Data Appendix to

Bargaining over Babies: Theory, Evidence, and Policy Implications*

Matthias Doepke
Northwestern University

Fabian Kindermann
Universität Regensburg

April 2019

Abstract

This document describes how to obtain and process data from the "Generations and Gender Survey" in order to generate figures and regression tables in the paper. In addition, we show how to use data from the "International Social Survey Program" to plot Figure A.2 in the appendix. Finally, we briefly discuss our simulation model, how to run it and how to read the respective output files.

*Doepke: Department of Economics, Northwestern University, 2211 Campus Drive, Evanston, IL 60208 (e-mail: doepke@northwestern.edu). Kindermann: Department of Economics, University of Regensburg, 93040 Regensburg, Germany (e-mail: fabian.kindermann@ur.de).

1 Analysis using GGP Data

We use micro data from the ”Generations and Gender Survey” in order to generate Figures 1-3 and Figure 5 in the paper as well as Figure A.1 in the online appendix. In addition, we use the data to get to Tables 1-6 in the paper as well as Table A.1 to A.7 in the online appendix.

The ”Generations and Gender program” provides the raw data through their website

`https://www.ggp-i.org`

Every person who wants to replicate our data analysis can register online on the website, file a data request and download the raw data. After approval of the registration, researchers should download the raw data files in Stata format. These files should be copied to the folder `GGP_analysis/81_RAW` in the data appendix that accompanies this readme file. Table 1 summarizes the Stata files that are necessary to execute the data analysis.

Table 1: Raw files for GGP data analysis

Wave 1 data files	*Wave 2 data files*
GGS_Wave1_Austria_V.4.3.dta	GGS_Wave2_Austria_V.1.3.dta
GGS_Wave1_Belgium_V.4.3.dta	
GGS_Wave1_Bulgaria_V.4.3.dta	GGS_Wave2_Bulgaria_V.1.3.dta
GGS_Wave1_CzechRepublic_V.4.3.dta	GGS_Wave2_CzechRepublic_V.1.3.dta
GGS_Wave1_France_V.4.3.dta	GGS_Wave2_France_V.1.3.dta
GGS_Wave1_Germany_V.4.3.dta	GGS_Wave2_Germany_V.1.3.dta
GGS_Wave1_Lithuania_V.4.3.dta	GGS_Wave2_Lithuania_V.1.3.dta
GGS_Wave1_Norway_V.4.3.dta	
GGS_Wave1_Poland_V.4.3.dta	
GGS_Wave1_Romania_V.4.3.dta	
GGS_Wave1_Russia_V.4.3.dta	GGS_Wave2_Russia_V.1.3.dta

Once the raw micro data files have been copied to the folder `GGP_analysis/81_RAW`, the full data analysis can be carried out. We coded each graph and table into a separate Stata `.do` file. These files are organized in four different folders. The files in folder

1_retrieve are used to merge, recode and select the raw data in order to construct the datasets needed for the data analysis. The files in folder 2_analysis can be used to generate the figures and regression tables in Section 2 of the paper. The files in folder 3_calibration deliver the tables and figures in the calibration section (Section 5). The files in folder 4_appendix can be used to reconstruct the figures and tables in the online appendix. All figures Stata generates are stored in 91_graphs, all regression tables in 92_texres.

The file 0_run.do manages the execution of the full data analysis. The file assumes that the Stata working directory is the folder in which the file 0_run.do is located. If you run this do file, all graphs and tables (except for Figure A.2) will be generated. All files contain detailed comments on what is done in them, so they should be self-explanatory. Note that the Stata package estout is needed to export the regression tables. If you first run the file 0_run.do, Stata will automatically install this package.

2 Figure A.2 on time use and disagreement

In order to construct Figure A.2 in the online appendix, we used a different data set from the ”International Social Survey Program”. The Stata code to replicate the figure is contained in the folder `ISSP_time_use`. In order to run this code, please download the raw data from

`http://dx.doi.org/10.4232/1.12661`

In the section `Data & Documents`, just click on the file `ZA5900_v4-0-0.dta.zip` and extract the file `ZA5900_v4-0-0.dta` directly into the folder `ISSP_time_use`. Note that an online registration is necessary in order to download the raw dataset. The Stata file `figure_A2.do` can then be used to directly re-draw Figure A.2 in the online appendix.

3 Simulation model

Our simulation model is written in Fortran. A Fortran development environment can be downloaded directly from

`https://www.ce-fortran.com`

for different operating systems.

