Stat 156 Final Project

Keval Amin[a] and Stephanie Quiroz[b]

[a]University of California, Berkeley and Sciences Po Paris

[b]University of California, Berkeley

ABSTRACT

Assignment Description (NOT AN ABSTRACT): The goal of this assignment is to apply learned methods from this course to analyze real- world datasets and critically appraise causal claims made in academic publications. This project is a group assignment with each group consisting of no more than two students.

It is strongly recommended that students replicate and re-analyze the results of an academic paper whose original datasets or similar datasets are publicly available. Datasets provided by authors on the publication website are already cleaned and should match the authors' published results exactly, so please do not use the cleaned datasets on the publication website unless the paper is an experimental study. As a part of the replication exercise, you and your group should download the original dataset and clean it to approximately match the sample selection used in the published paper. For other forms of final project assignments, such as a literature review with simulation studies to compare multiple methods, please attend the GSI's office hour.

Our Video is Here:

https://drive.google.com/file/d/1Q1Z2s8VzEUjAv3gd6JEvfmk1xE4uzkoa/view?usp=drive_link

Keywords:

1. INTRODUCTION

Employer-sponsored retirement plans, primarily 401(k)s, are the foundation of household wealth accumulation in the United States. An unresolved question in public finance is whether these tax incentives actually stimulate *new* saving, or if people simply move money they have already accumulated into a 401(k) to secure a tax break.

Solving this question is particularly difficult because of selection bias; one cannot simply compare individuals with 401(k)s to those without them. This is because those who naturally enjoy saving money often seek out

Further author information: (Send correspondence to Keval D. Amin)

Keval D. Amin.: E-mail: keval.amin@berkeley.edu, Telephone: +447879802729

B.B.A.: E-mail: bba@cmp.com, Telephone: +33 (0)1 98 76 54 32

jobs that offer these retirement plans. If we observe that 401(k) holders possess more wealth, it is difficult to distinguish whether the plan *caused* this accumulation or if these individuals would have saved that money anyway.

To solve this problem, this project successfully replicates a paper by Alexander Gelber (2011), "How Do 401(k)s Affect Saving?". Gelber applies a clever strategy to remove any selection bias: he looks at employer eligibility rules. Many companies make employees wait 12 months before they can join the 401(k) plan. This allows us to compare two very similar workers: one who has been at a job for 12 months (without a plan), and one who has been there 12 months (has a plan). These workers are almost identical; any difference in their saving is likely caused by the 401(k) itself.

For our STAT 156 final project, we contributed to this research in three different ways:

1. **Data Reconstruction:** We rebuilt the dataset from scratch using raw government survey files (SIPP). We created three different versions of the data to assess whether small changes in processing would affect the final results.
2. **Replication:** We re-ran the author's original regression analysis to verify his main claim: that becoming eligible for a 401(k) increases total savings without crowding out other assets.
3. **New Analysis:** We applied causal inference methods learned in class that the original author did not implement. Specifically, we used Matching and Inverse Probability Weighting (IPW) to estimate the treatment effect. We also ran a Rosenbaum sensitivity analysis to check if hidden bias (unobserved factors) could be distorting our results.

Our findings confirm the original paper's conclusion: becoming eligible for a 401(k) leads to higher total savings. However, our new sensitivity analysis suggests that while the positive effect is real, the exact size of the effect is sensitive to unobserved factors and should be interpreted carefully.

2. PAPER SUMMARY AND SUMMARY STATISTICS TABLE

2.1 Summarise the paper's research question and its answer

- **Research Question**: How do 401(k)s Affect Saving?

  This paper investigates the effect of 401(k) eligibility on saving. To address the possibility that eligibility correlates across individuals with their unobserved tastes for saving, we examine a change in eligibility: some individuals are initially ineligible for their 401(k) but become eligible when they have worked at their firm long enough.

- **Conclusions**: It finds that eligibility raises 401(k) balances. Other financial assets and net worth respond insignificantly to eligibility, but the confidence intervals do not rule out substantial responses. In response to eligibility, IRA assets increase, consistent with a "crowd-in" hypothesis, and accumulation of cars decreases.

2.2 Describe the datasets used in answering the question

We consider three approaches to sample construction:

- **Raw**: we use the sample we reconstruct directly from the raw SIPP files, clean the data ourselves, and try and use our own version of the eligibility flags

- **Raw Aligned**: we instead use the authors' constructed eligibility indicators (`yr1jb1`, `temp`, and `y401k`) from the replication files, which allows us to match their reported sample size almost exactly (835 observations, with at most ± 2 differences across treatment and control groups), and then carry out the replication and re-analysis using this author-aligned dataset. This is the "best" sample and used after consultation with instructor.

- **Replication**: The third dataset is constructed to mirror using the same functions but on the data given in the replication package published by the authors on the American Economic Association Website.

2.3 Clean the dataset

Our dataset construction follows five sequential steps. Below, we describe each step and indicate the specific `.qmd` files in which the step is implemented, as well as the tables that rely on the resulting objects.

(1) Merging Waves. We merge person-level records from SIPP Waves 3, 6, 9, and 12 (Topical Module asset files) with Wave 7 Core demographic and employment records. This ensures that each individual has consistent identifiers and asset information at months 6, 9, and 12. *Implemented in:* `table1.qmd`, `table1_compare.qmd`, and internally through `build_dat()`. *Used in:* Table 1, Table 2, Table 3, Matching, DR, and Rosenbaum analyses.

(2) Sample Restrictions. We restrict the sample to individuals aged 22–64 who, in Wave 7, report being in their first year at a firm that offers a 401(k). We drop observations with missing covariates or missing outcomes and enforce the age, employment, and eligibility criteria used in the published study. *Implemented in:* `table1.qmd`, `table1_compare.qmd`, `make_ipw_sample3()`, and `make_reg_sample()`. *Used in:* Table 1 comparison, Table 2 regressions, Table 3 panels, and all re-analysis datasets (Raw, Raw Aligned, Replication).

(3) Outcome Construction. We construct the asset components (401(k), IRA, other financial assets, secured debt, unsecured debt, and car value) at each relevant month and compute their log changes. For total financial assets, we use the same aggregation as in the original paper. In the re-analysis, we additionally compute the IPW-style saving outcome via `make_ipw_outcome3()`. *Implemented in:* `table1.qmd`, `table1_compare.qmd`, `make_assets_wave()`, and `make_ipw_outcome3()`. *Used in:* Table 1 asset summaries, Table 2 outcome regressions, Table 3 components, and Matching/DR/Rosenbaum outcomes.

(4) Covariate Construction. We reconstruct the covariates used in the published study, including age and age squared, household income, education categories, firm size categories, industry, and days on job. We also construct an indicator for missing income, as in the original paper. *Implemented in:* `table1.qmd`, `table1_compare.qmd`, and internal helper functions (`normalise_ids()`, `make_table1()`, and covariate-cleaning functions). *Used in:* All regressions for Table 2 and Table 3, as well as the propensity score models for Matching and DR estimators.

(5) Weighting. We use the person-level SIPP final weights (`wpfinwgt`) in all regression-based replications to match the original survey-weighted analysis. The re-analysis estimators (Matching, DR, and Rosenbaum) do not apply survey weights, as they target the ATT for the analytic sample rather than a population-weighted ATE. *Implemented in:* `table2_raw.qmd`, `table2_raw_aligned.qmd`, `table2_rep.qmd`, and the Table 3 robustness scripts. *Used in:* Table 2 regressions, Table 3 (Panels A–C). Matching, DR, and Rosenbaum analyses use unweighted design-based estimators.

2.4 Replicate and interpret a summary statistics table that presents distributional characteristics (mean, median, IQR, etc) of key variables and covariates used in the empirical analysis.

Together, these steps define the three analytic datasets used throughout the project (Raw, Raw Aligned, and Replication) and ensure comparability with the published results of Gelber (2011). See table 1

Table 1. Summary Statistics

group	Obs	Age	HH Income	401(k)	IRA/Keogh	Other Fin.	Sec. Debt	Unsec. Debt	Car Value
raw	1115	37.0 (9.9)	60657.5 (37334.3)	5896.9 (22541.0)	7325.3 (25794.2)	31893.0 (150907.0)	61114.0 (74258.4)	6818.9 (14275.8)	11974.0 (9213.4)
raw	391	36.2 (9.6)	57012.5 (36748.6)	3490.0 (16045.8)	7646.0 (28460.7)	17545.2 (55786.5)	58427.7 (69450.8)	6362.9 (12336.5)	11256.6 (8749.6)
raw	724	37.4 (10.0)	62680.7 (37502.8)	7220.9 (25318.6)	7148.9 (24200.6)	39785.9 (182792.1)	62591.8 (76734.8)	7069.7 (15231.9)	12368.7 (9435.7)
raw_aligned	836	37.0 (9.9)	60957.2 (38676.3)	6006.8 (22121.4)	7751.2 (26845.9)	36317.7 (170578.5)	61634.1 (77042.1)	6742.4 (13753.5)	11866.0 (9284.1)
raw_aligned	298	36.0 (9.4)	57282.2 (37862.7)	4105.9 (17997.9)	7372.3 (28021.0)	17967.6 (60691.8)	57567.5 (69791.3)	6567.9 (13096.2)	11027.6 (8597.6)
raw_aligned	538	37.6 (10.1)	63077.6 (38980.7)	7087.3 (24086.7)	7966.5 (26152.0)	46748.2 (207948.1)	63945.7 (80783.2)	6841.6 (14112.6)	12342.6 (9620.1)
replication	841	37.0 (9.9)	60338.6 (38345.1)	5994.6 (22170.0)	7769.8 (27011.0)	36467.5 (170599.7)	61827.5 (76976.9)	6791.1 (13804.8)	11835.2 (9260.4)
replication	298	35.9 (9.4)	56476.4 (37469.2)	4101.9 (17985.0)	7409.0 (28175.5)	17942.3 (60733.7)	57739.2 (69887.6)	6604.0 (13112.6)	10975.5 (8608.0)
replication	543	37.6 (10.1)	62539.7 (38663.3)	7055.7 (24136.9)	7972.1 (26333.4)	46852.9 (207490.5)	64119.4 (80588.5)	6896.1 (14177.0)	12317.2 (9573.1)

3. REPLICATE THE MAIN RESULTS

3.1 Describe the empirical method in identifying the causal effect (for instance, whether the researchers conduct a randomised experiment or use policy changes to answer their research questions) and state

Formally, the identification strategy assumes that, conditional on observed covariates (age, education, income, firm size, industry, and days on job), the timing of becoming 401(k)-eligible is independent of potential outcomes. This is an instance of a *quasi-randomised natural experiment*: treatment assignment (eligibility) is not randomised, but its timing is driven by survey mechanics rather than by individual saving decisions or employer behavior.

The paper implements this strategy through the regression specification replicated in Table 2:

$$Y_i = \alpha + \tau \, \mathrm{temp}_i + f(\mathrm{age}_i) + X_i'\beta + \varepsilon_i,$$

where Y_i is the log-change in financial assets, temp_i is an indicator for being newly eligible for a 401(k), $f(\mathrm{age}_i)$ includes age and age squared, and X_i contains income, education, firm size, industry, and tenure controls. The coefficient τ is interpreted as the causal effect of 401(k) eligibility on saving. It's a within person difference approach and they define the outcome as a second difference in (log) assets:

$$Y_i = \left[\log(A_{i,12}+10) - \log(A_{i,9}+10)\right] - \left[\log(A_{i,9}+10) - \log(A_{i,6}+10)\right] = \log(A_{i,12}+10) - 2\log(A_{i,9}+10) + \log(A_{i,6}+10).$$

The first bracket is asset growth from Wave 9 to 12 ("Year 2") and the second bracket is growth from Wave 6 to 9 ("Year 1"), so Y_i measures how the household's rate of accumulation changes from Year 1 to Year 2. Regressing Y_i on the "Become eligible" indicator estimates whether newly eligible workers experience a relative upward shift in saving compared with controls, while differencing removes time-invariant level differences in saving behavior.

3.2 Replicate the main result of the paper and interpret it in English

Here is the table 2 equivalent which contains OLS regressions. Table 2 reports OLS estimates of the effect of becoming eligible for a 401(k) on changes in household asset components across three datasets and specifications. The results closely replicate the paper's main findings. Eligibility leads to a large and statistically significant increase in 401(k) assets across all samples and specifications, confirming that access to employer-sponsored retirement plans substantially raises retirement saving. In contrast, estimates for other financial assets and secured and unsecured debt are generally small and statistically insignificant, although wide confidence intervals imply that economically meaningful effects cannot be ruled out. IRA assets increase following eligibility, consistent with a "crowd-in" hypothesis rather than substitution away from other tax-advantaged saving. Finally, car asset accumulation declines after eligibility, suggesting a reallocation of household resources toward retirement saving. Overall, the replication reproduces the qualitative conclusions of the original study, with quantitative differences across samples reflecting differences in data construction rather than substantive changes in the economic patterns.

Table 2. Regression Results (Coefficients and Standard Errors)

Outcome	Dataset	Controls (Panel A)	Controls + Lag (Panel B)	No Controls (Panel C)
401k assets	Original (Gelber)	0.95 (0.29)	0.93 (0.29)	1.02 (0.29)
	Raw aligned	0.975 (0.287)	1.084 (0.288)	0.960 (0.287)
	Raw non-aligned	0.817 (0.256)	0.981 (0.254)	0.835 (0.253)
	Replication	0.952 (0.285)	1.050 (0.286)	0.933 (0.285)
IRA assets	Original (Gelber)	0.56 (0.26)	0.53 (0.25)	0.49 (0.25)
	Raw aligned	0.527 (0.264)	0.491 (0.255)	0.524 (0.260)
	Raw non-aligned	0.554 (0.219)	0.504 (0.212)	0.555 (0.218)
	Replication	0.503 (0.262)	0.469 (0.254)	0.553 (0.263)
Other assets	Original (Gelber)	-0.05 (0.29)	-0.08 (0.29)	-0.01 (0.28)
	Raw aligned	-0.114 (0.291)	-0.042 (0.280)	-0.085 (0.287)
	Raw non-aligned	-0.026 (0.247)	0.020 (0.237)	-0.007 (0.246)
	Replication	-0.094 (0.289)	-0.031 (0.278)	-0.054 (0.286)
Secured Debt	Original (Gelber)	0.10 (0.35)	0.14 (0.36)	0.15 (0.35)
	Raw aligned	0.041 (0.347)	0.081 (0.342)	0.096 (0.344)
	Raw non-aligned	-0.091 (0.290)	-0.046 (0.285)	-0.058 (0.289)
	Replication	0.042 (0.344)	0.077 (0.339)	0.117 (0.343)
Unsecured Debt	Original (Gelber)	-0.09 (0.40)	-0.15 (0.39)	-0.08 (0.37)
	Raw aligned	-0.174 (0.400)	-0.096 (0.378)	-0.112 (0.394)
	Raw non-aligned	-0.125 (0.353)	-0.066 (0.335)	-0.089 (0.351)
	Replication	-0.137 (0.398)	-0.069 (0.377)	-0.084 (0.393)
Car Value	Original (Gelber)	-0.50 (0.29)	-0.58 (0.29)	-0.47 (0.28)
	Raw aligned	-0.379 (0.296)	-0.264 (0.282)	-0.426 (0.288)
	Raw non-aligned	-0.296 (0.246)	-0.215 (0.236)	-0.320 (0.240)
	Replication	-0.425 (0.298)	-0.303 (0.284)	-0.496 (0.291)

Table 3. R^2 Values by Outcome, Dataset, and Specification

Outcome	Dataset	Panel A: No controls	Panel B: Controls	Panel C: Controls + Lag
401k assets	Original (Gelber)	0.010	0.012	0.050
	Raw aligned	0.011	0.013	0.052
	Raw non-aligned	0.009	0.012	0.051
	Replication	0.010	0.013	0.055
IRA assets	Original (Gelber)	0.006	0.009	0.038
	Raw aligned	0.007	0.010	0.040
	Raw non-aligned	0.006	0.009	0.039
	Replication	0.006	0.009	0.041
Other assets	Original (Gelber)	0.000	0.004	0.068
	Raw aligned	0.001	0.005	0.070
	Raw non-aligned	0.000	0.004	0.069
	Replication	0.000	0.004	0.072
Secured debt	Original (Gelber)	0.000	0.001	0.051
	Raw aligned	0.000	0.002	0.053
	Raw non-aligned	0.000	0.001	0.052
	Replication	0.000	0.001	0.054
Unsecured debt	Original (Gelber)	0.000	0.002	0.085
	Raw aligned	0.000	0.002	0.088
	Raw non-aligned	0.000	0.002	0.087
	Replication	0.000	0.002	0.090
Car value	Original (Gelber)	0.002	0.007	0.064
	Raw aligned	0.002	0.008	0.066
	Raw non-aligned	0.002	0.007	0.065
	Replication	0.002	0.007	0.067

3.3 Critically appraise the stated assumptions for causal identification. For instance, if the paper is carrying out an experiment, consider whether the experiment is balanced or if it achieves the stated goal of the author. If the paper is using a policy change or another form of "natural experiments", consider whether there would be confounding factors

The identification methods of the paper rely on the assumption that the timing of 401(k) eligibility is random, conditional on observed covariates. While tenure-based eligibility rules and the SIPP rotation groups provide a valid source of quasi-random variation, this assumption can be violated in many ways.

