Armenia CCDR Microsimulation

Renato Vargas Julie Rozenberg Colin Lenoble
Natsuko Kiso Nozaki Thomas Farole

Table of contents

0.1 Introduction

In this calculation file, we "age" the household survey according to demographic projections and different macroeconomic scenarios to explore the impact of climate-related risks and policy measures on the consumption expenditure distribution.

As a convention, code is presented in the following format in this guide:

```
# Some comment that is not evaluated by R
some_variable <- some_function(some_object, some_parameter = TRUE)
```

We assume that the reader has created an Rstudio project and is familiar with basic R functions. Within that project we recommend the following file structure:

```
root/
   scripts
      my_script.R
   data/
|      my_data.sav
```

```
|      my_data.dta
|      my_data.csv
   output
       my_output1.csv
       my_output2.xlsx
```

Using RStudio project makes it possible to not use `setwd()` to establish the root directory and refer to subdirectories in a relative manner, making interoperability easier within teams and not hard coding a particular computer's file structure into the code. If you are not using RStudio, just add `setwd(r'(C:\My\path\to\project\root)')` at the beginning of your coding session.

0.2 Preamble

0.3 Preamble

We start with a clean environment, making sure that any objects from a previous session are not present. We take this opportunity to keep our country ISO code in a variable `iso` in case we need it later.

```
# Clean workspace
rm(list = ls())

# Armenia country ISO code
iso <- "ARM"

# Exchange rate USD per dram
er <- 0.002310
```

We call the appropriate libraries.

Rather than calling our libraries as we go, we will make sure we have everything we need from the beginning.

```
# Load packages
library(tidyverse) # includes dplyr, ggplot2 and others
library(haven)     # to read SPSS and Stata datasets
library(readxl)    # to read from MS-Excel
library(openxlsx)  # to write to MS-Excel.
library(gt)        # pretty tables
library(car)       # Companion to applied regression
```

```
library(modelr)     # regression models
library(janitor)    # pretty subtotals
library(purrr)      # map vectors (aggregation)

# Geopackages
library(sf)         # to read and write shapefile maps
library(terra)      # to perform geocalculations
library(tmap)       # for static and interactive maps
```

0.4 Datasets

We then load the datasets that we need for this study. We are lucky that the World Bank has processed some of these already for poverty analysis and so we have the original SPSS datasets with all variables for Households `hh` and for Individuals `pp`, as well as a consumption aggregate `ca` and a household income `ic` dataset, which are Stata datasets. This is for the year 2022. These are imported using the `haven` package. These are based on Armenia Integrated Living Conditions Survey 2022 (ARMSTAT, 2023).

```
# Original SPSS datasets
# Households (hh)
hh <- read_sav(
  "data/ARM-HH-survey/original-spss-files/ILCS-ARM-2022-Households.sav")
# Persons (pp)
pp <- read_sav(
  "data/ARM-HH-survey/original-spss-files/ILCS-ARM-2022-Persons.sav")

# Processed WB datasets
# Consumption aggregate at household level (ca)
ca  <- read_dta("data/ARM-HH-survey/CONSAGG2022.dta")
# Processed income at household level (ic)
ic  <- read_dta("data/ARM-HH-survey/totinc.dta")
```

We will work non-destructively, meaning we will not rewrite these data sets and we will only create intermediate data frame objects from them to perform transformations, selections and other data management tasks. For example, we will keep household assignment to poverty status and consumption deciles handy by creating a subset of our `ca` data with only our household identifiers, deciles, and poverty.

```
# From the WB processed dataset, we extract deciles and poverty
deciles <- ca |>
```

```
  select( hhid, decile, poor_Avpovln2022,
          poor_Foodpovln2022, poor_Lpovln2022, poor_Upovln2022)
```

Our population data comes from UN's projections.

```
# Population projections UN 2022
pop_proj <- read_dta("data/UN2022_population.dta") |>
  filter(country == iso)
```

We also have geographical information for level 1 in Shapefile format, which we import with the `sf` package. We rename the column with the name of the administrative region to match our household survey data set conventions to ease mergers. The `dplyr` package from the `tidyverse` meta package allows us to "pipe" or link processing steps using the `|>` pipe, which can be inserted using **Ctrl + m.** Although there is no geoprocessing in this analysis, this will come in handy for graphical presentations. Let's have a look at it.

```
# Geodata
# Armenia marzes or administrative level 1 shapefile
adm1 <- read_sf("data/ARM-Geodata/ARM-ADM1.shp") |>
  select(NAM_1, COD_HH_SVY, geometry) |>
    # Make sure that names match the rest of datasets
  mutate(NAM_1 = if_else(NAM_1 == "Gergharkunik", "Gegharkunik", NAM_1))
names(adm1)[2] <- "hh_02"

tm_shape(adm1)+
  tm_polygons("NAM_1", legend.show = FALSE) +
  tm_text("NAM_1", size = 3/4)
```

Marzes names are more accurate in the shapefile than in the survey. We will use them from here on instead of the survey factor labels.

```
hh <- hh |>
  left_join(adm1, join_by(hh_02 == hh_02)) |>
  select(-geometry)

ic <- ic |>
  left_join(adm1, join_by(hh_02 == hh_02)) |>
  select(-geometry)
```

