README for
Inertia, Market Power, and Adverse Selection in Health Insurance:
Evidence from the ACA Exchanges

1 Overview

This document provides a description of the data and code used to replicate the results in "Inertia, Market Power, and Adverse Selection in Health Insurance: Evidence from the ACA Exchanges." Data files (if shareable) are stored in the *Data* folder and code files are stored in the *Code* folder. Most code is run in either the `R` programming language (version 3.5) or `Julia` programming language (version 1.8). All `R` code files have the suffix `.R` and all `Julia` code files have the suffix `.jl`. One program, which creates a table, was run in Stata (version 17) and has the suffix `.do`.

2 Raw Data

The paper uses the following raw data sources:

1. *Enrollment Data*

   This data file contains 2014-2019 consumer-level enrollment data from Covered California. It was obtained through a Public Records Act (PRA) request by emailing `pra@covered.ca.gov`. We are unable to upload it to this replication package. In the code, the data file is loaded as `pra_07192019.csv`. Raw data variables include:

   (a) `ahbx_case_id_x` - Household id
   (b) `indv_id_x` - Individual ID
   (c) `enrlee_enrlmnt_yr` - Year
   (d) `enrlee_status` - Enrolled at time of data extract
   (e) `subsidy_eligible` - Eligible for subsidies
   (f) `metal_level_enhanced` - Metal Tier, including AV if eligible for CSRs
   (g) `plan_name` - Plan name
   (h) `issuer_name` - Insurer
   (i) `plan_network_type` - Type of network (e.g., HMO, PPO)
   (j) `hios_id_16` - Health Insurance Oversight System (HIOS) ID
   (k) `race_ethnicity` - Race/ethnicity
   (l) `subsidy_fpl_bracket` - FPL bracket
   (m) `age` - Age
   (n) `service_channel` - Method used to enroll

(o) `language_spoken` - Language

(p) `subsidy_fpl_percent_int` - FPL

(q) `gross_premium_amt_int` - Gross (before subsidy) premium

(r) `net_premium_amt_int` - Net (after subsidy) premium

(s) `ENRLEE_PLAN_SELECT_DT_WK` - Week plan selected

(t) `cov_start_dt_WK` - Week coverage started

(u) `cov_end_dt_WK` - Week coverage ended

(v) `region` - Rating area

2. *Plan Data*

These data contain plan characteristics for each plan offered in Covered California during the 2014-2019 time frame. These files (one per year) were downloaded from https://hbex.coveredca.com/data-research/
(`20XX Product Prices for all Health Insurance Companies.xlsx`). These files are saved as `201X Plans.xlsx` in the *COVCAL Plan Data* subfolder in the *Data* folder. These files have inconsistent variable names, but most include the following:

(a) `Plan ID` - Plan identifier

(b) `Rating Area ID` - Rating area

(c) `Plan ID` - Plan identifier

(d) `Age` - Age

(e) `Individual Rate` - Premium

(f) `Applicant` - Insurer

(g) `HIOS ID` - Health Insurance Oversight System (HIOS) ID

(h) `Metal Level` - Metal tier

(i) `Network` - Plan network type

(j) `Product` - Coinsurance or copay plan

(k) `Index Name` - Name of plan

(l) `Full Plan Name` - Name of plan including metal tier

Because of inconsistent formatting in these raw data files, we merge these files in Excel to form the data file `ca_plan_data2.csv`. This file includes several additional fields that we created. Fields in this file include:

(a) `ENROLLMENT_YEAR` - Year

(b) `region` - Rating Area

(c) Rating_Area - Rating Area (exactly the same as region)

(d) Issuer_Name - Insurer

(e) HIOS_ID_10 - 10-digit HIOS

(f) HIOS - Full HIOS

(g) HIOSYR - Concatenation of HIOS and year

(h) PLAN_NETWORK_TYPE - Plan network type (e.g. HMO, PPO)

(i) metal_level - Metal tier

(j) Network_num - Network identifier (only relevant for plans sold by Sharp)

(k) Premium - Premium for a 40-year-old

(l) HSA - Indicator for whether plan is a health savings account plan

(m) MSP - Indictor for whether plan is a multi-state plan

(n) Plan_Name - Assigned plan name that groups plan across regions and years

(o) Plan_Name_NOCSR - Assigned plan name starting from Plan_Name that combines CSR plans

(p) Plan_Name2 - Assigned plan name starting from Plan_Name that combines PPO, EPO, and HSP plans

(q) Plan_Name2_NOCSR - Assigned plan name starting from Plan_Name2 that combines silver CSR plans

(r) Plan_Name_Small - Assigned plan name starting from Plan_Name2 that combines regional/small insurer plans

(s) Plan_Name_Small_NOCSR - Assigned plan name starting from Plan_Name_Small that combines silver CSR plans

(t) Plan_Name_Small_NOHSACAT - Assigned plan name starting from Plan_Name_Small_NOCSR that combines bronze plans, bronze HSA plans, and catastrophic plans (primary classification used in estimation and simulation)

(u) Base_Plan - Indicator for whether plan is the "base plan" for fixing plan premium ratios across regions by year (usually the bronze plan in rating area 15 unless insurer does not offer coverage in rating area 15)

(v) Pricing_Value - Ratio of plan premium to insurer's bronze plan premium in year and rating area

(w) Pricing_Factor - Ratio of plan premium to insurer's base plan premium in year

(x) Plan_ID_Order - Unique plan identifier that is used to reorder the plans in this .csv file

Several fields above are used to combine plans. For example, all regional/small insurer plans are combined by metal tier and Bronze, Bronze HSA, and Catastrophic plans are combined. The primary motivation for combining these plans is to save on computational costs. Most rating areas only have one regional/small insurer. Bronze and Bronze HSA plans usually have very similar premiums. Less than 1% of consumers choose catastrophic (largely because most consumers are not eligible to purchase catastrophic and catastrophic plans are not eligible for subsidies). Catastrophic plans generally are only slightly cheaper than bronze plans.

3. *Choice Set Data*

   These data indicate the 5-digit zip codes and counties where each plan was offered. For 2017-2019, three files (`20XX Products by Zip Code.xlsx`) were downloaded from https://hbex.coveredca.com/data-research/. These files are saved as `201X Choice Sets.xlsx` in the *COVCAL Choice Sets* subfolder in the *Data* folder. Raw data variables (names vary slightly across the three files) include:

   (a) `Covered California Rating Region` - Rating area
   (b) `County` - County
   (c) `FIPS County Code` - FIPS county code
   (d) `Zip Code` - Zip code
   (e) `Zip – County Code` - Concatenation of FIPS county code and zip code
   (f) `Zip Code Split Across Counties` - indicates if zip code belongs to two counties
   (g) `Anthem EPO` - indicates whether Anthem EPO plans are offered
   (h) `Anthem HMO` - indicates whether Anthem HMO plans are offered
   (i) `Anthem PPO` - indicates whether Anthem PPO plans are offered
   (j) `Blue Shield HMO` - indicates whether Blue Shield HMO plans are offered
   (k) `Blue Shield PPO` - indicates whether Blue Shield PPO plans are offered
   (l) `CCHP HMO` - indicates whether Chinese Community Health Plan is offered
   (m) `HealthNet of CAHMO` - indicates whether Health Net HMO plans are offered
   (n) `HealthNet of CAHSP` - indicates whether Health Net HSP plans are offered
   (o) `HealthNet Life EPO` - indicates whether Health Net EPO plans are offered
   (p) `HealthNet Life PPO` - indicates whether Health Net PPO plans are offered
   (q) `Kaiser Permanente` - indicates whether Kaiser plans are offered
   (r) `LA Care` - indicates whether LA Care plans are offered
   (s) `Molina` - indicates whether Molina plans are offered
   (t) `Oscar` - indicates whether Oscar plans are offered
   (u) `Sharp 1 Copay` - indicates whether Sharp copay plans are offered

(v) `Sharp 2 Coinsuance` - indicates whether Sharp coinsurance plans are offered

(w) `Valley Health Plan` - indicates whether Valley Health Plan is offered

(x) `Western Health Advantage` - indicates whether Western Health Advantage plans are offered

For 2014-2016, we obtained the choice set data through a PRA request by emailing pra@covered.ca.gov. Data for these years are saved in `2014 − 2016ChoiceSets.xlsx`, which is not included in the replication package. Raw data variables include:

(a) `RNAME` - Rating area

(b) `COUNTY_PSERVICE` - County

(c) `ZIP` - zip code

(d) `Anthem Blue Cross` - indicates whether Anthem plans are offered

(e) `Blue Shield` - indicates whether Blue Shield plans are offered

(f) `Chinese Community` - indicates whether Chinese Community Health plans are offered

(g) `Contra Costa Health Plan` - indicates whether Contra Costa plans are offered

(h) `Health Net − HMO` - indicates whether Health Net HMO plans are offered

(i) `Health Net − PPO` - indicates whether Health Net PPO plans are offered

(j) `Kaiser` - indicates whether Kaiser plans are offered

(k) `LA Care` - indicates whether LA Care plans are offered

(l) `Molina Health Care` - indicates whether Molina plans are offered

(m) `SHARP Health Plan` - indicates whether Sharp plans are offered

(n) `Valley Health` - indicates whether Valley plans are offered

(o) `Western Health` - indicates whether Western Health Advantage plans are offered

Because of unusual and highly inconsistent formatting in these raw data files, we merge these files in Excel to form the data file `CA Entry − Exit.csv`.

