Replication File Guide for

"Are Poor Cities Cheap for Everyone? Non-Homotheticity and the Cost of Living Across U.S. Cities"

Jessie Handbury[1]

April 25, 2021

Accessing the Data

This project uses the following data:

- RMS Scanner and HMS Panel data provided by Nielsen and accessed through the Kilts Center for Marketing at the University of Chicago Booth School of Business. Interested researchers may register to gain access to the RMS scanner data and the HMS panel data used herein online at https://www.chicagobooth.edu/research/kilts/datasets/nielseniq-nielsen. Upon approval, Kilts will provide instructions on how to access each of these data sets. For additional questions not addressed on the website, please contact marketingdata@chicagobooth.ed

- Census data on the population, income, land area, and centroid locations of different geographies (CBSA, county, Census tract, zipcode). These data are sourced from public data sources including the Michigan Population Studies Center, NHGIS (Manson et al. (2018)), and various Census websites. These data are included in the rawdata_inputs folder for replication purposesand described in rawdata_inputs/README_filelist.txt.

Replication Summary

The replication code is divided into six steps, all of which are run in Stata unless otherwise specified. The first three steps (Step 0a, Step 0b, and Step 1) prepare the data, while the latter three (Step 2, Step 3, and Step 4) run the analysis presented in the paper:

- Step 0a imports the raw HMS and RMS data creating and saving the data as dta files.

- Step 0b categorizes UPCs into products, cleans the HMS data and calculates size-adjusted household income, and collapses the RMS data to the store-month-product level.

[1] handbury@wharton.upenn.edu

- Step 1 uses the HMS purchase data and Census data to approximate the income distribution around each store; calculates the price instruments used in estimation; runs first stage regressions testing these instruments; and outputs the estimation data that will be read into Matlab.
- Step 2 creates the stylized facts plots and tables.
- Step 3 estimates demand within each product module in Matlab and then reads the results from this estimation back into Stata.
- Step 4 uses the demand parameter estimates to calculate income- and market-specific price indexes, and studies how these price indexes vary across markets differentially for households at different income levels.

Beyond the core functions outlined above, each of these steps also includes code to summarize and describe the data, in some cases producing tables and figures included in the manuscript or appendices. In what follows you can find a full accounting of the data used and created by each step in the pipeline.

This code was run via a series of bash script files. The full sequence of script files, accounting for dependencies between steps (and sub-steps), is listed in README_Pipeline.txt. Script files and any data read in by them may need to be converted to unix to be run on a unix machine with the dos2unix command. The file directories, specified as globals macros below, are set in `SetFilePaths.do`. Other macros (either local, e.g., ‘year’, or global, ${year}) included in the file names below indicate that the code generates versions of that data for each value of the macro. These macros are specified in the arguments listed in the script files, or in the Stata code itself.

All of the files located in ${raw‘suf’} directories are included in the rawdata_inputs folder and described in rawdata_inputs/README_filelist.txt.

Replication File Details

Step 0a: Data Construct

- `HMS/0_dataimport.do`: This code imports the raw HMS data.
  - Run from `HMS/0_dataimport.sh`
  - Input data (all located in year-specific HMS file directories, ${rawHMS}/‘year’/Annual_Files/):
    - panelists_‘year’.tsv contains demographic and geographic location of Nielsen panelists
    - trips_‘year’.tsv contains summary information about each shopping trip recorded by panelists
    - purchases_‘year’.tsv contains information about products purchased on each trip with quantity and price paid
    - products_extra_‘year’.tsv contains extra product attributes
  - Output data (all outputted to ${HMSdata}): panelists_‘year’.dta; trips_‘year’.dta; purchases_‘year’.dta; products_‘year’.dta
- `RMS/0a_prepcharacteristicinfo.do`: This code imports the raw RMS product and store characteristic data.
  - Run from `RMS/0a_prepcharacteristicinfo.sh`
  - Input data:
    - ${rawRMS}/Master_Files/Latest/products.tsv contains product characteristics, downloaded in 2014 import of Kilts data.
    - ${rawRMS}/‘year’/Annual_Files/rms_versions_‘year’.tsv contains Kilts version numbers for UPCs in each year of data
    - ${rawRMS}/‘year’/Annual_Files/stores_‘year’.tsv contains store characteristics
  - Output data (all outputted to ${RMSdata}): products_‘year’.data; stores_‘year’.dta, rms_versions_2006-2014.dta
- `RMS/0b_listprodmods.do`: This code creates a list of all product modules in the RMS data and the product groups into which they are categorized.
  - Run from `RMS/0b_listprodmods.sh`

    - Input data: ${RMSdata}/products.dta created in RMS/0a_prepcharacteristicinfo.do
    - Output data: ${RMSdata}/allmodulesrms_2006-2014.dta

- `RMS/1_createworkingfiles.do`: This code collapses RMS sales data to the monthly frequency.
    - Run from `RMS/1_createworkingfiles.sh` which runs code in parallel for all modules in file ${RMSdata}/allmodules_2006-2014.txt
    - Input data:
        - ${RMSdata}/rms_versions_2006-2014.dta created in 0a_prepcharacteristicinfo.do
        - ${RMSdata}/allmodulesrms_2006-2014.dta created in 0b_listprodmods.do
        - ${rawRMS}/${year}/Movement_Files/${pgrp}_${year}/‘pmod’_${year}.tsv contains store- and UPC-level sales data for products in module ‘pmod’ located in product group ${pgrp}
    - Output data: ${RMSdata}/WorkingFiles/monthly_‘pmod’_${year}.dta

Step 0b: Data Import

- `0a_rawdataconvert.do`: This code defines the chain code and prepares miscellaneous inputs to other programs in 0b_dataimport and beyond.
    - Run from `STEP0a_rawdataconvert.sh`
    - Input data:
        - ${RMSdata}/stores_‘year’.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfo.do
        - ${RMSdata}/products_2006-2014.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfo.do
    - Output data:
        - ${step0bdata}/stores_‘year’.dta