The complete simulation model can be found in the folder `simulation_model`. The source files (ending with `.f90`) are all stored in the sub-folder `src`. Table 2 summarizes what each of the source files is good for. The main program of interest is the file `fertdec.f90`. The lines 22-24 in this program contain a couple of "switches" that can be turned on and off in order to run different parts of our simulation model. By default, the variable `estimate` and `simulate_new_shocks` are set to a value of `.false.` and `policy` is set to `.true.`. Under this configuration, the model simulates our benchmark outcomes reported in Section 5 of the paper and performs a policy analysis to arrive at Figures 6 and 7.

Executing the simulation model The Fortran files need to be compiled in a certain order for the program building process to be successful. To manage the compilation process, we provide a file `makefile` that contains all necessary compiler information. For the building process to work, you need to install GNU's `make` environment, see `https://www.gnu.org/software/make/`. On macOS, this can be done for example via `homebrew`, on Windows machines using `cygwin`. Depending on the Fortran compiler used, the variable `f90comp` in the `makefile` can be adjusted. By default, it is set to GNUs `gfortran` compiler, which will be installed on your system if you obtained Fortran from the link provided above. In order to compile and run the program, simply open a terminal (or command prompt on Windows machines) and navigate to the folder where the file `makefile` is located. Then type the command

```
make run
```

into the terminal and the simulation model is compiled and executed.

Table 2: Source files of the simulation model

File	*Purpose*
`fertdec.f90`	This is the main program file. It manages the execution of the single program modules and sets all relevant model parameters. In addition, it can be used to simulate a new set of shocks for the model simulation.
`globals.d90`	The global variable file in which all variables are stored that need to be accessed by different modules of the simulation model.
`model.d90`	The code in this file derives partners' optimal decisions for a given set of parameters and policies and simulates the model for a certain set of shocks.
`policy_analysis.d90`	This module manages the evaluation of different policies. The results of this evaluation process are reported in Figure 6 and 7 of the paper.
`estimation.d90`	With this model, one can re-calibrate the parameters of the model to a different set of targets. The calibration targets can be adjusted in the subroutine `setup_targets()`. Detailed comments guide through this process.
`output.d90`	This module contains all output printing subroutines.
`wage_distribution`	The code in this module generates a discretized distribution of female wages.
`toolbox.d90`	This toolbox contains ample routines for numerical purposes.
`simulated_annealing.d90`	We use a parallelized simulated annealing method in order to calibrate parameters of the model. This routine turns out very efficient in finding a global solution to the calibration process. The module `estimate.f90` draws on this simulation module.

Simulating new shocks Setting `simulate_new_shocks` to `.true.`, the model will draw a new series of shocks for the simulation of the model. The series of shocks we used for simulation is stored in the file `random.inp` in the folder `input`. Once a new series of shocks is drawn, these new shocks are stored in the file `random.inp` and the program stops. In order to simulate our model with the new series of shocks, again set `simulate_new_shocks` to `.false.` and re-run the program.

Re-calibrating the model In order to re-calibrate the model with a new set of targets, set the variable `estimate` to a value of `.true.`. Note, however, that this re-calibration might take a long time. It best works on a computer with several cores. We used a standard machine with 8 workers. Yet, in principle the re-calibration process could be done on any machine.

Model output The model generates several output files, stored in the folder `output`. Table 3 summarized their content.

Table 3: Output files

File	*Purpose*
`res_output.tex`	Contains the data of Tables 4 to 8 of the paper in latex format.
`res_output_x.out`	Detailed output on agreement and disagreement, labor force participation, total fertility rate and persistence for the total population. File `x` = `all` contains information for total population, files `x` = `hs` and `x` = `col` only for high school workers and college graduate, respectively.
`res_policy.out`	The effectiveness of different policies to foster child bearing. These data are used for creating Figures 6 and 7 in the paper, see below.
`estim_parameters.out`	A summary of the parameters used. In case of a re-calibration of the model, this file will contain the final calibrated parameters.
`estim_targets_x.out`	A summary of the calibration targets. File `x` = `all` contains information for total population, files `x` = `hs` and `x` = `col` only for high school workers and college graduate, respectively.
`estim_values.out`	Prints the difference between calibration targets and model simulated output values.

Re-drawing Figures 6 and 7 Figures 6 and 7 can be re-constructed using the two Matlab source files in the folder `matlab`. These source files read the data from the file `res_policy.out` and plot the respective figures. Note that an active license of Matlab (`https://www.mathworks.com`) is needed to generate these plots.