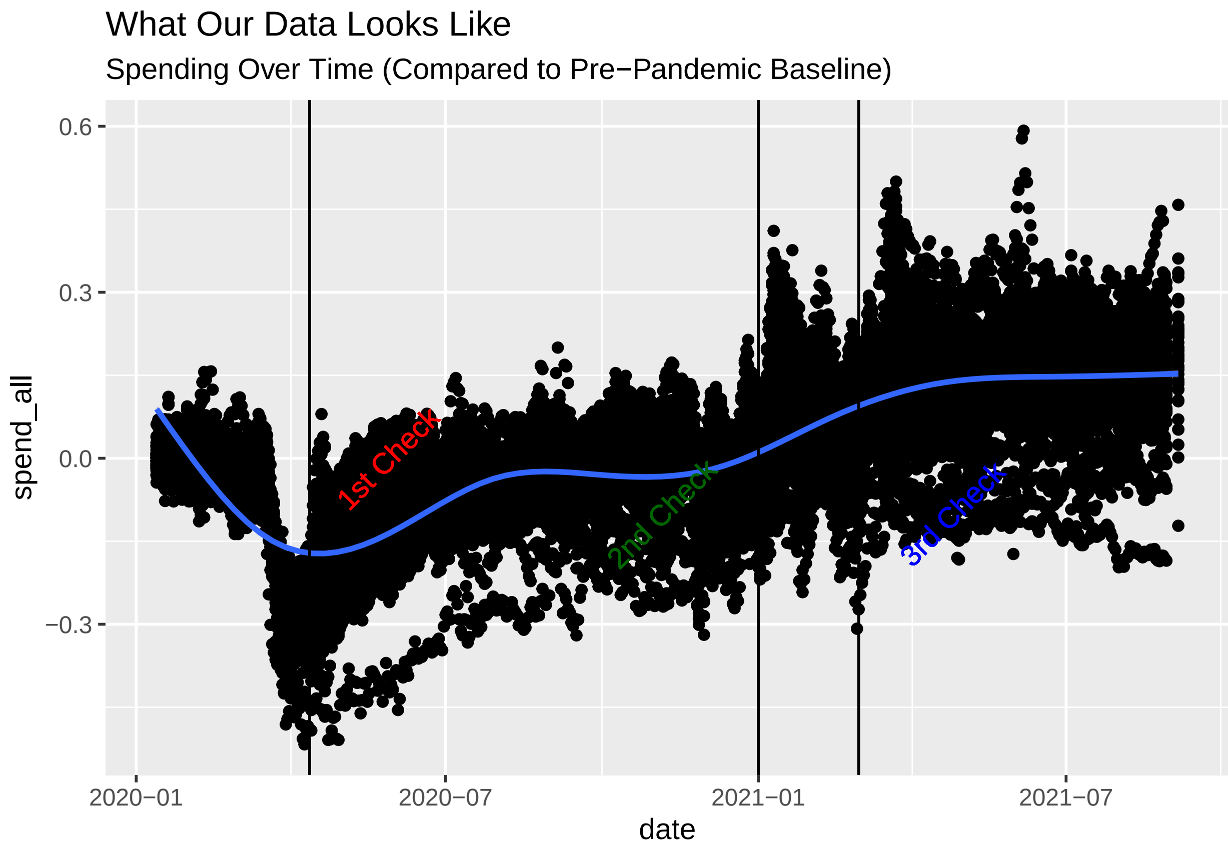
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



