SAM Shares Data Request

Georgia Country Climate and Development Report

Renato Vargas

Table of contents

1 Introduction

In this data request, we estimate shares to disaggregate the Social Accounting Matrix for the Macroeconomic team.

1.1 Objectives

1. HH split: Split our representative household into income quintiles and rural/urban households.
2. Labor split: Split labor into 6 types according to gender and skill level.

3. Economic activity split: Split wages and capital by economic activities.
4. Split household consumption by urban/rural, quintiles and economic activity.

1.2 Preliminaries

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

# Georgia country ISO code
iso <- "GEO"

# Survey year
survey_year <- 2023

# Exchange rate USD per GEL
er <- 0.37

# Years of interest for our macroeconomic scenario analysis
# analysis_years <- c(2030, 2050)
```

We will use the following libraries for this exercise.

```
library(tidyverse)
library(haven)
library(readxl)
library(openxlsx)
library(gt)
```

1.3 Datasets and correspondence tables

We use data from the 2023 survey for the Labor Split and data from the 2022 survey for the household expenditure and capital shares in In mil. GEL per year per household type.

```
 #| lst-label: original-datasets

# Household Unique ID, Weights, Location and other basic variables
hh_basics <- read_sav(
  "data/ilcs_2023/sysschedule.sav") |>
  mutate(
    UID = as.integer(UID))

# Household size (includes no. of family members)
hh_size <- read_sav(
  "data/ilcs_2023/familysize.sav")|>
  mutate(
    UID = as.integer(UID))

# Processed income at household level
hh_income <- read_sav(
  "data/ilcs_2023/tblincomes.sav")|>
  mutate(
    UID = as.integer(UID))

# Consumption aggregate at household level
hh_expenditure <- read_sav(
  "data/ilcs_2023/tblexpenditures.sav")|>
  rename(# rename total expenditure variables
         total_expenditure = MTlianixarjebi_,
         total_expenditure_aeq06 = MTlianimoxmareba_EqAdScale,
         total_expenditure_aeq08 = Mtlianimoxmareba_EqAdScale_08) |>
  mutate(
    UID = as.integer(UID))

# Characteristics of the dwelling
hh_chars <- read_sav(
  "data/ilcs_2023/tblshinda01.sav")|>
  mutate(
    UID = as.integer(UID))

# Persons (pp)
pp <- read_sav(
  "data/ilcs_2023/tblshinda02.sav") |>
  mutate(
    UID = as.integer(UID),
    MemberNo = as.integer(MemberNo))

# Labor (pp)
pp_labor <- read_sav(
  "data/ilcs_2023/tblshinda05_1.sav") |>
  mutate(
    UID = as.integer(UID),
```

```r
    MemberNo = as.integer(MemberNo),
    Q5  = as.integer(Q5),
    Q12 = as.integer(Q12))

# Poverty
poverty <- read_dta(
  "data/ilcs_2023/POVERTY_stata.dta") |>
  mutate(
    UID = as.integer(UID))

# Ind. Poverty
ind_poverty <- read_dta(
  "data/ilcs_2023/IND_POVERTY_stata.dta") |>
  rename(MemberNo = memberno) |>
  mutate(
    UID = as.integer(UID),
    MemberNo = as.integer(MemberNo))
```

We also need look-up tables.

```r
sam_activities <- read_excel(
    "data/sam/classifications.xlsx",
    sheet = "SAM Correspondence",
    col_names = T,
    col_types = "text",
  )

sam_factors <- read_excel(
    "data/sam/classifications.xlsx",
    sheet = "SAM factors",
    col_names = T,
    col_types = "text",
  )

coicop <- read_excel(
    "data/sam/classifications.xlsx",
    sheet = "COICOP",
    col_names = T,
    col_types = "text",
  ) |>
  mutate(simple_code = as.integer(gsub("\\.", "", Coicop)))
```

```
coicop_filtered <- coicop |>
  filter( nchar(as.character(simple_code)) >= 5)
```

1.4 Population totals

First we check that our dataset amounts to population totals.