4. *Network data*

These data files include information on the insurers' provider networks. We obtained these data from Ideon using the instructions at the website https://ideonapi.com/researchers/. In the code, the data on network breadth and average overlap are loaded as `ca_network_data10.csv` and the data on pairwise inclusivity are loaded as `pairwise_inclus.csv`. The fields in `ca_network_data10.csv` are

(a) `insurer` - Insurer

(b) `network_type` - Plan network type (e.g., HMO, PPO)

(c) `network_num` - Network number (only relevant for Sharp plans)

(d) `hsa` - Indicator of HSA plan

(e) `year` - Year

(f) `metal` - Metal tier

(g) `zip3` - 3-digit zip code

(h) `unique_id2` - Unique plan ID

(i) `network_id` - Network ID

(j) `missing_net_ID` - Indicator of whether plan data is missing

(k) `total_enrollees` - Enrollees in plan (values 10 or less are coded as "< 11")

(l) `netbreadth_zip3` - Average network breadth measure

(m) `net_inclusivity_zip3` - Average inclusivity measure

The fields in `pairwise_inclus.csv` are

(a) `year` - Year

(b) `zip3` - 3-digit zip code

(c) `plan_a` - First plan of pair

(d) `plan_b` - Second plan of pair

(e) `inclusivity` - Pairwise inclusivity

5. *Age Rating Factor Data*

   This file contains the age rating factors published by CMS. The factors were downloaded from https://www.cms.gov/cciio/programs-and-initiatives/health-insurance-market-reforms/state-rating. Note that the age rating factors changed slightly in 2018. In the code, the data file is loaded as `age_rating_factors.csv`. Variables include:

   (a) `Age` - Age

   (b) `Rating_Factor` - Age rating factor for 2014-2017

   (c) `Rating_Factor2018` - Age rating factor 2018-Present

6. *Poverty Guideline Data*

   This file contains the federal poverty guidelines published by ASPE. The guidelines were downloaded from https://aspe.hhs.gov/topics/poverty-economic-mobility/poverty-guideli Each year the poverty guidelines are adjusted slightly. In the code, the data file is loaded as `poverty_guidelines.csv`. Variables include:

   (a) `Family_Size` - number of people in household

(b) `YR201X` - the federal poverty level in year 201X, where X ranges from 4 to 9

7. *ACA Contribution Percentage Data*

   This file shows the maximum required percentage contribution to the benchmark plan by income and by year. In the code, the data file is loaded as `contribution_percentages.csv`. The 2014 contribution percentages are in the text of the ACA: https://housedocs.house.gov/energycommerce/ppacacon.pdf. Each year, these percentages are slightly updated by the IRS. Links to IRS updates are provided below:

   - 2015: https://www.irs.gov/pub/irs-drop/rp-14-37.pdf
   - 2016: https://www.irs.gov/pub/irs-drop/rp-14-62.pdf
   - 2017: https://www.irs.gov/pub/irs-drop/rp-16-24.pdf
   - 2018: https://www.irs.gov/pub/irs-drop/rp-17-36.pdf
   - 2019: https://www.irs.gov/pub/irs-drop/rp-18-34.pdf

   Variables include:

   (a) `FPL` - Federal poverty level

   (b) `YR201X` - the required contribution percentage for the benchmark silver plan in year X, where X ranges from 4 to 9

8. *California Rating Area Definitions*

   This file maps California counties to rating areas. The data were downloaded from https://www.cms.gov/cciio/programs-and-initiatives/health-insurance-market-reforms/ca-gra. In the code, the data file is loaded as `ca_rating_areas.csv`. The variables are:

   (a) `County` - County

   (b) `Region` - Rating area

9. *Plan Definitions*

   This file defines each plan in *Choice Set Data*. It was prepared in Excel based on *Plan Data*. In the code, the data file is loaded as `product_definitions.csv`. Variables include:

   (a) `Product` - Concatenation of insurer, plan network type, MSP designation

   (b) `insurer` - Insurer

   (c) `plan_network_type` - Plan network type (e.g., PPO, HMO)

   (d) `MSP` - Multi-state plan

   (e) `Network` - Only relevant for Sharp, which has two different provider networks

10. *ACS Data*

These files includes single-year data from the U.S Census American Community Survey (ACS) for the years 2014-2019. It was downloaded from the Integrated Public Use Microdata Series (IPUMS) USA database from the website https://usa.ipums.org/usa/ (Ruggles et al., 2016). The ACS is used to help with constructing the outside option population. In the code, the raw data file is loaded as `usa_00011.dat.gz`. For the imputation of undocumented immigrants and affordable employer offers, we load a different raw data extract from IPUMS, `ipumsCA2.gz`. These files are not included in the replication package following IPUMS restrictions. Raw data variables include:

(a) `SERIALNO` - Household ID
(b) `SPORDER` - Person Number
(c) `PUMA` - PUMA (geographic identifier)
(d) `CIT` - Citizenship status
(e) `CITWP` - Year of naturalization
(f) `YOEP` - Year of entry
(g) `NATIVITY` - Nativity
(h) `COW` - Class of worker
(i) `HINS1` - ESI/group flag
(j) `HINS2` - Individual/nongroup flag
(k) `HINS3` - Medicare flag
(l) `HINS4` - Medicaid flag
(m) `HINS5` - TRICARE/military flag
(n) `HINS6` - VA flag
(o) `HINS7` - Indian Health Service flag
(p) `HICOV` - Health insurance coverage flag
(q) `PRIVCOV` - Private health insurance coverage flag
(r) `PUBCOV` - Public health insurance coverage flag
(s) `AGEP` - Age
(t) `PWGTP` - Person weight
(u) `SEX` - Gender
(v) `HISP` - Hispanic
(w) `PINCP` - Total personal income
(x) `POVPIP` - FPL

(y) RAC1P - Race

(z) RACAIAN - American Indian/Alaska Native flag

11. *SIPP Data*

    This file includes data from all available waves of the SIPP 2014. We use these data to impute "market eligibility" as described in the Online Appendix. The primary data files for each of the four waves were downloaded from the U.S. Census website https://www.census.gov/programs-surveys/sipp/data/datasets/2014-panel.html. The data are combined and loaded as Final_Workspace_v3.RData, but not included in this replication package following data restrictions. The data dictionary is available at the U.S. Census website above.

12. *SIPP Topical Module Data*

    These files include data from the SIPP 2008 topical modules 2 and 6. We use these data to impute undocumented immigrants and affordable employer offers. The core and topical module data for waves 2 and 6 were downloaded from the U.S. Census website https://www.census.gov/programs-surveys/sipp/data/datasets/2008-panel.html. In the code, the data file for topical module 2 loaded as p08putm2.zip. The core data for that wave 2 are loaded as l08puw2.zip. The data file for topical module 6 loaded as p08putm6.zip. The core data for that wave 6 are loaded as l08puw6.zip. These files are not included in this replication package following data restrictions. The data dictionary is available at the U.S. Census website above.

