README

Reproducibility Package for

" Survey-to-Survey Poverty Monitoring under Economic Shocks:

The Role of Proxies and Model Flexibility"

1. Overview

This reproducibility package contains all code and instructions needed to replicate the following outputs from the paper:

- Table 1 — Simulation results: model with proxy Z (MI-based poverty estimation)
- Table 2 — Simulation results: with and without proxy Z, including structural change
- Table 3 — Simulation results: nonlinear models (OLS vs. XGBoost with varying nonlinearity)
- Table 4 — Comparison of poverty estimates with and without proxies (Afghanistan, Rwanda, Uganda)
- Annex 7 — The role of consumption dummies as a proxy (egg/milk simulation)
- Annex 8 — Extended Table 4 results with standard errors and OLS coefficients

Tables 1, 2, 3, and Annex 7 are based on Monte Carlo simulations and do not require external data. Table 4 and Annex 8 use household survey microdata from Afghanistan, Rwanda, and Uganda.

2. Data

2.1 Data and Starting Point

All analyses in this paper start from processed household survey datasets constructed within the World Bank's Poverty and Equity Global Practice (including contributions from Poverty Economists and the SWIFT team). These datasets serve as the starting point for the reproducibility package for all countries (Afghanistan, Rwanda, and Uganda).

The processed datasets contain harmonized variables and constructed welfare aggregates derived from underlying household survey microdata using standardized procedures implemented within the Poverty and Equity GP.

This reproducibility package focuses on reproducing all analytical results (tables and figures) from these processed datasets. Reconstruction of these datasets from raw survey

microdata is not required to reproduce the results and is not fully included for all countries in the current package.

For Uganda (UGA), the package includes scripts that reconstruct the processed dataset from the underlying survey microdata, providing additional verification of the data construction process. For Afghanistan (AFG) and Rwanda (RWA), reconstruction from raw data is not included due to data access constraints and limited availability of upstream processing information.

2.2 Data Availability Statement

Original Data Sources:

1. National Risk and Vulnerability Survey 2011-2012 (Afghanistan)

   https://catalog.ihsn.org/index.php/catalog/5230

2. Living Conditions Surveys 2016-17 (Afghanistan)

   https://catalog.ihsn.org/index.php/catalog/8014

3. Rwanda - Integrated Household Living Conditions Survey 2010-2011

   https://microdata.statistics.gov.rw/index.php/catalog/36

4. Uganda National Household Survey 2019/2020

   https://www.ubos.org/about-us/guidelines-on-usage-and-reproduction-of-data/

5. Rwanda - Integrated Household Living Conditions Survey 2005-2006

   https://microdata.statistics.gov.rw/index.php/catalog/39

All analyses in this paper start from processed household survey datasets constructed within the World Bank's Poverty and Equity Global Practice (including contributions from Poverty Economists and the SWIFT team). These datasets serve as the starting point for the reproducibility package for all countries (Afghanistan, Rwanda, and Uganda).

The processed datasets contain harmonized variables and constructed welfare aggregates derived from underlying household survey microdata using standardized procedures implemented within the Poverty and Equity GP.

This reproducibility package focuses on reproducing all analytical results (tables and figures) from these processed datasets. Reconstruction of these datasets from raw survey

microdata is not required to reproduce the results and is not fully included for all countries in the current package.

For two countries (Rwanda and Uganda), the package includes scripts that reconstruct the processed datasets from the underlying survey microdata, providing additional verification of the data construction process. For Afghanistan (AFG), reconstruction from raw data is not included due to data access constraints and limited availability of upstream processing information

Dataset Name	File Name(s)	Link/Access Instructions	Access Date	Data Access
Afghanistan National Risk and Vulnerability Survey (NRVS) 2011-2012 Household processed data	dataset_2011_12_swift.dta	The World Bank Group	2014	Restricted
Afghanistan Living Conditions Surveys 2011-12 Household processed data	dataset_2016_17_swift.dta	The World Bank Group	2014	Restricted
Rwanda Integrated Household Living Conditions Survey (EICV), 2005/06 and 2010/11				
Household processed data	RW_Dataset.dta	The World Bank Group	2013	Restricted
Consumption and poverty processed data	rw_povdata_2006_11.dta	The World Bank Group	2020	Restricted
Rwanda Integrated Household Living Conditions Survey (EICV), 2010/11, Non-food consumption module	S09A1_consnonfood.dta	The World Bank Group	2020	Restricted
Uganda National Household Survey 2019/2020				

Front page	GSEC1.dta	The World Bank Group	2024	Restricted
HH Roster data	GSEC2.dta			
HH Members Info	GSEC3A.dta			
Education module	GSEC4.dta			
Consumption module	GSEC6A_2.dta			
Non-food module	GSEC6c.dta			
Finance	GSEC7_2.dta			
Dwelling characteristics module	GSEC9.dta			
Household member characteristics module	GSEC12.dta			
Consumption aggregate and poverty line data	indv_2020.dta			

3. Statement about Rights

I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript. Permission to redistribute is not granted beyond the authors. Replication verification therefore requires confidential agreement.

