

Branch: master ▾

ui\_public / README.md

Find file

Copy path



ganong123 add DOI badge

4871a95 an hour ago

1 contributor

144 lines (108 sloc) 10.1 KB

# Ganong and Noel "Consumer Spending During Unemployment: Positive and Normative Implications" (2018)

By Peter Ganong and Pascal Noel

Please send feedback and questions to [ganong@uchicago.edu](mailto:ganong@uchicago.edu).

DOI 10.5281/zenodo.2531618

## empirics

### Empirical analysis of public use data

Master.R runs all the following scripts.

- `prelim.R` loads all the necessary packages required to run the scripts.
- `1_chase_state_map.R` produces online Appendix Figure 1 (`out/chase_footprint.png`), map of states with Chase branches and direct deposit of UI benefits.
- `2_scf.R` uses SCF to produces
  - statistics for statements in the text:
    - "We calculate using the SCF that about 15 percent of labor income is paid by paper checks and pre-paid debit cards rather than by direct deposit."
    - "90 percent of households reporting UI income in the past year in the SCF had a bank account at the time of the survey."
    - "We estimate using the SCF that 64 percent of UI recipients with a bank account have an outside credit card."
    - "According to the Department of Labor Benefit Accuracy Measurement system, in states where the maximum potential duration of UI benefits is six months, about 15 percent of UI recipients are eligible for five months of benefits."
  - checking account balance in online Appendix Figure 3 used for representativeness analysis (`out/for_jpmci/scf_ck_bal.csv`).
  - online Appendix Table 6 (`out/for_jpmci/at6_scf_liq_asset.csv`), the distribution of all liquid assets and

checking account balances for representativeness analysis.

- `3_sipp_extract_cols.R` cleans raw SIPP data by extracting columns we use in our analysis.
- `4_sipp_build_ui_event_study.R` further cleans SIPP data by identifying people receiving UI benefits.
- `5_sipp_who_are_ui_recipients.R` produces
  - online appendix Table 5 ( `out/for_jpmci/at5_sipp_inc.csv` ), representativeness analysis in income between JPMC and SIPP.
  - age distribution in online Appendix Figure 3 used for representativeness analysis ( `out/sipp_wt_age.csv` ).
  - distribution of income prior to UI receipt in online Appendix Figure 3 used for representativeness analysis ( `out/sipp_wt_inc_10k.csv` ).
- `6_spending_crosswalk.R` produces spending coverage analysis:
  - online appendix Table 2 ( `out/at2_jpmc.csv` ), comparison of monthly spending between JPMC and external CE and PCE surveys.
  - online appendix Table 3 ( `out/at3_cex.csv` ), comparison of monthly spending between JPMC and CE surveys.
  - online appendix Table 4 ( `out/at4_pce.csv` ), comparison of monthly spending between JPMC and PCE surveys.
- `7_dcpc_benchmark.R` produces online Appendix Table 21 ( `out/at21_dcpc_dur_share.csv` ), payment methods analysis.
- `8_ui_benefits.R` calculates average weekly UI benefits in 2015 for online Appendix Figure 2 ( `out/for_jpmci/avg_weekly_ui_benefit.csv` ).

Auxiliary code is not needed for the replication, but we provide it if you want to start from scratch. Both scripts should be run after `prelim.R` and before analysis code.

- `download_data.R` downloads raw SCF and DCPC data and provides download instructions of SIPP data, and convert them to Rdata format.

## **sources**

- 2013 Survey of Consumer Finances [full public dataset](#) and [summary extract public data](#).
- 2012 Diary of Consumer Payment Choice [public use datasets](#).
- 2004 Survey of Income and Program Participation core wave files. We use Stata files created by [NBER](#).
  - [2004w1](#)
  - [2004w2](#)
  - [2004w3](#)
  - [2004w4](#)
  - [2004w5](#)
  - [2004w6](#)
  - [2004w7](#)
  - [2004w8](#)
  - [2004w9](#)
  - [2004w10](#)
  - [2004w11](#)
  - [2004w12](#)

## JPMCI data

---

Some of the data used for this paper were prepared in JPMorganChase Institute's (JPMCI) secure computing facilities. Due to JPMCI's rules on access and confidentiality, the programming code and analysis files cannot be made available publicly. The analysis files and programming code created by the authors will be available within JPMCI's secure computing facilities until 2021, and can be requested by researchers with approved projects (email [institute@jpmchase.com](mailto:institute@jpmchase.com)). We grant any researchers with appropriate approval to conduct research on JPMCI's secure computing facilities access to these files. Below, we list the tables needed to replicate the analysis as well as some key variables