13. *PUMA Data*

    This file maps public use microdata areas (PUMAs) to counties. The "2010 PUMA Names File" was downloaded from https://www.census.gov/programs-surveys/geography/guidance/geo-areas/pumas.html. The region and simplified county fields were added manually to this file in Excel. In the code, the data file is loaded as ca_pumas.csv. Data variables include:

    (a) STATEFP - State FIPS Code

    (b) PUMA5CE - PUMA

    (c) RATING_AREA - Rating Area

    (d) COUNTY - County

    (e) PUMA Name - Description of geographic counties/cities in PUMA

14. *SAHIE Data*

    This file contains data on the uninsured from the U.S. Census Small Area Health Insurance Estimates (SAHIE) program. It is used to check the uninsured population estimates in the ACS at small geographic levels. The data for the years of our study were downloaded from https://www.census.gov/data/datasets/time-series/demo/sahie/estimates-acs.html. In the code, the data file is loaded as sahie.csv. Relevant data variables include:

(a) `year` - Year

(b) `statefips` - State FIPS code

(c) `countyfips` - County FIPS code

(d) `geocat` - 50 indicates county, 40 indicates state

(e) `agecat` - 0 (under 65), 1 (18 to 64), 2 (40 to 64), 3 (50 to 64), 4 (Under 19), 5 (21 to 64)

(f) `racecat` - 0 (all races), 1 (White alone), 2 (Black/African American alone), 3 (Hispanic), 4 (American Indian or Native American), 5 (Asian alone), 6 (Native Hawaiian or Pacific Islander), 7 (Two or more races)

(g) `sexcat` - 0 (both), 1 (male), 2 (female)

(h) `iprcat` - 0 (all), 1 (200% of FPL or below), 2 (250% of FPL or below), 3 (138% of FPL or below), 4 (400% of FPL or below), 5 (138% to 400% of FPL)

(i) `NIPR` - number in demographic group

(j) `NUI` - number uninsured

(k) `NIC` - number insured

(l) `PCTUI` - percent uninsured

(m) `PCTIC` - percent insured

(n) `PCTELIG` - percent uninsured in demographic group for all income levels

(o) `PCTLIIC` - percent insured in demographic group for all income levels

15. *MLR Data*

These data come from 2014-2018 CMS Medical Loss Ratio (MLR) Reports. They provide financial information for each insurer, including data used to calculate fixed costs. The zip file data were downloaded from the CMS website https://www.cms.gov/marketplace/resources/data/medical-loss-ratio-data-systems-resources. The data are saved in one subfolder for each year (`201X MLR Data`) within the `Data` folder. We call multiple files within each subfolder in the code. The zip file downloads include a data dictionary.

16. *Risk Adjustment-Reinsurance Data*

These data in the RA-RI-RC subfolder in the Data folder summarize insurer-state level risk adjustment and reinsurance payments. The data were downloaded from the CMS website https://www.cms.gov/marketplace/health-plans-issuers/premium-stabilization-programs Specifically, we downloaded the PDFs titled "Summary Report on Transitional Reinsurance Payments and Permanent Risk Adjustment Transfers for the 201X Benefit Year (PDF)" and the Appendix B Excel files. We combined the relevant tables for risk adjustment and reinsurance in the PDF documents across the six years in the file `ra_reins.csv`. Data variables include:

(a) `HIOS` - HIOS identifier

(b) `Insurer` - Insurer

(c) `STATE` - State

(d) `Reinsurance` - Reinsurance received (dollars)

(e) `RA_IND` - Risk Adjustment received in individual market (dollars)

(f) `RA_SMALL` - Risk Adjustment received in small group market (dollars)

(g) `Year` - Year

We also use Appendix B of these reports to obtain data on the geographic cost factors that CMS uses. Data variables include:

(a) `Market` - Individual or small group

(b) `State` - State

(c) `Exchange Rating Area` - Rating area

(d) `GCF` - Geographic cost factor

(e) `Billable Member Months` - Member months

17. *Supplemental Rate Review Template (SRRT) Data*

These provide supplemental information to the standard CMS rate filings. Covered California insurers must submit various information on the SRRT forms that are not included in the standard unified rate review template (URRT) that CMS requires (see CMS Rate Filing Data below). We use information on plan risk scores from the SRRT Data. We downloaded these data from California's Department of Managed Health Care at the website `https://wpso.dmhc.ca.gov/premiumratereview/FilingList.aspx`. The data are not included in this replication package following data restrictions. While the layout of these files is somewhat inconsistent, we generally use data from the "Risk Scores" tab. This tab continues two tables:

(a) Member months, by metal level and rating area

(b) Risk scores, by metal level andd rating area

18. *CMS Rate Filing Data*

These data come from the CMS unified rate review template (URRT) that every exchange insurer must complete. We downloaded these zip files (Worksheet I and II Data for 201X Single Risk Pool Filings from the CMS website `https://www.cms.gov/CCIIO/Resources/Data-Resources/ratereview`. These zip files are saved in the `Rate Filings` subfolder of the `Data` folder. The zip file downloads include a data dictionary. Files for 2014 through 2020 were merged into the file `2014-2020.RData`.

3 Code

All code is written in the R programming language or Julia programming language. Most data cleaning and output processing scripts are in R and most estimation and simulation scripts are in Julia. The subsections below summarize each script used in the paper. All R code files have the suffix .R and all Julia code files have the suffix .jl.

3.1 Cleaning/Processing

3.1.1 Demand-Side

1. make.choice.set.file.R

   This file reads in CA Entry-Exit.csv (which is at the 5-digit zip code level). It outputs two files:

   (a) *Processed Choice Set Data*: Choice sets for all but 34 of the 666 3-digit zip code/rating area combinations. The output is saved in choice_sets.csv. The 34 missing combinations occur where a zip barely overlaps a rating area. These are added manually in Excel to create zip3_choices2.csv.

   (b) *California County Insurer Participation Files*: Choice sets by insurer, county, and year. One output file is saved for each year (2014-2019) as ca_counties_YYYY.csv.

2. process.COVCAL.data.R

   The *Enrollment Data* is cleaned and processed in process.COVCAL.data.R. Inputs include:

   (a) *Enrollment Data* (pra_07192019.csv)

   (b) *Processed Plan Data* (ca_plan_data2.csv)

   (c) *Processed Choice Set Data* (zip3_choices2.csv)

   (d) *Age Rating Factor Data* (age_rating_factors.csv)

   (e) *Poverty Guideline Data* (poverty_guidelines.csv)

   (f) *ACA Contribution Percentage Data* (contribution_percentages.csv)

   (g) *California Rating Area Definitions* (ca_rating_areas.csv)

   (h) *Plan Definitions* (product_definitions.csv)

   Outputs include:

   (a) ca_enrollment_data_AUG012019
   This R data file contains cleaned enrollment data at the individual level.

   (b) ca_household_data_AUG012019
   This R data file contains cleaned enrollment data at the household level.

3. `process.sipp.R`

   This code is used to impute undocumented immigrants and survey respondents with an affordable employer offer in the ACS population. Inputs include:

   (a) *SIPP Topical Module Data* (several files, see above)

   (b) *ACS Data* (several files, see above)

   Outputs include:

   (a) `ca_acs_immigration_OCT012018`
   This `R` data file contains the undocumented immigrant imputation.

   (b) `ca_acs_emp_offer_OCT012018`
   This `R` data file contains the affordable employer offer imputation.

4. `impute.with.SIPP.R`

   This code imputes "market eligibility" using the *SIPP data* (`Final_Workspace_v3.RData`). The file outputs the logit parameters in the `R` data file (`sipp_logit`).

5. `process.ACS.data.R`

   This code creates the outside option (uninsured and exchange-eligible) population and merges it with the exchange population. These are individuals that meet our definition of exchange-eligible, but choose not to purchase insurance. Inputs include:

   (a) *Processed Individual Exchange Enrollment Data* (`ca_enrollment_data_AUG012019`)

   (b) *Processed Household Exchange Enrollment Data* (`ca_household_data_AUG012019`)

   (c) *ACS Data* (several files, see above)

   (d) *Processed Plan Data* (`ca_plan_data2.csv`)

   (e) *Processed Choice Set Data* (`zip3_choices2.csv`)

   (f) *Age Rating Factor Data* (`age_rating_factors.csv`)

   (g) *Poverty Guideline Data* (`poverty_guidelines.csv`)

   (h) *ACA Contribution Percentage Data* (`contribution_percentages.csv`)

   (i) *California County Insurer Participation Files* (`ca_counties_YYYY.csv`)

   (j) *Plan Definitions* (`product_definitions.csv`)

   (k) *PUMA Data* (`ca_pumas.csv`)

   (l) *SAHIE Data* (`sahie.csv`)

   (m) *Affordable Employer Offer Imputation* (`ca_acs_emp_offer_OCT012018`)

   (n) *Undocumented Immigrant Imputation* (`ca_acs_immigration_OCT012018`)

Outputs include:

(a) `ca_enrollment_data_AUG022019`
This `R` data file contains cleaned enrollment data at the individual level.

(b) `ca_household_data_AUG022019`
This `R` data file contains cleaned enrollment data at the household level.

