Stimulus Checks and Consumer Spending

https://github.com/swapnilsethi/DTSC_5301_Project_Economics

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

9/2/2021

Note: We are using the tidyverse and lubridate libraries for our analysis. Make sure, before knitting report these libraries are installed on your system.

See this session info for more insights on the packages we are using. If you are not able to knit the report then you might consider to update your packages to the below versions.

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Big Sur 11.5.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/
    libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/
    libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.33
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.1.1    magrittr_2.0.1    tools_4.1.1       htmltools_0
    .5.1.1
##  [5] yaml_2.2.1        stringi_1.7.3     rmarkdown_2.10    stringr_1.4.0
##  [9] xfun_0.25         digest_0.6.27     rlang_0.4.11      evaluate_0.14
```

INDEX

Motivation and questions of interest

An economic crisis is among those times when a significant number of people are at their most vulnerable, owing to the sudden loss of financial security. In such a situation, the policies put in place by the government to support the economy, and their efficacy, becomes essential. In the past, owing to the time lag in conventional economic data reporting, quantitatively analyzing the efficacy of such policies would take a considerable amount of time. This would mean that the analysis could not help in changing or better tuning the policies during the crisis itself.

However, this lack of data to analyze economic policies in near real-time has not been a constraint during the coronavirus crisis. In this project, we aim to analyze the effectiveness of some of the economic policies applied for the first time using the available data. More specifically, we are interested to answer the following two questions:

1. Did the stimulus checks provided by the US government have a significant positive impact on consumer spending?
2. Did the increase in unemployment compensation provided by the US government cause the unemployment rate to be higher than it would have been otherwise?

The second question is especially interesting as more than 20 states stopped providing enhanced unemployment benefits to citizens in June 2021 while the rest of the states continued to provide it until September. This created a natural experiment to test the effect of unemployment benefits on the unemployment rate.

Why should YOU care how effective the government's economic response is?

Why should we care about consumer spending? It seems like a big picture idea that won't really affect any of us specifically, right? Wrong. Effective economic policies lead to a faster recovery and a healthier economy. 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.

There is also a continuous 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. Our analysis, however, will provide us some insight into the debate between political ideologies on if the enhanced unemployment benefits significantly affect businesses' ability to hire employees.

Data sources and description

Nearly all of the data we use comes from private firms which engage in particular economic areas, and all of this data has been aggregated by Opportunity Insights, which is an NGO at Harvard University. Consumer spending data in our analysis has been originally sourced from Affinity Solutions, visits to retail and recreational establishments from Google, job postings data from Burning Glass, unemployment data from the US Department of Labour, and employment data from Paychex, Intuit, Earnin, and Kronos

The data can be found at- https://github.com/OpportunityInsights/EconomicTracker

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

Reading-in the data

In this code chunk, we read the columns that we're interested in from the data sources. We could use all of the data in the data sources, but we choose not to since not all of the features will be helpful in answering the questions we have.

```r
move_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/Google%20Mobility%20-%20State%20-%20Daily.csv",show_col_types = FALSE, col_select = c(year, month, day, statefips, gps_retail_and_recreation))

affinity_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/Affinity%20-%20State%20-%20Daily.csv", show_col_types = FALSE, col_select = c(year, month, day, statefips, spend_all))

affinity_national_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/Affinity%20-%20National%20-%20Daily.csv", show_col_types = FALSE, col_select = c(year, month, day, spend_all, spend_all_q1, spend_all_q4))

job_listings_weekly_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/Burning%20Glass%20-%20State%20-%20Weekly.csv", show_col_types = FALSE, col_select = c(year, month, day_endofweek, statefips, bg_posts))

employment_daily_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/Employment%20-%20State%20-%20Daily.csv", show_col_types = FALSE, col_select = c(year, month, day, statefips, emp))

employment_national_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/Employment%20-%20National%20-%20Daily.csv", show_col_types = FALSE, col_select = c(year, month, day, emp_incq1, emp_incq4))

ui_claims_weekly_df <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/UI%20Claims%20-%20State%20-%20Weekly.csv", show_col_types = FALSE, col_select = c(year, month, day_endofweek, statefips, contclaims_rate_combined))

state_id <- read_csv("https://raw.githubusercontent.com/OpportunityInsights/EconomicTracker/main/data/GeoIDs%20-%20State.csv",show_col_types = FALSE)
```

Data Cleaning and Transformation

Data cleaning for visualizations

First, let's prepare national level data for visualization. We need a date column with "date" datatype for further analysis and visualization, so let's create one using year, month, and day columns.

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

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