### tables

- ganong\_ui\_aer\_archive\_need\_res\_dir\_approval\_to\_delete\_wide
- ganong\_ui\_aer\_archive\_need\_res\_dir\_approval\_to\_delete\_long
- peter\_ganong\_do\_not\_delete\_ui\_jobid\_mo\_2017\_09\_29

### key variables

- Household-level variables
  - first\_check\_period\_index
  - last\_check\_period\_index
  - duration -- last\_check\_period\_index - first\_check\_period\_index
  - dur\_weeks
  - exhaust\_cust
  - first\_job\_period\_index
  - dur\_search\_emp -- first\_job\_period\_index - first\_check\_period\_index
  - dur\_stitch -- ifelse(duration <= 5 & !exhaust\_cust, duration, dur\_search\_emp)
- Monthly variables:
  - mos\_since\_start
  - ui
  - labor\_broad

## model

---

We thank Xian Ng for outstanding research assistance.

## Directory Structure

---

1. code/ - all estimation and analysis code
2. input/ - model estimation targets from JP Morgan Chase Institute (JPMCI) data
3. out/ - code output including figures, logfiles, and raw tables
4. Parameters/ - environment parameters for estimation and analysis; stored model estimation results
5. tables/ - formatted tables

## Files in the code/ directory

---

**Master Scripts: do\_min.py, do\_mid.py, do\_all.py**

The master scripts in the code directory re-run the code. To call the script from the command line, go to the `code/` directory and enter 'python '. Note that this project is built on python 2.7

1. `do_min.py` : Solves the consumption and job search models using estimated parameters in `Parameters/model_params_main.json` to replicate the plots in the paper.
  2. `do_mid.py` : `do_min.py` + computes the standard errors on the parameter estimates, and performs the welfare simulations in the paper.
  3. `do_all.py` : `do_mid.py` + estimates the model parameters again using model targets from JPMCI data.
- To simply replicate the results, it is sufficient to run one of the scripts above.

## Setup scripts

1. `setup_estimation_parameters.py` reads in `Parameters/params_ui_reemp.json` to build the estimation environment
2. `build_JPMC_targets.py` builds the model estimation targets from files in `input/` and writes to `Parameters/JPMC_inputs.py`. It only needs to be run once.

## Model Solving, Estimation, and Simulation scripts

3. `solve_cons.py` - contains a function that takes environment and preference parameters and computes optimal consumption in each period as a function of cash-on-hand. It solves this problem using backwards induction for a finite horizon.
4. `job_search.py` extends `solve_cons.py`. Takes environment and preference parameters and computes optimal consumption and job search in each period as a function of cash-on-hand.
  - Agent chooses search effort with an isoelastic cost and with the gains from search equal to  $V_{emp}(a) - V_{unemp}(a)$ .
  - To accomplish the above, we compute a value function which sums over utility in each period. We do not need the value function to compute optimal consumption, but we do need it to compute optimal job search effort.
5. `sparsity.py` solves the sparse model from Gabaix (2016) in the UI context.
6. `estimate_models.py` takes environmental parameters and consumption and job search moments and solves for the preference parameters that generate consumption and job search behavior similar to the moments. Relies on the class in `job_search.py`. By default, solves for the models in the paper one at a time. Can also be used to solve multiple models at once on a cluster.
7. `agent_history.py` With a given set of environmental and preference parameters, simulates employment histories, consumption behavior, and job search behavior for N agents.

## Plotting and Replication scripts

8. `model_plotting.py` contains plotting functions. `make_plots.py` uses the `rrpy2` package to create some wrappers for making PDF plots in R.
9. `model_plots.py` produces plots using the estimated preference parameters contained in `Parameters/model_params_main.json` and `Parameters/model_params_sec.json`.
10. `model_welfare.py` performs the simulations for lifecycle welfare analysis.
11. `comp_SEs.py` calculates standard errors for estimated models. Note: all output is in the `/out/` directory.

## Other files

- `est_robust_gamma.py` estimates models with different risk aversion parameters.
- `estimate_models_helper.py` contains helper functions for estimating many models simultaneously on a cluster.
- `grid_submit.sh` shell script for submitting jobs to a cluster. Edit as necessary.

- `make_plots.py` contains aesthetic options for plots.
- `prelim.py` is a helper script for setting up the model environment.
- `est_models_in/` contains `initial_conditions_master.json` for estimating the models in the paper. csv files in the directory are examples of different initial conditions for model estimation. Convert to JSON for estimation using function in `estimate_models_helper.py`
- `est_models_out` logs final estimates as JSON files when estimating preference parameters for multiple models at the same time on a cluste. Convert to csv using function in `estimate_models_helper.py`