6. `make.data.objects.R`

This code (along with `make.julia.R`) creates the data objects that are needed to perform estimation and simulation in `Julia`. Inputs include:

(a) *Processed Individual Enrollment Data* (`ca_enrollment_data_AUG022019`)

(b) *Processed Household Enrollment Data* (`ca_household_data_AUG022019`)

(c) *Processed Plan Data* (`ca_plan_data2.csv`)

(d) *Processed Choice Set Data* (`zip3_choices2.csv`)

(e) *Age Rating Factor Data* (`age_rating_factors.csv`)

(f) *Poverty Guideline Data* (`poverty_guidelines.csv`)

(g) *ACA Contribution Percentage Data* (`contribution_percentages.csv`)

(h) *Plan Definitions* (`product_definitions.csv`)

(i) *California Rating Area Definitions* (`ca_rating_areas.csv`)

(j) *Market Eligibility Imputation* (`sipp_logit`)

Outputs include:

(a) `ca_premium_matrix_AUG032019_small`
Premiums for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

(b) `ca_choice_matrix_AUG032019_small`
An indicator matrix for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for each household's chosen plan, 0 for plan's not chosen by the household, and NA for plans not in household's choice set.

(c) `ca_new_choice_matrix_AUG032019_small`
An indicator matrix for every **new** plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for every new plan in each household's choice set, 0 for existing plans, and NA for plans not in household's choice set.

(d) `ca_previous_choice_matrix_AUG032019_small`

An indicator matrix for the previously chosen plan. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for the plan chosen in the previous year, 0 for all other plans in the household's choice set, and NA for plans not in household's choice set. For a household new to the exchange, all plans will be 0.

(e) `ca_subsidy_matrix_AUG032019_small`

Subsidies for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

(f) `ca_AV_matrix_AUG032019_small`

Actuarial value (AV) for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The AV for the same silver plan can vary by household due to CSRs.

(g) `ca_household_characteristics_AUG032019_small`
`ca_household_characteristics_AUG032019_small.csv`

Household characteristics that will be used for estimation.

(h) `ca_plan_characteristics_AUG032019_small`
`ca_plan_characteristics_AUG032019_small.csv`

Plan characteristics that will be used for estimation.

(i) `years_in_panel_1418.csv`

Indicates how many years each household appears in our panel between 2014 and 2018.

7. `make.julia.R` - This code (along with `make.data.objects.R`) creates the data objects that are needed to perform estimation and simulation in `Julia`. Specifically, this code combines premiums, choices, previous choices, new plans, subsidies, and AVs into a single data object to send to `Julia`. Inputs include:

(a) *Premium Object* (`ca_premium_matrix_AUG032019_small`)

(b) *Choice Object* (`ca_choice_matrix_AUG032019_small`)

(c) *New Plan Object* (`ca_new_choice_matrix_AUG032019_small`)

(d) *Previous Choice Object* (`ca_previous_choice_matrix_AUG032019_small`)

(e) *Subsidy Object* (`ca_subsidy_matrix_AUG032019_small`)

(f) *AV Object* (`ca_AV_matrix_AUG032019_small`)

(g) *Household Characteristics* (`ca_household_characteristics_AUG032019_small`)

Outputs include:

(a) `ca_julia_data_AUG032019_small.csv`

This object includes premiums, choices, previous choices, new plans, subsidies, and AVs to read into `Julia`. This is the primary data object for demand estimation.

(b) `unique_choice_sets_MAY112021.csv`

This file contains the unique choice sets by geography. This helps facilitate faster code in `Julia`.

8. `make.data.objects.churn.R`

This code (along with `make.julia.churn.R`) creates the data objects that are needed to perform estimation and simulation in `Julia` for the churn simulations. Inputs are the same as for `make.data.objects..R`. Outputs include:

(a) `ca_premium_matrix_AUG032019_churn`

Premiums for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

(b) `ca_choice_matrix_AUG032019_churn`

An indicator matrix for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for each household's chosen plan, 0 for plan's not chosen by the household, and NA for plans not in household's choice set.

(c) `ca_new_choice_matrix_AUG032019_churn`

An indicator matrix for every **new** plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for every new plan in each household's choice set, 0 for existing plans, and NA for plans not in household's choice set.

(d) `ca_previous_choice_matrix_AUG032019_churn`

An indicator matrix for the previously chosen plan. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for the plan chosen in the previous year, 0 for all other plans in the household's choice set, and NA for plans not in household's choice set. For a household new to the exchange, all plans will be 0.

(e) `ca_subsidy_matrix_AUG032019_churn`

Subsidies for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

(f) `ca_AV_matrix_AUG032019_churn`

Actuarial value (AV) for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The AV for the same silver plan can vary by household due to CSRs.

(g) `ca_household_characteristics_AUG032019_churn`
`ca_household_characteristics_AUG032019_churn.csv`
Household characteristics that will be used for estimation.

9. `make.julia.churn.R` - This code (along with `make.data.objects.churn.R`) creates the data objects that are needed to perform estimation and simulation in `Julia`. Specifically, this code combines premiums, choices, previous choices, new plans, subsidies, and AVs into a single data object to send to `Julia`. Inputs include:

   (a) *Premium Object* (`ca_premium_matrix_AUG032019_churn`)

   (b) *Choice Object* (`ca_choice_matrix_AUG032019_churn`)

   (c) *New Plan Object* (`ca_new_choice_matrix_AUG032019_churn`)

   (d) *Previous Choice Object* (`ca_previous_choice_matrix_AUG032019_churn`)

   (e) *Subsidy Object* (`ca_subsidy_matrix_AUG032019_churn`)

   (f) *AV Object* (`ca_AV_matrix_AUG032019_churn`)

   (g) *Household Characteristics* (`ca_household_characteristics_AUG032019_churn`)

   Outputs include:

   (a) `ca_julia_data_AUG032019_churn.csv`
   This object includes premiums, choices, previous choices, new plans, subsidies, and AVs to read into `Julia`. This is the primary data object for demand estimation.

   (b) `unique_choice_sets_MAY112021_churn.csv`
   This file contains the unique choice sets by geography. This helps facilitate faster code in `Julia`.

10. `make.data.objects.net.R`

    This code (along with `make.julia.R`) creates the data objects that are needed to perform estimation and simulation in `Julia` for the runs where we include the network variables. Inputs include all of those used in `make.data.objects.R` plus the following:

    (a) *Network Breadth Data* (`ca_network_data10.csv`)

    (b) *Pairwise Inclusivity Data* (`pairwise_inclus.csv`)

    Outputs include:

    (a) `ca_premium_matrix_AUG032019_net`
    Premiums for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

(b) `ca_choice_matrix_AUG032019_net`

An indicator matrix for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for each household's chosen plan, 0 for plan's not chosen by the household, and NA for plans not in household's choice set.

(c) `ca_new_choice_matrix_AUG032019_net`

An indicator matrix for every **new** plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for every new plan in each household's choice set, 0 for existing plans, and NA for plans not in household's choice set.

(d) `ca_previous_choice_matrix_AUG032019_net`

An indicator matrix for the previously chosen plan. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The value is 1 for the plan chosen in the previous year, 0 for all other plans in the household's choice set, and NA for plans not in household's choice set. For a household new to the exchange, all plans will be 0.

(e) `ca_subsidy_matrix_AUG032019_net`

Subsidies for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

(f) `ca_AV_matrix_AUG032019_net`

Actuarial value (AV) for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans. The AV for the same silver plan can vary by household due to CSRs.

(g) `ca_household_characteristics_JUN182021_net`
`ca_household_characteristics_JUN182021_net.csv`

Household characteristics that will be used for estimation.

(h) `ca_plan_characteristics_JUN182021_net`
`ca_plan_characteristics_JUN182021_net.csv`

Plan characteristics that will be used for estimation.

(i) `ca_network_breadth_matrix_JUN182021_net`

Network breadth for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

(j) `ca_network_inclus_matrix_JUN182021_net`

Average overlap for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

(k) `ca_network_pairinclus_matrix_JUN182021_net`

Pairwise inclusivity for every plan in every household's choice set. It is saved as a matrix of dimension $I \times J$, where I is the number of households and J is the number of plans.

11. `make.julia.net.R` - This code (along with `make.data.objects.net.R`) creates the data objects that are needed to perform estimation and simulation in `Julia` for the runs where we include the network variables. Specifically, this code combines premiums, choices, previous choices, new plans, subsidies, AVs, and the network variables into a single data object to send to `Julia`. Inputs include:

    (a) *Premium Object* (`ca_premium_matrix_AUG032019_net`)

    (b) *Choice Object* (`ca_choice_matrix_AUG032019_net`)

    (c) *New Plan Object* (`ca_new_choice_matrix_AUG032019_net`)

    (d) *Previous Choice Object* (`ca_previous_choice_matrix_AUG032019_net`)

    (e) *Subsidy Object* (`ca_subsidy_matrix_AUG032019_net`)

    (f) *AV Object* (`ca_AV_matrix_AUG032019_net`)

    (g) *Network Breadth Object* (`ca_network_breadth_matrix_JUN182021_net`)

    (h) *Average Overlap Object* (`ca_network_inclus_matrix_JUN182021_net`)

    (i) *Pairwise Inclusivity* (`ca_network_pairinclus_matrix_JUN182021_net`)

    (j) *Household Characteristics* (`ca_household_characteristics_JUN182021_net`)

    Outputs include:

    (a) `ca_julia_data_JUN182021_net.csv`
    This object includes premiums, choices, previous choices, new plans, subsidies, and AVs to read into `Julia`. This is the primary data object for demand estimation.