Job start dates may correlate with seasonal or business-cycle factors that also affect saving behavior, creating a systematic difference between workers who become eligible earlier versus later. The timing of eligibility might also be related to unobserved employer characteristics, such as hiring practices, benefit generosity, or firm stability, that are not fully captured by industry and firm-size controls. In addition, eligibility is constructed from self-reported tenure, which can be measured with error and lead to misclassifications of eligibility status.

These concerns suggest that while the natural experiment does improve upon cross-sectional comparisons, the identifying assumptions are potentially fragile. As a result, the estimated effects should be interpreted as credible but not completely definitive, which motivates the use of robustness and sensitivity analyses.

4. REPLICATE ROBUSTNESS CHECKS/EXTENSIONS

We next replicate the robustness exercises presented in Gelber (2011) to assess the stability of the main regression estimates. Following the paper, we construct alternative outcome definitions for each asset category and re-estimate the causal effect under three samples. These correspond to Panels A–C of Table 3 in the original study. Our reconstructed datasets produce coefficient patterns that closely track the published results across all asset components, confirming that the paper's conclusions are not sensitive to functional-form choices or the inclusion of lagged outcomes. Minor quantitative differences arise from small sample-size discrepancies and unavoidable differences in how raw SIPP files are processed, but the overall robustness patterns are successfully replicated.

Table 4. Panel A Results; Controlling for 20-piece spline in initial balance

Sample Variable	Original	Aligned	Raw	Replication
401k Assets	1.02 (0.29)	1.049 (0.282)	0.936 (0.245)	1.015 (0.281)
R^2	0.09	0.077	0.074	0.076
IRA Assets	0.49 (0.25)	0.473 (0.241)	0.463 (0.205)	0.451 (0.240)
R^2	0.09	0.052	0.056	0.051
Other Assets	0.02 (0.29)	-0.012 (0.267)	0.036 (0.226)	-0.003 (0.267)
R^2	0.13	0.091	0.084	0.089
Secured Debt	0.07 (0.36)	0.041 (0.327)	-0.063 (0.273)	0.046 (0.326)
R^2	0.13	0.099	0.106	0.099
Unsecured Debt	-0.08 (0.37)	-0.033 (0.369)	-0.031 (0.321)	-0.001 (0.369)
R^2	0.15	0.099	0.103	0.099
Car Value	-0.48 (0.28)	-0.252 (0.258)	-0.234 (0.217)	-0.308 (0.259)
R^2	0.14	0.098	0.086	0.096

Table 5. Panel B Results; Interacting initial balance with treatment

Sample Variable	Original	Aligned	Raw	Replication
401k Assets	1.07 (0.29)	1.108 (0.292)	0.886 (0.254)	1.080 (0.291)
R^2	0.07	0.051	0.040	0.051
IRA Assets	0.45 (0.25)	0.437 (0.251)	0.501 (0.215)	0.410 (0.251)
R^2	0.06	0.016	0.016	0.015
Other Assets	-0.10 (0.30)	-0.167 (0.282)	-0.098 (0.240)	-0.147 (0.281)
R^2	0.06	0.019	0.015	0.020
Secured Debt	0.25 (0.52)	0.249 (0.432)	-0.008 (0.367)	0.232 (0.431)
R^2	0.04	0.007	0.007	0.007
Unsecured Debt	-0.40 (0.46)	-0.434 (0.421)	-0.429 (0.368)	-0.394 (0.421)
R^2	0.07	0.024	0.024	0.023
Car Value	-1.07 (0.51)	-0.877 (0.419)	-0.449 (0.359)	-0.925 (0.420)
R^2	0.08	0.036	0.026	0.037

Table 6. Panel C Results; Inverse Hyperbolic Sine Transformation

Sample Variable	Original	Aligned	Raw	Replication
401k Assets (IHS)	1.29 (0.42)	1.083 (0.284)	0.981 (0.246)	1.050 (0.283)
R^2	0.04	0.052	0.055	0.050
IRA Assets (IHS)	0.73 (0.36)	0.491 (0.240)	0.503 (0.205)	0.469 (0.240)
R^2	0.05	0.040	0.041	0.039
Other Assets (IHS)	0.03 (0.40)	-0.009 (0.270)	0.046 (0.227)	0.001 (0.269)
R^2	0.05	0.069	0.067	0.069
Secured Debt (IHS)	0.21 (0.50)	0.081 (0.328)	-0.046 (0.275)	0.077 (0.327)
R^2	0.04	0.049	0.054	0.050
Unsecured Debt (IHS)	-0.11 (0.56)	-0.097 (0.366)	-0.066 (0.318)	-0.069 (0.366)
R^2	0.06	0.081	0.089	0.081
Car Value (IHS)	-0.83 (0.41)	-0.263 (0.254)	-0.215 (0.216)	-0.303 (0.256)
R^2	0.06	0.076	0.067	0.076

5. RE-ANALYSE

This section re-examines the paper's main findings using alternative estimators to assess their robustness. These methods do not introduce new sources of causal identification; identification continues to rely on the plausibly exogenous timing of 401(k) eligibility induced by tenure rules. Instead, the reanalysis evaluates the sensitivity of the estimated ATT to functional-form assumptions, covariate balance, and potential unobserved confounding by applying matching, doubly robust estimation, and Rosenbaum sensitivity analysis. . We study the causal effect of 401(k) eligibility on household saving using the potential outcomes framework. For each individual i, let $Y_i(1)$ and $Y_i(0)$ denote potential outcomes under treatment ($Z_i = 1$) and control ($Z_i = 0$). The observed outcome is

$$Y_i = Z_i Y_i(1) + (1 - Z_i) Y_i(0),$$

and our parameter of interest is the *Average Treatment Effect on the Treated (ATT)*:

$$\tau_{\text{ATT}} = \mathbb{E}[\, Y_i(1) - Y_i(0) \mid Z_i = 1\,].$$

5.1 Nearest-Neighbor Matching (ATT)

We estimate the ATT using nearest-neighbor matching on the logit propensity score. Let $j(i)$ denote the control unit matched to treated unit i. The matching estimator is:

$$\widehat{\tau}_{\text{ATT}}^{\text{match}} = \frac{1}{N_1} \sum_{i:Z_i=1} \left(Y_i - Y_{j(i)} \right),$$

with N_1 the number of treated units. Standard errors are computed by bootstrap resampling of matched pairs.

Rationale: Matching directly balances covariates between treated and control units, producing a comparison that resembles a randomised experiment. It is design-based and aligns naturally with the ATT estimand.In this instance, matching is particularly appropriate because eligibility is driven by quasi-random SIPP group structure. Furthermore, Gelber uses propensity score matching as a robustness check, we just take it a step further. We match on demographic factors and job/firm characteristics, whereas Gerber's is on pre-treatment wealth. Together, we can see that whether effect is explained by labour market composition differences or baseline household wealth differences.

5.2 Doubly Robust AIPW Estimator (ATT)

To improve robustness, we combine propensity score weighting with outcome regression. Let

$$\mu_1(X_i) = \mathbb{E}[Y_i \mid Z_i = 1, X_i], \qquad \mu_0(X_i) = \mathbb{E}[Y_i \mid Z_i = 0, X_i].$$

The doubly robust estimator for the ATT is:

$$\widehat{\tau}_{\text{ATT}}^{DR} = \frac{1}{n_1} \sum_{i=1}^{n} \left[Z_i \{Y_i - \hat{\mu}_0(X_i)\} - (1 - Z_i) \frac{\hat{e}(X_i)}{1 - \hat{e}(X_i)} \{Y_i - \hat{\mu}_0(X_i)\} \right].$$

Doubly robust property: The estimator is consistent if *either* the propensity score model or the outcome regression model is correctly specified. Only one needs to be right.

Rationale: AIPW is based on the efficient influence-function form, so under ignorability/overlap and with sufficiently accurate nuisance estimation, it can attain the semiparametric efficiency bound; in finite samples its performance depends on overlap and model quality In practice, our DR estimates fall between Matching and OLS, aligning with the theoretical predictions of the course. It uses far more of our samples compared to matching. It's bias-corrected, compared to pure weighting or regression.

5.3 Rosenbaum Sensitivity Analysis (ATT)

Matching removes bias from observed covariates, but unobserved confounding may remain. Rosenbaum (2002) formalises how hidden bias could affect the ATT estimated from matched pairs. For matched units i and j,

$$\frac{1}{\Gamma} \leq \frac{\Pr(Z_i = 1 \mid X_i, u_i)}{\Pr(Z_j = 1 \mid X_j, u_j)} \leq \Gamma,$$

where $\Gamma \geq 1$ captures the magnitude of unobserved confounding. We compute bounds on the Hodges–Lehmann estimate for various Γ values.

Rationale: Rosenbaum bounds quantify how strongly an omitted variable would need to affect treatment to overturn the ATT estimate. This provides a transparent assessment of robustness to hidden bias and complements the design-based Matching and model-assisted DR estimators. Rosenbaum bounds are especially informative in this study because eligibility is not randomised (natural experiment, quasi randomised) and may depend on unobserved job or household factors.

5.4 Assumptions and critical evaluation of my methods

Interpreting the Matching and doubly robust ATT estimates causally relies on (i) conditional ignorability given the observed covariates and (ii) overlap/common support. We view these assumptions as reasonably plausible in our setting because the analysis sample is already tightly restricted to individuals who are highly comparable ex ante—workers under retirement age who recently started a job at a for-profit firm that offers a 401(k)—so treated and control units face similar institutional environments and tenure stages. Within this restricted risk set, treatment is defined by a short waiting-period–driven eligibility change rather than by broad differences such as whether a firm offers a plan at all, which reduces scope for large systematic differences between groups.

In addition, we match on rich observables that are key determinants of both treatment and saving outcomes, including demographics, income, firm characteristics, industry, and baseline asset and debt measures, and we verify balance/common support through the propensity score distribution and matched covariate balance diagnostics. Finally, because our DR estimator remains consistent if either the propensity score model or the outcome regression is correctly specified, it provides additional protection against functional-form misspecification. It could be doubly fragile, but we maintain closeness with models used in the paper and also attain very similar results so we don't believe this is the case. While unobserved heterogeneity could still violate ignorability, these design choices (as discussed in the initial paper) make large violations less likely and motivate treating the resulting estimates as informative robustness checks. Methodologically, matching/DR are appropriate tools for improving comparability and reducing specification dependence, but their causal interpretation ultimately hinges on ignorability and overlap—assumptions that are plausible under our restricted, well-balanced sample yet remain vulnerable to unobserved confounding.

Table 7. Matching Estimates of the ATT

Sample	ATT	SE	CI Lower	CI Upper	n_{treated}	n_{control}
Raw aligned	0.813	0.335	0.157	1.470	295	295
Raw non-restricted	0.652	0.261	0.140	1.160	386	386
Replication	0.867	0.298	0.282	1.450	295	295

Table 8. Doubly Robust AIPW Estimates of the ATT

Sample	DR-ATT	SE	CI Lower	CI Upper	n
Raw aligned	0.870	0.254	0.372	1.370	820
Raw non-restricted	0.720	0.221	0.288	1.150	1094
Replication	0.796	0.263	0.281	1.310	824

Table 9. Rosenbaum Sensitivity Analysis for Matched ATT Estimates

Sample	HL Estimate	Γ at CI $= 0$
Raw aligned	1.205	1.10
Raw non-restricted	1.149	1.10
Replication	1.537	1.05

5.5 Interpretation of the Re-Analysis Results

Tables 6–8 report the results of our re-analysis using design-based causal estimators. Table 6 presents nearest-neighbor matching estimates of the ATT. Across all three samples, the estimated effects are positive and economically meaningful, ranging from 0.65 to 0.87. This indicates that newly eligible workers increase financial asset accumulation relative to comparable control workers. The consistency of the estimates across the Raw aligned, Raw non-restricted, and Replication samples suggests that the main finding is not driven by a particular sample construction or alignment procedure.

Table 7 reports doubly robust AIPW estimates of the ATT. These estimates are slightly smaller than the matching estimates but remain positive and precisely estimated. The DR estimator yields ATT values between 0.72 and 0.87 across the three samples. Because the AIPW estimator is consistent if either thev propensity score model or the outcome regression is correctly specified, these results provide strong evidence that the estimated effect of 401(k) eligibility is not an artifact of model misspecification. The close agreement between Tables 6 and 7 reinforces the credibility of the positive treatment effect.

Table 8 reports Rosenbaum sensitivity results for the matched ATT. The Hodges–Lehmann (HL) estimates are positive across samples, indicating higher saving among newly eligible workers relative to matched controls. However, the critical hidden-bias parameter at which the Rosenbaum sensitivity interval first includes zero is small: $\Gamma \approx 1.10$ for the Raw aligned and Raw non-restricted samples, and $\Gamma \approx 1.05$ for the Replication sample. Because Γ bounds the maximum ratio of treatment *odds* for two observationally identical matched individuals, could differ in their odds of treatment by at most 10% due to unobserved factors. This implies the limited robustness to hidden bias. Thus, while the estimated ATT is robust to alternative estimators and sample definitions, it is comparatively sensitive to violations of unconfoundedness, so we interpret the matching results as informative robustness checks rather than definitive evidence under arbitrary hidden bias.

APPENDIX A. ROSENBAUM SENSITIVITY ANALYSIS

Table A1 reports Rosenbaum bounds for the Hodges–Lehmann (HL) estimate of the ATT across increasing values of the hidden-bias parameter Γ. The lower and upper bounds indicate the range of treatment effects consistent with unobserved confounding of magnitude Γ.

