STAT 156 Final Project: Replication and Improvement on "How do 401(k)s Affect Saving? Evidence from Change in 401(k) Eligibility"

Yehchan Yoo and Xinyi Zi

December 16, 2022

The video presentation URL is https://youtu.be/Tpw7Ch7XJDg.

1 Paper and Data Summary

1.1 Paper Summary

In 2011, a research paper titled "How do 401(k)s Affect Saving? Evidence from Changes in 401(k) Eligibility" was published with Alexander M. Gelber as the author. The 401(k) plans are currently the primary way for Americans to save up for retirement and began in 1978 with the passage of the Revenue Act in the United States Congress (Elkins, 2017). These plans are sponsored by American companies. They work by having employees of these companies contribute a proportion of their income to a retirement investment account with

the employer potentially matching this contribution; these plans allow Americans to save up for their retirement with assistance from their employers and with major tax savings (Fernando, 2022).

The purpose of Gelber's paper is to investigate the impact that 401(k) eligibility has on saving. In theory, being eligible for a 401(k) might discourage personal saving because some people might see it as a substitute for other forms of saving. Plus, the 401(k) program is costly to the government and contains a significant portion of personal savings in America. Also, the erratic changes in 401(k) balances due to volatility of asset prices since 2008 put the social value of 401(k) plans and defined contribution pensions into question. So, it is important to understand how the 401(k) program actually affects the saving rate in the United States so that the United States government can develop proper policies in regards to saving rates. Before this paper, there was no consensus on the effect of 401(k) program on personal savings. Also, according to the author, past academic work had some empirical limitations such as unobserved taste for saving that is correlated with 401(k) eligibility, changes in composition of 401(k)-eligible and 401(k)-ineligible populations, and potentially confounding cohort differences. This paper contributed to the existing literature on the effect of 401(k) by trying to improve upon these limitations and by concluding that being eligible for 401(k) significantly increases 401(k) savings. Although the impact on other forms of savings is not statistically significant, the study notes that the confidence intervals (or CI's) are so large that the possibility of 401(k) eligibility having a noticeable impact on 401(k) savings still remains (Gelber, 2011, pp. 103-105).

For our paper, we hope to replicate, critique, and re-analyze the data analysis methods found in "How do 401(k)s Affect Saving? Evidence from Changes in 401 (k) Eligibility" by

Alexander M. Gelber.

1.2 Data Summary

This study uses the 1996 Survey of Income and Program Participation (SIPP) data (Gelber, 2011, p.105). To get a hint at what data was used in this paper, we downloaded the author's published replication files for this paper from OpenICPSR. The replication files consisted of a README file, a `do` file allowing for replication of the study's programming methods in STATA, and multiple `dta` files containing the SIPP data the author used (Gelber, 2019).

The 1996 SIPP data consist of two types of data files: panel longitudinal core data and topical module data (*Survey of Income and Program Participation (SIPP)*, 2017). Exploring the published data files, we found that the core data focused on demographic and employment information while the topical data focused on the liabilities and financial assets that the respondents own. We also found that the published data contained topical data from waves 3, 6, 7, 9, and 12 along with panel longitudinal core data from waves 7, 8, and 9 (Gelber, 2019).

The data is collected in waves, and there are 4 months between two adjacent waves, which means there are a total of 3 waves in each year. Respondents' financial assets are recorded in wave 3, 6, 9, and 12. As visualized in Figure 1 below directly taken from the paper, the respondents' eligibility for 401(k) is measured in wave 7. The study refers to 1996 as "year 0", 1997 as "year 1", and 1998 as "year 2". Correspondingly, waves 3-6 are observed in year 0, waves 6-9 are observed in year 1, and waves 9-12 are observed in year 2 (Gelber,

2011, pp. 105-106).

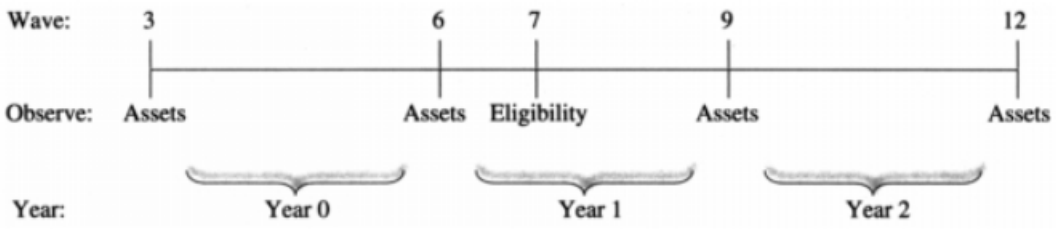


FIGURE 1. TIMING OF OBSERVATIONS

In order to fully replicate the data analysis done in Gelber's paper, we decided to use raw 1996 SIPP data for our replication of Gelber's paper. Zip folders containing the `dat` raw data files from SIPP could be found on the United States Census website and on the National Bureau for Economic Research (NBER) website. As we did find using Powershell that the raw 1996 SIPP data files from the United States Census were identical to corresponding data files from NBER, we downloaded both the `dat` and the guiding `dct` files from NBER for convenience (*SIPP 1996 Panel Data*, 2022; *Survey of Income and Program Participation (SIPP)*, 2017).

We used the `dct` dictionary files for these data files from NBER for guidance with the cleaning process. The dictionary files contain information on how many characters each column takes up, what the name of each column is, what type of data each column contains, and what information each column represents. Due to the massive size of the data files, these data and dictionary files were uploaded to Deepnote to allow for memory-intensive analysis of the data. Also, instead of Deepnote's basic computer with 5 gigabytes (GB) of random access memory (RAM) and two virtual central processing units (vCPUs), we used a more advanced computer from Deepnote's computer selection with 4 vCPUs, 60 GB of RAM, and 12 GB of K80 graphical processing unit (GPU) memory. From here, we used both our own

R code and R code based off of the author's published `do` file to clean the raw SIPP data files.

Much of the data cleaning effort was spent on converting the fixed-width data in `dat` files to R `data.frame` objects and on filtering for the respondents that met the following five criteria of the study as done in the paper:

1. The respondent must work at a for-profit firm.
2. The firm must offer 401(k).
3. The respondent started the current job one year or less before wave 7.
4. The respondent is under 65 years of age but over 21.
5. The respondent did not switch job between year 1 and year 2.

Another major component of our data cleaning process was the creation of new variables. As done by the author in his `do` file, we created an indicator variable for whether the respondent was temporarily ineligible for 401(k) because they had not worked long enough for the firm. Also, as done by the author, we also created variables that aggregated the financial assets each respondent had in each of the relevant waves. In addition, as done in the paper, we log-transformed the variables for financial assets; the author log-transformed these variables due to the approximate log-normal distribution of these variables, and we decided to follow suit (Gelber, 2011, pp. 105-107; Gelber, 2019).

1.3 Summary Statistics

After cleaning the data, we calculated the summary statistics for some of the main variables and covariates from Wave 6 for all authors as done by the author in Table 1.

	Name	Weighted Mean	Mean	Median	IQR	SD
1	Age	36.80	37.30	36.00	15.00	9.90
2	Yearly Household Income	58967.90	58263.90	50501.00	43017.00	39220.70
3	401(k)	6044.70	6064.00	0.00	0.00	22560.10
4	IRA	7834.80	7397.30	0.00	0.00	25833.70
5	Other Assets	36745.80	37915.70	2000.00	11269.00	182031.60
6	Secured Debt	61681.10	60480.40	33500.00	95186.50	76150.00
7	Unsecured Debt	6838.00	6854.10	2000.00	8000.00	14169.80
8	Car Value	11875.00	11860.20	11000.00	12584.50	9443.60

Table 1: Summary Statistics

According to author's `do` file, the means of all the variables in Table 1 of his paper are weighted (Gelber, 2019). So, to get a better understanding of the data, we included both weighted and non-weighted means in our summary table. We also included not only standard deviation but also median and interquartile range (IQR) to get a better understanding of the spread of our data variables.