    (b) `unique_choice_sets_JUN182021_net.csv`
    This file contains the unique choice sets by geography. This helps facilitate faster code in `Julia`.

3.1.2 Supply-Side

1. `process.MLR.data.R`

    This file processes 2014-2018 data from the medical loss ratio (MLR) reports. Inputs include:

    (a) *MLR Data* (suite of files, see above)

    (b) *Risk Adjustment-Reinsurance Data* (`ra_reins.csv`)

    Outputs include:

    (a) `mlr_data.csv`
    Processed MLR data for the California insurers during the study time frame.

2. `process.risk.scores.R`

   This code: (1) processes the risk score data from the *SRRT Data* and (2) processes the geographic cost factor data from the *Risk Adjustment-Reinsurance Data*. Inputs include:

   (a) *SRRT Data* (suite of files, see above)

   (b) *Risk Adjustment/Reinsurance Data* (`201X_App_B.xlsx`)

   Outputs include:

   (a) `risk_score_data.csv`
   Processed risk score data, saved for later usage in other code scripts.

   (b) `ca_cms_geographic_factors.csv`
   Geographic cost factors, saved for later usage in other code scripts.

   (c) `gcf_table.csv`
   A table of geographic cost factors, formatted for the paper.

3. `process.rate.data.R`

   This code processes the CMS rate filing data and prepares the risk score/claims moments for the `Julia` supply estimation code. Note that this code has to be run after demand estimation. There are flag variables at the beginning of the code that should be modified depending on the demand specification. Inputs include:

   (a) *CMS Rate Filing Data* ($2014 - 2020$.`RData`)

   (b) *Processed Demand Data for Julia*
   (`ca_julia_data_AUG032019_small.csv` or
   `ca_julia_data_JUN182021_net.csv`)

   (c) *Processed Household Characteristics for Julia*
   (`ca_household_characteristics_AUG032019_small`
   or `ca_household_characteristics_JUN182021_net`)

   (d) *MLR Data* (`mlr_data.csv`)

   (e) *Plan Data* (`ca_plan_data2.csv`)

   (f) *Initial risk score parameter estimates* (`rs_coefficients_XXXX.csv`)

   (g) *Initial claims parameter estimates* (`claims_first_stage_coefficients_XXXX.csv`)

   (h) *Estimated Household Choice Probabilities* (depends on demand specification)

   Outputs include:

   (a) `moments_data_XXXX.csv`
   Data object used to construct risk score and claims moments in `Julia` scripts, where $XXXX$ refers to the demand and supply specification.

(b) `convergence_analysis_updated.csv`
Data object with basic financial information by insurer and year.

4. `process.supply.data.R`

This code prepares the final files needed for estimating the supply-side parameters. Note that this code has flag variables at the beginning of the code that should be modified depending on the demand specification. Inputs include:

(a) *Processed Demand Data for Julia*
(`ca_julia_data_AUG032019_small.csv` or
`ca_julia_data_JUN182021_net.csv`)

(b) *Processed Household Characteristics for Julia*
(`ca_household_characteristics_AUG032019_small`
or `ca_household_characteristics_JUN182021_net`)

(c) *MLR Data* (`mlr_data.csv`)

(d) *Rate Filing Data* (`moments_data_XXXX.csv`)

(e) *Basic financial info* (`convergence_analysis_updated.csv`)

(f) *Processed Plan Characteristics for Julia*
(`ca_plan_characteristics_AUG032019_small.csv` or
`ca_plan_characteristics_JUN182021_net.csv`)

(g) *Processed Plan Data* (`ca_plan_data2.csv`)

(h) *Geographic Cost Factors* (`ca_cms_geographic_factors.csv`)

(i) *Age Rating Factor Data* (`age_rating_factors.csv`)

Outputs include:

(a) `ca_julia_supply_data_JUN262020_learn.csv`
`ca_julia_supply_data_JUN262020_churn.csv`
`ca_julia_supply_data_JUN182021_net.csv`
This is the main household-level data object for the supply-side estimation, as well as for simulation.

(b) `ca_plan_year_JUN262020_learn.csv`
`ca_plan_year_JUN262020_churn.csv`
`ca_plan_year_JUN182021_net.csv`
This data object includes all plan-year level variables required for supply-side estimation, as well as for simulation.

(c) `ca_plan_market_year_JUN262020_learn.csv`
`ca_plan_market_year_JUN262020_churn.csv`
`ca_plan_market_year_JUN182021_net.csv`
This data object includes all plan-market-year level variables required for supply-side estimation, as well as for simulation.

3.2 Estimation

3.2.1 Demand

1. `perform.reduced.form.R`

   This file runs the first stage of the control function approach. Note that this code has flag variables at the beginning of the code that should be modified depending on the demand specification. Inputs include:

   (a) *Processed Plan Characteristics for Julia* (`ca_plan_characteristics_AUG032019_small.csv` or `ca_plan_characteristics_JUN182021_net.csv`)

   (b) *Geographic Cost Factors* (`ca_cms_geographic_factors.csv`)

   (c) *Processed Demand Data for Julia* (`ca_julia_data_AUG032019_small.csv` or `ca_julia_data_AUG032019_churn.csv` or `ca_julia_data_JUN182021_net.csv`)

   Outputs include:

   (a) (`ca_julia_data_AUG032019_small.csv` or `ca_julia_data_AUG032019_churn.csv` or `ca_julia_data_JUN182021_net.csv`)

   Processed demand data for `Julia` with residual variable added to implement control function approach

   (b) `reduced_form_results.csv`

   Results from the first stage for paper.

2. `estimate.demand.inertia.jl`

   Main script that performs demand estimation. Note that this code has flag variables at the beginning of the code that should be modified depending on the demand specification. These flag variables include:

   (a) `cluster_run` - indicates whether to run the script on the high performance computing cluster. In general, this code will not be able to run on a local machine.

   (b) `use_newton` - indicates whether to use Newton's method or Nelder-Mead. The former is much faster, but often does not converge unless the starting solution is reasonably close to the optimal solution.

   (c) `eq_robust_flag` - indicates whether to only use the last three years of data for estimation

   (d) `network_flag` - indicates whether to include the network variables in estimation

(e) `control_function` - indicates whether to use the control function approach in estimation

(f) `random_parameters` - indicates whether to include random parameters in estimation

(g) `sequence_flag` - indicates whether to the maximize the likelihood function as a sequence of choices

(h) `inattention` - indicates whether to use the inattention specification

(i) `nested` - indicates whether to use a logit or nested logit

(j) `add_premium_interactions` - indicates whether to interact the premium variable with demographic variables

(k) `add_previous_demographics_interactions` - indicates whether to interact previous choice variable with demographic variables

(l) `add_previous_firm_interactions` - indicates whether to interact previous choice variable with plan characteristics, including brand names

(m) `add_av_interactions` - indicates whether to interact the actuarial value variable with demographic variables

(n) `add_intercepts` - indicates whether to add demographic intercepts

(o) `add_rating_areas` - indicates whether to add rating area (market) fixed effects

(p) `add_insurer_market_fe` - indicates whether to add insurer-market fixed effects

(q) `add_complex_interactions` - indicates whether to add interactions between (1) premium, income, and age; (1) previous choice, income, and age; (3) AV, income, and age and (4) plan type and age.

(r) `shrink_sample` - indicates whether to perform estimation on a 1% sample of the data (can usually be run on a local machine)

(s) `do_estimation_flag` - indicates whether to perform estimation (if not, the output files are generated assuming the starting solution is optimal)

There are additional flag variables in the code that are no longer actively used. These should not be adjusted. Inputs to this script include:

(a) *Starting solution* (`starting_point.csv`)

This file contains the starting solution for demand estimation. There needs to be a covariate column with the header `Variable` and a column of values with the header `Coef`. The code uses both the specified covariates and corresponding values to build the data objects and initialize the optimization routine. The list of covariates must correspond to the specification specified at the top of `estimate.demand.inertia.jl`.

(b) *Setup script* (`setup.demand.data.jl`)

This script loads all other data, as detailed below.

(c) *Demand function script* (`demand.functions.inertia.jl`)
This script contains support functions that are called in `estimate.demand.inertia.jl`.

(d) *Elasticity function script* (`elasticity.functions.CA.jl`)
This script contains the functions used to calculate elasticities and switching costs that are called in `estimate.demand.inertia.jl`.