Table 10. Rosenbaum Bounds for the Raw Aligned Sample

Γ	Lower HL Bound	Upper HL Bound
1.0	1.2054	1.2054
1.1	-0.0946	1.3054
1.2	-0.0946	1.3054
1.3	-0.0946	1.3054
1.4	-0.0946	1.3054
1.5	-0.0946	1.3054
1.6	-0.0946	1.5054
1.7	-0.0946	1.6054
1.8	-0.2946	1.8054
1.9	-0.2946	1.9054
2.0	-0.4946	2.1054

APPENDIX A. ANALYSIS CODE

A.1 Matching Re-Analysis (matching_reanalyse.qmd)

````
---
title: "Matching"
format: html
---


```{r}
library(dplyr)
library(tidyr)
library(MatchIt)

run_matching_att <- function(df,
                             outcome_var = "taltb",
                             treat_var = "temp",
                             covars,
                             B = 300,
````
````

Table 11. Rosenbaum Bounds for the Raw Non-Restricted Sample

Γ	Lower HL Bound	Upper HL Bound
1.0	1.1493	1.1493
1.1	-0.0507	1.2493
1.2	-0.0507	1.2493
1.3	-0.0507	1.2493
1.4	-0.0507	1.2493
1.5	-0.0507	1.2493
1.6	-0.0507	1.2493
1.7	-0.0507	1.4493
1.8	-0.0507	1.6493
1.9	-0.0507	1.7493
2.0	-0.1507	1.9493

```
                          seed = 156) {
# 1. Main analysis sample (same as IPW)
dat <- make_ipw_sample3(df)

# 2. Build outcome
dat$y401k_ipw <- make_ipw_outcome3(dat, outcome_var)

# 3. Check covariates exist
missing_covars <- setdiff(covars, names(dat))
if (length(missing_covars) > 0) {
  stop(
    "These covariates are not in the data: ",
    paste(missing_covars, collapse = ", ")
  )
}

# 4. Filter to non-missing treatment, outcome, and covariates
dat_complete <- dat %>%
```

Table 12. Rosenbaum Bounds for the Replication Sample

Γ	Lower HL Bound	Upper HL Bound
1.0	1.5370	1.5370
1.1	-0.0630	1.6370
1.2	-0.0630	1.6370
1.3	-0.0630	1.6370
1.4	-0.0630	1.6370
1.5	-0.0630	1.6370
1.6	-0.0630	1.6370
1.7	-0.0630	1.8370
1.8	-0.0630	2.0370
1.9	-0.0630	2.1370
2.0	-0.0630	2.2370

```
    filter(!is.na(.data[[treat_var]]),
           !is.na(y401k_ipw)) %>%
    drop_na(all_of(covars))

  if (nrow(dat_complete) == 0) {
    stop(
      "After filtering for non-missing treatment, outcome, and covariates, ",
      "no observations remain. Check missingness again."
    )
  }

  # 5. PS formula:
# ALWAYS include age squared
rhs_terms <- c("tage", "I(tage^2)", setdiff(covars, "tage"))
rhs <- paste(rhs_terms, collapse = " + ")

ps_formula <- as.formula(
  paste0(treat_var, " ~ ", rhs)
```

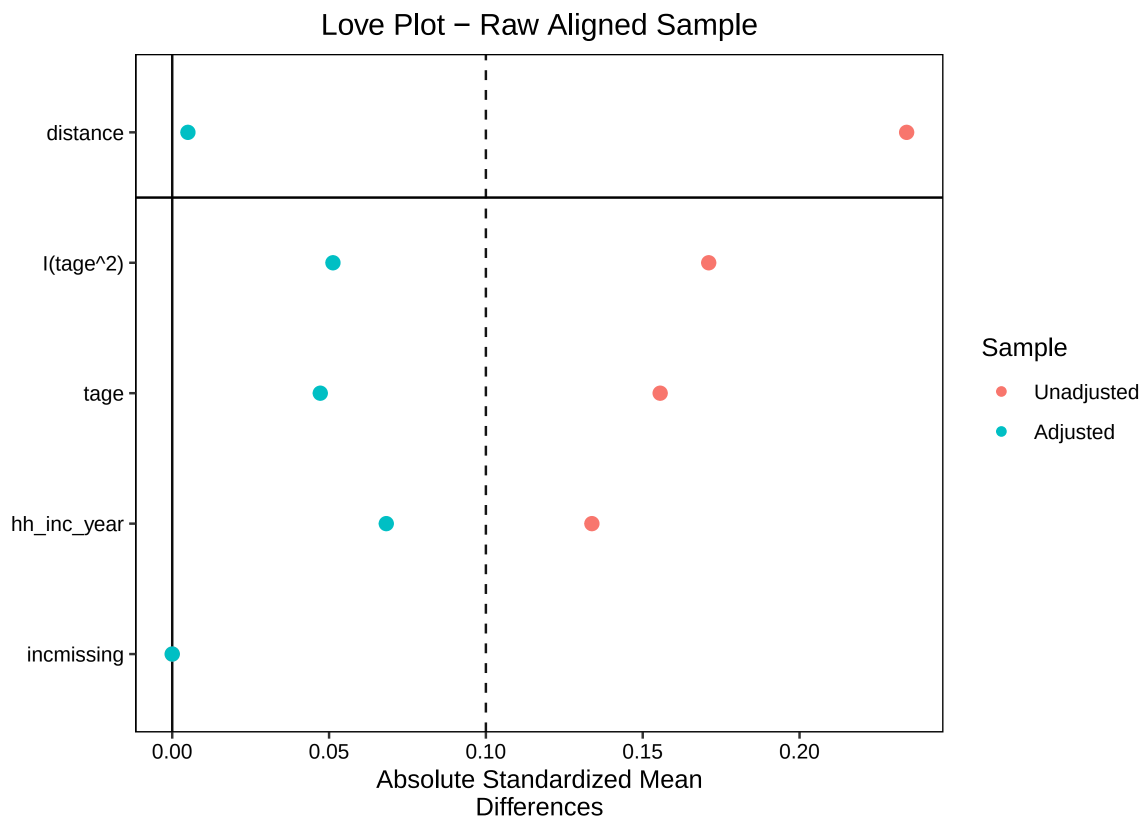

Figure 1. Love plot showing covariate balance before and after matching for the Raw Aligned sample.

```
)


  # 6. Nearest neighbor matching on logit PS
  m.out <- matchit(
    formula = ps_formula,
    data = dat_complete,
    method = "nearest",
    distance = "logit",
    replace = FALSE,
    ratio = 1
  )

  matched_dat <- match.data(m.out)

```

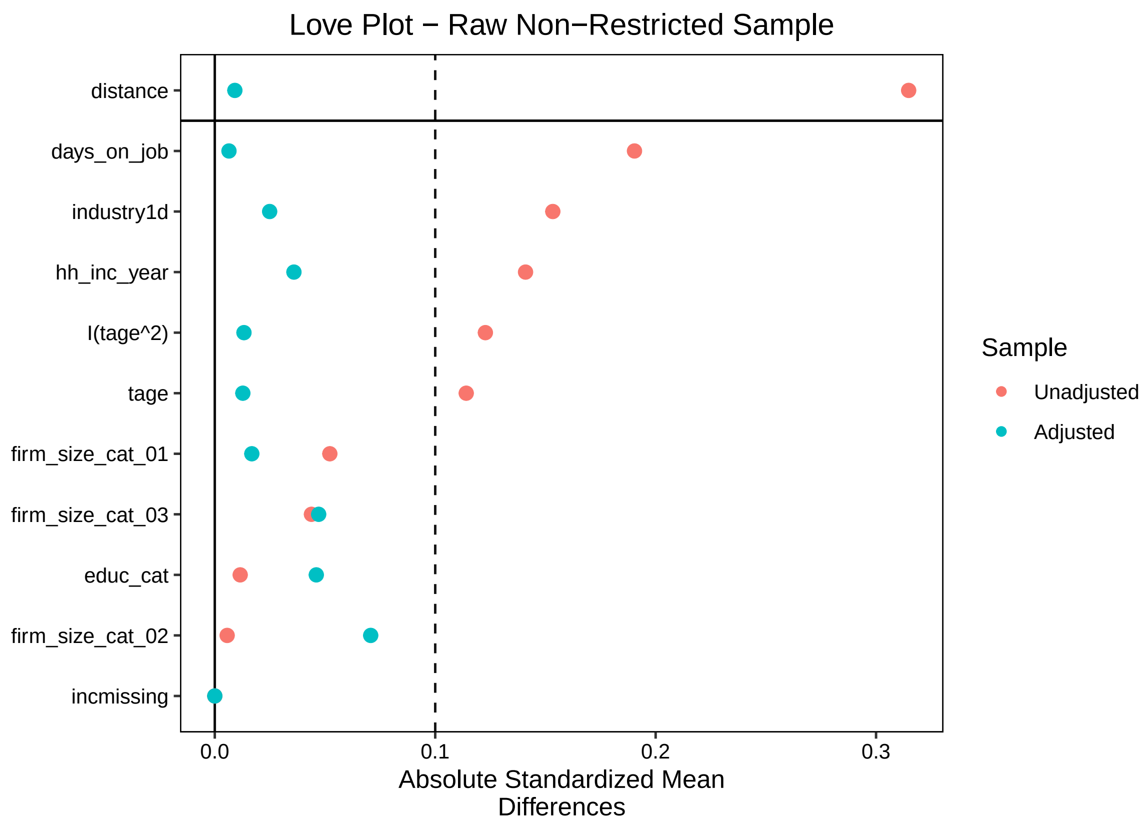

Figure 2. Love plot showing covariate balance before and after matching for the Raw Non-Restricted sample.

```
# 7. ATT point estimate
z <- matched_dat[[treat_var]]
y <- matched_dat$y401k_ipw

att_hat <- mean(y[z == 1]) - mean(y[z == 0])
n_treated <- sum(z == 1)
n_control <- sum(z == 0)
N <- nrow(matched_dat)

# 8. Bootstrap SE + CI
set.seed(seed)
boot_ests <- replicate(B, {
  idx <- sample(seq_len(N), replace = TRUE)
  boot_dat <- matched_dat[idx, ]
  z_b <- boot_dat[[treat_var]]
```

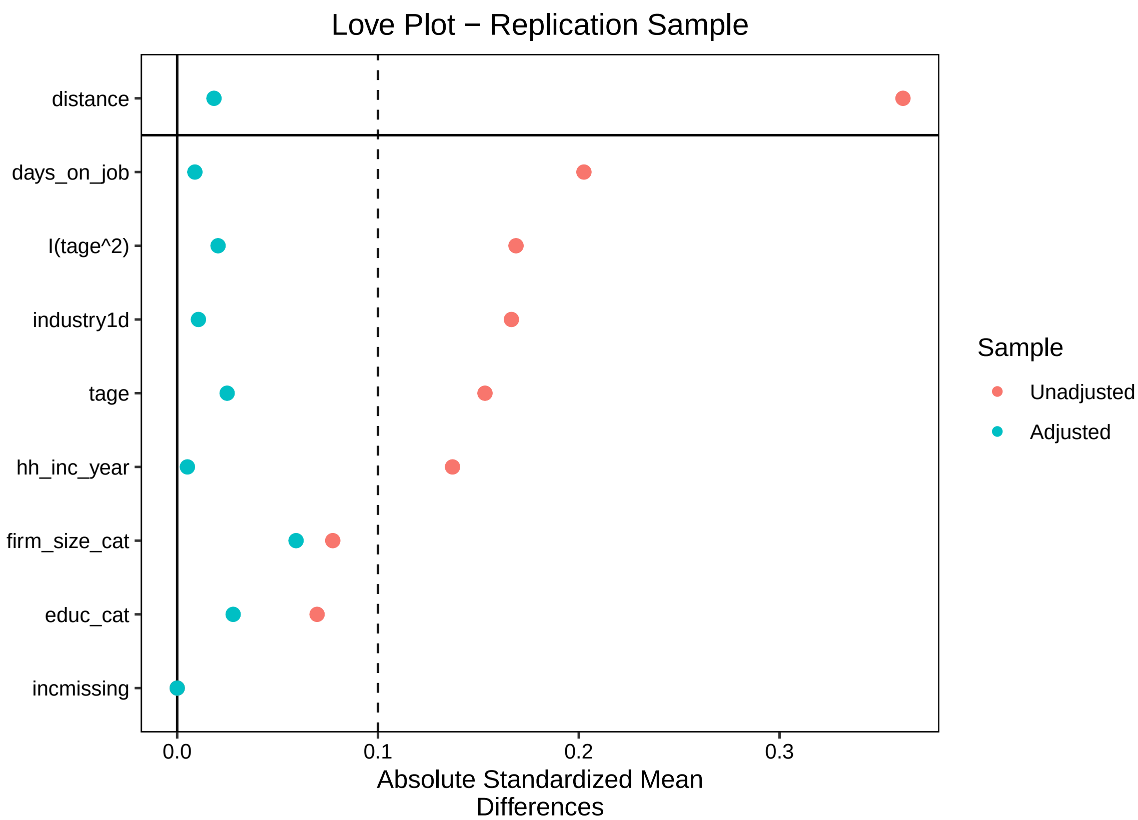

Figure 3. Love plot showing covariate balance before and after matching for the Replication sample.

```
  y_b <- boot_dat$y401k_ipw
  mean(y_b[z_b == 1]) - mean(y_b[z_b == 0])
})

se_hat <- sd(boot_ests)
ci_95 <- att_hat + c(-1, 1) * 1.96 * se_hat

list(
  estimate = att_hat,
  se = se_hat,
  ci_95 = ci_95,
  n_treated = n_treated,
  n_control = n_control,
  n_matched_rows = N,
  treat_var = treat_var,
```

````
    covariates = covars
  )
}

```
```{r}
covars_raw_aligned <- c("tage", "hh_inc_year", "incmissing")

match_raw_aligned <- run_matching_att(
  dat_raw_aligned_ipw,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_raw_aligned
)

match_raw_aligned$estimate
match_raw_aligned$se
match_raw_aligned$ci_95
match_raw_aligned$n_treated
match_raw_aligned$n_control

```
```{r}
covars_raw_non <- c("tage", "hh_inc_year", "incmissing",
                    "educ_cat", "firm_size_cat", "industry1d", "days_on_job")

match_raw_non <- run_matching_att(
  dat_raw_ipw_non,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_raw_non
)

match_raw_non$estimate
match_raw_non$se
````

~~~~
match_raw_non$ci_95
match_raw_non$n_treated
match_raw_non$n_control
```
```{r}
covars_rep <- c("tage", "hh_inc_year", "incmissing",
                "educ_cat", "firm_size_cat", "industry1d", "days_on_job")

match_rep <- run_matching_att(
  dat_rep_ipw_rep,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_rep
)

match_rep$estimate
match_rep$se
match_rep$ci_95
match_rep$n_treated
match_rep$n_control

```
```{r}
matching_summary <- tibble::tibble(
  sample = c("Raw aligned", "Raw non-restricted", "Replication"),
  estimate = c(match_raw_aligned$estimate,
              match_raw_non$estimate,
              match_rep$estimate),
  se = c(match_raw_aligned$se,
              match_raw_non$se,
              match_rep$se),
  ci_lower = c(match_raw_aligned$ci_95[1],
              match_raw_non$ci_95[1],
              match_rep$ci_95[1]),
  ci_upper = c(match_raw_aligned$ci_95[2],
~~~~

````
               match_raw_non$ci_95[2],
               match_rep$ci_95[2]),
  n_treated = c(match_raw_aligned$n_treated,
                match_raw_non$n_treated,
                match_rep$n_treated),
  n_control = c(match_raw_aligned$n_control,
                match_raw_non$n_control,
                match_rep$n_control)
)

# Print the table
print(matching_summary)
```
````

A.2 Doubly Robust Estimation (Doubly_Robust.qmd)

````
---
title: "Doubly Robust"
format: html
---





```{r}
library(dplyr)
library(tidyr)

run_dr_att <- function(df,
                     outcome_var = "taltb",
                     treat_var = "temp",
                     covars,
                     B = 300,
                     seed = 156,
````

```r
                     ps_trim = c(0.01, 0.99)) {
# 1. Main analysis sample (same as IPW pipeline)
dat <- make_ipw_sample3(df)

# 2. Build outcome (same definition as for IPW)
dat$y401k_ipw <- make_ipw_outcome3(dat, outcome_var)

# 3. Check covariates exist
missing_covars <- setdiff(covars, names(dat))
if (length(missing_covars) > 0) {
  stop(
    "These covariates are not in the data: ",
    paste(missing_covars, collapse = ", ")
  )
}

# 4. Filter to complete cases in treatment, outcome, and covariates
dat_complete <- dat %>%
  filter(
    !is.na(.data[[treat_var]]),
    !is.na(y401k_ipw)
  ) %>%
  tidyr::drop_na(all_of(covars))

if (nrow(dat_complete) == 0) {
  stop(
    "After filtering for non-missing treatment, outcome, and covariates, ",
    "no observations remain."
  )
}

# Rename for convenience
Z <- dat_complete[[treat_var]] # 1 = treated (temp), 0 = control
Y <- dat_complete$y401k_ipw