        - ${step0bdata}/RMS_retailer_groupchain.dta
        - ${step0bdata}/storechainloc.dta
        - ${step0bdata}/product_module_department_concordance.dta
        - ${step0bdata}/productmodule_list.dta and ${step0bdata}/productmodule_list.txt

- `0b_createproductids_ctlbrandsRMS.do`: This code assigns a retailer-specific brand ID for control brands in the RMS data, in preparation for the create product IDS step.
    - Run from `STEP0b1_createproductids_ctlbrandsRMS.sh`, `STEP0b2_createproductids_ctlbrandsRMS.sh`, and `STEP0b3_createproductids_ctlbrandsRMS.sh`
    - Input data:
        - ${RMSdata}/WorkingFiles/month_${pmodule}_${year}.dta created in 0a_dataconstruct/RMS/1_createworkingfiles.do
        - ${step0bdata}/storechainloc.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
        - ${step0bdata}/RMS_retailer_groupchain.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - Output data:
        - ${step0bdata}/RMSbrand_bymodule/RMSupcbrand_${pmodule}_${year}.dta
        - ${step0bdata}/RMSbrands.dta
- `0c_createproductids.do`: This code identifies modal units in each module and assigns a product ID to each product module-brand.
    - Run from `STEP0c_createproductids.sh`
    - Input data:
        - ${HMSdata}/trips‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
        - ${HMSdata}/purchases‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do

    - ${RMSdata}/products.dta and ${HMSdata}/products_extra_‘year’.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfo.do
    - ${HMSdata}/products_extra_‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
    - ${step0bdata}/productmodule_list.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - ${step0bdata}/RMSbrands.dta created in 0b_dataimport/Code/0b_createproductids_ctlbrandsRMS.do
    - ${step0bdata}/product_module_department_concordance.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - measurementunitconversion.xlsx
  - Output data:
    - ${step0bdata}/HMSbrands.dta
    - ${step0bdata}/size1_units.dta
    - ${step0bdata}/size1_units_pm.dta
    - ${step0bdata}/productmodule_list_noncount.dta/txt
    - ${step0bdata}/productmodule_year_list_noncount.dta/txt
    - ${step0bdata}/RMSproductids.dta
    - ${step0bdata}/HMSproductids.dta
    - ${step0bdata}/productids.dta
    - ${step0bdata}/RMSproductid_bymodule/RMSproductids_‘p’.dta

- `1a_HMS_purchasedata.do`: This code cleans the HMS data by selecting households who report their purchases continuously and generating purchases datasets based on cleaned households.
  - Run from `STEP1a_HMS_purchasedata.sh`
  - Input data:
    - ${HMSdata}/trips‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do

        - ${HMSdata}/purchases‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
        - ${step0bdata}/HMSbrands.dta created in 0b_dataimport/Code/0c_createproductids.do
        - ${step0bdata}/HMSproductids.dta created in 0b_dataimport/Code/0c_createproductids.do
    - Output data:
        - ${step0bdata}/HHtrips.dta
        - ${step0bdata}/households‘freq’.dta
        - ${step0bdata}/HMSstoreleveldata_‘freq’.dta
        - ${step0bdata}/HMSdata_‘freq’.dta
        - ${step2bdata}/HMS_droppedbrands_‘freq’.dta

- `1b_HMS_demographicdata.do`: This code cleans and exports household demographic data and calculates household size-adjusted income.
    - Run from `STEP1b_HMS_purchasedata.sh`
    - Input data:
        - ${HMSdata}/panelists‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
        - ${step0bdata}/householdsmonth.dta created in 0b_dataimport/Code/1a_HMS_purchasedata.do
        - ${step0bdata}/HMSdata_month.dta created in 0b_dataimport/Code/1a_HMS_purchasedata.do
    - Output data:
        - ${step0bdata}/hhid_char.dta
        - ${step0bdata}/incomeadj.dta

- `1c_RMS_datasetup.do`: This code calculates RMS product sales value and amount at the store-freq level.
    - Run from `STEP1c_RMS_datasetup.sh`

  - Input data:
    - ${RMSdata}/WorkingFiles/month_${pmodule}_${year}.dta created in 0a_dataconstruct/RMS/1_createworkingfiles.do
    - ${step0bdata}/storechainloc.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - ${step0bdata}/RMSbrand_bymodule/RMSupcbrand_${pmodule}_${year}.dta created in 0b_dataimport/Code/0b_createproductids_ctlbrands
    - ${step0bdata}/RMSproductids.dta created in 0b_dataimport/Code/0c_createproductids.do
  - Output data:
    - ${step0bdata}/store_${freq}_product/${freq}_${pmodule}_${year}_productlevel.dta

- `1d_RMS_monthlysales.do`:
  - Run in `STEP1d1_RMS_monthlysales_chunk.sh` and `STEP1d2_RMS_monthlysales_agg.sh`
  - Input data:
    - ${RMSdata}/WorkingFiles/month_‘p’_‘y’.dta created in 0a_dataconstruct/RMS/1_createworkingfiles.do
    - ${step0bdata}/productmodule_list_noncount.dta created in 0b_dataimport/Code/0c_createproductids.do
  - Output data:: ${step0bdata}/store_monthlysales.dt

- `2_incomedistchecks.do`: This code compares the distribution of household income in HMS data with that in ACS and the distribution of income around stores to that around stores in the CBP.
  - Run from `STEP2_incomedistchecks.sh`
  - Input data:
    - ${rawcensus}/ACS/ACS_12_1YR_S1901.csv
    - ${rawcbp}/2016/cbp16co/cbp16co.txt
    - ${rawNHGIS}/2014/nhgis0008_ds206_20145_2014_county.csv

