

Stat 156 Final Project

Keval Amin^a and Stephanie Quiroz^b

^aUniversity of California, Berkeley and Sciences Po Paris

^bUniversity 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:

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

where Y_i is the log-change in financial assets, temp_i is an indicator for being newly eligible for a 401(k), $f(\text{age}_i)$ 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:

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

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	Original	Aligned	Raw	Replication
Variable				
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	Original	Aligned	Raw	Replication
Variable				
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)*:

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

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:

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

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

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

The doubly robust estimator for the ATT is:

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

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 the 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)

```
1 ----  
2 title: "Matching"  
3 format: html  
4 ----  
5  
6  
7 '``'{r}  
8 library(dplyr)  
9 library(tidyr)  
10 library(MatchIt)  
11  
12 run_matching_att <- function(df,  
13                                 outcome_var = "taltb",  
14                                 treat_var = "temp",  
15                                 covars,  
16                                 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

```

17         seed = 156) {
18
19 # 1. Main analysis sample (same as IPW)
20
21 dat <- make_ipw_sample3(df)
22
23
24 # 2. Build outcome
25
26 dat$y401k_ipw <- make_ipw_outcome3(dat, outcome_var)
27
28
29 # 3. Check covariates exist
30
31 missing_covars <- setdiff(covars, names(dat))
32
33 if (length(missing_covars) > 0) {
34
35   stop(
36     "These covariates are not in the data: ",
37     paste(missing_covars, collapse = ", "))
38
39 }
40
41
42 # 4. Filter to non-missing treatment, outcome, and covariates
43
44 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

```

35 filter(!is.na(.data[[treat_var]]),
36     !is.na(y401k_ipw)) %>%
37 drop_na(all_of(covars))
38
39 if (nrow(dat_complete) == 0) {
40   stop(
41     "After filtering for non-missing treatment, outcome, and covariates, ",
42     "no observations remain. Check missingness again."
43   )
44 }
45
46 # 5. PS formula:
47 # ALWAYS include age squared
48 rhs_terms <- c("tage", "I(tage^2)", setdiff(covars, "tage"))
49 rhs <- paste(rhs_terms, collapse = " + ")
50
51 ps_formula <- as.formula(
52   paste0(treat_var, " ~ ", rhs)

```

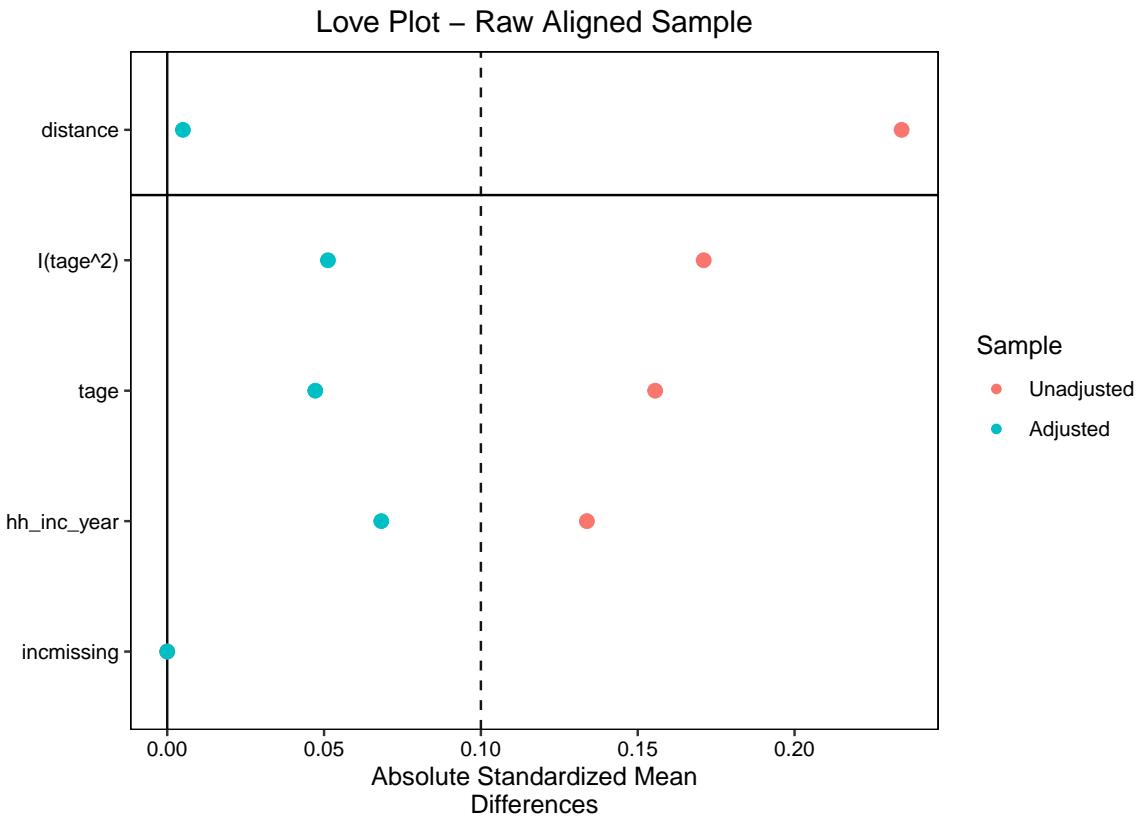


Figure 1. Love plot showing covariate balance before and after matching for the Raw Aligned sample.

```

53 )
54
55
56 # 6. Nearest neighbor matching on logit PS
57 m.out <- matchit(
58   formula = ps_formula,
59   data = dat_complete,
60   method = "nearest",
61   distance = "logit",
62   replace = FALSE,
63   ratio = 1
64 )
65
66 matched_dat <- match.data(m.out)
67

```

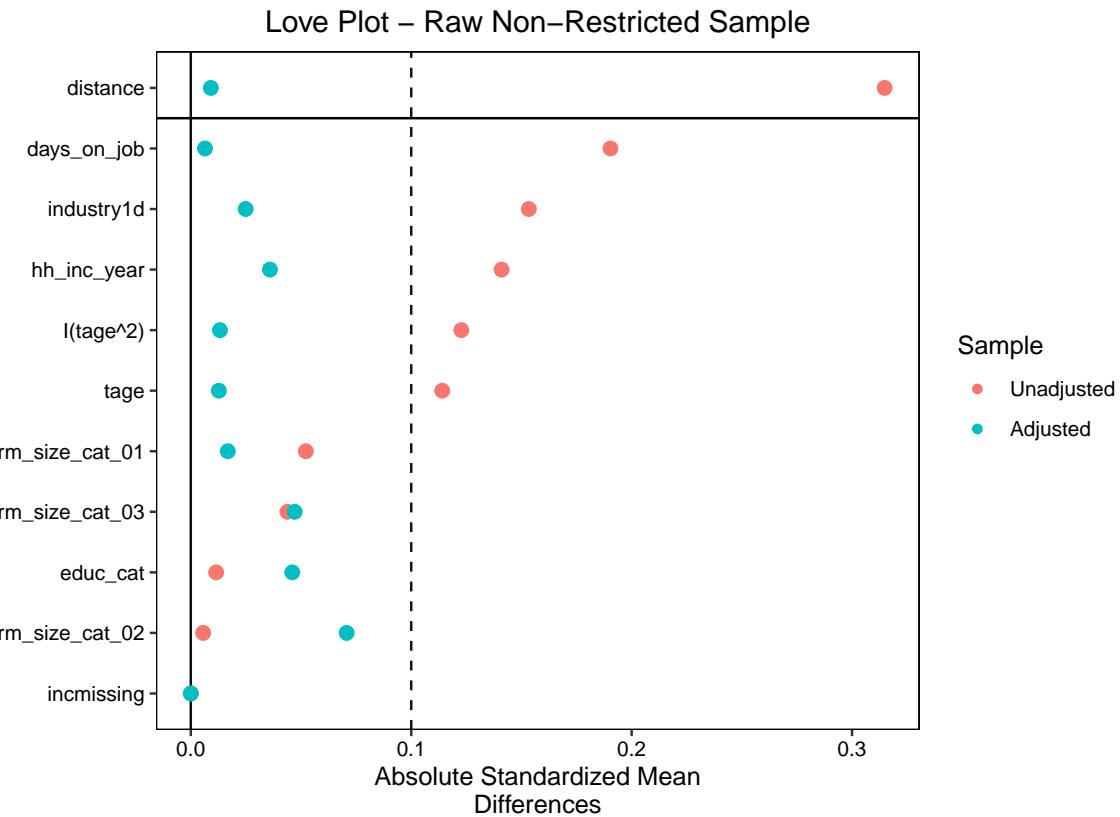


Figure 2. Love plot showing covariate balance before and after matching for the Raw Non-Restricted sample.

```

68 # 7. ATT point estimate
69 z <- matched_dat[[treat_var]]
70 y <- matched_dat$y401k_ipw
71
72 att_hat <- mean(y[z == 1]) - mean(y[z == 0])
73 n_treated <- sum(z == 1)
74 n_control <- sum(z == 0)
75 N <- nrow(matched_dat)
76
77 # 8. Bootstrap SE + CI
78 set.seed(seed)
79 boot_est <- replicate(B, {
80   idx <- sample(seq_len(N), replace = TRUE)
81   boot_dat <- matched_dat[idx, ]
82   z_b <- boot_dat[[treat_var]]

```

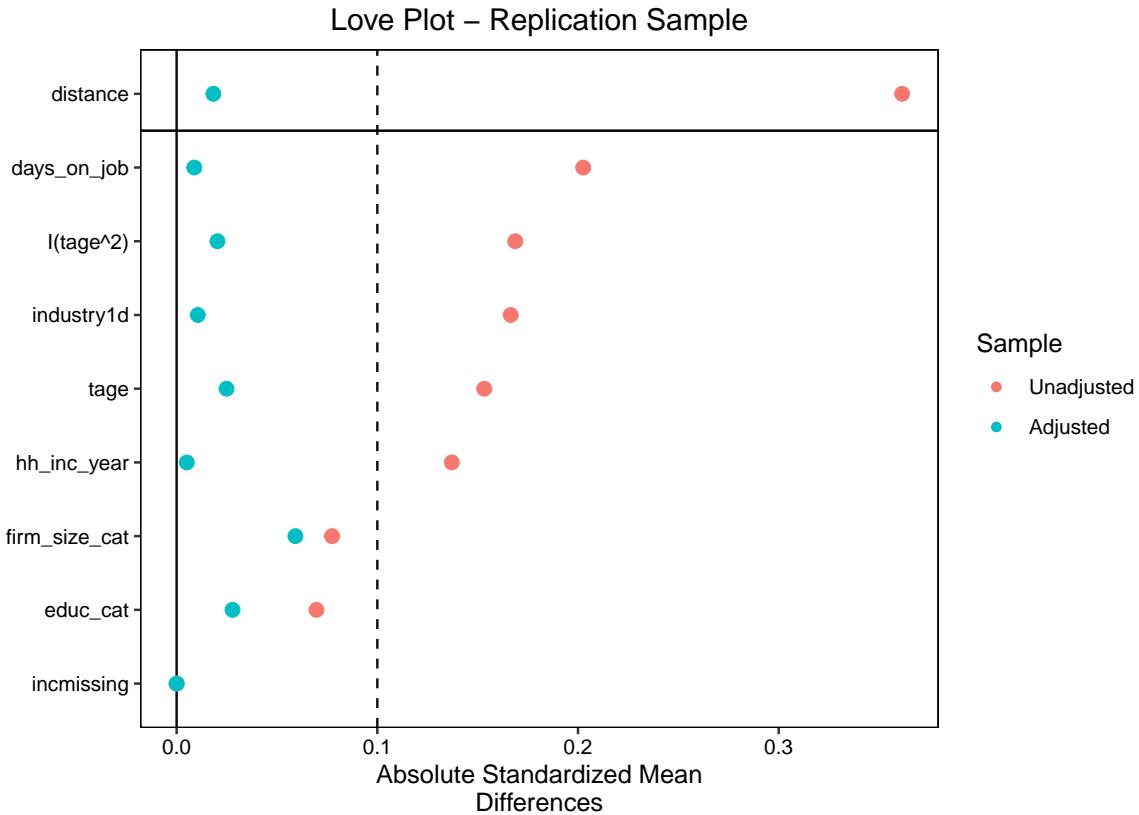


Figure 3. Love plot showing covariate balance before and after matching for the Replication sample.

```

83 y_b <- boot_dat$y401k_ipw
84 mean(y_b[z_b == 1]) - mean(y_b[z_b == 0])
85 }
86
87 se_hat <- sd(boot_ests)
88 ci_95 <- att_hat + c(-1, 1) * 1.96 * se_hat
89
90 list(
91   estimate = att_hat,
92   se = se_hat,
93   ci_95 = ci_95,
94   n_treated = n_treated,
95   n_control = n_control,
96   n_matched_rows = N,
97   treat_var = treat_var,
```

```

98     covariates = covars
99   )
100 }
101 /**
102 **'{r}
103 covars_raw_aligned <- c("tage", "hh_inc_year", "incmissing")
104
105 match_raw_aligned <- run_matching_att(
106   dat_raw_aligned_ipw,
107   outcome_var = "taltb",
108   treat_var = "temp",
109   covars = covars_raw_aligned
110 )
111
112
113 match_raw_aligned$estimate
114 match_raw_aligned$se
115 match_raw_aligned$ci_95
116 match_raw_aligned$n_treated
117 match_raw_aligned$n_control
118
119 /**
120 **'{r}
121 covars_raw_non <- c("tage", "hh_inc_year", "incmissing",
122   "educ_cat", "firm_size_cat", "industry1d", "days_on_job")
123
124 match_raw_non <- run_matching_att(
125   dat_raw_ipw_non,
126   outcome_var = "taltb",
127   treat_var = "temp",
128   covars = covars_raw_non
129 )
130
131 match_raw_non$estimate
132 match_raw_non$se