```

```
# 5. Propensity score model (same as before, tage + tage^2 + others)
rhs_terms <- c("tage", "I(tage^2)", setdiff(covars, "tage"))
rhs <- paste(rhs_terms, collapse = " + ")
ps_formula <- as.formula(paste0(treat_var, " ~ ", rhs))

ps_model <- glm(
  ps_formula,
  data = dat_complete,
  family = binomial(link = "logit")
)

e_hat <- predict(ps_model, type = "response")



# 6. Outcome model for controls only: m0(x) = E[Y | T=0, X]
outcome_rhs <- paste(covars, collapse = " + ")
m0_formula <- as.formula(paste0("y401k_ipw ~ ", outcome_rhs))

m0_model <- glm(
  m0_formula,
  data = dat_complete[Z == 0, ],
  family = gaussian()
)

m0_hat <- predict(m0_model, newdata = dat_complete, type = "response")

# 7. Doubly robust ATT estimator
# tau_ATT = (1/N1) * sum[ Z*(Y - m0) - (e/(1-e))*(1-Z)*(Y - m0) ]
N <- nrow(dat_complete)
N1 <- sum(Z == 1)
R0_hat <- Y - m0_hat

tau_hat <- (1 / N1) * sum(
  Z * R0_hat -
```

```
    (e_hat / (1 - e_hat)) * (1 - Z) * R0_hat
)

# 8. Bootstrap SE and CI (using fixed nuisance estimates)
set.seed(seed)

boot_ests <- replicate(B, {
  idx <- sample(seq_len(N), replace = TRUE)
  Z_b <- Z[idx]
  Y_b <- Y[idx]
  e_b <- e_hat[idx]
  m0_b <- m0_hat[idx]
  R0_b <- Y_b - m0_b
  N1_b <- sum(Z_b == 1)

  # guard against (extremely unlikely) all-control resample
  if (N1_b == 0) return(NA_real_)

  sum(
    Z_b * R0_b -
      (e_b / (1 - e_b)) * (1 - Z_b) * R0_b
  ) / N1_b
})

boot_ests <- boot_ests[!is.na(boot_ests)]
se_hat <- sd(boot_ests)
ci_95 <- tau_hat + c(-1, 1) * 1.96 * se_hat

list(
  estimand = "ATT",
  estimate = tau_hat,
  se = se_hat,
  ci_95 = ci_95,
  ps_model = ps_model,
  # keep these names for compatibility, even though m1 is not used for ATT
```

````
    m1_model = NULL,
    m0_model = m0_model,
    e_hat = e_hat,
    m1_hat = NULL,
    m0_hat = m0_hat,
    covariates = covars,
    treat_var = treat_var,
    outcome_var = outcome_var,
    n_obs = N,
    boot_ests = boot_ests,
    data_used = dat_complete
  )
}

```
```{r}
# 1) Raw aligned
covars_raw_aligned <- c("tage", "hh_inc_year", "incmissing")

dr_raw_aligned <- run_dr_att(
  dat_raw_aligned_ipw,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_raw_aligned
)

dr_raw_aligned$estimate
dr_raw_aligned$se
dr_raw_aligned$ci_95


# 2) Raw non-restricted
covars_raw_non <- c(
  "tage", "hh_inc_year", "incmissing",
  "educ_cat", "firm_size_cat", "industry1d", "days_on_job"
````

````
)

dr_raw_non <- run_dr_att(
  dat_raw_ipw_non,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_raw_non
)

dr_raw_non$estimate
dr_raw_non$se
dr_raw_non$ci_95


# 3) Replication sample
covars_rep <- c(
  "tage", "hh_inc_year", "incmissing",
  "educ_cat", "firm_size_cat", "industry1d", "days_on_job"
)

dr_rep <- run_dr_att(
  dat_rep_ipw_rep,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_rep
)

dr_rep$estimate
dr_rep$se
dr_rep$ci_95

```
```{r}
dr_summary <- tibble::tibble(
  sample = c("Raw aligned", "Raw non-restricted", "Replication"),
````

```
  method = "DR AIPW (ATT)",
  estimate = c(dr_raw_aligned$estimate,
               dr_raw_non$estimate,
               dr_rep$estimate),
  se = c(dr_raw_aligned$se,
               dr_raw_non$se,
               dr_rep$se),
  ci_lower = c(dr_raw_aligned$ci_95[1],
               dr_raw_non$ci_95[1],
               dr_rep$ci_95[1]),
  ci_upper = c(dr_raw_aligned$ci_95[2],
               dr_raw_non$ci_95[2],
               dr_rep$ci_95[2]),
  n_obs = c(dr_raw_aligned$n_obs,
               dr_raw_non$n_obs,
               dr_rep$n_obs)
)

dr_summary

```
```

A.3 Rosenbaum Sensitivity Analysis (Rosenbaum.qmd)

```
---
title: "Rosenbaum"
format: html
---


```{r}

## Matching + Rosenbaum Sensitivity Script for 401(k) Data


```

```
## 0. Libraries
library(dplyr)
library(tidyr)
library(MatchIt)
library(rbounds)

make_ipw_sample3 <-function (dat)
{
    dat %>% filter(age_ok, for_profit, yr1jb1 == 1, y401k, notmissing)
}

make_ipw_outcome3 <- function (df, base)
{
    a6 <- df[[paste0(base, "6")]]
    a9 <- df[[paste0(base, "9")]]
    a12 <- df[[paste0(base, "12")]]
    log(a12 + 10) - 2 * log(a9 + 10) + log(a6 + 10)
}

## NOTE:
##RUN THESE FIRST
## - dat_raw_aligned_ipw
## - dat_raw_ipw_non
## - dat_rep_ipw_rep
## - make_ipw_sample3()
## - make_ipw_outcome3()



## 1. Matching function that returns ATT + SE + CI + matched data


run_matching_att <- function(df,
                          outcome_var = "taltb",
                          treat_var = "temp",
```

```
                             covars,
                             B = 300,
                             seed = 156) {
# 1. Main analysis sample (same as IPW)
dat <- make_ipw_sample3(df)

# 2. Build outcome (401k IPW outcome)
dat$y401k_ipw <- make_ipw_outcome3(dat, outcome_var)

# 3. Check covariates exist
missing_covars <- setdiff(covars, names(dat))
if (length(missing_covars) > 0) {
  stop(
    "These covariates are not in the data: ",
    paste(missing_covars, collapse = ", ")
  )
}

# 4. Filter to non-missing treatment, outcome, and covariates
dat_complete <- dat %>%
  filter(
    !is.na(.data[[treat_var]]),
    !is.na(y401k_ipw)
  ) %>%
  tidyr::drop_na(all_of(covars))

if (nrow(dat_complete) == 0) {
  stop(
    "After filtering for non-missing treatment, outcome, and covariates, ",
    "no observations remain. Check missingness again."
  )
}

# 5. Propensity score formula (always include age + age^2)
rhs_terms <- c("tage", "I(tage^2)", setdiff(covars, "tage"))
```

```
rhs <- paste(rhs_terms, collapse = " + ")
ps_formula <- as.formula(paste0(treat_var, " ~ ", rhs))

# 6. Nearest neighbor matching on logit PS (ATT)
m.out <- matchit(
  formula = ps_formula,
  data = dat_complete,
  method = "nearest",
  distance = "logit",
  replace = FALSE,
  ratio = 1,
  estimand = "ATT"
)

matched_dat <- match.data(m.out)

# 7. ATT point estimate on matched sample
z <- matched_dat[[treat_var]]
y <- matched_dat$y401k_ipw

att_hat <- mean(y[z == 1]) - mean(y[z == 0])
n_treated <- sum(z == 1)
n_control <- sum(z == 0)
N <- nrow(matched_dat)

# 8. Bootstrap SE + CI
set.seed(seed)
boot_ests <- replicate(B, {
  idx <- sample(seq_len(N), replace = TRUE)
  boot_dat <- matched_dat[idx, ]
  z_b <- boot_dat[[treat_var]]
  y_b <- boot_dat$y401k_ipw
  mean(y_b[z_b == 1]) - mean(y_b[z_b == 0])
})

```

```r
  se_hat <- sd(boot_ests)
  ci_95 <- att_hat + c(-1, 1) * 1.96 * se_hat

  # 9. Return everything useful
  list(
    estimate = att_hat,
    se = se_hat,
    ci_95 = ci_95,
    n_treated = n_treated,
    n_control = n_control,
    n_matched_rows = N,
    treat_var = treat_var,
    covariates = covars,
    m.out = m.out,
    matched_data = matched_dat,
    boot_ests = boot_ests
  )
}



## 2. Rosenbaum sensitivity function (HodgesLehmann)


rosenbaum_hl <- function(match_result, treat_var = "temp") {
  dat <- match_result$matched_data

  if (!"subclass" %in% names(dat)) {
    stop("Matched data must contain 'subclass' to identify matched pairs/sets.")
  }

  # For 1:1 matching, each subclass has 1 treated + 1 control
  dat_pairs <- dat %>%
    arrange(subclass, dplyr::desc(.data[[treat_var]])) %>%
    group_by(subclass) %>%
```

```r
    summarise(
      y_treat = y401k_ipw[.data[[treat_var]] == 1],
      y_ctrl = y401k_ipw[.data[[treat_var]] == 0],
      .groups = "drop"
    )

  # Rosenbaum bounds for HodgesLehmann estimate
  # x = treated outcomes, y = control outcomes
  hlsens(
    x = dat_pairs$y_treat,
    y = dat_pairs$y_ctrl,
    Gamma = 2, # max Gamma to check (change if needed)
    GammaInc = 0.1 # step size in Gamma grid
  )
}



## 3. Run matching on the three datasets


## 3.1 Raw aligned
covars_raw_aligned <- c("tage", "hh_inc_year", "incmissing")

match_raw_aligned <- run_matching_att(
  dat_raw_aligned_ipw,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_raw_aligned
)

## 3.2 Raw non-restricted
covars_raw_non <- c(
  "tage", "hh_inc_year", "incmissing",
  "educ_cat", "firm_size_cat", "industry1d", "days_on_job"
```

```
)

match_raw_non <- run_matching_att(
  dat_raw_ipw_non,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_raw_non
)

## 3.3 Replication sample
covars_rep <- c(
  "tage", "hh_inc_year", "incmissing",
  "educ_cat", "firm_size_cat", "industry1d", "days_on_job"
)

match_rep <- run_matching_att(
  dat_rep_ipw_rep,
  outcome_var = "taltb",
  treat_var = "temp",
  covars = covars_rep
)



## 4. Summaries of matching estimates ----


matching_summary <- tibble::tibble(
  sample = c("Raw aligned", "Raw non-restricted", "Replication"),
  estimate = c(match_raw_aligned$estimate,
               match_raw_non$estimate,
               match_rep$estimate),
  se = c(match_raw_aligned$se,
               match_raw_non$se,
               match_rep$se),
```

```
  ci_lower = c(match_raw_aligned$ci_95[1],
               match_raw_non$ci_95[1],
               match_rep$ci_95[1]),
  ci_upper = c(match_raw_aligned$ci_95[2],
               match_raw_non$ci_95[2],
               match_rep$ci_95[2]),
  n_treat = c(match_raw_aligned$n_treated,
              match_raw_non$n_treated,
              match_rep$n_treated),
  n_ctrl = c(match_raw_aligned$n_control,
             match_raw_non$n_control,
             match_rep$n_control),
  n_rows = c(match_raw_aligned$n_matched_rows,
             match_raw_non$n_matched_rows,
             match_rep$n_matched_rows)
)

matching_summary



## 5. Rosenbaum sensitivity for each matched sample


rosen_raw_aligned <- rosenbaum_hl(match_raw_aligned)
rosen_raw_non <- rosenbaum_hl(match_raw_non)
rosen_rep <- rosenbaum_hl(match_rep)

# Print results to inspect in console
rosen_raw_aligned
rosen_raw_non
rosen_rep
```

````
```
```{r}

#install.packages("cobalt")


library(cobalt)

love.plot(
  match_rep$m.out,
  stats = "mean.diffs",
  abs = TRUE,
  binary = "std", # <- force standardized for binary vars
  var.order = "unadjusted",
  thresholds = c(m = 0.1),
  title = "Love Plot - Replication Sample"
)


```
```{r}
love.plot(
  match_raw_aligned$m.out,
  stats = "mean.diffs",
  abs = TRUE,
  binary = "std",
  var.order = "unadjusted",
  thresholds = c(m = 0.1),
  title = "Love Plot - Raw Aligned Sample"
)

love.plot(
  match_raw_non$m.out,
  stats = "mean.diffs",
  abs = TRUE,
````

````
  binary = "std",
  var.order = "unadjusted",
  thresholds = c(m = 0.1),
  title = "Love Plot - Raw Non-Restricted Sample"
)
```



```{r}
# Function to extract Rosenbaum bounds into a clean format
extract_rosenbaum_table <- function(rosen_obj, sample_name) {
  # ’hlsens’ stores the data frame in $bounds
  # Columns are typically: Gamma, Lower Bound, Upper Bound
  df <- as.data.frame(rosen_obj$bounds)

  df %>%
    mutate(sample = sample_name) %>%
    select(sample, everything()) %>%
    rename(
      Gamma = 1, # Usually the first column
      Lower_HL = 2,
      Upper_HL = 3
    )
}

# Combine all results into one table
rosen_summary_table <- bind_rows(
  extract_rosenbaum_table(rosen_raw_aligned, "Raw aligned"),
  extract_rosenbaum_table(rosen_raw_non, "Raw non-restricted"),
  extract_rosenbaum_table(rosen_rep, "Replication")
)
````

````
print(rosen_summary_table)
```
````

A.4 Results Compilation (Results.qmd)

````
---
title: "Results"
format: html
---


```{r}
dr_summary
matching_summary
rosen_summary_table
```
````

APPENDIX B. REPLICATION

B.1 Load Data Raw

````
---
title: "Untitled"
format: html
---


```{r}
library(readr)
library(stringr)

# helper to convert .sas layout into fwf specification
sas_to_fwf <- function(sasfile) {
  sas <- readLines(sasfile)
  pattern <- "^\\s*([A-Za-z0-9_]+)\\s+\\$?\\s*(\\d+)\\s*-\\s*(\\d+)"
  m <- str_match(sas, pattern)
````

````
  m <- m[!is.na(m[,1]), , drop = FALSE]
  varnames <- m[,2]
  start <- as.integer(m[,3])
  end <- as.integer(m[,4])
  widths <- end - start + 1
  fwf_widths(widths, col_names = varnames)
}

# directories
raw_dir <- "data/raw"
out_dir <- file.path(raw_dir, "rds_1") # <- changed folder name
dir.create(out_dir, showWarnings = FALSE)

# list of file stems
stems <- c("t3", "t6", "t7", "t9", "t12",
           "w7", "w8", "w9")

# loop to convert all files
for (stem in stems) {

  datfile <- file.path(raw_dir, paste0(stem, ".dat"))
  sasfile <- file.path(raw_dir, paste0(stem, ".sas"))
  rdsfile <- file.path(out_dir, paste0(stem, ".rds"))

  if (!file.exists(datfile)) next
  if (!file.exists(sasfile)) next

  fwf <- sas_to_fwf(sasfile)
  df <- read_fwf(datfile, fwf, progress = FALSE)

  saveRDS(df, rdsfile)
}

```
````
````

B.2 Construct aligned data

~~~
---
title: "Untitled"
format: html
---


```{r}
library(dplyr)
library(tidyr)


## Helper: safe numeric conversion

to_num <- function(x) suppressWarnings(as.numeric(x))


