Partial Replication Kit for "Spending and Job Finding Impacts of Expanded Unemployment Benefits: Evidence from Administrative Micro Data"

By Peter Ganong, Fiona Greig, Pascal Noel, Daniel M. Sullivan, and Joseph Vavra

Please send feedback and questions to ganong@uchicago.edu.

https://doi.org/10.1257/aer.20220973

Table of Contents

Overview

This code package allows researchers to replicate the analysis from the paper "Spending and Job Finding Impacts of Expanded Unemployment Benefits: Evidence from Administrative Micro Data." The package includes MATLAB scripts for model simulation, R scripts for benchmarking against public data, and additional scripts used within JPMorgan Chase Institute's secure computing environment. The publicly available code requires Stata, Matlab, and R. The confidential data and code (only available on JPMCI servers) requires Python. Pseudocode is provided, but these scripts will not reproduce the figures from the paper. The replication process involves running the JPMCI scripts first, followed by the benchmarking scripts, and finally the model scripts. The package covers various stages of data processing, model calibration, simulation, and result generation to reproduce much of the findings presented in the paper. Details about what figures can and cannot be produced are provided in `pandemic_ui_public/figure_table_mapping.xlsx`.

Data Availability and Provenance Statements

Statement about Rights

I certify that the authors of the manuscript have legitimate access to and permission to use the data used in this manuscript.

I certify that the authors of the manuscript have permission to redistribute/publish the data contained within this replication package.

Summary of Availability

- □ All data **are** publicly available.
- ☑ Some data **cannot be made** publicly available.
- □ **No data can be made** publicly available.

Data Availability Statement

Some of the data used for this paper were prepared in JPMorgan Chase Insitute'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 2028, and can be requested by researchers with approved projects (email `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 describe the key tables needed to replicate the analysis

Details on each Data Source

This table describes the raw data used in the analysis and their corresponding directories.

Data.Name	Data.Files	Location	Provided	Citations
Department of Labor (DOL)	ar539.csv; ar5159.csv; ae5159.csv; ap5159.csv; ap902.csv; weekly_pandemic_claims.xlsx	/pandemic_ui_public/analysis/input/public_data	TRUE	(Employment and Training Administration 2019 - 2020a); (--- 2019 - 2020b); (--- 2019 - 2020c); (--- 2019 – 2020d)
Federal Reserve Economic Data (FRED)	PAYEMS.xls; PCE.xls	/pandemic_ui_public/analysis/input/public_data/	TRUE	(U.S. Bureau of Labor Statistics 2016-2021); (U.S. Bureau of Economic Analysis 2016-2021)
Bureau of Labor Statistics (BLS)	bls_payroll_emp_nonfarm_no_adj.xlsx	/pandemic_ui_public/analysis/input/public_data/	TRUE	(Bureau of Labor Statistics 2019 - 2021)

Data.Name	Data.Files	Location	Provided	Citations
Schmieder and Von Wachter 2016	literature_elasticities.csv	/pandemic_ui_public/analysis/input/public_data/	TRUE	(Schmieder and Von Wachter 2016)
JPMorgan Chase Institute Data Assets	n/a	n/a	FALSE	(JPMorgan Chase Institute 2018-2021)

Public Data Table

This table explains all the files in the `/pandemic_ui_public/analysis/input/public_data` directory in more detail, excluding the `us_states_hexagrid.geojson`, which is used for the hexmap in figure A-1.