```

```

133 match_raw_non$ci_95
134 match_raw_non$n_treated
135 match_raw_non$n_control
136 ''
137 '''
138 covars_rep <- c("tage", "hh_inc_year", "incmissing",
139                 "educ_cat", "firm_size_cat", "industryid", "days_on_job")
140
141 match_rep <- run_matching_att(
142   dat_rep_ipw_rep,
143   outcome_var = "taltb",
144   treat_var = "temp",
145   covars = covars_rep
146 )
147
148 match_rep$estimate
149 match_rep$se
150 match_rep$ci_95
151 match_rep$n_treated
152 match_rep$n_control
153
154 '''
155 '''
156 matching_summary <- tibble::tibble(
157   sample = c("Raw aligned", "Raw non-restricted", "Replication"),
158   estimate = c(match_raw_aligned$estimate,
159               match_raw_non$estimate,
160               match_rep$estimate),
161   se = c(match_raw_aligned$se,
162         match_raw_non$se,
163         match_rep$se),
164   ci_lower = c(match_raw_aligned$ci_95[1],
165               match_raw_non$ci_95[1],
166               match_rep$ci_95[1]),
167   ci_upper = c(match_raw_aligned$ci_95[2],

```

```

168         match_raw_non$ci_95[2] ,
169
170         match_rep$ci_95[2]), ,
171
172 n_treated = c(match_raw_aligned$n_treated,
173
174         match_raw_non$n_treated,
175
176         match_rep$n_treated),
177
178 n_control = c(match_raw_aligned$n_control,
179
180         match_raw_non$n_control,
181
182         match_rep$n_control)
183
184 )
185
186
187
188 # Print the table
189
190 print(matching_summary)
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2190
2191
2192
2193
2194
2195
2196
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2296
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2388
2389
2390
2391
2392
2393
2394
2395
2396
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2488
2489
2490
2491
2492
2493
2494
2495
2496
2496
2497
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2558
2559
2560
2561
2
```

A.2 Doubly Robust Estimation (Doubly_Robust.qmd)

```
1 ----  
2 title: "Doubly Robust"  
3 format: html  
4 ----  
5  
6  
7  
8  
9  
10 ` `` {r}  
11 library(dplyr)  
12 library(tidyr)  
13  
14 run_dr_att <- function(df,  
15  
16  
17  
18  
19 outcome_var = "taltb",  
treat_var = "temp",  
covars,  
B = 300,  
seed = 156,
```

```

20         ps_trim = c(0.01, 0.99) {
21 
22     # 1. Main analysis sample (same as IPW pipeline)
23 
24     dat <- make_ipw_sample3(df)
25 
26 
27     # 2. Build outcome (same definition as for IPW)
28 
29     dat$y401k_ipw <- make_ipw_outcome3(dat, outcome_var)
30 
31 
32     # 3. Check covariates exist
33 
34     missing_covars <- setdiff(covars, names(dat))
35 
36     if (length(missing_covars) > 0) {
37 
38         stop(
39             "These covariates are not in the data: ",
40             paste(missing_covars, collapse = ", ")
41         )
42 
43     }
44 
45 
46     # 4. Filter to complete cases in treatment, outcome, and covariates
47 
48     dat_complete <- dat %>%
49 
50         filter(
51 
52             !is.na(.data[[treat_var]]),
53 
54             !is.na(y401k_ipw)
55 
56         ) %>%
57 
58         tidyrr::drop_na(all_of(covars))
59 
60 
61     if (nrow(dat_complete) == 0) {
62 
63         stop(
64 
65             "After filtering for non-missing treatment, outcome, and covariates, ",
66 
67             "no observations remain."
68         )
69 
70     }
71 
72 
73     # Rename for convenience
74 
75     Z <- dat_complete[[treat_var]] # 1 = treated (temp), 0 = control
76 
77     Y <- dat_complete$y401k_ipw
78 
79 
```

```

55 # 5. Propensity score model (same as before, tage + tage^2 + others)
56 rhs_terms <- c("tage", "I(tage^2)", setdiff(covars, "tage"))
57 rhs <- paste(rhs_terms, collapse = " + ")
58 ps_formula <- as.formula(paste0(treat_var, " ~ ", rhs))
59
60 ps_model <- glm(
61   ps_formula,
62   data = dat_complete,
63   family = binomial(link = "logit")
64 )
65
66 e_hat <- predict(ps_model, type = "response")
67
68
69
70 # 6. Outcome model for controls only: m0(x) = E[Y | T=0, X]
71 outcome_rhs <- paste(covars, collapse = " + ")
72 m0_formula <- as.formula(paste0("y401k_ipw ~ ", outcome_rhs))
73
74 m0_model <- glm(
75   m0_formula,
76   data = dat_complete[Z == 0, ],
77   family = gaussian()
78 )
79
80 m0_hat <- predict(m0_model, newdata = dat_complete, type = "response")
81
82 # 7. Doubly robust ATT estimator
83 # tau_ATT = (1/N1) * sum[ Z*(Y - m0) - (e/(1-e))*(1-Z)*(Y - m0) ]
84 N <- nrow(dat_complete)
85 N1 <- sum(Z == 1)
86 R0_hat <- Y - m0_hat
87
88 tau_hat <- (1 / N1) * sum(
89   Z * R0_hat -

```

```

90   (e_hat / (1 - e_hat)) * (1 - Z) * R0_hat
91 )
92
93 # 8. Bootstrap SE and CI (using fixed nuisance estimates)
94 set.seed(seed)
95
96 boot_est <- replicate(B, {
97   idx <- sample(seq_len(N), replace = TRUE)
98   Z_b <- Z[idx]
99   Y_b <- Y[idx]
100  e_b <- e_hat[idx]
101  m0_b <- m0_hat[idx]
102  R0_b <- Y_b - m0_b
103  N1_b <- sum(Z_b == 1)
104
105  # guard against (extremely unlikely) all-control resample
106  if (N1_b == 0) return(NA_real_)
107
108  sum(
109    Z_b * R0_b -
110    (e_b / (1 - e_b)) * (1 - Z_b) * R0_b
111  ) / N1_b
112 })
113
114 boot_est <- boot_est[!is.na(boot_est)]
115 se_hat <- sd(boot_est)
116 ci_95 <- tau_hat + c(-1, 1) * 1.96 * se_hat
117
118 list(
119   estimand = "ATT",
120   estimate = tau_hat,
121   se = se_hat,
122   ci_95 = ci_95,
123   ps_model = ps_model,
124   # keep these names for compatibility, even though m1 is not used for ATT

```

```

125     m1_model = NULL,
126     m0_model = m0_model,
127     e_hat = e_hat,
128     m1_hat = NULL,
129     m0_hat = m0_hat,
130     covariates = covars,
131     treat_var = treat_var,
132     outcome_var = outcome_var,
133     n_obs = N,
134     boot_ests = boot_ests,
135     data_used = dat_complete
136   )
137 }
138
139 /**
140 **{r}
141 # 1) Raw aligned
142 covars_raw_aligned <- c("tage", "hh_inc_year", "incmissing")
143
144 dr_raw_aligned <- run_dr_att(
145   dat_raw_aligned_ipw,
146   outcome_var = "taltb",
147   treat_var = "temp",
148   covars = covars_raw_aligned
149 )
150
151 dr_raw_aligned$estimate
152 dr_raw_aligned$se
153 dr_raw_aligned$ci_95
154
155
156 # 2) Raw non-restricted
157 covars_raw_non <- c(
158   "tage", "hh_inc_year", "incmissing",
159   "educ_cat", "firm_size_cat", "industry1d", "days_on_job"

```

```

160  )
161
162 dr_raw_non <- run_dr_att(
163   dat_raw_ipw_non,
164   outcome_var = "taltb",
165   treat_var = "temp",
166   covars = covars_raw_non
167 )
168
169 dr_raw_non$estimate
170 dr_raw_non$se
171 dr_raw_non$ci_95
172
173
174 # 3) Replication sample
175 covars_rep <- c(
176   "tage", "hh_inc_year", "incmissing",
177   "educ_cat", "firm_size_cat", "industry1d", "days_on_job"
178 )
179
180 dr_rep <- run_dr_att(
181   dat_rep_ipw_rep,
182   outcome_var = "taltb",
183   treat_var = "temp",
184   covars = covars_rep
185 )
186
187 dr_rep$estimate
188 dr_rep$se
189 dr_rep$ci_95
190
191 """
192 """{r}
193 dr_summary <- tibble::tibble(
194   sample = c("Raw aligned", "Raw non-restricted", "Replication"),

```

```

195 method = "DR AIPW (ATT)",
196 estimate = c(dr_raw_aligned$estimate,
197               dr_raw_non$estimate,
198               dr_rep$estimate),
199 se = c(dr_raw_aligned$se,
200        dr_raw_non$se,
201        dr_rep$se),
202 ci_lower = c(dr_raw_aligned$ci_95[1],
203               dr_raw_non$ci_95[1],
204               dr_rep$ci_95[1]),
205 ci_upper = c(dr_raw_aligned$ci_95[2],
206               dr_raw_non$ci_95[2],
207               dr_rep$ci_95[2]),
208 n_obs = c(dr_raw_aligned$n_obs,
209            dr_raw_non$n_obs,
210            dr_rep$n_obs)
211 )
212
213 dr_summary
214
215 """

```

A.3 Rosenbaum Sensitivity Analysis (Rosenbaum.qmd)

```

1 ---
2 title: "Rosenbaum"
3 format: html
4 ---
5
6
7 """{r}
8
9 ## Matching + Rosenbaum Sensitivity Script for 401(k) Data
10
11

```

```

12 ## 0. Libraries
13 library(dplyr)
14 library(tidyr)
15 library(MatchIt)
16 library(rbounds)
17
18 make_ipw_sample3 <- function (dat)
19 {
20   dat %>% filter(age_ok, for_profit, yr1jb1 == 1, y401k, notmissing)
21 }
22
23 make_ipw_outcome3 <- function (df, base)
24 {
25   a6 <- df[[paste0(base, "6")]]
26   a9 <- df[[paste0(base, "9")]]
27   a12 <- df[[paste0(base, "12")]]
28   log(a12 + 10) - 2 * log(a9 + 10) + log(a6 + 10)
29 }
30
31 ## NOTE:
32 ##RUN THESE FIRST
33 ## - dat_raw_aligned_ipw
34 ## - dat_raw_ipw_non
35 ## - dat_rep_ipw_rep
36 ## - make_ipw_sample3()
37 ## - make_ipw_outcome3()
38
39
40
41 ## 1. Matching function that returns ATT + SE + CI + matched data
42
43
44 run_matching_att <- function(df,
45                               outcome_var = "taltb",
46                               treat_var = "temp",

```

```

47             covars,
48             B = 300,
49             seed = 156) {
50
# 1. Main analysis sample (same as IPW)
51 dat <- make_ipw_sample3(df)
52
53 # 2. Build outcome (401k IPW outcome)
54 dat$y401k_ipw <- make_ipw_outcome3(dat, outcome_var)
55
56 # 3. Check covariates exist
57 missing_covars <- setdiff(covars, names(dat))
58 if (length(missing_covars) > 0) {
59   stop(
60     "These covariates are not in the data: ",
61     paste(missing_covars, collapse = ", "))
62 }
63
64
# 4. Filter to non-missing treatment, outcome, and covariates
65 dat_complete <- dat %>%
66   filter(
67     !is.na(.data[[treat_var]]),
68     !is.na(y401k_ipw)
69   ) %>%
70   tidyr::drop_na(all_of(covars))
71
72
73 if (nrow(dat_complete) == 0) {
74   stop(
75     "After filtering for non-missing treatment, outcome, and covariates, ",
76     "no observations remain. Check missingness again."
77   )
78 }
79
80 # 5. Propensity score formula (always include age + age^2)
81 rhs_terms <- c("tage", "I(tage^2)", setdiff(covars, "tage"))