```
weights <- hh_basics |>
  select(UID, QuartNo, Weights)

hh_size |>
  left_join(weights, join_by(UID)) |>
#  filter(QuartNo == 110) |>
  summarize(
    "Population" = sum(FamilySize * Weights, na.rm = T),
    "Households" = sum(Weights, na.rm = T)) |>
  gt()
```

Population	Households
14861930	4499690

Upon first exploration, we see that the population amounts to 14,861,930 individuals, living in 4,499,690 households, when in reality we have a total population estimate of 3,702,130 individuals, living in 1,109,130 households. This is because the survey covers four quarters and households are interviewed four times in the year. So we need to drop households for our estimates and keep only those related to one quarter. Since we need information for 2022, but our dataset is for 2023, we will use the first quarter (Q1), which is closer to the required year.

```
pop_by_quarter <- hh_size |>
  left_join(weights, join_by(UID)) |>
  group_by(QuartNo) |>
  summarize(
    "Population" = sum(FamilySize * Weights, na.rm = T),
    "Households" = sum(Weights, na.rm = T))

pop_by_quarter |>
  gt()
```

Quarter ID	Population	Households
107	3713876	1122110
108	3654991	1114584
109	3729558	1130628
110	3763505	1132368

2 Generating descriptor variables (SAM disaggregates)

2.1 Skill level

For skill level, we will use information on schooling from `pp$Education` (`TblShinda02`), which has the following levels:

1. Illiterate
2. Do not have primary education but can read and write
3. Pre-primary education
4. Primary education
5. Lower secondary education
6. Upper secondary education
7. Vocational education without secondary general education
8. Vocational education on the base of lower secondary education with secondary general education certificate
9. Vocational education on the base of secondary general education (except higher professional education)
10. Higher professional program
11. Bachelor or equivalent
12. Master or equivalent
13. Doctor or equivalent

We need three skill levels for our SAM template, so we map these levels to:

Low skill (1 - 5): Illiterate through lower secondary. Medium skill (6 - 9): Upper secondary through vocational education. High skill (10 - 13): Higher professional program through Doctor.

```
pp_factor_descriptors <- pp |>
  select(UID, MemberNo, Gender, Age, Education) |>
  mutate(
    MemberId =
      paste0(sprintf("%06d", UID), sprintf("%02d", MemberNo))) |>
  mutate(Gender = factor(
    Gender,
    levels = c(1, 2),
    labels = c("Female", "Male")
  )) |>
  mutate(
    SkillLevel = case_when(
      Education >= 0 & Education <= 5 ~ 1,
      Education > 5 & Education <= 9 ~ 2,
      Education > 9 & Education <= 13 ~ 3,
      TRUE ~ NA ) ) |>
  mutate(
    SkillLevel = factor(
      SkillLevel,
      levels = c( 1, 2, 3),
      labels = c( "Low Skill", "Medium Skill", "High Skill"))
  )
```

Now that we have skill levels, we need to add information on urban/rural (from `hh_basics`) and quintile (from `ind_poverty`), and type of income earner (from `pp_labor`).

```
urb_rur <- hh_basics |>
  select(UID, UrbanOrRural, RegNo, Weights) |>
  mutate(
    UrbanOrRural = factor(
      UrbanOrRural,
      levels = c(2,1),
      labels = c("Rural", "Urban")
    )
  )

quintiles <- poverty |>
  select(UID, quintilc) |>
  rename(Quintile = quintilc) |>
  mutate(
    Quintile = factor(
      Quintile,
      levels = c(1:5),
      labels = c("Q1", "Q2", "Q3", "Q4", "Q5")
    )
  )
```

```
pp_factor_descriptors <- pp_factor_descriptors |>
  left_join(urb_rur, join_by(UID)) |>
  left_join(quintiles, join_by(UID))
```

2.2 Labor status and Economic Activities

We work with labor status from `Shinda05_1`. Since, upon import NACE 1 codes are converted to numbers, we need to convert them back to text, so that we can keep zeros to the left for proper order. We then extract the first two digits and match with 2-digit NACE Rev. 1 and find the correspondence to Rev. 2 from the SAM using the look-up table `sam_activities`. For proper order, we convert the SAM activities columns for job 1 and job 2 to factor, using the order from the dataset `sam_factors`.

