Generating a relational synthetic dataset for an imaginary country

Technical Documentation

Olivier Dupriez and Aivin Solatorio, World Bank, June 2023

Table of Contents

1 Introduction

This document describes the creation of two synthetic datasets. The first one, which we refer to as the ***core dataset*** or ***synthetic census dataset*** is a relational dataset of 10,003,891 individuals (2,501,755 households) representing the entire population of an imaginary middle-income country. The dataset contains two data files: one with variables at the household level, the other one with variables at the individual level. It includes variables that are typically collected in population censuses (demography, education, occupation, dwelling characteristics, fertility, mortality, and migration) and in household surveys (household expenditure, anthropometric data for children, assets ownership). The data only includes ordinary households (no institutional households). From this core dataset, we extracted a stratified sample of 8,000 households. We refer to this second dataset as the ***synthetic survey dataset***. Both datasets were created for the purposes of training and simulation and are not representative of any specific country; they are intended to represent an imaginary middle-income country.

These data present no disclosure risk and can thus be safely shared as open data. Both datasets are published the World Bank Microdata Library, and are available in English and in French under a CC-BY 4.0 license.[1]

To produce these data, we developed our own synthetic data generation models, leveraging deep learning methods. The models were subsequently packaged into REaLTabFormer[2], a model openly available and published in a GitHub repository.[3]

2 Hierarchical generative model

The core population dataset was generated using a four-level hierarchical generative model. The first-level model is the *household composition generator*, which generates variables that define each household's composition (household size and basic demographic profile of members, including age and relationship to the head of household). The second-level model is the *household-level variables generator*, which generates the variables whose values are common to all household members (such as dwelling characteristics) based on the household composition. The third-level model is the *household-head generator*, which generates observations for the

[1] See www.microdata.worldbank.org;

- Census, EN: https://microdata.worldbank.org/index.php/catalog/study/WLD_2023_SYNTH-CEN-EN_v01_M
- Census, FR: https://microdata.worldbank.org/index.php/catalog/study/WLD_2023_SYNTH-CEN-FR_v01_M
- Survey, EN: https://microdata.worldbank.org/index.php/catalog/study/WLD_2023_SYNTH-SVY-EN_v01_M
- Survey, FR: https://microdata.worldbank.org/index.php/catalog/study/WLD_2023_SYNTH-SVY-FR_v01_M

[2] A detailed description of the model is available in our paper "*REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers*" available on ArXiv at https://arxiv.org/abs/2302.02041.

[3] See https://github.com/avsolatorio/REaLTabFormer

head of the households based on the output of the previous two models. The fourth-level model is the *household member generator*, which generates data on the household members, excluding the head, for households of size two and above. The *household member generator* model uses the data generated by the household composition, household-level variables, and household head generator models. This hierarchical model provides relational dependencies within a household that would not be guaranteed if all records were generated independently.

To implement the different models, we adopted a transformer architecture. The household composition generator is a decoder model that generates data from normally distributed noise. The other three models use a sequence-to-sequence model inspired by the application of deep learning to language translation.

Each column (variable) of the dataset is encoded independently, which means that each column has a distinct set of token vocabulary. Different token identifiers are used to represent the same value found across different columns. This encoding strategy inspired by the IBM TabFormer[4] model allowed us to implement constraints during the data generation for each column. We can impose zero probability for tokens or values that are invalid for a given variable, which reduces the production of invalid samples.

The four models were subsequently packaged into REaLTabFormer, a model that generates parent and child tables for the production of tabular and relational datasets.

3 Model implementation

This section provides an overview of the synthetic data generation process and training data sources. More detailed information is provided in the subsequent sections.

We implemented the above-mentioned models using public-use census micro-samples provided by the IPUMS International program as training datasets. We obtained IPUMS data on over 236 million individuals from 30 countries, which by far exceeded what was needed to train the models. From this large dataset, we drew a uniform sample of 1,594,414 households that formed the training data for the production of (part of) the core dataset. We then augmented the core synthetic dataset by imputing a small number of variables extracted from other data sources (sample surveys), using random forest models. We also added geographic variables and the enumeration area variable using specific procedures described in this document.

The process of generating the core synthetic population dataset using the trained models followed a four-step hierarchical sequence:

1. Generate synthetic data on the composition of the synthetic households.

[4] See https://github.com/IBM/TabFormer

2. Generate the household-level synthetic variables using the synthetic household composition as input.
3. Combine the output of the two previous steps and use it as input to generate the data for the heads of households.
4. Combine the output of the three previous steps and use it as input to create the data for individual household members.

Due to the probabilistic nature of the data generation process, the models may create synthetic observations with inconsistencies across variables. To avoid such inconsistencies, we embedded "validators" in the process. Validators are consistency rules within an observation (e.g., a 4-year-old cannot have a tertiary education level) or across observations for a same household (e.g., if one member is declared as spouse of the head of household, the head may not be “never married”, “divorced”, or “widow”). Observations that violate any of the validation rules are automatically rejected and replaced.

3.1 Household composition generator model

The first step in the creation of the core dataset was to create a data file with variables representing the household composition. The model provides flexibility on what variables represent a household composition. We included the following IPUMS variables:

IPUMS Variable	***Description***
HHSIZE	Household size
RELATE	Relationship to the head of the household (frequency)
SEX	Sex of the household members (frequency)
AGE	Age structure of the household (count by 10-year age bins)
MARST	Marital status of the household members (frequency)
LIT	Literacy of the household members (frequency)
EDATTAIN	Educational attainment of the household members (frequency)

We included education variables to ensure consistency across household members (e.g., if the head and spouse have a tertiary education level, it is unlikely that a child would be illiterate at age 15). We generated a vector that captures the distribution of household composition and trained the generative model using this data. We then used the trained model to generate synthetic household composition vectors. The generation process is stochastic and differs from the approach implemented in a package like simPop, which generates the household structure by copying data from the training dataset. In our approach, the composition of households is fully synthetic.

3.2 Household-level variables generator model

Traditional methods for generating synthetic data tend to lose inter-variable dependencies, which reduces the utility of the resulting data. We seek to address this issue by utilizing a transformer-based model (GPT 2). The transformer architecture uses the multi-head attention mechanism that learns long-term relationships across variables. The household-level variables included in our model are listed below, although not all of them are retained in the distributed synthetic data.

IPUMS Variable	***Description***
URBAN	Urban-rural status
OWNERSHIPD	Ownership of dwelling [detailed version]
ELECTRIC	Electricity
WATSUP	Water supply
SEWAGE	Sewage
FUELCOOK	Cooking fuel
FUELHEAT	Fuel for heating
PHONE	Telephone availability
CELL	Cellular phone availability
INTERNET	Internet access
AUTOS	Automobiles available
REFRIG	Refrigerator
TV	Television set
RADIO	Radio in household
ROOMS	Number of rooms
BEDROOMS	Number of bedrooms
TOILET	Toilet
FLOOR	Floor material
WALL	Wall or building material
ROOF	Roof material
MORTNUM	Number of deaths in household last
ANYMORT	Any deaths in household last year
HHTYPE	Household classification
NFAMS	Number of families in household
NCOUPLES	Number of married couples in household
NMOTHERS	Number of mothers in household
NFATHERS	Number of fathers in household

We trained the model on the IPUMS sample dataset (household composition combined with household variables). Using this model (and the output of the household composition generator model as input), we then created the synthetic household-level variables.

3.3 Head generator and member generator models

We built separate models for generating the profile of the head of household (the "head generator" model) and for generating data for the other members of the household (the "member generator" model). Both models follow a Seq2Seq architecture. The head generator model creates a fixed set of data, while the member generator model creates data according to the size of the household. To accomplish this, we concatenate the variables for each household member to produce the training data for the member generator model. We generate a "wide record" for each household, which contains information on all members. We embed special tokens ("[BMEM]" and "[EMEM]") to distinguish members in this wide record. The special tokens are placed before and after the sequence of variables for each household member, marking the beginning and end of an observation for each member, respectively.

The following variables are used to train both models.

IPUMS Variable	***Description***
RELATE	Relationship to household head [general version]
AGE	Age
SEX	Sex
MARST	Marital status [general version]
CHBORN	Children ever born
CHSURV	Children surviving
BIRTHSLYR	Number of births last year
RELIGION	Religion [general version]
INDIG	Member of an indigenous group
SCHOOL	School attendance
LIT	Literacy
EDATTAIN	Educational attainment, international recode [general version]
YRSCHOOL	Years of schooling
EMPSTAT	Activity status (employment status) [general version]
LABFORCE	Labor force participation
OCCISCO	Occupation, ISCO general
INDGEN	Industry, general recode
MIGRATE1	Migration status, 1 year
MIGRATE5	Migration status, 5 years
MIGRATE0	Migration status, 10 years
MIGRATEP	Migration status, previous residence
DISABLED	Disability status
DISBLND	Blind or vision-impaired
DISDEAF	Deaf or hearing-impaired
DISMNTL	Mental disability

The output of the two previous models (household composition and household-level variables) is used as input to generate the synthetic data for the head of household. We then used the outputs of the three models (household composition, household variables, and head of household) as input to generate the synthetic data for the members. All variables listed above are included in the synthetic data being generated.

3.4 Additional variables

To enrich this dataset, we imputed additional variables using other data sources as training data, namely the Demographic and Health Surveys (DHS) program and national household expenditure surveys. This was done using more traditional approaches (random forest model). We describe these imputations in detail in this report.

4 Training datasets

The creation of the core synthetic dataset is a multi-stage process that requires multiple training datasets. The project made use of public data to the extent possible. We first trained a model to generate the core population dataset using IPUMS data, which includes most but not all variables. These are variables typically collected in population censuses: demographic, education, occupation, disability, and housing variables. We then added variables typically collected in DHS surveys, such as detailed information on water sources, anthropometric variables for children aged 0 to 4 years, ownership of a bank account, and additional assets ownership. Finally, we added variables collected from household expenditure or consumption surveys that provide information on households' consumption by category of products and services. IPUMS and DHS are publicly available to registered users, whereas access to the consumption/expenditure datasets is restricted.

4.1 IPUMS International census samples

The training data utilized to generate the core synthetic population dataset is a compilation of sample census datasets obtained from IPUMS International. IPUMS data are publicly available to registered users from https://international.ipums.org/international/.

We selected 43 census datasets from the IPUMS collection, originating from 30 countries, most of which are middle-income countries, and extracted a subset of the IPUMS harmonized variables. The list of countries and census years that were extracted is shown in the table below. One selection criterion was the availability of most of the variables we are interested in including in our core data. As no single dataset contains all these variables, the training dataset contains missing values. The dataset obtained by merging the selected samples contained 236 million observations, which greatly exceeds our needs. From this dataset, we randomly selected a sample of 1,594,414 households to train our models, representing approximately 6.4 million observations. The datasets from which we extracted our sample includes the following IPUMS public use files:

		Households		Individuals	
Country	Year	Count	%	Count	%
		58,807,161	100.00%	236,108,388	100.00%
Argentina	2001	1,040,852	1.77%	3,626,103	1.54%
Argentina	2010	1,217,166	2.07%	3,966,245	1.68%
Bangladesh	2001	2,625,959	4.47%	12,442,115	5.27%
Bangladesh	2011	1,654,631	2.81%	7,205,720	3.05%
Bolivia	2012	292,117	0.50%	1,003,516	0.43%
Botswana	2001	42,375	0.07%	168,676	0.07%
Botswana	2011	61,792	0.11%	201,752	0.09%
Brazil	1991	4,024,553	6.84%	17,045,712	7.22%
Brazil	2000	5,304,711	9.02%	20,274,412	8.59%
Brazil	2010	6,192,502	10.53%	20,635,472	8.74%
Chile	2002	437,766	0.74%	1,513,914	0.64%
Colombia	2005	1,054,812	1.79%	4,006,168	1.70%
Ghana	2000	379,372	0.65%	1,894,133	0.80%
Ghana	2010	570,234	0.97%	2,466,289	1.04%
Indonesia	2010	6,151,164	10.46%	23,603,049	10.00%
Jordan	2004	97,343	0.17%	510,646	0.22%
Kenya	1999	317,106	0.54%	1,407,547	0.60%
Kenya	2009	895,230	1.52%	3,841,935	1.63%
Laos	2005	99,098	0.17%	560,480	0.24%
Malawi	2008	298,607	0.51%	1,341,977	0.57%
Mali	2009	235,834	0.40%	1,451,856	0.61%
Mexico	2005	2,546,985	4.33%	10,284,550	4.36%
Mexico	2010	2,903,640	4.94%	11,938,402	5.06%
Mexico	2015	2,927,196	4.98%	11,344,365	4.80%
Mozambique	2007	463,420	0.79%	2,047,048	0.87%
Myanmar	2014	1,237,712	2.10%	5,032,818	2.13%
Nepal	2001	411,851	0.70%	2,067,609	0.88%
Nepal	2011	669,492	1.14%	3,238,842	1.37%
Peru	2007	705,498	1.20%	2,745,895	1.16%
Philippines	2000	1,511,890	2.57%	7,417,810	3.14%
Philippines	2010	2,066,824	3.51%	9,411,256	3.99%
South Africa	2007	345,170	0.59%	1,047,657	0.44%
South Africa	2011	1,326,354	2.26%	4,418,594	1.87%
South Africa	2016	984,627	1.67%	3,328,793	1.41%
South Sudan	2008	92,592	0.16%	542,765	0.23%
Sudan	2008	922,816	1.57%	5,066,530	2.15%

Tanzania	2012	950,776	1.62%	4,498,022	1.91%
Thailand	2000	165,417	0.28%	604,519	0.26%
Turkey	2000	934,627	1.59%	3,444,456	1.46%
Venezuela	2001	543,475	0.92%	2,306,489	0.98%
Vietnam	2009	3,692,042	6.28%	14,177,590	6.00%
Zambia	2010	250,805	0.43%	1,321,973	0.56%
Zimbabwe	2012	160,728	0.27%	654,688	0.28%

The variables used for our models are the harmonized (recoded) variables produced by IPUMS International. To ensure that census data from multiple countries can be mapped to a harmonized category, the harmonized variables contain categories that accommodate the specificity of each country's data. For example, the variable "cooking fuel" (*fuelcook*) may have categories such as "coal", "charcoal", "coal or charcoal", and "wood or charcoal", which may have overlap that would not be found in a census dataset. Some variables therefore contain more categories than would typically be found in a national census dataset.

We trained our models using the variable categories as provided by IPUMS. After generating the synthetic observations, we recoded some of these categories to make the population dataset more representative of a typical country dataset. The nested coding system adopted by IPUMS made this easy. For example, codes 31 to 34 could be mapped to code 30, and codes 41 to 47 to code 40. We also grouped some codes in a somewhat arbitrary manner, considering the frequencies of each category to generate less perturbative groupings. Our objective was not to generate data representative of a specific country, so we were able to make these groupings as needed. An example of such groupings is shown in the table below for the variable "cooking fuel", where codes 53 to 56 were grouped into one category "Coal or charcoal".

0	NIU (not in universe)		
10	None		
20	Electricity		
30	Petroleum gas, unspecified		
		31	Gas -- piped/utility
		32	Gas -- tanked or bottled
		34	Liquefied petroleum gas
40	Petroleum liquid		
		41	Oil, kerosene, and other liquid fuels
		42	Kerosene/paraffin
		47	Diesel
51	Wood and other plant fuels		
52	Non-wood plant materials		
53	Coal or charcoal		
54	Charcoal		
55	Coal		

56 Wood or charcoal
61 Bottled gas and wood
66 Other combinations
70 Other
  72 Biogas
  74 Dung/manure
  76 Solar energy
99 Unknown/missing

IPUMS country datasets do not contain all variables we were interested in. Therefore, our IPUMS training dataset contains missing values. The model used to generate synthetic data is able to learn the distribution of missing values for each observation (although the resulting synthetic data does not include missing values).

4.2 Demographic and Health Survey (DHS) datasets

We use a selection of DHS datasets to build a second training dataset. This dataset contains a set of variables that are common with the IPUMS data, and additional variables that we want to add to the core synthetic dataset. The common variables are intended to be used as predictors in the imputation process.

DHS datasets are freely available to registered users. We used data from the following 15 surveys with a total of 1,326,054 observations:

Country	Year	Individuals
Total		1,326,054
Armenia	2015	27,768
Colombia	2015	162,459
Egypt	2014	120,276
Ghana	2014	43,945
Indonesia	2017	197,723
Jordan	2017	93,347
Kenya	2014	153,840
Namibia	2013	41,646
Nepal	2016	49,064
Pakistan	2017	100,869
Philippines	2017	120,273
Turkey	2013	45,660
Tanzania	2015	64,880
South Africa	2016	38,850
Zambia	2018	65,454

The variables of interest are (mostly) harmonized across DHS surveys. They were recoded as necessary to match the variables in the IPUMS training data (the variables must be made consistent for the imputation process).[5]

4.3 Global Consumption Database microdata

To incorporate information on household expenditure into the core synthetic dataset, we created a third training dataset out of microdata from the World Bank's Global Consumption Database (GCD). These data are not publicly accessible, as they are owned by the respective countries, and the World Bank is not authorized or mandated to publish them.