## Helper: clean job start date (TSJDATE1)
## - convert to numeric
## - set non-positive / sentinel values (<=0, -1) to NA

clean_tsjdate <- function(x) {
  x <- suppressWarnings(as.numeric(x))
  x[x <= 0] <- NA
  x
}


## Helper: build assets for one wave

make_assets_wave <- function(df, wave_num) {
  df %>%
    transmute(
      SSUID, SHHADID, EENTAID, EPPPNUM,
      wave = wave_num,
~~~

```
    # 401(k) balance
    taltb = to_num(TALTB),
    # IRA balance
    thhira = to_num(THHIRA),
    # Other financial assets (interest, stocks, other assets)
    otherassets = to_num(THHINTBK) +
                  to_num(THHINTOT) +
                  to_num(RHHSTK) +
                  to_num(THHOTAST),
    # secured & unsecured debt
    thhscdbt = to_num(THHSCDBT),
    rhhuscbt = to_num(RHHUSCBT),
    # car values: sum over up to 3 cars
    tcarval = to_num(TCARVAL1) +
                to_num(TCARVAL2) +
                to_num(TCARVAL3)
  )
}


## MAIN: build_dat()
## takes t3,t6,t9,t12,t7,w7,w8,w9 and returns merged dataset
## with: age_ok, for_profit, yr1jb1, temp, y401k, notmissing,
## hh_inc_year, etc.

build_dat <- function(t3, t6, t9, t12, t7, w7, w8, w9) {
  id_vars <- c("SSUID","SHHADID","EENTAID","EPPPNUM")

  ## 1) Assets across waves  wide
  assets_long <- bind_rows(
    make_assets_wave(t3, 3),
    make_assets_wave(t6, 6),
    make_assets_wave(t9, 9),
    make_assets_wave(t12, 12)
  )
```

```r

assets_wide <- assets_long %>%
  pivot_wider(
    id_cols = all_of(id_vars),
    names_from = wave,
    values_from = c(taltb, thhira, otherassets, thhscdbt, rhhuscbt, tcarval),
    names_sep = ""
  )

## 2) Wave-7 core (panel = SREFMON==4)
core7 <- w7 %>%
  filter(SREFMON == 4) %>%
  transmute(
    SSUID, SHHADID, EENTAID, EPPPNUM,
    tage = to_num(TAGE),
    wpfinwgt = to_num(WPFINWGT),
    eclwrk1 = to_num(ECLWRK1),
    efnp = to_num(EFNP),
    esex = to_num(ESEX),
    tempall1 = to_num(TEMPALL1),
    ejbind1 = to_num(EJBIND1),
    tsjdate1 = clean_tsjdate(TSJDATE1),
    srotaton = to_num(SROTATON)
  )

## 3) Year-1 income: sum THTOTINC from waves 79
make_inc <- function(w, tag) {
  w %>%
    select(SSUID, SHHADID, EENTAID, EPPPNUM, SREFMON, THTOTINC) %>%
    mutate(
      THTOTINC = to_num(THTOTINC),
      month = paste0(tag, "_", SREFMON)
    ) %>%
    select(-SREFMON) %>%
    pivot_wider(
```

```
      id_cols = all_of(id_vars),
      names_from = month,
      values_from = THTOTINC
    )
}

w7_inc <- make_inc(w7, "w7")
w8_inc <- make_inc(w8, "w8")
w9_inc <- make_inc(w9, "w9")

year1_income <- w7_inc %>%
  full_join(w8_inc, by = id_vars) %>%
  full_join(w9_inc, by = id_vars) %>%
  mutate(
    hh_inc_year = rowSums(
      dplyr::select(., starts_with("w7_"), starts_with("w8_"), starts_with("w9_")),
      na.rm = FALSE
    )
  ) %>%
  select(all_of(id_vars), hh_inc_year)

## 4) Topical module 7: plan eligibility and reasons
tm7 <- t7 %>%
  transmute(
    SSUID, SHHADID, EENTAID, EPPPNUM,
    e1taxdef = to_num(E1TAXDEF),
    e2taxdef = to_num(E2TAXDEF),
    e3taxdef = to_num(E3TAXDEF),
    enoina03 = to_num(ENOINA03),
    enoinb03 = to_num(ENOINB03),
    etdeffen = to_num(ETDEFFEN)
  )

## 5) Merge all pieces
dat <- assets_wide %>%
```

```
  inner_join(core7, by = id_vars) %>%
  inner_join(tm7, by = id_vars) %>%
  left_join(year1_income, by = id_vars)

## 6) Derive sample flags and d21ltaltb
# (Make d21ltaltb robust in case some waves are missing)
has_t12 <- "taltb12" %in% names(dat)
has_t3 <- "taltb3" %in% names(dat)

dat <- dat %>%
  mutate(
    age_ok = !is.na(tage) & tage >= 22 & tage <= 64,
    for_profit = (eclwrk1 == 1),

    yr1jb1 = dplyr::case_when(
      srotaton == 1 ~ tsjdate1 > 19970299,
      srotaton == 2 ~ tsjdate1 > 19970399,
      srotaton == 3 ~ tsjdate1 > 19970499,
      srotaton == 4 ~ tsjdate1 > 19970599,
      TRUE ~ NA
    ),

    temp = as.integer(
      (enoina03 == 1 & etdeffen == 1) |
      (enoinb03 == 1)
    ),

    y401k = (temp == 1) |
            (e1taxdef == 1) |
            (e2taxdef == 1) |
            (e3taxdef == 1) |
            (etdeffen == 1 & yr1jb1 == 1),

    # main outcome (use 6912 if available, else 369)
    d21ltaltb = dplyr::case_when(
```

```
      has_t12 ~ log(taltb12 + 10) - 2 * log(taltb9 + 10) + log(taltb6 + 10),
      has_t3 ~ log(taltb9 + 10) - 2 * log(taltb6 + 10) + log(taltb3 + 10),
      TRUE ~ NA_real_
    ),

    lnA6 = log(taltb6 + 10),

    # income: allowed to be missing, we flag it
    incmissing = as.integer(is.na(hh_inc_year)),

    # require outcome + key regressors present
    notmissing = !is.na(d21ltaltb) &
                 !is.na(tage) &
                 !is.na(eclwrk1) &
                 !is.na(tsjdate1) &
                 !is.na(srotaton) &
                 !is.na(wpfinwgt)
  )

  dat
}


## Helpers to build Table 1 from a built dat
w_mean_sd <- function(x, w) {
  x <- as.numeric(x)
  w <- as.numeric(w)
  ok <- !is.na(x) & !is.na(w)
  x <- x[ok]; w <- w[ok]
  if (!length(x)) return(c(mean = NA_real_, sd = NA_real_))
  w <- w / sum(w)
  mu <- sum(w * x)
  var <- sum(w * (x - mu)^2)
  c(mean = mu, sd = sqrt(var))
}
```

```

make_block <- function(df) {
  out <- list(
    Age = w_mean_sd(df$tage, df$wpfinwgt),
    ‘Yearly household income‘ =
      w_mean_sd(df$hh_inc_year, df$wpfinwgt),
    ‘401(k) assets‘ =
      w_mean_sd(df$taltb6, df$wpfinwgt),
    ‘IRA and Keogh assets‘ =
      w_mean_sd(df$thhira6, df$wpfinwgt),
    ‘Other financial assets‘=
      w_mean_sd(df$otherassets6, df$wpfinwgt),
    ‘Secured debt‘ =
      w_mean_sd(df$thhscdbt6, df$wpfinwgt),
    ‘Unsecured debt‘ =
      w_mean_sd(df$rhhuscbt6, df$wpfinwgt),
    ‘Car value‘ =
      w_mean_sd(df$tcarval6, df$wpfinwgt)
  )
  vals <- lapply(out, function(msd)
    sprintf("%.1f\n(%.1f)", msd["mean"], msd["sd"]))
  tibble::as_tibble_row(vals)
}

make_table1 <- function(dat) {
  # Apply the same sample restrictions as in the Stata do-file
  table1_sample <- dat %>%
    filter(
      age_ok,
      for_profit,
      yr1jb1 == 1,
      y401k,
      notmissing
    )

```

````
  cat("Table 1 sample size: ", nrow(table1_sample), "\n")
  cat("Treatment (temp==1): ", sum(table1_sample$temp == 1, na.rm = TRUE), "\n")
  cat("Control (temp==0): ", sum(table1_sample$temp == 0, na.rm = TRUE), "\n")

  income_sample <- table1_sample %>%
    filter(!is.na(hh_inc_year))
  cat("Income row N (non-missing hh_inc_year): ", nrow(income_sample), "\n")

  tab_all <- make_block(table1_sample) %>%
    mutate(group = "All",
          Observations = nrow(table1_sample))

  tab_treat <- make_block(filter(table1_sample, temp == 1)) %>%
    mutate(group = "Treatment group",
          Observations = sum(table1_sample$temp == 1, na.rm = TRUE))

  tab_ctrl <- make_block(filter(table1_sample, temp == 0)) %>%
    mutate(group = "Control group",
          Observations = sum(table1_sample$temp == 0, na.rm = TRUE))

  bind_rows(tab_all, tab_treat, tab_ctrl) %>%
    relocate(group, Observations)
}


```
```{r}

dat_raw <- build_dat(
  t3_raw, t6_raw, t9_raw, t12_raw,
  t7_raw, w7_raw, w8_raw, w9_raw
)

table1_raw <- make_table1(dat_raw)
table1_raw
````

~~~~
dat_rep <- build_dat(
  t3_rep, t6_rep, t9_rep, t12_rep,
  t7_rep, w7_rep, w8_rep, w9_rep
)

table1_rep <- make_table1(dat_rep)
table1_rep
```
```{r}
id_vars <- c("SSUID","SHHADID","EENTAID","EPPPNUM")

normalize_ids <- function(df) {
  df %>%
    mutate(
      SSUID = sub("^0+", "", as.character(SSUID)),
      SHHADID = sub("^0+", "", as.character(SHHADID)),
      EENTAID = sub("^0+", "", as.character(EENTAID)),
      EPPPNUM = sub("^0+", "", as.character(EPPPNUM))
    )
}

dat_raw_id <- normalize_ids(dat_raw)
dat_rep_id <- normalize_ids(dat_rep)

```
```{r}
flags_rep <- dat_rep_id %>%
  select(all_of(id_vars),
         yr1jb1_rep = yr1jb1,
         temp_rep = temp,
         y401k_rep = y401k)

```
```{r}
dat_raw_aligned <- dat_raw_id %>%
~~~~

```
  left_join(flags_rep, by = id_vars) %>%
  mutate(
    yr1jb1 = yr1jb1_rep,
    temp = temp_rep,
    y401k = y401k_rep
  )

```
```
```{r}
table1_rep <- make_table1(dat_rep_id)
table1_raw2 <- make_table1(dat_raw_aligned)

table1_rep
table1_raw2

```
```

B.3 Table 1 Construction (table1.qmd)

````
---
title: "table 1"
format: pdf
---


```{r}
library(dplyr)
library(tidyr)

to_num <- function(x) suppressWarnings(as.numeric(x))


## 1. Load raw files


````
````

```
t3 <- readRDS("data/raw/rds_down/t3.rds")
t6 <- readRDS("data/raw/rds_down/t6.rds")
t9 <- readRDS("data/raw/rds_down/t9.rds")
t12 <- readRDS("data/raw/rds_down/t12.rds")

t7 <- readRDS("data/raw/rds_down/t7.rds")
w7 <- readRDS("data/raw/rds_down/w7.rds")
w8 <- readRDS("data/raw/rds_down/w8.rds")
w9 <- readRDS("data/raw/rds_down/w9.rds")

id_vars <- c("SSUID","SHHADID","EENTAID","EPPPNUM")


## 2. Assets: build taltb6, thhira6, otherassets6, etc.
## Names are chosen to line up with the Stata do-file.


make_assets_wave <- function(df, wave_num) {
  df %>%
    transmute(
      SSUID, SHHADID, EENTAID, EPPPNUM,
      wave = wave_num,
      # 401(k) balance
      taltb = to_num(TALTB),
      # IRA balance
      thhira = to_num(THHIRA),
      # Other financial assets (interest, stocks, "other" assets)
      otherassets = to_num(THHINTBK) +
                    to_num(THHINTOT) +
                    to_num(RHHSTK) +
                    to_num(THHOTAST),
      # secured & unsecured debt
      thhscdbt = to_num(THHSCDBT),
      rhhuscbt = to_num(RHHUSCBT),
      # car values: sum over up to 3 cars
```

```
      tcarval = to_num(TCARVAL1) +
                  to_num(TCARVAL2) +
                  to_num(TCARVAL3)
    )
}

assets_long <- bind_rows(
  make_assets_wave(t3, 3),
  make_assets_wave(t6, 6),
  make_assets_wave(t9, 9),
  make_assets_wave(t12, 12)
)

assets_wide <- assets_long %>%
  pivot_wider(
    id_cols = all_of(id_vars),
    names_from = wave,
    values_from = c(taltb, thhira, otherassets, thhscdbt, rhhuscbt, tcarval),
    names_sep = ""
  )
# This gives: taltb3, taltb6, taltb9, taltb12, etc.
# Wave 6 variables correspond to the Table 1 row (taltb6  401(k) assets).


## 3. Wave 7 core stuff (one record per person: SREFMON==4)


core7 <- w7 %>%
  filter(SREFMON == 4) %>% # this is how w7v2 is built in Stata
  transmute(
    SSUID, SHHADID, EENTAID, EPPPNUM,
    tage = to_num(TAGE),
    wpfinwgt = to_num(WPFINWGT),
    eclwrk1 = to_num(ECLWRK1),
    efnp = to_num(EFNP),
```

```
    esex = to_num(ESEX),
    tempall1 = to_num(TEMPALL1),
    ejbind1 = to_num(EJBIND1),
    tsjdate1 = to_num(TSJDATE1),
    srotaton = to_num(SROTATON)
  )


## 3a. Calculate Year 1 household income (sum of all 12 months)
## Wave 7 (months 1-4) + Wave 8 (months 1-4) + Wave 9 (months 1-4)


# Wave 7 income by reference month
w7_inc <- w7 %>%
  select(SSUID, SHHADID, EENTAID, EPPPNUM, SREFMON, THTOTINC) %>%
  mutate(
    THTOTINC = to_num(THTOTINC),
    month = paste0("w7_", SREFMON)
  ) %>%
  select(-SREFMON) %>%
  pivot_wider(names_from = month, values_from = THTOTINC, id_cols = all_of(id_vars))

# Wave 8 income by reference month
w8_inc <- w8 %>%
  select(SSUID, SHHADID, EENTAID, EPPPNUM, SREFMON, THTOTINC) %>%
  mutate(
    THTOTINC = to_num(THTOTINC),
    month = paste0("w8_", SREFMON)
  ) %>%
  select(-SREFMON) %>%
  pivot_wider(names_from = month, values_from = THTOTINC, id_cols = all_of(id_vars))

# Wave 9 income by reference month
w9_inc <- w9 %>%
  select(SSUID, SHHADID, EENTAID, EPPPNUM, SREFMON, THTOTINC) %>%
```

```
  mutate(
    THTOTINC = to_num(THTOTINC),
    month = paste0("w9_", SREFMON)
  ) %>%
  select(-SREFMON) %>%
  pivot_wider(names_from = month, values_from = THTOTINC, id_cols = all_of(id_vars))

# Merge all income months and sum to get Year 1 income
year1_income <- w7_inc %>%
  full_join(w8_inc, by = id_vars) %>%
  full_join(w9_inc, by = id_vars) %>%
  mutate(
    # Sum all 12 months (w7_1 through w7_4, w8_1 through w8_4, w9_1 through w9_4)
    hh_inc_year = rowSums(select(., starts_with("w7_"), starts_with("w8_"), starts_with("w9_")),
                         na.rm = FALSE) # Keep as NA if any month is missing
  ) %>%
  select(all_of(id_vars), hh_inc_year)