        - ${HMSdata}/panelists2012.dta created in 0a_dataconstruct/HMS/0_dataimport.do
        - ${RMSdata}/stores_2012.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfo.do
    - Tables/Figures:
        - Appendix Figure A.2a: Distribution of Household Income in Nielsen vs ACS data (NielsenCBP_income_equalweight.eps)
        - Appendix Figure A.2b: Distribution of Store Local Income in Nielsen RMS and County Business Patterns (NielsenCBP_income_equalweight.e

Step 1: Data Prep

- `0a_infer_storezip_from_HMS.do`:
    - Run from `STEP0a_infer_storezip.sh`
    - Input data:
        - ${HMSdata}/trips‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
        - ${HMSdata}/panelists‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
    - Output data:
        - ${step1data}/storezip_from_HMS_nhouseholds.dta
- `0b_storedensityestimation.do`: This code estimates density of household shopping trips/expenditure around stores.
    - Run from `STEP0b_storedensityestimation.sh`
    - Input data:
        - ${HMSdata}/panelists‘year’.dta created in 0b_dataimport/1b_HMS_demographicdata.do
        - ${HMSdata}/trips‘year’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
        - ${RMSdata}/stores_2012.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfo.do

        - ${step1data}/storezip_from_HMS_nhouseholds.dta created in 1_dataprep/Code/0a_infer_storezip_from_HMS.do
        - ${rawcensus}/Gaz_zcta_national.txt (zip locations)
        - ${rawNHGIS}/zipcode_population/nhgis0020_ds‘num’_‘acsyr’_zcta.csv (zip population)
        - ${rawsocial}/R11667808.dct (county population and area)
        - ${rawNHGIS}/Gaz_tract_national.txt (tract locations)
        - ${rawcensus}/us2010trf.txt (2000 to 2010 tract relationship file)
        - ${rawNHGIS}/nhgis_medinc_‘acsyr’_tract.csv (tract income distribution)
    - Output data:
        - ${step1data}/household_tripexpshare.dta
        - ${step1data}/storetypeConcordance.dta
        - ${step1data}/storeexpnditure_tractDist_interpolate.dta
        - ${step1data}/storeexpenditure_byincome.dta
    - Tables/Figures:
        - Appendix Figure A.1: Sales Density by Store Type (salesdensity_storetype1_bypopdens.eps and salesdensity_storetype2_bypopdens.eps)

- `0c_store_incomeparameterest.do`: This code estimates income parameters for the store income distributions.
    - Run from `STEP0c1_nation_incomeparameterest.sh` and `STEP0c2_store_incomeparameterest.sh`
    - Input data:
        - ${step1data}/storeexpenditure_byincome.dta created in 1_dataprep/Code/0b_storedensityestimation.do
    - Output data:
        - ${step1data}/ParameterEstimation/incpctiles_‘store’_ln.dta

- `1_priceinstruments.do`: This code creates price instruments for products, to be merged into estimation data in step 2.

  - Run from `STEP1_priceinstruments.sh`
  - Input data:
    - ${step0bdata}/store_${freq}_product/${freq}_${pmodule}_${year}_productlevel.dta created in 0b_dataimport/Code/1c_RMS_datasetup.do
    - ${step0bdata}/storechainloc.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - ${rawMABEL}/county2cbsa10.xlsx
  - Output data:
    - ${step1output}/priceinstr_avguv_${freq}/${freq}_${pmodule}_${year}_product_prinstr1_avguv.dta
    - ${step1output}/priceinstr_avguv_${freq}/${freq}_${pmodule}_${year}_product_prinstr2_avguv.dta

- `2a_outputforMATLAB_prep.do`: This code defines a unique market id and estimates an income distribution, for both store-freq and cbsa-freq markets.
  - Run from `STEP2a_outputforMATLAB_prep.sh`
  - Input data:
    - ${RMSdata}/stores_‘year’.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfo.do
    - ${step0bdata}/storechainloc.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - ${step0bdata}/store_monthlysales.dta created in 0b_dataimport/Code/1d_RMS_monthlysales.do
    - ${step1data}/ParameterEstimation/incpctiles_‘store’_ln.dta created in 1_dataprep/Code/0c_store_incomeparameterest.do
    - ${step1data}/storetypeConcordance.dta created in 1_dataprep/Code/0b_storedensityestimation.do
    - ${rawMABEL}/county2cbsa10.xlsx
    - ${rawmisc}/SocialExplorer/county_income_acs_2010_2014.csv
  - Output data:
    - ${step1data}/store_‘freq’_mktid_correspondence.dta

        - ${step1data}/storeincdist_logincpctiles.dta
        - ${step1data}/storeincdist_logincpctiles_meaninc.dta
        - ${step1data}/storeincdist_logincpctiles_‘freq’_nocd.dta
        - ${step1data}/List_mktid_withincdist_logincpctiles_‘freq’_nocd.dta
        - ${step1output}/dataMatLab_inbrkt_nocd/storeincdist_logincpctiles_‘freq’_nocd.csv
        - ${step1output}/dataMatLab_inbrkt_nocd/List_mktid_withincdist_logincpctiles_‘freq’_nocd.csv
        - ${step1data}/incomedist_lognormal_moments_bycbsa.dta
        - ${step1output}/dataMatLab_cbsa/storeincdist_‘freq’.dta/csv
        - ${step1output}/dataMatLab_cbsa/List_mktid_withincdist_‘freq’.dta/csv