Data file	Source	Notes
`weekly_pandemic_claims.xlsx`	(Employment and Training Administration 2019–2020d)	Unemployment Insurance Weekly Claims Data https://oui.doleta.gov/unemploy/claims.asp
`ae5159.csv`	(Employment and Training Administration 2019–2020b)	ETA 5159 (Extended benefits) https://oui.doleta.gov/unemploy/DataDownloads.asp
`ap902.csv`	(Employment and Training Administration 2019–2020a)	ETA 902P https://oui.doleta.gov/unemploy/DataDownloads.asp
`ap5159.csv`	(Employment and Training Administration 2019–2020b)	ETA 5159 (PEUC) https://oui.doleta.gov/unemploy/DataDownloads.asp
`ar539.csv`	(Employment and Training Administration 2019–2020c)	ETA 539 https://oui.doleta.gov/unemploy/DataDownloads.asp
`ar5159.csv`	(Employment and Training Administration 2019–2020b)	ETA 5159 (Regular program) https://oui.doleta.gov/unemploy/DataDownloads.asp
`bls_payroll_emp_nonfarm_no_adj.xlsx`	(U.S. Bureau of Labor Statistics 2019–2021)	select "Multi Screen" under "Employment, Hours, and Earnings - National" in section "Employment" of the following link https://www.bls.gov/data/home.htm
`decompose_pua.csv`	-	Produced by `decompose_pua.do`
`elig_ui_reg_pua.csv`	(ETA 2019–2020b); (ETA 2019–2020d)	Adds up the number of 'regular' UI payments (ETA 5159) and PUA payments (ETA 902P)
`literature_elasticities.csv`	(Schmieder and Von Wachter 2016)	Table 2 from this paper. Set of elasticities from previous literature.
`monthly_exit_rates.dta`	-	Intermediate dataset saved by `decompose_pua.do`

Data file	Source	Notes
`PAYEMS.xls`	(U.S. Bureau of Labor Statistics 2016–2021)	From https://fred.stlouisfed.org/series/payems
`PCE.xls`	(U.S. Bureau of Economic Analysis 2016–2021)	From https://fred.stlouisfed.org/series/PCE
`ui_receipt_benchmarks.xlsx`	(Employment and Training Administration 2019–2020b)	Sheets `month_final_payments` and `month_initial_claims` are from ETA 5159, and sheet `week_continued_claims` is from Employment and Training Administration 2019–2020d

Computational requirements

Software Requirements

These are the software configurations in which the code was last run.

- Stata (code was last run with version 17)
- Python (only for JPMCI scripts, which will not run)
- Matlab (code was run with Matlab Release 2021b)
- R 4.3.0
  - RColorBrewer version 1.1-3
  - yaml version 2.3.9
  - testthat version 3.2.1.1
  - scales version 1.3.0
  - readxl version 1.4.2
  - ggrepel version 0.9.5
  - geojsonio version 0.11.3
  - broom version 1.0.6 (note: the hexmap requires broom version earlier than 1.0.3)
  - lubridate version 1.9.3
  - tidyverse version 2.0.0
  - rgeos version 0.6-3 (note--this package was depricated in 2023 and is not available in CRAN. It also requires downloading dependency `geos` separately)

Memory and Runtime Requirements

Summary

Approximate time needed to reproduce the analyses on a standard 2024 desktop machine:

- □ <10 minutes
- □ 10-60 minutes
- ☑ 1-2 hours
- □ 2-8 hours
- □ 8-24 hours
- □ 1-3 days
- □ 3-14 days
- □ > 14 days
- □ Not feasible to run on a desktop machine, as described below.

Details

The code was last run on a **3.5 GHz Dual-Core Intel Core i7 with MacOS version 13.6.6 (22G630)**.

The matlab code is not particularly computationally intensive. Everything but the best fit parameters can be run in less than an hour on a standard laptop or desktop computer. Solving for the best fit parameters takes around an hour or two on the same computer but this does not need to be done to replicate results since the results of this parameter search are saved in the replication code.

Data for the python scripts are not included and should not be considered for runtime estimates or memory requirements.

Description of programs/code

Directory Structure

Inputs

1. `analysis/input/disclose/` - Model inputs and targets from JPMorgan Chase Institute (JPMCI) data.
2. `analysis/input/public_data/` - Model inputs and targets from publicly available data. See below for description of the datasets and their sources.

Code

1. `pgm/` - JPMC code (does not run)
2. `analysis/source/` - Benchmarking code
3. `analysis/source/joint_spend_search_model/` - Model code (including subdirectory `robustness/`)

Outputs (Exhibits)

1. `analysis/release/joint_spend_search_model/paper_figures` - Model outputs
2. `analysis/release/ui_benchmarking` - Benchmarking outputs

License for Code

The code is licensed under a MIT license. See `/pandemic_ui_public/LICENSE` for details.

Instructions to Replicators

To replicate the results:

1. The JPMCI scripts would be executed using `ui_driver.sh` first. (Note: these scripts require confidential data not included in the public repository so they will not run.)
2. Run the benchmarking scripts using the R script `driver.R`. You will need to adjust the file paths in the script to match your local directory structure.
3. Execute the model scripts by running `shell.m` in MATLAB. Ensure all references to paths in program prelim.m include `rootdir` using the full file function.