Outputs include:

(a) *Parameter output* (filenames vary depending on specification)
This file contains the demand estimates, including the point estimates, standard errors, and p-values.

(b) *Household choice probabilities* (filenames vary depending on specification)
This file contains the estimated choice probabilities for every household-plan combination.

(c) *Elasticity table* (filenames vary depending on specification)
This file contains the estimated elasticities and semi-elasticities (both own and exchange coverage) implied by the parameter estimates.

(d) *Switching cost table* (filenames vary depending on specification)
This file contains the estimated annual switching costs implied by the parameter estimates.

(e) *Average parameter output* (filenames vary depending on specification)
This file saves the average premium and inertia parameters (averaged across all households).

3. `setup.demand.data.jl`

This script is called from `estimate.demand.inertia.jl`. It loads packages and data, initializes variables, and includes functions to form the main covariate objects. Inputs include:

(a) *Processed Demand Data for Julia*
(`ca_julia_data_AUG032019_small.csv` or `ca_julia_data_JUN182021_net.csv`)

(b) *Processed Household Characteristics for Julia*
(`ca_household_characteristics_AUG032019_small` or `ca_household_characteristics_JUN182021_net`)

(c) *Processed Plan Characteristics for Julia*
(`ca_plan_characteristics_AUG032019_small.csv` or `ca_plan_characteristics_JUN182021_net.csv`)

There are no outputs from this script.

4. `demand.functions.inertia.jl`

   Demand-side functions that are called in many scripts, mostly notably the main demand estimation script `estimate.demand.inertia.jl`. There are no input or output files.

5. `elasticity.functions.CA.jl`

   Functions to compute elasticities and switching costs. This script is loaded from the main demand estimation script `estimate.demand.inertia.jl`. There are no input or output files.

3.2.2 Supply

1. `estimate.risk.scores.R`

   This code provides initial estimates of the supply-side parameters (without considering the equilibrium conditions) using the estimated demand parameters. Note that this code has flag variables at the beginning of the code that should be modified depending on the demand specification. Inputs include:

   (a) *Processed Demand Data for Julia* (`ca_julia_data_AUG032019_small.csv` or `ca_julia_data_JUN182021_net.csv`)

   (b) *Processed Household Characteristics for Julia* (`ca_household_characteristics_AUG032019_small` or `ca_household_characteristics_JUN182021_net`)

   (c) *Processed Plan Characteristics for Julia* (`ca_plan_characteristics_AUG032019_small.csv` or `ca_plan_characteristics_JUN182021_net.csv`)

   (d) *Processed Household Enrollment Data* (`ca_household_data_AUG022019`)

   (e) *Processed Plan Data* (`ca_plan_data2.csv`)

   (f) *Processed Risk Score Data* (`risk_score_data.csv`)

   (g) *Market Eligibility Imputation* (`sipp_logit`)

   (h) *Estimated Household Choice Probabilities* (depends on demand specification)

   Outputs include:

   (a) `rs_coefficients_XXXX.csv`

   Initial risk score parameter estimates, where XXXX is specific to demand and supply specification.

   (b) `claims_first_stage_coefficients_XXXX.csv`

   Initial claims parameter estimates, where XXXX is specific to demand and supply specification.

2. `estimate.supply.jl`

   Main script that performs estimation of supply-side parameters. Note that this code has flag variables at the beginning of the code that should be modified depending on the specification. These flag variables include:

   (a) `eq_robust_flag` - indicates whether to only use the last three years of data for estimation

   (b) `network_flag` - indicates whether to include the network variables in estimation

   (c) `control_function` - indicates whether to use the control function approach in estimation

   (d) `random_parameters` - indicates whether to include random parameters in estimation

   (e) `choice_seq_flag` - indicates whether to the maximize the likelihood function as a sequence of choices

   (f) `nested` - indicates whether to use a logit or nested logit

   (g) `add_av_interactions` - indicates whether to interact the actuarial value variable with demographic variables

   (h) `add_insurer_market_fe` - indicates whether to add insurer-market fixed effects

   (i) `add_complex_interactions` - indicates whether to add interactions between (1) premium, income, and age; (1) previous choice, income, and age; (3) AV, income, and age and (4) plan type and age.

   (j) `int_rs_variables` - indicates whether to use the streamlined risk score regression specification

   (k) `all_rs_variables` - indicates whether to use the full/rich risk score regression specification

   There are additional flag variables in the code that are no longer actively used. These should not be adjusted. Inputs include:

   (a) `load.supply.estimation.data.jl`
   Script that loads all data for supply estimation, as detailed below.

   (b) `estimation.functions.jl`
   Script containing support functions that are called in `estimate.supply.jl`.

   Outputs include:

   (a) *Supply-Side Parameter Estimates* (filename varies depending on specification)

3. `load.supply.estimation.data.jl`

   This script is called from `estimate.supply.jl`. It loads packages and data, as well as additional functions in `estimation.functions.jl`. Inputs include:

(a) *Processed Supply Data for Julia*
`ca_julia_supply_data_JUN262020_learn.csv`
`ca_julia_supply_data_JUN262020_churn.csv`
`ca_julia_supply_data_JUN182021_net.csv`

(b) *Processed Household Characteristics for Julia*
(`ca_household_characteristics_AUG032019_small` or
`ca_household_characteristics_JUN182021_net`)

(c) *Unique Choice Set Data*
(`unique_choice_sets_MAY112021.csv`
`unique_choice_sets_MAY112021_churn.csv` or
`unique_choice_sets_JUN182021_net.csv`)

(d) *Processed Plan-Year Characteristics for Julia*
`ca_plan_year_JUN262020_learn.csv`
`ca_plan_year_JUN262020_churn.csv`
`ca_plan_year_JUN182021_net.csv`

(e) *Processed Plan-Market-Year Characteristics for Julia*
`ca_plan_market_year_JUN262020_learn.csv`
`ca_plan_market_year_JUN262020_churn.csv`
`ca_plan_market_year_JUN182021_net.csv`

(f) *Demand Parameter Estimates*
Estimates were aggregated in `demand_parameter_estimates.csv` using Excel.

(g) *Data for Moment Conditions*
(`moments_data_XXXX.csv`)

(h) *Initial risk score parameter estimates* (`rs_coefficients_XXXX.csv`)

(i) *Initial claims parameter estimates* (`claims_first_stage_coefficients_XXXX.csv`)

4. `estimation.functions.jl`

   Supply-side functions that are called in many scripts, most notably for estimating the supply-side parameters. There are no input or output files.

3.3 Simulation

1. `run.counterfactuals.jl`

   This is the main code for running counterfactuals. Note that this code has flag variables at the beginning of the code to define the scenario and specification used. These variables include:

   (a) `year` - specifies year to run. This should be between 2014 and 2018.

   (b) `scenario` - this references all scenario flag variables in `scenario_definitions_inertia.csv`. It must reference a scenario defined in `scenario_definitions_inertia.csv`.

(c) `cluster_run` - indicates whether to run the script on the high performance computing cluster. This code will generally run on a local machine, but there may be limited capacity for conducting parallel runs.

(d) `control_function` - indicates whether to use the control function approach in estimation

(e) `random_parameters` - indicates whether to include random parameters in estimation

(f) `choice_seq_flag` - indicates whether to the maximize the likelihood function as a sequence of choices

(g) `nested` - indicates whether to use a logit or nested logit

(h) `add_av_interactions` - indicates whether to interact the actuarial value variable with demographic variables

(i) `add_insurer_market_fe` - indicates whether to add insurer-market fixed effects

(j) `add_complex_interactions` - indicates whether to add interactions between (1) premium, income, and age; (1) previous choice, income, and age; (3) AV, income, and age and (4) plan type and age.

(k) `int_rs_variables` - indicates whether to use the streamlined risk score regression specification

(l) `all_rs_variables` - indicates whether to use the full/rich risk score regression specification

(m) `run_counterfactuals_flag` - indicates whether to run counterfactual or calculate output metrics based on the initial solution.

(n) `process_output_flag` - indicates whether to process output based on equilibrium vector of premiums

There are additional flag variables in the code that are no longer actively used. These should not be adjusted. Inputs include:

(a) *Initial Solution*

The default is the observed vector of base premiums, but these can be adjusted.

(b) `setup.counterfactual.data.jl`

Script that loads all data for counterfactuals, as detailed below.

(c) `counterfactual.functions.jl`

Script containing support functions that are called in `run.counterfactuals.jl`

(d) `create.supply.output.jl`

Script that takes new equilibrium premiums and calculates various outcome measures, including social welfare.