4. Software Requirements

Stata

Stata version 18 or later is required for the Stata do-files. The following Stata built-in commands are used: mi impute regress, mi estimate, svy, stepwise, putexcel. No additional user-written Stata packages (ado-files) are required.

R

R version 4.0 or later is required for Table 3. The following R packages must be installed:

- xgboost (version 3.2.0.1 recommended for reproducibility)
- writexl
- MASS (included with base R)

Install with: install.packages(c("xgboost", "writexl"))

Python

Python 3.10 is required for the machine learning notebooks (Table 4 / Annex 8). TensorFlow 2.12 does not support Python 3.11 or later. The following Python packages are required (exact versions pinned in requirements.txt):

- pandas, version2.2.3
- numpy, version 1.23.5
- scikit-learn, version 1.2.2
- xgboost, version 3.0.0
- statsmodels, version 0.14.4
- tensorflow, version 2.12.0
- joblib, version 1.4.2
- matplotlib, version 3.10.8
- scipy, version 1.15.3

A requirements.txt file with pinned versions is provided in the package root. To install all dependencies automatically, run setup_env.py (see Section 6.2).

Jupyter

Jupyter Notebook or JupyterLab is required to run the .ipynb machine learning notebooks for Table 4 and Annex 8. Jupyter is included in setup_env.py and requirements.txt.

5. Folder Structure

The package is organized as follows (S2S_Shock_WP_pk is the root folder):

```
S2S_Shock_WP_pk/
├─ config.do # Stata configuration file defining project paths
├─ config.py # Python configuration file defining project paths
├─ README.docx # Documentation and replication instructions
├─ requirements.txt # Python packages required for ML notebooks
├─ setup_env.py # Script to create Python virtual environment (.venv)

├─ Data prep/ # Data preparation stage
│  ├─ Code/
│  │  ├─ Data preparation main.do # Master script that runs all country
data preparation
│  │  ├─ AFG_data_prep.do # Data preparation for Afghanistan
│  │  ├─ RWA_data_prep.do # Data preparation for Rwanda
│  │  └─ UGA_data_prep.do # Data preparation for Uganda
│  │
│  └─ Data/
│  └─ Datain/
│  ├─ AFG/ # Raw input data for AFG (user must supply)
│  ├─ RWA/ # Raw input data for RWA (user must supply)
│  └─ UGA/ # Raw input data for UGA (user must supply)
```

```
├─ Table1/ # Table 1 simulation
│  ├─ Code/
│  │  └─ Table1_simulation_mi.do
│  │
│  └─ Output/
│  └─ Table1_simulation_results.xlsx

├─ Table2/ # Table 2 simulation
│  ├─ Code/
│  │  └─ Table2_simulation_mi.do
│  │
│  └─ Output/
│  └─ Table2_simulation_results.xlsx

├─ Table3/ # Table 3 simulation
│  ├─ Code/
│  │  └─ simulation_table_3_new.R
│  │
│  └─ Output/
│  └─ Table3_results.xlsx

├─ Table4_Annex8/ # Table 4 and Annex Table A8 empirical analysis
│  ├─ Code/
│  │  ├─ Generate_Table4.do
│  │  ├─ Generate_Annex8.do
│  │  │
│  │  └─ {AFG,RWA,UGA}/
│  │  ├─ {country}_ML.ipynb
│  │  ├─ {country}_OLS.do
│  │  └─ {country}_MI.do
│  │
│  ├─ Data/
│  │  ├─ {AFG,RWA,UGA}/ # Cleaned datasets and prediction outputs
│  │  │
│  │  └─ Temp/
│  │  └─ {AFG,RWA,UGA}/ # Intermediate datasets generated during
estimation
│  │
│  └─ Output/
│  ├─ Table4.xlsx
│  ├─ Annex8.xlsx
│  │
│  └─ {AFG,RWA,UGA}/ # Estimation log files

└─ Annex7/ # Annex 7 simulation
├─ Code/
│  └─ annex7_simulation.do
│
```

```
└─ Manuscript/
└─ Improving_shock-responsiveness_codereview.docx
```