Details

1. JPMCI scripts

Note: This section describes the entire internal JPMCI repository used for this project. The files in the public repository submitted to the American Economics Association includes only the analysis `.R` scripts. The `.py` scripts and their driver script `ui_driver.sh` are not included.

JPMC open source repo

Build

- `ui_driver.sh`: This driver script produces the entire build. Command-line options in `ui_driver.sh` are passed on to the main python script `pgm/daily_inflow_outflow_ui_recip.py` to specify the parts of the build and the time period for which the build should be executed.
- `pgm/daily_inflow_outflow_ui_recip.py`: This is the main python script and the only script called by `ui_driver.sh`. The output is a set of `hdfs` tables, which are also saved as `.rds` tables for analysis:
  - `demog`: tables with customer-by-month info on balances, demographics, and flows,
  - `eips`:
    - `eips_list`: tables of customer-level EIP transactions for UI customers, where EIP here refers exclusively to the April 2020 EIP round,
    - `eip_rounds_list`: tables of customer-level EIP transactions for UI customers with all 3 rounds of EIPs,

  - `weekly_cp`: customer-by-week-by-counterparty tables of labor and UI inflows,
  - `weekly_flows`: customer-by-week flows tables.
- `pgm/funcs/inflow_outflow_helper_funcs.py`: This script defines the helper functions called by `pgm/daily_inflow_outflow_ui_recip.py`.

Analysis

The main driver script is:

- `pgm/R_driver_script.R`: produces a large number of plots, tables and statistics which appear in the July 2023 draft.

Non-Chase inputs:

- DOL ETA Form 203: state-month level count of unemployment insurance claims by NAICS 2-digit industry. File path: `xxx/gnlab/ui_covid/scratch/2021-08-19claimant_industry.csv`

Description of Script `pgm/R_driver_script.R`:

The driver script, `pgm/R_driver_script.R`, run the following scripts in the following order:

Sample Set up

- To run the analysis on a 1% sample, set the vector `small_samp` to *TRUE*. Otherwise, the default is *FALSE* which runs the scripts on the full sample.
- `pgm/data_readin_1pct.R`: If there are new builds made, and there is need to make a new 1% sample, then, set the vector `create_new_1pct_sample` to *TRUE*, which runs this script. It reads in the new full sample builds, and saves new 1% sample builds.

Setting up Functions:

- `pgm/funcs/ui_functions.R`: a number of functions that are common across many later files. Functions include:
  - `gg_walk_save`: writes a ggplot object to PDF, and produces a CSV of the underlying data
  - `gg_point_line`: creates a line plot in ggplot, with a dot at each point on the line.
  - `diff_in_diff`: computes a difference-in-difference estimator, measured as the ratio of (change in treatment group)/(change in control group). The numerator and denominator of the ratio are themselves fractions corresponding to the year-on-year change in the treatment and control groups, respectively.
  - `yoy_change`: computes year-on-year change (or any ratio) estimator.
  - `fte_theme`: theme to construct plots with standardized aesthetic elements
  - `get_median_benefits`: Takes a customer week dataframe and returns the median benefits of the customer within a timeframe given by dates for start and end
  - `grouped_exit_rates`: produce exit rates by time or duration (including by recall status) for those who we observe a separation
  - `estimate`: find difference between average job-finding rate in two weeks prior to policy change to the first four weeks after the policy change.
  - `weekly_summary`: produces a weekly summary dataframe
- `pgm/funcs/prelim.R`: makes function, `winsor`, to winsorize data
- `pgm/funcs/xtile_ten.R`: makes a function, `xtile_ten`, that finds values at a specific percentile (but usually median) within JPMCI data while meeting data aggregation standards by taking the average of the ten values around the entered percentile.
- `pgm/funcs/test_that_modified.R`: this is a modification to the `test_that` functions used in scripts, where instead of returning an error, as is usual, if this is run it gives a warning. To use this, set the vector `warnings` to *TRUE*. This is used extensively while running R batch submission scripts.

