README for Replication Package Accompanying "Understanding the Size of the Government Spending Multiplier: It's in the Sign"

Regis Barnichon[(a)] *Davide Debortoli*[(b)] *Christian Matthes*[(c)]

[(a)] Federal Reserve Bank of San Francisco and CEPR

[(b)] Universitat Pompeu Fabra, CREi, Barcelona GSE and CEPR

[(c)] Indiana University

April 23, 2021

1 Software

All files were executed on a 2019 MacBook Pro running Matlab R2020b, except for the Stata codes, which were executed on a Windows virtual environment running Stata SE 16.1.
Each code listed below ran in less than 8 hours each, with the recursive identification-scheme codes taking the longest (multiple hours).

2 Stata Packages and Matlab Toolboxes

In order to execute the code below, the following packages/toolboxes are required:

- **Stata:**
  1. **ivreg2**. This package can be installed via the command

     ```
     ssc install ivreg2;
     ```

     Our Stata code already includes this command.
  2. **ranktest**. This package can be installed via the command

     ```
     ssc install ranktest;
     ```

     Our Stata code already includes this command.

- **Matlab**

  1. **Statistics and Machine Learning Toolbox**

3 Content

This section lists all codes that have to be executed to generate tables and figures in the text. Note that figures 1 and 9 are just for exposition (they were not generated with Matlab or Stata), hence there is no code here to generate those figures. We describe the data files used in a separate section below. Codes should be run in the order listed below so that the necessary estimation results are created before the creating the corresponding figures. All figures are saved as both Matlab fig files and eps files using the same numbering as in the paper.

1. To generate Local Projection-based results, run `LP_paper.do` in folder `Ramey_Zubairy_replication_codes`. Note that the user will have to set the Stata working directory to `Ramey_Zubairy_replication_codes` before executing this code. This code will generate results that will be used below to create figure 4.

2. To estimate the linear FAIR model with recursive identification, run `Principal_sym.m` in folder `FAIR/FAIR_Sym`. This code will save results for later use in figures and also display the relevant entries of table 1 in the Matlab command window.

3. To estimate the asymmetric FAIR model with recursive identification, run `Principal.m` in folder `FAIR/FAIR_Asym`. This code will create figure 2 and also display the relevant entries of table 1 in the Matlab command window.

4. To estimate the asymmetric FAIR model with state dependence and recursive identification, run `Principal_final.m` in folder `FAIR/FAIR_AsymState`. This code will create figure 6 and also display the relevant entries of table 2 in the Matlab command window.

5. To estimate the asymmetric FAIR model with state dependence and recursive identification on the sample starting in 1947, run `Principal.m` in folder `FAIR/FAIR_AsymState_47`. This code will display the relevant entries of table 3 in the Matlab command window.

6. To estimate the linear FAIR model with narrative identification, run `Principal_sym_paper.m` in folder `SUR-FAIR/FAIR_Sym`. This code will save results for later use in figures and also display the relevant entries of table 1 in the Matlab command window.

7. To estimate the asymmetric FAIR model with narrative identification, run `Principal_asym_paper.m` in folder `SUR-FAIR/FAIR_Asym`. This code will display the relevant entries of table 1 in the Matlab command window.

8. To estimate the asymmetric FAIR model with state dependence and narrative identification, run `Principal_state_asym.m` in folder `SUR-FAIR/FAIR_AsymState`. This code will display the relevant entries of table 2 in the Matlab command window.

9. To estimate the asymmetric FAIR model with state dependence and narrative identification on the sample starting in 1947, run `Principal_state_asym.m` in folder `SUR-FAIR/FAIR_AsymState_47`. This code will display the relevant entries of table 3 in the Matlab command window.

10. To generate figures 3,4, and 7, run `figures.m` in the folder `SUR-FAIR`.

11. To generate figure 5, run `Plot_indicator.m` in folder `figure 5`.

12. To generate figure 8, run `HistShocks.m` in folder `figure 8`.

13. To solve the benchmark equilibrium model and create figure 10, run `main.m` in the folder `theoretical_model`. The relevant entries of table 4 will be displayed in the Matlab command window.

14. To solve the equilibrium model with perfect insurance, run `main_perfect_insurance.m` in the folder `theoretical_model`. The relevant entries of table 4 will be displayed in the Matlab command window.

15. To solve the equilibrium model with constant elasticity, run `main_constant_elasticity.m` in the folder `theoretical_model`. The relevant entries of table 4 will be displayed in the Matlab command window.

16. To create XLSX files that contain the same information as the tables in the text, run `create_tables.m` in the folder `FAIR`

4 Data Availability Statement

The data we use are based on Auerbach and Gorodnichenko (2012a) and Ramey and Zubairy (2018a), as outlined in the main text. In each folder that contains estimation procedures, there is a file called data.mat that contains the data used in estimation. The exact structure of this data file depends on whether narrative or recursive identification is used.

For recursive identification, the variables are stored in an $N \times T$ array called data. The order of the variables is as outlined in the text: $\left(\Delta g^F_{t|t-1}, g_t, \tau_t, y_t\right)'$ where g is real government purchases (consumption plus investment at the federal, state, and local levels), τ is real government receipts of direct and indirect taxes net of transfers to businesses and individuals, y is gross domestic product (in chained 2000 dollars), and $\Delta g^F_{t|t-1}$ is the growth rate of government spending at time t forecasted at time $t-1$, which is obtained combining the Greenbook and Survey of Professional Forecasters (SPF) quarterly forecasts. As in Ramey and Zubairy (2018a), we ex-ante re-scale all variables by a measure of "potential" output taken from Ramey and Zubairy (2018a).

For narrative identification, we borrow the underlying data from Ramey and Zubairy (2018a). The rows of our data files are ordered as follows (the columns represent time periods):

1. left-hand-side variables
2. ones for the intercept
3. lagged endogenous variables (if any)
4. lagged exogenous variables (if any)
5. measures of shocks

The data from Ramey and Zubairy (2018a) (Ramey and Zubairy (2018b))is available at `https://econweb.ucsd.edu/~vramey/research/Ramey_Zubairy_replication_codes.zip`.

The data from Auerbach and Gorodnichenko (2012a) (Auerbach and Gorodnichenko (2012b)) is available at `http://doi.org/10.3886/E114783V1`.

References

Auerbach, A. J. and Y. Gorodnichenko (2012a). Measuring the output responses to fiscal policy. *American Economic Journal: Economic Policy 4*(2), 1–27.

Auerbach, A. J. and Y. Gorodnichenko (2012b). Replication data for: "Measuring the output responses to fiscal policy". `http://doi.org/10.3886/E114783V1`.

Ramey, V. A. and S. Zubairy (2018a). Government spending multipliers in good times and in bad: evidence from us historical data. *Journal of Political Economy 126*(2), 850–901.

Ramey, V. A. and S. Zubairy (2018b). Replication data for "Government spending multipliers in good times and in bad: evidence from US historical data". `https://econweb.ucsd.edu/~vramey/research/Ramey_Zubairy_replication_codes.zip`.