What Determines Consumer Financial Distress?

Place- and Person-Based Factors

Readme

Benjamin J. Keys* Neale Mahoney† Hanbin Yang‡

February 28, 2022

1 Data

Our main data comes from credit report data provided by TransUnion, a global information solutions company, through a relationship with the Kilts Center for Marketing at The University of Chicago Booth School of Business. The data was stored in a server at Chicago Booth and we provide the code to extract data for our analysis.

We supplement our main data with geographic covariates from various sources. Raw data are stored in /Data/Raw. These data are publicly available and we provide some instructions to access them

- Zip level 2011-2015 5-year ACS from America Fact Finder
- ACS and CPS for moving patterns in 2005 and 2015
- Zip level 2013-2015 5 year ACS from census (Figure A21)
- 2014 Provider of Services (POS)
- Bank branches from 2014 FDIC data

*The Wharton School and NBER. Email: benkeys@wharton.upenn.edu
†Stanford University and NBER. Email: neale.mahoney@gmail.com
‡Harvard Business School. Email: hanbin.v.yang@gmail.com

- Chapter 7 and 13 bankruptcy fees from Lupica (2012)
- State median seizable assets from Mahoney (2015); wage garnishment from `nolo.com`

We also use the following crosswalk, in /Data/Crosswalk. These crosswalk can be downloaded from, for example, `https://www.psc.isr.umich.edu/dis/data/resource/`

- 2000 and 2010 puma to cz (puma_cz_cross_2000, puma_cz_cross_2010)
- county to cz (cw_zip_to_county)
- zip to county (cw_cty00_cz)
- court district to county (cw_cty_courtdistrict)
- cz to state (cz_name)
- county to state with county level population and region as weights (county_population_area)

2 Code

2.1 Extract TransUnion data from server

Data in the Booth server are stored in Parquet format and we use python scripts to extract data using sparkSQL packages. See /Code/1_Spark. Copy outputs from these scripts to /Data/Clean/

SparkDataTools.py: include functions to extract variables from parquet data tables in the Booth server and construct financial distress measures

01_extract.py: extract snapshots of financial distress measures at various geographic levels. Outputs are **sumstats_yyyymm_<geo>.csv**. <geo> are possible geographic aggregation levels including commuting zone "cz", state "st", zipcode "zip", and individual level without aggregation "indi"

02_mover.py: classify individuals into once movers and non-movers, extract financial distress measures at quarterly level for movers and non-movers. Output is **mv_nmv.csv**

The following codes extract data for exhibits in the appendix:

03_count_with_trade.py: extract number of observations with and without trade information as of 6/2016. The output is **June2015_count_cz.dta** and used for Figure A8.

04_move_pattern.py: calculate the percent moved across zip/county/state in the past 1 and 5 years. Outputs are **moving_200509.dta** and **moving_201509.dta**. These are used to compute move frequency in Table A2.

05_credit_limit.py: construct credit score bins of 10 $\times$ CZ level credit limit and current balance. Outputs are **cl_score_yyyymm.dta**. These are inputs to compute credit supply measures for the correlates of place effects in Appendix C.

06_cc_decay.py track the status of credit card accounts that were current in 1/2009 and non-current in 6/2009. Output is **cc_decay_200906.dta** and used for Figure A1.

07_coll_decay.py track the status of collections accounts in 6/2009 that were opened in the past six months. . Output is **coll_decay_200906.dta** and used for Figure A1.

08_collection_breakdown.py: show breakdown of collection by creditor types as of 2015. The output is **coll_201506.dta** and used for Table A1.

09_bkrptcy.py: extract public record at individual level. The output is **Bkrptcy_BH_yyyymm.dta** and used for Figure A21.

2.2 Aggregate supplementary data

Code to aggregate supplementary data are in /Code/2_Build

01_acs_2015_cov process zip level demographic information from 2011-2015 5-year ACS. We downloaded zip level information from various tables in American Fact Finder (`https://factfinder.census.gov/faces/affhelp/jsf/pages/metadata.xhtml?lang=en&type=dataset&id=dataset.en.ACS_15_5YR`). American Fact Finder is no longer supported and please use the current census website to access these information. Raw data stored in /Data/Raw/ACS_15_5YR_XXXX. We create state, CZ, and zip level covariates: **acs_2015_st.dta, acs_2015_cz.dta, acs_2015_zip.dta**

02_medical_cov process medicare provider of service (POS) from CMS (which we downloaded from NBER `https://www.nber.org/research/data/provider-services-files`) to construct the fraction of for profit hospital. Raw data stored as Data/Raw/POS_OTHER_DEC14. We also included mortality due to prescription opioid poisoning from CDC but did not use this data in our analysis. Generate **TU_Geo_Debt_Hospital_2014.dta**.

03_bank_cov process FDIC banking data from `https://www7.fdic.gov/sdi/download_large_list_outside.asp` to construct measures of banking access as CZ covariates. Raw data stored as Data/Raw/All_Reports_20140630_Assets and Liabilities.csv, FDIC_2014.csv. Generate **TU_Geo_Debt_Banking_Access_2014.dta**.