Outputs include:

(a) *New equilibrium premiums* (filename varies depending on scenario)

(b) *Scenario output measures* (filename varies depending on scenario)

2. `setup.counterfactual.data.jl`

   This file loads package and data for running the counterfactuals. It also creates the necessary data objects. Inputs include:

   (a) Scenario Definitions (`scenario_definitions_inertia.csv`)
   This CSV file defines every counterfactual scenario as a list of flag variables. Every scenario specified in `run.counterfactuals.jl` must refer to a scenario defined in this file.

   (b) Counterfactual Functions (`counterfactual.functions.jl`)

   (c) *Processed Supply Data for Julia*
   `ca_julia_supply_data_JUN262020_learn.csv`
   `ca_julia_supply_data_JUN262020_churn.csv`
   `ca_julia_supply_data_JUN182021_net.csv`

   (d) *Processed Household Characteristics for Julia*
   (`ca_household_characteristics_AUG032019_small` or
   `ca_household_characteristics_JUN182021_net`)

   (e) *Unique Choice Set Data*
   (`unique_choice_sets_MAY112021.csv`
   `unique_choice_sets_MAY112021_churn.csv` or
   `unique_choice_sets_JUN182021_net.csv`)
   This file contains the unique choice sets by geography. This helps facilitate faster code in `Julia`.

   (f) *Processed Plan-Year Characteristics for Julia*
   `ca_plan_year_JUN262020_learn.csv`
   `ca_plan_year_JUN262020_churn.csv`
   `ca_plan_year_JUN182021_net.csv`

   (g) *Processed Plan-Market-Year Characteristics for Julia*
   `ca_plan_market_year_JUN262020_learn.csv`
   `ca_plan_market_year_JUN262020_churn.csv`
   `ca_plan_market_year_JUN182021_net.csv`

   (h) *Demand Parameter Estimates*
   Estimates were aggregated in `demand_parameter_estimates.csv` using Excel.

   (i) *Risk score parameter estimates* (`rs_coefficients_XXXX.csv`)

   (j) *Claims parameter estimates* (`claims_estimates_XXXX.csv`)
   This file is from `estimate.supply.jl`.

There are no outputs for this script.

3. `counterfactual.functions.jl`

   Functions used for running counterfactuals. Loaded in `setup.counterfactual.data.jl`. There are no input or output files.

4. `create.supply.output.jl`

   This script reads in the final equilibrium vector of base premiums from `run.counterfactuals.jl` and computes various output metrics, including social welfare. The beginning of the script contains a series of support functions. The output metrics to be computed are listed in a vector called `output_fields` following the last support function. Inputs for the script include:

   (a) *Equilibrium premiums* (`base_premium`)
   This is the final vector of equilibrium premiums from running the counterfactual in `run.counterfactuals.jl`.

   (b) *Base scenario premiums* (filename depends on year and specification)
   This is the vector of equilibrium premiums from running the *Base* scenario.

   (c) *Base setting premiums* (filename depends on year and specification)
   This is the vector of equilibrium premiums from running the same scenario, but with inertia. It will be the same as *equilibrium premiums* if the scenario includes inertia.

   The output metrics are saved in a file whose name depends on the scenario/year/specification being run.

5. `simulate.std.errors.jl`

   This code calculates standard errors for the simulation results by running `run.counterfactuals.jl` 100 times. Note that there are flag variables at the beginning of the code to set up the scenario. There are no data inputs beyond those in `run.counterfactuals.jl`. For each of the 100 simulation runs, there are two files (one with the equilibrium premiums and one with output metrics). Each file is appended with the run number. These output files are loaded in `make.inertia.tables.R` to calculate standard errors. Note that `simulate.std.errors.jl` takes a very long time to run, but can be sped up if runs are performed in parallel.

3.4 Output Processing

1. `assess.model.fit.holdout.R`

   This code creates the data objects needed to assess the fit in `Julia` of our estimated alternative models. While `make.data.objects.R` uses a 5% "estimation" sample, this file prepares the remaining 95% "holdout" sample. Inputs include:

   (a) *Processed Individual Enrollment Data* (`ca_enrollment_data_AUG022019`)

(b) *Processed Household Enrollment Data* (ca_household_data_AUG022019)

(c) *Processed Plan Data* (ca_plan_characteristics_AUG032019_small)

(d) *Processed Choice Set Data* (zip3_choices2.csv)

(e) *Age Rating Factor Data* (age_rating_factors.csv)

(f) *Geographic Cost Factors* (ca_cms_geographic_factors.csv)

(g) *Poverty Guideline Data* (poverty_guidelines.csv)

(h) *ACA Contribution Percentage Data* (contribution_percentages.csv)

(i) *Plan Definitions* (product_definitions.csv)

(j) *Market Eligibility Imputation* (sipp_logit)

Outputs include:

- fit_julia_data.csv
  This file (at the household-plan level) contains data needed to predict choices in the holdout sample.
- fit_households.csv
  This file contains household-level characteristics for the holdout sample.

2. calculate.model.fit.holdout.forward.jl

   This file calculates the fit of each alternative model using the estimated demand parameters and the output data from assess.model.fit.holdout.R. Note that there are a large number of flag variables at the beginning of the code that needed to be adjusted, depending on which model fit is being calculated. Inputs include:

   (a) *Processed Holdout Sample Data* (fit_julia_data.csv)

   (b) *Processed Holdout Household Characteristics Data* (fit_households.csv)

   (c) *Processed Plan Data* (ca_plan_characteristics_AUG032019_small)

   (d) *Demand Parameter Estimates*
   Estimates were aggregated in demand_parameter_estimates.csv using Excel.

   The output file (filename depends on specification) contains information on how well the model fits the data.

3. study.prem.dynamics.R

   This code assesses how premiums change in the exchanges. Inputs include:

   (a) *Processed Household Enrollment Data* (ca_household_data_AUG022019)

   (b) *Processed Plan Data* (ca_plan_data2.csv)

(c) *Market Eligibility Imputation* (sipp_logit)

The output includes regression tables of plan percentage premium increase and lagged market share.

4. create.summary.stats.R

   This code creates the summary statistics table in the paper. Inputs include:

   (a) *Processed Individual Enrollment Data* (ca_enrollment_data_AUG022019)
   (b) *Processed Household Enrollment Data* (ca_household_data_AUG022019)
   (c) *Processed Plan Data* (ca_plan_data2.csv)
   (d) *Processed Choice Set Data* (zip3_choices2.csv)
   (e) *Plan Definitions* (product_definitions.csv)
   (f) *Network Breadth Data* (ca_network_data10.csv)
   (g) *Market Eligibility Imputation* (sipp_logit)

   Outputs include:

   (a) basic_demographics_inertia.csv
   Summary table with basic demographics.
   (b) new_renewal_premiums.csv
   Table summarizing premiums for new and returning enrollees.

5. make.trans.matrices.R

   This code makes transition matrices between metal tiers and insurers. Inputs include

   (a) *Processed Individual Enrollment Data* (ca_enrollment_data_AUG022019)
   (b) *Processed Household Enrollment Data* (ca_household_data_AUG022019)
   (c) *Processed Plan Data* (ca_plan_data2.csv)
   (d) *Processed Choice Set Data* (zip3_choices2.csv)
   (e) *Plan Definitions* (product_definitions.csv)
   (f) *Market Eligibility Imputation* (sipp_logit)

   Outputs include the transition matrices (saved in output.csv and output.perc.csv).

6. make.inertia.paper.tables.R

   This code makes most of the tables and figures in the paper. Much of the input is read in directly from .csv files produced by the counterfactual code run.counterfactuals.zeroresid.jl. Some of the output is saved as .csv files. These .csv files are combined in the file Paper Figures BW.xlsx, where we create figures for the paper. Using the stargazer or xtable packages, the code also produces LaTex code that can be copied and pasted directly into the paper LaTex document (e.g., most of our regression tables are made using this code).

4 Figures and Tables

4.1 Main Paper

- Figure 1: Annual Enrollee Plan Transitions by Metal and Insurer

  This figure includes transition matrices between metal tiers and between insurers. The data for this figure is created in `make.trans.matrices.R`. The figures are made in the Excel file `Paper Figures BW.xlsx`.

- Figure 2: Impact of Inertia on Premiums by Setting

  This figure summarizes the main simulation results on premiums that are produced by running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`. The figures are made in the Excel file `Paper Figures BW.xlsx`.

- Figure 3: Impact of Inertia on Coverage by Setting—Overall, by Metal, and by Insurer

  This figure summarizes the main simulation results on coverage that are produced by running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`. The figures are made in the Excel file `Paper Figures BW.xlsx`.

- Figure 4: Impact of Inertia and Setting on Consumer Switching

  This figure summarizes the main simulation results on consumer switching that are produced by running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`. The figures are made in the Excel file `Paper Figures BW.xlsx`.