There are some differences between this table of summary statistics and Table 1 in author's paper. For instance, the weighted mean of the yearly household income is $60,389.70 in the author's paper, but the weighted mean of the yearly household income is $58,967.90 in our summary statistics table. The weighted mean for individual retirement account (IRA)

assets were the same for both tables, but the standard deviation was different with the value on the author's table being $27,130.50 and the value on our table being $25,833.70 (Gelber, 2011, p. 110).

The differences potentially come from the author's lack of transparency in his cleaning process. While the author's `do` file shows how to deal with the given datasets in `dta` format, the author does not include any code in his published files or in his paper on how he obtained and cleaned the original `dat` files (Gelber, 2019). So, there may be steps in the cleaning process that the author might not have included in either the paper or the published files; such steps might be the reason why there are some differences between our and the author's summary statistics tables.

One interesting observation is that the medians and the IQRs for 401(k) saving and IRA saving are 0. This means that less than 25% of the respondents had assets in these two categories, so the distribution of these two assets are highly skewed to the right. In addition, "Other Assets" is also right-skewed with a very large dispersion. Overall, we can see that financial assets are concentrated on a small percentage of wealthy respondents.

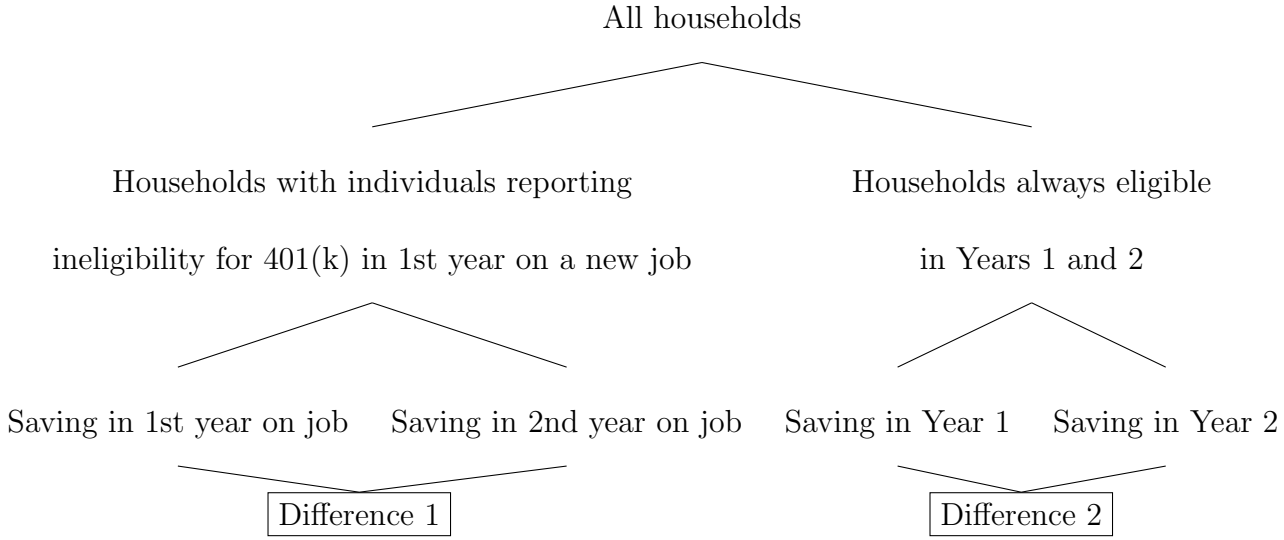
2 Replication of the Main Results

2.1 Author's Methodology

In the paper, Gelber uses a difference-in-difference strategy for causal inference. He defines the treatment group as the group of 1996 SIPP respondents who reported to be ineligible for 401(k) in their first year of a job because they have not worked long enough.

In contrast, the control group is the group of 1996 SIPP respondents who have always been eligible for the 401(k) plan. Therefore, the treatment is becoming eligible in the second year on the job, while the control is always being eligible for 401(k) on both years. The treatment is coded as a dummy variable "becoming eligible", which equals 1 if the respondent belongs to the treatment group, or 0 if the respondent belongs to the control group (Gelber, 2011, p. 106).

Gelber obtains respondents' savings over their first year of employment and savings over the second year. He then obtains the difference between savings in these two years. Finally, he compares the difference in savings between the treatment group and the control group (Gelber, 2011, p. 105). Note that the underlying assumption here is that people who were ineligible in the first year would automatically become eligible in the second year. Similarly, people who were eligible in the first year would remain eligible in the second year. A diagram was added below to make his methodology more clear.



Linear regression is used to implement the author's difference-in-difference method. The dependent variable is the difference between the natural log of savings on a type of assets

in the second year on a job and in the first year on the job. Or, in mathematical terms, the dependent variable is:

$$\ln A_{12} - \ln A_9 - [\ln A_9 - \ln A_6]$$

where A stands for a type of asset such as 401(k) savings and the subscript stands for wave number. So, A_9 would represent the amount of a type of asset from Wave 9.

The independent variables are the dummy variable that equals 1 if the respondent is temporarily ineligible; and the confounders that we wish to control for. The confounding variables include age, household income, education level, size of the firm that the respondents work for, industry, and the number of days the respondents have worked on the job. Gelber ran three regressions in his main results: one without any of the controls, one with the controls, and one with respondents' initial assets in wave 6 in addition to all of the controls (Gelber, 2011, pp. 110-112).

Main Results			
	Becoming eligible without controls	**Becoming eligible + covariates**	**Becoming eligible + covariates + initial assets**
401(k) Saving	0.949	0.932	1.03
95% CI for 401(k) Saving	[0.386, 1.51]	[0.364, 1.5]	[0.462, 1.6]
IRA Saving	0.557	0.526	0.46
95% CI for IRA Saving	[0.038, 1.08]	[0.027, 1.03]	[-0.03, 0.951]

Table 2: Coefficients and Confidence Intervals for the "Become Eligible" Dummy Variable

Through replication, we obtained similar results as in the paper. For the regression without any controls, the coefficients on "become eligible" are only significant for 401(k) savings and equity in IRA accounts, but not for the other types of financial assets. Specifically, the coefficient for "become eligible" is 0.949 for 401(k) saving, and 0.557 for IRA assets in the regressions without controls. Also, the results with and without controls appear to be very similar, which also aligns with the author's results. Finally, the causal effect on 401(k)

saving becomes larger when the initial balance is included, together with all the controls. In contrast, the causal effect for IRA saving decreases. The results are still only statistically significant for 401(k) saving and IRA saving. The coefficient for "become eligible" is 1.03 for 401(k) saving, and 0.46 for IRA assets.

Note that we omit elaborating on the other types of assets here and in later parts of the report because the causal effect on those assets were never statistically significant.

We included below a derivation of what the coefficient means to help interpret the result. Let τ denote the coefficient for the "become eligible" dummy variable.

$$\begin{aligned}\hat{\tau} &= [\ln A_{12} - \ln A_9 - (\ln A_9 - \ln A_6)]_{\text{treatment}} - [\ln A_{12} - \ln A_9 - (\ln A_9 - \ln A_6)]_{\text{control}} \\ &= \left[\ln \frac{A_{12}}{A_9} - \ln \frac{A_9}{A_6}\right]_{\text{treatment}} - \left[\ln \frac{A_{12}}{A_9} - \ln \frac{A_9}{A_6}\right]_{\text{control}}\end{aligned}$$

As seen above, the coefficient can be interpreted as the difference in difference between increase in saving between year 1 and year 2, caused by becoming eligible for the 401(k) plan, as a ratio.