```

```

82   rhs <- paste(rhs_terms, collapse = " + ")
83   ps_formula <- as.formula(paste0(treat_var, " ~ ", rhs))
84
85   # 6. Nearest neighbor matching on logit PS (ATT)
86   m.out <- matchit(
87     formula = ps_formula,
88     data = dat_complete,
89     method = "nearest",
90     distance = "logit",
91     replace = FALSE,
92     ratio = 1,
93     estimand = "ATT"
94   )
95
96   matched_dat <- match.data(m.out)
97
98   # 7. ATT point estimate on matched sample
99   z <- matched_dat[[treat_var]]
100  y <- matched_dat$y401k_ipw
101
102  att_hat <- mean(y[z == 1]) - mean(y[z == 0])
103  n_treated <- sum(z == 1)
104  n_control <- sum(z == 0)
105  N <- nrow(matched_dat)
106
107  # 8. Bootstrap SE + CI
108  set.seed(seed)
109  boot_est <- replicate(B, {
110    idx <- sample(seq_len(N), replace = TRUE)
111    boot_dat <- matched_dat[idx, ]
112    z_b <- boot_dat[[treat_var]]
113    y_b <- boot_dat$y401k_ipw
114    mean(y_b[z_b == 1]) - mean(y_b[z_b == 0])
115  })
116
```

```

117   se_hat <- sd(boot_est)
118   ci_95 <- att_hat + c(-1, 1) * 1.96 * se_hat
119
120 # 9. Return everything useful
121 list(
122   estimate = att_hat,
123   se = se_hat,
124   ci_95 = ci_95,
125   n_treated = n_treated,
126   n_control = n_control,
127   n_matched_rows = N,
128   treat_var = treat_var,
129   covariates = covars,
130   m.out = m.out,
131   matched_data = matched_dat,
132   boot_est = boot_est
133 )
134 }
135
136
137
138 ## 2. Rosenbaum sensitivity function (HodgesLehmann)
139
140
141 rosenbaum_hl <- function(match_result, treat_var = "temp") {
142   dat <- match_result$matched_data
143
144   if (!"subclass" %in% names(dat)) {
145     stop("Matched data must contain 'subclass' to identify matched pairs/sets.")
146   }
147
148   # For 1:1 matching, each subclass has 1 treated + 1 control
149   dat_pairs <- dat %>%
150     arrange(subclass, dplyr::desc(.data[[treat_var]])) %>%
151     group_by(subclass) %>%

```

```

152   summarise(
153     y_treat = y401k_ipw[.data[[treat_var]] == 1],
154     y_ctrl = y401k_ipw[.data[[treat_var]] == 0],
155     .groups = "drop"
156   )
157
158 # Rosenbaum bounds for HodgesLehmann estimate
159 # x = treated outcomes, y = control outcomes
160 hlsens(
161   x = dat_pairs$y_treat,
162   y = dat_pairs$y_ctrl,
163   Gamma = 2, # max Gamma to check (change if needed)
164   GammaInc = 0.1 # step size in Gamma grid
165 )
166 }
167
168
169
170 ## 3. Run matching on the three datasets
171
172
173 ## 3.1 Raw aligned
174 covars_raw_aligned <- c("tage", "hh_inc_year", "incmissing")
175
176 match_raw_aligned <- run_matching_att(
177   dat_raw_aligned_ipw,
178   outcome_var = "taltb",
179   treat_var = "temp",
180   covars = covars_raw_aligned
181 )
182
183 ## 3.2 Raw non-restricted
184 covars_raw_non <- c(
185   "tage", "hh_inc_year", "incmissing",
186   "educ_cat", "firm_size_cat", "industry1d", "days_on_job"

```

```

187  )
188
189 match_raw_non <- run_matching_att(
190   dat_raw_ipw_non,
191   outcome_var = "taltb",
192   treat_var = "temp",
193   covars = covars_raw_non
194 )
195
196 ## 3.3 Replication sample
197 covars_rep <- c(
198   "tage", "hh_inc_year", "incmissing",
199   "educ_cat", "firm_size_cat", "industryid", "days_on_job"
200 )
201
202 match_rep <- run_matching_att(
203   dat_rep_ipw_rep,
204   outcome_var = "taltb",
205   treat_var = "temp",
206   covars = covars_rep
207 )
208
209
210
211 ## 4. Summaries of matching estimates ----
212
213
214 matching_summary <- tibble::tibble(
215   sample = c("Raw aligned", "Raw non-restricted", "Replication"),
216   estimate = c(match_raw_aligned$estimate,
217     match_raw_non$estimate,
218     match_rep$estimate),
219   se = c(match_raw_aligned$se,
220     match_raw_non$se,
221     match_rep$se),

```

```

222 ci_lower = c(match_raw_aligned$ci_95[1],
223   match_raw_non$ci_95[1],
224   match_rep$ci_95[1]),
225 ci_upper = c(match_raw_aligned$ci_95[2],
226   match_raw_non$ci_95[2],
227   match_rep$ci_95[2]),
228 n_treat = c(match_raw_aligned$n_treated,
229   match_raw_non$n_treated,
230   match_rep$n_treated),
231 n_ctrl = c(match_raw_aligned$n_control,
232   match_raw_non$n_control,
233   match_rep$n_control),
234 n_rows = c(match_raw_aligned$n_matched_rows,
235   match_raw_non$n_matched_rows,
236   match_rep$n_matched_rows)
237 )
238
239 matching_summary
240
241
242
243 ## 5. Rosenbaum sensitivity for each matched sample
244
245
246 rosen_raw_aligned <- rosenbaum_hl(match_raw_aligned)
247 rosen_raw_non <- rosenbaum_hl(match_raw_non)
248 rosen_rep <- rosenbaum_hl(match_rep)
249
250 # Print results to inspect in console
251 rosen_raw_aligned
252 rosen_raw_non
253 rosen_rep
254
255
256

```

```

257   """
258   """{r}
259
260 #install.packages("cobalt")
261
262
263 library(cobalt)
264
265 love.plot(
266   match_rep$m.out,
267   stats = "mean.diffs",
268   abs = TRUE,
269   binary = "std", # <- force standardized for binary vars
270   var.order = "unadjusted",
271   thresholds = c(m = 0.1),
272   title = "Love Plot - Replication Sample"
273 )
274
275
276 """
277 """{r}
278 love.plot(
279   match_raw_aligned$m.out,
280   stats = "mean.diffs",
281   abs = TRUE,
282   binary = "std",
283   var.order = "unadjusted",
284   thresholds = c(m = 0.1),
285   title = "Love Plot - Raw Aligned Sample"
286 )
287
288 love.plot(
289   match_raw_non$m.out,
290   stats = "mean.diffs",
291   abs = TRUE,

```

```

292     binary = "std",
293     var.order = "unadjusted",
294     thresholds = c(m = 0.1),
295     title = "Love Plot - Raw Non-Restricted Sample"
296   )
297   """
298
299
300
301   '''{r}
302   # Function to extract Rosenbaum bounds into a clean format
303   extract_rosenbaum_table <- function(rosen_obj, sample_name) {
304     # 'hlsens' stores the data frame in $bounds
305     # Columns are typically: Gamma, Lower Bound, Upper Bound
306     df <- as.data.frame(rosen_obj$bounds)
307
308     df %>%
309       mutate(sample = sample_name) %>%
310       select(sample, everything()) %>%
311       rename(
312         Gamma = 1, # Usually the first column
313         Lower_HL = 2,
314         Upper_HL = 3
315       )
316   }
317
318   # Combine all results into one table
319   rosen_summary_table <- bind_rows(
320     extract_rosenbaum_table(rosen_raw_aligned, "Raw aligned"),
321     extract_rosenbaum_table(rosen_raw_non, "Raw non-restricted"),
322     extract_rosenbaum_table(rosen_rep, "Replication")
323   )
324
325
326

```

```
327 print(rosen_summary_table)
328 ````
```

A.4 Results Compilation (Results.qmd)

```
1 ---
2 title: "Results"
3 format: html
4 ---
5
6
7 ``{r}
8 dr_summary
9 matching_summary
10 rosen_summary_table
11 ````
```

APPENDIX B. REPLICATION

B.1 Load Data Raw

```
1 ---
2 title: "Untitled"
3 format: html
4 ---
5
6
7 ``{r}
8 library(readr)
9 library(stringr)
10
11 # helper to convert .sas layout into fwf specification
12 sas_to_fwf <- function(sasfile) {
13   sas <- readLines(sasfile)
14   pattern <- "^\s*([A-Za-z0-9_]+)\s+\$?\s*(\d+)\s*-\s*(\d+)"
15   m <- str_match(sas, pattern)
```

```

16 m <- m[!is.na(m[,1]), , drop = FALSE]
17 varnames <- m[,2]
18 start <- as.integer(m[,3])
19 end <- as.integer(m[,4])
20 widths <- end - start + 1
21 fwf_widths(widths, col_names = varnames)
22 }
23
24 # directories
25 raw_dir <- "data/raw"
26 out_dir <- file.path(raw_dir, "rds_1") # <- changed folder name
27 dir.create(out_dir, showWarnings = FALSE)
28
29 # list of file stems
30 stems <- c("t3", "t6", "t7", "t9", "t12",
31           "w7", "w8", "w9")
32
33 # loop to convert all files
34 for (stem in stems) {
35
36   datfile <- file.path(raw_dir, paste0(stem, ".dat"))
37   sasfile <- file.path(raw_dir, paste0(stem, ".sas"))
38   rdsfile <- file.path(out_dir, paste0(stem, ".rds"))
39
40   if (!file.exists(datfile)) next
41   if (!file.exists(sasfile)) next
42
43   fwf <- sas_to_fwf(sasfile)
44   df <- read_fwf(datfile, fwf, progress = FALSE)
45
46   saveRDS(df, rdsfile)
47 }
48
49 ***

```

B.2 Construct aligned data

```
1 ---  
2 title: "Untitled"  
3 format: html  
4 ---  
5  
6  
7 '```{r}  
8 library(dplyr)  
9 library(tidyr)  
10  
11  
12 ## Helper: safe numeric conversion  
13  
14 to_num <- function(x) suppressWarnings(as.numeric(x))  
15  
16  
17 ## Helper: clean job start date (TSJDATE1)  
18 ## - convert to numeric  
19 ## - set non-positive / sentinel values (<=0, -1) to NA  
20  
21 clean_tsjdate <- function(x) {  
22   x <- suppressWarnings(as.numeric(x))  
23   x[x <= 0] <- NA  
24   x  
25 }  
26  
27  
28 ## Helper: build assets for one wave  
29  
30 make_assets_wave <- function(df, wave_num) {  
31   df %>%  
32     transmute(  
33       SSUID, SHHADID, EENTAID, EPPNUM,  
34       wave = wave_num,
```

```

35   # 401(k) balance
36   taltb = to_num(TALTB),
37   # IRA balance
38   thhira = to_num(THHIRA),
39   # Other financial assets (interest, stocks, other assets)
40   otherassets = to_num(THHINTBK) +
41     to_num(THHINTOT) +
42     to_num(RHHSTK) +
43     to_num(THHOTAST),
44   # secured & unsecured debt
45   thhscdbt = to_num(THHSCDBT),
46   rhhuscbt = to_num(RHHUSCBT),
47   # car values: sum over up to 3 cars
48   tc当地 = to_num(TCARVAL1) +
49     to_num(TCARVAL2) +
50     to_num(TCARVAL3)
51 )
52 }
53
54
55 ## MAIN: build_dat()
56 ## takes t3,t6,t9,t12,t7,w7,w8,w9 and returns merged dataset
57 ## with: age_ok, for_profit, yr1jb1, temp, y401k, notmissing,
58 ## hh_inc_year, etc.
59
60 build_dat <- function(t3, t6, t9, t12, t7, w7, w8, w9) {
61   id_vars <- c("SSUID", "SHHADID", "EENTAID", "EPPNUM")
62
63   ## 1) Assets across waves wide
64   assets_long <- bind_rows(
65     make_assets_wave(t3, 3),
66     make_assets_wave(t6, 6),
67     make_assets_wave(t9, 9),
68     make_assets_wave(t12, 12)
69   )

```

```

70
71 assets_wide <- assets_long %>%
72   pivot_wider(
73     id_cols = all_of(id_vars),
74     names_from = wave,
75     values_from = c(talrb, thhira, otherassets, thhscdbt, rhhuscbt, tcarval),
76     names_sep = ""
77   )
78
79 ## 2) Wave-7 core (panel = SREFMON==4)
80 core7 <- w7 %>%
81   filter(SREFMON == 4) %>%
82   transmute(
83     SUID, SHHADID, EENTAID, EPPNUM,
84     tage = to_num(TAGE),
85     wpfinwgt = to_num(WPFINWGT),
86     eclwrk1 = to_num(ECLWRK1),
87     efnp = to_num(EFNP),
88     esex = to_num(ESEX),
89     tempall1 = to_num(TEMPALL1),
90     ejbind1 = to_num(EJBIND1),
91     tsjdate1 = clean_tsjdate(TSJDATE1),
92     srotaton = to_num(SROTATON)
93   )
94
95 ## 3) Year-1 income: sum THTOTINC from waves 79
96 make_inc <- function(w, tag) {
97   w %>%
98     select(SUID, SHHADID, EENTAID, EPPNUM, SREFMON, THTOTINC) %>%
99     mutate(
100       THTOTINC = to_num(THTOTINC),
101       month = paste0(tag, "_", SREFMON)
102     ) %>%
103     select(-SREFMON) %>%
104     pivot_wider(

```

```

105     id_cols = all_of(id_vars),
106     names_from = month,
107     values_from = THTOTINC
108   )
109 }
110
111 w7_inc <- make_inc(w7, "w7")
112 w8_inc <- make_inc(w8, "w8")
113 w9_inc <- make_inc(w9, "w9")
114
115 year1_income <- w7_inc %>%
116   full_join(w8_inc, by = id_vars) %>%
117   full_join(w9_inc, by = id_vars) %>%
118   mutate(
119     hh_inc_year = rowSums(
120       dplyr::select(., starts_with("w7_"), starts_with("w8_"), starts_with("w9_")),
121       na.rm = FALSE
122     )
123   ) %>%
124   select(all_of(id_vars), hh_inc_year)
125
126 ## 4) Topical module 7: plan eligibility and reasons
127 tm7 <- t7 %>%
128   transmute(
129     SSUID, SHHADID, EENTAID, EPPPNUM,
130     e1taxdef = to_num(E1TAXDEF),
131     e2taxdef = to_num(E2TAXDEF),
132     e3taxdef = to_num(E3TAXDEF),
133     enoina03 = to_num(ENOINA03),
134     enoinb03 = to_num(ENOINB03),
135     etdeffen = to_num(ETDEFFEN)
136   )
137
138 ## 5) Merge all pieces
139 dat <- assets_wide %>%

