Pay-As-You-Go Insurance: Experimental Evidence on Consumer Demand and Behavior Replication Kit README

Raymond Kluender*

October 23, 2023

1 Overview

This document describes the data, codebase, and process for replicating the exhibits in the article, "Pay-As-You-Go Insurance: Experimental Evidence on Consumer Demand and Behavior." Please reference the paper and its Internet Appendix for additional details. If you have any further questions or issues, please email the author, Raymond Kluender (rkluender@hbs.edu).

Section 2 describes our codebase, gives instructions for running the code, and discusses computational requirements. In Section 3, we provide citation and a data availability statement for our data sources.

2 Instructions

The replication kit includes `code map.xlsx`, containing a list of the code in the codebase and their inputs and outputs. Outputs include datasets used in analysis or exhibits used in the paper. The replication kit includes the code needed to replicate all figures and tables from the raw data. For confidentiality reasons, I am unable to share the underlying data and instead provide pseudodata that can be used to run the analysis programs. To run this code in Stata, please follow the instructions below.

2.1 Computational Requirements

Code is written in Stata version 17 and executed on a machine running MacOS Ventura. The following Stata packages are used and their associated .ado files must be installed in order to execute the full replication kit. The following .ado file installations are included in the replication kit under "`code/ado`" and can be moved to the user's own "ado" folder or installed separately:

- `binscatter`
- `carryforward`
- `ftools`
- `reghdfe`
- `regsave`
- `unique`

2.2 Preliminaries

Before running, do the following:

- Update the global variable "`root`" in "`code/set_environment.do`" to the filepath of the replication kit.
- Update the global variable "`hugo_repkit_root`" in "`build.do`" to the same filepath.
- Update the global variable "`hugo_repkit_root`" in "`analysis.do`" to the same filepath.

*I thank Calvin Jahnke and Eleanor Jenke for invaluable research assistance in compiling the code base and constructing this replication kit.

2.3 Running the Code

The code for the paper is run using two main files which each call a series of subprograms as described. They also create the relevant subfolders within the repository for interim and clean data files, as well as the output for the analysis.

- `code/build.do`: Converts raw inputs into cleaned data files (excluded from replication kit)
  - `code/build/hugo/Clean Mturk data.do`: Processes string entries and selects median vehicle value.
  - `code/build/hugo/hugo data build.do`: Imports users included in the experiment, selects variables from an internal Hugo dashboard and merges them to users, and merges in the vehicle values from MTurk.
  - `code/build/insurance actions/Create action level datasets.do`: Merges user-level data with ledger of purchase actions (balance activity), orders transactions and processes timestamps, and saves out datasets with balance actions and insurance coverage actions as measured by ledger of balance activity.
  - `code/build/insurance actions/Create collapsed action hourly.do`: Uses the balance actions dataset, merges in cancellation dates, creates indicators for bundled purchases, uses `tsfill` to generate a fully populated panel of activity at the user-hour level.
  - `code/build/insurance actions/Create hourly any insurance.do`: Merges in indicators for other coverage (prior to and during the experiment) and outputs one dataset for heterogeneity analysis and another with start and end hours of coverage.
  - `code/build/insurance actions/Collapsed action any insurance.do`: Merges insurance actions within Hugo through the experiment with actions through any insurance carrier and aligns timing, saves out a user-hour level dataset, collapses dataset to the daily level and saves out a user-day level dataset, collapses to the user level and defines and labels outcomes for analysis at the user level.
  - `code/build/insurance actions/Merge stripe payments.do`: Merges in Stripe transactions data and align with transactions observed in the balance ledger.
  - `code/build/credit data/Merge experian data.do`: Crosswalks experiment IDs to Experian IDs and merges in credit reports for the study sample, additionally merges in the random sample, processes and labels the credit report variables, and saves out datasets for the random sample and just the experiment sample (alongside the user-level outcomes created above).
  - `code/build/credit data/Merge clarity data.do`: Loads inquiries datasets for a random sample of Clarity users, the random sample of Experian users for whom I received credit reports, and the experiment sample, processes and saves the inquiries, do the same for the tradeline datasets, process variables, and merge into the relevant datasets. Ultimately, this creates the "`collapsed action with credit.dta`" dataset which is used for the majority of the analysis and is represented by the "`userlevel_pseudodata.dta`" provided in the replication kit.
- `code/analysis.do`: Runs analysis .do files. See Table and Figure notes in the paper for additional details. Note: I did not generate pseudodata for some appendix exhibits that rely on data that is not used in the main text of the paper. The .do files in italics can only be partially run on the pseudodata (or not at all), but the .do files are included in the replication kit for reference.
  - `code/analyze/main/premium distributions.do`: Produces Figure 1.
  - `code/analyze/main/sumstats.do`: Produces Table 1.
  - `code/analyze/main/ITT daily regression and barchart.do`: Produces Figure 2, Tables 2 & 4.
  - `code/analyze/main/ITT dayrate regression and plots.do`: Produces Figure 3.
  - `code/analyze/main/demand elasticities.do`: Produces Table 3 and Appendix Table A6.
  - `code/analyze/main/bundle choice.do`: Produces Figure 4.
  - `code/analyze/main/ITT over time.do`: Produces Figure 5.
  - `code/analyze/appendix/Hourly viz.do`: Produces Appendix Figure A4.
  - *`code/analyze/appendix/Payment and use day distributions.do`*: Produces Appendix Figure A5.
  - *`code/analyze/appendix/appendix_sumstats.do`*: Produces Appendix Tables A3 and A4.
  - `code/analyze/appendix/appendix_ITT_specs.do`: Produces Appendix Figure A3 and Appendix Tables A5, A7, A9, and A10.
  - `code/analyze/appendix/cost of credit.do`: Produces Appendix Tables A2 and *A8*.
  - *`code/analyze/appendix/ITT Renewals.do`*: Produces Appendix Figure A6.
  - *`code/analyze/In text values.do`*: Produces analysis for numbers cited in the paper.