2.2 Assumptions and Critiques

The assumption made in carrying out the causal identification is that people who were not eligible for 401(k) in their first year on a job all became eligible in the second year. At the same time, those who were eligible in the first year are assumed to remain eligible. While the second assumption is reliable to a large extent, as employers are not likely to cancel the 401(k) program for those who are already enrolled, the first assumption deserves some critiques. It is not guaranteed that employers who do not offer a 401(k) plan in the first year would offer it in the second year. For example, they might decide to only offer it

to employees who have worked for 2 or 3 years in order to encourage employees to stay on the job longer. Thus, a violation of the assumption is likely.

This assumption of becoming eligible in the second year is also problematic because it is possible that a respondent who was ineligible at the time of wave 7 became eligible shortly after wave 7 but long before wave 9. As mentioned before, the author defined year 1 as wave 6 to wave 9, and year 2 is defined as wave 9 to wave 12. The respondents' eligibility for 401(k) is surveyed at wave 7 (Gelber, 2011, p. 105). However, note that there are 8 months between wave 7 and wave 9. Even though the author only selected respondents who worked on their jobs fewer than one year at the time of wave 7, the data does not capture exactly how many months they had worked on the job. As a result, it is very likely that some respondents in the treatment group became eligible shortly after wave 7 because they had reached the second year of their employment. In this case, the difference between savings in wave 6 to wave 9 and wave 9 to wave 12 does not capture the changes in saving due to change in eligibility, as the respondent in the treatment group already became eligible in year 1, and no change in eligibility happened in year 2. This would mean that the difference-in-difference the author captures is due to some factors other than change in eligibility.

Another weakness in the method the author used is that it does not account for variables that are potentially inconsistent across time at all. These time-inconsistent variables refer to variables that can change from one year to the next, and could influence the respondents' willingness to save. For example, the profitability of the firms the respondents worked for is a time-inconsistent variable. If a firm did very well in year 1 of the study, but started to lose a lot of money in year 2, it is natural that the respondents working at this firm would want to save more in year 2 because they would be worried about a potential layoff.

Failure to account for this type of variables suggests that there potential exists uncontrolled confoundedness that could seriously undermine the causal relationship.

3 Replication of Robustness Checks

Robustness Checks			
	1st check: Winsorize outliers	**2nd check: p score stratification**	**3rd check: accumulation of assets in year 1**
401(k) saving	0.743	0.65	-0.446
95% CI for 401(k) saving	[0.288, 1.2]	[-0.624, 1.924]	[-0.724, -0.169]
IRA saving	0.412	0.57	-0.186
95% CI for IRA saving	[0.068, 0.756]	[-0.547, 1.687]	[-0.436, 0.064]

Table 3: Robustness Check Results

We replicated three robustness checks the author conducted. The first robustness check dealt with outliers. There are some observations with large amounts of assets. To test the robustness of the causal effect against outliers, the asset data were winsorized at 5th and 95th percentiles, meaning that all values below the 5th percentile were set to the 5th percentile and all values above the 95th percentile were set to the 95th percentile (Gelber, 2011, pp. 114-115). The causal effects for 401(k) saving and IRA saving were still statistically significant; but the magnitude became smaller, especially for 401(k) saving. The coefficients are 0.743 for 401(k) saving and 0.412 for IRA, compared to 1.03 and 0.46 before removing the outliers. This suggests that the causal effect on IRA saving is more robust than that on 401(k) saving.

Next, propensity score stratification was done to help to account for other potential differences between the control and treatment groups (Gelber, 2011, p. 116). Propensity

score is mathematically defined as:

$$e(X, Y(1), Y(0)) = \text{pr}\{Z = 1|X, Y(1), Y(0)\}$$

but often estimated with formula $e(X) = \text{pr}\{Z = 1|X\}$ under strong ignorability. Demographic and socio-economic information were accounted for when we stratified the data by propensity score. Although we got coefficients similar to what the author has for his propensity score match in Panel B of Table 4, we had larger standard errors and the 95% confidence intervals for both 401(k) saving and IRA saving contained 0. This means that the causal effect might be quite small and there might not be much of a difference at all between the two groups after accounting for more observable characteristics.

The third robustness check focused on savings accumulated in the first year (wave 6 to wave 9). In the third robustness check, log of wave 9 asset is regressed on "become eligible" dummy, log of wave 6 asset, and the control variables. The purpose of this robustness check was to see if the control group actually saved less when they were not eligible for the 401(k) plan (Gelber, 2011, pp. 115-116). Only the coefficient in the regression for 401(k) saving was statistically significant. The coefficient was -0.446, indicating that the control group saved more than the treatment group in year 1, when the treatment group was theoretically ineligible for 401(k).

Re-Analysis		
	Propensity score matching	**Doubly robust estimator**
401(k) saving	0.686	1.044
95% CI for 401(k) saving	[-0.157, 1.529]	[0.287, 1.802]
IRA saving	0.431	0.206
95% CI for IRA saving	[-0.347, 1.209]	[-0.345, 0.756]

Table 4: Re-Analysis Results

4 Re-Analysis

4.1 Propensity Score Matching

We applied propensity score matching to make the control and treatment groups more comparable and used simple ordinary least squares regression to check if the covariates were balanced by propensity score matching. Although the point estimate for the causal effect on 401(k) saving and IRA saving obtained through matching were similar to the point estimate obtained through stratification, their standard errors became even larger. As a result, the 95% confidence intervals crossed 0, suggesting that there might not be a causal effect at all.

By regressing the "become eligible" dummy variable on the covariates, we observe that six of the covariates are unbalanced before matching. After matching, two of the covariates remain unbalanced: value of the car owned by the respondents in wave 3 and wave 6. The negative coefficients on these two variables suggests that the control group had more valuable cars and this difference cannot be mitigated through matching. The values of the cars owned by respondents indicate not only their financial capaacity but also their consumption

behaviors. Having significant imbalance in this variable undermines the causal identification because it is possible that the observed causal effect is simply a result of different spending habits. Thus, we need to come up with an analysis that addresses this, as detailed in the next section.

4.2 Doubly Robust Estimator

As some covariates continued to be unbalanced even after matching, we decided to calculate doubly robust estimators to calculate the causal effect of the treatment variable on 401(k) and IRA savings. The doubly robust estimator involves using both propensity score calculations and linear regression to calculate causal effect. The estimator is called "doubly robust" because it is consistent if either the propensity score model or the outcome model is correctly specified. This means that the estimator is correct if only one of the two aforementioned models is correct and the other is misspecified (Alves, n.d.). Doubly robust estimator has been suggested as a solution for doing causal inference if covariate imbalance remained even after propensity score matching (Nguyen et al., 2017). The formula for the doubly robust estimator is $\hat{\tau}^{\text{dr}} = \hat{\mu}_1^{\text{dr}} - \hat{\mu}_0^{\text{dr}}$, where

$$\hat{\mu}_1^{\text{dr}} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{Z_i\{Y_i - \mu_1(X_i - \hat{\beta}_1)\}}{e(X_i, \hat{\alpha})} + \mu_1(X_i, \hat{\beta}_1)\right]$$

$$\hat{\mu}_2^{\text{dr}} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{(1 - Z_i)\{Y_i - \mu_0(X_i - \hat{\beta}_0)\}}{1 - e(X_i, \hat{\alpha})} + \mu_0(X_i, \hat{\beta}_0)\right]$$

with $e(X, \hat{\alpha})$ being the fitted values of propensity scores and $\mu_1(X_i - \hat{\beta}_1)$ and $\mu_0(X_i - \hat{\beta}_0)$ being the fitted values of outcome means.

We calculated the doubly robust estimators for 401(k) and IRA assets using the same treatment and covariate variables used by the author for his original propensity score calculations – that is, the "become eligible" dummy variable was used as the treatment variable with outcome variables being the difference-in-difference values of logs of 401(k) assets and of IRA assets and with covariate variables being age, education, household income, gender, household size, firm size, 1-digit industry, and the values in both Waves 3 and 6 of 401(k) balance, IRA balance, other assets, secured debt, unsecured debt, and cars.