- Figure 5: Impact of Inertia on Premiums and Enrollment by Subsidy Design and Consumer Churn

  This figure summarizes the simulation results on consumer churn that are produced by running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`. The figures are made in the Excel file `Paper Figures BW.xlsx`.

- Figure 6: Impact of Inertia on Average Annual Per-Capita Welfare by Setting

  This figure summarizes the main welfare results that are produced by running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`. The figures are made in the Excel file `Paper Figures BW.xlsx`.

- Figure 7: Average Switching Costs and Premium and Welfare Impacts of Inertia, Accounting for Provider Network

  This figure summarizes the demand estimates (switching costs) and main simulation results when we include network variables that are produced by running `estimate.demand.inertia.jl` and `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`. The figures are made in the Excel file `Paper Figures BW.xlsx`.

- Table 1a: Average Premiums for New and Renewing Enrollees, 2015-2018

  This table runs compares premiums of new and renewing enrollees. The table is created in `descriptive_inertia_tabfig.do`.

- Table 1b: Association between Plan Market Share and Percentage Premium Increases

  This table runs descriptive regressions of percentage premium increases on plan market share. The table is created in `study.prem.dynamics.R`.

- Table 2: Annual Switching Costs ($)

  This table summarizes annual switching cost estimates. The data come from `estimate.demand.inertia.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table 3: Supply Parameter Estimates

  This table contains the estimated supply-side parameters. The data come from `estimate.supply.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

4.2 Online Appendix

- Figure A1: ACA Age Rating Factors

  Figure shows the CMS age rating factors from Centers for Medicare and Medicaid Services (2013).

- Figure A2: California Geographic Rating Areas

  This figure is from California Health Benefit Exchange (2016).

- Figure A3: Silver Premiums for a 40-Year-Old by Insurer and Rating Area in 2017

  This figure reports silver premiums for a 40-year-old by insurer and rating area in the 2017 plan year. The data come from `study.prem.dynamics.R`.

- Figure A4: Simulation Results: Variability

  This figure shows the variability of the simulation results in Appendix Table A8 for the Base setting. The data come from running `simulate.std.errors.jl`. The data are compiled in `make.inertia.tables.R`.

- Figure A5: Impact of Inertia on Claims by Setting

  This figure shows the impact of inertia on average claims by setting and average claims by metal tier. The data come from running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`.

- Figure A6: Impact of Inertia on Average Premiums, Coverage, and Claims—Alternative Demand Parameters

  This figure summarizes the simulation results based on demand parameters from specification (6) of Table A4, and supply parameters from specification (1) of Table A7. The data come from running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`.

- Figure A7: Impact of Inertia on Average Premiums, Coverage, and Claims—Alternative Supply Parameters

  This figure summarizes the simulation results based on demand parameters from specification (5) of Table A4, and supply parameters from specification (2) of Table 3. The data come from running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`.

- Figure A8: Consumer Surplus Impact of Inertia by Demographic Group

  This figure reports the impact of removing inertia on consumer surplus by demographic group. The data come from running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`.

- Figure A9: Inattention Specification—Premium, Claims and Welfare Impacts of Inertia

  This figure compares the impact of removing inertia on average premiums, average claims, and welfare for two different models: (1) the switching cost model, which includes a lagged choice variable in the utility specification; and (2) the inattention model, which models the probability that consumers become attentive and search for a new plan. The data come from running `run.counterfactuals.jl`. The data are compiled in `make.inertia.tables.R`.

- Table A1: Plan Choices and Enrollee Demographics

  This table summarizes enrollee-weighted unsubsidized and subsidized premiums, enrollee plan choices, and demographic distributions using California administrative data. The data come from `create.summary.stats.R`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A2: Provider Network Variable Summary Statistics

  This table presents summary statistics on the network variables. The data come from `create.summary.stats.R`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A3: CMS Geographic Cost Factors for California (2014-2018)

  This table shows the geographic cost factors that CMS used in California between 2014 and 2018 for determining risk adjustment transfers. The data come from `process.risk.scores.R`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A4: Estimated Demand Parameters

  This table presents the estimated demand parameters. The data come from `estimate.demand.inertia.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A5: Assessing Out-Of-Sample Model Fit

  This table compares our data to the out-of-sample fit of six alternative specifications. The data come from `calculate.model.fit.holdout.forward.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A6: Own-Premium and Exchange Coverage Elasticities

  This table reports average elasticities by demographic group for six specifications. The data come from `estimate.demand.inertia.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A7: Supply Parameter Estimates—Alternative Demand Specification

  This table presents supply parameter estimates based on specification (6) of Table A4. The data come from `estimate.supply.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A8: Simulation Results: Premiums, Coverage, and Claims

  This table reports the impact on premiums, coverage, and claims for each main counterfactual scenario. The data come from running `run.counterfactuals.jl`. The code `make.inertia.tables.R` compiles the data and produces `LaTex` code for the paper using the `xtable` package.

- Table A9: Sensitivity to Equilibrium Assumption: Premiums, Coverage, and Claims

  This table reports the impact on premiums, coverage, and claims using only data from 2016-2018 to estimate the model parameters. The table data come from running `run.counterfactuals.jl`. The code `make.inertia.tables.R` compiles the data and produces `LaTex` code for the paper using the `xtable` package.

- Table A10: Impact of Subsidy Design and Churn: Premiums, Coverage, and Claims

  This table reports the impact on premiums, coverage, and claims for each of the churn counterfactual scenarios. The data come from running `run.counterfactuals.jl`. The code `make.inertia.tables.R` compiles the data and produces `LaTex` code for the paper using the `xtable` package.

- Table A11: Sensitivity to Provider Networks—Demand Parameters

  This table reports key demand parameter estimates, incorporating provider network preferences. The data come from `estimate.demand.inertia.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A12: Impact of Subsidy Design and Churn: Premiums, Coverage, and Claims

  This table reports the impact on welfare for each of the churn counterfactual scenarios. The data come from running `run.counterfactuals.jl`. The code `make.inertia.tables.R` compiles the data and produces `LaTex` code for the paper using the `xtable` package.

- Table A13: Simulation Results: Change in Annual Per-Capita Social Welfare

  This table reports the impact on welfare for each main counterfactual scenario. The data come from running `run.counterfactuals.jl`. The code `make.inertia.tables.R` compiles the data and produces `LaTex` code for the paper using the `xtable` package.

- Table A14: Simulation Results: Sensitivity to Inertia Role in Welfare

  This table reports the sensitivity of the change in annual per-capita consumer surplus and social welfare for each scenario relative to the Base (or ACA) scenario. The data come from running `run.counterfactuals.jl`. The code `make.inertia.tables.R` compiles the data and produces `LaTex` code for the paper using the `xtable` package.

- Table A15: Sensitivity of Annual Switching Costs to Inclusion of Network Variables

  This table shows the sensitivity of annual switching costs to alternative specifications of utility that include various measures of the plan provider network. The data come from running `run.counterfactuals.jl`. The code `make.inertia.tables.R` compiles the data and produces `LaTex` code for the paper using the `xtable` package.

- Table A16: Sensitivity to Provider Networks: Simulation Results

  This table reports the impact of inertia and provider networks on monthly premiums, coverage, switching rates, monthly average claims, and annual per-capita welfare. The data come from running `run.counterfactuals.jl`. The code `make.inertia.tables.R` compiles the data and produces `LaTex` code for the paper using the `xtable` package.

- Table A17: Inattention Specification—Estimated Demand Parameters

  This table reports the key estimated demand parameters for the inattention specification. The data come from `estimate.demand.inertia.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A18: Inattention Specification—Elasticities and Semi-Elasticities

  This table reports the estimated elasticities and semi-elasticities for the inattention specification. The data come from `estimate.demand.inertia.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A19: Inattention Specification—Supply Parameter Estimates

  This table reports the estimated supply parameters for the inattention specification. The data come from `estimate.supply.jl`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

- Table A20: Control Function—First Stage Parameter Estimates

  This table reports the first stage parameter estimates for the control function approach. The data come from `perform.reduced.form.R`. The `R` code `make.inertia.tables.R` reads in the compiled data and produces `LaTex` code for the paper using the `xtable` package.

References

California Health Benefit Exchange (2016). *Data and Research*. http://hbex.coveredca.com/data-research/.

Centers for Medicare and Medicaid Services (2013, August). *Market Rating Reforms*. http://www.cms.gov/CCIIO/Programs-and-Initiatives/Health-Insurance-Market-Reforms/state-rating.html.

Ruggles, S., K. Genadek, R. Goeken, J. Grover, and M. Sobek (2016). Integrated public use microdata series: Version 6.0 [machine-readable database]. *University of Minnesota*.