```
employment_national_df <- employment_national_df %>% unite("date", day:month:
    year, remove = FALSE, sep = "-")
```

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

Changing the data type of date column to "date"

```
# https://lubridate.tidyverse.org/reference/ymd.html
affinity_national_df <- affinity_national_df %>% mutate(date = dmy(affinity_
    national_df$date))
employment_national_df <- employment_national_df %>% mutate(date = dmy(
    employment_national_df$date))
```

We also need a week column for our analysis and visualization. In the below code chunk we are creating one in both dataframes.

```
affinity_national_df <- affinity_national_df %>% mutate(week = epiweek(date))
employment_national_df <- employment_national_df %>% mutate(week = epiweek(
    date))
```

Now, let's take look at a summary of the data to see if there are problems

```
glimpse(affinity_national_df)
```

```
## Rows: 967
## Columns: 8
## $ date          <date> 2019-01-07, 2019-01-08, 2019-01-09, 2019-01-10,
    2019-01-~
## $ year          <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019,
    201~
## $ month         <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, ~
## $ day           <dbl> 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
    21, ~
## $ spend_all     <chr> ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".",
     ".~
## $ spend_all_q1 <chr> ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".",
     ".~
## $ spend_all_q4 <chr> ".", ".", ".", ".", ".", ".", ".", ".", ".", ".", ".",
     ".~
## $ week          <dbl> 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4,
    4, ~
```

```
glimpse(employment_national_df)
```

```
## Rows: 557
## Columns: 7
## $ date       <date> 2020-01-14, 2020-01-15, 2020-01-16, 2020-01-17,
   2020-01-18,~
## $ year       <dbl> 2020, 2020, 2020, 2020, 2020, 2020, 2020, 2020, 2020,
   2020, ~
## $ month      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
   2, ~
## $ day        <dbl> 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
   28, ~
## $ emp_incq1 <dbl> -2.23e-03, -1.26e-03, -2.48e-04, 6.96e-04, 1.49e-03, 2.17
   e-0~
## $ emp_incq4 <dbl> 1.22e-03, 6.94e-04, 2.16e-04, -2.00e-04, -5.59e-04, -8.67
   e-0~
## $ week       <dbl> 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5,
   6, ~
```

After taking a closer look at the above output we realized the spend_all, spend_all_q1 and spend_all_q4 columns have “char” datatype instead of “dbl”. Let’s change their data type to a “dbl”.

```
df_national <- affinity_national_df %>%
  select(date, year, month, day, week, spend_all, spend_all_q1, spend_all_q4)
     %>%
  mutate(
    spend_all_q1 = as.double(spend_all_q1),
    spend_all_q4 = as.double(spend_all_q4),
    spend_all = as.double(spend_all)
  ) %>% filter(date > ymd("2020-01-15"))

## 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

emp_national <- employment_national_df %>%
  select(date, year, month, day, week, emp_incq1, emp_incq4) %>%
  mutate(
    emp_incq1 = as.double(emp_incq1),
    emp_incq4 = as.double(emp_incq4),
  ) %>% filter(date > ymd("2020-01-15"))
```

Data cleaning for the first regression model

Now, let's prepare data for our first regression model

```
consumer_spending <- affinity_daily_df
visits_to_retail <- move_daily_df
employment <- employment_daily_df
```

We now join the dataframes of our interest based on a shared date and state of measurements.