The GCD microdata comprises a set of harmonized datasets derived from national surveys of various types, including Household Income and Expenditure Surveys, Household Budget Surveys, Household Consumption Surveys, Living Standards Measurement Surveys, and equivalent. All surveys in this collection have nationwide coverage and contain variables on the demographic composition and other characteristics of household members and dwellings. Expenditure data are available for each household by COICOP group (105), class (37), and category (12) of products and services.

Initially, we selected the following 58 datasets to build the training dataset for household expenditure:

Country	Year	Survey
Bangladesh	2000	Household Income and Expenditure Survey
Bangladesh	2005	Household Income and Expenditure Survey
Bangladesh	2010	Household Income and Expenditure Survey
Bulgaria	2007	Multi-topic Household Survey
Bhutan	2007	Living Standards Survey
Bolivia	2007	Encuesta de Hogares
Brazil	2008	Pesquisa de Orçamentos Familiares
Cambodia	2006	Socio-economic survey
Cambodia	2008	Socio-economic survey
Cambodia	2012	Socio-economic survey
Cameroon	2007	Enquête Camerounaise auprès des Ménages
Cameroon	2014	Enquête Camerounaise auprès des Ménages
Cape Verde	2007	Questionário Unificado de Indicadores Básicos de Bem-Estar
Colombia	2008	Encuesta Nacional de Calidad de Vida
Colombia	2010	Encuesta Nacional de Calidad de Vida
Egypt	2009	Household Expenditure and Consumption Survey
El Salvador	2004	Encuesta de Hogares de Propósitos Múltiples

[5] This recoding and data preparation was done using Stata (script *DHS_prepare_data_for_synthetic_data.do*). The resulting dataset that serves as training dataset was named *DHS_all_selected.dta*.

Ethiopia	2010	Household Income Consumption and Expenditure
Gabon	2005	Enquête Gabonaise pour l'Evaluation et le Suivi de la Pauvreté
Georgia	2013	Welfare Monitoring Survey
Ghana	2006	Living Standards Survey
Ghana	2012	Living Standards Survey
Guatemala	2006	Encuesta Nacional sobre Condiciones de Vida
Honduras	2004	Encuesta Nacional de Condiciones de Vida
India	2004	National Sample Survey
India	2009	National Sample Survey
India	2011	National Sample Survey
Indonesia	2002	National Socio-Economic Survey
Indonesia	2012	National Socio-Economic Survey
Iraq	2012	Household Socio Economic Survey
Jordan	2002	Household Income and Expenditure Survey
Jamaica	2007	Survey of Living Conditions
Kazakhstan	2011	Household Budget Survey
Kenya	2005	Integrated Household Budget Survey
Kyrgyz Republic	2010	Integrated Household Survey
Lao PDR	2007	Household Expenditure and Consumption Survey
North Macedonia	2008	Household Budget Survey
Mexico	2010	Encuesta Nacional de Ingreso-Gasto de los Hogares
Mexico	2012	Encuesta Nacional de Ingreso-Gasto de los Hogares
Moldova	2012	Household Budget Survey
Mongolia	2012	Household Income and Expenditure Survey
Morocco	2006	Enquête Nationale sur la Consommation et les Dépenses des Ménage
Namibia	2009	Household Income Expenditure Survey
Nepal	2010	Living Standards Survey
Pakistan	2013	Social and Living Standards Measurement Survey
Peru	2005	Encuesta Nacional de Hogares
Peru	2008	Encuesta Nacional de Hogares
Peru	2010	Encuesta Nacional de Hogares
Philippines	2006	Family Income and Expenditure Survey
Philippines	2012	Family Income and Expenditure Survey
South Africa	2000	Income and Expenditure Survey
South Africa	2010	Income and Expenditure Survey
Sri Lanka	2002	Household Income and Expenditure Survey
Sri Lanka	2006	Household Income and Expenditure Survey
Sri Lanka	2009	Household Income and Expenditure Survey
Sri Lanka	2012	Household Income and Expenditure Survey
Uganda	2013	National Household Survey
Ukraine	2013	Household Budget Survey

The datasets comprise values in local currency unit and for different reference years. We converted all local currency values into 2020 $PPP values, utilizing household nominal consumption growth obtained from the World Bank's World Development Indicators database and purchasing power parities (PPP) conversion factors from the same source. We also scaled the expenditure values by applying a simple multiplying factor for each survey, ensuring that the annual mean per capita expenditure in each survey was 3,500 $PPP. The resulting data file comprises the consumption profiles of 1,207,951 households. Some variables in this data file overlap with the IPUMS and DHS training data and will be utilized as predictors in the imputation process. The content of this third training dataset is as follows:

Variable name	**Variable label**
hhno	Unique household ID
svy	Survey
hid	Household ID
stratum	Stratum
psu	Primary sampling unit
geo_1	Geographic code (level 1)
geo_2	Geographic code (level 2)
urbrur	Area of residence
hhsize	Household size
m_00_15	Nb males, 0 to 15 years
m_16_59	Nb males, 16 to 59 years
m_60p	Nb males, 60 years and over
f_00_15	Nb females, 0 to 15 years
f_16_59	Nb females, 16 to 59 years
f_60p	Nb females, 60 years and over
nb_0_4	Members aged 0-4 years
nb_0_17	Members aged 0-17 years
nb_18_59	Members aged 18-59 years
nb_60_	Members aged 60+ years
nb_mal	Number of male members
nb_fem	Number of female members
adeq_fao	Adults equivalent (FAO scale)
hhcomp	Household type
hhsex	Sex of the head
hhagey	Age of the head
hhcivil	Marital status of the head
hheduc	Level of education of the head
ownhouse	Ownership of dwelling unit
roof	Main material used for roof
walls	Main material used for external walls
floor	Main material used for floor
rooms	Number of habitable rooms

water	Main source of drinking water
fuelligh	Main source of lighting
toilet	Main toilet facility
ownland	Ownership of land
landsize	Land size owned (ha)
llivesk	Nb of large-sized livestock owned
mlivesk	Nb of medium-sized livestock owned
poultry	Nb of poultry owned
radio	Ownership of a radio
tv	Ownership of a television
phone	Ownership of a telephone (fix or cell)
cell	Ownership of a cell phone
refrigerator	Ownership of a refrigerator
sewmach	Ownership of a sewing machine
computer	Ownership of computer
stove	Ownership of a stove
bicycle	Ownership of a bicycle
motorcycle	Ownership of a motorcycle
car	Ownership of a private car
oxcart	Ownership of an animal cart
boat	Ownership of a boat
c37_0111	Bread and cereals
c37_0112	Meat
c37_0113	Fish and seafood
c37_0114	Milk, cheese, and eggs
c37_0115	Oils and fats
c37_0116	Fruits
c37_0117	Vegetables
c37_0118	Sugar, jam, honey, chocolate, and confectionery
c37_0119	Food products n.e.c.
c37_0120	Non-alcoholic beverages
c37_0210	Alcoholic beverages
c37_0310	Clothing
c37_0320	Footwear
c37_0401	Housing, water, electricity, gas, and other fuels
c37_0440	Water supply and miscellaneous services relating to the dwelling
c37_0450	Electricity, gas, and other fuels
c37_0510	Furniture and furnishings, carpets, and other floor coverings
c37_0530	Household appliances
c37_0550	Tools and equipment for house and garden
c37_0560	Goods and services for routine household maintenance
c37_0610	Medical products, appliances, and equipment
c37_0640	Out-patient and hospital services

c37_0710	Purchase of vehicles
c37_0720	Operation of personal transport equipment
c37_0730	Transport services
c37_0801	Communication
c37_0910	Audio-visual, photographic and information processing equipment
c37_0920	Other major durables for recreation and culture
c37_0930	Other recreational items and equipment, garden, and pets
c37_0940	Recreational and cultural services
c37_1011	Education
c37_1111	Catering services
c37_1121	Accommodation services
c37_1210	Personal care
c37_1230	Personal effects n.e.c.
c37_1260	Financial services n.e.c.
c37_1271	Other services n.e.c.
pcexp	Per capita household expenditure
wta_hh	Household weighting coefficient
wta_pop	Population weighting coefficient (= wta_hh * hhsize)
piped_water	Recode of water (Main source of drinking water)
electricity	Recode of electcon (Connection to electricity in dwelling)
cook_fuel	Recode of fuelcook (Main cooking fuel)
flush_toilet	Recode of toilet (Main toilet facility)

The distribution of log per capita expenditure in this combined dataset is quasi-normal and provides a credible household consumption dataset for an imaginary country (Figure 1).

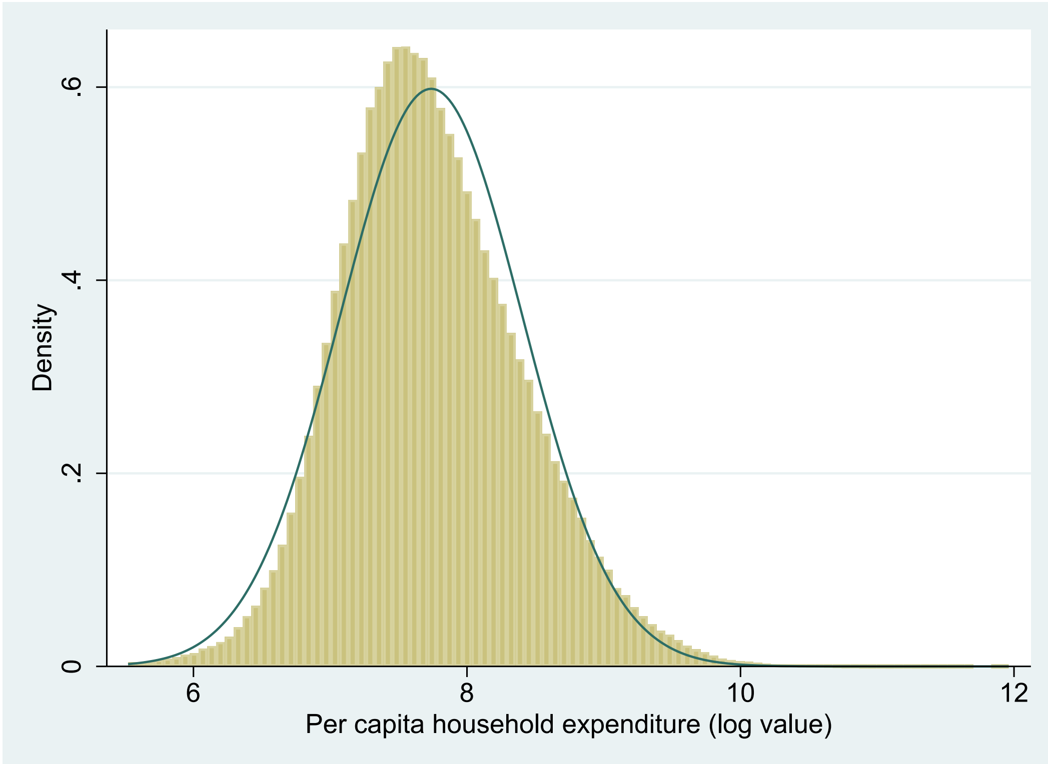

Figure 1 - Log per capita expenditure, training dataset (all observations)

5 Training and running the synthetic data generation model

5.1 Practical implementation

We previously described in broad terms the process of generating the synthetic population from the trained generative models. We provide here some more technical information on the implementation of the process.[6]

The scripts are all written in Python, and the generative models are implemented using PyTorch. We used the ipumspy[7] package to load the DDI file (metadata in XML format) included in the data dump downloaded from IPUMS. This allowed us to easily parse the raw file containing the individual information. We converted the Stata [.dat] file downloaded from IPUMS into a parquet

[6] The main purpose of this section is to record information useful to replicate or adapt the process in the future.
[7] See https://ipumspy.readthedocs.io/en/latest/

file partitioned on the SAMPLE variable for efficient processing.[8] Below is a snippet for using the code:

```
Prerequisite to the pipeline is the IPUMS_SAMPLE.parquet file. This needs to be first generated using the
`IPUMSDDIProcessor` which is implemented in `synthetic_data.ipums.processor`.

```python
# cd data/01_raw/ipums
from synthethic_data.ipums.processor import IPUMSDDIProcessor

idp = IPUMSDDIProcessor(ddi_xml="ipums_ddi.xml")
idp.split_by_sample("ipumsi_00079.dat.gz", to_parquet=True, base_name="IPUMS_SAMPLE", batch_size=500_000)
```
```

After generating the above parquet file, we created a unique household identifier based on the SAMPLE and SERIAL variables. A hash identifier derived from the unique household identifier is used to randomly bucket households across census samples. The hashing and sampling are implemented as a kedro pipeline.[9] The shuffled parquet data is generated by running the kedro pipeline:

```
kedro run –pipeline ip
```

Note the following data artefacts:

- Input
    - ipums_ddi.xml: this is renamed from ipumsi_00079.xml file received from IPUMS.
    - IPUMS_SAMPLE.parquet: this is generated using the steps above.
- Output
    - IPUMS_SHUFFLED.parquet: stored in data/01_raw/ipums/

We implemented another pipeline that processes the shuffled IPUMS data to derive household composition variables and the main input data for the generative models. The output data is another parquet file containing information for each household, namely: *hid*, *hh_comp*, *hh*, *head*, *members*, *valid*, *split*. The *hid* variable represents the unique identifier for the household, the *hh_comp* variable is a vector of token ids representing the derived household structure, *hh* contains the household variables for the household, *head* is a vector of the head of household information, *members* is a combined vector of information of the household members. We also note if the household is valid by checking that the household contains one and only one head. We store that information in the *valid* column. We filter the data used in the model on this variable to ensure that all households in the training data have exactly one head. We split the

[8] The parquet file is stored in the IPUMS_SAMPLE.parquet. The relevant code for this section is implemented in the module src/synthetic_data/ipums/processor.py.

[9] Script: nodes.py

household data into training (*train)*, validation (*val*), and *test* groups indicated by the *split* variable.

The pipeline generates the vocabulary and all the derived mappings for the values present in the data. These mappings will be used in model training and for decoding the generated data by the model into the final raw synthetic data.[10] This can be run (assuming all previous steps have been done) by:

```
kedro run -pipeline td
```

Once the raw data for training the model is ready, we can start generating the final training data formatted for the generative models. The pipeline creates the necessary input-output pairs for the Seq2Seq models.[11] To generate all the training data, we execute the kedro pipeline as follows:

```
kedro run -tag=train_dataset_tag
```

The training data is now available, so the models can be trained. The snippet below was used to train the models.[12] The pipeline saves the models for use later in generating the synthetic data.

```
# # train_models.sh
# Train the hh_comp model and the hh_comp_hh model
kedro run --pipeline tm -n train_gen_hh_comp_model && kedro run --pipeline tm -n train_gen_hh_comp_hh_model

# Train the seq2seq_hh_comp_hh model, the seq2seq_head model, and the seq2seq_members model
kedro run --pipeline tm -n train_seq2seq_hh_comp_hh_model && kedro run --pipeline tm -n train_seq2seq_head_model
&& kedro run --pipeline tm -n train_seq2seq_members_model
```

Once the models are trained, the raw synthetic data can be generated using a Jupyter notebook named 01-Test Generate Synthetic Samples.ipynb.

The hardware used to process the data, train the models, and generate the raw synthetic data was an on-prem workstation running on Ubuntu 22.04 with 2x AMD EPYC H12 64-core CPU (256 threads), 2x GPU RTX 3090 24GB VRAM, and 1TB of RAM.

5.2 Dealing with missing values

The IPUMS data used to train the core population generator models contain missing values. While we preserved the missing values in the training of the model, we do not want the synthetic dataset to contain missing values, as the synthetic dataset is intended to provide an accurate and full representation of the population. To address this, we imputed the missing values during the generative process of synthetic observations. This was done by explicitly removing the token that represents a missing value in the list of candidate tokens used to generate the value of the

[10] The pipeline described above is implemented in the kedro node: nodes.py

[11] More information about the processing implementation is available in code accessible in script: models_dataset.py

[12] The kedro nodes are defined in model_training.py

variable. This suppression guarantees that the model will not generate missing values for the variable while preserving the distribution for the valid values.

