Data code for: “Climate Shocks and Their Effects on Food Security, Prices, and Agricultural Wages in Afghanistan”

Tosin Gbadegesin, Bo Pieter Johannes Andree, and Ademola Braimoh

Instructions for Replicators:

To successfully reproduce the analysis, follow these steps:

1. Ensure you have the file **AFG Data.xlsx** in the directory
2. Open the **AFG Analysis Study.ipynb** notebook and execute all code cells up to line 200 where AFG Stata Analysis Final.xlsx is being generated.
3. Next, run the entire code in **AFG Paper Study.do** file in Stata which generated AFG Stata Analysis Result-09-27.csv file.
4. Return to the **AFG Analysis Study.ipynb** notebook and execute the code from line 206 to line 255.

Data and Code Availability Statement

The paper uses public and non-confidential data on food insecurity that was published by the Famine Early Warning Systems Network (FEWS NET). We downloaded the data on January 2024 from https://datacatalog.worldbank.org/int/search/dataset/0064614/harmonized-sub-national-food-security-data.

The paper uses public and non-confidential data on domestic food prices and the local agricultural wages that were published by Andrée (2023b; 2023c). We downloaded the Microdata on January 2024 from https://microdatalib.worldbank.org/index.php/catalog/12415; https://microdata.worldbank.org/index.php/catalog/4484.

The paper uses public and non-confidential data on administrative distribution, rainfall, and the normalized difference vegetation index (NDVI) that was published by the Humanitarian Data Exchange. We downloaded the Microdata on January 2024 from https://data.humdata.org/dataset/afg-ndvi-subnational, https://data.humdata.org/dataset/afg-rainfall-subnational, and https://data.humdata.org/dataset/cod-ab-afg.

The paper uses public and non-confidential data on exchange rate and fuel price data that was published by Andrée (2023b; 2023c). We downloaded the Microdata on January 2024 from https://microdatalib.worldbank.org/index.php/catalog/14848; https://microdatalib.worldbank.org/index.php/catalog/14847.

The paper uses public and non-confidential data on global food prices that was published by World Bank Food Price Monitor. We downloaded the Microdata on January 2024 from https://www.worldbank.org/en/research/commodity-markets.

The paper uses public and non-confidential data on population, area, terrain ruggedness, cropland, and pastureland, that was published by Andrée (2020). We downloaded the Microdata on January 2024 from https://microdata.worldbank.org/index.php/catalog/3811.

Computational Requirements

Software and Hardware Requirements

- **Software**:
    - **Python**: We used Jupyter Notebook from Anaconda Distribution 2023.03. This version uses the base environment Python 3.10 and offers Python 3.8 and Python 3.9 support. It also includes conda 23.1.01 and Anaconda Navigator 2.4.01.
    - **Stata**: We used Stata 17 for our regression analysis.
    - **Packages**: Numerous packages are required, and they are all specified at the beginning of each code script.
- **OS**:
    - **Windows 11 Pro**: This was our primary operating system. Other versions of Windows, as well as Mac and Linux, should be compatible.
- **Hardware**:
    - **Model:** Dell Latitude 7420
    - **CPU**: 11th Gen Intel(R) Core(TM) i5-1145G7 @ 2.60GHz   1.50 GHz
    - 16.0 GB / 64-bit operating system, x64-based processor

Downloading and opening the replication files

Link to the data and replicate code is available at: https://github.com/TosinSDGs/AFG-Climate-Analysis

Data Preparation Summary

All the datasets used in this analysis are manually added to the *AFG Data.xlsx* Excel document, with each dataset stored in a specific sheet as follows:

- Rainfall data is manually added to the sheet named *rainfall*.
- NDVI data is manually added to the sheet named *ndvi*.
- Administrative data is manually added to the sheet named *ADM2*.
- Domestic food prices are manually added to the sheet named *FoodP*.
- Exchange rates and agricultural wage are manually added to the sheet named *exchangeR*.

- Fuel prices are manually added to the sheet named *FuelP*.
- Global food prices are manually added to the sheet named *GFoodP*.
- IPC data is manually added to the sheet named *AFG Others*.
- Data on cropland, ruggedness, pastureland, area and population are manually added to the sheet named *foodcrises*.