3 Data

3.1 Data Availability

Data used in the paper come from several sources with administrative data from Hugo Insurance Services as the primary source. This data include insurance application information, a ledger of purchase actions, and a ledger of coverage actions. To supplement the application information, I hired three contractors on Mechanical Turk to obtain each vehicle's value from its make, model, and year using the CARFAX valuation tool. To supplement the ledger of purchase actions with all attempted purchases, I also incorporate transaction-level data from Hugo Insurance Services collected by Stripe, their payment processor. To provide more context on the participants in the randomized control trial, I additionally purchased credit reports from Experian and alternative credit reports tracking subprime borrowing from Clarity Services. For the Experian and Clarity datasets, we received data for all Hugo users with a credit report and for a random sample of 1,000,000 US individuals for comparison.

I am prohibited from sharing any of these datasets publicly and instead provide pseudo-data using approximate distributions of the underlying variables to demonstrate how the code functions. Please note that this means the exhibits produced by the replication kit will not match those in the article.

3.2 Input Datasets

The following list contains all input datasets to our codebase. The included `code map.xlsx` shows where each data file is called and the resulting output.

- MTurk
  - **MTurk/makemodel_to_carvalues**
- Hugo
  - **hugo/users**
  - **hugo/hugo_dashboard**
  - **hugo/new_balance**
  - **hugo/new_coverage_activity**
  - **hugo/all_insurance**
  - **hugo/stripe/stripe_payments.csv**
- Credit Reports
  - Experian
    - **Experian/202007301A_Reseq_NoPII**
    - **Experian/202007301A_MEMBER_022019**
    - **Experian/202007301A_SAMPLE_022019**
  - Clarity
    - **Clarity/Clarity_Sample_Inquiries_Return_to_Harvard_20200724**
    - **Clarity/Harvard_Sample_Inquiries_Return_from_Clarity_20200724**
    - **Clarity/Members_Inquiries_Return_to_Harvard_20200724**
    - **Clarity/Clarity_Sample_Tradelines_Return_to_Harvard_20200724**
    - **Clarity/Harvard_Sample_Tradelines_Return_from_Clarity_20200724**
    - **Clarity/Members_Tradelines_Return_to_Harvard_20200724**

3.3 Pseudodata

The pseudodata included in the replication kit is divided among three files:

- **userlevel_pseudodata.do**: User-level data
- **userdaylevel_pseudodata.do**: User-day-level data
- **userhourlevel_pseudodata.do**: User-hour-level data

A description of each variable is included below:

- **id**: User ID

- **daily**: Whether the user is offered pay-as-you-go (0 or 1)
- **price**: Price treatment group (control, base, low, or high)
- **bundle**: Whether the user is offered the bundle discount (0 or 1)
- **totalagentpremium**: The user's quoted 3-month premium
- **basedayrate**: Translated daily premium (equals 1.667*totalagentpremium/90)
- **dayrate**: Base day adjusted for price treatment
- **bound**: Insurance take-up through Hugo (0 or 1)
- **anybound7**: Take-up of any insurance within 7 days of sign-up (0 or 1)
- **daysnonzeroreserve90**: Days with Hugo coverage (integer between 0 and 92)
- **daysnonzeroreserve90cr**: Days with any insurance coverage (integer between 0 and 92)
- **totdayscoverage90**: Days insured by Hugo (integer between 0 and 92)
- **totdayscoverage90cr**: Days insured by any provider (integer between 0 and 92)
- **threemonther**: Insured by Hugo at the end of the 3-month experiment (0 or 1)
- **threemonthcrer**: Insured by any provider at the end of the 3-month experiment (0 or 1)
- **num3**: Number of 3-day PAYG purchases
- **num7**: Number of 7-day PAYG purchases
- **num14**: Number of 14-day PAYG purchases
- **num30**: Number of 30-day PAYG purchases
- **paydays1**: Number of days in the user's first PAYG purchase (3, 7, 14, or 30)
- **payments**: Number of payments
- **makemodelyearvalue_private**: Vehicle value
- **car_year**: Vehicle year
- **age**: Age of user
- **income_insight_score**: User's Income Insight Score
- **vantage_v4_score**: User's VantageScore 4.0
- **totinq**: User's total number of credit inquiries
- **totalrevlimit**: User's revolving trade limit
- **cardlimit**: User's credit limit
- **cardbalance**: User's credit card balance
- **isconstrained**: Whether user's credit utilization is above 100%
- **autoloan**: Whether the user has an auto loan
- **autoloanamount**: User's auto loan amount
- **medcoll**: User's medical collections amount
- **nonmedcoll**: User's non-medical collections amount
- **clartotinq**: User's total number of credit inquiries (from Clarity)
- **clar_limits**: User's credit limit (from Clarity)
- **clar_balance**: User's credit card balance (from Clarity)

- **daycr**: Day index where day 0 equals Hugo account creation date
- **day**: Day index where day 0 equals Hugo insurance take-up date if applicable, otherwise equals account creation date
- **hourcr**: Hour index where hour 0 equals Hugo account creation hour
- **hour**: Hour index where hour 0 equals Hugo insurance take-up hour if applicable, otherwise equals account creation hour
- **hascov**: Whether the user is covered by Hugo
- **hasres**: Whether the user has Hugo reserve days
- **anyins**: Whether the user is covered by any insurance