5.3 Validations embedded in the process

We embedded a set of *validators* in the process of generating our synthetic dataset. The validators are rules that verify the consistency across variables of a same observation and across observations of a same household. Records that violate any of the rules are automatically rejected. The following validators were embedded in the data generation process[13]:

- There must be one and only one head in each household.
- The head of household must be >= 16 years old
- If the relationship to the head for one or more member(s) of a household is declared as "spouse", the marital status of the head must be "married / in union"
- If the relationship to the head of a person is "spouse", the marital status of that person must be "married/in union"
- If the relationship to the head of a person is "spouse", *age* must be >= 14
- If *age* is < 12, then the marital status of that person must be "single"
- The age difference between the head of household and members declared as children of the head must be > 14
- Years of schooling must be 0 for persons who never attended school
- Years of schooling must be <= age - 4
- Years of schooling must be >= 5 if the education attained is "primary completed"
- Years of schooling must be >= 10 if the education attained is "secondary completed"
- Years of schooling must be > 12 if the education attained is "higher completed"
- If education attained is "primary completed", age must be >= 10
- If education attained is "secondary completed", age must be >= 15
- If education attained is "higher completed", age must be >= 18
- If education attained is "primary completed" or "secondary completed" or "higher completed", *literacy* must be "yes"
- If years of schooling >= 5, *literacy* must be "yes"
- If cooking fuel is "electricity", electricity must be "yes"
- If *sex* is "male" OR (*sex* is "female" and *age* < 12), the number of children ever born and surviving must be 0 or NA
- If *sex* is "female" and number of children ever born > 1, *age* must be > 11 + number of children ever born
- If not NA, number of children ever born must be >= number of surviving children
- Births last year must be <= children ever born
- If *age* > 49 births last year must be 0
- The number of children ever born, and number of surviving children must be < 20
- If *bedrooms* must be at most the number of *rooms*

[13] Script: synthetic-population-data/src/synthetic_population_data/validators/validator.py

We instructed the model to generate 5,000,000 households. This is more than what we needed for a target population of 10 million individuals. The excess was intended to account for rejections and to obtain a pool of households from which we can extract our final synthetic population. The model created 4,435,035 households that satisfied the validation criteria, which corresponds to a rejection rate of 11.29%.

6 Correction of age heaping, and age in months

The training data shows significant age heaping, as was expected. The models used to create the synthetic data was able to learn this pattern and the synthetic data show a similar age heaping issue (Figure 2).

We created two versions of the *age* variable: the original one (variable *age*, representing age data as collected), and one that has been partially corrected for age heaping (variable *age_fix*).[14] We used the Whipple's index to quantify the age heaping in the synthetic dataset. The Whipple's index is 118 for the *age* variable in the synthetic dataset, which corresponds to age data of "approximate" quality.[15]

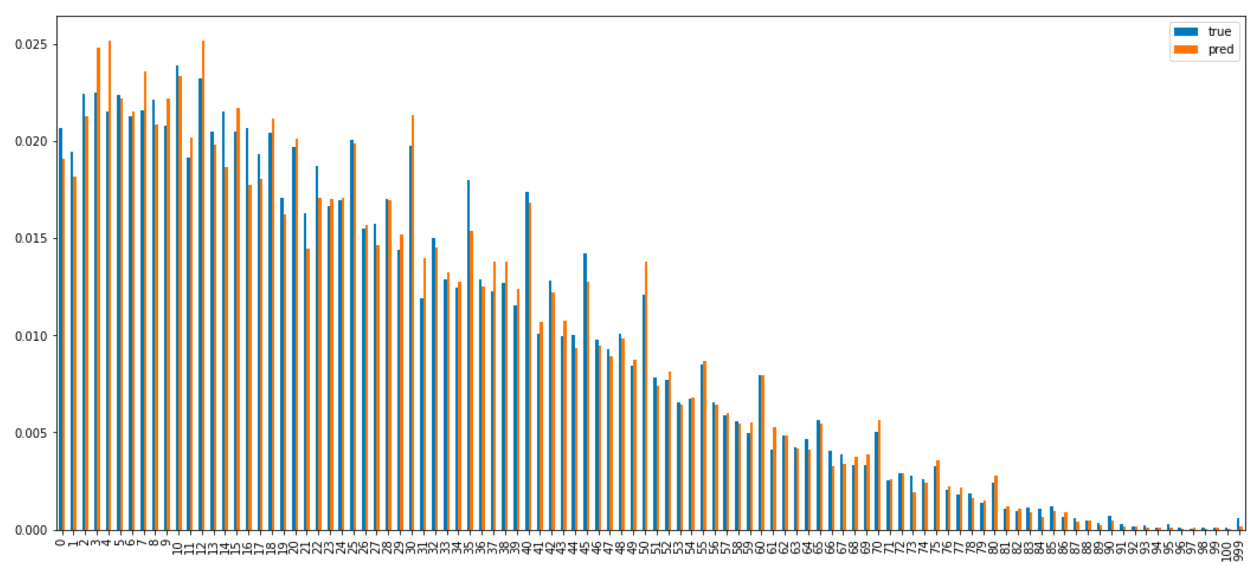

Figure 2 - Age distribution, original vs synthetic, showing age heaping

We corrected the heaping by smoothing the distribution and generating an *age_fix* variable (keeping variable *age* as generated by the model). Both age variables are included in the synthetic dataset. To generate the *age_fix* variable, we implemented an algorithm that redistributes the

[14] We only included the original variable in the published synthetic datasets.
[15] Based on United Nations recommendation. See https://en.wikipedia.org/wiki/Whipple%27s_index.

age for ages that are divisible by 5 (denoted as c_{age}) to age values around it, starting from age 25. We did not fix the age for the population below 24 to avoid creating inconsistencies with variables such as school attendance or school attainment. The algorithm operates as follows:

- Get the count of individuals with age +- 1 year with respect to the central age (denote as: c_{age+1} and c_{age-1})
- Take the mean of c_{age+1} and c_{age-1}, let this mean be denoted by m_{age}.
- Calculate the excess allocation for the central age by subtracting m_{age} from the value of the central age.
- Define a distribution parameter δ which dictates the proportion of the excess that will be redistributed to the other ages. The number of individuals whose age must be reassigned is given by:

$$d = \delta\left(c_{age} - m_{age}\right)$$

  This means that prior to redistribution, the count of individuals in the central age is $m_{age} \cdot (1 - \delta) + c_{age}$. This provides a guarantee that in the immediate locality of c_{age}, the slope is negative.
- Define the redistribution probability of reassigning the age of an individual in c_{age} to any age in the range $c_{age-N/2}$ to $c_{age+N/2}$:

$$p_{age_i} = \frac{\log space(1,\ 0,\ N)_i}{\sum_i \log space(1,\ 0,\ N)_i}$$

  This probability distribution provides a logarithmically decreasing allocation of individuals by age, which again preserves the negative slope in the locality of c_{age}.

Applying this algorithm to the synthetic data produces an age distribution (Figure 3) with a Whipple's index of 100.09 which represents a "highly accurate" age distribution (by United Nations standards).

As we planned to add anthropometric variables (height and weight) to the dataset for children aged less than 5 (imputed from DHS survey data, see section 2.8), whose analysis requires availability of age in months for children aged < 5 years, we also created an *age_month* variable as follows:

$$age_{month} = 12 \cdot age + R;\ R \in [0,\ 11]$$

where R is a random number of months with a range from 0 to 11.

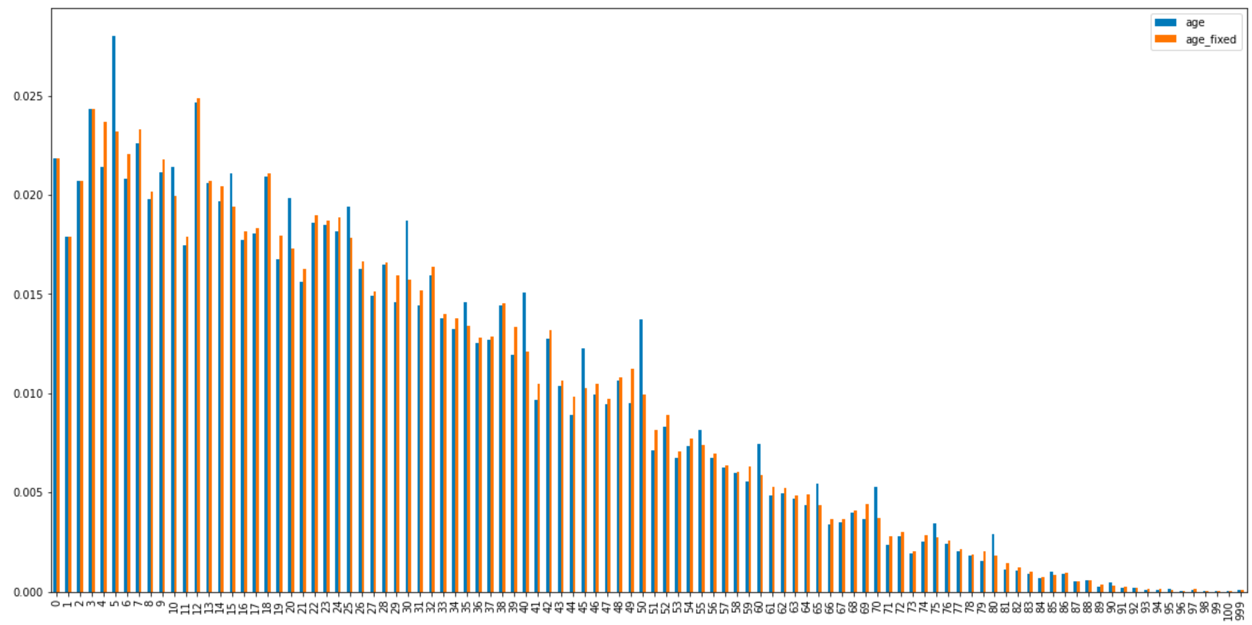

Figure 3 - Age distribution before (blue) and after (orange) heaping correction

7 Generating enumeration areas

We distributed the population by enumeration area (variable *ea*) to obtain a core dataset that can be used as a realistic sample frame to draw stratified samples.

"*For interviewer-based censuses, enumerators assigned to different enumeration areas cover all households and persons in the enumeration area during a specified and usually short period of time in order to meet the requirements of universality and simultaneity. Correctly delineated, enumeration areas will: (a) Be mutually exclusive (non-overlapping) and exhaustive (cover the entire country); (b) Have boundaries that are easily identifiable on the ground; (c) Be consistent with the administrative hierarchy; (d) Be compact and have no pockets or disjoined sections; (e) Have populations of approximately equally size; (f) Be small and accessible enough to be covered by an enumerator within the census period. The chosen population size varies from country to country and is generally determined on the basis of pretest results. Average population size may also vary between rural and urban areas since enumeration can proceed more quickly in towns and cities than in the countryside*."[16]

Creating the enumeration areas required to first decide on a distribution of enumeration areas by size (number of households), then to allocate observations by enumeration area in a meaningful manner.

---

[16] United Nations Statistics Division. 2007. Principles and Recommendations for Population and Housing Censuses Revision 3.

We randomly selected the population of each enumeration area from a negative binomial distribution with a standard deviation of 100 and with a mean of 350 for rural areas, and 500 in urban areas. To distribute the population into enumeration areas in a realistic way, we conducted some analysis of enumeration areas in DHS survey datasets.

7.1 Analysis of DHS data for enumeration area insights

Households in an enumeration area tend to have a somewhat similar profile, compared with households in other enumeration areas (clustering effect). To confirm this, we analyzed the enumeration area data available in the 15 DHS surveys we selected.[17] We perform a K-Means clustering using variables common to the IPUMS and DHS. We chose to use K=50 clusters to have high granularity.

The clustering effect within enumeration areas may be detected by comparing the characteristics of households within the same enumeration areas. The K-Means clustering allowed us to represent household characteristics into vectors. The vector values are based on the distance to cluster centroids. We use the vectors to compute the cosine similarity between households. Knowing the true enumeration area in the survey data, we derive the mean similarity of households within the same enumeration area (intra-EA similarities). We then segment this value by urban and rural to validate another hypothesis that there exist differences in household characteristics between urban and rural areas.

We first visualize the principal components (PCA) of the dataset obtained by appending the 15 selected DHS datasets into one data file. Figure 4 shows that the surveys mix well with each other for the first 2 principal components. Some clustering structures appear. We then fit a clustering model with an arbitrarily large number of clusters (50) on this dataset. For each household, we generate a vector representing the distance of the household to each cluster's centroid. Using these vectors, we measure the cosine similarity between households.

A summary of our empirical investigation is shown in Figure 5. The distributions of average similarity of household characteristics in urban and rural areas (orange and blue lines, respectively) are distinct with different means. The distributions, however, overlap significantly.

The hypothesis that clustering within enumeration areas exists is validated. This is supported by the wide distribution in blue, showing the household characteristic similarities for all pairs of households, with its mean depicted by the vertical red line in the graph. The vertical red line corresponds to the average similarity of household characteristics, regardless of the enumeration area. We observe that the urban and rural enumeration areas have mean intra-EA similarities higher than average.

[17] Script: 04-DHS Analyze Enumeration Areas.ipynb

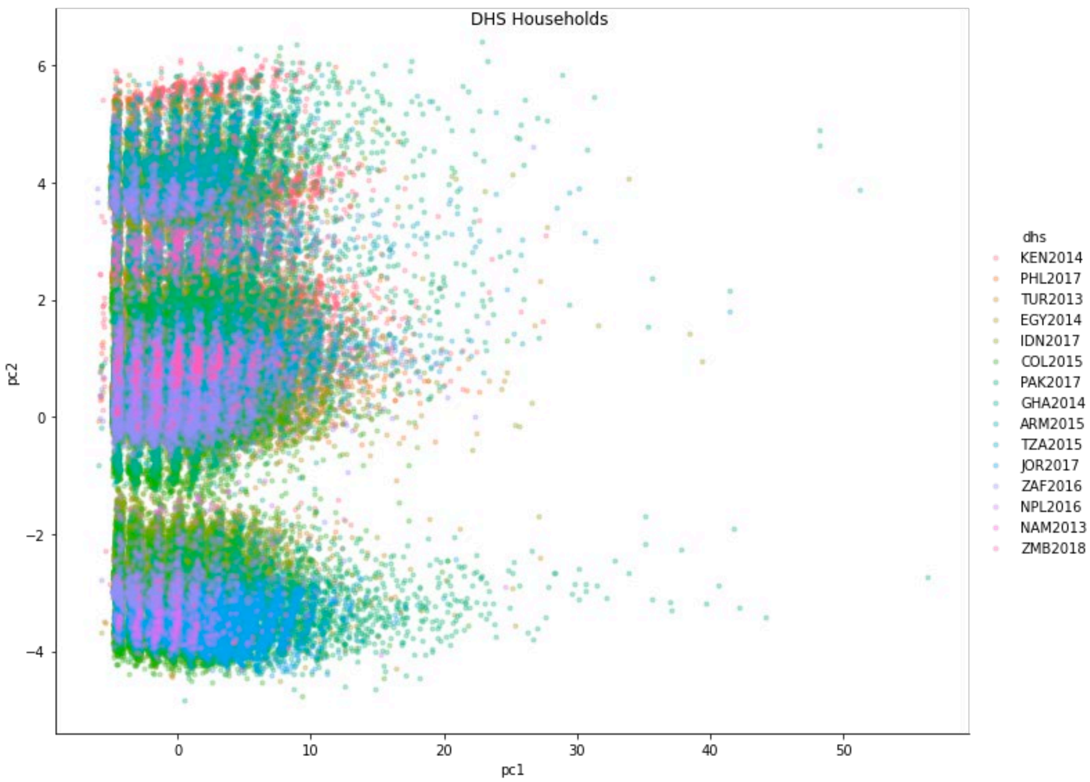

Figure 4 – Plot of first and second principal components, 15 DHS surveys

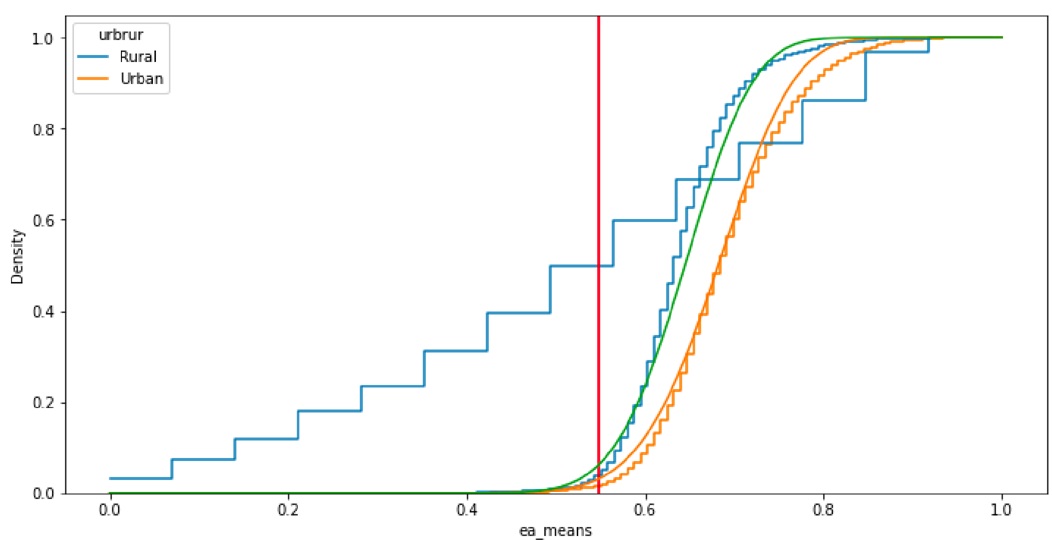

Figure 5 – Empirical distribution of average cosine similarities of household characteristics within the same cluster. The data are is segmented by urban and rural. The vertical red line corresponds to the mean similarities of household characteristics when no aggregation is considered.