```
regression1_df <- left_join(consumer_spending, visits_to_retail, by = c("year"
  , "month", "day", "statefips"))
regression1_df <- left_join(regression1_df, employment, by =c("year", "month",
   "day", "statefips"))
```

We need a date column with "date" datatype for our analysis, so let's create one using year, month, and day columns.

```
regression1_df <- regression1_df %>% unite("date", day:month:year, remove =
  FALSE, sep = "-")

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

regression1_df$date <- dmy(regression1_df$date)
```

Now 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, then $1200 for two weeks after the first check, then zero, then $600 for two weeks after the second check, then zero, then $1400 for two weeks after the third check, and then zero.

```
regression1_df<- regression1_df%>% mutate(
  stimulus_checks = ifelse (date < ymd("2020-04-15"), 0,
                            ifelse (date < ymd("2020-05-01"), 1200,
                                    ifelse (date < ymd("2021-01-04"),0,
                                            ifelse (date < ymd("2021-01-19")
                                                ,600,
                                                    ifelse (date < ymd("
                                                       2021-03-17"),0,
                                                            ifelse (date < ymd
                                                               ("2021-04-01")
                                                               ,1400, 0))))))
                                                               )

head(regression1_df)

## # A tibble: 6 x 9
##   date        year month   day statefips spend_all gps_retail_and_recreat~
   emp
##   <date>     <dbl> <dbl> <dbl>     <dbl> <chr>                       <dbl>
   <chr>
## 1 2020-01-01  2020     1     1         1 .                              NA
   <NA>
## 2 2020-01-01  2020     1     1         2 .                              NA
   <NA>
## 3 2020-01-01  2020     1     1         4 .                              NA
   <NA>
```

```
## 4 2020-01-01   2020     1     1         5  .                          NA
    <NA>
## 5 2020-01-01   2020     1     1         6  .                          NA
    <NA>
## 6 2020-01-01   2020     1     1         8  .                          NA
    <NA>
## # ... with 1 more variable: stimulus_checks <dbl>
```

Taking a closer look at the regression1_df shows that spend_all and emp are both "chr" instead of "dbl" Let's change their data type to a "dbl".

```
regression1_df$spend_all <- ifelse(regression1_df$spend_all == ".", NA,
    regression1_df$spend_all)
regression1_df$spend_all <- as.double(regression1_df$spend_all)

regression1_df$emp <- ifelse(regression1_df$emp == ".", NA, regression1_df$emp
    )
regression1_df$emp <- as.double(regression1_df$emp)
```

Data cleaning for the second regression model

Here we repeat the cleaning and transformation steps that we did for the first linear model, except now we're doing it on weekly data instead of daily data. Specifically for job postings data, we add 1 to the date of the data point, as it will make creating the "regression2_df" easier as the dates will match up with other dataframes that we will be joining.

```
visits_to_retail_weekly <- move_daily_df
visits_to_retail_weekly <- visits_to_retail_weekly %>% unite("date", day:month
    :year, remove = FALSE, sep="-")

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

visits_to_retail_weekly$date = dmy(visits_to_retail_weekly$date)
visits_to_retail_weekly <- mutate(visits_to_retail_weekly,week =isoweek(date))
visits_to_retail_weekly <-  visits_to_retail_weekly %>% group_by(year, week,
    statefips) %>% summarise(gps_retail_and_recreation = mean(gps_retail_and_
    recreation), date =max(date))

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

unemployment_claims <-  ui_claims_weekly_df
unemployment_claims <- unemployment_claims %>% unite("date", day_endofweek:
    month:year, remove = FALSE, sep="-")

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

unemployment_claims$date <- dmy(unemployment_claims$date)
unemployment_claims$contclaims_rate_combined <- as.double(unemployment_claims$
    contclaims_rate_combined)

## Warning: NAs introduced by coercion

job_postings <-  job_listings_weekly_df
job_postings <- job_postings %>% unite("date", day_endofweek:month:year,
    remove = FALSE, sep="-")