```

```

140 inner_join(core7, by = id_vars) %>%
141 inner_join(tm7, by = id_vars) %>%
142 left_join(year1_income, by = id_vars)
143
144 ## 6) Derive sample flags and d21ltalb
145 # (Make d21ltalb robust in case some waves are missing)
146 has_t12 <- "taltb12" %in% names(dat)
147 has_t3 <- "taltb3" %in% names(dat)
148
149 dat <- dat %>%
150   mutate(
151     age_ok = !is.na(tage) & tage >= 22 & tage <= 64,
152     for_profit = (eclwrk1 == 1),
153
154     yr1jb1 = dplyr::case_when(
155       srotaton == 1 ~ tsjdate1 > 19970299,
156       srotaton == 2 ~ tsjdate1 > 19970399,
157       srotaton == 3 ~ tsjdate1 > 19970499,
158       srotaton == 4 ~ tsjdate1 > 19970599,
159       TRUE ~ NA
160     ),
161
162     temp = as.integer(
163       (enoina03 == 1 & etdeffen == 1) |
164       (enoinb03 == 1)
165     ),
166
167     y401k = (temp == 1) |
168       (e1taxdef == 1) |
169       (e2taxdef == 1) |
170       (e3taxdef == 1) |
171       (etdeffen == 1 & yr1jb1 == 1),
172
173     # main outcome (use 6912 if available, else 369)
174     d21ltalb = dplyr::case_when(

```

```

175   has_t12 ~ log(talrb12 + 10) - 2 * log(talrb9 + 10) + log(talrb6 + 10),
176   has_t3 ~ log(talrb9 + 10) - 2 * log(talrb6 + 10) + log(talrb3 + 10),
177   TRUE ~ NA_real_
178 ),
179
180 lnA6 = log(talrb6 + 10),
181
182 # income: allowed to be missing, we flag it
183 incmissing = as.integer(is.na(hh_inc_year)),
184
185 # require outcome + key regressors present
186 notmissing = !is.na(d21ltalrb) &
187   !is.na(tage) &
188   !is.na(eclwrk1) &
189   !is.na(tsjdate1) &
190   !is.na(srotaton) &
191   !is.na(wpfinwgt)
192 )
193
194 dat
195 }
196
197
198 ## Helpers to build Table 1 from a built dat
199 w_mean_sd <- function(x, w) {
200   x <- as.numeric(x)
201   w <- as.numeric(w)
202   ok <- !is.na(x) & !is.na(w)
203   x <- x[ok]; w <- w[ok]
204   if (!length(x)) return(c(mean = NA_real_, sd = NA_real_))
205   w <- w / sum(w)
206   mu <- sum(w * x)
207   var <- sum(w * (x - mu)^2)
208   c(mean = mu, sd = sqrt(var))
209 }
```

```

210
211 make_block <- function(df) {
212   out <- list(
213     Age = w_mean_sd(df$tage, df$wpfinwgt),
214     'Yearly household income' =
215       w_mean_sd(df$hh_inc_year, df$wpfinwgt),
216     '401(k) assets' =
217       w_mean_sd(df$talrb6, df$wpfinwgt),
218     'IRA and Keogh assets' =
219       w_mean_sd(df$thhira6, df$wpfinwgt),
220     'Other financial assets'=
221       w_mean_sd(df$otherassets6, df$wpfinwgt),
222     'Secured debt' =
223       w_mean_sd(df$thhscdbt6, df$wpfinwgt),
224     'Unsecured debt' =
225       w_mean_sd(df$rhhuscbt6, df$wpfinwgt),
226     'Car value' =
227       w_mean_sd(df$tcarval6, df$wpfinwgt)
228   )
229   vals <- lapply(out, function(msd)
230     sprintf("%.1f\n%.1f", msd["mean"], msd["sd"]))
231   tibble::as_tibble_row(vals)
232 }
233
234 make_table1 <- function(dat) {
235   # Apply the same sample restrictions as in the Stata do-file
236   table1_sample <- dat %>%
237     filter(
238       age_ok,
239       for_profit,
240       yr1jb1 == 1,
241       y401k,
242       notmissing
243     )
244

```

```

245 cat("Table 1 sample size: ", nrow(table1_sample), "\n")
246 cat("Treatment (temp==1): ", sum(table1_sample$temp == 1, na.rm = TRUE), "\n")
247 cat("Control (temp==0): ", sum(table1_sample$temp == 0, na.rm = TRUE), "\n")
248
249 income_sample <- table1_sample %>%
250   filter(!is.na(hh_inc_year))
251 cat("Income row N (non-missing hh_inc_year): ", nrow(income_sample), "\n")
252
253 tab_all <- make_block(table1_sample) %>%
254   mutate(group = "All",
255         Observations = nrow(table1_sample))
256
257 tab_treat <- make_block(filter(table1_sample, temp == 1)) %>%
258   mutate(group = "Treatment group",
259         Observations = sum(table1_sample$temp == 1, na.rm = TRUE))
260
261 tab_ctrl <- make_block(filter(table1_sample, temp == 0)) %>%
262   mutate(group = "Control group",
263         Observations = sum(table1_sample$temp == 0, na.rm = TRUE))
264
265 bind_rows(tab_all, tab_treat, tab_ctrl) %>%
266   relocate(group, Observations)
267 }
268
269
270 """
271 """
272
273 dat_raw <- build_dat(
274   t3_raw, t6_raw, t9_raw, t12_raw,
275   t7_raw, w7_raw, w8_raw, w9_raw
276 )
277
278 table1_raw <- make_table1(dat_raw)
279 table1_raw

```

```

280 dat_rep <- build_dat(
281   t3_rep, t6_rep, t9_rep, t12_rep,
282   t7_rep, w7_rep, w8_rep, w9_rep
283 )
284
285 table1_rep <- make_table1(dat_rep)
286 table1_rep
287 /**
288 ***{r}
289 id_vars <- c("SSUID","SHHADID","EENTAID","EPPNUM")
290
291 normalize_ids <- function(df) {
292   df %>%
293     mutate(
294       SSUID = sub("^0+", "", as.character(SSUID)),
295       SHHADID = sub("^0+", "", as.character(SHHADID)),
296       EENTAID = sub("^0+", "", as.character(EENTAID)),
297       EPPNUM = sub("^0+", "", as.character(EPPNUM))
298     )
299 }
300
301 dat_raw_id <- normalize_ids(dat_raw)
302 dat_rep_id <- normalize_ids(dat_rep)
303
304 /**
305 ***{r}
306 flags_rep <- dat_rep_id %>%
307   select(all_of(id_vars),
308         yr1jb1_rep = yr1jb1,
309         temp_rep = temp,
310         y401k_rep = y401k)
311
312 /**
313 ***{r}
314 dat_raw_aligned <- dat_raw_id %>%

```

```

315 left_join(flags_rep, by = id_vars) %>%
316   mutate(
317     yr1jb1 = yr1jb1_rep,
318     temp = temp_rep,
319     y401k = y401k_rep
320   )
321
322 /**
323 `{{r}}
324 table1_rep <- make_table1(dat_rep_id)
325 table1_raw2 <- make_table1(dat_raw_aligned)
326
327 table1_rep
328 table1_raw2
329
330 /**

```

B.3 Table 1 Construction (table1.qmd)

```

1 ---
2 title: "table 1"
3 format: pdf
4 ---
5
6
7 `{{r}}
8 library(dplyr)
9 library(tidyr)
10
11 to_num <- function(x) suppressWarnings(as.numeric(x))
12
13
14 ## 1. Load raw files
15
16

```

```

17 t3 <- readRDS("data/raw/rds_down/t3.rds")
18 t6 <- readRDS("data/raw/rds_down/t6.rds")
19 t9 <- readRDS("data/raw/rds_down/t9.rds")
20 t12 <- readRDS("data/raw/rds_down/t12.rds")

21
22 t7 <- readRDS("data/raw/rds_down/t7.rds")
23 w7 <- readRDS("data/raw/rds_down/w7.rds")
24 w8 <- readRDS("data/raw/rds_down/w8.rds")
25 w9 <- readRDS("data/raw/rds_down/w9.rds")

26
27 id_vars <- c("SSUID", "SHHADID", "EENTAID", "EPPNUM")

28
29
30 ## 2. Assets: build taltb6, thhira6, otherassets6, etc.
31 ## Names are chosen to line up with the Stata do-file.

32
33
34 make_assets_wave <- function(df, wave_num) {
35   df %>%
36     transmute(
37       SSUID, SHHADID, EENTAID, EPPNUM,
38       wave = wave_num,
39       # 401(k) balance
40       taltb = to_num(TALTB),
41       # IRA balance
42       thhira = to_num(THHIRA),
43       # Other financial assets (interest, stocks, "other" assets)
44       otherassets = to_num(THHINTBK) +
45         to_num(THHINTOT) +
46         to_num(RHHSTK) +
47         to_num(THHOTAST),
48       # secured & unsecured debt
49       thhscdbt = to_num(THHSCDBT),
50       rhhuscbt = to_num(RHHUSCBT),
51       # car values: sum over up to 3 cars

```

```

52     tcarval = to_num(TCARVAL1) +
53         to_num(TCARVAL2) +
54         to_num(TCARVAL3)
55     )
56 }
57
58 assets_long <- bind_rows(
59   make_assets_wave(t3, 3),
60   make_assets_wave(t6, 6),
61   make_assets_wave(t9, 9),
62   make_assets_wave(t12, 12)
63 )
64
65 assets_wide <- assets_long %>%
66   pivot_wider(
67     id_cols = all_of(id_vars),
68     names_from = wave,
69     values_from = c(talrb, thhira, otherassets, thhscdbt, rhhuscbt, tcarval),
70     names_sep = ""
71   )
72 # This gives: talrb3, talrb6, talrb9, talrb12, etc.
73 # Wave 6 variables correspond to the Table 1 row (talrb6 401(k) assets).
74
75
76 ## 3. Wave 7 core stuff (one record per person: SREFMON==4)
77
78
79 core7 <- w7 %>%
80   filter(SREFMON == 4) %>% # this is how w7v2 is built in Stata
81   transmute(
82     SUID, SHHADID, EENTAID, EPPNUM,
83     tage = to_num(TAGE),
84     wpfinwgt = to_num(WPFINWGT),
85     eclwrk1 = to_num(ECLWRK1),
86     efnp = to_num(EFNP),

```

```

87     esex = to_num(ESEX),
88     tempall1 = to_num(TEMPALL1),
89     ejbind1 = to_num(EJBIND1),
90     tsjdate1 = to_num(TSJDATE1),
91     srotaton = to_num(SROTATON)
92 )
93
94
95 ## 3a. Calculate Year 1 household income (sum of all 12 months)
96 ## Wave 7 (months 1-4) + Wave 8 (months 1-4) + Wave 9 (months 1-4)
97
98
99 # Wave 7 income by reference month
100 w7_inc <- w7 %>%
101   select(SSUID, SHHADID, EENTAID, EPPNUM, SREFMON, THTOTINC) %>%
102   mutate(
103     THTOTINC = to_num(THTOTINC),
104     month = paste0("w7_", SREFMON)
105   ) %>%
106   select(-SREFMON) %>%
107   pivot_wider(names_from = month, values_from = THTOTINC, id_cols = all_of(id_vars))
108
109 # Wave 8 income by reference month
110 w8_inc <- w8 %>%
111   select(SSUID, SHHADID, EENTAID, EPPNUM, SREFMON, THTOTINC) %>%
112   mutate(
113     THTOTINC = to_num(THTOTINC),
114     month = paste0("w8_", SREFMON)
115   ) %>%
116   select(-SREFMON) %>%
117   pivot_wider(names_from = month, values_from = THTOTINC, id_cols = all_of(id_vars))
118
119 # Wave 9 income by reference month
120 w9_inc <- w9 %>%
121   select(SSUID, SHHADID, EENTAID, EPPNUM, SREFMON, THTOTINC) %>%