We want to reflect this clustering effect in our synthetic data. For this purpose, we extracted additional information from the DHS data to analyze the empirical characteristics of an enumeration area, with the intent to inform a probabilistic model to distribute households into enumeration areas. Enumeration areas in DHS surveys typically consist of a small number of households. The cumulative density distribution plot (Figure 6) shows that 50 percent of the enumeration areas in our DHS datasets have less than 15 households, and 80 percent have less than 25 households. This limits the statistical stability of the metrics computed from enumeration areas. Despite this constraint, we derived general insights to guide the probabilistic generative model for distributing households to enumeration areas.

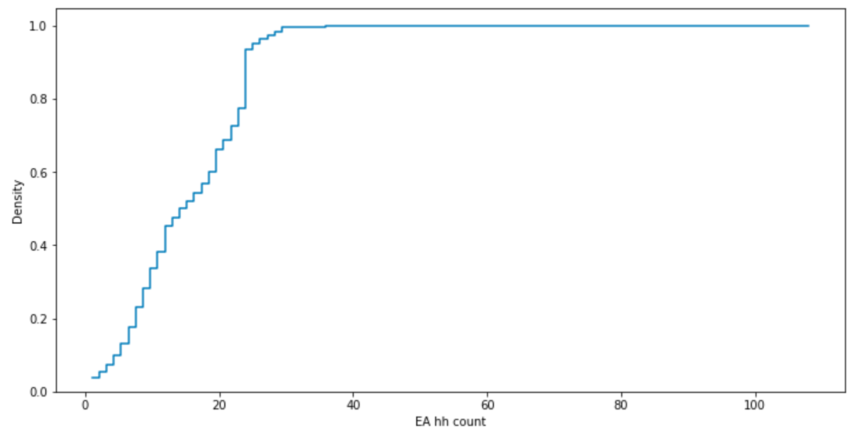

Figure 6 -Cumulative density plot of DHS enumeration areas by number of households

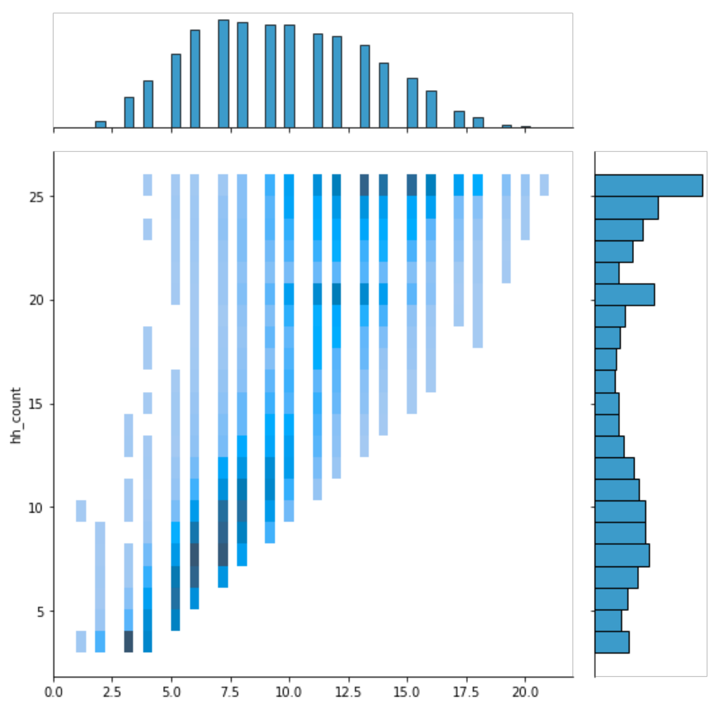

Figure 7 – Joint distribution of household count by enumeration area and the number of households for each cluster in the DHS surveys.

We analyzed the frequency distribution of households within enumeration areas. We found a distribution that resembles a truncation at the higher tail, with a cut-off above 25 (Figure 7). This was expected as DHS surveys typically sample around 25 households per enumeration area. We are most interested in the distribution of the number of distinct clusters of households that belong in the same enumeration area. This distribution, while not perfect, appears to be approximated by a Poisson distribution which we used in our probabilistic model.

7.2 Construction of the synthetic enumeration area variable

Guided by the empirical analysis of DHS surveys, we used a hierarchical generative probabilistic model to assign households to enumeration areas. The same process was applied to urban and rural areas, but with different sets of parameters. Analogous to the method we applied to the empirical data, we fit a K-means clustering model with 50 clusters to obtain a granular grouping of households in the synthetic data. A singular value decomposition (SVD) with 10 components was then used to reduce the dimension of the input to the clustering model. After training the model, we predicted the cluster identifier for each household. This cluster identifier was used to assign households to an enumeration area. Since the model is generative, we had to specify each component responsible for conditioning the generation of certain parameters.

A high-level description of the generative process of an enumeration area is as follows:

- Get the number of households that will be assigned to the enumeration area.
- Get the "seed cluster" for this enumeration area. Households will be sampled from this cluster.
- Use the information learned from the empirical data that an enumeration area is composed of households coming from different clusters. Get the number of distinct clusters for this enumeration area.
- Get other "related clusters" from which households will be sampled. Related clusters are identified proportional to the similarity of the clusters. The more similar the cluster is with the seed cluster (using cosine similarity), the more likely it will be chosen as a related cluster. The number of related clusters chosen is based on the number drawn above.
- Randomly sample households from each cluster, where a drawn household is more likely to come from samples in clusters similar to the seed cluster.
- Continue the process until the required number of households for the enumeration area is sampled, or until no more households is available.

The above steps group all households into enumeration areas. We then use these enumeration areas to distribute households by geography.

The process of generating an enumeration area is segmented by the urban/rural variable, the mechanism is the same, but the parameters are different for the two segments. We take households that are identified under each segment. Then the process proceeds based on this generative model:

- Number of households ~ NB (ea_hh_size_mean, ea_hh_size_std)
    - We use a negative binomial distribution to model the number of households for each enumeration area. The model is parameterized by the mean and standard deviation parameters that are unique for urban and rural segments.
- Number of clusters in EA (N) ~ Poisson(mean_ea_cluster_num)
    - We consider the Poisson distribution for representing the distribution of the number of clusters in an enumeration area. This is guided by the findings in the empirical analysis (Figure 7).
- Choose a seed cluster based on the probability density of households in each cluster.
- Using the seed cluster, get the other clusters (N-1) based on the cosine similarity vector of cluster centroids.
- The cosine similarity vector is transformed into probabilities using the softmax transformation.
- The probability vector is then used to condition the likelihood of a cluster being chosen together with the seed cluster for the current EA.

While the clustering within enumeration areas is guaranteed, the reallocation will not be perfect as the available number of households varies as the reallocation proceeds. We expect that some enumeration areas will consist of households belonging entirely to the same cluster. The shape of the distribution is also parameterized by applying a temperature parameter to the softmax.

The plots of the mean intra-EA cosine similarities of households is shown in Figure 8. The distributions capture the general properties of the empirical data. The urban and rural enumeration areas differ distinctly from the unsegmented pool of household characteristic similarities. Also, the distribution for the urban segment shows more diversity compared with the rural segment, which is also exhibited in the empirical data.[18]

[18] Script: 08-Household Clustering and Enumeration Area Generation.ipynb

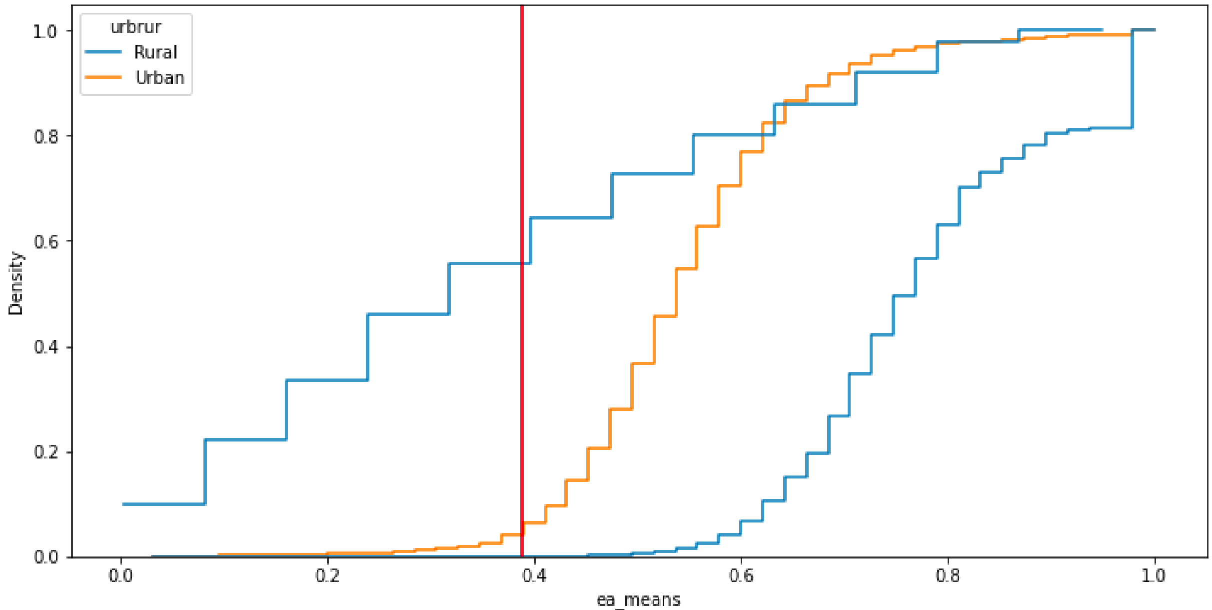

Figure 8 - Distribution of average household similarities within the same enumeration area, segmented by urban and rural, produced by the generative model.

8 Adding geographic variables

The generative models trained using the IPUMS, DHS and GCD data do not generate geographic attributes (we did not include geographic variables in the training datasets), except for the related *urbrur* (urban/rural) variable. To represent the geographic distribution of the population, we added two variables (*geo1* and *geo2*) to the synthetic data, which represent the geographic locations equivalent to admin1 and admin2 administrative area levels in our imaginary country.

We assign the synthetic observations to geographic areas based on a target distribution provided as a table of population distribution (%) by *geo1*, *geo2*, and urban/rural (*urbrur*). We created a table loosely inspired by the Philippines 2015 census data.[19] We generated a target distribution into 10 *geo1* (admin1) areas, 61 *geo2* (admin2) areas, and with an urban/rural breakdown. All regions have urban and rural population, except for *geo_01* which represents the capital city of our imaginary country and is entirely urban. The target distribution of the synthetic population by geographic area and urban/rural is provided in the table below.

[19] Source: https://psa.gov.ph/content/urban-population-philippines-2020-census-population-and-housing. Table A. Total Population, Urban Population, and Percent Urban by Region, Province, and Highly Urbanized City: Philippines, 2020 and 2015

Target distribution of the population by geographic area (admin1 and admin2 levels), and urban/rural (% of total population)

GEO1	GEO2	ALL	URBAN	RURAL
ALL	**ALL**	**100.00%**	**51.23%**	**48.77%**
geo_01	geo_01_01	2.15%	2.15%	0.00%
geo_01	geo_01_02	1.19%	1.19%	0.00%
geo_01	geo_01_03	3.03%	3.03%	0.00%
geo_01	geo_01_04	1.93%	1.93%	0.00%
geo_01	geo_01_05	0.86%	0.86%	0.00%
geo_01	geo_01_06	1.16%	1.16%	0.00%
geo_01	geo_01_07	1.16%	1.16%	0.00%
geo_01	geo_01_08	1.27%	1.27%	0.00%
geo_02	geo_02_01	0.12%	0.00%	0.12%
geo_02	geo_02_02	0.44%	0.02%	0.42%
geo_02	geo_02_03	0.87%	0.08%	0.79%
geo_02	geo_02_04	1.34%	0.19%	1.15%
geo_02	geo_02_05	0.80%	0.13%	0.67%
geo_02	geo_02_06	1.23%	0.25%	0.98%
geo_02	geo_02_07	1.60%	0.37%	1.23%
geo_02	geo_02_08	3.71%	1.17%	2.55%
geo_03	geo_03_01	2.34%	0.78%	1.56%
geo_03	geo_03_02	1.94%	0.83%	1.11%
geo_03	geo_03_03	2.93%	1.93%	1.00%
geo_03	geo_03_04	3.26%	2.68%	0.58%
geo_03	geo_03_05	0.64%	0.63%	0.01%
geo_04	geo_04_01	4.51%	1.66%	2.85%
geo_04	geo_04_02	3.64%	2.62%	1.02%
geo_04	geo_04_03	3.27%	2.52%	0.75%
geo_04	geo_04_04	2.86%	2.67%	0.18%
geo_05	geo_05_01	0.52%	0.02%	0.50%
geo_05	geo_05_02	1.14%	0.12%	1.03%
geo_05	geo_05_03	1.62%	0.30%	1.33%
geo_05	geo_05_04	1.30%	0.35%	0.95%
geo_05	geo_05_05	1.93%	0.58%	1.36%
geo_05	geo_05_06	1.42%	0.45%	0.97%
geo_05	geo_05_07	0.74%	0.42%	0.32%
geo_06	geo_06_01	2.09%	0.18%	1.91%
geo_06	geo_06_02	1.90%	0.33%	1.57%
geo_06	geo_06_03	2.47%	1.51%	0.97%
geo_06	geo_06_04	1.00%	0.83%	0.17%
geo_07	geo_07_01	1.40%	0.22%	1.17%
geo_07	geo_07_02	1.34%	0.50%	0.84%

geo_07	geo_07_03	2.91%	1.31%	1.60%
geo_07	geo_07_04	1.68%	1.59%	0.09%
geo_08	geo_08_01	0.88%	0.04%	0.84%
geo_08	geo_08_02	0.80%	0.06%	0.74%
geo_08	geo_08_03	1.71%	0.21%	1.49%
geo_08	geo_08_04	1.01%	0.22%	0.80%
geo_09	geo_09_01	0.97%	0.14%	0.84%
geo_09	geo_09_02	2.04%	0.37%	1.67%
geo_09	geo_09_03	0.58%	0.13%	0.45%
geo_09	geo_09_04	2.18%	0.61%	1.57%
geo_09	geo_09_05	1.62%	0.59%	1.04%
geo_09	geo_09_06	2.00%	0.81%	1.19%
geo_09	geo_09_07	1.02%	0.56%	0.45%
geo_09	geo_09_08	1.44%	1.35%	0.10%
geo_10	geo_10_01	0.80%	0.10%	0.70%
geo_10	geo_10_02	0.95%	0.21%	0.73%
geo_10	geo_10_03	0.79%	0.21%	0.58%
geo_10	geo_10_04	1.28%	0.42%	0.87%
geo_10	geo_10_05	1.52%	0.63%	0.89%
geo_10	geo_10_06	2.13%	1.03%	1.10%
geo_10	geo_10_07	1.63%	0.96%	0.67%
geo_10	geo_10_08	1.95%	1.69%	0.26%
geo_10	geo_10_09	1.01%	0.94%	0.07%

This table was not used in the synthetic data generation process, which means that we did not know, when the synthetic data were generated, how many observations would be needed for each area. To ensure that we would have enough urban and rural households to meet the target distribution in the table, we generated a large pool of households. The model had been instructed to generate a dataset of 5 million households, which resulted in a dataset of 4,435,035 households after rejecting observations that did not pass the validation rules). We extracted the population from this pool according to the target allocation by *geo1*, *geo2*, and *urbrur*. The allocation of households/population by geographic area was done by allocating entire enumeration areas to a geographic location to guarantee that enumeration areas do not span over two different geographic areas.

9 Adding DHS variables

The variables derived from DHS datasets that we want to add to our synthetic data are the height and weight of children aged 0 to 5 years, the main source of drinking water, the type of toilet used by the household, the ownership of a bicycle and motorcycle, and a variable indicating whether any member of the household has a bank account. These variables are available in many of the Demographic and Health Survey (DHS) datasets. We acquired data from 15 DHS and recoded some of their variables to match variables available from the IPUMS datasets. These recoded variables are used as predictors for the imputation of additional variables to our synthetic data.