- `2b_outputforMATLAB_RMS.do`: This code cleans the RMS data sample for estimation and outputs into a csv file for analysis in MATLAB.
    - Run from `STEP2b_outputforMATLAB_RMS.sh`
    - Input data:
        - ${step0bdata}/store_${freq}_product/${freq}_${pmodule}_‘year’_productlevel.dta created in 0b_dataimport/Code/1c_RMS_datasetup.do
        - ${step0bdata}/storechainloc.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
        - ${step0bdata}/RMSproductid_bymodule/RMSproductids_${pmodule}.dta created in 0b_dataimport/Code/0c_createproductids.do
        - ${step1output}/priceinstr_avguv_${freq}/${freq}_${pmodule}_${year}_product_prinstr1_avguv.dta created in 1_dataprep/Code/1_priceinstr
        - ${step1output}/priceinstr_avguv_${freq}/${freq}_${pmodule}_${year}_product_prinstr2_avguv.dta created in 1_dataprep/Code/1_priceinstr
        - ${step1data}/List_mktid_withincdist_logincpctiles_${freq}_nocd.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
        - ${step1output}/dataMatLab_cbsa/storeincdist_${freq}.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
        - ${rawMABEL}/county2cbsa10.xlsx
    - Output data:

    - ${step1output}/dataMatLab${marketlevelsuf}/storecsv_${freq}/${year}/outside${outsidegoodscut}/store_module${pmodule}_data.csv
    - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/outliers_${pmodule}.dta
    - ${step1output}/outsidegoods${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/outsidegoods_${pmodule}_${freq}.dta
    - ${step1data}/RMSmkts_bymodule${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/mktscount_${pmodule}_${freq}.dta
    - ${step1data}/RMSmkts_bymodule${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/mkts_${pmodule}_${freq}.dta
    - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/'stage'_prods_${pmodule}.dta
      where stage is preclean, postmerge, postdropmiss, postoutlier, postoutsidegood1, postoutsidegood2
    - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/'stage'_mkts_${pmodule}.dta
      where stage is preclean, postmerge, postdropmiss, postoutlier, postoutsidegood1, postoutsidegood2

- `2c_outputforMATLAB_HMS.do`: This code cleans the HMS data sample based on markets left in RMS estimation data and outputs into a csv file for analysis in MATLAB.
  - Run from `STEP2c_outputforMATLAB_HMS.sh`
  - Input data:
    - ${HMSdata}/purchases'year'.dta created in 0a_dataconstruct/HMS/0_dataimport.do
    - ${HMSdata}/trips'year'.dta created in 0a_dataconstruct/HMS/0_dataimport.do
    - ${step0bdata}/households${freq}.dta created in 0b_dataimport/Code/1a_HMS_purchasedata.do
    - ${step0bdata}/HMSbrands.dta created in 0b_dataimport/Code/0c_createproductids.do
    - ${step0bdata}/HMSproductids.dta created in 0b_dataimport/Code/0c_createproductids.do
    - ${step0bdata}/incomeadj.dta created in 0b_dataimport/Code/1b_HMS_demographicdata.do
    - ${step0bdata}/hhid_char.dta created in 0b_dataimport/Code/1b_HMS_demographicdata.do
    - ${step0bdata}/storechainloc.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - ${step1data}/store_${freq}_mktid_correspondence.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do

        - ${step1data}/List_mktid_withincdist_logincpctiles_${freq}_nocd.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
        - ${step1output}/dataMatLab_cbsa/storeincdist_${freq}.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
        - ${step1data}/RMSmkts_bymodule${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/mkts_'p'_${freq}.dta created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
        - ${step1output}/outsidegoods${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/outsidegoods_'p'_${freq}.dta created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
        - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/postoutlier_mkts_${pmodule}.dta created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
        - ${rawMABEL}/county2cbsa10.xlsx
    - Output data:
        - ${step1output}/dataMatLab${marketlevelsuf}/HHcsv_${freq}/${year}/outside${outsidegoodscut}//HH_module'p'_data_outside.csv
        - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/HMS_'stage'_hhmkts.dta where stage is preprodiddrop, postprodiddrop, preclean, postmerge1, postmerge2
        - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/HMS_'stage'_hhmkts_'p'.dta where stage is postclean

- `3a_RMS_test2sls_regs.do`: This code tests 2SLS estimation using RMS data outputted for analysis in MATLAB.
    - Run from `STEP3a_RMS_test2sls_regs.sh`
    - Input data:
        - ${step1output}/dataMatLab${marketlevelsuf}/storecsv_${freq}/${year}/outside${outsidegoodscut}/store_module${pmodule}_data.csv created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
        - ${step1data}/storeincdist_logincpctiles_${freq}_nocd.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
        - ${step1output}/dataMatLab_cbsa/storeincdist_${freq}.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do

- Output data:
    - ${step1output}/RMS_test2sls${marketlevelsuf}/${freq}/outside${outsidegoodscut}/testestimates_newZ2_module${pmodule}.dta

- `3b_RMS_test2sls_compare.do`: This code puts together 2SLS estimates outputted in 3a and produces graphs comparing moments.
  - Run from `STEP3b_RMS_test2sls_compare.sh`
  - Input data:
    - ${step1output}/RMS_test2sls${marketlevelsuf}/${freq}/outside${outsidegoodscut}/testestimates_newZ2_module${pmodule}.dta created in 1_dataprep/Code/3a_RMS_test2sls_regs.do
    - ${step0bdata}/productmodule_list_noncount.txt created in 0b_dataimport/Code/0c_createproductids.do
    - ${step0bdata}/size1_units_pm.dta created in 0b_dataimport/Code/0c_createproductids.do
  - Tables/Figures:
    - Appendix Figure A.5: Summary Statistics for First Stage Results (month_dist.eps)