List of Exhibits:

The provided code reproduces:

Exhibit	Output filename	Script	Note
Table 1	AFG Data for Maps and Tables.xlsx	AFG Analysis Study.ipynb	Sheet name: Descriptive Stats
Figure 1	food_insecurity_plot.png	AFG Analysis Study.ipynb	
Figure 2	prices_wages_plot2.png	AFG Analysis Study.ipynb	
Figure 3	rainfall_ndvi_plot.png rainfall_ndvi_anomalies_plot.png	AFG Analysis Study.ipynb	
Table 2	AFG Data for Maps and Tables.xlsx	AFG Analysis Study.ipynb	Sheet name: Table 2_FStress
Table 3	AFG Data for Maps and Tables.xlsx	AFG Analysis Study.ipynb	Sheet name: Table 3_FCrisis
Table 4	AFG Data for Maps and Tables.xlsx	AFG Analysis Study.ipynb	Sheet name: Table 2_FStress
Table 5	AFG Data for Maps and Tables.xlsx	AFG Analysis Study.ipynb	Sheet name: Table 3_FCrisis
Figure 6	food_ stress_plota.png food_crisis_plota.png	AFG Analysis Study.ipynb	

Using Python (code in file "AFG Analysis Study.ipynb"), these datasets were processed and integrated to produce the final cleaned dataset, stored in AFG Final Cleaned Data.xlsx. From this output, the file was reloaded, and the final processing was conducted to create the dataset "AFG Stata Analysis Final.xlsx", which serves as the input for the regression analysis presented in this study. The Python code used for this processing includes adequate comments to ensure clarity and reproducibility, with descriptive explanations for each step. Additionally, break comments were added separate the first preprocessing steps from the next.

Some Final Processing Description (Python):

First, reload the data from ' AFG Final Cleaned Data.xlsx' into a DataFrame named afg, setting the foundation for all subsequent data manipulations. Calculate the vanilla z-scores for rainfall and NDVI as described above. Create dummy variables for positive and negative z-scores (positive_raina and negative_raina) and calculate their products with the z-scores, resulting in two additional columns: positive_rainfa and negative_rainfa. Next, follow similar process for NDVI. Compute the percentage changes for variables such as 'avgfuelP', 'avgexchangeR', 'avgWageNonQL', 'avgWageQL', 'aveWageQLNL', 'avgfood', and 'GFoodPrice' within each 'mkt_name' group, prefixing the new columns with 'p_'. Create dummy variables based on the 'IPC' column to indicate whether IPC values are greater than or equal to 2 (IPC_dummya) and 3 (IPC_dummyb). Finally, save the updated afg_f DataFrame to an Excel file named 'AFG Stata Analysis Final.xlsx'.

Food Insecurity
We obtained historical food insecurity data from assessments conducted across 34 districts in Afghanistan from 2009 to 2023 via the Famine Early Warning Systems Network (FEWS NET). FEWS NET uses the Integrated Food Security Phase Classification (IPC) system, which categorizes food insecurity into five phases: minimal, stressed, crisis, emergency, and famine. Since 2012, FEWS NET data has identified areas where humanitarian aid reduced the IPC phase by one. We created a new IPC rating by combining FEWS NET IPC Phase data with a binary indicator for humanitarian aid presence, offering insights into potential IPC phases without aid. This method focuses solely on food insecurity, excluding the effects of humanitarian response. Two dummy variables were then generated: the first equals 1 when the new IPC rating is ≥ 2, and the second equals 1 when IPC is ≥ 3.

Rainfall and NDVI
Rainfall and NDVI data, aggregated monthly at the administrative division level 2, are analyzed to detect anomalies using Z-scores, calculated by standardizing the values based on their mean and standard deviation. Positive and negative rainfall anomalies represent flood and drought risks, respectively, with dummy variables flagging these conditions. The magnitude of each anomaly is obtained by multiplying the Z-scores with the corresponding dummy variables. A similar process is applied to NDVI data, where positive anomalies indicate healthier vegetation (potential flood risk), and negative anomalies suggest drought or vegetation stress.

Agricultural wage, exchange rate, domestic food price, and fuel price
Data on agricultural wages, exchange rates, domestic food prices, and fuel prices are organized by market name, and then aggregated at the market level. These variables, which include monthly open, close, high, and low values, are averaged to obtain the mean agricultural wage, exchange rate, domestic food price, and fuel price. Additionally, percentage changes in these variables are used to capture their dynamic interactions with other factors.

Population density, terrain ruggedness, cropland, and pastureland

Population density, terrain ruggedness, cropland, and pastureland are variables that remain constant over time, with data available up to 2020. Population density is calculated by dividing the population by the area. Due to their static characteristics, these variables were extended to 2023 using forward-filling techniques and were then logarithmically transformed.