```
pp_emstatus <- pp_labor |>
  mutate(
    MemberId =
      paste0(sprintf("%06d", UID), sprintf("%02d", MemberNo))) |>
  mutate(
    # Job 1 NACE Rev 1 code.
    Q5  = paste0(sprintf("%04d", Q5)),
    # Job 2 NACE Rev 1 code.
    Q12 = paste0(sprintf("%04d", Q12))) |>
  mutate(
    job1 = substr(Q5, 1, 2),
    job2 = substr(Q12, 1, 2)
  ) |>
  # We match to Rev 2 and SAM classifications (for job 1 and job 2)
  left_join(
    sam_activities[,c(1,4)],
    join_by(job1 == rev1_2d)) |>
  left_join(
    sam_activities[,c(1,4)],
    join_by(job2 == rev1_2d),
    suffix = c("_job1", "_job2")) |>
  # And convert to factors for proper order
  mutate(
    SAM_job1 = factor(
      SAM_job1,
      levels = sam_factors$SAM,
      # labels = sam_factors$SAM_description
      ),
    SAM_job2 = factor(
      SAM_job2,
      levels = sam_factors$SAM,
      # labels = sam_factors$SAM_description
      )
  )

head(
  pp_emstatus[c(18, 34, 40, 41, 67),
  c("UID", "MemberNo", "SAM_job1", "SAM_job2")]) |>
  gt()
```

UID	MemberNo	SAM_job1	SAM_job2
386848	1	a-cns	a-agri
386855	1	a-agri	a-agri
386856	1	a-mine	a-agri
386856	2	a-agri	a-agri
386866	2	a-heal	a-heal

2.3 Types of income

Before making our multi-dimensional tables, we need to identify different types of income. f-lab (wages) and f-surp (capital income). The instruction is that f-surp needs to be split into wages to entrepreneurs/self employed and capital income.

```
pp_emstatus <- pp_emstatus |>
  mutate(
    # We add accross three months for each source (and coalesce the NAs to 0)
    flab_job1 =
      rowSums(
        across(starts_with("Q8_faqti_"), \(x) coalesce(x, 0))),
    flab_job2 =
      rowSums(
        across(starts_with("Q14_faqti_"), \(x) coalesce(x, 0))),
    fsurp =
      rowSums(
        across(starts_with("Q10_faqti_"), \(x) coalesce(x, 0)))
  ) |>
  # We also add factor labels to Employment Status
  mutate(
    Q7 = factor(
      Q7,
      levels = c(1:6),
      labels = c(
        "Employee", "Employer", "Own Account (Non-Ag.)",
        "Own Account (Ag.)", "Unpaid Worker", "Other"))
    ) |>
  mutate(
    Q13 = factor(
      Q13,
      levels = c(1:6),
      labels = c(
        "Employee", "Employer", "Own Account (Non-Ag.)",
        "Own Account (Ag.)", "Unpaid Worker", "Other"))
    )
```

And we add our labor market variables to our `pp_factor_descriptors` dataset.

```
pp_emstatus <- pp_emstatus |>
  select(-UID,-MemberNo) |>
  left_join(pp_factor_descriptors, join_by(MemberId)) |>
  relocate(c(UID, MemberNo, MemberId), .before = 1)
```

3 Multidimensional Tables

Now we can put together our tables.

3.1 Wages and Surplus income

```
flab1 <- pp_emstatus |>
  select(
    Q7,
    UrbanOrRural,
    Quintile,
    SkillLevel,
    Gender,
    flab_job1,
```

```r
    SAM_job1,
    Weights) |>
  rename(
    EmStatus = Q7,
    FLab = flab_job1,
    SAM = SAM_job1
  ) |>
  mutate(
    FLab = coalesce(FLab, 0) * Weights,
    FactorType = "f-lab"
  )

flab2 <- pp_emstatus |>
  select(
    Q13,
    UrbanOrRural,
    Quintile,
    SkillLevel,
    Gender,
    flab_job2,
    SAM_job2,
    Weights) |>
  rename(
    EmStatus = Q13,
    FLab = flab_job2,
    SAM = SAM_job2
  ) |>
  mutate(
    FLab = coalesce(FLab, 0) * Weights,
    FactorType = "f-lab"
  )

fsurp <- pp_emstatus |>
  select(
    Q7,
    UrbanOrRural,
    Quintile,
    SkillLevel,
    Gender,
    fsurp,
    SAM_job1,
    Weights) |>
  rename(
    EmStatus = Q7,
    FLab = fsurp,
    SAM = SAM_job1
  ) |>
  mutate(
    FLab = coalesce(FLab, 0) * Weights,
    FactorType = "f-surp"
  )

flab <- rbind(flab1, flab2, fsurp)
is.na(flab$FLab) <- 0
```

3.2 Filling out the templates

We first get a table with all the possible combinations, so we can make sure f-lab and f-surp have the data that we need.