- `4a_datacleaning_accounting.do`: This code produce tables on remaining products and markets after each step of data cleaning, in RMS and HMS data.
  - Run from `STEP4a_datacleaning_accounting.sh`
  - Input data:
    - ${RMSdata}/WorkingFiles/month_${pmodule}_${year}.dta created in 0a_dataconstruct/RMS/1_createworkingfiles.do
    - ${step0bdata}/productmodule_list.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - ${step0bdata}/productmodule_list_noncount.txt created in created in 0b_dataimport/Code/0c_createproductids.do
    - ${step0bdata}/storechainloc.dta created in 0b_dataimport/Code/0a_rawdataconvert.do
    - ${step0bdata}/RMSbrand_bymodule/RMSupcbrand_‘module’_${year}.dta created in 0b_dataimport/Code/0b_createproductids_ctlbrandsRM

  - ${step0bdata}/RMSproductids.dta created in 0b_dataimport/Code/0c_createproductids.do
  - ${step0bdata}/hhid_char.dta created in 0b_dataimport/Code/1b_HMS_demographicdata.do
  - ${step0bdata}/incomeadj.dta created in 0b_dataimport/Code/1b_HMS_demographicdata.do
  - ${step1data}/store_${freq}_mktid_correspondence.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
  - ${step1data}/storeincdist_logincpctiles_${freq}_nocd.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
  - ${step1output}/dataMatLab_cbsa/storeincdist_${freq}.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
  - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/'stage'_prods_${pmodule}.dta where stage is preclean, postmerge, postdropmiss, postoutlier, postoutsidegood1, postoutsidegood2, created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
  - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/'stage'_mkts_${pmodule}.dta where stage is preclean, postmerge, postdropmiss, postoutlier, postoutsidegood1, postoutsidegood2, created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
  - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/HMS_'stage'_hhmkts.dta where stage is preprodiddrop, postprodiddrop, preclean, postmerge1, postmerge2, created in 1_dataprep/Code/2c_outputforMATLAB_HMS.do
  - ${step1output}/dataMatLab${marketlevelsuf}/storecsv_${freq}/${year}/outside${outsidegoodscut}/store_module${pmodule}_data.csv created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
  - ${rawMABEL}/county2cbsa10.xlsx

- Tables/Figures:
  - Appendix Table A.1: Summary Statistics on the Share of Data Dropped in Each Step of Data Cleaning (datacleaning_accounting_cbsa_table_rms.tex and datacleaning_accounting_cbsa_table_hms.tex)
  - Appendix Table A.2: Correlation between Data Dropped and Store/Household Income (datacleaning_accounting_cbsa_rms_incregs.tex and datacleaning_accounting_cbsa_hms_incregs.tex)

- `4b_producttable.do`: This code checks which modules exist throughout the pipeline and produces a table of module counts.

  - Run from `STEP4b_producttable.sh`
  - Input data:
    - ${RMSdata}/products.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfo.do
    - ${step0bdata}/RMSproductids.dta created in 0b_dataimport/Code/0c_createproductids.do
    - ${step0bdata}/size1_units_pm.dta created in 0b_dataimport/Code/0c_createproductids.do
    - ${step1output}/dataMatLab${marketlevelsuf}/storecsv_${freq}/${year}/outside${outsidegoodscut}/store_module‘p’_data.csv created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
    - ${step1output}/dataMatLab${marketlevelsuf}/HHcsv_${freq}/${year}/outside${outsidegoodscut}/HH_module‘p’_data_outside.csv created in 1_dataprep/Code/2c_outputforMATLAB_HMS.do
    - ${step1output}/cleaning_acct${marketlevelsuf}/${freq}/${year}/outside${outsidegoodscut}/postoutsidegood1_prods_‘p’.dta created in 1_dataprep/Code/2b_outputforMATLAB_RMS.do
  - Tables/Figures:
    - Table 1: Summary Statistics (countstable_estimation.tex)

Step 2: Pre-estimation Analysis

- `1_stylizedfact1.do`: This code generates Figure 1 in the paper showing that high-income households purchase different, more expensive products that low-income households.
  - Run from `1_stylizedfact1_script.sh`
  - Input data:
    - ${rawMABEL}/county2cbsa10.xlsx
    - ${HMSdata}/panelists‘y’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
    - ${step0bdata}/hhid_char‘foodsuffix’.dta created in 0b_dataimport/Code/1b_HMS_demographicdata_vN.do

        - ${step0bdata}/HMSstoreleveldata‘foodsuffix’_‘time’.dta created by 0b_dataimport/Code/1a_HMS_purchasedata_vN.do
        - ${step0bdata}/month_${pmodule}_${year}_productlevel.dta created in 0b_dataimport/Code/1c_RMS_datasetup.do
        - ${step0bdata}/RMSproductids_${pmodule}.dta created in 0b_dataimport/Code/0c_createproductids.do
    - Sub-programs: main calls p_income_quality
    - Tables/Figures:
        - Figure 1: Average Log Price Paid (wu_income_quality_2012_month.eps)

- `2_stylizedfact2.do`: This code generates Figure 2 and Table 2 in the paper showing that product availability and prices charged in high-income cities and neighborhoods favor high-income household purchases.
    - Run from:
        - `2ai_stylizedfact2_cbsahhprep_script.sh` runs code to calculate product-level expenditure shares for each income group
        - `2aii_stylizedfact2_cbsabsprep_script.sh` runs code to calculate product-level sales for each bootstrap sample of stores in each CBSA
        - `2aiii_stylizedfact2_cbsabscalc_script.sh` runs code to calculate share of decile expenditures represented (variety index) and decile-specific hedonic price index for each bootstrap sample of stores in each CBSA
        - `2aiv_stylizedfact2_cbsabsplot_script.sh` runs code to analyze descriptive variety and hedonic price indexes to produce Figure 2
        - `2b_stylizedfact2_store_script.sh` runs all code required for Table 2
    - Input data:
        - ${rawMABEL}/county2cbsa10.xlsx
        - ${rawNHGIS}/2014/nhgis0007_ds206_20145_2014_cbsa.csv
        - ${rawNHGIS}/2014/nhgis0008_ds206_20145_2014_county.csv