04_bankruptcy_fee.do organize bankruptcy attorney fee information from Lupica (2012). Raw data stored as Data/Raw/TU_bankruptcy_fee. Generate **TU_bankruptcy_fee.dta**.

05_tu_credit_limit.do consolidate credit limit from 2011 to 2015 and aggregate average credit limit at CZ level, weighted by national credit score distributions. Generate **cz_credit_limit.dta**

06_migration_cps_acs.do migration pattern as of 2005 and 2015 from CPS and 1-year ACS sample. Raw sample extractions from ACS and CPS stored as Data/Raw/IPUMS_ACS_05_16, Data/Raw/cps. Outputs are used in **03_sumstats.do** which generate Table A2.

2.3 Mover analysis

Code to produce our main analysis are in /Code/3_Analysis

runall.do: master script to set up directory and run all or selected do files

00_config.do set up the configuration including global variables of geographic level, key variable lists, and sample restrictions. Also load functions in **01_programs.do**. The most important section is "run geographic level", where we select the geographic level. Always run this before any of the below do files.

01_programs.do: functions.

02_indi_to_cz.do load individual level data extracted from the server **sumstats_201506_indi.csv** from **01_extract.py**, winsorize balance variables and aggregate to CZ level (**sumstats_201506_inditoCZ.csv**). Also produce summary statistics Table 1. Set geographic level to individual in **00_config.do** .

03_sumstats.do load CZ level aggregates (**sumstats_201506_inditoCZ.csv**), merge demographic variables from ACS, geographic covariates from multiple sources, as well as state law measures (state median seizable asset and wage garnishment: Data/Raw/state_legal). Produce maps of financial distress measures (Figure 1, A2,A3,A4,A5), summary of variation (Table 3), and other summary statistics tables (Table A1, A2). Set geographic level to CZ in **00_config.do** .

04_decom_variation.do aggregate CZ level snapshots each year from the server **sumstats_yyyymm_cz.csv**, decompose variance of financial distress measures at CZ level into within and between, and generate Table 2.

05_rank_stability.do aggregate CZ level snapshots in 2001 and 2015 from the server.

Generate rank stability and correlation plots (Figure A6,A7).

06_decay_plots.do show decay pattern of credit card and collection to compare flow vs stock measures. Generate Figure A1.

07_mover_delta.do: load mover and non mover data **mv_nmv.csv** and compute δ of financial distress measures between geographic regions. Change geographic levels in **00_config.do** to generate δ at zipcode level (baseline), zipcode $\times$ 10 year age bin, and county level. Save the results in **tu_mover_delta_<geo>.dta**.

08_mover_event_study.do: load **tu_mover_delta_<geo>.dta**, and run event study regressions. Create histograms of δ (Figure A9,A8,A11), pre-trend plots (Figure 2,3), and event study plots (Figure 4, A12, A14). Change geographic levels and also sample restrictions to look at different horizon post moves (see global **evt_sample_version**) in **00_config.do**. See section "mover event study what to run" in **00_config.do** to select sections to run. Also save event study coefficients.

09_event_study_tables.do: aggregate event study results from **08_mover_event_study.do**, and create estimation tables (Table 4, A4, A6).

10_mover_sum_stats.do: generate the distribution of number of non-movers used to compute δ. Run this for different levels of geographic levels. The outputs are tex files **sumstats_<geo>_delta_numobs_mover_<geo_extension>_v1.tex**. See the "run geographic level" section in **00_config.do**. We then aggregate these into one single tex file **sumstats_delta_numobs_mover_v1.tex**

11_two_way_fe.do: load **tu_mover_delta_<geo>.dta**, and decompose place vs person using a two-way fixed effect model. Also run the model across bootstrap data.

12_two_way_fe_tables.do: load results from **11_two_way_fe.do**, and create estima-

tion table (Table A5).

13_state_level_correlates.do: run two way fixed effects at state level to estimate state level place effects. Then merge with state level correlates and prepare data to run variable selection using LASSO in R. The output is **TU_place_vs_customer_cov_std.dta**

14_place_vs_person_lasso.R. Note that this is an R script. Run LASSO to select correlates with each state level place effect of financial distress measures. Run both version 1 and 2 by setting variable in line 44. The outputs are **post_lasso_v<ver>.csv**, where <ver> is 1 or 2.

15_place_vs_person_project.do take variable selections from R to project place effect on correlates. Generate Figure A15-A20

16_maps_movers.do compute number of movers in each origin/destination CZ and produce maps. Produce Figure A8.

17_bankruptcy_income.do load **Bkrptcy_BH_yyyymm.dta** and create binned scatter plots of bankruptcy filing by income. We use zip level income from 2013-2017 5-year ACS sample accessed through census (`https://data.census.gov/cedsci/`). Raw data stored as Data/Raw/ZIP_vars_ACS. Produces Figure A21.