Finally, but not least important, we have our vulnerability information.

```
buildings_aal <-
  read_xlsx("data/ARM-Vulnerability-Analysis/Data_AAL_AAE.xlsx",
            sheet = "Building_AAL") |>
    # Make sure that names match the rest of datasets
  mutate(NAM_1 = if_else(NAM_1 == "Gergharkunik", "Gegharkunik", NAM_1))
buildings_1in100 <-
  read_xlsx("data/ARM-Vulnerability-Analysis/Data_AAL_AAE.xlsx",
            sheet = "Building_1in100")
crops_productivity <-
  read.csv("data/ARM-Vulnerability-Analysis/ARM_crops_combined_REF_shock_admin1.csv") |>
```

```
  rename(NAM_1 = Province)
crops_aal <-
  read_xlsx("data/ARM-Vulnerability-Analysis/Data_AAL_AAE.xlsx",
            sheet = "Agriculture_AAL")
crops_1in100 <-
  read_xlsx("data/ARM-Vulnerability-Analysis/Data_AAL_AAE.xlsx",
            sheet = "Agriculture_1in100")
```

0.5 Data preparation income outliers and missings

0.5.1 1.1 Household consumption aggregates and characteristics

This section was done outside the text environment, so for clarification, please read the inline comments at this time. It will be fixed later.

```
# ============================================================
#  1. DATAPREP
# ============================================================
#  1.1 Household consumption aggregates and characteristics
# ============================================================

consumption_aggregates <- ca |>
  mutate(rural = ifelse(urb_rur == 2, 1, 0),  # Create rural indicator
         yhh = totc,              # Total household expenditure
         wgt_adj = pweight) |>        # Make a copy of the weight variable
  select(hhid, rural, hhsize,hhsize_R, marz, aepc, yhh, wgt_adj, weight,
         Foodpovln2022, Lpovln2022, Upovln2022, Avpovln2022,
         poor_Foodpovln2022, poor_Lpovln2022, poor_Upovln2022,
         poor_Avpovln2022, decile )  # Keep only necessary columns
```

0.5.2 1.2 Demographic characteristics, education, Labor Force

Here the original code calls for Zone data, which is not present in our dataset, due to the different administrative structure of Armenia. However, we use urban/rural for this purpose.

```
# Zone data
zone_data <- hh |>
  select(interview__key, hh_01_code, hh_02, hh_03) |>
  mutate(
    hhid = interview__key,
```

```
    zone = hh_01_code,
    marz = hh_02,
    urb_rur = hh_03
  )
```

Demographic data, merge with zone data Note that ed_03 (educy) below is not years of education, but education level (primary, general, secondary, etc.) However, it is ordered in a way that higher levels imply more years of education. We perform several steps within the first pipe call.

```
demographics <- pp |>
  rename(hhid = interview__key) |>
  left_join(zone_data, join_by( hhid == hhid))  |>
  mutate(# Demographic characteristics
         pid = paste0(interview__key, "-",
                      str_pad(mem_001__id, 2, pad = "0")), # Unique person id
         gender = mem_02,
         age = mem_05,
         head = ifelse(mem_03 == 1, 1, 0),
         # Education level
         educy = ifelse(is.na(ed_03) | ed_03 == 8, 0, ed_03),
         # Labor Force Status
         lstatus = case_when(
           est_03 == 1 | est_04 == 1 | est_05 == 1 | est_06 == 1 | est_09 < 7 ~ 1L,
           est_10 == 1 ~ 2L,
           est_10 == 2 ~ 3L,
           .default = 4L # Default to OLF
         ),
         employed = (lstatus == 1),
         # Salaried status (1. paid employee; 2 self-employed)
         salaried = ifelse(!is.na(emp_11a), 1L,
                           ifelse(is.na(emp_11a) & employed == TRUE, 0L, NA_integer_))
         ) |>
  rename(rel = mem_03) # |>
  # select(hhid, pid, gender, age, head, rel, zone, marz, urb_rur, educy,
  #        lstatus, employed, salaried, )
```

Count the number of employed persons by household.

```
demographics <- demographics |>
  mutate(employed = (lstatus == 1)) |>
```

```
  group_by(hhid) |>
  mutate(employed_hh = sum(employed, na.rm = TRUE)) |>  # Count within each household
  ungroup()
```

Here the original Stata code calculates income variables and aggregates them by household. We skip that because the dataset "ic" already has these elements calculated by the WB poverty team. We'll add them later.

Primary and Secondary Job income

- emp_11 11.How much was %rostertitle%'s payment for wages/salary/income for last month?
- emp_12 12.What period of time was the wage/income for?
- emp_25 25.How much was %rostertitle%'s payment for wages/salary/income for last month?
- emp_26 26.What period of time was the wage/income for?

Bonus, In-Kind, and food from job was not asked in Armenia, If it were, you should add a mutate() statement like the ones below for each subcategory.

ARMSTAT. (2023). *Integrated Living Conditions Survey 2022*.