We used a random forests regression model for the imputation of the variables, which performed better than linear regression models with regularization. The histograms in Figure 9 and Figure 10 compare the prediction obtained by these two approaches for the prediction of children height and weight, and show that the random forest models capture the true distribution better than the linear regression models.[20]

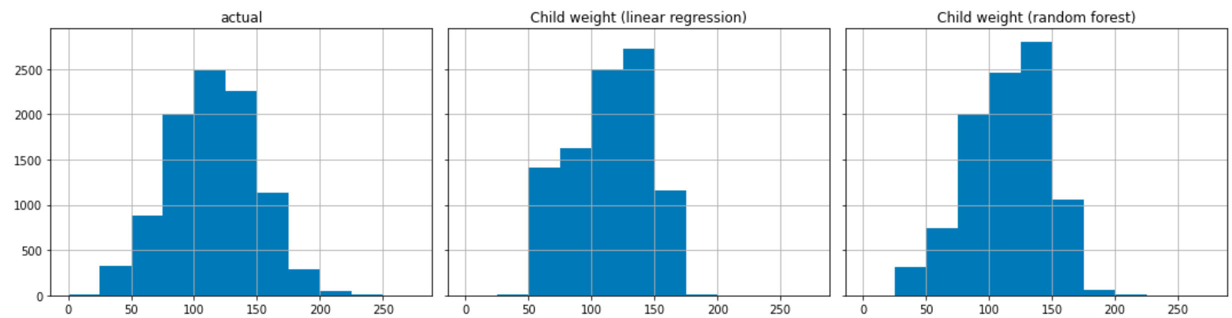

Figure 9 - Predicting children weight: linear regression vs random forest

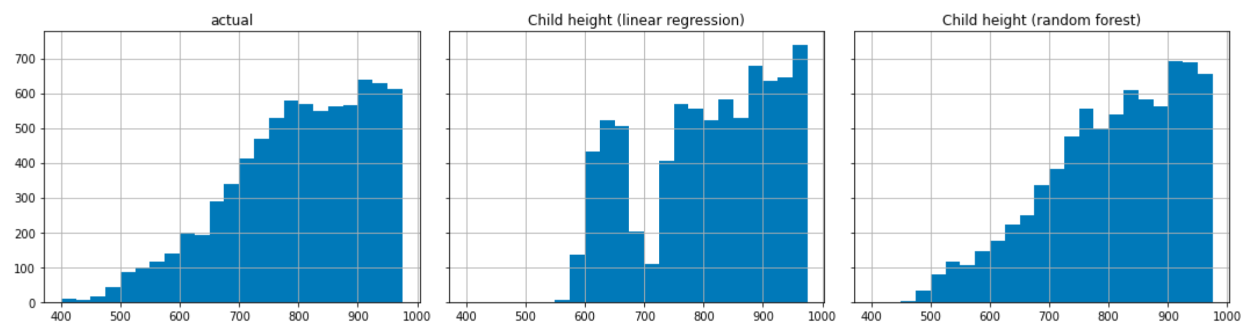

Figure 10 - Predicting children height: linear regression vs random forest

10 Adding consumption variables

We added variables describing each household's consumption to the synthetic dataset. We used data from the World Bank Global Consumption Database as input (see section 2.3.3). The variables we added to the synthetic dataset consist of variables representing the annual consumption of each household for 12 COICOP categories of products and services. We broke down the imputation of the consumption variables into two problems. A random forest

[20] Script: 09.01-Variable Imputation Model - DHS.ipynb

regression model aimed at modeling the total household expenditure and was used to impute the total consumption on the synthetic data.[21] We then used a transformer-based model to generate the consumption classes (shares by class of product/service). The transformer model captures the distribution of the consumption shares better than a random forests model (Figure 11). The model simultaneously estimates the proportions for the 37 COICOP classes.

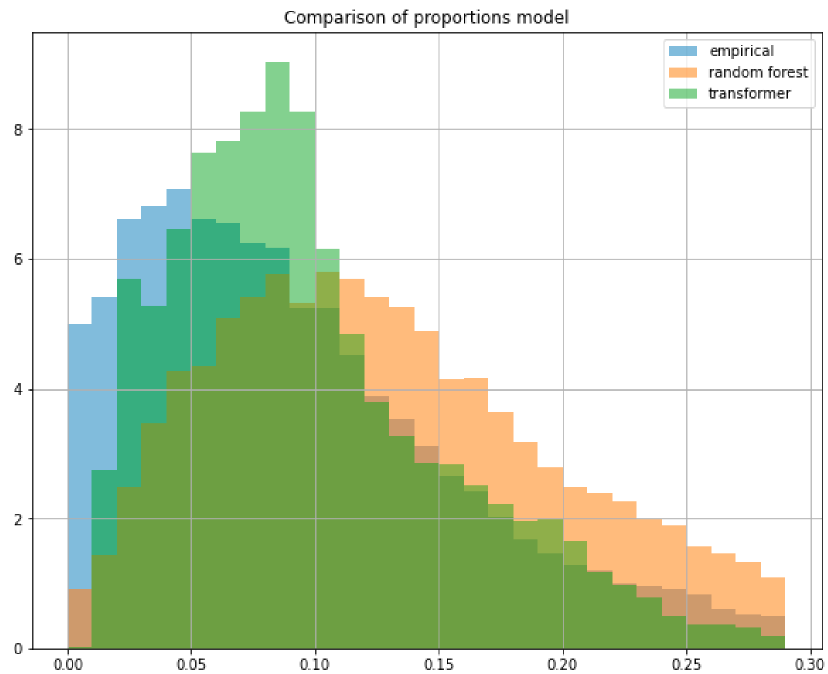

Figure 11 - Comparison between the transformer-based model and the random forest model inference on one of the expenditure categories.

We then aggregated the 37 classes into a smaller set of COICOP categories (from 37 classes to 12 categories), which are included in the final synthetic dataset. Some derived variables were calculated, including the share of food in the household expenditure, and population quintiles based on per capita expenditure (at the national level, and separately for urban and rural areas).

This resulted in the following variables being added to the synthetic dataset:

C12_01 to C12_12	Household annual expenditure by category of product/service (see data dictionary in section 2.10 for detail)
tot_exp	Annual household consumption (total), including home-consumption and use value of durable goods.
tot_food	Annual household food consumption (total)
pc_exp	Annual household expenditure per capita
food_share	Share of food and non-alcoholic beverages (not including catering) in household total expenditure

[21] Script: 09.02-Variable Imputation Model - Expenditure.ipynb

quint_nat	Population quintile of per capita expenditure (20% of population, not households), calculated at the national level
quint_rur	Population quintile of per capita expenditure (20% of population, not households), calculated for the rural population
quint_urb	Population quintile of per capita expenditure (20% of population, not households), calculated for the urban population

11 Data dictionary

The data dictionary of the final synthetic dataset is provided in the table below (in the French version of the dataset, the variable names have been translated). The dataset contains two files: one at the household level, one at the individual level. Variable *hid* is the unique household identifier and is the key variable to be used to merge the two data files. The combination of variables *hid* and *indid* forms a unique individual identifier.

Synthetic dataset - Data dictionary

[0] Generated by the synthetic data model or derived from modelled variables
[1] Imputed from Demographic and Health Surveys (DHS)
[2] Imputed from Global Consumption Database (GCD)
* Variables common to IPUMS and DHS recoded data

Name	Label	Description
HOUSEHOLD LEVEL DATA FILE		
version	Version of the dataset	A version number assigned to the synthetic data file. Not included in published dataset (version is part of the metadata)
hid [0]	Household No	Household unique identifier
geo1 [0]	Geographic level admin 1	Geographic area corresponding to a state or province (admin1 level), This variable is created by distributing households according to a pre-defined distribution (a table of admin1 and admin2 areas and population size by urban/rural with population of each). Some clustering is applied to obtain some degree of homogeneity within the areas.
geo2 [0]	Geographic level admin 2	Geographic area corresponding to districts (admin2 level). Created based on a pre-defined population distribution (see geo1).
ea [0]	Enumeration area	Census enumeration area number. Enumeration areas are areas within districts that have a population between 400 and 1000 persons (more in urban than in rural). Some clustering is applied before distributing the population by EA to have some homogeneity within EAs.
urbrur [0]	Urban/rural	Modeled from IPUMS *urban* variable. 1 = Rural 2 = Urban

hhsize [0] *	Household size	The household size is derived from the IPUMS dataset.
statocc [0]	Status of occupation of the dwelling	Recoded from IPUMS *ownership* then generated by the core model. 1 = Owned 2 = Rented 3 = Occupied for free
rooms [0]	Number of rooms in the dwelling	Recoded from IPUMS *rooms* 1 to 20 (20 = 20 or more)
bedrooms [0] *	Number of bedrooms in the dwelling	Recoded from IPUMS *bedrooms* 1 to 20 (20 = 20 or more)
floor [0] *	Main materials of the floor	Recoded from IPUMS *floor* 1 = Earth 2 = Cement/concrete 3 = Tile 4 = Stone 5 = Wood 6 = Other
walls [0] *	Main materials of the walls	Recoded from IPUMS *wall* 1 = Cardboard/scrap 2 = Wood/straw 3 = Bricks 4 = Concrete/cement 5 = Adobe/mud 6 = Stone 7 = Metal 8 = Other
roof [0] *	Main materials of the roof	Recoded from IPUMS *roof* 1 = Concrete/cement 2 = Tile 3 = Asphalt/laminate 4 = Slate 5 = Metal 6 = Wood 7 = Thatch 8 = Scrap 9 = Other
water [1]	Main source of drinking water	Imputed from the DHS dataset. The variable water was recoded from DHS variable hvxxx. 11 = Piped into dwelling 12 = Piped to yard/plot 13 = Piped to neighbor 14 = Public tap/standpipe 21 = Tube well or borehole 31 = Protected well 32 = Unprotected well 41 = Protected spring 42 = Unprotected spring 43 = River/dam/lake/ponds/stream/canal/irrig 51 = Rainwater 61 = Tanker truck 62 = Cart with small tank 71 = Bottled water 96 = Other

piped_water [0] *	Piped water supply	Recoded from IPUMS *watsup* then generated by the core model. This variable is also extracted (and recoded) from the DHS dataset, to be used as a predictor. 0 = No piped water 1 = Piped into dwelling 2 = Piped outside the dwelling 3 = Public piped water
toilet [1]	Toilet facility	Recoded from DHS variable *hv205* 11 = Flush to piped sewer system 12 = Flush to septic tank 13 = Flush to pit latrine 14 = Flush to somewhere else 21 = Ventilated improved pit latrine (vip) 22 = Pit latrine with slab 23 = Pit latrine without slab/open pit 31 = No facility/bush/field 96 = Other
flush_toilet [0] *	Flush toilet	Recoded from IPUMS *toilet* 0 = No toilet 1 = Flush toilet 2 = Toilet/latrine with no flush
electricity [0] *	Electricity	Recoded from IPUMS *electric* then generated by the core model. This variable is also extracted (and recoded) from the DHS dataset, to be used as a predictor. 0 = No 1 = Yes
cook_fuel [0] *	Cooking fuel	Recoded from IPUMS *fuelcook* then generated by the core model. This variable is also extracted (and recoded) from the DHS dataset (variable hv226), to be used as a predictor. 1 = Electricity 2 = Gas 3 = Petroleum 4 = Wood 5 = Coal/charcoal 6 = Other
phone [0] *	Has a phone (landline)	Recoded from IPUMS *phone* 0 = No 1 = Yes
cell [0] *	Has a cell phone	Recoded from IPUMS *cell* 0 = No 1 = Yes
car [0] *	Has a car	Recoded from IPUMS *autos* 0 = No 1 = Yes
bicycle [1]	Has a bicycle	Imputed from DHS hv210, not in IPUMS 0 = No 1 = Yes
motorcycle [1]	Has a motorcycle or scooter	Imputed from DHS hv211; not in IPUMS 0 = No 1 = Yes

refrigerator [0] *	Has a refrigerator	Recoded from IPUMS *refrig* 0 = No 1 = Yes
tv [0] *	Has a television	Recoded from IPUMS *tv* 0 = No 1 = Yes
radio [0] *	Has a radio	Recoded from IPUMS *radio* 0 = No 1 = Yes
bank [1]	Any member has a bank account	Imputed from DHS; not in IPUMS 0 = No 1 = Yes
deaths_12m [0] *	Number of deaths in the household in the past 12 months	Recoded from IPUMS *mortnum* Value with range 0 to 7
exp_01 [2]	Expenditure on: Bread and cereals	Expenditure on: Food and non-alcoholic beverages
exp_02 [2]	Expenditure on: Meat	Expenditure on: Alcoholic beverages, tobacco, and narcotics
exp _03 [2]	Expenditure on: Fish and seafood	Expenditure on: Clothing and footwear
exp _04 [2]	Expenditure on: Milk, cheese and eggs	Expenditure on: Housing, water, electricity, gas, and other fuels
exp _05 [2]	Expenditure on: Oils and fats	Expenditure on: Furnishing, household equipment and routine household maintenance
exp _06 [2]	Expenditure on: Fruits	Expenditure on: Health
exp _07 [2]	Expenditure on: Vegetables	Expenditure on: Transport
exp _08 [2]	Expenditure on: Sugar, jam, honey, chocolate, and confectionery	Expenditure on: Communication
exp _09 [2]	Expenditure on: Food products n.e.c.	Expenditure on: Recreation and culture
exp _10 [2]	Expenditure on: Non-alcoholic beverages	Expenditure on: Education
exp _11 [2]	Expenditure on: Alcoholic beverages	Expenditure on: Catering and accommodation services
exp _12 [2]	Expenditure on: Clothing	Expenditure on: Miscellaneous goods and services
tot_exp [2]	Total expenditure	Annual household consumption (total), including home-consumption and use value of durable goods.
tot_food	Total food expenditure	Annual household food consumption (total)
share_food	Food share in total expenditure	Share of food and non-alcoholic beverages (not including catering) in household total expenditure
pc_exp [2]	Annual household expenditure per capita	Annual household expenditure per capita
quint_nat	Expenditure quintile, national	Population quintile of per capita expenditure (20% of population, not households), calculated at the national level
quint_urb	Expenditure quintile, urban	Population quintile of per capita expenditure (20% of population, not households), calculated for the urban population
quint_rur	Expenditure quintile, rural	Population quintile of per capita expenditure (20% of population, not households), calculated for the rural population

Name	Label	Description
INDIVIDUAL LEVEL DATA FILE		
hid [0]	Household No	Household unique identifier
idno [0]	Person number	Generated variable; sequential number from 1 (for head) to N within each household.
relation [0] *	Relation to the head	Recoded from IPUMS *relate* 1 = Head 2 = Spouse/partner 3 = Child 4 = Other relative 5 = Non related
sex [0] *	Sex	Recoded from IPUMS *sex* 1 = Male 2 = Female
age [0] *	Age	Recoded from IPUMS *age* Age in completed years, as reported Capped at age 95 (95 = 95+)
age_fix [0]	Age (fixed)	Age in completed years, fixed for heaping Not included in the published dataset.
age_month [0] *	Age in months	Reported age in months, for ages <= 5 years
marstat [0] *	Marital status	Recoded from IPUMS *marst* 1 = Single/never married 2 = Married/in union 3 = Divorced/separated 4 = Widowed
religion [0]	Religion	Recoded from IPUMS *religion* 1 = No religion 2 = Religion A 3 = Religion B 4 = Religion C 5 = Religion D 6 = Religion E 7 = Other
school_attend [0] *	School attendance status	Recoded from IPUMS *school* 0 = NIU (not in universe) 1 = Yes 2 = No, not specified if ever attended 3 = No, attended in the past 4 = No, never attended
educ_attain [0] *	Educational attainment	Recoded from IPUMS *edattain* 0 = NIU (not in universe) or no education 1 = Less than primary completed 2 = Primary completed 3 = Secondary completed 4 = University completed
yrs_school [0] *	Years of schooling	Recoded from IPUMS *yrschool* 0 to 18 = number of years (18 = 18+) 90 = Not specified 91 = Some primary 92 = Some technical after primary 93 = Some secondary

		94 = Some tertiary 95 = Adult literacy 96 = Special education 98 = Unknown 99 = Not in universe
literacy [0]	Literacy	Recoded from IPUMS *lit* 0 = Not in universe 1 = Yes 2 = No
act_status [0]	Activity status	Recoded from IPUMS *empstat* 0 = NIU (not in universe) 1 = Employed 2 = Unemployed 3 = Inactive
labor_force [0]	Labor force participation	Recoded from IPUMS *labforce* 0 = NIU (not in universe) 1 = Yes 2 = No
occupation [0]	Occupation, ISCO	Recoded from IPUMS *occisco* 0 = Not in universe 1 = Legislators, senior officials, and managers 2 = Professionals, 3 = Technicians and associate professionals 4 = Clerks 5 = Service workers and shop and market sales 6 = Skilled agricultural and fishery worker 7 = Crafts and related trades workers 8 = Plant and machine operators and assemblers 9 = Elementary occupations 10 = Armed forces 11 = Other occupations, unspecified or n.e.c.
industry [0]	Industry	Recoded from IPUMS *indgen* 0 = NIU (not in universe) 1 = Agriculture, fishing, and forestry 2 = Mining and extraction 3 = Manufacturing 4 = Electricity, gas, water, and waste management 5 = Construction 6 = Wholesale and retail trade 7 = Hotels and restaurants 8 = Transportation, storage, and communications 9 = Financial services and insurance 10 = Public administration and defense 11 = Business services and real estate, 12 = Education 13 = Health and social work 14 = Other services 15 = Private household services, 16 = Other industry, n.e.c.

migrate_recent [0]	Migration in past 12 months	Recoded from IPUMS *migrate (N-years)* 0 = NIU (not in universe) 10 = Same major administrative unit 11 = Same major, same minor administrative unit 12 = Same major, different minor administrative unit 20 = Different major administrative unit 30 = Abroad 99 = Unknown/missing
migrate_5yr [0]	*Migration in past 5 years*	*This variable was used in some simulations, but not included in the published synthetic dataset.*
disability [0]	Disability status	Recoded from IPUMS *disabled* 0 = No disability 1 = Has disability
blind [0]	Disability – Blind	Recoded from IPUMS *disblnd* 0 = No 1 = Yes
deaf [0]	Disability – Deaf	Recoded from IPUMS *disdeaf* 0 = No 1 = Yes
mental [0]	Disability – Mental	Recoded from IPUMS *dismntl* 0 = No 1 = Yes
ch_height[1]	Height in cm (children 0 to 59 months old)	For children aged < 5 years Imputed from DHS
ch_weight[1]	Weight in grams (children 0 to 59 months old)	For children aged < 5 years Imputed from DHS
children_born [0] *	Children ever born	Recoded from IPUMS *chborn* For women age 12+ (otherwise, NIU) 0 to 20 (20 = 20+)
children_surv [0] *	Children surviving	Recoded from IPUMS *chsurv* For women age 12+ (otherwise, NIU) 0 to 20 (20 = 20+)
births_12m [0] *	Births last year	Recoded from IPUMS *birthslyr* For women age 12+ (otherwise, NIU) 0 to 4
indigenous	Indigenous status	0 = Did not want to respond 1 = Indigeneous 2 = Not indigeneous 9 = Missing Not included in published dataset.