```

```

122   mutate(
123     THTOTINC = to_num(THTOTINC),
124     month = paste0("w9_", SREFMON)
125   ) %>%
126   select(-SREFMON) %>%
127   pivot_wider(names_from = month, values_from = THTOTINC, id_cols = all_of(id_vars))
128
129 # Merge all income months and sum to get Year 1 income
130 year1_income <- w7_inc %>%
131   full_join(w8_inc, by = id_vars) %>%
132   full_join(w9_inc, by = id_vars) %>%
133   mutate(
134     # Sum all 12 months (w7_1 through w7_4, w8_1 through w8_4, w9_1 through w9_4)
135     hh_inc_year = rowSums(select(., starts_with("w7_"), starts_with("w8_"), starts_with("w9_")),
136                           na.rm = FALSE) # Keep as NA if any month is missing
137   ) %>%
138   select(all_of(id_vars), hh_inc_year)
139
140
141 ## 4. Topical module 7: 401(k) eligibility & reasons
142
143
144 tm7 <- t7 %>%
145   transmute(
146     SSUID, SHHADID, EENTAID, EPPNUM,
147     e1taxdef = to_num(E1TAXDEF),
148     e2taxdef = to_num(E2TAXDEF),
149     e3taxdef = to_num(E3TAXDEF),
150     enoina03 = to_num(ENOINA03), # reason not covered: havent worked long enough (plan A)
151     enoinb03 = to_num(ENOINB03), # reason not covered: havent worked long enough (plan B)
152     etdeffen = to_num(ETDEFFEN) # firm offers any tax-deferred plan
153   )
154
155 ## 5. Merge assets + core + topical + Year 1 income into one person-level dataset
156

```

```

157
158 dat <- assets_wide %>%
159   inner_join(core7, by = id_vars) %>%
160   inner_join(tm7, by = id_vars) %>%
161   left_join(year1_income, by = id_vars)
162
163 cat("Rows in merged dat: ", nrow(dat), "\n")
164
165
166 ## 6. Reproduce Stata sample flags: yr1jb1, temp, y401k, notmissing
167
168
169 dat <- dat %>%
170   mutate(
171     # Age range: 2264 (matches: keep if tage>21 & tage<65)
172     age_ok = !is.na(tage) & tage >= 22 & tage <= 64,
173
174     # For-profit: eclwrk1 == 1
175     for_profit = (eclwrk1 == 1),
176
177     # yr1jb1: began current job within the past year at Wave 7 (Stata logic)
178     yr1jb1 = case_when(
179       srotaton == 1 ~ tsjdate1 > 19970299,
180       srotaton == 2 ~ tsjdate1 > 19970399,
181       srotaton == 3 ~ tsjdate1 > 19970499,
182       srotaton == 4 ~ tsjdate1 > 19970599,
183       TRUE ~ NA
184     ),
185
186     # Treatment: temp (exactly the Stata definition)
187     temp = as.integer(
188       (enoina03 == 1 & etdeffen == 1) | # "haven't worked long enough" & firm offers plan
189       (enoinb03 == 1)
190     ),
191

```

```

192 # y401k: firm offers 401(k) (Stata: y401k = temp | eitaxdef==1 | e2taxdef==1 | e3taxdef==1 | (
193   etdeffen==1 & yr1jb1))
194 y401k = (temp == 1) |
195   (e1taxdef == 1) |
196   (e2taxdef == 1) |
197   (e3taxdef == 1) |
198   (etdeffen == 1 & yr1jb1 == 1),
199
200 # Calculate d21ltalbt exactly as in Stata: ln(talbt12+10) - 2*(ln(talbt9+10)) + ln(talbt6+10)
201 d21ltalbt = log(talbt12 + 10) - 2 * log(talbt9 + 10) + log(talbt6 + 10),
202
203 # "notmissing" is defined via d21ltalbt ~= . in Stata
204 # Check if d21ltalbt is not missing (which happens when any component calculation fails)
205 notmissing = !is.na(d21ltalbt),
206
207 # Income missing dummy (Stata's incmissing); here we allow missing, we just flag it.
208 incmissing = as.integer(is.na(hh_inc_year))
209 )
210
211 cat("Counts: y401k TRUE/FALSE:\n")
212 print(table(dat$y401k, useNA = "ifany"))
213 cat("Counts: temp (treatment) 0/1:\n")
214 print(table(dat$temp, useNA = "ifany"))
215
216 ## 7. Table 1 *sample* = Stata's "main" restrictions
217 ## - age 2264
218 ## - for-profit (eclwrk1==1)
219 ## - yr1jb1 == 1 (started job within the previous year)
220 ## - y401k == TRUE
221 ## - notmissing == TRUE (has the data to construct d21ltalbt)
222
223
224 table1_sample <- dat %>%
225   filter(

```

```

226   age_ok,
227   for_profit,
228   yr1jb1 == 1,
229   y401k,
230   notmissing
231 )
232
233 cat("Table 1 sample size: ", nrow(table1_sample), "\n")
234 cat("Treatment (temp==1): ", sum(table1_sample$temp == 1, na.rm = TRUE), "\n")
235 cat("Control (temp==0): ", sum(table1_sample$temp == 0, na.rm = TRUE), "\n")
236
237 ## Separate N for *income* row (Stata has 818 instead of 835)
238 income_sample <- table1_sample %>%
239   filter(!is.na(hh_inc_year))
240
241 cat("Income row N (non-missing hh_inc_year): ", nrow(income_sample), "\n")
242
243 ## 8. Build Table 1 means/SDs (same variables as paper)
244
245
246 w_mean_sd <- function(x, w) {
247   x <- as.numeric(x)
248   w <- as.numeric(w)
249   ok <- !is.na(x) & !is.na(w)
250   x <- x[ok]; w <- w[ok]
251   if (!length(x)) return(c(mean = NA_real_, sd = NA_real_))
252   w <- w / sum(w)
253   mu <- sum(w * x)
254   var <- sum(w * (x - mu)^2)
255   c(mean = mu, sd = sqrt(var))
256 }
257
258 make_block <- function(df) {
259   out <- list(
260     Age = w_mean_sd(df$age, df$wpfinwgt),

```

```

261   'Yearly household income' =
262     w_mean_sd(df$hh_inc_year, df$wpfinwgt),
263   '401(k) assets' =
264     w_mean_sd(df$talrb6, df$wpfinwgt),
265   'IRA and Keogh assets' =
266     w_mean_sd(df$thhira6, df$wpfinwgt),
267   'Other financial assets'=
268     w_mean_sd(df$otherassets6, df$wpfinwgt),
269   'Secured debt' =
270     w_mean_sd(df$thhscdbt6, df$wpfinwgt),
271   'Unsecured debt' =
272     w_mean_sd(df$rhhuscbt6, df$wpfinwgt),
273   'Car value' =
274     w_mean_sd(df$tcarval6, df$wpfinwgt)
275 )
276 vals <- lapply(out, function(msd)
277   sprintf("%.1f\n%.1f", msd["mean"], msd["sd"]))
278 tibble::as_tibble_row(vals)
279 }
280
281 tab_all <- make_block(table1_sample) %>%
282   mutate(group = "All",
283         Observations = nrow(table1_sample))
284
285 tab_treat <- make_block(filter(table1_sample, temp == 1)) %>%
286   mutate(group = "Treatment group",
287         Observations = sum(table1_sample$temp == 1, na.rm = TRUE))
288
289 tab_ctrl <- make_block(filter(table1_sample, temp == 0)) %>%
290   mutate(group = "Control group",
291         Observations = sum(table1_sample$temp == 0, na.rm = TRUE))
292
293 table1 <- bind_rows(tab_all, tab_treat, tab_ctrl) %>%
294   relocate(group, Observations)
295

```

```

296 table1
297
298 """
299 """{r}
300 pivot_wider(names_from = month, values_from = THTOTINC, id_cols = all_of(id_vars))
301
302 """
303 """{r}
304 library(dplyr)
305 library(tidyr)
306
307 to_num <- function(x) suppressWarnings(as.numeric(x))
308
309
310
311
312 id_vars <- c("SSUID","SHHADID","EENTAID","EPPNUM")
313
314
315 ## 2. Assets: build taltb6, thhira6, otherassets6, etc.
316
317
318 make_assets_wave <- function(df, wave_num) {
319   df %>%
320     transmute(
321       SSUID, SHHADID, EENTAID, EPPNUM,
322       wave = wave_num,
323       # 401(k) balance
324       taltb = to_num(TALTB),
325       # IRA balance
326       thhira = to_num(THHIRA),
327       # Other financial assets (interest, stocks, other)
328       otherassets = to_num(THHINTBK) +
329         to_num(THHINTOT) +
330         to_num(RHHSTK) +

```

```

331         to_num(THHOTAST),
332 
333     # secured & unsecured debt
334 
335     thhscdbt = to_num(THHSCDBT),
336 
337     rhhuscbt = to_num(RHHUSCBT),
338 
339     # car values: sum over up to 3 cars
340 
341     tcarval = to_num(TCARVAL1) +
342 
343             to_num(TCARVAL2) +
344 
345             to_num(TCARVAL3)
346 
347     )
348 }
349 
350 assets_long <- bind_rows(
351 
352     make_assets_wave(t3, 3),
353 
354     make_assets_wave(t6, 6),
355 
356     make_assets_wave(t9, 9),
357 
358     make_assets_wave(t12, 12)
359 )
360 
361 assets_wide <- assets_long %>%
362 
363     pivot_wider(
364 
365         id_cols = all_of(id_vars),
366 
367         names_from = wave,
368 
369         values_from = c(talrb, thhira, otherassets, thhscdbt, rhhuscbt, tcarval),
370 
371         names_sep = ""
372 
373     )
374 
375 # Gives: talrb3, talrb6, talrb9, talrb12, etc.
376 
377 
378 
379 ## 3. Wave 7 core (SREFMON==4 only) + RMESR (employment status)
380 
381 
382 core7 <- w7 %>%
383 
384     filter(SREFMON == 4) %>% # rotation 4
385 
386     transmute(
387 
388         SSUID, SHHADID, EENTAID, EPPNUM,

```

```

366 tage = to_num(TAGE),
367 wppfinwgt = to_num(WPFINWGT),
368 eclwrk1 = to_num(ECLWRK1),
369 efnp = to_num(EFNP),
370 esex = to_num(ESEX),
371 tempall1 = to_num(TEMPALL1),
372 ejbind1 = to_num(EJBIND1),
373 tsjdate1 = to_num(TSJDATE1),
374 srotaton = to_num(SROTATON),
375 rmesr = to_num(RMESR), # <-- key new variable
376 # Year-1 household income proxy
377 hh_inc_year = to_num(THTOTINC)
378 )
379
380
381 ## 4. Topical module 7: 401(k) eligibility & reasons
382
383
384 tm7 <- t7 %>%
385 transmute(
386 SUID, SHHADID, EENTAID, EPPNUM,
387 e1taxdef = to_num(E1TAXDEF),
388 e2taxdef = to_num(E2TAXDEF),
389 e3taxdef = to_num(E3TAXDEF),
390 enoina03 = to_num(ENOINA03), # reason: haven't worked long enough (plan A)
391 enoinb03 = to_num(ENOINB03), # reason: haven't worked long enough (plan B)
392 etdeffen = to_num(ETDEFFEN) # firm offers any tax-deferred plan
393 )
394
395
396 ## 5. Merge assets + core + topical
397
398
399 dat <- assets_wide %>%
400   inner_join(core7, by = id_vars) %>%

```

```

401 inner_join(tm7, by = id_vars)

402

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

404

405

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

407

408

409 dat <- dat %>%
410   mutate(
411     # Age 2264
412     age_ok = !is.na(tage) & tage >= 22 & tage <= 64,
413
414     # For-profit: ECLWRK1 == 1
415     for_profit = (eclwrk1 == 1),
416
417     # Started job within previous year (Wave 7 rotation logic)
418     yr1jb1 = case_when(
419       srotaton == 1 ~ tsjdate1 > 19970299,
420       srotaton == 2 ~ tsjdate1 > 19970399,
421       srotaton == 3 ~ tsjdate1 > 19970499,
422       srotaton == 4 ~ tsjdate1 > 19970599,
423       TRUE ~ NA
424     ),
425
426     # Treatment: "haven't worked long enough" & firm offers plan
427     temp = as.integer(
428       (enoina03 == 1 & etdeffen == 1) |
429       (enoimb03 == 1)
430     ),
431
432     # Firm offers 401(k)
433     y401k = (temp == 1) |
434       (e1taxdef == 1) |
435       (e2taxdef == 1) |

```

```

436     (e3taxdef == 1) |
437     (etdeffen == 1 & yr1jb1 == 1),
438
439 # Notmissing asset growth (based on 6/9/12 balances)
440 notmissing = !is.na(talbtb6) & !is.na(talbtb9) & !is.na(talbtb12),
441
442 # Income missing dummy
443 incmissing = as.integer(is.na(hh_inc_year))
444 )
445
446 cat("Counts: y401k TRUE/FALSE/NA:\n")
447 print(table(dat$y401k, useNA = "ifany"))
448 cat("Counts: temp (treatment) 0/1:\n")
449 print(table(dat$temp, useNA = "ifany"))
450
451
452 ## 7. Table 1 sample + RMESR filter (employment status)
453
454 ## Stata analogue: keep if rmesr <= 4 (in labor force / recently worked)
455
456
457 table1_sample <- dat %>%
458   filter(
459     age_ok, # 2264
460     for_profit, # for-profit firm
461     yr1jb1 == 1, # started job within previous year
462     y401k, # firm offers 401(k)
463     notmissing, # have assets in 6/9/12
464     !is.na(rmesr),
465     rmesr <= 4 # <-- NEW: match authors labor-force restriction
466   )
467
468 cat("Table 1 sample size: ", nrow(table1_sample), "\n")
469 cat("Treatment (temp==1): ", sum(table1_sample$temp == 1, na.rm = TRUE), "\n")
470 cat("Control (temp==0): ", sum(table1_sample$temp == 0, na.rm = TRUE), "\n")