Also, when we ran the robustness check and the propensity score matching, we omitted all rows with missing values not only to prevent these rows from causing errors, but also because this was one of the two ways that the author dealt with missing data in his paper. Most notably, for some parts of his analysis, he finds that household income is missing for 17 observations; so, he tries doing his analysis after removing these observations. However, the author also tries doing the same analysis by creating a dummy column indicating 1 for rows with missing household income data and 0 for other rows (Gelber, 2011, p. 111). So, for doubly robust estimator calculation, we decided to do the same thing by imputing missing values in each column with -1 and indicating which values in that column were missing in another dummy column in order to account for rows with missing values.

Calculating the doubly robust estimator, we found that the coefficient for the treatment vector was pretty similar to those from the replicated main results. The doubly robust estimator for 401(k) assets was similar to that from panel C of table 2 in the paper, though the doubly robust estimator had a greater standard error; the result was also significant at the 5 percent level. The doubly robust estimator for IRA assets was much smaller than those from the replicated main results with similar standard errors; the doubly robust estimator

was not significant here at 5% level.

5 Conclusion

In this project, we attempted to replicate Alexander M. Gelber's 2011 study on the causal effect that being eligible for the 401(k) plan has on savings. We cleaned the data from scratch, replicated the author's main result, replicated three robustness checks, and used propensity score matching and doubly robust estimators to improve on the method. By replicating the author's method, we obtained similar results as in the paper: the causal effect is statistically significant for 401(k) saving and IRA saving, but not for the other types of assets.

We think the author's assumption that people who were not eligible in year 1 would all become eligible in year 2 is fragile for two reasons: first, there is no guarantee that they would actually become eligible; secondly, they could become eligible shortly after the survey about eligibility, which happened in year 1. Thus, it is not guaranteed that the causal effect identified is actually a result of becoming eligible for 401(k).

The robustness checks revealed more weaknesses. When propensity score stratification was performed to account for more observable covariates, the wide confidence intervals that covered 0 suggested there might not be a causal effect.

We tried to improve the author's method by conducting covariate balance check with propensity score matching. We found that the 95% CI for the causal effects contains 0, and two of the covariates remained unbalanced after matching. We decided to use doubly robust estimator to cope with covariate imbalance and with the potential misspecification of the

outcome model or the propensity score model. The doubly robust estimator for the causal effect on 401(k) saving is very similar to what we had from the replication of the author's main result. However, the causal effect on IRA saving is much smaller, and the 95% CI contains 0. Thus, we can only confidently say that 401(k) eligibility does have a positive causal effect on 401(k) saving, but not on other types of assets including IRA assets.

Future studies should consider accounting for variables that are likely to be inconsistent across time to rule out the effect from this type of potential confounding variables.

References

Alves, M. F. (n.d.). 12 - Doubly Robust Estimation. https://matheusfacure.github.io/python-causality-handbook/12-Doubly-Robust-Estimation.html

Elkins, K. (2017). A brief history of the 401(k), which changed how Americans retire. https://www.cnbc.com/2017/01/04/a-brief-history-of-the-401k-which-changed-how-americans-retire.html

Fernando, J. (2022). What is a 401(k) and how does it work? https://www.investopedia.com/terms/1/401kplan.asp

Gelber, A. M. (2011). How Do 401(k)s Affect Saving? Evidence from Changes in 401(k) Eligibility. *American Economic Journal: Economic Policy*, *3*(4), 103–122. https://doi.org/10.1257/pol.3.4.103

Gelber, A. M. (2019). *Replication data for: How Do 401(k)s Affect Saving? Evidence from Changes in 401(k) Eligibility* (No. V1). American Economic Association. https://doi.org/10.3886/E116534V1

Nguyen, T.-L., Collins, G. S., Spence, J., Daurès, J.-P., Devereaux, P. J., Landais, P., & Le Manach, Y. (2017). Double-adjustment in propensity score matching analysis: Choosing a threshold for considering residual imbalance. *BMC Medical Research Methodology*, *17*(1). https://doi.org/10.1186/s12874-017-0338-0

SIPP 1996 Panel Data. (2022). United States Census Bureau. https://www.census.gov/programs-surveys/sipp/data/datasets/1996-panel.html

Survey of Income and Program Participation (SIPP). (2017). National Bureau of Economic Research. https://www.nber.org/research/data/survey-income-and-program-participation-sipp

Appendix

This section includes all of the code we wrote for the replication project. All the code is in R. Note that all the code was run on Deepnote with its powerful GPU computer with 4 vCPUs, 60 GB of RAM, and 12 GB of K80 GPU memory.

```
# Importing the libraries we need
library(MASS)
library(dplyr)
library(stringr)
library(readr)
library(data.table)
library(lspline)
library(psych)
library(sandwich)
library(lmtest)
library(broom)
library(xtable)
library(Matching)
```

```
Attaching package: ‘MASS’


The following object is masked from ‘package:dplyr’:

    select



Attaching package: ‘data.table’


The following objects are masked from ‘package:dplyr’:

    between, first, last
```

Cleaning and Merging Data

This part of the appendix involved importing, cleaning, and merging raw data from NBER.