Logit and VAR Regression (See the Stata do file)
To begin, set the working directory to where your datasets are located. This step ensures that all subsequent file paths are relative to this directory, simplifying the file access process. Next, load the main dataset obtained from Python (AFG Stata Analysis Final.xlsx) into Stata. After importing the datasets, sort them by 'adm2_name' and 'dates' to ensure the data is properly ordered for time series analysis. Following this, generate a time trend variable, which will be used for time series operations, and set the dataset as time series.

Table 2
For the food stress analysis, perform logistic regression (logit) to model the probability of food stress (IPC_dummya). This regression includes several independent variables such as positive_rainfa, negative_rainfa, vim_zscore, various percentage change variables (e.g., p_avgWageQL, p_avgfood), and seasonal dummy variables (winter, summer, spring). Robust standard errors (vce(robust)) are used to account for potential heteroskedasticity, ensuring more reliable statistical inferences. After running the logistic regression, use the predict command to calculate the predicted probabilities of food stress and store them in a new variable logit1a, representing the estimated likelihood of food stress for each observation in the dataset based on the specified model.

To evaluate the model's performance, use the estat gof command to assess the goodness of fit and estat classification to evaluate the classification accuracy of the model. For a comprehensive analysis, use the nlcom command to compute the total effect of various variables such as positive_rainfa, negative_rainfa, vim_zscore, p_avgfood, and p_avgWageQL, summing the coefficients of the current and lagged values (up to 12 lags) of each variable to understand their cumulative impact. Calculate the overall marginal effects of all included variables using the margin, dydx(*) command, and specifically compute the marginal effects for climate-related variables (positive_rainfa, negative_rainfa, vim_zscore, p_avgWageQL, p_avgfood) to understand their specific impact on food stress. Repeat a similar process to analyze food crisis and obtain logit1b using IPC_dummyb as the dependent variable. Then, export the dataset which now has column logit1a and logit1b as csv ('AFG Stata Analysis Result-09-27.csv') to further analysis in Python.

Table 3

For analyzing food prices while still using AFG Stata Analysis Final.xlsx dataset, use the VAR command to run a Vector Autoregression (VAR) model with the percentage change in average food prices (p_avgfood) as the dependent variable, including lagged values and other independent variables such as average wages, rainfall anomalies, NDVI anomalies, fuel prices, global food prices, exchange rates, interaction terms, humanitarian dummy, time trend, and control variables like cropland percentage, ruggedness mean, population density, pasture percentage, and seasonal dummy variables.

Next, calculate the total effect of the explanatory variables over time using the nlcom command, summing the coefficients of the current and lagged values (up to 12 lags) for variables such as positive and negative rainfall anomalies, NDVI anomalies, average food prices, and average wages. Similarly, for analyzing agricultural wages, run another VAR model with the percentage change in average wages (p_avgWageQL) as the dependent variable, using the same independent variables. Calculate the total effect of these variables on average wages with the nlcom command, summing the coefficients of the current and lagged values. This cumulative impact analysis helps understand the influence of these variables on agricultural wages over time.

Table 4&5; Figure 4 & 5
First, load the data exported from Stata where the prediction for food stress (logit1a) and food crises (logit1b) were stored, 'AFG Stata Analysis Result-09-27.csv' into a DataFrame named Obj1 (code in file "AFG Analysis Study.ipynb") in Python.

The process begins by grouping the data by adm1_name to calculate the mean values of logit1a and logit1b, which are stored in a new DataFrame called logit1. Population data, including male and female populations for each province, is then merged with the logit1 DataFrame to create a combined dataset. For Table 4, the logit1a values are categorized into bins ranging from 0.31 to 0.71 and above. The data is grouped by these bins, and the number of provinces, as well as the total, male, and female populations, are summed for each bin to produce a summary of food stress probabilities. Population data here are scraped from AFG Pop.pdf (these two pages are extracted from the whole document "NSIA-Report on Population (2021-2022).pdf"). Similarly, for Table 5, logit1b values are categorized into bins ranging from 0 to 0.31 and above. The data is again grouped by these bins to generate a summary of food crisis probabilities. Both results are written to an Excel file (AFG Data for Maps and Tables.xlsx). Sheet "Table 2_FStress" and "Table 3_FCrisis" has data for Table 4 and 5, while sheet "Descriptive Stats" has Table 1. Data in sheet named "Map Data" was used for creating Figure 4 and 5 in Excel, the Map for food stress and food crises, respectively using the Excel map function.