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

job_postings$date <- dmy(job_postings$date)
job_postings$date <- job_postings$date+1
job_postings$day_endofweek <- job_postings$day_endofweek+1

regression2_df <- left_join(unemployment_claims, job_postings, by =c("date", "
    statefips", "month", "year", "day_endofweek"))
regression2_df <- regression2_df %>% mutate(week = isoweek(date))
regression2_df <- left_join(regression2_df, visits_to_retail_weekly, by = c("
    week","statefips", "year"))
regression2_df <- left_join(regression2_df, state_id, by = c("statefips"))
regression2_df <- rename(regression2_df, c(visits_to_retail_and_recreation = "
    gps_retail_and_recreation", date = "date.x"))
```

We'll also need to add the column "unemployment_checks", which has values of the enhanced unemployment checks provided in different states since 2020. Some states stopped enhanced unemployment benefits in June and hence we will also have to make sure that the "unemployment_checks" column is a positive number after June only for the states that continued to provide these benefits.

```
states_which_stopped_early_benefits <- c("Arizona", "Indiana", "Maryland", "
    Tennessee", "Alaska", "Iowa", "Mississippi", "Missouri", "Alabama", "Idaho
    ", "Nebraska", "New Hampshire", "North Dakota", "West Virginia", "Wyoming"
    , "Arkansas", "Florida", "Georgia", "Ohio", "Oklahoma", "South Carolina",
    "South Dakota", "Texas", "Utah", "Montana")

regression2_df <- regression2_df %>% mutate(unemployment_checks =