6. Initial Setup

6.1 Edit config.do

A config.do file is provided in the root of the S2S_Shock_WP_pk folder. It sets the global path macros used by all Stata do-files (Tables 1, 2, Table 4, Annex 7, and Annex 8). By default it sets the project root to the current working directory:

```
global dir "`c(pwd)'"
```

If you always cd to S2S_Shock_WP_pk before running any do-file, no edits are required. If you prefer a fixed absolute path, uncomment and edit the alternative line inside config.do:

```
global dir "C:/Users/YOUR_USERNAME/S2S_Shock_WP_pk"
```

config.do also creates the following globals: $code1, $rdata1, $cdata1 (data preparation); $code2, $pdata2, $output2 (Table 4 / Annex 8); $table1out, $table2out (simulation output); $annex7out (Annex 7 output). It ensures output folders exist and sets the working directory to $dir.

Note: Generate_Table4.do and Generate_Annex8.do will automatically search for config.do in both the current working directory and two levels up (i.e. from Table4_Annex8/Code). You can therefore either cd to the S2S_Shock_WP_pk root before running them, or open them directly from their Code folder. Table 3 (R script) does not use config.do; it resolves its output path relative to its own directory.

6.2 Edit config.py and run setup_env.py (required for Table 4 and Annex 8)

A config.py file is provided in the root of the S2S_Shock_WP_pk folder. It sets the path variables used by all Python Jupyter notebooks. Open config.py and change line 17 to your local path:

```
PROJECT_ROOT = Path(r"C:\Users\YOUR_USERNAME\path\to\S2S_Shock_WP_pk")
```

Then run the automated setup script from a terminal. Any Python 3.x version can run this script:

```
cd S2S_Shock_WP_pk
python setup_env.py
```

This script automatically: (a) finds or installs Python 3.10 (via winget on Windows), (b) creates a virtual environment (.venv) in each notebook directory (AFG, RWA, UGA), (c) installs all pinned packages from requirements.txt, and (d) registers a Jupyter kernel

named “Python 3.10 (S2S)”. The script prints a verification message when all imports succeed.

6.3 Confidential microdata (required for Data prep, Table 4, and Annex 8)

The analysis in this package does not require reconstruction from raw microdata to reproduce the reported results, as it starts from processed datasets described above. The household survey microdata used in this study are confidential and cannot be distributed with this replication package. The authors accessed these datasets under restricted conditions, and the exact versions used in the analysis are not publicly available.

Public-use versions of some of the datasets may be available through the World Bank Microdata Library (MDL). However, these datasets typically undergo anonymization and additional processing, and therefore may differ from the internal versions used in this study. As a result, running the replication code with publicly available microdata may not reproduce the exact numerical results reported in the paper.

Researchers who have authorized access to the same internal datasets may place the files in the corresponding folders under:

Data prep/Data/Datain/

and run the data preparation scripts to reproduce the analysis.

- Afghanistan (AFG): Afghanistan National Risk and Vulnerability Survey 2011-12 and Afghanistan Living Conditions Survey 2016-17 (ALCS). Place Stata files in Data prep/Data/Datain/AFG/.
- Rwanda (RWA): Rwanda Integrated Household Living Conditions Survey (EICV) 2005/06 and 2010/11. Place Stata files in Data prep/Data/Datain/RWA/.
- Uganda (UGA): Uganda National Panel Survey (UNPS) 2019/20. Place Stata files in Data prep/Data/Datain/UGA/.

7. Replication Instructions

The tables can be replicated independently. Tables 1, 2, 3, and Annex 7 use simulated data and can be run without external datasets. Table 4 and Annex 8 require the microdata described above.

7.1 Table 1 — Simulation with Proxy Z

Software: Stata 18+

Data required: None (uses simulated data)

Steps:

1. Open Stata and cd to the S2S_Shock_WP_pk root.

2. Run: do config.do

3. Run: do "Table1/Code/Table1_simulation_mi.do"

4. Output: Table1/Output/Table1_simulation_results.xlsx and console output. Use Table1 Sheet.

Description: This script simulates a DGP where Y = 2 + 3X + 1.5U + e, U = 1 + 1.5Z + e, with Z shifting from N(0,1) to N(2,1) in Period 3. It uses Stata's mi impute regress to draw from the posterior predictive distribution and applies Rubin's combining rules to estimate poverty rates across three periods. N = 10,000 observations, M = 10 imputations. Random seed: 42.

