

LIS Self Teaching Package 2018

SPSS version

Part I: Inequality, poverty, and social policy



Part I

Inequality, poverty, and social policy

Overall Plan and Structure of the Exercise

The next seven exercises demonstrate the use of the LIS data. These exercises will lead you through the process of developing a comparative research project that examines inequality and poverty across countries. Each of the exercises introduces new concepts related to the datasets and the programming techniques needed to make use of the data. By the end of the last exercise, you will produce a complete program that returns results on poverty and inequality for a selection of six LIS countries.

Each exercise builds on the one that comes before it. It is intended that you will begin each new activity by returning to the program you have written in the previous exercise, and modifying it to satisfy the requirements of the new exercise. Each exercise contains questions that you can answer with the new results you produce. The solutions included with each exercise include an example program, with **bolded** sections indicating code that has been added for that exercise.

Unless otherwise instructed in the lesson, you should not remove variables or lines of code from your program, even if it is not needed for the current exercise. Variables and procedures that are introduced in earlier lessons may be needed later. In particular, all of the variables that are introduced in Exercise 2 will eventually be needed to complete the final exercise, even if they are not all required for the lessons in between.

The analysis shown here is simplified somewhat, compared to what might be done in an actual LIS Working Paper. Some choices have also been made in order to demonstrate particular aspects of the LIS data. However, these exercises provide a starting point for researchers who want to develop an analysis of the data based on their own research questions.

Research Questions

Since the beginning of the LIS project, one of the most prominent objects of research using the data has been the effect of government tax and transfer programs on poverty and inequality. The first substantive paper in the LIS

Working Papers series, published in 1985, analyzed the pre- and post-transfer poverty rate in Sweden, the United Kingdom, Israel, the United States, Norway, Canada, and West Germany. ¹ The second substantive paper compares the distribution of income across these same seven countries.²

States affect the income distribution through several different types of policy, which are captured in the LIS data:

- Progressive taxation
- Social insurance programmes linked to employment, such as public pensions, unemployment insurance, and sickness pay
- Universal benefits provided irrespective of employment, income or assets
- Social assistance benefits for especially needy individuals or households

The exercises in this section assess the impact of these policies in different countries on both poverty and income inequality. We will measure incomes due to labour market income, capital income, and private transfers, and compare this to incomes after income taxes and social insurance contributions as well as government transfers are accounted for. Within the category of government policies, we will separate the effect of payroll taxes, social insurance, and universal benefits on the one hand, and social assistance benefits on the other. We will measure the proportion of the population that is poor according to these different income measures, where poverty is defined relative to median level of income within a country. We will also compare income inequality using one of the most popular and longstanding inequality measures, the Gini coefficient.

As LIS has grown, the analysis of government policy, poverty, and inequality has been updated for more countries and more recent years.³ Recently, LIS expanded beyond the rich countries that have long made up the core of the project, and began adding data from middle income countries. This gives researchers the opportunity to compare income and poverty in these countries

1 Richard Hauser, Lee Rainwater, Martin Rein, Gaston Schaber, Timothy Smeeding. "Poverty in Major Industrialized Countries". LIS Working Paper No. 2 – Jul 1985. <http://www.lisdatacenter.org/wps/liswps/2.pdf>

2 Michael O'Higgins, Gunther Schmaus, Geoffrey Stephenson. "Income Distribution and Redistribution". LIS Working Paper No. 3 – Jun 1985. <http://www.lisdatacenter.org/wps/liswps/3.pdf>

3 For a recent example, see Timothy Smeeding, "Government Programs and Social Outcomes: The United States in Comparative Perspective". LIS Working Paper No. 426 – May 2005. <http://www.lisdatacenter.org/wps/liswps/426.pdf>

to the patterns seen in rich countries that have been much more heavily studied.

For this exercise, we will begin by analyzing the data from Guatemala, one of the recently-added middle income countries. We will then compare Guatemala to five other countries: the United States, Denmark, Poland, Hungary, and Israel. These specific countries have been chosen in part for pedagogical reasons. However, they will also produce substantively interesting comparisons, because they represent a wide range of national income levels and welfare state regime types. After completing the final exercise, you will be able to answer the following questions:

- Which of the six countries in the study have the highest levels of income inequality and poverty before taxes and transfers are accounted for? Does this change when taxes and transfers are included?
- Which type of government policy has a larger impact on inequality and poverty in each country: taxes, social insurance, and universal benefits, or targeted social assistance?