                        ifelse((date >= ymd("2020-03-29") & date<= ymd("2020-07-31")
                            ), 600,
                        ifelse ((date >= ymd("2020-07-26") & date<=ymd("2020-09-05")
                            ), 300,
                        ifelse ((date >= ymd("2020-12-26") & date<=ymd("2021-03-13")
                            ), 300,
                        ifelse ((date >= ymd("2021-03-14") & date <= ymd("2021-06-21
                            ") &
                                (statename %in% states_which_stopped_early_benefits))
                                    ,300,
                        ifelse((date >= ymd("2021-03-14") & date <= ymd("2021-09-04"
                            ) &
                              !(statename %in% states_which_stopped_early_benefits)),
                                   300, 0))))))
```

We, again, replace missing values (such as ".") with NA.

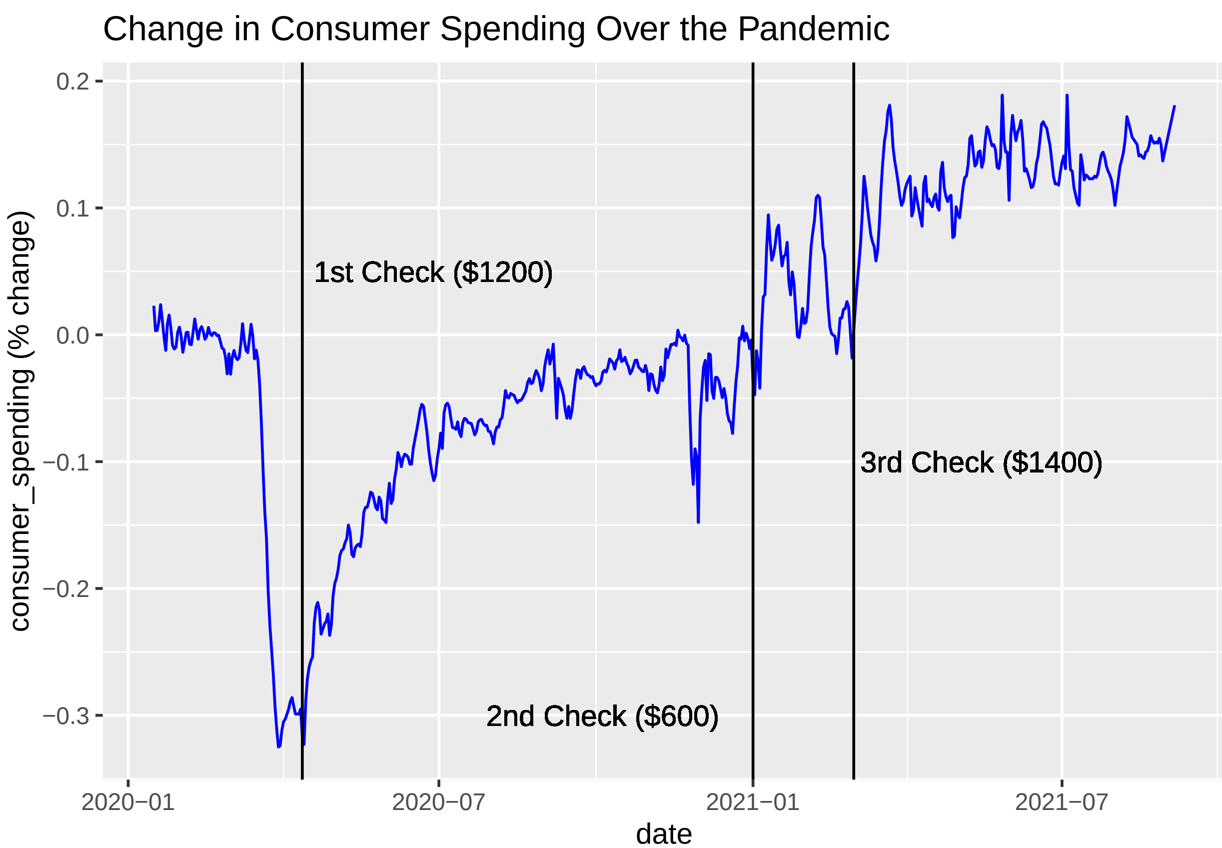
```
regression2_df$contclaims_rate_combined <- ifelse(regression2_df$contclaims_
    rate_combined == ".", NA, regression2_df$contclaims_rate_combined)
regression2_df$contclaims_rate_combined <- as.double(regression2_df$contclaims
    _rate_combined)
```

Visualization and analysis

National Spending Over Time

Now, that we have our data prepared, we can start with visualization. We can start off by charting the change in consumer spending, one of our variables of primary interest. (The dates for the stimulus checks were approximated from this article.)