```

```

471
472 income_sample <- table1_sample %>%
473   filter(!is.na(hh_inc_year))
474
475 cat("Income row N (non-missing hh_inc_year): ", nrow(income_sample), "\n")
476
477
478 ## 8. Build Table 1 means/SDs
479
480
481 w_mean_sd <- function(x, w) {
482   x <- as.numeric(x)
483   w <- as.numeric(w)
484   ok <- !is.na(x) & !is.na(w)
485   x <- x[ok]; w <- w[ok]
486   if (!length(x)) return(c(mean = NA_real_, sd = NA_real_))
487   w <- w / sum(w)
488   mu <- sum(w * x)
489   var <- sum(w * (x - mu)^2)
490   c(mean = mu, sd = sqrt(var))
491 }
492
493 make_block <- function(df) {
494   out <- list(
495     Age = w_mean_sd(df$age, df$wpfinwgt),
496     'Yearly household income' =
497       w_mean_sd(df$hh_inc_year, df$wpfinwgt),
498     '401(k) assets' =
499       w_mean_sd(df$talrb6, df$wpfinwgt),
500     'IRA and Keogh assets' =
501       w_mean_sd(df$thhira6, df$wpfinwgt),
502     'Other financial assets'=
503       w_mean_sd(df$otherassets6, df$wpfinwgt),
504     'Secured debt' =
505       w_mean_sd(df$thhscdbt6, df$wpfinwgt),

```

```

506   'Unsecured debt' =
507     w_mean_sd(df$rhhuscbt6, df$wpfinwgt),
508   'Car value' =
509     w_mean_sd(df$tcarval6, df$wpfinwgt)
510 )
511 vals <- lapply(out, function(msd)
512   sprintf("%.1f\n%.1f", msd["mean"], msd["sd"]))
513 tibble::as_tibble_row(vals)
514 }
515
516 tab_all <- make_block(table1_sample) %>%
517   mutate(group = "All",
518         Observations = nrow(table1_sample))
519
520 tab_treat <- make_block(filter(table1_sample, temp == 1)) %>%
521   mutate(group = "Treatment group",
522         Observations = sum(table1_sample$temp == 1, na.rm = TRUE))
523
524 tab_ctrl <- make_block(filter(table1_sample, temp == 0)) %>%
525   mutate(group = "Control group",
526         Observations = sum(table1_sample$temp == 0, na.rm = TRUE))
527
528 table1 <- bind_rows(tab_all, tab_treat, tab_ctrl) %>%
529   relocate(group, Observations)
530
531 table1
532
533 """

```

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

```

1 ---
2 title: "Untitled"
3 format: html
4 ---

```

```

5
6  ``'{r}
7
8
9
10
11 # 1. LOAD THE DATA (RAW)
12
13
14 t3_raw <- readRDS("data/raw/rds_1/t3.rds")
15 t6_raw <- readRDS("data/raw/rds_1/t6.rds")
16 t9_raw <- readRDS("data/raw/rds_1/t9.rds")
17 t12_raw <- readRDS("data/raw/rds_1/t12.rds")
18
19 t7_raw <- readRDS("data/raw/rds_1/t7.rds")
20 w7_raw <- readRDS("data/raw/rds_1/w7.rds")
21 w8_raw <- readRDS("data/raw/rds_1/w8.rds")
22 w9_raw <- readRDS("data/raw/rds_1/w9.rds")
23
24
25 ```
26 ``'{r}
27 library(dplyr)
28 library(tidyr)
29 library(haven)
30
31 to_num <- function(x) suppressWarnings(as.numeric(x))
32
33
34 # 1. LOAD THE DATA *FIRST*
35
36
37 t3_raw <- readRDS("data/raw/rds_1/t3.rds")
38 t6_raw <- readRDS("data/raw/rds_1/t6.rds")
39 t9_raw <- readRDS("data/raw/rds_1/t9.rds")

```

```

40 t12_raw <- readRDS("data/raw/rds_1/t12.rds")
41
42 t7_raw <- readRDS("data/raw/rds_1/t7.rds")
43 w7_raw <- readRDS("data/raw/rds_1/w7.rds")
44 w8_raw <- readRDS("data/raw/rds_1/w8.rds")
45 w9_raw <- readRDS("data/raw/rds_1/w9.rds")
46
47
48 # 2. LOAD REPLICATION FILES
49
50
51 t3_rep <- read_dta("data/replication/dta/t3.dta") %>% rename_with(toupper)
52 t6_rep <- read_dta("data/replication/dta/t6.dta") %>% rename_with(toupper)
53 t9_rep <- read_dta("data/replication/dta/t9.dta") %>% rename_with(toupper)
54 t12_rep <- read_dta("data/replication/dta/t12.dta") %>% rename_with(toupper)
55
56 t7_rep <- read_dta("data/replication/dta/t7.dta") %>% rename_with(toupper)
57 w7_rep <- read_dta("data/replication/dta/w7.dta") %>% rename_with(toupper)
58 w8_rep <- read_dta("data/replication/dta/w8.dta") %>% rename_with(toupper)
59 w9_rep <- read_dta("data/replication/dta/w9.dta") %>% rename_with(toupper)
60
61
62
63 /**
64 /**
65 w9_rep <- read_dta(
66   "data/replication/dta/w9.dta",
67   col_select = c("ssuid", "shhadid", "eentaid", "epppnum", "srefmon", "thtotinc")
68 ) %>%
69   rename_with(toupper)
70 w7_rep <- read_dta(
71   "data/replication/dta/w7.dta",
72   col_select = c("ssuid", "shhadid", "eentaid", "epppnum",
73     "srefmon", "thtotinc", "tage", "wpfinwgt",
74     "eclwrk1", "efnp", "esex", "tempall1",

```

```

75         "ejbind1","tsjdate1","srotaton")
76 ) %>% rename_with(toupper)
77
78 w8_rep <- read_dta(
79   "data/replication/dta/w8.dta",
80   col_select = c("ssuid","shhadid","eentaid","epppnum","srefmon","thtotinc")
81 ) %>% rename_with(toupper)
82
83
84 /**
85
86 `{{r}}
87 library(dplyr)
88 library(tidyr)
89
90
91 ## Helper: safe numeric conversion
92
93 to_num <- function(x) suppressWarnings(as.numeric(x))
94
95
96 ## Helper: build assets for one wave
97
98 make_assets_wave <- function(df, wave_num) {
99   df %>%
100     transmute(
101       SSUID, SHHADID, EENTAID, EPPPNUM,
102       wave = wave_num,
103       # 401(k) balance
104       taltb = to_num(TALTB),
105       # IRA balance
106       thhira = to_num(THHIRA),
107       # Other financial assets (interest, stocks, other assets)
108       otherassets = to_num(THHINTBK) +
109         to_num(THHINTOT) +

```

```

110         to_num(RHHSTK) +
111             to_num(THHOTAST),
112     # secured & unsecured debt
113     thhscdbt = to_num(THHSCDBT),
114     rhhuscbt = to_num(RHHUSCBT),
115     # car values: sum over up to 3 cars
116     tcarval = to_num(TCARVAL1) +
117             to_num(TCARVAL2) +
118             to_num(TCARVAL3)
119 )
120 }
121
122
123 ## MAIN: build_dat()
124 ## takes t3,t6,t9,t12,t7,w7,w8,w9 and returns merged dataset
125 ## with: age_ok, for_profit, yr1jb1, temp, y401k, notmissing,
126 ## hh_inc_year, etc.
127
128 build_dat <- function(t3, t6, t9, t12, t7, w7, w8, w9) {
129   id_vars <- c("SSUID", "SHHADID", "EENTAID", "EPPNUM")
130
131   # 1) Assets across waves wide
132   assets_long <- bind_rows(
133     make_assets_wave(t3, 3),
134     make_assets_wave(t6, 6),
135     make_assets_wave(t9, 9),
136     make_assets_wave(t12, 12)
137   )
138
139   assets_wide <- assets_long %>%
140     pivot_wider(
141       id_cols = all_of(id_vars),
142       names_from = wave,
143       values_from = c(talrb, thhira, otherassets, thhscdbt, rhhuscbt, tcarval),
144       names_sep = ""

```

```

145   )
146
147 # 2) Wave-7 core (panel = SREFMON==4)
148 core7 <- w7 %>%
149   filter(SREFMON == 4) %>%
150   transmute(
151     SUID, SHHADID, EENTAID, EPPNUM,
152     tage = to_num(TAGE),
153     wpfinwgt = to_num(WPFINWGT),
154     eclwrk1 = to_num(ECLWRK1),
155     efnp = to_num(EFNP),
156     esex = to_num(ESEX),
157     tempall1 = to_num(TEMPALL1),
158     ejbind1 = to_num(EJBIND1),
159     tsjdate1 = to_num(TSJDATE1),
160     srotaton = to_num(SROTATON)
161   )
162
163 # 3) Year-1 income: sum THTOTINC from waves 79
164 make_inc <- function(w, tag) {
165   w %>%
166     select(SUID, SHHADID, EENTAID, EPPNUM, SREFMON, THTOTINC) %>%
167     mutate(
168       THTOTINC = to_num(THTOTINC),
169       month = paste0(tag, "_", SREFMON)
170     ) %>%
171     select(-SREFMON) %>%
172     pivot_wider(
173       id_cols = all_of(id_vars),
174       names_from = month,
175       values_from = THTOTINC
176     )
177 }
178
179 w7_inc <- make_inc(w7, "w7")

```

```

180 w8_inc <- make_inc(w8, "w8")
181 w9_inc <- make_inc(w9, "w9")
182
183 year1_income <- w7_inc %>%
184   full_join(w8_inc, by = id_vars) %>%
185   full_join(w9_inc, by = id_vars) %>%
186   mutate(
187     hh_inc_year = rowSums(
188       dplyr::select(., starts_with("w7_"), starts_with("w8_"), starts_with("w9_")),
189       na.rm = FALSE
190     )
191   ) %>%
192   select(all_of(id_vars), hh_inc_year)
193
194 # 4) Topical module 7: plan eligibility and reasons
195 tm7 <- t7 %>%
196   transmute(
197     SSUID, SHHADID, EENTAID, EPPNUM,
198     e1taxdef = to_num(E1TAXDEF),
199     e2taxdef = to_num(E2TAXDEF),
200     e3taxdef = to_num(E3TAXDEF),
201     enoina03 = to_num(ENOINA03),
202     enoinb03 = to_num(ENOINB03),
203     etdeffen = to_num(ETDEFFEN)
204   )
205
206 # 5) Merge all pieces
207 dat <- assets_wide %>%
208   inner_join(core7, by = id_vars) %>%
209   inner_join(tm7, by = id_vars) %>%
210   left_join(year1_income, by = id_vars)
211
212 # 6) Derive sample flags and d21ltalb
213 dat <- dat %>%
214   mutate(

```

```

215 age_ok = !is.na(tage) & tage >= 22 & tage <= 64,
216 for_profit = (eclwrk1 == 1),
217 yr1jb1 = dplyr::case_when(
218   srotaton == 1 ~ tsjdate1 > 19970299,
219   srotaton == 2 ~ tsjdate1 > 19970399,
220   srotaton == 3 ~ tsjdate1 > 19970499,
221   srotaton == 4 ~ tsjdate1 > 19970599,
222   TRUE ~ NA
223 ),
224 temp = as.integer(
225   (enoina03 == 1 & etdeffen == 1) |
226   (enoinb03 == 1)
227 ),
228 y401k = (temp == 1) |
229   (e1taxdef == 1) |
230   (e2taxdef == 1) |
231   (e3taxdef == 1) |
232   (etdeffen == 1 & yr1jb1 == 1),
233
234 d21ltalb = log(talbt12 + 10) - 2 * log(talbt9 + 10) + log(talbt6 + 10),
235 lnA6 = log(talbt6 + 10),
236
237 # income: allowed to be missing, we flag it
238 incmissing = as.integer(is.na(hh_inc_year)),
239
240 # tweak: require regression vars (except hh_inc_year) to be non-missing
241 notmissing = !is.na(d21ltalb) &
242   !is.na(tage) &
243   !is.na(eclwrk1) &
244   !is.na(tsjdate1) &
245   !is.na(srotaton) &
246   !is.na(wpfinwgt)
247 )
248
249 dat

```

```

250  }
251
252
253 ## Helpers to build Table 1 from a built dat
254
255 w_mean_sd <- function(x, w) {
256   x <- as.numeric(x)
257   w <- as.numeric(w)
258   ok <- !is.na(x) & !is.na(w)
259   x <- x[ok]; w <- w[ok]
260   if (!length(x)) return(c(mean = NA_real_, sd = NA_real_))
261   w <- w / sum(w)
262   mu <- sum(w * x)
263   var <- sum(w * (x - mu)^2)
264   c(mean = mu, sd = sqrt(var))
265 }
266
267 make_block <- function(df) {
268   out <- list(
269     Age = w_mean_sd(df$stage, df$wpfinwgt),
270     'Yearly household income' =
271       w_mean_sd(df$hh_inc_year, df$wpfinwgt),
272     '401(k) assets' =
273       w_mean_sd(df$talrb6, df$wpfinwgt),
274     'IRA and Keogh assets' =
275       w_mean_sd(df$thhira6, df$wpfinwgt),
276     'Other financial assets'=
277       w_mean_sd(df$otherassets6, df$wpfinwgt),
278     'Secured debt' =
279       w_mean_sd(df$thhscdbt6, df$wpfinwgt),
280     'Unsecured debt' =
281       w_mean_sd(df$rhhuscbt6, df$wpfinwgt),
282     'Car value' =
283       w_mean_sd(df$tcarval6, df$wpfinwgt)
284   )