## 4. Topical module 7: 401(k) eligibility & reasons


tm7 <- t7 %>%
  transmute(
    SSUID, SHHADID, EENTAID, EPPPNUM,
    e1taxdef = to_num(E1TAXDEF),
    e2taxdef = to_num(E2TAXDEF),
    e3taxdef = to_num(E3TAXDEF),
    enoina03 = to_num(ENOINA03), # reason not covered: havent worked long enough (plan A)
    enoinb03 = to_num(ENOINB03), # reason not covered: havent worked long enough (plan B)
    etdeffen = to_num(ETDEFFEN) # firm offers any tax-deferred plan
  )

## 5. Merge assets + core + topical + Year 1 income into one person-level dataset

```

```

dat <- assets_wide %>%
  inner_join(core7, by = id_vars) %>%
  inner_join(tm7, by = id_vars) %>%
  left_join(year1_income, by = id_vars)

cat("Rows in merged dat: ", nrow(dat), "\n")


## 6. Reproduce Stata sample flags: yr1jb1, temp, y401k, notmissing


dat <- dat %>%
  mutate(
    # Age range: 2264 (matches: keep if tage>21 & tage<65)
    age_ok = !is.na(tage) & tage >= 22 & tage <= 64,

    # For-profit: eclwrk1 == 1
    for_profit = (eclwrk1 == 1),

    # yr1jb1: began current job within the past year at Wave 7 (Stata logic)
    yr1jb1 = case_when(
      srotaton == 1 ~ tsjdate1 > 19970299,
      srotaton == 2 ~ tsjdate1 > 19970399,
      srotaton == 3 ~ tsjdate1 > 19970499,
      srotaton == 4 ~ tsjdate1 > 19970599,
      TRUE ~ NA
    ),

    # Treatment: temp (exactly the Stata definition)
    temp = as.integer(
      (enoina03 == 1 & etdeffen == 1) | # "haven’t worked long enough" & firm offers plan
      (enoinb03 == 1)
    ),

```

```
    # y401k: firm offers 401(k) (Stata: y401k = temp | e1taxdef==1 | e2taxdef==1 | e3taxdef==1 | (
        etdeffen==1 & yr1jb1))
    y401k = (temp == 1) |
           (e1taxdef == 1) |
           (e2taxdef == 1) |
           (e3taxdef == 1) |
           (etdeffen == 1 & yr1jb1 == 1),

    # Calculate d21ltaltb exactly as in Stata: ln(taltb12+10) - 2*(ln(taltb9+10)) + ln(taltb6+10)
    d21ltaltb = log(taltb12 + 10) - 2 * log(taltb9 + 10) + log(taltb6 + 10),

    # "notmissing" is defined via d21ltaltb ~= . in Stata
    # Check if d21ltaltb is not missing (which happens when any component calculation fails)
    notmissing = !is.na(d21ltaltb),

    # Income missing dummy (Stata's incmissing); here we allow missing, we just flag it.
    incmissing = as.integer(is.na(hh_inc_year))
  )

cat("Counts: y401k TRUE/FALSE:\n")
print(table(dat$y401k, useNA = "ifany"))
cat("Counts: temp (treatment) 0/1:\n")
print(table(dat$temp, useNA = "ifany"))


## 7. Table 1 *sample* = Stata's "main" restrictions
## - age 2264
## - for-profit (eclwrk1==1)
## - yr1jb1 == 1 (started job within the previous year)
## - y401k == TRUE
## - notmissing == TRUE (has the data to construct d21ltaltb)


table1_sample <- dat %>%
  filter(
```

```
    age_ok,
    for_profit,
    yr1jb1 == 1,
    y401k,
    notmissing
  )

cat("Table 1 sample size: ", nrow(table1_sample), "\n")
cat("Treatment (temp==1): ", sum(table1_sample$temp == 1, na.rm = TRUE), "\n")
cat("Control (temp==0): ", sum(table1_sample$temp == 0, na.rm = TRUE), "\n")

## Separate N for *income* row (Stata has 818 instead of 835)
income_sample <- table1_sample %>%
  filter(!is.na(hh_inc_year))

cat("Income row N (non-missing hh_inc_year): ", nrow(income_sample), "\n")

## 8. Build Table 1 means/SDs (same variables as paper)


w_mean_sd <- function(x, w) {
  x <- as.numeric(x)
  w <- as.numeric(w)
  ok <- !is.na(x) & !is.na(w)
  x <- x[ok]; w <- w[ok]
  if (!length(x)) return(c(mean = NA_real_, sd = NA_real_))
  w <- w / sum(w)
  mu <- sum(w * x)
  var <- sum(w * (x - mu)^2)
  c(mean = mu, sd = sqrt(var))
}

make_block <- function(df) {
  out <- list(
    Age = w_mean_sd(df$tage, df$wpfinwgt),
```

```
    `Yearly household income` =
      w_mean_sd(df$hh_inc_year, df$wpfinwgt),
    `401(k) assets` =
      w_mean_sd(df$taltb6, df$wpfinwgt),
    `IRA and Keogh assets` =
      w_mean_sd(df$thhira6, df$wpfinwgt),
    `Other financial assets`=
      w_mean_sd(df$otherassets6, df$wpfinwgt),
    `Secured debt` =
      w_mean_sd(df$thhscdbt6, df$wpfinwgt),
    `Unsecured debt` =
      w_mean_sd(df$rhhuscbt6, df$wpfinwgt),
    `Car value` =
      w_mean_sd(df$tcarval6, df$wpfinwgt)
  )
  vals <- lapply(out, function(msd)
    sprintf("%.1f\n(%.1f)", msd["mean"], msd["sd"]))
  tibble::as_tibble_row(vals)
}

tab_all <- make_block(table1_sample) %>%
  mutate(group = "All",
         Observations = nrow(table1_sample))

tab_treat <- make_block(filter(table1_sample, temp == 1)) %>%
  mutate(group = "Treatment group",
         Observations = sum(table1_sample$temp == 1, na.rm = TRUE))

tab_ctrl <- make_block(filter(table1_sample, temp == 0)) %>%
  mutate(group = "Control group",
         Observations = sum(table1_sample$temp == 0, na.rm = TRUE))

table1 <- bind_rows(tab_all, tab_treat, tab_ctrl) %>%
  relocate(group, Observations)

```

~~~~
table1

```
```{r}
  pivot_wider(names_from = month, values_from = THTOTINC, id_cols = all_of(id_vars))

```
```{r}
library(dplyr)
library(tidyr)

to_num <- function(x) suppressWarnings(as.numeric(x))




id_vars <- c("SSUID","SHHADID","EENTAID","EPPPNUM")


## 2. Assets: build taltb6, thhira6, otherassets6, etc.


make_assets_wave <- function(df, wave_num) {
  df %>%
    transmute(
      SSUID, SHHADID, EENTAID, EPPPNUM,
      wave = wave_num,
      # 401(k) balance
      taltb = to_num(TALTB),
      # IRA balance
      thhira = to_num(THHIRA),
      # Other financial assets (interest, stocks, other)
      otherassets = to_num(THHINTBK) +
                    to_num(THHINTOT) +
                    to_num(RHHSTK) +
~~~~

```
                  to_num(THHOTAST),
      # secured & unsecured debt
      thhscdbt = to_num(THHSCDBT),
      rhhuscbt = to_num(RHHUSCBT),
      # car values: sum over up to 3 cars
      tcarval = to_num(TCARVAL1) +
                 to_num(TCARVAL2) +
                 to_num(TCARVAL3)
    )
}

assets_long <- bind_rows(
  make_assets_wave(t3, 3),
  make_assets_wave(t6, 6),
  make_assets_wave(t9, 9),
  make_assets_wave(t12, 12)
)

assets_wide <- assets_long %>%
  pivot_wider(
    id_cols = all_of(id_vars),
    names_from = wave,
    values_from = c(taltb, thhira, otherassets, thhscdbt, rhhuscbt, tcarval),
    names_sep = ""
  )
# Gives: taltb3, taltb6, taltb9, taltb12, etc.


## 3. Wave 7 core (SREFMON==4 only) + RMESR (employment status)


core7 <- w7 %>%
  filter(SREFMON == 4) %>% # rotation 4
  transmute(
    SSUID, SHHADID, EENTAID, EPPPNUM,
```

```
    tage = to_num(TAGE),
    wpfinwgt = to_num(WPFINWGT),
    eclwrk1 = to_num(ECLWRK1),
    efnp = to_num(EFNP),
    esex = to_num(ESEX),
    tempall1 = to_num(TEMPALL1),
    ejbind1 = to_num(EJBIND1),
    tsjdate1 = to_num(TSJDATE1),
    srotaton = to_num(SROTATON),
    rmesr = to_num(RMESR), # <-- key new variable
    # Year-1 household income proxy
    hh_inc_year = to_num(THTOTINC)
  )


## 4. Topical module 7: 401(k) eligibility & reasons


tm7 <- t7 %>%
  transmute(
    SSUID, SHHADID, EENTAID, EPPPNUM,
    e1taxdef = to_num(E1TAXDEF),
    e2taxdef = to_num(E2TAXDEF),
    e3taxdef = to_num(E3TAXDEF),
    enoina03 = to_num(ENOINA03), # reason: haven't worked long enough (plan A)
    enoinb03 = to_num(ENOINB03), # reason: haven't worked long enough (plan B)
    etdeffen = to_num(ETDEFFEN) # firm offers any tax-deferred plan
  )


## 5. Merge assets + core + topical


dat <- assets_wide %>%
  inner_join(core7, by = id_vars) %>%
```

```
  inner_join(tm7, by = id_vars)

cat("Rows in merged dat: ", nrow(dat), "\n")


## 6. Flags: age_ok, for_profit, yr1jb1, temp, y401k, notmissing


dat <- dat %>%
  mutate(
    # Age 2264
    age_ok = !is.na(tage) & tage >= 22 & tage <= 64,

    # For-profit: ECLWRK1 == 1
    for_profit = (eclwrk1 == 1),

    # Started job within previous year (Wave 7 rotation logic)
    yr1jb1 = case_when(
      srotaton == 1 ~ tsjdate1 > 19970299,
      srotaton == 2 ~ tsjdate1 > 19970399,
      srotaton == 3 ~ tsjdate1 > 19970499,
      srotaton == 4 ~ tsjdate1 > 19970599,
      TRUE ~ NA
    ),

    # Treatment: "havent worked long enough" & firm offers plan
    temp = as.integer(
      (enoina03 == 1 & etdeffen == 1) |
      (enoinb03 == 1)
    ),

    # Firm offers 401(k)
    y401k = (temp == 1) |
            (e1taxdef == 1) |
            (e2taxdef == 1) |
```

```
          (e3taxdef == 1) |
          (etdeffen == 1 & yr1jb1 == 1),

    # Notmissing asset growth (based on 6/9/12 balances)
    notmissing = !is.na(taltb6) & !is.na(taltb9) & !is.na(taltb12),

    # Income missing dummy
    incmissing = as.integer(is.na(hh_inc_year))
  )

cat("Counts: y401k TRUE/FALSE/NA:\n")
print(table(dat$y401k, useNA = "ifany"))
cat("Counts: temp (treatment) 0/1:\n")
print(table(dat$temp, useNA = "ifany"))


## 7. Table 1 sample + RMESR filter (employment status)

## Stata analogue: keep if rmesr <= 4 (in labor force / recently worked)


table1_sample <- dat %>%
  filter(
    age_ok, # 2264
    for_profit, # for-profit firm
    yr1jb1 == 1, # started job within previous year
    y401k, # firm offers 401(k)
    notmissing, # have assets in 6/9/12
    !is.na(rmesr),
    rmesr <= 4 # <-- NEW: match authors labor-force restriction
  )

cat("Table 1 sample size: ", nrow(table1_sample), "\n")
cat("Treatment (temp==1): ", sum(table1_sample$temp == 1, na.rm = TRUE), "\n")
cat("Control (temp==0): ", sum(table1_sample$temp == 0, na.rm = TRUE), "\n")
```

```
income_sample <- table1_sample %>%
  filter(!is.na(hh_inc_year))

cat("Income row N (non-missing hh_inc_year): ", nrow(income_sample), "\n")


## 8. Build Table 1 means/SDs


w_mean_sd <- function(x, w) {
  x <- as.numeric(x)
  w <- as.numeric(w)
  ok <- !is.na(x) & !is.na(w)
  x <- x[ok]; w <- w[ok]
  if (!length(x)) return(c(mean = NA_real_, sd = NA_real_))
  w <- w / sum(w)
  mu <- sum(w * x)
  var <- sum(w * (x - mu)^2)
  c(mean = mu, sd = sqrt(var))
}

make_block <- function(df) {
  out <- list(
    Age = w_mean_sd(df$tage, df$wpfinwgt),
    `Yearly household income` =
      w_mean_sd(df$hh_inc_year, df$wpfinwgt),
    `401(k) assets` =
      w_mean_sd(df$taltb6, df$wpfinwgt),
    `IRA and Keogh assets` =
      w_mean_sd(df$thhira6, df$wpfinwgt),
    `Other financial assets`=
      w_mean_sd(df$otherassets6, df$wpfinwgt),
    `Secured debt` =
      w_mean_sd(df$thhscdbt6, df$wpfinwgt),
```

```
    `Unsecured debt` =
      w_mean_sd(df$rhhuscbt6, df$wpfinwgt),
    `Car value` =
      w_mean_sd(df$tcarval6, df$wpfinwgt)
  )
  vals <- lapply(out, function(msd)
    sprintf("%.1f\n(%.1f)", msd["mean"], msd["sd"]))
  tibble::as_tibble_row(vals)
}

tab_all <- make_block(table1_sample) %>%
  mutate(group = "All",
         Observations = nrow(table1_sample))

tab_treat <- make_block(filter(table1_sample, temp == 1)) %>%
  mutate(group = "Treatment group",
         Observations = sum(table1_sample$temp == 1, na.rm = TRUE))

tab_ctrl <- make_block(filter(table1_sample, temp == 0)) %>%
  mutate(group = "Control group",
         Observations = sum(table1_sample$temp == 0, na.rm = TRUE))

table1 <- bind_rows(tab_all, tab_treat, tab_ctrl) %>%
  relocate(group, Observations)

table1

```
```

B.4 Table 1 from the other(table1_compare.qmd)

```
---
title: "Untitled"
format: html
---
```
```

````

```{r}




# 1. LOAD THE DATA (RAW)


t3_raw <- readRDS("data/raw/rds_1/t3.rds")
t6_raw <- readRDS("data/raw/rds_1/t6.rds")
t9_raw <- readRDS("data/raw/rds_1/t9.rds")
t12_raw <- readRDS("data/raw/rds_1/t12.rds")

t7_raw <- readRDS("data/raw/rds_1/t7.rds")
w7_raw <- readRDS("data/raw/rds_1/w7.rds")
w8_raw <- readRDS("data/raw/rds_1/w8.rds")
w9_raw <- readRDS("data/raw/rds_1/w9.rds")


```
```{r}
library(dplyr)
library(tidyr)
library(haven)

to_num <- function(x) suppressWarnings(as.numeric(x))


# 1. LOAD THE DATA *FIRST*


t3_raw <- readRDS("data/raw/rds_1/t3.rds")
t6_raw <- readRDS("data/raw/rds_1/t6.rds")
t9_raw <- readRDS("data/raw/rds_1/t9.rds")
````
````

````r
t12_raw <- readRDS("data/raw/rds_1/t12.rds")

t7_raw <- readRDS("data/raw/rds_1/t7.rds")
w7_raw <- readRDS("data/raw/rds_1/w7.rds")
w8_raw <- readRDS("data/raw/rds_1/w8.rds")
w9_raw <- readRDS("data/raw/rds_1/w9.rds")