12 Assessment of the synthetic dataset

We assess four aspects of the synthetic dataset: safety, internal consistency, correlations, and comparison of the synthetic data with actual data at aggregated level.

12.1 Safety

The sources and pre-processing of the data used for training our models guarantee a high level of safety in the synthetic data. The training data is a combination of data from many sources, the variables they contain have been recoded and harmonized variables in multiple ways, and we re-sampled a small fraction of the original observations from the core datasets. Linking a synthetic observation to any of the sources is made almost impossible and highly uncertain by these very processes.

The approach we developed to generate the synthetic data adds a layer of protection. The approach is designed to ensure that no "data copying" occurs, i.e., that the data generation model does not reproduce full records from the training data. First, a procedure that assesses and controls the risk of overfitting is embedded in the model implementation. Second, a procedure assesses the closeness between the training data and the synthetic observations generated by the model. More information on these two safety measures is available in Solatorio and Dupriez (2023) where the REaLTabFormer model is described in more detail.[22]

12.2 Internal consistency checks

Internal consistency is guaranteed by the application of validators in the process. The validators automatically reject synthetic observations that violate any of the consistency checks embedded, as the model runs. By design, the number of observations in the synthetic population that violate any of the validation rules is 0.

12.3 Correlations

We compare the correlation between categorical variables in the actual data with the correlations in the synthetic data, using the following measure:

(actual correlation / synthetic correlation) – 1

Values close to 0 are ideal. The absolute value of the correlation difference is also measured. Figure 11 shows that for most variables, the measure is quasi-ideal. This also shows in the

[22] REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers. https://arxiv.org/abs/2302.02041

summarized version of the chart presented in Figure 12, which shows that most measures are close to 0.[23]

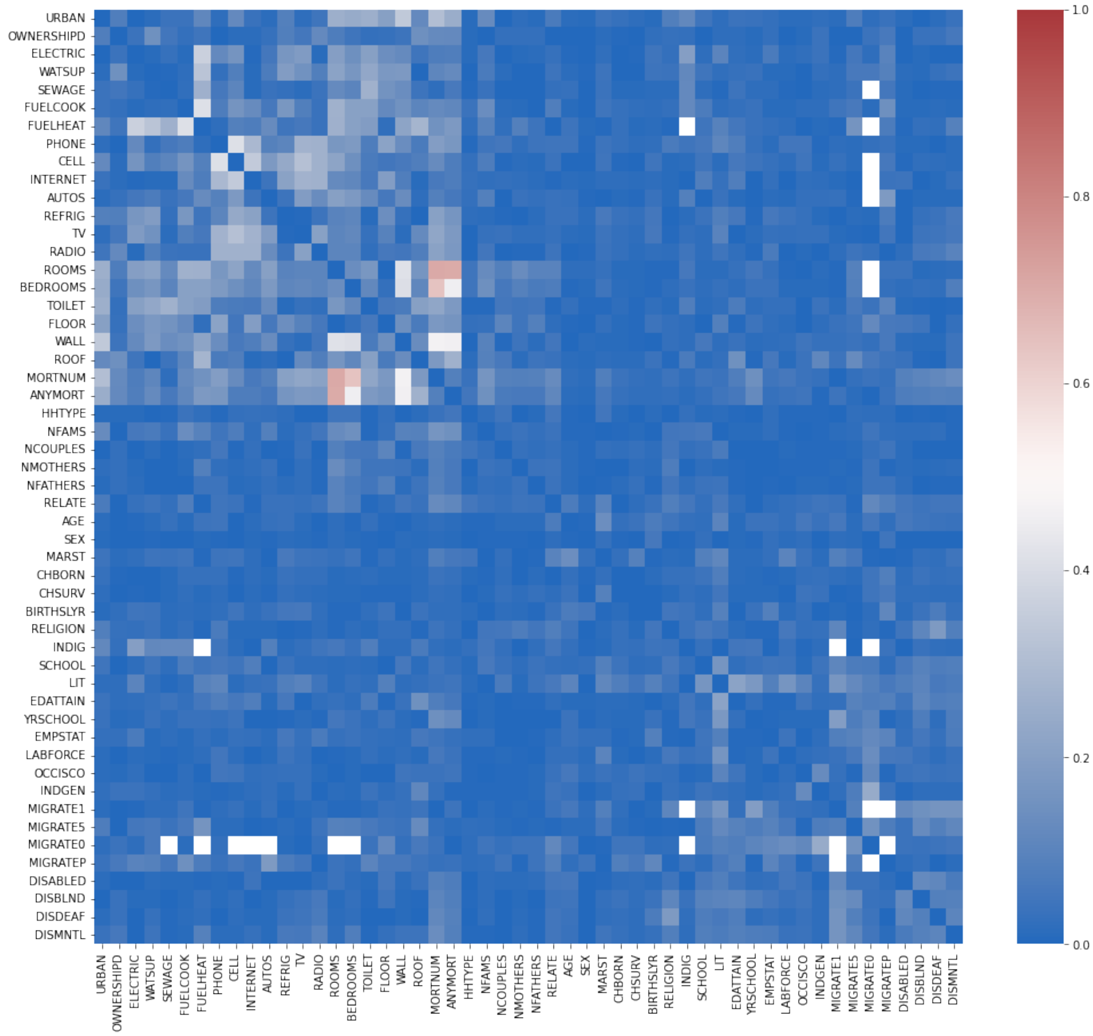

Figure 12 – Absolute value of (actual correlation / synthetic correlation) – 1

The scatter plot of the correlation ratio and difference gives more insight on the level of variation in the correlation. We find that most of the correlation of the variables in the synthetic data is as close as the correlation of the variables in the training data within the range -0.2 to 0.2 (Figure 13). Still, the graphs in Figure 14 show that some pairs of variables significantly differ in correlation values when computed using the synthetic data in comparison with the values derived from the empirical data. We list the top 20 variable pairs with the largest correlation ratios to get some clue on the variables for which the generative model fails to learn the inter-variable relationships well.

[23] Script: 001-Analyze IPUMS Dataset.ipynb

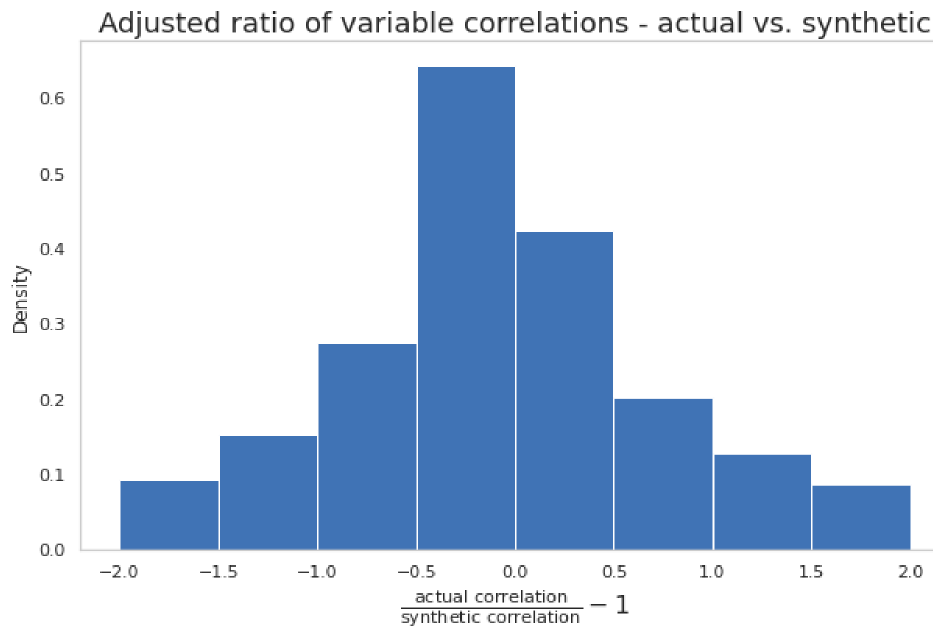

Figure 13 – Summary of the measure of correlation between categorical variables

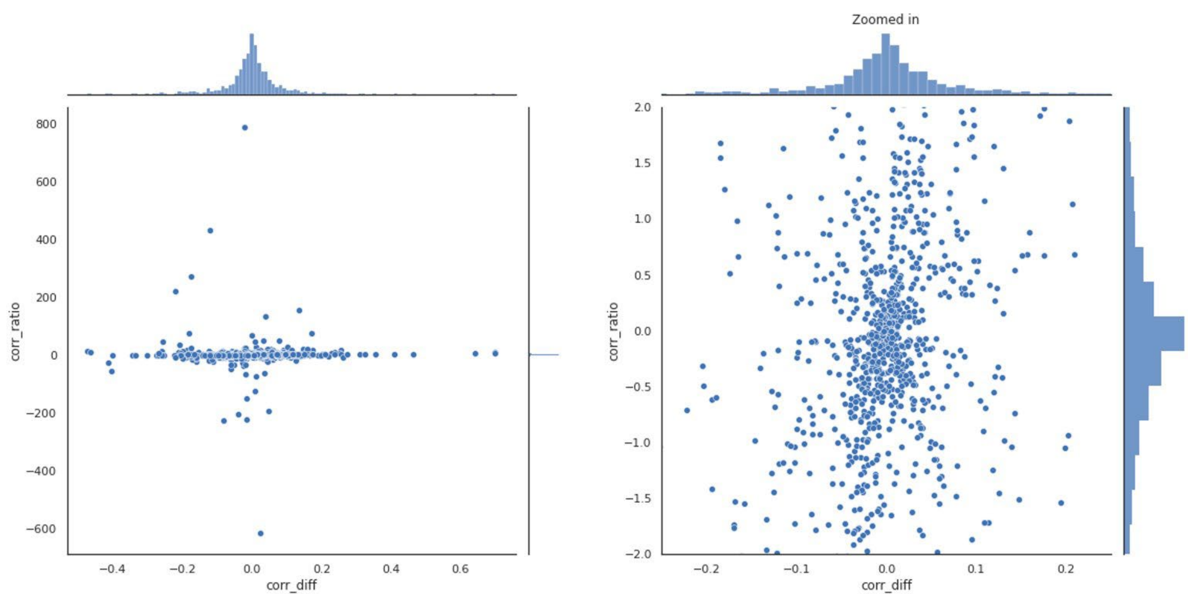

Figure 14 – Comparison of the correlation of variables between the empirical and the synthetic data. The graphs contain the absolute difference and the ratio of the correlations.

The top five pairs consist of variables that are weakly correlated. The correlation values range between -0.05 to 0.05 when computed from the empirical data. The synthetic data correlations are much lower for these pairs. Additionally, most of the variable pairs that have the largest errors, by ratio, when viewed intuitively are expected to have low correlations. For example, the variable pair ANYMORT::OCCISCO corresponds to "did any death occur in the household in the past year" and "occupation of the household member". Intuitively, we do not expect these variables to be strongly correlated. We suppose that the generative model prioritized learning strong correlations of other variables in the data.

One pair of variables that we intuitively expected to be correlated, i.e., the variable pair BEDROOMS::WALL (respectively "number of bedrooms", and "materials of the wall") was not learned by the model well.[24]

	corr_diff	corr_ratio	train_corr	samp_corr
INDGEN::MIGRATEP	-0.034135	-444.356344	-0.034058	0.000077
FLOOR::DISBLND	0.049916	266.681789	0.050104	0.000187
WALL::MIGRATEP	-0.057635	-257.575756	-0.057411	0.000224
INTERNET::RELIGION	-0.015525	230.898959	-0.015592	-0.000067
ANYMORT::BIRTHSLYR	0.026718	198.098844	0.026853	0.000135
MORTNUM::OCCISCO	0.035151	190.044345	0.035336	0.000185
TOILET::NMOTHERS	0.041561	189.816835	0.041780	0.000219
MORTNUM::YRSCHOOL	0.137428	-162.384451	0.136582	-0.000846
ANYMORT::NCOUPLES	-0.059012	-148.454577	-0.058615	0.000398
BEDROOMS::WALL	-0.396054	-136.161214	-0.393146	0.002909
TOILET::MORTNUM	-0.222046	128.489945	-0.223775	-0.001728
ANYMORT::OCCISCO	0.043410	-119.297991	0.043047	-0.000364
RADIO::ANYMORT	-0.170535	113.298461	-0.172041	-0.001505
SCHOOL::DISABLED	0.058430	89.034266	0.059086	0.000656
MORTNUM::DISBLND	-0.108251	-83.352698	-0.106952	0.001299
CELL::WALL	0.060080	75.590515	0.060875	0.000795
NMOTHERS::DISMNTL	-0.023305	-71.077751	-0.022977	0.000328
MORTNUM::DISDEAF	-0.120951	-68.764460	-0.119192	0.001759
INTERNET::CHSURV	0.015753	-67.827273	0.015520	-0.000232
BIRTHSLYR::DISMNTL	-0.024800	-57.698624	-0.024370	0.000430

In general, the pairs of variables for which the model did not perform well in capturing the relationships as measured by correlation are those that are not expected to have high correlations.

12.4 Aggregated data – Center and spread measures

The synthetic data we generated is representative of a specific country. It is not intended and may not be used for statistical inference. It is only intended to be used for simulation and training purposes.

[24] This may also in part be a consequence of the "hybrid" nature of the training data, which combined datasets from very different countries.

As we used data from multiple countries and sources for training our models, we cannot use similarity with a specific dataset to assess the quality of the synthetic data. We can however generate the frequencies for key categorical variables and compare them with the frequencies in the training datasets. For the continuous variables (household expenditures, and height and weight of children), we can compare means and distributions with the means and distributions of similar variables in actual country datasets (not expecting exact similarity, but to assess whether the means and spreads are reasonable). We present below a series of summary tables that confirm that the synthetic data is of high quality and fit for its intended purposes of training, simulation, and SDC assessment.