```

```

285 vals <- lapply(out, function(msd)
286   sprintf("%.1f\n%.1f", msd["mean"], msd["sd"]))
287 tibble::as_tibble_row(vals)
288 }
289
290 make_table1 <- function(dat) {
291   # Apply same sample restrictions
292   table1_sample <- dat %>%
293     filter(
294       age_ok,
295       for_profit,
296       yr1jb1 == 1,
297       y401k,
298       notmissing
299     )
300
301   cat("Table 1 sample size: ", nrow(table1_sample), "\n")
302   cat("Treatment (temp==1): ", sum(table1_sample$temp == 1, na.rm = TRUE), "\n")
303   cat("Control (temp==0): ", sum(table1_sample$temp == 0, na.rm = TRUE), "\n")
304
305   income_sample <- table1_sample %>%
306     filter(!is.na(hh_inc_year))
307   cat("Income row N (non-missing hh_inc_year): ", nrow(income_sample), "\n")
308
309   tab_all <- make_block(table1_sample) %>%
310     mutate(group = "All",
311           Observations = nrow(table1_sample))
312
313   tab_treat <- make_block(filter(table1_sample, temp == 1)) %>%
314     mutate(group = "Treatment group",
315           Observations = sum(table1_sample$temp == 1, na.rm = TRUE))
316
317   tab_ctrl <- make_block(filter(table1_sample, temp == 0)) %>%
318     mutate(group = "Control group",
319           Observations = sum(table1_sample$temp == 0, na.rm = TRUE))

```

```

320
321 bind_rows(tab_all, tab_treat, tab_ctrl) %>%
322   relocate(group, Observations)
323 }
324
325 """
326
327 """{r}
328
329 dat_raw <- build_dat(
330   t3_raw, t6_raw, t9_raw, t12_raw,
331   t7_raw, w7_raw, w8_raw, w9_raw
332 )
333
334 table1_raw <- make_table1(dat_raw)
335 table1_raw
336 dat_rep <- build_dat(
337   t3_rep, t6_rep, t9_rep, t12_rep,
338   t7_rep, w7_rep, w8_rep, w9_rep
339 )
340
341 table1_rep <- make_table1(dat_rep)
342 table1_rep
343
344 """
345
346
347 """{r}
348 id_vars <- c("SSUID", "SHHADID", "EENTAID", "EPPNUM")
349
350 normalize_ids <- function(df) {
351   df %>%
352     mutate(
353       SSUID = sub("^0+", "", as.character(SSUID)),
354       SHHADID = sub("^0+", "", as.character(SHHADID)),

```

```

355     EENTAID = sub("^0+", "", as.character(EENTAID)),
356     EPPNUM = sub("^0+", "", as.character(EPPNUM))
357   )
358 }
359
360
361 /**
362 **'{r}
363 # Rebuild from scratch using existing build_dat()
364 dat_raw <- build_dat(
365   t3_raw, t6_raw, t9_raw, t12_raw,
366   t7_raw, w7_raw, w8_raw, w9_raw
367 )
368
369 dat_rep <- build_dat(
370   t3_rep, t6_rep, t9_rep, t12_rep,
371   t7_rep, w7_rep, w8_rep, w9_rep
372 )
373
374 # Normalize IDs
375 dat_raw_id <- normalize_ids(dat_raw)
376 dat_rep_id <- normalize_ids(dat_rep)
377
378 # Check ID sets really match
379 raw_ids <- dat_raw_id %>% select(all_of(id_vars)) %>% distinct()
380 rep_ids <- dat_rep_id %>% select(all_of(id_vars)) %>% distinct()
381
382 rep_not_raw <- dplyr::anti_join(rep_ids, raw_ids, by = id_vars)
383 raw_not_rep <- dplyr::anti_join(raw_ids, rep_ids, by = id_vars)
384
385 nrow(rep_not_raw) # should be 0
386 nrow(raw_not_rep) # should be 0
387
388 /**

```

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

```
1 ---  
2 title: "Untitled"  
3 format: html  
4 ---  
5  
6  
7 {{{r}}}  
8  
9  
10 library(dplyr)  
11  
12 main <- dat_raw_aligned # NOW USING dat_raw_aligned, but switch to data_raw to use 1115 sample size  
13  
14 to_num <- function(x) suppressWarnings(as.numeric(x))  
15  
16 safe_log <- function(x) {  
17   ifelse(!is.na(x) & x > -10, log(x + 10), NA_real_)  
18 }  
19  
20 main <- main %>%  
21   mutate(  
22     weight = wpfinwgt,  
23     tagesq = tage^2,  
24  
25     ## log levels at wave 6  
26     ltalrb6 = safe_log(talrb6),  
27     lthhira6 = safe_log(thhira6),  
28     lotherassets6 = safe_log(otherassets6),  
29     lthhsedb6 = safe_log(thhsedb6),  
30     lrhhuscbt6 = safe_log(rhhuscbt6),  
31     ltcarval6 = safe_log(tcarval6),  
32  
33     ## second differences (12 29 + 6)  
34     d21lthhira = safe_log(thhira12) - 2*safe_log(thhira9) + safe_log(thhira6),
```

```

35   d21lotherassets = safe_log(otherassets12) - 2*safe_log(otherassets9) + safe_log(otherassets6),
36   d21lhhscdbt = safe_log(thhscdbt12) - 2*safe_log(thhscdbt9) + safe_log(thhscdbt6),
37   d21lrhuscbt = safe_log(rhhuscbt12) - 2*safe_log(rhhuscbt9) + safe_log(rhhuscbt6),
38   d21ltcarval = safe_log(tcarval12) - 2*safe_log(tcarval9) + safe_log(tcarval6)
39 ) %>%
40 ## apply papers required sample restrictions
41 filter(
42   age_ok,
43   for_profit,
44   yr1jb1
45 )
46
47 /**
48 ````{r}
49
50 library(sandwich)
51 library(lmtest)
52
53 run_cluster_reg <- function(df, formula) {
54
55   df <- df %>%
56     dplyr::group_by(SSUID, SHHADID) %>%
57     dplyr::mutate(hid = dplyr::cur_group_id()) %>%
58     dplyr::ungroup()
59
60   mod <- lm(formula, data = df, weights = df$weight)
61
62   vc <- sandwich::vcovCL(mod, cluster = df$hid)
63   ct <- lmtest::coeftest(mod, vcov. = vc)
64
65   list(model = mod, coeftest = ct)
66 }
67
68 /**
69 ````{r}

```

```

70
71
72 var_roots <- c("taltb", "thhira", "otherassets",
73   "thhscdbt", "rhhuscbt", "tcarval")
74
75 table2_results <- lapply(var_roots, function(vr) {
76
77   dv <- paste0("d21l", vr)
78   lag6 <- paste0("l", vr, "6")
79
80   df <- main %>%
81     filter(
82       y401k == 1,
83       notmissing,
84       !is.na(.data[[dv]]))
85   )
86
87   ## Formulae:
88   f1 <- reformulate("temp", response = dv)
89
90   covars2 <- c("temp", "tage", "tagesq", "hh_inc_year", "incmissing")
91   f2 <- reformulate(covars2, response = dv)
92
93   covars3 <- c("temp", lag6, "tage", "tagesq", "hh_inc_year", "incmissing")
94   f3 <- reformulate(covars3, response = dv)
95
96   s1 <- run_cluster_reg(df, f1)
97   s2 <- run_cluster_reg(df, f2)
98   s3 <- run_cluster_reg(df, f3)
99
100  extract_temp <- function(ct) {
101    if (!"temp" %in% rownames(ct)) return(c(coef=NA, se=NA))
102    c(coef = ct["temp","Estimate"],
103      se = ct["temp","Std. Error"])
104  }

```

```

105
106 c1 <- extract_temp(s1$coeftest)
107 c2 <- extract_temp(s2$coeftest)
108 c3 <- extract_temp(s3$coeftest)

109
110 data.frame(
111   outcome = vr,
112   spec = c("no_controls", "controls", "controls_plus_lag"),
113   coef_temp = c(c1["coef"], c2["coef"], c3["coef"]),
114   se_temp = c(c1["se"], c2["se"], c3["se"]),
115   n = c(nobs(s1$model), nobs(s2$model), nobs(s3$model))
116 )
117 })
118
119 """
120
121 """
122
123
124 table2_raw <- bind_rows(table2_results)
125 table2_raw
126
127 """
128 """
129
130
131 library(dplyr)
132
133 # Rename columns to avoid name collisions
134 t_rep <- table2 %>%
135   rename(
136     coef_rep = coef_temp,
137     se_rep = se_temp,
138     N_rep = n
139   )

```

```

140
141 t_raw <- table2_raw %>%
142   rename(
143     coef_raw = coef_temp,
144     se_raw = se_temp,
145     N_raw = n
146   )
147
148 # Merge side-by-side by outcome + spec
149 table2_compare <- t_rep %>%
150   inner_join(t_raw, by = c("outcome", "spec")) %>%
151   arrange(outcome, spec)
152
153 table2_compare
154
155 '''

```

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

```

1 ---
2 title: "Untitled"
3 format: html
4 ---
5
6 '''{r}
7 library(dplyr)
8
9 main <- dat_raw_aligned # NOW USING dat_raw_aligned, but switch to data_raw to use 1115 sample size
10
11 to_num <- function(x) suppressWarnings(as.numeric(x))
12
13 safe_log <- function(x) {
14   ifelse(!is.na(x) & x > -10, log(x + 10), NA_real_)
15 }
16

```

```

17 main <- main %>%
18   mutate(
19     weight = wptfinwgt,
20     tagesq = tage^2,
21
22     ## log levels at wave 6
23     ltaltb6 = safe_log(taltb6),
24     lthhira6 = safe_log(thhira6),
25     lotherassets6 = safe_log(otherassets6),
26     lthhscdbt6 = safe_log(thhscdbt6),
27     lrhhuscbt6 = safe_log(rhhuscbt6),
28     ltcarval6 = safe_log(tcarval6),
29
30     ## second differences (12 29 + 6)
31     d21lthhira = safe_log(thhira12) - 2*safe_log(thhira9) + safe_log(thhira6),
32     d21lotherassets = safe_log(otherassets12) - 2*safe_log(otherassets9) + safe_log(otherassets6),
33     d21lthhscdbt = safe_log(thhscdbt12) - 2*safe_log(thhscdbt9) + safe_log(thhscdbt6),
34     d21lrhhuscbt = safe_log(rhhuscbt12) - 2*safe_log(rhhuscbt9) + safe_log(rhhuscbt6),
35     d21ltcarval = safe_log(tcarval12) - 2*safe_log(tcarval9) + safe_log(tcarval6)
36   ) %>%
37
38   ## apply papers required sample restrictions
39   filter(
40     age_ok,
41     for_profit,
42     yr1jb1
43   )
44
45   '''
46   '''
47   ``{r}
48
49   library(sandwich)
50   library(lmtest)
51
52   run_cluster_reg <- function(df, formula) {

```

```

52 df <- df %>%
53   dplyr::group_by(SSUID, SHHADID) %>%
54   dplyr::mutate(hid = dplyr::cur_group_id()) %>%
55   dplyr::ungroup()
56
57 mod <- lm(formula, data = df, weights = df$weight)
58
59 vc <- sandwich::vcovCL(mod, cluster = df$hid)
60 ct <- lmtest::coeftest(mod, vcov. = vc)
61
62 list(model = mod, coeftest = ct)
63 }
64
65 /**
66 /**
67 /**
68 /**
69 var_roots <- c("taltb", "thhira", "otherassets",
70               "thhscdbt", "rhhuscbt", "tcarval")
71
72 table2_results <- lapply(var_roots, function(vr) {
73
74   dv <- paste0("d21l", vr)
75   lag6 <- paste0("l", vr, "6")
76
77   df <- main %>%
78     filter(
79       y401k == 1,
80       notmissing,
81       !is.na(.data[[dv]])
82     )
83
84   ## Formulae:
85   f1 <- reformulate("temp", response = dv)
86

```

```

87 covars2 <- c("temp", "tage", "tagesq", "hh_inc_year", "incmissing")
88 f2 <- reformulate(covars2, response = dv)
89
90 covars3 <- c("temp", lag6, "tage", "tagesq", "hh_inc_year", "incmissing")
91 f3 <- reformulate(covars3, response = dv)
92
93 s1 <- run_cluster_reg(df, f1)
94 s2 <- run_cluster_reg(df, f2)
95 s3 <- run_cluster_reg(df, f3)
96
97 extract_temp <- function(ct) {
98   if (!"temp" %in% rownames(ct)) return(c(coef=NA, se=NA))
99   c(coef = ct["temp","Estimate"],
100    se = ct["temp","Std. Error"])
101 }
102
103 c1 <- extract_temp(s1$coeftest)
104 c2 <- extract_temp(s2$coeftest)
105 c3 <- extract_temp(s3$coeftest)
106
107 data.frame(
108   outcome = vr,
109   spec = c("no_controls", "controls", "controls_plus_lag"),
110   coef_temp = c(c1["coef"], c2["coef"], c3["coef"]),
111   se_temp = c(c1["se"], c2["se"], c3["se"]),
112   n = c(nobs(s1$model), nobs(s2$model), nobs(s3$model))
113 )
114 })
115 """
116 """
117 """
118 """
119
120
121 table2_raw_aligned <- bind_rows(table2_results)