Also, note that a lot of comments came from the author's published `do` file, as we tried to replicate the author's data cleaning process (originally done in STATA) using R code.

```
# Functions for converting the dat and dct files into R data.frame objects

# All columns were converted to numeric form if possible, as we often saved the R.dataframes in
# csv form and not converting all columns to numeric type often led to data importing issues
# when trying to import these csv files back into R.

convert.to.numeric <- function(x){
    return(ifelse(is.na(as.numeric(x)), x, as.numeric(x)))
}
read.dat.dct <- function(dat, dct) {
    temp <- readLines(dct)
    temp <- temp[grepl("_column", temp)]
    colwidth_uncleaned <- unlist(lapply(temp, function(x){as.numeric(substr(str_extract(x, "%[0-9.]+[a-z]"), 2, nchar(str_extract(x, "%[0-9.]+[a-z]"))-1))}), use.names = FALSE)
    colnames <- unlist(lapply(temp, function(x){str_extract_all(x, "[a-z0-9]+", simplify=TRUE)[4]}), use.names = FALSE)
    division_factor <- unlist(lapply(colwidth_uncleaned, function(x){x%%1*10}), use.names = FALSE)
    colwidth <- unlist(lapply(colwidth_uncleaned, as.integer), use.names = FALSE)

    output_df <- read_fwf(dat, fwf_widths(colwidth, col_names = colnames))
    for(i in 1:ncol(output_df)){
        if(division_factor[i] != 0){
            col <- colnames(output_df)[i]
            output_df[ , col] <- sapply(output_df[ , col], as.numeric) * (10^(-1*division_factor[i]))
        }
    }
    output_df <- data.frame(lapply(output_df, convert.to.numeric))
    return(output_df)
}
```

```
# Import and sort t7
t7 <- read.dat.dct("/work/raw_data/raw_data/sipp96t7.dat", "/work/raw_data_dct/raw_data_dct/sip96t7.dct")
t7 <- t7[with(t7, order(ssuid, shhadid, epppnum)), ]
```

```
# Relevant variables used later are duplicated four times in w7 and w9 (once for each reference month),
# so for merging purposes keep only one of these four months;
w7 <- read.dat.dct("/work/raw_data/raw_data/sipp96l7.dat", "/work/raw_data_dct/raw_data_dct/sip96l7.dct")
w7v2 <- data.frame(w7)
w7v2 <- w7v2 %>% filter(srefmon == 4)
w7v2 <- w7v2[with(w7v2, order(ssuid, shhadid, epppnum)), ]
w9 <- read.dat.dct("/work/raw_data/raw_data/sipp96l9.dat", "/work/raw_data_dct/raw_data_dct/sip96l9.dct")
w9v2 <- data.frame(w9)
w9v2 <- w9v2 %>% filter(srefmon == 4)
w9v2 <- w9v2[with(w9v2, order(ssuid, shhadid, epppnum)), ]
```

```
# Rename variables so they are identified by their wave
for(t_num in c('3', '6', '9', '12')){
    t <- read.dat.dct(paste0("/work/raw_data/raw_data/sipp96t", t_num, ".dat"), paste0("/work/raw_data_dct/raw_data_dct/sip96t", t_num, ".dct"))
    old_col_names <- c('taltb', 'thhintbk', 'thhintot', 'thhotast', 'thhira', 'thhscdbt', 'rhhuscbt', 'rhhstk',
                       'tcarval1', 'tcarval2', 'tcarval3')
    new_col_names <- sapply(old_col_names, function(x){return(paste0(x, t_num))})
    t <- setnames(t, old = old_col_names, new = new_col_names)
    t <- t[with(t, order(ssuid, shhadid, epppnum)), ]
    new_t_file_path <- paste0("/work/raw_data/processed_raw_data/sipp96t", t_num, "v2.csv")
```

```
    fwrite(t, file=new_t_file_path)
}
```

```
# Keep relevant variables
t6 <- read_csv(file = "/work/raw_data/processed_raw_data/sipp96t6v2.csv")
t6 <- select(t6, starts_with(c('ssuid', 'shhadid', 'epppnum', 'tage', 'eeducate', 'thhintbk', 'thhintot', 'thhotast',
                               'rhhstk', 'taltb', 'thhira', 'thhscdbt', 'rhhuscbt', 'tcarval1', 'tcarval2', 'tcarval3')))
t6 <- data.frame(lapply(t6, convert.to.numeric))
t6 <- t6[with(t6, order(ssuid, shhadid, epppnum)), ]
```

```
# Merge in relevant variables from other waves - merging wave 3, 6, 9, 12, but not 7
for(t_num in c('3', '9', '12')){
    t_file_path <- paste0("/work/raw_data/processed_raw_data/sipp96t", t_num, "v2.csv")
    t <- read_csv(t_file_path)
    t <- data.frame(lapply(t, convert.to.numeric))
    old_col_names <- c('taltb', 'thhintbk', 'thhintot', 'thhotast', 'thhira', 'thhscdbt', 'rhhuscbt', 'rhhstk',
                       'tcarval1', 'tcarval2', 'tcarval3')
    new_col_names <- sapply(old_col_names, function(x){return(paste0(x, t_num))}, USE.NAMES = FALSE)
    t <- select(t, all_of(c(new_col_names, c('ssuid', 'shhadid', 'epppnum'))))
    t6 <- merge(t6, t, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
    t6 <- t6[with(t6, order(ssuid, shhadid, epppnum)), ]
}
```

```
# Merge in relevant variables from t7 and w7
t7 <- select(t7, all_of(c('enoina03', 'enoinb03', 'epensnyn', 'etdeffen', 'e1taxdef', 'e2taxdef', 'e3taxdef',
                          'ssuid', 'shhadid', 'epppnum')))
t6 <- merge(t6, t7, by=c('ssuid', 'shhadid', 'epppnum'))
t6 <- t6[with(t6, order(ssuid, shhadid, epppnum)), ]

w7v2_filtered <- data.frame(w7v2)
w7v2_filtered <- w7v2_filtered %>% filter(srefmon == 4)
w7v2_filtered <- select(w7v2_filtered, all_of(c('tsjdate1', 'srotaton', 'wpfinwgt', 'eclwrk1', 'efnp', 'esex', 'tempall1',
                                               'ejbind1', 'ssuid', 'shhadid', 'epppnum')))
t6 <- merge(t6, w7v2_filtered, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
t6 <- t6[with(t6, order(ssuid, shhadid, epppnum)), ]
```

```
# Generate variable measuring whether individuals have been on their job one year or less
t6$yr1jb1 = sapply(1:nrow(t6), function(i){switch(t6$srotaton[i], t6$tsjdate1[i] > 19970299, t6$tsjdate1[i] > 19970399, t6$tsjdate1[i] > 19970499, t6$tsjdate1[i] > 19970599)})
```

```
# Generate variable measuring how many days individuals have been on their job
t6$year = as.integer(t6$tsjdate1/10000)
t6$mo = as.integer((t6$tsjdate1-t6$year*10000)/100)
t6$day = t6$tsjdate1-t6$year*10000-t6$mo*100
daysonjob_func <- function(i){
    if(t6$year[i] >= 1947 & t6$year[i] <= 1998){
        return(((1998-t6$year[i])*12+(t6$srotaton[i]+2)-t6$mo[i])*30 + (30-t6$day[i]))
    }
    else{
        return(NA)
    }
}
t6$daysonjob = sapply(1:nrow(t6), daysonjob_func)
```

```
# Generate variable measuring whether individuals have been on their job at most eight months by wave 7, for use in Table 5
t6$yr1v2 = sapply(1:nrow(t6), function(i){switch(t6$srotaton[i], t6$tsjdate1[i] > 19970699, t6$tsjdate1[i] > 19970799, t6$tsjdate1[i] > 19970899, t6$tsjdate1[i] > 19970999)})
```

```
# Keep if in a private, for-profit firm
# Keep if in the relevant age range
t6 <- t6 %>% filter(eclwrk1 == 1 & tage > 21 & tage < 65)
```

```
# General overall car value in each wave as the sum of the value of individuals' different cars
t6$tcarval3 = t6$tcarval13+t6$tcarval23+t6$tcarval33
t6$tcarval6 = t6$tcarval16+t6$tcarval26+t6$tcarval36
t6$tcarval9 = t6$tcarval19+t6$tcarval29+t6$tcarval39
t6$tcarval12 = t6$tcarval112+t6$tcarval212+t6$tcarval312
```

```
# Generate other financial assets variable for each wave;
t6$otherassets3 = t6$thhintbk3+t6$thhintot3+t6$rhhstk3+t6$thhotast3
t6$otherassets6 = t6$thhintbk6+t6$thhintot6+t6$rhhstk6+t6$thhotast6
t6$otherassets9 = t6$thhintbk9+t6$thhintot9+t6$rhhstk9+t6$thhotast9
t6$otherassets12 = t6$thhintbk12+t6$thhintot12+t6$rhhstk12+t6$thhotast12
```

```
# Done with job date variable: drop it in order to merge in a different job date variable;
t6 <- subset(t6, select = -c(tsjdate1))
t6 <- t6[with(t6, order(ssuid, shhadid, epppnum)), ]
```

```
# Merge in variable measuring when they started their job in wave 9,
# in order to determine whether they stayed on the same job from Year 1 to Year 2;
w9_for_merge <- data.frame(w9v2)
w9_for_merge <- select(w9, all_of(c('tsjdate1', 'ssuid', 'shhadid', 'epppnum')))
t6 <- merge(t6, w9_for_merge, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
yr2jbdate_col <- sapply(1:nrow(t6), function(i){switch(t6$srotaton[i], t6$tsjdate1[i] > 19980299, t6$tsjdate1[i] > 19980399, t6$tsjdate1[i] > 19980499, t6$tsjdate1[i] > 19980599)})
t6$yr2jbdate <- replace(yr2jbdate_col, yr2jbdate_col == 'NULL', NA) %>% unlist()
```

```
# Generate dummy variables for education categories
sorted_unique_eeducate <- sort(unique(t6$eeducate))
for(i in 1:(length(unique(t6$eeducate))-1)){
    t6[ , paste0("educ", i)] <- (t6$eeducate == sorted_unique_eeducate[i])