12.4.1 Distribution of households by size, and mean household size, urban/rural and by quintile

Per capita expenditure quintiles, national	Household size 1	2	3	4	5	6	7	8	9	10	Total
1	1,435 0.44	9,341 2.89	23,443 7.24	46,151 14.26	66,687 20.61	52,704 16.29	41,044 12.68	35,710 11.03	18,311 5.66	28,802 8.90	323,628 100.00
2	12,110 2.88	26,027 6.20	65,895 15.69	103,469 24.64	90,686 21.60	52,276 12.45	31,678 7.55	20,199 4.81	8,751 2.08	8,762 2.09	419,853 100.00
3	27,601 5.70	57,925 11.96	87,496 18.06	130,313 26.90	92,859 19.17	44,674 9.22	23,068 4.76	12,028 2.48	4,801 0.99	3,738 0.77	484,503 100.00
4	57,548 10.14	100,812 17.76	135,756 23.91	137,768 24.27	76,903 13.55	33,244 5.86	14,094 2.48	6,986 1.23	2,550 0.45	2,033 0.36	567,694 100.00
5	144,918 20.52	176,171 24.95	173,368 24.55	123,994 17.56	57,642 8.16	18,156 2.57	7,135 1.01	3,232 0.46	934 0.13	527 0.07	706,077 100.00
Total	243,612 9.74	370,276 14.80	485,958 19.42	541,695 21.65	384,777 15.38	201,054 8.04	117,019 4.68	78,155 3.12	35,347 1.41	43,862 1.75	2,501,755 100.00

	Mean
Per capita expenditure quintiles, national	
1	6.027031
2	4.745775
3	4.12344
4	3.521822
5	2.8325
Total	3.973268

Residence (urban/rural)	Household size 1	2	3	4	5	6	7	8	9	10	Total
Rural	82,108 7.31	136,988 12.20	202,418 18.02	242,156 21.56	188,612 16.79	108,152 9.63	65,374 5.82	47,780 4.25	21,711 1.93	27,846 2.48	1,123,145 100.00
Urban	161,504 11.71	233,288 16.92	283,540 20.57	299,539 21.73	196,165 14.23	92,902 6.74	51,645 3.75	30,375 2.20	13,636 0.99	16,016 1.16	1,378,610 100.00
Total	243,612 9.74	370,276 14.80	485,958 19.42	541,695 21.65	384,777 15.38	201,054 8.04	117,019 4.68	78,155 3.12	35,347 1.41	43,862 1.75	2,501,755 100.00

	Mean
Residence (urban/rural)	
Rural	4.307236
Urban	3.701186
Total	3.973268

12.4.2 Percentage of female headed households, urban/rural

Sex of head	Freq.	Percent	Cum.
Male	1,909,091	76.31	76.31
Female	592,664	23.69	100.00
Total	2,501,755	100.00	

12.4.3 Mean age, urban/rural

Variable	Obs	Mean	Std. dev.	Min	Max
age	10,003,891	27.60168	20.06089	0	100

-> urbrur = Rural

Variable	Obs	Mean	Std. dev.	Min	Max
age	4,879,725	26.32096	20.12738	0	100

-> urbrur = Urban

Variable	Obs	Mean	Std. dev.	Min	Max
age	5,124,166	28.8213	19.92097	0	100

12.4.4 Population by sex, urban/rural

Sex	Residence (urban/rural) Rural	Urban	Total
Male	2,436,797 49.94	2,490,235 48.60	4,927,032 49.25
Female	2,442,928 50.06	2,633,931 51.40	5,076,859 50.75
Total	4,879,725 100.00	5,124,166 100.00	10,003,891 100.00

12.4.5 Age dependency ratio, by quintile

Quintile	Dependency ratio
1	1.01
2	0.73
3	0.60
4	0.50
5	0.37
All	0.61

12.4.6 Literacy (age 15+), urban/rural, by sex, and by quintile

Residence (urban/rural)	Literacy status Yes	No	Total
Rural	2,370,522 75.67	762,260 24.33	3,132,782 100.00
Urban	3,341,294 91.91	294,228 8.09	3,635,522 100.00
Total	5,711,816 84.39	1,056,488 15.61	6,768,304 100.00

Sex	Literacy status Yes	No	Total
Male	2,876,463	413,423	3,289,886
	87.43	12.57	100.00
Female	2,835,353	643,065	3,478,418
	81.51	18.49	100.00
Total	5,711,816	1,056,488	6,768,304
	84.39	15.61	100.00

Per capita expenditure quintiles, national	Literacy status Yes	No	Total
1	677,346	394,281	1,071,627
	63.21	36.79	100.00
2	972,359	277,903	1,250,262
	77.77	22.23	100.00
3	1,174,104	190,758	1,364,862
	86.02	13.98	100.00
4	1,339,134	135,452	1,474,586
	90.81	9.19	100.00
5	1,548,873	58,094	1,606,967
	96.38	3.62	100.00
Total	5,711,816	1,056,488	6,768,304
	84.39	15.61	100.00

12.4.7 School attendance for ages 6 to 15, urban/rural and by quintile

"No, atten" = "No, attended in the past" ; "No, not s" = "No, not specified if ever attended"

Sex	School attendance Yes	No, never	No, atten	No, not s	Total
Male	925,893 86.92	60,836 5.71	20,475 1.92	58,063 5.45	1,065,267 100.00
Female	903,159 87.36	55,381 5.36	18,714 1.81	56,608 5.48	1,033,862 100.00
Total	1,829,052 87.13	116,217 5.54	39,189 1.87	114,671 5.46	2,099,129 100.00

Residence (urban/rural)	School attendance Yes	No, never	No, atten	No, not s	Total
Rural	910,308 81.83	89,858 8.08	20,507 1.84	91,727 8.25	1,112,400 100.00
Urban	918,744 93.11	26,359 2.67	18,682 1.89	22,944 2.33	986,729 100.00
Total	1,829,052 87.13	116,217 5.54	39,189 1.87	114,671 5.46	2,099,129 100.00

Per capita expenditure quintiles, national	School attendance Yes	No, never	No, atten	No, not s	Total
1	428,751 74.33	72,026 12.49	13,331 2.31	62,684 10.87	576,792 100.00
2	413,044 85.99	27,644 5.76	9,979 2.08	29,662 6.18	480,329 100.00
3	394,024 92.77	9,943 2.34	7,804 1.84	12,951 3.05	424,722 100.00
4	334,016 95.13	4,965 1.41	5,437 1.55	6,702 1.91	351,120 100.00
5	259,217 97.39	1,639 0.62	2,638 0.99	2,672 1.00	266,166 100.00
Total	1,829,052 87.13	116,217 5.54	39,189 1.87	114,671 5.46	2,099,129 100.00

12.4.8 Access to electricity, urban/rural and by quintile

Residence (urban/rural)	Electricity Yes	Electricity No	Total
Rural	399,840 35.60	723,305 64.40	1,123,145 100.00
Urban	63,142 4.58	1,315,468 95.42	1,378,610 100.00
Total	462,982 18.51	2,038,773 81.49	2,501,755 100.00

- quintiles, national	Electricity Yes	Electricity No	Total
1	206,344 63.76	117,284 36.24	323,628 100.00
2	147,984 35.25	271,869 64.75	419,853 100.00
3	65,301 13.48	419,202 86.52	484,503 100.00
4	33,860 5.96	533,834 94.04	567,694 100.00
5	9,493 1.34	696,584 98.66	706,077 100.00
Total	462,982 18.51	2,038,773 81.49	2,501,755 100.00

12.4.9 Average years of schooling by age group and sex

	Sex Male	Sex Female	Total
Age group			
20-29	6.687116	6.318169	6.49536
30-39	7.008762	6.131592	6.560185
40-49	6.823653	5.307524	6.04464
50-59	4.991733	4.005016	4.497198
60+	3.438855	2.731709	3.061209
Total	6.118915	5.263057	5.67712

12.4.10 Gini coefficient and other inequality indicators

The inequality indicators calculated for this combined dataset, as well as the consumption profiles by urban/rural area of residence, are consistent with what can be expected from a middle-income country. As a comparison, the Gini coefficient in our dataset (0.37) is close to the Gini coefficient of Indonesia 2012 (0.38) and India 2019 (0.36).[25]

Percentile ratios

All obs	p90/p10	p90/p50	p10/p50	p75/p25
	5.021	2.461	0.490	2.374

Generalized Entropy indices GE(a), where a = income difference sensitivity parameter, and Gini coefficient

All obs	GE(-1)	GE(0)	GE(1)	GE(2)	Gini
	0.24905	0.21719	0.23147	0.30998	0.36511

Atkinson indices, A(e), where e > 0 is the inequality aversion parameter

All obs	A(0.5)	A(1)	A(2)
	0.10624	0.19522	0.33249

12.4.11 Consumption by main category of products/services, urban/rural and quintile

The structure of household expenditure is also in-line with what is found in other countries.

Synthetic data	Rural	Urban	National
Food and non-alcoholic beverages	51%	36%	43%
Alcoholic beverages, tobacco, and narcotics	3%	2%	2%
Clothing and footwear	6%	5%	5%
Housing, water, electricity, gas, and other fuels	13%	24%	19%
Furnishing, household equipment and routine household maintenance	4%	4%	4%
Health	4%	3%	3%
Transport	6%	7%	7%
Communication	2%	3%	3%
Recreation and culture	2%	3%	2%
Education	2%	4%	3%
Catering and accommodation services	2%	4%	3%
Miscellaneous goods and services	5%	5%	5%
	100%	100%	100%

[25] Source: World Bank, World Development Indicators (https://data.worldbank.org/indicator/SI.POV.GINI)

Bangladesh 2010			Indonesia 2002		
Rural	Urban	National	Rural	Urban	National
55%	47%	51%	61%	46%	52%
3%	2%	3%	8%	6%	7%
5%	5%	5%	4%	4%	4%
12%	18%	15%	12%	20%	17%
5%	5%	5%	2%	3%	3%
3%	2%	3%	2%	2%	2%
4%	5%	4%	2%	4%	3%
2%	3%	2%	0%	2%	1%
2%	3%	2%	1%	2%	2%
3%	5%	4%	1%	2%	2%
2%	2%	2%	2%	5%	4%
4%	4%	4%	4%	4%	4%

Ethiopia 2010			Philippines 2010		
Rural	Urban	National	Rural	Urban	National
52%	37%	41%	47%	34%	40%
6%	2%	3%	2%	1%	2%
6%	6%	6%	3%	2%	3%
21%	29%	27%	18%	24%	21%
6%	5%	5%	3%	3%	3%
1%	1%	1%	3%	3%	3%
2%	4%	4%	7%	8%	8%
1%	4%	3%	2%	3%	3%
0%	1%	1%	1%	1%	1%
0%	1%	1%	3%	4%	4%
4%	6%	6%	5%	9%	7%
3%	3%	3%	7%	7%	7%

12.4.12 Histograms of log per capita expenditure

The distribution of the log per capita household total expenditure in the synthetic data does not have a perfect bell shape (figures 15 to 17). The urban distribution in particular shows a multi-modal distribution, which would unlikely be found in a country dataset. This is due to the "hybrid" nature of our training dataset, which contain data for significantly diverse populations and is obtained by merging national samples of very different sizes. Although the overall distribution of the per capita expenditure is "normal", we cannot expect predicted values to result in a similar ideal shape. This is not considered as a problem in our synthetic data, as it does not affect its utility as a straining and SDC simulation dataset.

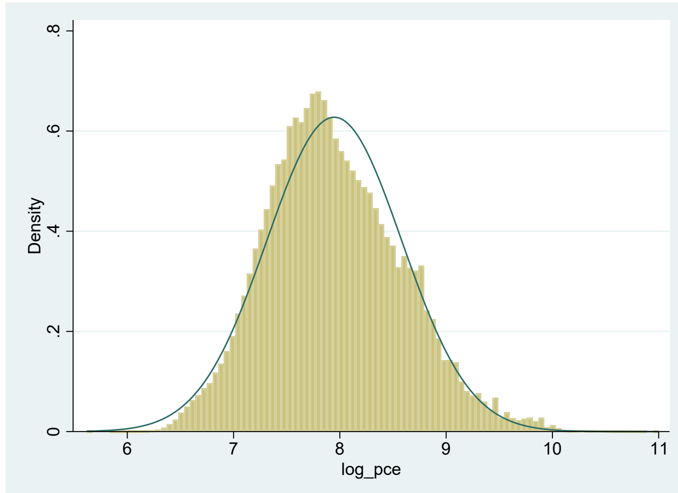

Figure 15 - Log per capita expenditure, national

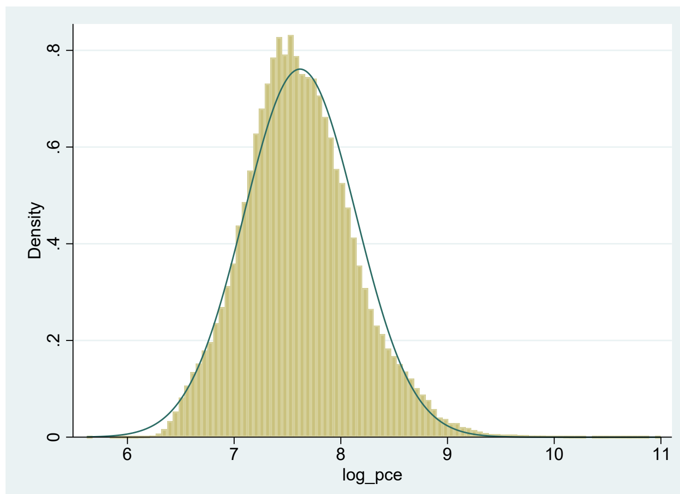

Figure 16 - Log per capita expenditure, rural

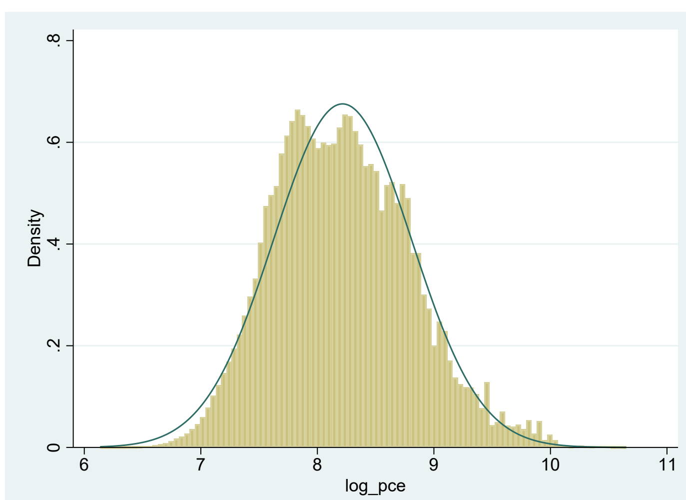

Figure 17 - Log per capita expenditure, urban

13 Post-processing and dissemination

The dataset as generated by the process described above contains some "anomalies". First, it may contain some inconsistencies. Although we implemented a set of validators to reject observations with anomalies, this set is not exhaustive. Also, the training data, although edited, may contain some errors which could have been "learned" and thus replicated by the model. Ideally, a thorough control and editing of the training data should be the first step in the process of synthetic data generation. Last, the IPUMS data contain some values that would typically not be found in a real country dataset. For example, the individual-level variable "*school_yrs*" (number of years of schooling) accepts values from 0 to 18, and 90 to 99. The values 90 to 99 represent "unknown" or missing responses. In a country dataset, unknown or non-applicable would have one or two values, not 10.

We therefore implemented some post-processing procedures to "clean" the datasets intended for dissemination. This post-processing did not significantly change the data. The procedures were limited to a few global recoding (for examples, recoding all values from 90 to 99 into missing for the number of years of schooling), or setting values to "missing" in cases where a value was imputed but a "non-applicable" code was expected (for example, the education attainment for a 2-year-old person). We also added variable and value labels in Stata.

The resulting dataset is available as open data (published under a CC-BY 4.0 license) in the World Bank Microdata Library (in English and in French). See links in footnote of page 1 of this document.

14 Sample open dataset for training

The core "census" dataset represents the full population of the imaginary country. Out of this large dataset, a sample of 8,000 households (32,396 individuals) was selected. Enumeration areas were first selected in strata (by geo1 and urbrur) then a fixed number of households (25) was randomly selected from each selected enumeration area. The R script used to extract the sample and generate the sample weights is available in Annex 2.

The sample "survey" datasets are also made publicly available. For training purposes, a "fake questionnaire" was generated for this dataset.