Build Script:

Before you run these scripts, there are two set up vectors that will determine how the driver script is run. If you would like to re-run the build scripts, then set the vector `re_run_build_scripts` to *TRUE*. Further, if you would like to run the disaggregated version of the build, which splits consumption into its constituent categories, then set the vector `run_categories` to *TRUE*.

- `pgm/ui_eip_data_read_in.R`: imports weekly counterparty files from `/data/jpmci/teams/gnlab/ui_covid`. This script reads in and lightly cleans RDS files from the PySpark build.

- `pgm/ui_eip_data_build.R`: cleans up the imported data so that it is in a form useful for analysis
- `pgm/jobfind_build_1_of_2.R` and `pgm/jobfind_build_2_of_2.R`: builds the following dataframes:
    - `df_labor_cust_week` which is a dataframe at the customer-by-week level. Shows whether the customer has exited labor or exited UI to a new job or to recall.
    - `df_ui_cust_week_add_spells` which feeds into `df_ui_cust_week`, which is created in `jobfind_build_2_of_2`
    - `df_ui_cust_week_alt_horizon_basic` which feeds into `df_ui_cust_week_alt_horizon` (used as an end product for a plot in `timeseries_plots.R`), and compares various lengths of job seperation.
- `pgm/jobfind_build_2_of_2.R`: uses a number of sample screens to further clean up the dfs from previous build scripts.

NOTE: can skip the first three files and run straight from `pgm/jobfind_build_2_of_2.R` since the prior three builds and saves the relevant rds files and `pgm/jobfind_build_2_of_2.R` reads the files straight in. To run everything from `pgm/jobfind_build_2_of_2.R`, set `re_run_step1 <- FALSE` at the start.

Jobfind Analysis:

- Prep scripts to create controls and dataframes ready for analysis:
    - `pgm/control_prep.R`: this creates controls such as industry (based on organization that paid your last paycheck before separation), age (spell-level), gender.
    - `pgm/rep_rate_prep.R`: calculates the median benefits and % benefit change in two time periods: "expiration" (expiration of $600 FPUC at the end of August) and "onset" (onset of $300 at the start of January 2021).
- Output scripts produce timeseries plots, DID plots, regression tables, etc.
    - `pgm/timeseries_plots.R`: make timeseries plots of exit rates for jobfind analysis using tmp_for_hazard_plot_expanded
        - Outputs: Figures 4, 5, A13ab, A14, A15, A16, A21
    - `pgm/summer_expiration.R`: makes timeseries plots for summer expirations, including exit rates and binscatters.
        - Outputs: Figures A24ab, A25, Table A15
    - `pgm/rep_rate_tables.R`
        - Outputs: Tables 3, A2, A11b, A12, A13b, A14
    - `pgm/marginal_effects_hazard_calc.R`: calculates inputs for hazard elasticity calculations done outside the firewall.
    - `pgm/rep_rate_figs.R`: This script produces plots for event study by above/below median rep rate as well as binscatter plots.
        - Outputs: Figures 6ab, 7ab, A17abcdef
    - `pgm/weekly_coef_figs.R`: This runs regressions with weekly coefficients to new job for binary (above vs below median) and weekly DID, then plots the coefficients.
        - Outputs: Figures A23ab
    - `pgm/ui_universe_read_in_plot.R`: Analyzes all UI recipients for comparison to those who meet the primacy screen (this is run after running all the analysis of the primacy screen)
        - Outputs: Figure A2a
    - `pgm/jobfind_tables.R`: make tables for job-finding analysis
- Robustness checks on controls, e.g. benchmarking our industry mix and interacting our 'main' regression with liquidity:
    - `pgm/industry_mix_change.R`: assess the quality of the industry variable tagging in JPMCI by comparing to an external benchmark (Department of Labor ETA form 203) which gives data on UI claims by industry
        - Outputs: Figure A3
    - `pgm/jobfind_liquidity.R`: This runs regressions interacting with liquidity variable, which is measured as pre-period balance
        - Outputs: Tables A4, A5
- `pgm/save_time_series_for_model.R`: produces model outputs that Joe Vavra uses on the outside
- `pgm/jobfind_stats_export_jan22.R`: creates stats for text for export, minimum aggregation standards tables, other model input that is used on the outside, and a workbook (`[date]_ui_jobfind_for_export.xls`) which also includes any other data frame needed on the outside.