```r
factor_income <- flab |>
  filter(
    as.numeric(EmStatus) < 6
  ) |>
  group_by(
    FactorType,
    EmStatus,
    UrbanOrRural,
    Quintile,
    SkillLevel,
    Gender) |>
  summarize(FLab = sum(FLab, na.rm = T)) |>
  ungroup() |>
  pivot_wider(
    id_cols = c(FactorType, EmStatus, UrbanOrRural, Quintile, SkillLevel),
    names_from = c(Gender),
    values_from = FLab ) |>
  pivot_wider(
    id_cols = c(FactorType, EmStatus, UrbanOrRural, Quintile),
    names_from = c(SkillLevel),
    values_from = c(Female, Male) ) |>
  mutate(across(5:10, ~replace_na(., 0)))
```

Salaries from employees, salaries paid to business owners, and own account workers (mixed income).

```r
factor_income_comprehensive <- flab |>
  filter(
    as.numeric(EmStatus) < 6
  ) |>
  group_by(
    FactorType,
    EmStatus,
    UrbanOrRural,
    Quintile,
    SkillLevel,
    Gender) |>
  summarize(FLab = sum(FLab, na.rm = T)) |>
  ungroup() |>
  pivot_wider(
    id_cols = c(FactorType, EmStatus, UrbanOrRural, Quintile, SkillLevel),
    names_from = c(Gender),
    values_from = FLab ) |>
  pivot_wider(
    id_cols = c(FactorType, EmStatus, UrbanOrRural, Quintile),
    names_from = c(SkillLevel),
    values_from = c(Female, Male) ) |>
  mutate(across(5:10, ~replace_na(., 0)))

# Sneak peak
head(factor_income_comprehensive[,c(1:7)]) |>
  gt()
```

FactorType	EmStatus	UrbanOrRural	Quintile	Female_Low Skill	Female_Medium Skill	Fem
f-lab	Employee	Rural	Q1	6570741	73766574	
f-lab	Employee	Rural	Q2	3136840	68728533	
f-lab	Employee	Rural	Q3	3384631	66745137	
f-lab	Employee	Rural	Q4	2903820	89122463	
f-lab	Employee	Rural	Q5	3978437	66745042	
f-lab	Employee	Urban	Q1	10106489	140823577	

Now we single out wages.

```
factor_income_wages <- flab |>
  filter(
    as.numeric(EmStatus) < 6,
    FactorType == "f-lab"
  ) |>
  group_by(
    UrbanOrRural,
    Quintile,
    SkillLevel,
    Gender) |>
  summarize(FLab = sum(FLab, na.rm = T)) |>
  ungroup() |>
  pivot_wider(
    id_cols = c(UrbanOrRural, Quintile, SkillLevel),
    names_from = c(Gender),
    values_from = FLab ) |>
  pivot_wider(
    id_cols = c(UrbanOrRural, Quintile),
    names_from = c(SkillLevel),
    values_from = c(Female, Male) ) |>
  mutate(across(3:8, ~replace_na(., 0)))

factor_income_wages |>
  gt()
```

UrbanOrRural	Quintile	Female_Low Skill	Female_Medium Skill	Female_High Skill	Male_Lo
Rural	Q1	7071449	73976896	40873369	13
Rural	Q2	3136840	69199593	62840081	21
Rural	Q3	3384631	68131258	62812282	9
Rural	Q4	2903820	89812175	101771027	9
Rural	Q5	3978437	67394473	213109248	6
Urban	Q1	10106489	140832561	74981955	9
Urban	Q2	14984587	143794440	180076456	23
Urban	Q3	3875826	251290586	353147864	5
Urban	Q4	3120367	236333189	531182422	2
Urban	Q5	10523005	215233168	1088737655	10

And Capital.