}
t6[ , paste0("educ", length(unique(t6$eeducate)))] <- is.na(t6$eeducate)
```

```
# Generate variable "temp" measuring whether the individual is
# temporarily ineligible for their 401(k);
```

```
# "temp" is the main treatment variable;
t6$temp <- (t6$enoina03==1&t6$etdeffen==1)|(t6$enoinb03==1)
```

```
# Generate transformations of the basic variables
# to be used in regressions later;
for(var in c('taltb', 'thhira', 'otherassets', 'thhscdbt', 'rhhuscbt', 'tcarval')){
    var3 <- t6[, paste0(var, 3)]
    var6 <- t6[, paste0(var, 6)]
    var9 <- t6[, paste0(var, 9)]
    var12 <- t6[, paste0(var, 12)]
    t6[ , paste0("d21ihs", var)] <- log(var12+sqrt(var12^2+1))-2*log(var9+sqrt(var9^2+1))+log(var6+sqrt(var6^2+1))
    t6[ , paste0("d21l", var)] <- log(var12+10) - 2*(log(var9+10)) + log(var6+10)
    t6[ , paste0("d10l", var)] <- log(var9+10) - 2*(log(var6+10)) + log(var3+10)
    t6[ , paste0("temp", var)] <- t6$temp * var6
    spline_df <- as.data.frame(qlspline(var6, 20, na.rm = TRUE))
    old_col_names <- colnames(spline_df)
    new_col_names <- sapply(old_col_names, function(x){paste0('spl', var, x)}, USE.NAMES = FALSE)
    setnames(spline_df, old = old_col_names, new = new_col_names)
    t6 <- cbind(t6, spline_df)
    t6[ , paste0("d21l", var, "w")] <- winsor(t6[ , paste0("d21l", var)], trim = 0.05)
    t6[ , paste0("l", var, 9)] <- log(var9+10)
    t6[ , paste0("l", var, 6)] <- log(var6+10)
}
```

```
t6 <- t6[with(t6, order(ssuid, shhadid, epppnum)), ]
# Duplicate rows were created in this process, so we removed duplicate rows here
t6 <- t6[!duplicated(t6), ]
fwrite(t6, file="/work/raw_data/processed_raw_data/main.csv")
```

```
# Add together household income in each month of Year 1 to generate overall Year 1 income;
# first add together income in each month of Wave 7;
for(i in 1:4){
    w7i <- data.frame(w7)
    w7i <- w7i %>% filter(srefmon == i)
    setnames(w7i, old = "thtotinc", new = paste0("thtotinc7",i))
    fwrite(w7i, file=paste0("/work/raw_data/processed_raw_data/w7",i,".csv"))
}
```

```
# Merge data from each month of Wave 7;
w74 <- read_csv("/work/raw_data/processed_raw_data/w74.csv")
for(i in 1:3){
    using_file_path <- paste0("/work/raw_data/processed_raw_data/w7",i,".csv")
    w7i <- read_csv(file = using_file_path)
    w7i <- select(w7i, c('ssuid', 'shhadid', 'epppnum', paste0('thtotinc7', i)))
    w74 <- merge(w74, w7i, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
}
```

```
# Generate overall Wave 7 income;
w74$thtotinc7 = w74$thtotinc71 + w74$thtotinc72 + w74$thtotinc73 + w74$thtotinc74
```

```
fwrite(w74, file="/work/raw_data/processed_raw_data/w7totinc.csv")
```

```
# Add together income in each month of Wave 8
w8 <- read.dat.dct("/work/raw_data/raw_data/sipp96l8.dat", "/work/raw_data_dct/raw_data_dct/sip96l8.dct")
for(i in 1:4){
    w8i <- data.frame(w8)
    w8i <- w8i %>% filter(srefmon == i)
    setnames(w8i, old = "thtotinc", new = paste0("thtotinc8",i))
    fwrite(w8i, file=paste0("/work/raw_data/processed_raw_data/w8",i,".csv"))
}
```

```
# Merge data from each month of Wave 8;
w84 <- read_csv("/work/raw_data/processed_raw_data/w84.csv")
for(i in 1:3){
    using_file_path <- paste0("/work/raw_data/processed_raw_data/w8",i,".csv")
    w8i <- read_csv(file = using_file_path)
    w8i <- select(w8i, c('ssuid', 'shhadid', 'epppnum', paste0('thtotinc8', i)))
    w84 <- merge(w84, w8i, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
}
```

```
# Generate overall Wave 8 income
w84$thtotinc8 = w84$thtotinc81 + w84$thtotinc82 + w84$thtotinc83 + w84$thtotinc84
```

```
fwrite(w84, file="/work/raw_data/processed_raw_data/w8totinc.csv")
```

```
# Add together income in each month of Wave 9
for(i in 1:4){
    w9i <- data.frame(w9)
    w9i <- w9i %>% filter(srefmon == i)
    setnames(w9i, old = "thtotinc", new = paste0("thtotinc9",i))
    fwrite(w9i, file=paste0("/work/raw_data/processed_raw_data/w9",i,".csv"))
}
w94 <- read_csv("/work/raw_data/processed_raw_data/w94.csv")
for(i in 1:3){
    using_file_path <- paste0("/work/raw_data/processed_raw_data/w9",i,".csv")
    w9i <- read_csv(file = using_file_path)
    w9i <- select(w9i, c('ssuid', 'shhadid', 'epppnum', paste0('thtotinc9', i)))
    w94 <- merge(w94, w9i, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
}
# Generate overall Wave 9 income
w94$thtotinc9 = w94$thtotinc91 + w94$thtotinc92 + w94$thtotinc93 + w94$thtotinc94
```

```
# Merge in overall Wave 7 and Wave 8 income
w7totinc_selected <- select(w74, c('ssuid', 'shhadid', 'epppnum', 'thtotinc7'))
w94 <- merge(w94, w7totinc_selected, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
w8totinc_selected <- select(w84, c('ssuid', 'shhadid', 'epppnum', 'thtotinc8'))
```

```
w94 <- merge(w94, w8totinc_selected, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
# Generate overall Year 1 income
w94$thtotincyr1 <- w94$thtotinc7 + w94$thtotinc8 + w94$thtotinc9
fwrite(w94, file="/work/raw_data/processed_raw_data/w9totinc.csv")
```

```
# Merge in Year 1 income
main <- read_csv("/work/raw_data/processed_raw_data/main.csv")
main <- main[with(main, order(ssuid, shhadid, epppnum)), ]
w9totinc_selected <- read_csv("/work/raw_data/processed_raw_data/w9totinc.csv")
w9totinc_selected <- select(w9totinc_selected, all_of(c('thtotincyr1', 'ssuid', 'shhadid', 'epppnum')))
main <- merge(main, w9totinc_selected, by=c('ssuid', 'shhadid', 'epppnum'), all=TRUE)
```

```
# Generate variable "y401k" measuring whether the individual is in a firm that offers a 401(k);
main$y401k <- main$temp | (main$e1taxdef==1) | (main$e2taxdef==1) | (main$e3taxdef==1) | ((main$etdeffen==1) & main$yr1jb1)
main <- main[with(main, order(ssuid, shhadid, epppnum)), ]
# Duplicate rows were created in this process, so let's remove duplicate rows
main <- main[!duplicated(main), ]
```

```
# Generate a household ID variable so that clustering can be performed at the household level;
# ssuid shhadid together uniquely identify households;
fwrite(main, file="/work/raw_data/processed_raw_data/temp.csv")
main <- main %>% group_by(ssuid, shhadid) %>% summarise(tage = mean(tage, na.rm = TRUE))
main <- main[with(main, order(ssuid, shhadid)), ]
main$hid = 1:nrow(main)
fwrite(main, file="/work/raw_data/processed_raw_data/hid.csv")
```

```
temp <- read.csv("/work/raw_data/processed_raw_data/temp.csv")
hid <- read.csv("/work/raw_data/processed_raw_data/hid.csv")
```

```
hid <- select(hid, all_of(c('ssuid', 'shhadid', 'hid')))
temp <- merge(temp, hid, by=c('ssuid', 'shhadid'), all=TRUE)
```

```
# Generate age squared
temp$tagesq <- temp$tage^2
# Generator 1-digit industry dummies
temp$industry1digit <- as.integer(temp$ejbind1/100)
# Generate dummy variables for industrial categories
sorted_unique_industry1digit <- sort(unique(temp$industry1digit))
for(i in 1:(length(unique(temp$industry1digit))-1)){
    temp[ , paste0("ind1dig", i)] <- (temp$industry1digit == sorted_unique_industry1digit[i])
}
temp[ , paste0("ind1dig", length(unique(temp$industry1digit)))] <- is.na(temp$industry1digit)
```

```
# Generate firm size dummies
sorted_unique_tempall1 <- sort(unique(temp$tempall1))
for(i in 1:(length(unique(temp$tempall1))-1)){
    temp[ , paste0("tempallnew", i)] <- (temp$tempall1 == sorted_unique_tempall1[i])
}
temp[ , paste0("tempallnew", length(unique(temp$tempall1)))] <- is.na(temp$tempall1)
```

```
# Create dummy for missing income
temp$incmissing <- is.na(temp$thtotincyr1)
```

```
# Set income equal to -1 if it is missing;
# These values will be "dummied out" of the regression using the incising variable;
temp$thtotincyr1 <- ifelse(is.na(temp$thtotincyr1), -1, temp$thtotincyr1)
```

```
# Generate dummy variable measuring whether d21ltaltb is non-missing;
temp$notmissing <- (is.na(temp$d21ltaltb) == FALSE)
```

```
fwrite(temp, file="/work/raw_data/processed_raw_data/mainall.csv")
```

```
# Create dataset main.csv, which is the same as mainall.csv but only including year 1 observations;
fwrite(temp %>% filter(yr1jb1), file="/work/raw_data/processed_raw_data/main.csv")
```