Spend Analysis

- `pgm/spend_build.R`: build data needed for the analysis of spending around UI.
- `pgm/spend_plots.R`: create plots of spending for various event studies/ID strategies around UI.
    - Outputs: Figures 1, 2, 9ab, A4, A5, A6, A7, A8, A9, A10
- `pgm/spend_summer_weekly.R`: produce summer expiration spend plots.
    - Outputs: Figures A11, A12

- `pgm/mpc_robustness.R`: MPC calculations
    - Outputs: Tables 1, A10, A11a
- `pgm/mpc_cats.R`: MPC calculations with disaggregated categories sample
    - Outputs: Tables A7, A8
- `pgm/mpcs_more_controls.R`: MPC calculations with controls
    - Outputs: Table A9
- `pgm/spend_by_liquidity_buffer.R`: Spending by pre-pandemic liquidity group
    - Outputs: Figure 3, Table 2
- `pgm/table2_V2.R`: Create another version of table 2
- `pgm/spend_by_ever_recall.R`: Spending of recalled vs non-recalled workers
    - Outputs: Figure A22
- `pgm/liquidity_distribution.R`: compute some statistics to summarise the magnitude of the reversal of liquidity between unemployed and employed households during the pandemic.
- `pgm/liquidity_changes.R`: Produce liquidity change outputs for different treatment samples
    - Outputs: Table A6\
- `pgm/low_prepand_liq.R`: Low pre-pandemic liquidity group characteristics
- `pgm/spend_summary_stats.R`: Calculate some summary stats on spending and the spend samples

Note: In the repo, there is a folder `r_batch_submission_scripts` with the same R scripts as in `pgm/` to run as a bash job on the edgenode, instead of on Rstudio.

Prior to running the driver script, the pre-processing script `pgm/cust_labor_filter_table.py` creates a count of transactions at the customer-month level that is used in `pgm/daily_inflow_outflow_ui_recip.py` to filter the customer list to primary customers.`

Important note on data structure of `cust_demo` There are 4 'cust_types': `202021_ui_recipient, 2019_ui_recipient, nonui_2020, nonui_2019`. A `2019_ui_recipient` got UI in 2019, but they may also get UI in 2020.

Input tables for Build - List of customers with 2018 and 2019 JPMC activity as well as customer metadata - `institute_consumer.mwl_cust_covid_filters`: filtered customer list with 2018 and 2019 labor inflows - `institute_retail_curated.jpmci_customer_profile`: customer profile table - `institute_consumer.eip_cohort_info`: customer with EIP transaction details - `institute_consumer.mwl_daily_income_rollup_for_covid_inc_updated`: daily inflows table - `institute_consumer.outflows_rollup_by_day_granular`: daily outflows table - `institute_retail_curated.jpmci_deposit_account`: deposit accounts table - `institute_retail_curated.jpmci_customer_account_relationship`: customer-account relationship table - `institute_retail_curated.jpmci_deposit_transaction`: : deposit transaction table (transaction-level) - `institute_retail_curated.jpmci_transaction_counterparty_lookup`: firm-id crosswalk for deposit transaction table - `institute_consumer.ui_nonui_cust_list`: list of UI and non-UI customers - `institute_consumer.industry_classification_w4_sa`: cleaned at_counterparty values (including industries) - `institute_consumer.mwl_ui_cp_raw_lookup_mar2021`: table with UI counterparties matched up with their respective state

These input tables are used to create three tables which are then used in analysis: - weekly file with receipt of UI benefits and labor income including surrogate id for employer - monthly file with UI benefits, other income, several measures of spending, and checking account assets - file with demographics such as age, gender, states of residence, and Economic Impact Payment amount

2. Benchmarking scripts

The paper has a few plots which compare JPMCI data to public data. The R script `driver.R` in `analysis/source/` runs the script `diagnostic_benchmarking_plots.R`, also in `analysis/source/`, to produce plots benchmarking JPMCI numbers to public data. It produces four figures which are in `analysis/release/ui_benchmarking`: - Figure A-1: `hexmap_jpmci_sample.png` - Figure A-2: `diagnostic_levels_norm.png`; `state_hetero_inc_scatter.png`; `weekly_benefits_median_2019_mthly.png`.