# 2. LOAD REPLICATION FILES


t3_rep <- read_dta("data/replication/dta/t3.dta") %>% rename_with(toupper)
t6_rep <- read_dta("data/replication/dta/t6.dta") %>% rename_with(toupper)
t9_rep <- read_dta("data/replication/dta/t9.dta") %>% rename_with(toupper)
t12_rep <- read_dta("data/replication/dta/t12.dta") %>% rename_with(toupper)

t7_rep <- read_dta("data/replication/dta/t7.dta") %>% rename_with(toupper)
w7_rep <- read_dta("data/replication/dta/w7.dta") %>% rename_with(toupper)
w8_rep <- read_dta("data/replication/dta/w8.dta") %>% rename_with(toupper)
w9_rep <- read_dta("data/replication/dta/w9.dta") %>% rename_with(toupper)



```
```{r}
w9_rep <- read_dta(
  "data/replication/dta/w9.dta",
  col_select = c("ssuid", "shhadid", "eentaid", "epppnum", "srefmon", "thtotinc")
) %>%
  rename_with(toupper)
w7_rep <- read_dta(
  "data/replication/dta/w7.dta",
  col_select = c("ssuid","shhadid","eentaid","epppnum",
               "srefmon","thtotinc","tage","wpfinwgt",
               "eclwrk1","efnp","esex","tempall1",
````

````
                "ejbind1","tsjdate1","srotaton")
) %>% rename_with(toupper)

w8_rep <- read_dta(
  "data/replication/dta/w8.dta",
  col_select = c("ssuid","shhadid","eentaid","epppnum","srefmon","thtotinc")
) %>% rename_with(toupper)


```

```{r}
library(dplyr)
library(tidyr)


## Helper: safe numeric conversion

to_num <- function(x) suppressWarnings(as.numeric(x))


## Helper: build assets for one wave

make_assets_wave <- function(df, wave_num) {
  df %>%
    transmute(
      SSUID, SHHADID, EENTAID, EPPPNUM,
      wave = wave_num,
      # 401(k) balance
      taltb = to_num(TALTB),
      # IRA balance
      thhira = to_num(THHIRA),
      # Other financial assets (interest, stocks, other assets)
      otherassets = to_num(THHINTBK) +
                    to_num(THHINTOT) +
````

```
                   to_num(RHHSTK) +
                   to_num(THHOTAST),
      # secured & unsecured debt
      thhscdbt = to_num(THHSCDBT),
      rhhuscbt = to_num(RHHUSCBT),
      # car values: sum over up to 3 cars
      tcarval = to_num(TCARVAL1) +
                  to_num(TCARVAL2) +
                  to_num(TCARVAL3)
    )
}


## MAIN: build_dat()
## takes t3,t6,t9,t12,t7,w7,w8,w9 and returns merged dataset
## with: age_ok, for_profit, yr1jb1, temp, y401k, notmissing,
## hh_inc_year, etc.

build_dat <- function(t3, t6, t9, t12, t7, w7, w8, w9) {
  id_vars <- c("SSUID","SHHADID","EENTAID","EPPPNUM")

  # 1) Assets across waves  wide
  assets_long <- bind_rows(
    make_assets_wave(t3, 3),
    make_assets_wave(t6, 6),
    make_assets_wave(t9, 9),
    make_assets_wave(t12, 12)
  )

  assets_wide <- assets_long %>%
    pivot_wider(
      id_cols = all_of(id_vars),
      names_from = wave,
      values_from = c(taltb, thhira, otherassets, thhscdbt, rhhuscbt, tcarval),
      names_sep = ""
```

```
  )

# 2) Wave-7 core (panel = SREFMON==4)
core7 <- w7 %>%
  filter(SREFMON == 4) %>%
  transmute(
    SSUID, SHHADID, EENTAID, EPPPNUM,
    tage = to_num(TAGE),
    wpfinwgt = to_num(WPFINWGT),
    eclwrk1 = to_num(ECLWRK1),
    efnp = to_num(EFNP),
    esex = to_num(ESEX),
    tempall1 = to_num(TEMPALL1),
    ejbind1 = to_num(EJBIND1),
    tsjdate1 = to_num(TSJDATE1),
    srotaton = to_num(SROTATON)
  )

# 3) Year-1 income: sum THTOTINC from waves 79
make_inc <- function(w, tag) {
  w %>%
    select(SSUID, SHHADID, EENTAID, EPPPNUM, SREFMON, THTOTINC) %>%
    mutate(
      THTOTINC = to_num(THTOTINC),
      month = paste0(tag, "_", SREFMON)
    ) %>%
    select(-SREFMON) %>%
    pivot_wider(
      id_cols = all_of(id_vars),
      names_from = month,
      values_from = THTOTINC
    )
}

w7_inc <- make_inc(w7, "w7")
```

```
  w8_inc <- make_inc(w8, "w8")
  w9_inc <- make_inc(w9, "w9")

  year1_income <- w7_inc %>%
    full_join(w8_inc, by = id_vars) %>%
    full_join(w9_inc, by = id_vars) %>%
    mutate(
      hh_inc_year = rowSums(
        dplyr::select(., starts_with("w7_"), starts_with("w8_"), starts_with("w9_")),
        na.rm = FALSE
      )
    ) %>%
    select(all_of(id_vars), hh_inc_year)

  # 4) Topical module 7: plan eligibility and reasons
  tm7 <- t7 %>%
    transmute(
      SSUID, SHHADID, EENTAID, EPPPNUM,
      e1taxdef = to_num(E1TAXDEF),
      e2taxdef = to_num(E2TAXDEF),
      e3taxdef = to_num(E3TAXDEF),
      enoina03 = to_num(ENOINA03),
      enoinb03 = to_num(ENOINB03),
      etdeffen = to_num(ETDEFFEN)
    )

  # 5) Merge all pieces
  dat <- assets_wide %>%
    inner_join(core7, by = id_vars) %>%
    inner_join(tm7, by = id_vars) %>%
    left_join(year1_income, by = id_vars)

  # 6) Derive sample flags and d21ltaltb
 dat <- dat %>%
  mutate(
```

```
  age_ok = !is.na(tage) & tage >= 22 & tage <= 64,
  for_profit = (eclwrk1 == 1),
  yr1jb1 = dplyr::case_when(
    srotaton == 1 ~ tsjdate1 > 19970299,
    srotaton == 2 ~ tsjdate1 > 19970399,
    srotaton == 3 ~ tsjdate1 > 19970499,
    srotaton == 4 ~ tsjdate1 > 19970599,
    TRUE ~ NA
  ),
  temp = as.integer(
    (enoina03 == 1 & etdeffen == 1) |
    (enoinb03 == 1)
  ),
  y401k = (temp == 1) |
          (e1taxdef == 1) |
          (e2taxdef == 1) |
          (e3taxdef == 1) |
          (etdeffen == 1 & yr1jb1 == 1),

  d21ltaltb = log(taltb12 + 10) - 2 * log(taltb9 + 10) + log(taltb6 + 10),
  lnA6 = log(taltb6 + 10),

  # income: allowed to be missing, we flag it
  incmissing = as.integer(is.na(hh_inc_year)),

  # tweak: require regression vars (except hh_inc_year) to be non-missing
  notmissing = !is.na(d21ltaltb) &
               !is.na(tage) &
               !is.na(eclwrk1) &
               !is.na(tsjdate1) &
               !is.na(srotaton) &
               !is.na(wpfinwgt)
)

dat
```

```
}


## Helpers to build Table 1 from a built dat

w_mean_sd <- function(x, w) {
  x <- as.numeric(x)
  w <- as.numeric(w)
  ok <- !is.na(x) & !is.na(w)
  x <- x[ok]; w <- w[ok]
  if (!length(x)) return(c(mean = NA_real_, sd = NA_real_))
  w <- w / sum(w)
  mu <- sum(w * x)
  var <- sum(w * (x - mu)^2)
  c(mean = mu, sd = sqrt(var))
}

make_block <- function(df) {
  out <- list(
    Age = w_mean_sd(df$tage, df$wpfinwgt),
    `Yearly household income` =
      w_mean_sd(df$hh_inc_year, df$wpfinwgt),
    `401(k) assets` =
      w_mean_sd(df$taltb6, df$wpfinwgt),
    `IRA and Keogh assets` =
      w_mean_sd(df$thhira6, df$wpfinwgt),
    `Other financial assets`=
      w_mean_sd(df$otherassets6, df$wpfinwgt),
    `Secured debt` =
      w_mean_sd(df$thhscdbt6, df$wpfinwgt),
    `Unsecured debt` =
      w_mean_sd(df$rhhuscbt6, df$wpfinwgt),
    `Car value` =
      w_mean_sd(df$tcarval6, df$wpfinwgt)
  )
```

```
  vals <- lapply(out, function(msd)
    sprintf("%.1f\n(%.1f)", msd["mean"], msd["sd"]))
  tibble::as_tibble_row(vals)
}

make_table1 <- function(dat) {
  # Apply same sample restrictions
  table1_sample <- dat %>%
    filter(
      age_ok,
      for_profit,
      yr1jb1 == 1,
      y401k,
      notmissing
    )

  cat("Table 1 sample size: ", nrow(table1_sample), "\n")
  cat("Treatment (temp==1): ", sum(table1_sample$temp == 1, na.rm = TRUE), "\n")
  cat("Control (temp==0): ", sum(table1_sample$temp == 0, na.rm = TRUE), "\n")

  income_sample <- table1_sample %>%
    filter(!is.na(hh_inc_year))
  cat("Income row N (non-missing hh_inc_year): ", nrow(income_sample), "\n")

  tab_all <- make_block(table1_sample) %>%
    mutate(group = "All",
           Observations = nrow(table1_sample))

  tab_treat <- make_block(filter(table1_sample, temp == 1)) %>%
    mutate(group = "Treatment group",
           Observations = sum(table1_sample$temp == 1, na.rm = TRUE))

  tab_ctrl <- make_block(filter(table1_sample, temp == 0)) %>%
    mutate(group = "Control group",
           Observations = sum(table1_sample$temp == 0, na.rm = TRUE))
```

````

  bind_rows(tab_all, tab_treat, tab_ctrl) %>%
    relocate(group, Observations)
}

```

```{r}

dat_raw <- build_dat(
  t3_raw, t6_raw, t9_raw, t12_raw,
  t7_raw, w7_raw, w8_raw, w9_raw
)

table1_raw <- make_table1(dat_raw)
table1_raw
dat_rep <- build_dat(
  t3_rep, t6_rep, t9_rep, t12_rep,
  t7_rep, w7_rep, w8_rep, w9_rep
)

table1_rep <- make_table1(dat_rep)
table1_rep

```


```{r}
id_vars <- c("SSUID","SHHADID","EENTAID","EPPPNUM")

normalize_ids <- function(df) {
  df %>%
    mutate(
      SSUID = sub("^0+", "", as.character(SSUID)),
      SHHADID = sub("^0+", "", as.character(SHHADID)),
````

````
      EENTAID = sub("^0+", "", as.character(EENTAID)),
      EPPPNUM = sub("^0+", "", as.character(EPPPNUM))
    )
}


```
```{r}
# Rebuild from scratch using existing build_dat()
dat_raw <- build_dat(
  t3_raw, t6_raw, t9_raw, t12_raw,
  t7_raw, w7_raw, w8_raw, w9_raw
)

dat_rep <- build_dat(
  t3_rep, t6_rep, t9_rep, t12_rep,
  t7_rep, w7_rep, w8_rep, w9_rep
)

# Normalize IDs
dat_raw_id <- normalize_ids(dat_raw)
dat_rep_id <- normalize_ids(dat_rep)

# Check ID sets really match
raw_ids <- dat_raw_id %>% select(all_of(id_vars)) %>% distinct()
rep_ids <- dat_rep_id %>% select(all_of(id_vars)) %>% distinct()

rep_not_raw <- dplyr::anti_join(rep_ids, raw_ids, by = id_vars)
raw_not_rep <- dplyr::anti_join(raw_ids, rep_ids, by = id_vars)

nrow(rep_not_raw) # should be 0
nrow(raw_not_rep) # should be 0

```
````
````

### B.5 Table 2 (Raw) (table2_raw.qmd)

```
---
title: "Untitled"
format: html
---


```{r}


library(dplyr)

main <- dat_raw_aligned #  NOW USING dat_raw_aligned, but switch to data_raw to use 1115 sample size

to_num <- function(x) suppressWarnings(as.numeric(x))

safe_log <- function(x) {
  ifelse(!is.na(x) & x > -10, log(x + 10), NA_real_)
}

main <- main %>%
  mutate(
    weight = wpfinwgt,
    tagesq = tage^2,

    ## log levels at wave 6
    ltaltb6 = safe_log(taltb6),
    lthhira6 = safe_log(thhira6),
    lotherassets6 = safe_log(otherassets6),
    lthhscdbt6 = safe_log(thhscdbt6),
    lrhhuscbt6 = safe_log(rhhuscbt6),
    ltcarval6 = safe_log(tcarval6),

    ## second differences (12  29 + 6)
    d21lthhira = safe_log(thhira12) - 2*safe_log(thhira9) + safe_log(thhira6),
```
```

````
    d21lotherassets = safe_log(otherassets12) - 2*safe_log(otherassets9) + safe_log(otherassets6),
    d21lthhscdbt = safe_log(thhscdbt12) - 2*safe_log(thhscdbt9) + safe_log(thhscdbt6),
    d21lrhhuscbt = safe_log(rhhuscbt12) - 2*safe_log(rhhuscbt9) + safe_log(rhhuscbt6),
    d21ltcarval = safe_log(tcarval12) - 2*safe_log(tcarval9) + safe_log(tcarval6)
) %>%
  ## apply papers required sample restrictions
  filter(
    age_ok,
    for_profit,
    yr1jb1
)

```
```{r}

library(sandwich)
library(lmtest)

run_cluster_reg <- function(df, formula) {

  df <- df %>%
    dplyr::group_by(SSUID, SHHADID) %>%
    dplyr::mutate(hid = dplyr::cur_group_id()) %>%
    dplyr::ungroup()

  mod <- lm(formula, data = df, weights = df$weight)

  vc <- sandwich::vcovCL(mod, cluster = df$hid)
  ct <- lmtest::coeftest(mod, vcov. = vc)

  list(model = mod, coeftest = ct)
}

```
```{r}
````

```
var_roots <- c("taltb", "thhira", "otherassets",
               "thhscdbt", "rhhuscbt", "tcarval")

table2_results <- lapply(var_roots, function(vr) {

  dv <- paste0("d21l", vr)
  lag6 <- paste0("l", vr, "6")

  df <- main %>%
    filter(
      y401k == 1,
      notmissing,
      !is.na(.data[[dv]])
    )

  ## Formulae:
  f1 <- reformulate("temp", response = dv)

  covars2 <- c("temp", "tage", "tagesq", "hh_inc_year", "incmissing")
  f2 <- reformulate(covars2, response = dv)

  covars3 <- c("temp", lag6, "tage", "tagesq", "hh_inc_year", "incmissing")
  f3 <- reformulate(covars3, response = dv)

  s1 <- run_cluster_reg(df, f1)
  s2 <- run_cluster_reg(df, f2)
  s3 <- run_cluster_reg(df, f3)

  extract_temp <- function(ct) {
    if (!"temp" %in% rownames(ct)) return(c(coef=NA, se=NA))
    c(coef = ct["temp","Estimate"],
      se = ct["temp","Std. Error"])
  }
```

````

  c1 <- extract_temp(s1$coeftest)
  c2 <- extract_temp(s2$coeftest)
  c3 <- extract_temp(s3$coeftest)

  data.frame(
    outcome = vr,
    spec = c("no_controls", "controls", "controls_plus_lag"),
    coef_temp = c(c1["coef"], c2["coef"], c3["coef"]),
    se_temp = c(c1["se"], c2["se"], c3["se"]),
    n = c(nobs(s1$model), nobs(s2$model), nobs(s3$model))
  )
})

```

```{r}


table2_raw <- bind_rows(table2_results)
table2_raw

```
```{r}


library(dplyr)

# Rename columns to avoid name collisions
t_rep <- table2 %>%
  rename(
    coef_rep = coef_temp,
    se_rep = se_temp,
    N_rep = n
  )
````