Analysis

Summary Statistics

```
# Importing main.csv
main <- read_csv("/work/raw_data/processed_raw_data/main.csv")
```

```
# The following lines create the summary statistics table
mean_var <- c('tage', 'thtotincyr1', 'taltb6', 'thhira6', 'otherassets6', 'thhscdbt6', 'rhhuscbt6', 'tcarval6')
main_filtered <- main %>% filter(y401k & notmissing) %>% dplyr::select(c(all_of(mean_var), 'wpfinwgt', 'temp'))
means_all <- c()
weighted_means_all <- c()
medians_all <- c()
iqr_all <- c()
sd_all <- c()
for(x in mean_var){
    weighted_means_all <- append(weighted_means_all, weighted.mean(main_filtered[[x]], main_filtered$wpfinwgt))
    means_all <- append(means_all, mean(main_filtered[[x]]))
    medians_all <- append(medians_all, median(main_filtered[[x]]))
    iqr_all <- append(iqr_all, IQR(main_filtered[[x]]))
    sd_all <- append(sd_all, sd(main_filtered[[x]]))
}
summary.table <- data.frame(
    names <- c("Age", "Yearly Household Income", "401(k)", "IRA", "Other Assets", "Secured Debt", "Unsecured Debt", "Car Value"),
    weighted_means <- round(weighted_means_all, 1),
```

```
    means <- round(means_all, 1),
    medians <- round(medians_all, 1),
    iqrs <- round(iqr_all, 1),
    sds <- round(sd_all, 1)
)
colnames(summary.table) <- c("Name", "Weighted Mean", "Mean", "Median", "IQR", "SD")
summary.table
```

```
# The following line creates the LaTeX code for the summary statistics table
# Note that other tables were manually written into LaTeX;
# summary statistics table was the only table whose LaTeX code was produced by R
# due to the sheer size of the table
print(xtable(summary.table, type="latex"))
```

Replication of the Main Results

```
# "become eligible" without controls
var_list <- c('taltb', 'thhira', 'otherassets', 'thhscdbt', 'rhhuscbt', 'tcarval')
main_filtered <- main %>% filter(y401k)
result_table <- data.frame()
for(var in var_list){
    ols_formula <- as.formula(paste0('d21l', var, ' ~ temp'))
    print(var)
    model <- lm(ols_formula, data = main_filtered, weights = main_filtered$wpfinwgt)
    model_robust_clustered <- coeftest(model, vcov = vcovCL, type = "HC1", cluster = ~hid)
    result_table <- rbind(result_table, tidy(model_robust_clustered, conf.int = TRUE)[2, -c(4)])
}
result_table$term <- paste(var_list, "(without controls)")
print(result_table)
```

```
# "become eligible" with controls
var_list <- c('taltb', 'thhira', 'otherassets', 'thhscdbt', 'rhhuscbt', 'tcarval')
x_cols <- c('temp', 'tage', 'tagesq', 'thtotincyr1', 'incmissing', 'daysonjob')
for(pfx in c("educ", "tempallnew", "ind1dig")){
    x_cols <- append(x_cols, colnames(main)[startsWith(colnames(main), pfx)])
}
new_result_table <- data.frame()
main_filtered <- main %>% filter(y401k)
for(var in var_list){
    ols_formula <- as.formula(paste0('d21l', var, ' ~ ', paste(x_cols, collapse=" + ")))
    print(var)
    model <- lm(ols_formula, data = main_filtered, weights = main_filtered$wpfinwgt)
    model_robust_clustered <- coeftest(model, vcov = vcovCL, type = "HC1", cluster = ~hid)
    new_result_table <- rbind(new_result_table, tidy(model_robust_clustered, conf.int = TRUE)[2, -c(4)])
}
new_result_table$term <- paste(var_list, "(with covariates)")
result_table <- rbind(result_table, new_result_table)
print(result_table)
```

```
# "become eligible" with controls and initial assets
var_list <- c('taltb', 'thhira', 'otherassets', 'thhscdbt', 'rhhuscbt', 'tcarval')
new_result_table <- data.frame()
x_cols <- append(x_cols, colnames(main)[grepl("^l.+6$", colnames(main))])
for(var in var_list){
    ols_formula <- as.formula(paste0('d21l', var, ' ~ ', paste(x_cols, collapse=" + ")))
    print(var)
    model <- lm(ols_formula, data = main_filtered, weights = main_filtered$wpfinwgt)
    model_robust_clustered <- coeftest(model, vcov = vcovCL, type = "HC1", cluster = ~hid)
    new_result_table <- rbind(new_result_table, tidy(model_robust_clustered, conf.int = TRUE)[2, -c(4)])
}
new_result_table$term <- paste(var_list, "(with covariates and initial assets)")
result_table <- rbind(result_table, new_result_table)
print(result_table)
```