```
ggplot(df_national, aes(x = date, y = spend_all)) +
  geom_line(color="blue") +
  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("2020-06-28"), label = "1st Check ($1200)"),
    color = "black", angle = 0, y = .05
  ) +
  geom_text(aes(x = as.Date("2020-10-05"), label = "2nd Check ($600)"),
    color = "black", angle = 0, y = -.3
  ) +
  geom_text(aes(x = as.Date("2021-05-15"), label = "3rd Check ($1400)"),
    color = "black",
    angle = 0, y = -.1
  )+
  ylab("consumer_spending (% change)") +
  labs(title = "Change in Consumer Spending Over the Pandemic")
```

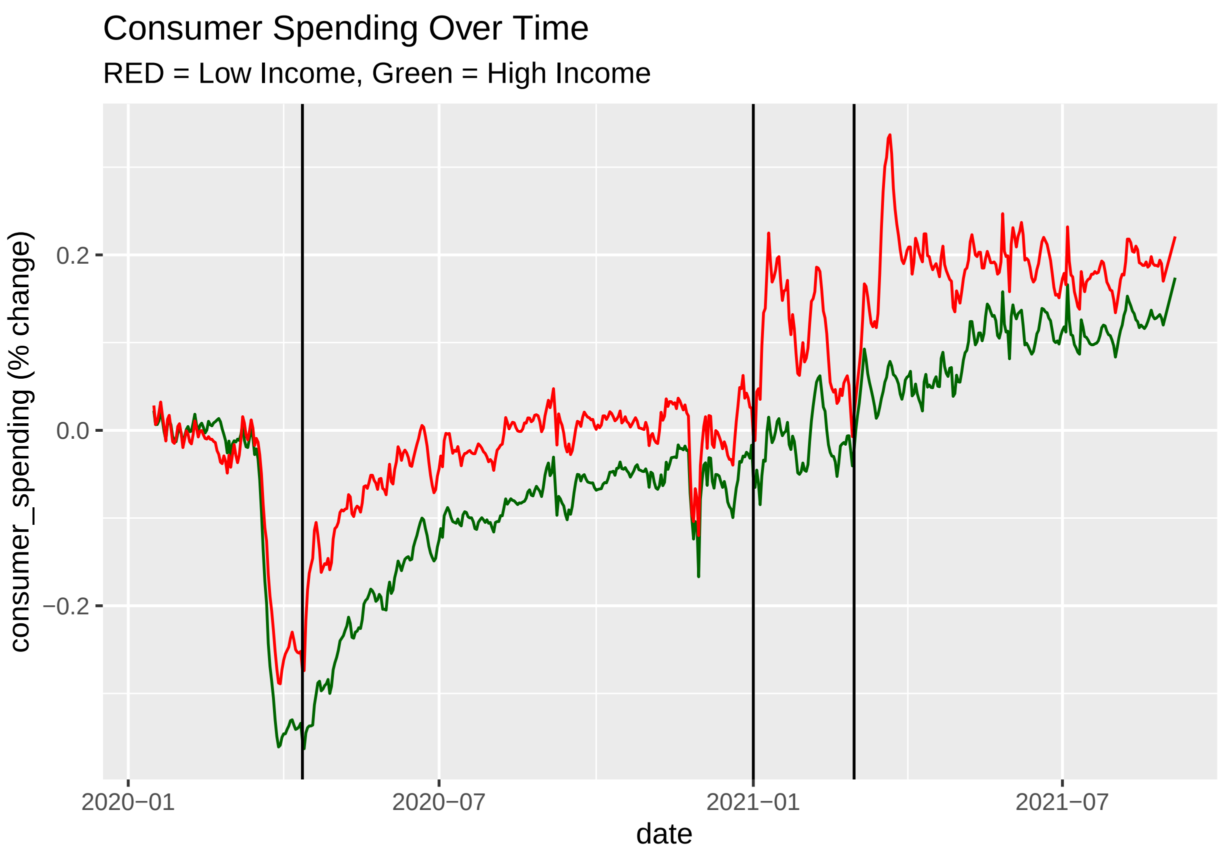


National Spending Over Time (Split by Income)

From the chart, we can see that consumer spending fell sharply when the lockdowns were put in place in early 2020 and have continued to recover ever since as the economy has opened up. From the chart, we also see that consumer spending jumps up right after stimulus checks are sent out by the government. This cursory analysis tells us that stimulus checks did have a relatively significant effect on consumption.

We now make the spending data more granular to see the effect of stimulus checks on the consumer spending of households in the first income quartile and the fourth income quartile.