    - ${rawNHGIS}/2014/nhgis0004_shapefile_tl2014_us_cbsa_2014/US_cbsa_2014.csv
    - ${rawNHGIS}/2014/nhgis0005_shapefile_tl2014_us_county_2014/US_county_2014.csv
    - ${raw}/cbsa_abbr.dta
    - ${rawcbp}/county_retail_counts_from_cbp.dta created directly from CBP raw data downloaded at https://www.census.gov/programs-surveys/cbp/data/datasets.html
    - ${step2data}/RMSmarketcoverage.dta created manually using DMA descriptions and coverage table in Kilts-Nielsen documentation, contact the author for these data if you have access to Kilts.
    - ${HMSdata}/panelists‘y’.dta created in 0a_dataconstruct/HMS/0_dataimport.do
    - ${RMSdata}/1_moduleyear/Data/stores_‘y’.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfo.do
    - ${step1data}/storeincdist_logincpctiles.dta created in 1_dataprep/Code/2a_outputforMATLAB_prep.do
    - ${step0bdata}/store_year_product/year_‘p’_‘y’_productlevel.dta created by 0b_dataimport/Code/1c_RMS_datasetup.do
    - ${step0bdata}/incomeadj.dta created in: 0b_dataimport/1b_HMS_demographicdata.do
    - ${step0bdata}/HMSdata_year.dta created by 0b_dataimport/Code/1a_HMS_purchasedata.do

  - Sub-programs: main calls...
    - p_mktdata preps the income and population data for the markets included in the analysis
    - p_incspecprodshares calculates product-level expenditure shares for each income group (‘market’specificprodshare.dta and natlprodshare.dta)
    - p_productmarketsales calculates product-level sales for each market (product_‘market’_‘y’_no9104‘channel’_sales‘sample’‘sampleV’.dta)
    - p_cityincomeshares calculates the share of high- and low-income expenditures represented in what is sold in given markets (tmp0_‘mktspecificsh’sharesdiffby‘market’‘channel’_‘incg’_‘y’_RMSavail‘sample’‘sampleV’.dta)
    - p_hedonic runs a hedonic regression to identify the hedonic price index for each market, weighting by the purchases of each income group (hedonic_‘market’‘channel’_‘incg’_‘y’_RMSavail‘sample’‘sampleV’.dta)
    - p_reg_plot creates output tables and plots

  - Output data:
    - ${step2data}/StoreSample_Crosswalks/storecodesample‘sampleN’_‘market’‘channel’‘sampleV’_‘year’.dta used in Step 4: Post-Estimation Analysis
  - Tables/Figures:
    - Figure 2: Differences in the Availability and Prices (wu_mktspecificsharediffbycbsa_lfit_productid_10_2012_RMSavail_limit50.eps and wu_mktspecifichedonicdiffbycbsa_lfit_productid_10_2012_RMSavail_limit50.eps)
    - Table 2: Differences in Availability and Relative Price of High-Income and Low-Income Baskets Across Stores (reg_sharediffbystores_F_regres
and reg_hedonicdiffbystores_F_regression_productid_10_2012_RMSavail_storeinc.tex)
    - Appendix Figure A.3: Availability Differences for Alternative Sampling Methods (wu_mktspecificsharediffbycbsa_lfit_productid_10_2012_RM
and wu_mktspecificsharediffbycbsa_lfit_productid_10_2012_RMSavail_limitcbp20.eps)
    - Appendix Figure A.4: Availability and Relative Price of High- and Low-Income Baskets Across CBSAs (wu_mktspecificsharelevelsbycbsa_lfit_
and wu_mktspecifichedoniclevelsbycbsa_lfit_productid_10_2012_RMSavail_limit50.eps)
    - Appendix Table A.4: Sample Size, Population, and Income by CBSA (MarketStats.tex)

- `3_nonRMSpurchasedatacheck.do`: This code generates Appendix Table A.3, confirming that household expenditures on non-RMS products do not vary with CBSA income differentially for high- vs. low-income households.
  - Run from `3_nonRMSpurchasedatacheck.sh`
  - Input data: a subset of the datasets used in 2_stylizedfact2.do, listed above.
  - Sub-programs: main calls...
    - p_mktdata preps the income and population data for the markets included in the analysis
    - p_productsbyCBSA identifies products available in RMS sample for each city (productsinRMS_bycbsa_‘y’.dta)
    - p_productmarketsales calculates product-level sales for each market (expshareinRMS_bycbsa_‘y’.dta)
    - nonRMSregcheck regresses expenditure shares on city income by income decile

- Tables/Figures:
  - Appendix Table A.3: Share of Each Income Decile's Non-RMS Store Purchases on Products Available in RMS Retailers by CBSA against CBSA Income (reg_shareinRMS_lnpcinc_2012_byinc.tex)