Robustness Checks

```
var_list <- c('taltb', 'thhira', 'otherassets', 'thhscdbt', 'rhhuscbt', 'tcarval')
main_filtered <- main %>% filter(y401k)
x_cols <- c('temp', 'tage', 'tagesq', 'thtotincyr1', 'incmissing', 'daysonjob')
for(pfx in c("educ", "tempallnew", "ind1dig")){
    x_cols <- append(x_cols, colnames(main)[startsWith(colnames(main), pfx)])
}
```

```
# 1st robustness check
result_table <- data.frame()
for(var in var_list){
    ols_formula <- as.formula(paste0('d21l', var, 'w ~ ', paste(x_cols, collapse=" + ")))
    print(var)
    model <- lm(ols_formula, data = main_filtered, weights = main_filtered$wpfinwgt)
    model_robust_clustered <- coeftest(model, vcov = vcovCL, type = "HC1", cluster = ~hid)
    result_table <- rbind(result_table, tidy(model_robust_clustered, conf.int = TRUE)[2, -c(4)])
}
result_table$term <- paste(var_list, "(winsorize outliers)")
print(result_table)
```

```
# 2nd robustness check
Neyman_SRE = function(z, y, x)
{
  xlevels = unique(x)
  K       = length(xlevels)
  PiK     = rep(0, K)
  TauK    = rep(0, K)
  varK    = rep(0, K)
  for(k in 1:K)
  {
    xk         = xlevels[k]
    zk         = z[x == xk]
    yk         = y[x == xk]
    PiK[k]     = length(zk)/length(z)
    TauK[k]    = mean(yk[zk==1]) - mean(yk[zk==0])
    varK[k]    = var(yk[zk==1])/sum(zk) +
      var(yk[zk==0])/sum(1 - zk)
```

```
  }

  return(c(sum(PiK*TauK), sqrt(sum(PiK^2*varK))))
}
main_filtered <- main %>% filter(y401k)
main_filtered <- main_filtered[complete.cases(main_filtered), ]
x_cols <- c('taltb3', 'thhira3', 'otherassets3', 'tcarval3', 'thhscdbt3', 'rhhuscbt3', 'tage', 'taltb6', 'thhira6',
            'otherassets6', 'tcarval6', 'thhscdbt6', 'rhhuscbt6', 'efnp', 'esex', 'thtotincyr1', 'incmissing')
for(pfx in c("educ", "tempallnew", "ind1dig")){
    x_cols <- append(x_cols, colnames(main_filtered)[startsWith(colnames(main_filtered), pfx)])
}
z <- main_filtered$temp
glm_formula <- as.formula(paste("temp", "~", paste(x_cols, collapse=" + ")))
pscore = glm(glm_formula, data = main_filtered, family = binomial)$fitted.values
for(var in var_list){
    y <- main_filtered[ , paste0('d21l', var)]
    n.strata = c(5)
    strat.res = sapply(n.strata, FUN = function(nn){
        q.pscore = quantile(pscore, (1:(nn-1))/nn)
        ps.strata = cut(pscore, breaks = c(0,q.pscore,1),
                        labels = 1:nn)
        Neyman_SRE(z, y, ps.strata)})
    rownames(strat.res) = c("est", "se")
    colnames(strat.res) = n.strata
    print(var)
    print(round(strat.res, 2))
}
```

```
# 3rd robustness check
x_cols <- c('temp', 'tage', 'tagesq', 'thtotincyr1', 'incmissing', 'daysonjob')
for(pfx in c("educ", "tempallnew", "ind1dig")){
    x_cols <- append(x_cols, colnames(main)[startsWith(colnames(main), pfx)])
}
for(var in var_list){
    ols_formula <- as.formula(paste0('l', var, '9 ~ l', var, '6 + ', paste(x_cols, collapse=" + ")))
    print(var)
    model <- lm(ols_formula, data = main_filtered, weights = main_filtered$wpfinwgt)
    model_robust_clustered <- coeftest(model, vcov = vcovCL, type = "HC1", cluster = ~hid)
    print(tidy(model_robust_clustered, conf.int = TRUE))
}
```

Re-Analysis

Propensity Score Matching

```
main_filtered <- main %>% filter(y401k)
main_filtered <- main_filtered[complete.cases(main_filtered), ]
z <- main_filtered$temp
for(var in var_list){
    y <- unlist(main_filtered[ , paste0('d21l', var)])
    x_cols <- c('taltb3', 'thhira3', 'otherassets3', 'tcarval3', 'thhscdbt3', 'rhhuscbt3', 'tage', 'taltb6', 'thhira6',
                'otherassets6', 'tcarval6', 'thhscdbt6', 'rhhuscbt6', 'efnp', 'esex', 'thtotincyr1', 'incmissing')
    for(pfx in c("educ", "tempallnew", "ind1dig")){
        x_cols <- append(x_cols, colnames(main_filtered)[startsWith(colnames(main_filtered), pfx)])
    }
    x <- as.matrix(main_filtered %>% dplyr::select(x_cols))
    print(var)
    matchest = Match(Y=y, Tr=z, X=x)
    print(summary(matchest))
    lm.before = lm(z ~ x)
    print(summary(lm.before))
    lm.after = lm(z ~ x, subset = c(matchest$index.treated, matchest$index.control))
    print(summary(lm.after))
    }
```

Doubly Robust Estimator

```
# Defining necessary functions for calculating the doubly robust estimator
OS_est = function(z, y, x, out.family = gaussian,
                  truncpscore = c(0, 1))
{
     ## fitted propensity score
     pscore   = glm(z ~ x, family = binomial)$fitted.values
     pscore   = pmax(truncpscore[1], pmin(truncpscore[2], pscore))

     ## fitted potential outcomes
     outcome1 = glm(y ~ x, weights = z,
                    family = out.family)$fitted.values
     outcome0 = glm(y ~ x, weights = (1 - z),
                    family = out.family)$fitted.values

     ## regression imputation estimator
     ace.reg  = mean(outcome1 - outcome0)
     ## IPW estimators
     ace.ipw0 = mean(z*y/pscore - (1 - z)*y/(1 - pscore))
     ace.ipw  = mean(z*y/pscore)/mean(z/pscore) -
                   mean((1 - z)*y/(1 - pscore))/mean((1 - z)/(1 - pscore))
     ## doubly robust estimator
     res1     = y - outcome1
     res0     = y - outcome0
     ace.dr   = ace.reg + mean(z*res1/pscore - (1 - z)*res0/(1 - pscore))

     return(c(ace.reg, ace.ipw0, ace.ipw, ace.dr))
}


OS_ATE = function(z, y, x, n.boot = 2*10^2,
                     out.family = gaussian, truncpscore = c(0, 1))
{
     point.est  = OS_est(z, y, x, out.family, truncpscore)

     ## nonparametric bootstrap
     n.sample   = length(z)
     x          = as.matrix(x)
     boot.est   = replicate(n.boot,
                   {id.boot = sample(1:n.sample, n.sample, replace = TRUE)
                   OS_est(z[id.boot], y[id.boot], x[id.boot, ],
                          out.family, truncpscore)})
```

```
    boot.se    = apply(boot.est, 1, sd)

    res        = rbind(point.est, boot.se)
    rownames(res) = c("est", "se")
    colnames(res) = c("reg", "HT", "Hajek", "DR")

    return(res)
}
```

```
# Prepping data for doubly robust estimator analysis
main <- read.csv("AEJPol2009-0165_data-3/AEJPol2009-0165_data-3/new_csv_data/main.csv")
main_filtered <- main %>% filter(y401k&notmissing)

# Imputing missing values with -1
# and adding dummy variables representing the presence of missing values
x_cols <- c('taltb3', 'thhira3', 'otherassets3', 'tcarval3', 'thhscdbt3', 'rhhuscbt3', 'tage',
            'taltb6', 'thhira6', 'otherassets6', 'tcarval6', 'thhscdbt6', 'rhhuscbt6', 'efnp',
            'esex', 'thtotincyr1', 'incmissing')
for(pfx in c("educ", "tempallnew", "ind1dig")){
    x_cols <- append(x_cols, colnames(main_filtered)[startsWith(colnames(main_filtered), pfx)])
}
for(col in x_cols){
    if(sum(is.na(main_filtered[ , col])) > 0){
        x_cols <- append(x_cols, paste0(col, "missing"))
    }
}
for(col in colnames(main_filtered)){
    if(sum(is.na(main_filtered[ , col])) > 0){
        main_filtered[ , paste0(col, 'missing')] <- as.integer(is.na(main_filtered[ , col]))
        main_filtered[ , col] <- ifelse(is.na(main_filtered[ , col]), -1, main_filtered[ , col])
        print(paste("Missing column made for", col))
    }
}
```

```
# Calculating the doubly robust estimator values
var_list <- c('taltb', 'thhira', 'otherassets', 'thhscdbt', 'rhhuscbt', 'tcarval')
z = as.integer(main_filtered$temp)
x = as.matrix(main_filtered[, x_cols])

for(var in var_list){
    print(paste0('d211', var))
    y <- main_filtered[ , paste0('d211', var)]
    causaleffects = OS_ATE(z, y, x, n.boot = 10^2)
    print(round(causaleffects, 3)[ , 4])
    print(rbind(causaleffects[1, ] - qnorm(1-0.05/2)*causaleffects[2, ],
        causaleffects[1, ] + qnorm(1-0.05/2)*causaleffects[2, ])[ , 4])
}
```