```

```

122 | table2_raw_aligned
123 |
124 | """
125 | """{r}
126 |
127 |
128 | library(dplyr)
129 |
130 | # Rename columns to avoid name collisions
131 | t_rep <- table2 %>%
132 |   rename(
133 |     coef_rep = coef_temp,
134 |     se_rep = se_temp,
135 |     N_rep = n
136 |   )
137 |
138 | t_raw <- table2_raw %>%
139 |   rename(
140 |     coef_raw = coef_temp,
141 |     se_raw = se_temp,
142 |     N_raw = n
143 |   )
144 |
145 | # Merge side-by-side by outcome + spec
146 | table2_compare <- t_rep %>%
147 |   inner_join(t_raw, by = c("outcome", "spec")) %>%
148 |   arrange(outcome, spec)
149 |
150 | table2_compare

```

B.7 Table 2 (Replication) (table2.rep.qmd)

```

1 ---
2 title: "Untitled"
3 format: html

```

```

4   ---
5
6   '''{r}
7
8
9 library(dplyr)
10
11 main <- dat_rep # use dat_rep as the analysis dataset
12
13 to_num <- function(x) suppressWarnings(as.numeric(x))
14
15 safe_log <- function(x) {
16   ifelse(!is.na(x) & x > -10, log(x + 10), NA_real_)
17 }
18
19 main <- main %>%
20   mutate(
21     weight = wpfinwgt,
22     tagesq = tage^2,
23
24     ## log levels at wave 6
25     ltalrb6 = safe_log(talrb6),
26     lthhira6 = safe_log(thhira6),
27     lotherassets6 = safe_log(otherassets6),
28     lthhscdbt6 = safe_log(thhscdbt6),
29     lrhhuscbt6 = safe_log(rhhuscbt6),
30     ltcarval6 = safe_log(tcarval6),
31
32     ## second differences
33     d21lthhira = safe_log(thhira12) - 2*safe_log(thhira9) + safe_log(thhira6),
34     d21lotherassets = safe_log(otherassets12) - 2*safe_log(otherassets9) + safe_log(otherassets6),
35     d21lthhscdbt = safe_log(thhscdbt12) - 2*safe_log(thhscdbt9) + safe_log(thhscdbt6),
36     d21lrhhuscbt = safe_log(rhhuscbt12) - 2*safe_log(rhhuscbt9) + safe_log(rhhuscbt6),
37     d21ltcarval = safe_log(tcarval12) - 2*safe_log(tcarval9) + safe_log(tcarval6)
38 )

```

```

39
40 ## Restrict to the papers working sample
41 main <- main %>%
42   filter(
43     age_ok,
44     for_profit,
45     yr1jb1
46   )
47
48
49
50 /**
51
52 /**{r}
53
54
55 library(sandwich)
56 library(lmtest)
57
58 run_cluster_reg <- function(df, formula) {
59
60   df <- df %>%
61     dplyr::group_by(SSUID, SHHADID) %>%
62     dplyr::mutate(hid = dplyr::cur_group_id()) %>%
63     dplyr::ungroup()
64
65   mod <- lm(formula, data = df, weights = df$weight)
66
67   vc <- sandwich::vcovCL(mod, cluster = df$hid)
68   ct <- lmtest::coeftest(mod, vcov. = vc)
69
70   list(model = mod, coeftest = ct)
71 }
72
73 /**

```

```

74  ' ' '{r}
75
76
77 var_roots <- c("taltb", "thhira", "otherassets",
78           "thhscdbt", "rhhuscbt", "tcarval")
79
80 table2_results <- lapply(var_roots, function(vr) {
81
82   dv <- paste0("d211", vr)
83   lag6 <- paste0("l", vr, "6")
84
85   df <- main %>%
86     filter(
87       y401k == 1,
88       notmissing,
89       !is.na(.data[[dv]]))
90
91
92   ## Spec 1
93   f1 <- reformulate("temp", response = dv)
94
95   ## Spec 2
96   covars2 <- c("temp", "tage", "tagesq", "hh_inc_year", "incmissing")
97   f2 <- reformulate(covars2, response = dv)
98
99   ## Spec 3
100  covars3 <- c("temp", lag6, "tage", "tagesq", "hh_inc_year", "incmissing")
101  f3 <- reformulate(covars3, response = dv)
102
103  s1 <- run_cluster_reg(df, f1)
104  s2 <- run_cluster_reg(df, f2)
105  s3 <- run_cluster_reg(df, f3)
106
107  extract_temp <- function(ct) {
108    if (!"temp" %in% rownames(ct)) return(c(coef=NA, se=NA))

```

```

109   c(coef = ct["temp", "Estimate"],
110     se = ct["temp", "Std. Error"])
111 }
112
113 c1 <- extract_temp(s1$coeftest)
114 c2 <- extract_temp(s2$coeftest)
115 c3 <- extract_temp(s3$coeftest)
116
117 data.frame(
118   outcome = vr,
119   spec = c("no_controls", "controls", "controls_plus_lag"),
120   coef_temp = c(c1["coef"], c2["coef"], c3["coef"]),
121   se_temp = c(c1["se"], c2["se"], c3["se"]),
122   n = c(nobs(s1$model), nobs(s2$model), nobs(s3$model))
123 )
124 })
125
126 """
127 '''{r}
128
129
130 table2 <- bind_rows(table2_results)
131 table2
132
133 """

```

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

```

1 ---
2 title: "Table 3"
3 format: pdf
4 ---
5
6 '''{r}
7 library(dplyr)

```

```

8 library(splines)
9 library(purrr)
10 library(tibble)
11
12 ## ---- helpers ----
13
14 make_d21 <- function(df, base) {
15   a6 <- df[[paste0(base, "6")]]
16   a9 <- df[[paste0(base, "9")]]
17   a12 <- df[[paste0(base, "12")]]
18   log(a12 + 10) - 2 * log(a9 + 10) + log(a6 + 10)
19 }
20
21 make_d21_ihs <- function(df, base) {
22   a6 <- df[[paste0(base, "6")]]
23   a9 <- df[[paste0(base, "9")]]
24   a12 <- df[[paste0(base, "12")]]
25   asinh(a12 + 10) - 2 * asinh(a9 + 10) + asinh(a6 + 10)
26 }
27
28 # same sample as main analysis
29 make_reg_sample <- function(dat) {
30   dat %>%
31     filter(
32       age_ok,
33       for_profit,
34       yr1jb1 == 1,
35       y401k,
36       notmissing
37     )
38 }
39
40 grab_become <- function(fit, name = "temp") {
41   s <- summary(fit)$coef
42   tibble(

```

```

43     estimate = s[name, "Estimate"],
44     se = s[name, "Std. Error"]
45   )
46 }
47
48 ## Panel A: temp + spline in Wave 6 + controls
49 run_panelA <- function(df, dep, base6) {
50   f <- as.formula(
51     paste0(
52       dep, " ~ temp + bs(", base6, ", df = 20) + ",
53       "tage + I(tage^2) + hh_inc_year + incmissing"
54     )
55   )
56   lm(f, data = df, weights = wpfinwgt)
57 }
58
59 ## Panel B: temp + Wave6 + temp:Wave6 + controls
60 run_panelB <- function(df, dep, base6) {
61   f <- as.formula(
62     paste0(
63       dep, " ~ temp + ", base6, " + temp:", base6, " + ",
64       "tage + I(tage^2) + hh_inc_year + incmissing"
65     )
66   )
67   lm(f, data = df, weights = wpfinwgt)
68 }
69
70 ## Panel C: IHS outcome, temp + log(A6) + controls
71 run_panelC <- function(df, dep_ihs, base6) {
72   f <- as.formula(
73     paste0(
74       dep_ihs, " ~ temp + log(", base6, " + 10) + ",
75       "tage + I(tage^2) + hh_inc_year + incmissing"
76     )
77   )

```

```

78 lm(f, data = df, weights = wpdfinwgt)
79 }
80
81 ## Table 3 rep func
82
83 run_table3_rep <- function(dat_rep) {
84   df <- make_reg_sample(dat_rep)
85
86   # build outcomes (log 2nd diffs)
87   df$d21_401k <- make_d21(df, "taltb")
88   df$d21_ira <- make_d21(df, "thhira")
89   df$d21_other <- make_d21(df, "otherassets")
90   df$d21_sec <- make_d21(df, "thhscdbt")
91   df$d21_unsec <- make_d21(df, "rhhuscbt")
92   df$d21_car <- make_d21(df, "tcarval")
93
94   # IHS for Panel C
95   df$d21_401k_ihs <- make_d21_ihs(df, "taltb")
96   df$d21_ira_ihs <- make_d21_ihs(df, "thhira")
97   df$d21_other_ihs <- make_d21_ihs(df, "otherassets")
98   df$d21_sec_ihs <- make_d21_ihs(df, "thhscdbt")
99   df$d21_unsec_ihs <- make_d21_ihs(df, "rhhuscbt")
100  df$d21_car_ihs <- make_d21_ihs(df, "tcarval")
101
102  outcomes <- c("d21_401k", "d21_ira", "d21_other",
103                "d21_sec", "d21_unsec", "d21_car")
104  bases6 <- c("taltb6", "thhira6", "otherassets6",
105              "thhscdbt6", "rhhuscbt6", "tcarval6")
106  ihs_out <- c("d21_401k_ihs", "d21_ira_ihs", "d21_other_ihs",
107                "d21_sec_ihs", "d21_unsec_ihs", "d21_car_ihs")
108
109 ## Panel A
110 fitsA <- Map(function(y, b6) run_panelA(df, y, b6), outcomes, bases6)
111 panelA <- map_dfr(fitsA, grab_become, .id = "col")
112 panelA$R2 <- map_dbl(fitsA, ~ summary(.x)$r.squared)

```

```

113
114 ## Panel B
115 fitsB <- Map(function(y, b6) run_panelB(df, y, b6), outcomes, bases6)
116 panelB <- map_dfr(fitsB, grab_become, .id = "col")
117 panelB$R2 <- map_dbl(fitsB, ~ summary(.x)$r.squared)

118
119 wave6_int <- map_dfr(fitsB, function(fit) {
120   s <- summary(fit)$coef
121   int_row <- grep("^temp:", rownames(s)) # temp:base6
122   base_row <- grep("6$", rownames(s))[1] # base6 term
123   tibble(
124     coef_wave6x = s[int_row, "Estimate"],
125     se_wave6x = s[int_row, "Std. Error"],
126     coef_wave6 = s[base_row, "Estimate"],
127     se_wave6 = s[base_row, "Std. Error"]
128   )
129 }, .id = "col")

130
131 panelB <- bind_cols(panelB, wave6_int[,-1])

132
133 ## Panel C
134 fitsC <- Map(function(y, b6) run_panelC(df, y, b6), ihs_out, bases6)
135 panelC <- map_dfr(fitsC, grab_become, .id = "col")
136 panelC$R2 <- map_dbl(fitsC, ~ summary(.x)$r.squared)

137
138 list(
139   panelA = panelA,
140   panelB = panelB,
141   panelC = panelC
142 )
143 }
144
145 """
146
147 """

```

```

148 ## use it
149 res3_rep <- run_table3_rep(dat_rep)
150
151 res3_rep$panelA
152 res3_rep$panelB
153 res3_rep$panelC
154 """
155 """
156 res3_aligned <- run_table3_rep(dat_raw_aligned)
157
158 res3_aligned$panelA
159 res3_aligned$panelB
160 res3_aligned$panelC
161 """
162 """
163 res3_raw <- run_table3_rep(dat_raw)
164 res3_raw$panelA
165 res3_raw$panelB
166 res3_raw$panelC
167
168 """
169 """
170 library(dplyr)
171
172 ## Panel A: only 4 cols + sample/panel
173 panelA_combined <- bind_rows(
174   res3_rep$panelA %>% mutate(sample = "replication", panel = "A"),
175   res3_aligned$panelA %>% mutate(sample = "aligned", panel = "A"),
176   res3_raw$panelA %>% mutate(sample = "raw", panel = "A")
177 ) %>%
178   relocate(sample, panel)
179
180 ## Panel B: has extra wave6 columns (same structure across samples)
181 panelB_combined <- bind_rows(
182   res3_rep$panelB %>% mutate(sample = "replication", panel = "B"),

```

```
183   res3_aligned$panelB %>% mutate(sample = "aligned", panel = "B"),
184   res3_raw$panelB %>% mutate(sample = "raw", panel = "B")
185 ) %>%
186   relocate(sample, panel)
187
188 ## Panel C: same idea
189 panelC_combined <- bind_rows(
190   res3_rep$panelC %>% mutate(sample = "replication", panel = "C"),
191   res3_aligned$panelC %>% mutate(sample = "aligned", panel = "C"),
192   res3_raw$panelC %>% mutate(sample = "raw", panel = "C")
193 ) %>%
194   relocate(sample, panel)
195
196 write.csv(panelA_combined, "table3A.csv", row.names = FALSE)
197 write.csv(panelB_combined, "table3B.csv", row.names = FALSE)
198 write.csv(panelC_combined, "table3C.csv", row.names = FALSE)
199 ````
```