Step 3: Estimation

- `MATLAB/runmicroblp.m`: This code estimates the parameters for each product module and is run for the baseline and robustness specs from `STEP3_0_runmicroblp.sh`. The main estimation code runs all of the sub-programs whose files start with prefixes `step0` through `step 4bi`. The other files without a step prefix are functions used in the code. This code was run on MATLAB R2020a, using Artelys Knitro 12.3.0 with the options specified in `knitroOptions2.opt`.
  - Input data, all located in 1_dataprep/STATAOutput/dataMatlab_cbsa, with arguments specified in `STEP3_0_runmicroblp.sh`:
    - Market sales data: storecsv_'time'/2012/outside'outsidegoodscutoff'/storemodule'modid'_data.csv created in `1_dataprep/Code/2b_out`
    - Household purchase data: HHcsv_'time'/2012/outside'outsidegoodscutoff'/HH_module'modid'_data_outside.csv created in `1_dataprep/Code/2c_outputforMATLAB_HMS.do`
    - Market-level income distribution data: storeincdist_'time'.csv created in `1_dataprep/2a_outputforMATLAB_prep.do`
  - Output data, saved to specification-specific directories in 3_estimation/MATLAB/OutputData/:
    - Dimensions: MATLABdimensions_mod'modid'_micro2_'time'_outside'outsidegoodscutoff'.txt
    - Model selection criteria inputs: MATLABmodel_nl_mod'modid'_micro2_'time'_outside'outsidegoodscutoff'.txt
    - Non-linear parameters: MATLABthetas_nl_mod'modid'_micro2_'time'_outside'outsidegoodscutoff'.txt
    - Product quality parameters: MATLABdeltas_nl_mod'modid'_micro2_'time'_outside'outsidegoodscutoff'.txt
- `STATA/1_estimatessave.do`: This code imports within-module parameter estimates and is run from `STATA/STEP3_1_runestimatessave`
  - Input data:

        - Market sales data: storecsv_‘time’/2012/outside‘outsidegoodscutoff’/storemodule‘modid’_data.csv created in `1_dataprep/Code/2b_out`
        - Estimates and estimation diagnostics from specification-specific directories in 3_estimation/MATLAB/OutputData.
    - Output data, saved to specification-specific directories in 3_estimation/STATA/Data/estimates:
        - Non-linear parameters only: withinestimates_module${pmodule}.dta
        - Non-linear parameters and mean product quality estimates: withinestimates_betas_module${module}.dta
        - Model selection criteria inputs: withinestimates_modelselection_module${pmodule}.dta

- `STATA/2_estimatesoutput.do`: This code creates plots and tables summarizing the within-module parameter estimates and runs the model-selection tests. It is run for the baseline spec only from `STATA/STEP3_2_outputestimates.sh`.
    - Input data:
        - Estimates and estimation diagnostics from STATA/1_estimatessave.do (withinestimates‘filetype’_module${pmodule}.dta)
        - Module-level sales: ${step0bdata}/HMSdata_year_modulesales.dta
        - Household-level purchases: ${step0bdata}/HMSdata_year.dta created in 0b_dataimport/Code/1a_HMS_purchasedata.do
    - Sub-programs: main calls...
        - collapsemodulesales to calculate module-level sales for weighting purposes
        - appendestimates to append estimates across modules
        - summarizeestimates to summarize estimates and then to create Table 1 summarizing the estimates from the main specifications
        - modelselection to run the model selection analysis (to obtain numbers reported in the manuscript)
        - pcastats to study principal components analysis on instruments (to obtain numbers reported in the manuscript)
        - studyquality to look at correlation between decile-specific brand expenditure shares and quality estimates
        - comparebetas to look at correlation between base and alternative relative product quality (see Appendix D.2)

  - Tables/Figures:
    - Table 3: Summary Statistics for Parameter Estimates (maincoefficientsnonbound_w_fulltable.tex)
    - Figure 4: Product Quality Estimates and High vs. Low Income expenditures (qualitycorrelations_plot_10groups.eps)
    - Appendix Figure A.7: Correlation between Base and Alternative Relative Product Quality Estimates (betadiffcorr_model3.eps)
    - Appendix Figure A.8: Correlation between HMS Panelist Expenditures and Errors in Relative Quality by Income Decile (qualityerrcorrelations_plot_10groups.eps)

- `STATA/3_identificationchecks.do`: This code compares parameter estimates across various robustness specifications to produce appendix figures. It is run from `STATA/STEP3_3_identificationchecks.sh`.
  - Input data: Estimates and estimation diagnostics from STATA/1_estimatessave.do (withinestimates‘filetype’_module${pmodule}.dta)
  - Sub-programs:
    - est_results_compkdensity to plot distribution of estimates from different specifications
    - est_results_Ctest to conduct overidentification test
    - Each of these programs call dataprep and bounds_check, which read in the estimates and identify estimates hitting bounds.
  - Tables/Figures:
    - Appendix Figure A.6: Distribution Price Coefficients Across Modules with Different Price Instruments (triple_scatcomp_OLS_model3_alpha0_

Step 4: Post-Estimation Analysis

- `1_availability_modbymod.do`: This code aggregates the Nielsen RMS store sales data for a given module and year to the store-product level.
  - Run from `STEP4_1_availability.sh`
  - Input data:

        - ${step0bdata}/month_${pmodule}_${year}_productlevel.dta created in 0b_dataimport/Code/1c_RMS_datasetup.do
        - ${step0bdata}/RMSproductids_${pmodule}.dta created in 0_dataimport/0c_createproductids.do
    - Output data: store_module${pmodule}_${year}_data.dta