```
ggplot(df_national, aes(x = date, y = spend_all_q1)) +
  geom_line(aes(y = spend_all_q4), color="dark green") +
  geom_line(color="red") +
  geom_vline(xintercept = as.Date("2020-04-12")) +
  geom_vline(xintercept = as.Date("2021-01-01")) +
  geom_vline(xintercept = as.Date("2021-03-01"))+
  ylab("consumer_spending (% change)") +
  labs(title = "Consumer Spending Over Time", subtitle = "RED = Low Income,
      Green = High Income")
```



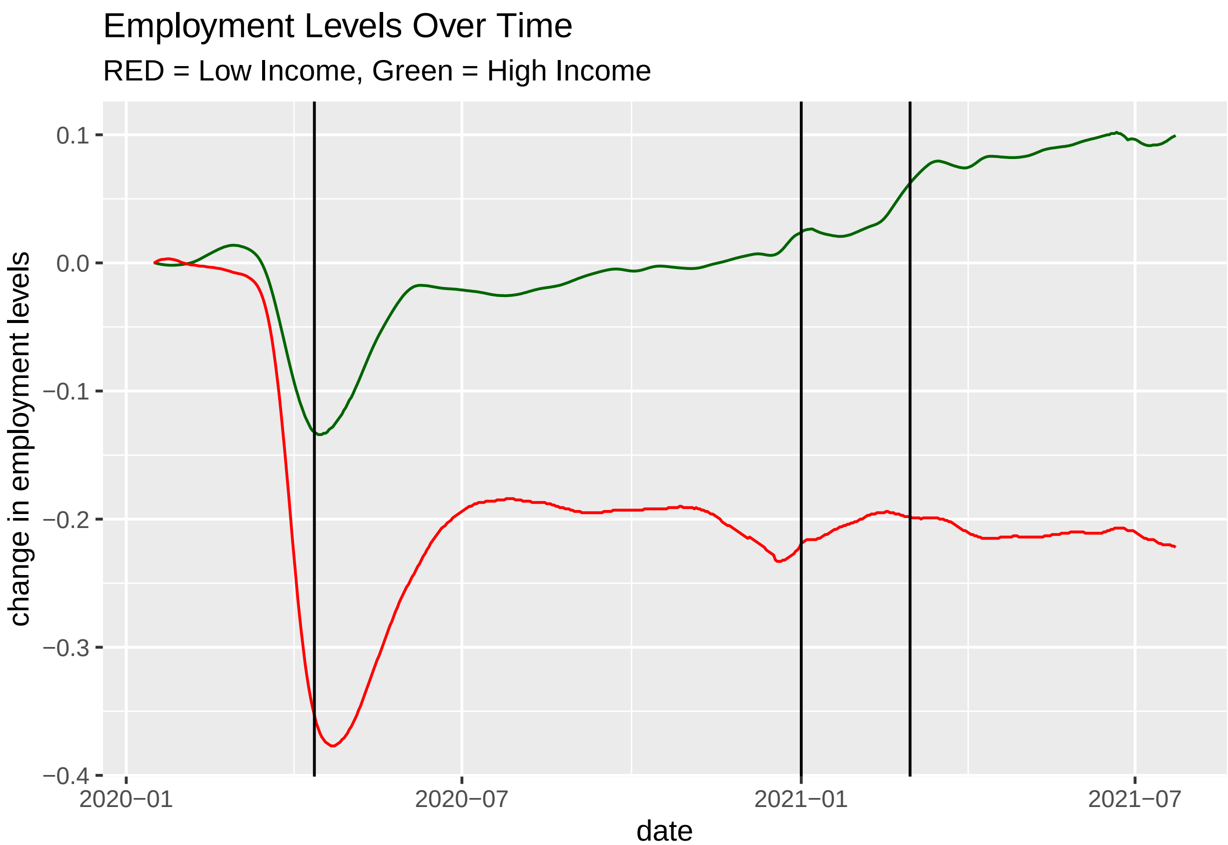
From the above graph, we see that the stimulus checks have supported consumer spending for low-income households much more strongly than it has for high-income households. This makes sense as a large part of the high-income households would not have received stimulus checks in the first place and also because spending by low-income households would be much more dependent on the stimulus checks than that of high-income households.

Interestingly, we also see in this graph that the fall in consumer spending by low-income households was shallower than that of high-income households. Furthermore, the spending growth of low-income households has remained above that of high-income households since the lockdowns. This is interesting because we would normally expect spending of high-income households to fall less as they would be more financially secure.

National unemployment rate over time (split by income)

To understand why this is happening, we next take a look at employment levels over time in high-income and low-income households, as maybe that would explain the difference.