7.2 Table 2 — With and Without Proxy Z

Software: Stata 18+

Data required: None (uses simulated data)

Steps:

1. Open Stata and cd to the S2S_Shock_WP_pk root.

2. Run: do config.do

3. Run: do "Table2/Code/Table2_simulation_mi.do"

4. Output: Table2/Output/Table2_simulation_results.xlsx and console output. Use Table2 Sheet.

Description: Extends the Table 1 simulation to four periods and compares models with and without proxy Z. The output has three rows (Scenarios): Scenario A "No proxy (sorting instability)" — Period 3 without Z; Scenario B "Stable proxy restores invariance" — Period 3 with Z; Scenario C "Imperfect proxy (partial restoration)" — Period 4 with Z. In Period 3, Z shifts from N(0,1) to N(2,1) while b_u remains 1.5; in Period 4, Z shifts to N(2,1) and b_u changes from 1.5 to 2.0, introducing a structural change in the U-Z relationship. The script runs three separate MI procedures, one for each panel. N = 10,000, M = 10. Random seed: 42.

7.3 Table 3 — Nonlinear Models

Software: R 4.0+

Data required: None (uses simulated data)

Steps:

1. Open R or RStudio.

2. Ensure the xgboost, writexl, and MASS packages are installed.

3. Open and run S2S_Shock_WP_pk/Table3/Code/simulation_table_3_new.R.

4. Output: Table3/Output/Table3_results.xlsx.

Description: Introduces nonlinearity via a Box-Cox-like transformation Y = h(X + U; lambda) + e, with lambda in {0.001, 0.01, 0.1, 0.5}. Compares three approaches: (1) XGBoost without Z (two-stage calibration MI), (2) XGBoost with Z (two-stage calibration MI), and (3) OLS with Z (one-stage direct MI). Z shifts from N(0,1) to N(1,1) between periods. N = 10,000, M = 10. Random seed: 42.

7.4 Table 4 — Empirical Poverty Estimates

Software: Stata 18+, Python 3.10 with Jupyter (see Section 6.2 for setup)

Data required: Household survey microdata (see Section 6.3)

Steps:

1. Complete the initial setup (Section 6): edit config.do, edit config.py, run setup_env.py (explained in Section 6.2), and place microdata.

2. Run data preparation: In Stata, cd to the S2S_Shock_WP_pk root and run:
   do "Data prep/Code/Data preparation main.do"
   This calls AFG_data_prep.do, RWA_data_prep.do, and UGA_data_prep.do.

3. Run the three machine learning Jupyter notebooks to generate prediction CSV files. Activate the virtual environment, launch Jupyter, and select the “Python 3.10 (S2S)” kernel, then run all cells:
   cd Table4_Annex8\Code\AFG
   .venv\Scripts\activate
   jupyter notebook AFG_ML.ipynb
   Repeat for RWA (RWA_ML.ipynb) and UGA (UGA_ML.ipynb).
   In each notebook, go to Kernel > Change kernel > Python 3.10 (S2S), then Cell > Run All.
   Each notebook takes approximately 30–60 minutes depending on hardware.
   These notebooks train OLS, Random Forest, XGBoost, and Deep Learning models and save predicted values as CSV files in the Table4_Annex8/Data/ sub-folders.

4. Generate Table 4: In Stata, cd to the S2S_Shock_WP_pk root and run:
   do "Table4_Annex8/Code/Generate_Table4.do"
   This calls the 6 sub-scripts (3 OLS + 3 MI) and writes results to Table4_Annex8/Output/Table4.xlsx.

Description: For each country (Afghanistan 2011-2016, Rwanda 2005-2010, Uganda 2019-2020), the analysis (a) estimates OLS models via stepwise selection with and without proxy variables, (b) uses mi impute regress for multiple imputation of consumption, (c) computes poverty rates from imputed values, and (d) compares direct estimates, OLS, Random Forest, XGBoost, and Deep Learning approaches. The OLS do-files (AFG_OLS.do, RWA_OLS.do, UGA_OLS.do) handle model selection and

imputation. The MI do-files (AFG_MI.do, RWA_MI.do, UGA_MI.do) process the ML predictions through a calibration MI framework.