- `2_moduleindex_allmods.do`: This code calculates module-level price indexes for each market and income group.
    - Run from `STEP4_2_cbsa_moduleindex.sh` for each of 100 versions for bootstrapped baseline CBSA-level indexes; `STEP4_2_store_moduleindex.sh` for single sample of store-level indexes; and `STEP4_2_robust_moduleindex.sh` for each of 100 versions for bootstrapped CBSA-level indexes for robustness specifications.
    - Input data:
        - ${step0bdata}/productmodule_list_noncount.txt created in 0b_dataimport/Code/0c_createproductids.do
        - store_module‘pmodule’_‘year’_data.dta created in 4_postestimation_analysis/1_availability_modbymod.do
        - ${rawNHGIS}/2010/nhgis0004_ds191_20125_2012_county.csv
        - ${RMSdata}/1_moduleyear/Data/stores_‘year’.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfor_2017.do
        - ${step2data}/storetocbsa_‘year’‘channel’.dta created in 2_preestimation_analysis/2_stylizedfact2.do
        - ${step2data}/StoreSample_Crosswalks/storecodesample‘sampleN’_‘market’‘channel’‘sampleV’_‘year’.dta created in 2_preestimation_analysis/2_stylizedfact2.do
        - ${estimates}/withinestimates_betas_module‘pmodule’.dta created in 3_estimation/STATA/2_estimatessave_vN.do
    - Output data: mkt_priceindices_‘market’_‘year’_module‘pmodule’‘sample’‘sampleV’‘incfile’.dta
    - Sub-programs:
        - main calls main_module, either looping through each of the modules in 0_dataimport/Code/productmodule_list_noncount.txt or for the specified module
        - main_module reads in the raw price data created in 1_availability_modbymod.do, collapses it to the CBSA-level for the relevant samples of stores (where relevant), joins the resulting data with the estimates from 3_estimation; and calculates and saves the module-level price indexes (calling calc_indexes)

        - calc_indexes calculates the module-level price indexes

- `3_aggindex.do`: This code aggregates module-level price indexes across modules.
    - Run from `STEP4_3_cbsa_aggindexes.sh` to aggregate bootstrapped baseline CBSA-level indexes; `STEP4_3_store_aggindexes.sh` to aggregate single sample of store-level indexes; and `STEP4_3_robust_aggindexes.sh` to aggregate bootstrapped CBSA-level indexes for robustness specifications.
    - Input data:
        - ${step2data}/product_${year}_sales_storelevel.dta created in 2_preestimation_analysis/2_stylizedfact2.do
        - mkt_priceindices_'market'_'year'_module*'sample''incfile'.dta created in 2_moduleindex_allmods.do
        - StoreSample_Crosswalks/storecodesample${sampleN}_'market''channel'${sampleV}_'year'.dta created in 2_preestimation_analysis/2_stylizedfact2.do
    - Output data:
        - mkt_module_priceindices_'market'_'year''test''sample''incfile'.dta (appended module-level indexes)
        - mkt_priceindices_'market'_'year''test''sample''incfile'.dta (aggregated indexes)
    - Sub-programs - main calls the following programs:
        - modulesales aggregates RMS sales data to the module-level for the calculation of Cobb-Douglas module weights
        - appendmodindexes appends the module-level market-income-specific indexes across modules
        - calcindexes drops modules with extreme outliers, any modules that do not have indexes available in all markets and income groups and then aggregates across modules

- `4_indexcomparison.do`: This code merges the market-level income-specific price indexes with market-level population and income data to estimate the cross-elasticities of these price indexes with respect to household and market income.

- Run from `STEP4_4_cbsa_indexcomparison.sh` for analysis of bootstrapped baseline CBSA-level indexes; `STEP4_4_store_indexco` for single sample of store-level indexes; and `STEP4_4_robust_indexcomparison.sh` for analysis of bootstrapped CBSA-level indexes for robustness specifications.
- Input data:
  - Market- and income-specific price indexes: ${indexes}/mkt_priceindices_‘market’_‘year’‘test’‘sample’‘sampleV’‘incfile’.dta created in 3_aggindexes.do
  - ${step0bdata}/storechainloc.dta created by 0b_dataimport/Code/0a_rawdataconvert.do
  - NHGIS income and population data:
    - CBSA: ${rawNHGIS}/2014/nhgis0007_ds206_20145_2014_cbsa.csv
    - County: ${rawNHGIS}/2010/nhgis0006_ds206_20145_2014_county.csv
  - Store-county concordance: ${RMSdata}/1_moduleyear/Data/stores_‘year’.dta created in 0a_dataconstruct/RMS/0a_prepcharacteristicinfor_20
  - Local incomes around stores: 1_dataprep/STATAData/storeincdist_logincpctiles.dta created in 1_dataprep/2a_outputforMATLAB_prep_vN.do
- Output data:
  - ${output}/temp_‘prices’city2city_‘market’_‘type’‘test’‘sample’‘incfile’.dta: processed price index data ready for robustness check regressions
- Sub-programs - main calls the following programs:
  - store_results calls sub-programs dataprep and output_regs_store
  - cbsa_results calls dataprep, output_excel, and p_comp_prices
- Tables/Figures:
  - Table 4: City-Income Specific Price Index Regressions
    - comp_p_citytocityregnon_cbsa___2012_3_limit50_incDetail_xbsfe.tex
  - Table 5: Store Price Index Regressions

        - HLreg_store_code_uc___2012_3_incDetail.tex
    - Excel file containing indexes (averaged across bootstrapped samples) for statistics referenced in the manuscript
        - ${output}/cityindexes_cbsa_bootstrap.xls

- 4a_indexcomparison_speccomptables.do: This code merges the market-level income-specific price indexes and market-level population and income data, as well as statistics from the demand estimates, for various robustness specifications to construct a table comparing the parameter estimates and price index cross-elasticities across specifications.
    - Run from STEP4_4a_indexrobustnesstable.sh.
    - Input data:
        - temp_‘prices’city2city_‘market’_‘type’‘test’‘sample’‘incfile’.dta: processed price index data ready for robustness check regressions from 4_indexcomparison.do, run via STEP4_4_robust_indexcomparison.sh for each of the main robustness specifications
        - appended_wihthinestimates.dta created in 3_estimation/STATA/1_estimatessave.do for each robustness specification
    - Tables/Figures:
        - Table 6: Robustness of Index Elasticities (comptoallrobust_withparmest_p_citytocityreg‘wgtsuf’_‘name’_‘m’‘test’‘sample’‘incfile’_xbsfe.tex)

References

Manson, S., J. Schroeder, D. Van Riper, and S. Ruggles (2018). IPUMS National Historical Geographic Information System: Version 13.0.