Replications notes on figure A-1:

Figure A-1 is a hexmap visualization of the types of JPMCI customer information available by state. It has different environment requirements for reproduction than the rest of the benchmarking plots. It depents on broom 1.0.0, which is incompatible with tidyverse 2.0.0, a package that is required for all other benhcmarking figures. It also uses a retired package called rgeos, which is no longer available on CRAN as of October 2023. Becuase of these unique envoronment needs, A-1 is not able to be replicated using the same environemnt as the rest of the benchmarking figures.

These were the steps taken in the last attempt to replicate this figure:

1. Download GEOS in terminal.

   ```
   brew install geos
   ```

2. Download rgeos package in R.

   ```
   remotes::install_version("rgeos", version = "0.6-3")
   ```

3. Load an earlier version of broom and tidyverse.

   ```
   devtools::install_version('broom', '1.0.0')
   devtools::install_version('tidyverse', '1.3.2')
   ```

4. Use appropriate libraries.

   ```
   library(broom)
   library(tidyverse)
   library(rgeos)
   library(geojsonio)
   ```

NOTES: The hexaplot may require these earlier package versions. Parts of the R benchmarking scrips were written in with these package versions: - yaml version 2.3.7 - testthat version 3.1.9 - scales version 1.2.1 - ggrepel version 0.9.3 - geojsonio version 0.11.1 - broom version 1.0.0 - lubridate version 1.9.2

It may also require an earlier version of R. The latest version of R available at the beginning of the benchmarking script construction was R 3.6.3.

3. Model code details

All in `analysis/source/joint_spend_search_model/`.

Driver Script

The driver script `shell.m` runs the code. The shell file comments also describes exactly which subroutines produce each model figure and table from the paper. The code producing most main text results is in lines 14-50 and only takes a few minutes to run. `solve_best_fit_params.m` is much more time consuming than the rest of the script but we provide intermediate files so this step can be skipped if so desired. Similarly, the `stimulus_check_size` related code and robustness are also more time consuming and can be skipped if not specifically interested in those results.

Setup Script

The script `prelim.m` defines data inputs to the model (paths point to `analysis/input/disclose/` and `analysis/input/public_data/`), sets parameters, and specifies plotting options.

Note about input paths: The Matlab code's relative paths are defined assuming that the current working directory when executing the code is `analysis/source/joint_spend_search_model`. For users executing the code locally in the Matlab GUI, this should be the default behavior. If the code is run from a different working directory, which may be the default when not using the Matlab GUI, users will likely need to instead define a root path and then pre-pend this to the relative file path references using the `fullfile` function. The code was written so that it should require no manual directory changes for the typical user executing locally.

Model Scripts

- solve_best_fit_params.m - Calibrate the search parameters of different expectations assumptions and MPC targets and save them in various intermediate .mat files like bestfit_target_waiting_MPC.mat. This file takes several hours to run, but we provide the .mat files, so shell.m can be run without this time consuming step by commenting out this solve_best_fit_params line. This code relies on functions pre_pandemic_fit_het_inf_horizon.m, sse_fit_het_inf_horizon_full.m, sse_fit_het_inf_horizon.m, and sse_fit_het_inf_horizon_onset.m which solve a prepandemic calibration and the pandemic model for expiration and onset.
- inf_horizon_het_results.m, inf_horizon_het_results_onset.m - Given search parameters calibrated above, this solves and simulates the main model for expiration and onset, respectively.
  - Plots in these scripts include results from other calibrations: prepandemic_results_target500MPC.m, inf_horizon_het_results_target500MPC.m, inf_horizon_het_results_nodiscountfactorshock.m, prepandemic_results_onset_target500MPC.m, inf_horizon_het_results_onset_target500MPC.
- inf_horizon_het_results_stimulus_check_size.m, inf_horizon_het_results_stimulus_check_size_onetenth.m, inf_horizon_het_counterfactuals.m - This code constructs the spending responses for stimulus checks vs. severance of various sizes. It is somewhat time consuming since not written very efficiently (about an hour), and results are only used for Figure 13, so could be skipped if not interested in these counterfactuals.
- liquidity_effects_prepandemic.m - Computes baseline effects of liquidity on job search to compare to Card, Chetty, Weber 2006
- make_table_agg_effects.m, make_table_mpc_for_paper.m, make_table_supplement_effects.m, make_table_alt_job_find_specs.m - Format model outputs for the paper.
- plot_duration_elasticities.m - Plot literature estimates of duration elasticities.
- pandemic_hazard_vs_duration_elasticity_constanteffects_v2.m - Decomposes role of different channels in low elasticity (paper figure 11)
- liquidity_effects_on_mpcs.m - some statistics related to liquidity changes that are briefly mentioned, otherwise this file is mostly deprecated
- inf_horizon_het_results_by_liquidity.m - This redoes the main results but splitting separately by high and low liquidity households constructed in various ways