Contents

1. Accessing the LIS databases: The Job Submission Interface (JSI)

- Submit a LIS job and get basic descriptive statistics for one variable

2. Sample selection and weighting

- Introduce the list of variables to be used
- Select the sample and eliminate cases with missing data
- Produce weighted and unweighted descriptive statistics for variables

3. Working with household income variables: top and bottom coding and equivalence scales

- Top and bottom coding income data to remove outliers
- Correcting income for household size using equivalence scales

4. Inequality: The Gini Index

- Calculating the Gini coefficient

5. Relative poverty rates

- Calculating relative poverty

6. Comparing income concepts

- Introducing three concepts of income
- Inequality and poverty before and after taxes and transfers

7. Comparing multiple countries

- Extending the analysis to multiple countries
- Working with net vs. gross income data

1. Accessing the LIS Database: The Job Submission Interface (JSI)

Goal

This exercise introduces the Job Submission Interface (JSI), which we will be using to work with LIS data in all of the subsequent exercises.

The JSI is a secure Java application that allows researchers to:

- write, submit and view job requests (and corresponding outputs);
- track the status of the job requests in process ('received', 'processing', 'set for review', 'refused', etc.); and
- access the history of all job requests ever sent.

In this exercise we will use the JSI to open a dataset and produce basic descriptive statistics.

Activity

Launch the Job Submission Interface (JSI) application and logon to it with your LIS account.

Submit a simple program to display descriptive statistics (number of valid observations, mean, minimum, maximum) for the household-level income variable **dhi**, for Guatemala 2006. **dhi**, or *disposable household income*, contains the total monetary and non-monetary current income for the household, net of income taxes and social security contributions. It is a harmonised variable that is available for all datasets.

Track the status of your job.

View the resulting listing.

Go to the Job Library window and discard the job; use the advanced search tool to get it back.

Question: How many valid observations (non-missing) are in this dataset for **dhi**?

Guidelines

- Once connected to the Job submission Interface, there are three main tasks that may be carried out:
 1. Submit jobs through the Job Session window.
 - Select a project (LIS, LWS).
 - Select a statistical package (SAS, **SPSS** or Stata or R).

- When submitting a job (Job Session window), always add a subject line.
- Write your code.
- Click on the submit button.

2. Work with Today's Jobs (Today Jobs window.)

- Watch the status of jobs currently sent to LISSY in the 'jobs in process' panel (top-left).
- View the jobs returned by LISSY.
- Click on a job in the 'jobs returned' panel (bottom-left).
- Click on the 'view job' button.
- Click on the 'job text' or 'listing' tabs, respectively, of the right panel to see the request and its output.
- Re-submit a selected job by clicking on the 'edit in job submission' button at the bottom-right of the window.

3. Manage (view, clean and search) all job requests ever sent in the Job Library window.

- View jobs sent over a specific time period.
- Clean the library by discarding useless job requests ('discard' button).
- Search jobs by keywords.
- Re-submit a selected job by clicking on the 'edit in job submission' button at the bottom-right of the window.

- When you open a LIS dataset, use the correct file reference for the country/year you wish to use. For example:

get file = gt06h

For more information about the syntax of country/year file reference, see the job submission instructions on the LIS web site (*Data Access → Job Submission*), accessed through this [link](#).

For a list of available data sets and their 2digit country codes, go to:

Our Data → LIS Database → List of Datasets, accessible through [LIS](#)

Our Data → LWS Database → List of Datasets, accessible through [LWS](#)

SPSS reminder: to run descriptive statistics, use **descriptive variables = <varlist>**

- If you do not receive your job in the expected amount of time, it means that there is a long queue of jobs on LISSY. In that case, resending your job, or sending several other ones while waiting to receive the first one, will only increase the queue and hence your waiting time. Remember to wait to get your results before sending a new job!

Program

get file = gt06h.

descriptive variables = dhi.