```r
factor_income_capital <- flab |>
  filter(
    as.numeric(EmStatus) < 6,
    FactorType == "f-surp"
  ) |>
  group_by(
    UrbanOrRural,
    Quintile,
    ) |>
  summarize(Capital = sum(FLab, na.rm = T)) |>
  ungroup()

factor_income_capital |>
  gt()
```

UrbanOrRural	Quintile	Capital
Rural	Q1	57510741
Rural	Q2	77436983
Rural	Q3	104578879
Rural	Q4	124723382
Rural	Q5	203923504
Urban	Q1	91338334
Urban	Q2	172982396
Urban	Q3	303575257
Urban	Q4	397204825
Urban	Q5	762309667

And now, differentiated by economic activity.

```r
factor_income_wages_activity <- flab |>
  filter(
    as.numeric(EmStatus) < 6,
    FactorType == "f-lab"
  ) |>
  group_by(
    SkillLevel,
    Gender,
    SAM) |>
  summarize(FLab = sum(FLab, na.rm = T)) |>
  ungroup() |>
  pivot_wider(
    id_cols = c(Gender,SkillLevel),
    names_from = c(SAM),
    names_expand = T,
    names_sort = T,
    values_from = FLab ) |>
  mutate(across(3:40, ~replace_na(., 0)))

factor_income_wages_activity |>
  gt()
```

Gender	SkillLevel	a-agri	a-mine	a-food	a-text	a-ppp	a-p__c	
Female	Low Skill	3277666	4589616	0.0	0.0	0	0	
Male	Low Skill	11116018	1766352	0.0	0.0	7116412	0	53
Female	Medium Skill	30258860	113953145	2998609.2	367597.7	5525985	1545872	116
Male	Medium Skill	110066917	163456438	2484429.9	1582986.1	23385206	46873481	5095
Female	High Skill	8979328	62554597	1164825.2	6153045.6	20555061	0	1400
Male	High Skill	43176973	81815750	488806.7	483916.5	8920082	13381519	1887

And Capital.

```
factor_income_capital_activity <- flab |>
  filter(
    as.numeric(EmStatus) < 6,
    FactorType == "f-surp"
  ) |>
  group_by(
    SAM) |>
  summarize(
    Capital = "Capital",
    FLab = sum(FLab, na.rm = T)) |>
  ungroup() |>
  pivot_wider(
    id_cols = c(Capital),
    names_from = c(SAM),
    names_expand = T,
    names_sort = T,
    values_from = FLab ) |>
  mutate(across(2:39, ~replace_na(., 0)))
```

```
factor_income_capital_activity |>
  gt()
```

Capital	a-agri	a-mine	a-food	a-text	a-ppp	a-p_c	a-che	a-bph	a-rnmm
Capital	45047563	67573351	9525115	1276846	42717980	28145458	0	0	30151417

4 Exports to Excel

Here we export to the same Excel we are using as template, but we don't overwrite the original template, just so we can compare the result to the request.

```
wb <- loadWorkbook("data/sam/SAMshares_GEO.xlsx")
names(wb)
```

```
[1] "SAM_temporary 2022"          "Accounts"
[3] "nace-rev1-en-divisions"      "factor payments"
[5] "factor income"               "factor income comprehensive"
[7] "HH Consumption"              "transfers_incomplete"
```

```
writeData(
  wb,
  "factor income comprehensive",
  factor_income_comprehensive ,
  startRow = 1,
  startCol = 1,
  rowNames = FALSE)
writeData(
  wb,
  "factor income",
  factor_income_wages ,
  startRow = 21,
  startCol = 2,
  rowNames = FALSE)
writeData(
  wb,
  "factor income",
  factor_income_capital ,
  startRow = 5,
  startCol = 13,
  rowNames = FALSE)
writeData(
  wb,
  "factor payments",
  factor_income_wages_activity ,
  startRow = 27,
  startCol = 1,
  rowNames = FALSE)
writeData(
  wb,
  "factor payments",
  factor_income_capital_activity ,
  startRow = 37,
  startCol = 2,
  rowNames = FALSE)
saveWorkbook(
  wb,
  "data/sam/SAMshares_GEO.xlsx",
  overwrite = T)
```