Figure 1
To create Figure 1, the process begins by making a copy of the original DataFrame (Obj1), selecting relevant columns such as date, adm1_name, adm2_name, pop, and IPC. The date column

is converted to datetime format to ensure correct handling of time data. Next, the total population is calculated by averaging the population values across each adm1_name group and summing the resulting averages to obtain the total population (total_pop_f). Proportions of the population in IPC phases >= 2 (food stress) and >= 3 (food crisis) are then computed and added as new columns (Proportion_IPC_>=2 and Proportion_IPC_>=3). The DataFrame is grouped by date and adm1_name, calculating the mean values for IPC and the population proportions, and sorted for clarity. Further aggregation by date sums the population proportions across all administrative regions, and the IPC values are rounded to the nearest whole number. Finally, a line plot is generated to display the proportion of the population in food stress and food crisis over time, with y-axis labels formatted in thousands and x-axis tick marks set at six-month intervals. The plot is saved as a PNG file and shown, clearly illustrating the trends in food insecurity.

Figure 2

To create Figure 2, the process begins by loading two datasets from Excel: one for food prices and another for wages. For food prices, multiple columns representing different commodities (bread, rice, and wheat) are averaged to create category-specific averages, and these are further averaged to calculate an overall avg_food column. For wages, separate columns for qualified and non-qualified labor wages are averaged to create new columns representing avgWageQL and aveWageQLNL (an overall average of the two). Rows containing only "Market Average" for each market are kept, and both datasets are filtered and merged based on mkt_name and price_date. The price_date column is converted to datetime format, and the data is filtered to include only entries between July 2009 and December 2023. The merged dataset is then used to create two side-by-side subplots: one for Domestic Food Price and the other for Agricultural Wages. The x-axis is formatted to display dates at monthly intervals, and y-axis labels are set to represent monthly prices and wages, respectively. The figure layout is adjusted to prevent overlap, saved as a PNG file named 'prices_wages_plot2.png', and displayed for visualization.

Figure 3a and 3b

To create Figures 3a and 3b, the relevant columns from the dataset (Obj1) were selected, including date, rfhmean (mean rainfall), vimmean (mean NDVI), and their Z-scores (rainfall and NDVI anomalies). These columns were renamed for clarity, and the date column was converted to datetime format. The data was then aggregated by date to calculate mean values for each variable. For Figure 3a, a side-by-side subplot layout was generated. The first subplot shows the mean rainfall over time in blue, with the y-axis labeled "Monthly Rainfall (mm)." The second subplot represents mean NDVI values over time in green, with the y-axis labeled "Monthly NDVI." Figure 3b, also presented as a side-by-side layout, shows anomalies. The first subplot displays rainfall anomalies in blue with a dashed line, and the second subplot shows NDVI anomalies in green with a dashed line. After adjusting the layout to prevent overlap, each figure is saved as a PNG file, named 'rainfall_ndvi_plot.png' for Figure 3a and 'rainfall_ndvi_anomalies_plot.png' for Figure 3b.

Figure 6a
To create Figure 4, the process starts by making a copy of relevant columns from the original dataset (Obj1), including date, pop, IPC, logit1a (Food Stress), and logit1b (Food Crisis). The columns are renamed, and the date column is converted to datetime format. The proportions of the population in IPC phases >= 2 and >= 3 are calculated and stored in new columns. These values are aggregated by date to compute the mean values for each day. Next, modeled IPC populations are calculated by multiplying population values with their respective probabilities of food stress and food crisis. Linear regression is performed without a constant to estimate the historical and future population impacts for food stress, and the results are stored in a DataFrame. A figure is then created with two subplots placed side by side: the first subplot shows historical and estimated food stress, while the second subplot combines the estimated population at risk with food stress probability. Y-axis labels are formatted in thousands for population and as proportions for the secondary axis, and the figure is saved as a PNG file.

Figure 6b
The process for Figure 5 mirrors that of Figure 4 but focuses on food crisis. Relevant columns (date, pop, IPC, logit1a, and logit1b) are copied from the original dataset, and the proportion of the population in IPC phases >= 3 is calculated. After aggregating the data by date, modeled IPC populations for food crisis are computed. Linear regression without a constant is performed to estimate historical and future population impacts, and the results are saved in a DataFrame. The figure consists of two side-by-side subplots: the first shows historical and estimated food crisis, while the second displays the estimated population at risk alongside food crisis probability. Y-axis labels are formatted in thousands for population, and the secondary axis is formatted as proportions. The figure is then saved as a PNG file for further use.