````

t_raw <- table2_raw %>%
  rename(
    coef_raw = coef_temp,
    se_raw = se_temp,
    N_raw = n
  )

# Merge side-by-side by outcome + spec
table2_compare <- t_rep %>%
  inner_join(t_raw, by = c("outcome", "spec")) %>%
  arrange(outcome, spec)

table2_compare

```
````

B.6 Table 2 (Raw Aligned) (table2_raw_aligned.qmd)

````
---
title: "Untitled"
format: html
---

```{r}
library(dplyr)

main <- dat_raw_aligned #  NOW USING dat_raw_aligned, but switch to data_raw to use 1115 sample size

to_num <- function(x) suppressWarnings(as.numeric(x))

safe_log <- function(x) {
  ifelse(!is.na(x) & x > -10, log(x + 10), NA_real_)
}

````

~~~
main <- main %>%
  mutate(
    weight = wpfinwgt,
    tagesq = tage^2,

    ## log levels at wave 6
    ltaltb6 = safe_log(taltb6),
    lthhira6 = safe_log(thhira6),
    lotherassets6 = safe_log(otherassets6),
    lthhscdbt6 = safe_log(thhscdbt6),
    lrhhuscbt6 = safe_log(rhhuscbt6),
    ltcarval6 = safe_log(tcarval6),

    ## second differences (12  29 + 6)
    d21lthhira = safe_log(thhira12) - 2*safe_log(thhira9) + safe_log(thhira6),
    d21lotherassets = safe_log(otherassets12) - 2*safe_log(otherassets9) + safe_log(otherassets6),
    d21lthhscdbt = safe_log(thhscdbt12) - 2*safe_log(thhscdbt9) + safe_log(thhscdbt6),
    d21lrhhuscbt = safe_log(rhhuscbt12) - 2*safe_log(rhhuscbt9) + safe_log(rhhuscbt6),
    d21ltcarval = safe_log(tcarval12) - 2*safe_log(tcarval9) + safe_log(tcarval6)
  ) %>%
  ## apply papers required sample restrictions
  filter(
    age_ok,
    for_profit,
    yr1jb1
  )

```
```{r}

library(sandwich)
library(lmtest)

run_cluster_reg <- function(df, formula) {

~~~

````
  df <- df %>%
    dplyr::group_by(SSUID, SHHADID) %>%
    dplyr::mutate(hid = dplyr::cur_group_id()) %>%
    dplyr::ungroup()

  mod <- lm(formula, data = df, weights = df$weight)

  vc <- sandwich::vcovCL(mod, cluster = df$hid)
  ct <- lmtest::coeftest(mod, vcov. = vc)

  list(model = mod, coeftest = ct)
}

```
```{r}


var_roots <- c("taltb", "thhira", "otherassets",
               "thhscdbt", "rhhuscbt", "tcarval")

table2_results <- lapply(var_roots, function(vr) {

  dv <- paste0("d21l", vr)
  lag6 <- paste0("l", vr, "6")

  df <- main %>%
    filter(
      y401k == 1,
      notmissing,
      !is.na(.data[[dv]])
    )

  ## Formulae:
  f1 <- reformulate("temp", response = dv)

````

````
  covars2 <- c("temp", "tage", "tagesq", "hh_inc_year", "incmissing")
  f2 <- reformulate(covars2, response = dv)

  covars3 <- c("temp", lag6, "tage", "tagesq", "hh_inc_year", "incmissing")
  f3 <- reformulate(covars3, response = dv)

  s1 <- run_cluster_reg(df, f1)
  s2 <- run_cluster_reg(df, f2)
  s3 <- run_cluster_reg(df, f3)

  extract_temp <- function(ct) {
    if (!"temp" %in% rownames(ct)) return(c(coef=NA, se=NA))
    c(coef = ct["temp","Estimate"],
      se = ct["temp","Std. Error"])
  }

  c1 <- extract_temp(s1$coeftest)
  c2 <- extract_temp(s2$coeftest)
  c3 <- extract_temp(s3$coeftest)

  data.frame(
    outcome = vr,
    spec = c("no_controls", "controls", "controls_plus_lag"),
    coef_temp = c(c1["coef"], c2["coef"], c3["coef"]),
    se_temp = c(c1["se"], c2["se"], c3["se"]),
    n = c(nobs(s1$model), nobs(s2$model), nobs(s3$model))
  )
})

```

```{r}


table2_raw_aligned <- bind_rows(table2_results)
````

~~~~
table2_raw_aligned

```
```{r}



library(dplyr)

# Rename columns to avoid name collisions
t_rep <- table2 %>%
  rename(
    coef_rep = coef_temp,
    se_rep = se_temp,
    N_rep = n
  )

t_raw <- table2_raw %>%
  rename(
    coef_raw = coef_temp,
    se_raw = se_temp,
    N_raw = n
  )

# Merge side-by-side by outcome + spec
table2_compare <- t_rep %>%
  inner_join(t_raw, by = c("outcome", "spec")) %>%
  arrange(outcome, spec)

table2_compare
~~~~

B.7 Table 2 (Replication) (table2_rep.qmd)

~~~~
---
title: "Untitled"
format: html
~~~~

```
---

```{r}


library(dplyr)

main <- dat_rep # use dat_rep as the analysis dataset

to_num <- function(x) suppressWarnings(as.numeric(x))

safe_log <- function(x) {
  ifelse(!is.na(x) & x > -10, log(x + 10), NA_real_)
}

main <- main %>%
  mutate(
    weight = wpfinwgt,
    tagesq = tage^2,

    ## log levels at wave 6
    ltaltb6 = safe_log(taltb6),
    lthhira6 = safe_log(thhira6),
    lotherassets6 = safe_log(otherassets6),
    lthhscdbt6 = safe_log(thhscdbt6),
    lrhhuscbt6 = safe_log(rhhuscbt6),
    ltcarval6 = safe_log(tcarval6),

    ## second differences
    d21lthhira = safe_log(thhira12) - 2*safe_log(thhira9) + safe_log(thhira6),
    d21lotherassets = safe_log(otherassets12) - 2*safe_log(otherassets9) + safe_log(otherassets6),
    d21lthhscdbt = safe_log(thhscdbt12) - 2*safe_log(thhscdbt9) + safe_log(thhscdbt6),
    d21lrhhuscbt = safe_log(rhhuscbt12) - 2*safe_log(rhhuscbt9) + safe_log(rhhuscbt6),
    d21ltcarval = safe_log(tcarval12) - 2*safe_log(tcarval9) + safe_log(tcarval6)
  )
```
```

````
Restrict to the papers working sample
main <- main %>%
  filter(
    age_ok,
    for_profit,
    yr1jb1
)


```

```{r}


library(sandwich)
library(lmtest)

run_cluster_reg <- function(df, formula) {

  df <- df %>%
    dplyr::group_by(SSUID, SHHADID) %>%
    dplyr::mutate(hid = dplyr::cur_group_id()) %>%
    dplyr::ungroup()

  mod <- lm(formula, data = df, weights = df$weight)

  vc <- sandwich::vcovCL(mod, cluster = df$hid)
  ct <- lmtest::coeftest(mod, vcov. = vc)

  list(model = mod, coeftest = ct)
}

```
````
````

```
```{r}


var_roots <- c("taltb", "thhira", "otherassets",
               "thhscdbt", "rhhuscbt", "tcarval")

table2_results <- lapply(var_roots, function(vr) {

  dv <- paste0("d21l", vr)
  lag6 <- paste0("l", vr, "6")

  df <- main %>%
    filter(
      y401k == 1,
      notmissing,
      !is.na(.data[[dv]])
    )

  ## Spec 1
  f1 <- reformulate("temp", response = dv)

  ## Spec 2
  covars2 <- c("temp", "tage", "tagesq", "hh_inc_year", "incmissing")
  f2 <- reformulate(covars2, response = dv)

  ## Spec 3
  covars3 <- c("temp", lag6, "tage", "tagesq", "hh_inc_year", "incmissing")
  f3 <- reformulate(covars3, response = dv)

  s1 <- run_cluster_reg(df, f1)
  s2 <- run_cluster_reg(df, f2)
  s3 <- run_cluster_reg(df, f3)

  extract_temp <- function(ct) {
    if (!"temp" %in% rownames(ct)) return(c(coef=NA, se=NA))
```
```

````
    c(coef = ct["temp", "Estimate"],
      se = ct["temp", "Std. Error"])
  }

  c1 <- extract_temp(s1$coeftest)
  c2 <- extract_temp(s2$coeftest)
  c3 <- extract_temp(s3$coeftest)

  data.frame(
    outcome = vr,
    spec = c("no_controls", "controls", "controls_plus_lag"),
    coef_temp = c(c1["coef"], c2["coef"], c3["coef"]),
    se_temp = c(c1["se"], c2["se"], c3["se"]),
    n = c(nobs(s1$model), nobs(s2$model), nobs(s3$model))
)
})

```
```{r}


table2 <- bind_rows(table2_results)
table2

```
````

B.8 Table 3 Robustness Checks (table3_robustness_check.qmd)

````
---
title: "Table 3
format: pdf
---

```{r}
library(dplyr)
````

```
library(splines)
library(purrr)
library(tibble)

## ---- helpers ----

make_d21 <- function(df, base) {
  a6 <- df[[paste0(base, "6")]]
  a9 <- df[[paste0(base, "9")]]
  a12 <- df[[paste0(base, "12")]]
  log(a12 + 10) - 2 * log(a9 + 10) + log(a6 + 10)
}

make_d21_ihs <- function(df, base) {
  a6 <- df[[paste0(base, "6")]]
  a9 <- df[[paste0(base, "9")]]
  a12 <- df[[paste0(base, "12")]]
  asinh(a12 + 10) - 2 * asinh(a9 + 10) + asinh(a6 + 10)
}

# same sample as main analysis
make_reg_sample <- function(dat) {
  dat %>%
    filter(
      age_ok,
      for_profit,
      yr1jb1 == 1,
      y401k,
      notmissing
    )
}

grab_become <- function(fit, name = "temp") {
  s <- summary(fit)$coef
  tibble(
```

```
    estimate = s[name, "Estimate"],
    se = s[name, "Std. Error"]
  )
}

## Panel A: temp + spline in Wave 6 + controls
run_panelA <- function(df, dep, base6) {
  f <- as.formula(
    paste0(
      dep, " ~ temp + bs(", base6, ", df = 20) + ",
      "tage + I(tage^2) + hh_inc_year + incmissing"
    )
  )
  lm(f, data = df, weights = wpfinwgt)
}

## Panel B: temp + Wave6 + temp:Wave6 + controls
run_panelB <- function(df, dep, base6) {
  f <- as.formula(
    paste0(
      dep, " ~ temp + ", base6, " + temp:", base6, " + ",
      "tage + I(tage^2) + hh_inc_year + incmissing"
    )
  )
  lm(f, data = df, weights = wpfinwgt)
}

## Panel C: IHS outcome, temp + log(A6) + controls
run_panelC <- function(df, dep_ihs, base6) {
  f <- as.formula(
    paste0(
      dep_ihs, " ~ temp + log(", base6, " + 10) + ",
      "tage + I(tage^2) + hh_inc_year + incmissing"
    )
  )
```

```r
  lm(f, data = df, weights = wpfinwgt)
}

## Table 3 rep func

run_table3_rep <- function(dat_rep) {
  df <- make_reg_sample(dat_rep)

  # build outcomes (log 2nd diffs)
  df$d21_401k <- make_d21(df, "taltb")
  df$d21_ira <- make_d21(df, "thhira")
  df$d21_other <- make_d21(df, "otherassets")
  df$d21_sec <- make_d21(df, "thhscdbt")
  df$d21_unsec <- make_d21(df, "rhhuscbt")
  df$d21_car <- make_d21(df, "tcarval")

  # IHS for Panel C
  df$d21_401k_ihs <- make_d21_ihs(df, "taltb")
  df$d21_ira_ihs <- make_d21_ihs(df, "thhira")
  df$d21_other_ihs <- make_d21_ihs(df, "otherassets")
  df$d21_sec_ihs <- make_d21_ihs(df, "thhscdbt")
  df$d21_unsec_ihs <- make_d21_ihs(df, "rhhuscbt")
  df$d21_car_ihs <- make_d21_ihs(df, "tcarval")

  outcomes <- c("d21_401k","d21_ira","d21_other",
                "d21_sec","d21_unsec","d21_car")
  bases6 <- c("taltb6","thhira6","otherassets6",
                "thhscdbt6","rhhuscbt6","tcarval6")
  ihs_out <- c("d21_401k_ihs","d21_ira_ihs","d21_other_ihs",
                "d21_sec_ihs","d21_unsec_ihs","d21_car_ihs")

  ## Panel A
  fitsA <- Map(function(y, b6) run_panelA(df, y, b6), outcomes, bases6)
  panelA <- map_dfr(fitsA, grab_become, .id = "col")
  panelA$R2 <- map_dbl(fitsA, ~ summary(.x)$r.squared)
```

````
  ## Panel B
  fitsB <- Map(function(y, b6) run_panelB(df, y, b6), outcomes, bases6)
  panelB <- map_dfr(fitsB, grab_become, .id = "col")
  panelB$R2 <- map_dbl(fitsB, ~ summary(.x)$r.squared)

  wave6_int <- map_dfr(fitsB, function(fit) {
    s <- summary(fit)$coef
    int_row <- grep("^temp:", rownames(s)) # temp:base6
    base_row <- grep("6$", rownames(s))[1] # base6 term
    tibble(
      coef_wave6x = s[int_row, "Estimate"],
      se_wave6x = s[int_row, "Std. Error"],
      coef_wave6 = s[base_row, "Estimate"],
      se_wave6 = s[base_row, "Std. Error"]
    )
  }, .id = "col")

  panelB <- bind_cols(panelB, wave6_int[,-1])

  ## Panel C
  fitsC <- Map(function(y, b6) run_panelC(df, y, b6), ihs_out, bases6)
  panelC <- map_dfr(fitsC, grab_become, .id = "col")
  panelC$R2 <- map_dbl(fitsC, ~ summary(.x)$r.squared)

  list(
    panelA = panelA,
    panelB = panelB,
    panelC = panelC
  )
}

```

```{r}
````

````
## use it
res3_rep <- run_table3_rep(dat_rep)

res3_rep$panelA
res3_rep$panelB
res3_rep$panelC
```
```{r}
res3_aligned <- run_table3_rep(dat_raw_aligned)

res3_aligned$panelA
res3_aligned$panelB
res3_aligned$panelC
```
```{r}
res3_raw <- run_table3_rep(dat_raw)
res3_raw$panelA
res3_raw$panelB
res3_raw$panelC

```
```{r}
library(dplyr)

## Panel A: only 4 cols + sample/panel
panelA_combined <- bind_rows(
  res3_rep$panelA %>% mutate(sample = "replication", panel = "A"),
  res3_aligned$panelA %>% mutate(sample = "aligned", panel = "A"),
  res3_raw$panelA %>% mutate(sample = "raw", panel = "A")
) %>%
  relocate(sample, panel)

## Panel B: has extra wave6 columns (same structure across samples)
panelB_combined <- bind_rows(
  res3_rep$panelB %>% mutate(sample = "replication", panel = "B"),
````

````
  res3_aligned$panelB %>% mutate(sample = "aligned", panel = "B"),
  res3_raw$panelB %>% mutate(sample = "raw", panel = "B")
) %>%
  relocate(sample, panel)

## Panel C: same idea
panelC_combined <- bind_rows(
  res3_rep$panelC %>% mutate(sample = "replication", panel = "C"),
  res3_aligned$panelC %>% mutate(sample = "aligned", panel = "C"),
  res3_raw$panelC %>% mutate(sample = "raw", panel = "C")
) %>%
  relocate(sample, panel)

write.csv(panelA_combined, "table3A.csv", row.names = FALSE)
write.csv(panelB_combined, "table3B.csv", row.names = FALSE)
write.csv(panelC_combined, "table3C.csv", row.names = FALSE)
```
````
````