Annex 1 - References and links

References

- Solatorio, Aivin V. and Olivier Dupriez. 2023. REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers. Available at https://arxiv.org/abs/2302.02041

Links

- REaLTabFormer github repository: https://github.com/worldbank/REaLTabFormer
- IPUMS International: https://international.ipums.org/international/
- Demographic and Health Surveys (DHS Program): https://dhsprogram.com/
- World Bank, Global Consumption Database 2010: https://microdata.worldbank.org/index.php/catalog/4424

Annex 2 – R script for sample extraction

```
## ---------------------------
##
## Script name: sample_synth_data.R
##
## Purpose of script: Create sample of synthetic dataset for use in documentation
##                    and anonymization training
## Author: Thijs Benschop
##
## Date Created: 2023-04-06
## Date updated: 2023-05-05 - recalibrate weights within strata
##                          - add variable labels
## ---------------------------
##
## Notes:
##   - Draw stratified sample of 8,000 households
##   - Two-stage sample: first stage stratified by geo1 and urbrur, second stage fixed number of households at random
##   - Input:  Synthetic household and individual census datasets
##   - Output: Household and individual level datasets
## ---------------------------

rm(list = ls())

# Set wd
setwd("___your path ___")

## Load packages
library(haven)
library(data.table)
library(sampling)
library(dplyr)
#library(reldist)
#library(survey)
```

```
#### Function for 2-stage sampling, first stage stratified, second stage fixed number n2  ####
# size      - sample size to be drawn (if size in [0,100] -> size is p %, if size > 100, size is n)
# ea_var    - variable specifying units to sample in stage 1
# strat_var - variable for stratifying first stage (needs to be 1 variable) - sample is drawn proportionally
# n2        - number of households sampled in each ea
# dat       - data.table with variable hid
# seed      - seed for random number generation to replicate samples

# for testing:
# size = 8000
# ea_var = "ea"
# strat_var = "stratvar" # stratify by both urbrur and geo1
# n2 = 25
# dat = synth_data_hh
# seed = 123

two_stage_sample <- function(size, ea_var, strat_var, n2, dat, seed = 123){
  set.seed(seed) # see for replicability

  # Calculate number of ea to be sampled based on size and n2
  if(size <= 0){
    break("Size cannot be negative or 0")
  }else if (size <= 100){
    size <- round(size * nrow(dat))
  }
  number_of_ea <- round(size / n2) # note that size should be a multiple of n2 to have exact results

  # List of eas
  dat_ea <- dat[, c(ea_var, strat_var), with = FALSE]
  dat_ea <- subset(unique(dat_ea)) # all records within same ea have no variation in strat_var

  # Size of sample per strata (proportional to size of strata)
  dat_ea <- dat_ea[order(dat_ea[, strat_var, with=FALSE])] # order list of eas by strata
  number_of_ea_by_strata <- round(number_of_ea * as.numeric(table(dat[, strat_var, with=FALSE]))/nrow(dat))

  # Correction: if total in number_of_ea_by_strata doesn't add up to number_of_ea due to rounding
  # add/substract from largest strata difference between sum(number_of_ea_by_strata) and number_of_ea
  if(sum(number_of_ea_by_strata) != number_of_ea){
    pos_to_update <- which(number_of_ea_by_strata == max(number_of_ea_by_strata))[1] # first of largest strata
    number_of_ea_by_strata[1] <- number_of_ea_by_strata[1] + (number_of_ea - sum(number_of_ea_by_strata))
  }
```

```
# Sample eas
sample_1 <- sampling::strata(data = dat_ea,
                   stratanames = strat_var,
                   size = number_of_ea_by_strata,
                   method = "srswor",
                   description = TRUE)

sample_1 <- cbind(dat_ea[sample_1$ID_unit, ea_var, with = FALSE], sample_1$Prob)
colnames(sample_1) <- c(colnames(sample_1)[1], "eaweight")

# Merge sample_1 and dat, selecting only selected eas
dat_selected <- merge.data.table(dat, sample_1, by = ea_var, all.x = FALSE, all.y = TRUE)

# Sample n2 households in each ea from dat_selected
#dat_selected

sample_2 <- sampling::strata(data = dat_selected,
                   stratanames = ea_var,
                   size = rep(n2, sum(number_of_ea_by_strata)),
                   method = "srswor",
                   description = TRUE)

sample_2 <- cbind(dat_selected[sample_2$ID_unit,], sample_2$Prob)

# Calculate weight
sample_2[, hhweight := 1/(eaweight * V2)]
sample_2[, V2 := NULL]
sample_2[, eaweight := NULL]

# Final weight adjustment within strata
num_obs_pop_by_strata <- dat %>% count(by = eval(stratvar))
colnames(num_obs_pop_by_strata) <- c(strat_var, "n")
num_obs_sample_by_strata <- sample_2 %>% group_by(eval(stratvar)) %>% summarise((sum = sum(hhweight)))
colnames(num_obs_sample_by_strata) <- c(strat_var, "sum_w")
num_obs_comb <- merge(num_obs_pop_by_strata, num_obs_sample_by_strata, by = strat_var)
num_obs_comb[, weight_factor := n/sum_w] # adjustment factor by strata

sample_2 <- merge(sample_2, num_obs_comb[, c(strat_var, "weight_factor"), with = FALSE], by = strat_var)
sample_2[, hhweight := hhweight * weight_factor]
#sample_2[, hhweight := hhweight *  (nrow(dat) / sum(hhweight))]
```

```
  sample_2[, .(hid, hhweight)] # Return hid and hhweight
}

#### Read in census data ####
synth_data_hh  <- as.data.table(read_dta("./training_data_household_census.dta"))
synth_data_ind <- as.data.table(read_dta("./training_data_individual_census.dta"))

dim(synth_data_hh) # 2,501,755 hhs
length(unique(synth_data_hh$geo1)) # 10 geo1
length(unique(synth_data_hh$geo2)) # 61 geo2
length(unique(synth_data_hh$ea))   # 5,940 eas

dim(synth_data_ind) # 10,003,891 individuals

colnames_hh  <- colnames(synth_data_hh)
colnames_ind <- colnames(synth_data_ind)

##### Draw hh sample #####
## Merge hh and ind level population files
synth_population <- merge(synth_data_hh, synth_data_ind, by = "hid")
rm(synth_data_ind) # remove ind data, as in synth_population

# Create stratification variable for both geo1 and urbrur
synth_data_hh[, stratvar := geo1 * 10 + urbrur]
table(synth_data_hh$stratvar)
# 20 strata, smallest strata only 14,036 hhs

## Sample 1: Two-stage sample, n = 8,000, 25 in each ea
sample_1 <- two_stage_sample(size = 8000,
                             ea_var = "ea",
                             strat_var = "stratvar", # stratify by both urbrur and geo1
                             n2 = 25,
                             dat = synth_data_hh,
                             seed = 123)

# sample_1 only contains selected hids and weight
length(unique(sample_1$hid))

# save selected hid
saveRDS(sample_1, file = "sampled_hhs.rds")
```

```
# select sample from pop
sample_1_dat  <- right_join(synth_population, sample_1, by = "hid")

# #### Replace hid with numeric hid ####
# problem with precisionnof float numbers > 16,777,215
# # hid includes information on geo2 -> intentionally
# setorder(sample_1_dat, cols = "geo1", "geo2")
# sample_1_dat[, hid_numeric_ea := rleid(hid), by = c("geo1", "geo2")]
# max(sample_1_dat$hid_numeric_ea)
# sample_1_dat[, hid_numeric    := 1000 * geo2 + hid_numeric_ea]
#
# #View(sample_1_dat[, .(geo1, geo2, hid, hid_numeric, hid_numeric_ea)])
#
# # save mapping hid and hid_numeric (idno in both pop and sample the same)
# saveRDS(unique(sample_1_dat[,.(hid, hid_numeric )]), "hid_mapping.rds")
#
# # keep only numeric hid
# sample_1_dat[, hid := hid_numeric]
# sample_1_dat[, ':='(hid_numeric = NULL, hid_numeric_ea = NULL)]

#### Checks on variables ####
colnames(sample_1_dat)

## Geo areas
# Not all geo2 areas are sampled
table(sample_1_dat$geo2) #  missing 9, 21, 32

# Proportionate in geo1
round(100 * table(sample_1_dat$geo1) / table(synth_population$geo1), digits = 2)

## Weights
# Sum of weights by hh -> 2,501,755 == number of hhs in pop
sample_1_dat[, .SD[1,], by = hid][, sum(hhweight)]

# Sum of weights at pop level -> 10,092,120 != pop size (10,003,891)
sample_1_dat[, sum(hhweight)]

# Weighted number of ind and hh by geo1
setorder(synth_population, cols = "geo1", "geo2")
```

```
geo1_dist <- cbind(sample_1_dat[, sum(hhweight), by=.(geo1)],
                   synth_population[, .N, by=.(geo1)]) #ind
geo1_dist$prop <- geo1_dist$V1 / geo1_dist$N
geo1_dist

geo1_dist_hh <- cbind(sample_1_dat %>% distinct(hid, .keep_all = T) %>%
  group_by(geo1) %>% summarise(sum(hhweight)),
synth_data_hh  %>%
  group_by(geo1) %>% summarise(n()))

geo1_dist_hh$prop <- geo1_dist_hh$`sum(hhweight)` / geo1_dist_hh$`n()`
geo1_dist_hh

# Weighted number of ind and hh by geo2 -> not stratified/representative at geo2 level
# setorder(synth_population, cols = "geo1", "geo2")
#
# geo2_dist <- cbind(sample_1_dat[, sum(hhweight), by=.(geo2)],
#                    synth_population[, .N, by=.(geo2)]) #ind
# geo2_dist$prop <- geo2_dist$V1 / geo2_dist$N
# geo2_dist
#
# sample_1_dat %>% distinct(hid, .keep_all = T) %>%
#   group_by(geo1) %>% summarise(sum(hhweight))
# synth_data_hh  %>%
#   group_by(geo1) %>% summarise(n())
#
# geo1_dist_hh <- cbind(sample_1_dat %>% distinct(hid, .keep_all = T) %>%
#                         group_by(geo1) %>% summarise(sum(hhweight)),
#                       synth_data_hh  %>%
#                         group_by(geo1) %>% summarise(n()))
#
# geo1_dist_hh$prop <- geo1_dist_hh$`sum(hhweight)` / geo1_dist_hh$`n()`
# geo1_dist_hh

# Compare hhsize
#sample_1_dat[, ]

#### Recalculate sample quintiles ####
# Need to weigh by hhweight, but can leave out hhsize when done on sample_1_dat

# 1) Sort by pc_exp (per capita expenditures)
```

```
setorder(sample_1_dat, cols = "pc_exp")

# 2) Generate cumulative sum of all weights
sample_1_dat[, cum_weight        := cumsum(hhweight)]
sample_1_dat[, cum_weight_urbrur := cumsum(hhweight), by = urbrur] # urbrur
sample_1_dat[, cum_weight_urb    := cum_weight_urbrur] # urb
sample_1_dat[, cum_weight_rur    := cum_weight_urbrur] # rur
sample_1_dat[urbrur == 1, cum_weight_urb    := NA] # set urban to NA if rural
sample_1_dat[urbrur == 2, cum_weight_rur    := NA] # set rural to NA if urban

# 3) Quintile cut off points and pc_exp values
cut_offs <- 1:4 * (max(sample_1_dat$cum_weight) / 5) # cut off points
cut_offs

cut_offs_urb <- 1:4 * (max(sample_1_dat %>% filter(urbrur == 2) %>% select(cum_weight_urb)) / 5) # cut off points
urban
#cut_offs_urb <- 1:4 * (max(sample_1_dat$cum_weight_urb, na.rm = T) / 5) # cut off points urban
cut_offs_urb
cut_offs_rur <- 1:4 * (max(sample_1_dat %>% filter(urbrur == 1) %>% select(cum_weight_rur)) / 5) # cut off points
rural
#cut_offs_rur <- 1:4 * (max(sample_1_dat$cum_weight_rur, na.rm = T) / 5) # cut off points urban
cut_offs_rur

# national
pc_exp_vals <- c(sample_1_dat[min(which(sample_1_dat$cum_weight > cut_offs[1])), pc_exp],
                 sample_1_dat[min(which(sample_1_dat$cum_weight > cut_offs[2])), pc_exp],
                 sample_1_dat[min(which(sample_1_dat$cum_weight > cut_offs[3])), pc_exp],
                 sample_1_dat[min(which(sample_1_dat$cum_weight > cut_offs[4])), pc_exp])
pc_exp_vals
sample_1_dat[, quint_nat_new := fcase(pc_exp < pc_exp_vals[1], 1,
                                      pc_exp < pc_exp_vals[2], 2,
                                      pc_exp < pc_exp_vals[3], 3,
                                      pc_exp < pc_exp_vals[4], 4,
                                      pc_exp >= pc_exp_vals[4], 5)]

table(sample_1_dat$quint_nat_new)
table(sample_1_dat$quint_nat_new, sample_1_dat$quint_nat)

# rural (value 1)
pc_exp_vals_rur <- c(sample_1_dat[min(which(sample_1_dat$cum_weight_urbrur > cut_offs_rur[1] & sample_1_dat$urbrur ==
1)), pc_exp],
```

```
                     sample_1_dat[min(which(sample_1_dat$cum_weight_urbrur > cut_offs_rur[2] & sample_1_dat$urbrur ==
1)), pc_exp],
                     sample_1_dat[min(which(sample_1_dat$cum_weight_urbrur > cut_offs_rur[3] & sample_1_dat$urbrur ==
1)), pc_exp],
                     sample_1_dat[min(which(sample_1_dat$cum_weight_urbrur > cut_offs_rur[4] & sample_1_dat$urbrur ==
1)), pc_exp])
pc_exp_vals_rur

sample_1_dat[, quint_rur_new := fcase(pc_exp < pc_exp_vals_rur[1], 1,
                                      pc_exp < pc_exp_vals_rur[2], 2,
                                      pc_exp < pc_exp_vals_rur[3], 3,
                                      pc_exp < pc_exp_vals_rur[4], 4,
                                      pc_exp >= pc_exp_vals_rur[4], 5)]

sample_1_dat[urbrur == 2, quint_rur_new := NA] # set all urban to missing (NA)

table(sample_1_dat$quint_rur_new, useNA = "always")
table(sample_1_dat$quint_rur_new, sample_1_dat$quint_urb, useNA = "always")

# urban (value 2)
pc_exp_vals_urb <- c(sample_1_dat[min(which(sample_1_dat$cum_weight_urb > cut_offs_urb[1] & sample_1_dat$urbrur ==
2)), pc_exp],
                     sample_1_dat[min(which(sample_1_dat$cum_weight_urb > cut_offs_urb[2] & sample_1_dat$urbrur ==
2)), pc_exp],
                     sample_1_dat[min(which(sample_1_dat$cum_weight_urb > cut_offs_urb[3] & sample_1_dat$urbrur ==
2)), pc_exp],
                     sample_1_dat[min(which(sample_1_dat$cum_weight_urb > cut_offs_urb[4] & sample_1_dat$urbrur ==
2)), pc_exp])
pc_exp_vals_urb

sample_1_dat[, quint_urb_new := fcase(pc_exp < pc_exp_vals_urb[1], 1,
                                      pc_exp < pc_exp_vals_urb[2], 2,
                                      pc_exp < pc_exp_vals_urb[3], 3,
                                      pc_exp < pc_exp_vals_urb[4], 4,
                                      pc_exp >= pc_exp_vals_urb[4], 5)]

sample_1_dat[urbrur == 1, quint_urb_new := NA] # set all rural to missing (NA)

table(sample_1_dat$quint_urb_new, useNA = "always")
table(sample_1_dat$quint_urb_new, sample_1_dat$quint_urb, useNA = "always")
```

```
# Check quintiles in ind sample (summing weights)
sample_1_dat[, sum(hhweight), by = quint_nat_new]
sample_1_dat[, sum(hhweight), by = quint_urb_new]
sample_1_dat[, sum(hhweight), by = quint_rur_new]
sample_1_dat[, sum(hhweight), by = urbrur]

# Replace old values with new values and drop newly created vars
# Cut in two as to not replace attributes
sample_1_dat[1:nrow(sample_1_dat), quint_nat := quint_nat_new]
sample_1_dat[1:nrow(sample_1_dat), quint_urb := quint_urb_new]
sample_1_dat[1:nrow(sample_1_dat), quint_rur := quint_rur_new]

sample_1_dat[ ,`:=`(quint_nat_new = NULL,
                    quint_urb_new = NULL,
                    quint_rur_new = NULL,
                    cum_weight = NULL,
                    cum_weight_urbrur = NULL,
                    cum_weight_rur = NULL,
                    cum_weight_urb = NULL)]
# Reorder data
setorderv(sample_1_dat, cols = c("geo2", "hid", "idno"))

#### Export data ####
# subset vars for distribution
# hh file
keepvars_h <- colnames_hh[!colnames_hh == "stratvar"]
keepvars_h <- c(keepvars_h, "hhweight") # add hhweight
sample_1_dat_hh <- sample_1_dat[,.SD[1], by = "hid"][, keepvars_h, with = FALSE]
sample_1_dat_hh[, popweight := hhsize * hhweight]

# Add variable names
attributes(sample_1_dat_hh$hid) <- list(label = "Unique household identifier",
                                        format.stata = "%12.0g")
attributes(sample_1_dat_hh$hhweight) <- list(label = "Household weight",
                                             format.stata = "%12.0g")
attributes(sample_1_dat_hh$popweight) <- list(label = "Population weight",
                                             format.stata = "%12.0g")
dim(sample_1_dat_hh)
colnames(sample_1_dat_hh)
write_dta(sample_1_dat_hh, "training_survey_data_hh.dta")
```

```
# ind file
keepvars_i <- c(colnames_ind, "hhweight")
sample_1_dat_ind <- sample_1_dat[, keepvars_i, with = FALSE]

# Add variable names
attributes(sample_1_dat_ind$hid) <- list(label = "Unique household identifier",
                                         format.stata = "%12.0g")
attributes(sample_1_dat_ind$hhweight) <- list(label = "Household weight",
                                             format.stata = "%12.0g")

dim(sample_1_dat_ind)
colnames(sample_1_dat_ind)
write_dta(sample_1_dat_ind, "training_survey_data_ind.dta")

# Check quintiles in hh sample
sample_1_dat_hh[, sum(popweight), by = quint_nat]
sample_1_dat_hh[, sum(popweight), by = quint_urb]
sample_1_dat_hh[, sum(popweight), by = quint_rur]

# Check min and max of quintiles
sample_1_dat_hh[, min(pc_exp), by = quint_nat]
sample_1_dat_hh[, max(pc_exp), by = quint_nat]
sample_1_dat_hh[, min(pc_exp), by = quint_urb]
sample_1_dat_hh[, max(pc_exp), by = quint_urb]
sample_1_dat_hh[, min(pc_exp), by = quint_rur]
sample_1_dat_hh[, max(pc_exp), by = quint_rur]
```