Robustness Scripts

- inf_horizon_het_results_timeaggregation_target500MPC.m, inf_horizon_het_results_timeaggregation.m - Code for constructing Figure A-18
- Subdirectory /robustness/beta_delta_revision_v2/ - Running the shell file in this folder creates robustness figure A-28. Note that the shell file provides the intermediate files and comments out the running of the model for a large set of parameters, which was run on a cluster with an array job. If you want to re-run this many hour grid search, see grid_search.sh and grid_search_append.sh
- test_homogeneity.m - Homogeneity results which are briefly mentioned in Appendix C.2 (not included in shell since there are no specific numbers reported/saved from this code)

Functions Written for this Project and Called by Routines Above

- average_duration.m - This computes the average duration of unemployment given an exit rate
- elasticity_distortions_and_aggregates.m - This computes duration elasticities as well as measures of aggregate distortions in response to the supplements in the pandemic
- search_elasticity_implications.m - This computes a duration elasticity to a small change in benefits (i.e. a normal benefit elasticity)
- share_remaining_survival.m - Computes share of unemployed remaining over time given exit hazard
- week_to_month_exit.m - Converts weekly exit rates in data to monthly exit rates used in model

Other Functions

- fminsearchbnd.m - Bounded optimization function. Used for optimization in solve_best_fit_params.m.
- cab.m - Close a subset of figures.
- hex2rgb.m - Convert hexadecimal color code to RGB values.
- table2latex_numbers_only.m - Convert MATLAB table to tex Table.

List of tables and programs

The provided code reproduces:

- ☐ All numbers provided in text in the paper
- ☐ All tables and figures in the paper
- ☑ Selected tables and figures in the paper, as explained and justified below.

Figures that are sourced by a file in the /pgm folder will not be reproduced, as they require restricted data. You can view information about what scripts produce what files in `pandemic_ui_public/figure_table_mapping.xlsx`.

References

JPMorgan Chase Institute. 2018 - 2021. JPMorgan Chase Institute De-Identified Data Assets. JPMorgan Chase Institute, New York, NY. https://www.jpmorganchase.com/institute (accessed July 16, 2024).

Schmieder, Johannes F., and Till von Wachter. 2016. Table 2 From "The Effects of Unemployment Insurance Benefits: New Evidence and Interpretation." Annual Review of Economics, 8(1):547–581 https://doi.org/10.1146/annurev-economics-080614-115758

U.S. Bureau of Economic Analysis. 2016-2021. "Personal Consumption Expenditures [PCE]." Retrieved from FRED, Federal Reserve Bank of St. Louis. https://fred.stlouisfed.org/series/PCE (accessed July 18th 2024).

U.S. Bureau of Labor Statistics. 2016-2021. "All Employees, Total Nonfarm [PAYEMS]." Retrieved from FRED, Federal Reserve Bank of St. Louis. https://fred.stlouisfed.org/series/PAYEMS (accessed July 18 2024).

Employment and Training Administration. 2019 - 2020a. “Characteristics of the Insured Unemployed.” United States Department of Labor. https://oui.doleta.gov/unemploy/DataDownloads.asp (accessed July 19, 2022).

———. 2019 - 2020b. “Claims and Payment Activities.” United States Department of Labor.https://oui.doleta.gov/unemploy/DataDownloads.asp (accessed July 19, 2022).

———. 2019 - 2020c. “Weekly Claims and Extended Benefits Trigger Data.” United States Department of Labor. https://oui.doleta.gov/unemploy/DataDownloads.asp (accessed July 19, 2022).

———. 2016–2021d. "Unemployment Insurance Weekly Claims Data." U.S. Department of Labor. https://oui.doleta.gov/unemploy/claims.asp (accessed July 19, 2024).