7.5 Annex 7 — Role of Consumption Dummies as Proxy

Software: Stata 18+

Data required: None (uses simulated data)

Steps:

1. Create config.do in the S2S_Shock_WP_pk root (see Section 6.1).

2. Open Stata and cd to the S2S_Shock_WP_pk root.

3. Run: do "Annex7/Code/annex7_simulation.do"

4. Output: Annex7/Output/annex7_results.log (console log with all results).

Description: Runs 500 Monte Carlo replications with 10,000 observations each. Simulates a DGP where an unobserved conflict variable affects consumption, and two observable food consumption dummies (egg, milk) serve as proxies. The simulation compares poverty imputation bias when using egg vs. milk as a proxy, with egg's correlation to the unobservable changing across periods while milk's remains stable. M = 10 imputations per replication. Random seed: 42. Expected runtime: several hours.

7.6 Annex 8 — Extended Results with Standard Errors

Software: Stata 18+, Python 3.10 with Jupyter (see Section 6.2 for setup)

Data required: Same as Table 4 (see Section 6.3)

Steps:

1. Complete all prerequisite steps for Table 4 (Steps 1-3 in Section 7.4).

2. In Stata, cd to the S2S_Shock_WP_pk root and run:
   do "Table4_Annex8/Code/Generate_Annex8.do"

3. Output: Table4_Annex8/Output/Annex8.xlsx with the following sheets:
   - Annex8_TableA: All poverty rates with standard errors
   - AFG_NoProxy_OLS, AFG_WithProxy_OLS: AFG OLS coefficients and variable means
   - RWA_NoProxy_OLS, RWA_WithProxy_OLS: RWA OLS coefficients and variable means
   - UGA_NoProxy_OLS, UGA_WithProxy_OLS: UGA OLS coefficients and variable means

Description: Generate_Annex8.do runs the same 6 sub-scripts as Table 4 but additionally exports standard errors for all poverty estimates and OLS regression coefficients with variable means for each country and model specification (with and without proxies).

8. Summary of Scripts and Outputs

Output	Script(s)	Software	Data	Approx. Runtime
Table 1	Table1/Code/Table1_simulation_mi.do	Stata	Simulated	< 1 min
Table 2	Table2/Code/Table2_simulation_mi.do	Stata	Simulated	< 1 min
Table 3	Table3/Code/simulation_table_3_new.R	R	Simulated	< 5 min
Table 4	Data prep scripts + ML notebooks + Generate_Table4.do	Stata + Python	Microdata	Python: ~4.5 h; Stata: ~30 min
Annex 7	Annex7/Code/annex7_simulation.do	Stata	Simulated	<20 min
Annex 8	Generate_Annex8.do (reruns Stata estimation used in Table 4)	Stata	Microdata	~1 hour

Note: Generate_Annex8.do reruns the same Stata estimation procedure used for Table 4 but does not execute the Python machine-learning notebooks. The script extracts additional statistics and intermediate results to produce the more detailed results reported in Table A8.

9. Expected Outputs

After running all scripts, the following output files will be generated:

- Table1/Output/Table1_simulation_results.xlsx
- Table2/Output/Table2_simulation_results.xlsx
- Table3/Output/Table3_results.xlsx
- Table4_Annex8/Output/Table4.xlsx
- Table4_Annex8/Output/Annex8.xlsx
- Annex7/Output/annex7_results.log

Note: Due to simulation randomness, results from Tables 1, 2, 3, and Annex 7 may differ slightly across runs despite the fixed random seed (42), depending on the Stata/R version and operating system. The qualitative findings should remain unchanged.

10. Additional Notes

Execution order: Tables 1, 2, 3, and Annex 7 can be run independently and in any order. For Table 4 and Annex 8, the execution order must be: (1) Data preparation, (2) ML notebooks, (3) Generate_Table4.do or Generate_Annex8.do.

Random seeds: All simulation scripts set a random seed of 42 at the beginning to ensure reproducibility. For Table 3 (R/XGBoost), per-iteration seeds are used to isolate the RNG state across lambda values, and seed = 42 is set in the XGBoost parameters. However, different versions of the xgboost R package may produce slightly different results due to changes in internal subsampling. To ensure exact reproducibility, use xgboost version 2.1.4.

Multiple imputation: The simulation scripts (Tables 1, 2, Annex 7) and the empirical analysis (Table 4, Annex 8) use Stata's mi impute regress command, which draws from the posterior predictive distribution of the regression model (perturbing both coefficients and residuals) and applies Rubin's combining rules.