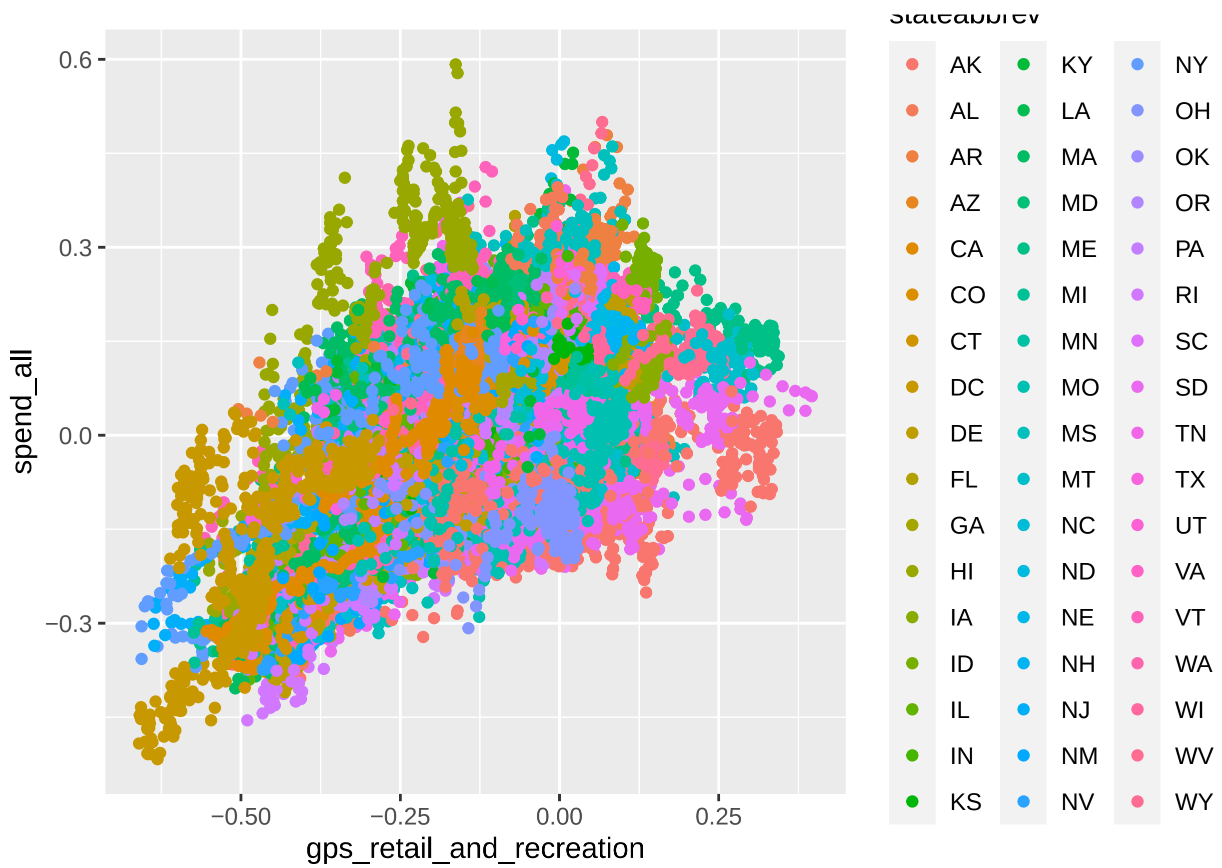

```
# https://stackoverflow.com/questions/38815996/r-adding-geom-vline-labels-to-geom-histogram-labels

ggplot(df, 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 = "dark green", angle = 45, y = -
  geom_text(aes(x = as.Date("2021-04-25"), label = "3rd Check"), color = "blue", angle = 45, y = -.1) +
  labs(title = "What Our Data Looks Like", subtitle = "Spending Over Time (Compared to Pre-Pandemic Base
```

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




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




**Summarize Our Biases**

We were expecting to find that the stimulus checks all had a positive impact on consumer spending, because this is the only outcome that makes any economic sense, as far as we can tell. This effected how we encoded our stimulus check data into feature variables, and it effected how accurate we thought any given model was. If a model said that a stimulus check caused a decrease in consumer spending, we considered that model to be almost certainly highly inaccurate.

**Summarize Biases in the Data**

Our data contains several potential biases that could skew our results. For example, we use movement data from Google. This data was likely pulled from android phones, not Apple products. Imagine, then, that more affluent people tend to purchase Apple products. If this is true, our movement data will be skewed away from affluent people who may be more likely to spend more, more casually.

Another example of possible bias in our data is that our stimulus check data does not fade out overtime, like the effect of the stimulus data presumably does. With the way we're encoding the stimulus data, if we had a fifty year period with a stimulus check in the first year, the data for the next forty nine years would still have the effect from the first check listed as a feature. This seems unrealistic, but we could not find a better way to encode the stimulus check data.

# Conclusion

## Result Summary

The coefficients of our linear model seem to indicate that, unsurprisingly, giving out stimulus checks is correlated with an increase in overall consumer spending.

Therefore, *if* high consumer spending is correlated with a healthier economy, then it seems reasonable to conclude that giving out stimulus checks is also good for the economy, and if they are good for the economy, they are also good for our chances of getting jobs that pay well post-graduation.

The "if" at the beginning of the previous sentence is a big one, though. If the government has to go further into debt to send out stimulus checks, then is the net effect of stimulus checks really good for the economy? We don't know, and this analysis has not considered the effect of increased government debt on the economy. This would be an interesting topic for future analysis.