```
ggplot(emp_national, aes(x = date, y = emp_incq1)) +
  geom_line(aes(y = emp_incq4), color="dark green") +
  geom_line(color="red") +
  geom_vline(xintercept = as.Date("2020-04-12")) +
  geom_vline(xintercept = as.Date("2021-01-01")) +
  geom_vline(xintercept = as.Date("2021-03-01")) + ylab("change in employment levels") +
  labs(title = "Employment Levels Over Time", subtitle = "RED = Low Income, Green = High Income")
```



In the above graph, we see that employment in high-income households has recovered and even risen above that of pre-pandemic levels. However, employment levels for low-income households have remained depressed. This stands in contrast with the initial shallower fall and later faster growth in consumption by low-income households. We believe that support provided by the government in terms of stimulus and unemployment checks which disproportionately benefit low-income households at least partly is responsible for this consumer spending discrepancy between the two.

Spending Regression Model

Training the first linear model on the data

In this code chunk, we fit a linear model to the *gps_retail_and_recreation*, *emp*, and *stimulus_checks* 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 three input variables is that *stimulus_checks* 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 the change in spending that could not necessarily be accounted for by the *stimulus_checks*. 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.

```
spending_regression <- lm(spend_all ~ gps_retail_and_recreation + emp +
    stimulus_checks, regression1_df)

summary(spending_regression)
```

```
##
## Call:
## lm(formula = spend_all ~ gps_retail_and_recreation + emp + stimulus_checks,
##     data = regression1_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33348 -0.07187 -0.00643  0.06591  0.57687
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               7.981e-02  1.003e-03   79.54   <2e-16 ***
## gps_retail_and_recreation 2.000e-01  7.927e-03   25.23   <2e-16 ***
## emp                       8.467e-01  1.409e-02   60.10   <2e-16 ***
## stimulus_checks           5.551e-05  2.139e-06   25.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1038 on 22536 degrees of freedom
##   (8468 observations deleted due to missingness)
## Multiple R-squared:  0.4067, Adjusted R-squared:  0.4067
## F-statistic:  5150 on 3 and 22536 DF,  p-value: < 2.2e-16
```

Based on the information displayed above, all of the variables we are regressing on are statistically significant for predicting overall spending. The positive coefficient for the *stimulus_checks* variable in the linear model seems to indicate that *stimulus_checks* are positively correlated with overall consumer spending (the variable we're predicting with our linear model).

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 three 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.

Unemployment Regression Model

Training the Second Linear Model on Our Data

In this code chunk, we fit a linear model to the *visits_to_retail_and_recreation*, *unemployment_checks*, and *bg_posts* features, with the goal of predicting the total continued unemployment claims variable. The working assumption here is that *contclaims_rate_combined* is a dependent variable, and the others are independent variables. Again, we chose only these three variables to regress on for the sake of model simplicity.

```
unemployment_regression <- lm(contclaims_rate_combined ~ 0 + bg_posts + visits
    _to_retail_and_recreation + unemployment_checks, regression2_df)

summary(unemployment_regression)

##
## Call:
## lm(formula = contclaims_rate_combined ~ 0 + bg_posts +
    visits_to_retail_and_recreation +
##     unemployment_checks, data = regression2_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.258  -2.238   1.828   5.305  56.342
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## bg_posts                         -1.01087    0.53023  -1.906   0.0567 .
## visits_to_retail_and_recreation -21.59414    0.86394 -24.995   <2e-16 ***
## unemployment_checks               0.01850    0.00044  42.039   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.409 on 3873 degrees of freedom
##   (612 observations deleted due to missingness)
## Multiple R-squared:  0.6359, Adjusted R-squared:  0.6356
## F-statistic:  2254 on 3 and 3873 DF,  p-value: < 2.2e-16
```

Of the variables we regressed over visits to retail and recreational establishments are significant at the 99% confidence interval, while job postings (bg_posts) are significant at 94% confidence interval for predicting the change in unemployment since the pre-pandemic baseline. The positive coefficient for the *umemployment_checks* variable in the linear model seems to indicate that *unemployment_checks* are positively correlated with overall unemployment rates. This is a rather odd way of saying that higher unemployment checks are correlated with higher unemployment rates.

This model is also fairly limited for a couple of reasons: we're using linear modeling and only regressing on a small set of features for the sake of model interpretability and simplicity.

As with the other model, the 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.

Biases

Model 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 affected how we encoded our stimulus check data into feature variables, and it affected how accurate we thought any given model was while tweaking it. If a model said that a stimulus check caused a decrease in consumer spending, we considered that model to be almost certainly highly inaccurate.

Data Biases

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.

Conclusion

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

Additionally, our second linear model seems to indicate that giving out extra unemployment checks is correlated with higher unemployment levels.

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 and unemployment checks, then is the net effect of these 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.