Results

	Number of valid observations	Mean	Minimum	Maximum
dhi	13,664	40,670	-186,010	2,729,298

Question: How many valid observations (non-missing) are in this dataset for **dhi**?

- There are 13,664 valid observations.

Comments

- It is important to pay close attention to sample sizes in order to make sure you have enough data to make stable estimates. When working with small datasets, or small sub-samples of datasets, always check the size of the sample underlying each statistic you have computed.
- As you can see in the results, the disposable household income **dhi** can be negative. This happens in cases where the data provider kept losses as such rather than applying bottom coding techniques. This typically happens with incomes from self-employment or capital income; in rare circumstances it happens that taxes are higher than gross income due to different income reference periods or miscalculation of taxes.

2. Sample selection and weighting

Goal

This exercise introduces the variables that we will be using in the rest of our analysis and concentrates on sample selection and the use of weights.

Sample selection – The final objective of this exercise is to compare incomes before and after government intervention. In order to be sure that the comparison is correct, users should ensure that they use exactly the same sample when calculating statistics for the pre- and post-government intervention. It is thus important to begin by selecting a sample which can be used for the entire analysis. For this reason, we will drop from the analysis not only cases which have missing values in the variable of interest to the specific statistic being calculated at each step, but all the cases that have a missing value in any of the variables of interest for the whole analysis.

Use of weights - Comparative researchers are typically interested in the characteristics of national populations, not the samples provided. It is very important to understand and use sample weights correctly in order to get representative results for the total underlying population. This exercise shows the differences in statistics between the unweighted sample and the weighted population.

Activity

As in the previous exercise, we will continue working with the Guatemala 2006 (GT06) data. Modify your previous program to select only the following variables in addition to **dhi**: unique country/year number (**did**), household weight (**hpopwgt**), number of household members (**nhhmem**), gross or net income information (**grossnet**), factor income (**factor**), work-related insurance transfers (**hitsi**), assistance benefit income (**hitsa**), private transfers (**hitp**) and tax and social security contribution expenditures (**hxit**).

- (1) Create an indicator variable that is equal to 1 if **dhi** or any of its income and expenditure sub-components is missing, and equal to zero otherwise.
- (2) Produce a frequency table for this variable to see how many cases have missing data.
- (3) Produce two different sets of descriptive statistics for the variables you have selected: unweighted, and weighted by person. For continuous variables, the statistics should include the number of observations, the mean, the median, the minimum and the maximum. For categorical variables, you should produce a frequency table.

(4) Drop all cases for which the indicator variable is equal to 1 and produce again the set of weighted descriptive statistics for all variables selected.

Question: What currency unit is used for the income variables in this dataset?

Question: What effect does applying the weights have on the median of disposable household income?

Question: What percentage of cases is being dropped from the dataset?

Question: In subsequent exercises, we will be comparing incomes before and after accounting for different kinds of government transfers. What is the difference between *work-related insurance transfers*, *universal benefits*, and *assistance benefits*?

Question: In subsequent exercises, we will separate *social assistance* transfers from social insurance/universal benefits. Based on the statistics you have produced, which type of transfer do you think has a larger effect on inequality and poverty in Guatemala?

Question: What specific government programs are included within the “social assistance” variable for Guatemala?

Question: How many different values does the variable **grossnet** take in this dataset? How might this variable be useful?

Guidelines

➤ To keep only the variables you will be using, use the **keep** options within the file command:

/keep = <varlist>

This avoids unnecessary burden on the machine so that submitted jobs will run faster. Combine the option and the command:

get file = gt06h / keep = <varlist>

➤ To run summary statistics use the **descriptives** command :

descriptives variables = <varlist>

➤ The descriptive command does not produce the median. A simple way to get to get the median is by using the **frequencies** command with the **/statistics=median** option. **Be aware** that when applying this to a continuous variable, you will get the full frequency table which will produce huge output. To avoid this, you need to add a further option **/format=notable**.

frequencies variables = <varlist>

- LIS Weights:
 - LIS records the household level weights in the variable **hpopwgt**. The person-level file contains a person-level weight variable, **ppopwgt**. When we are using the household file but want to weight by person rather than household, we can multiply the household weight by the number of household members, contained in the variable **nhhmem**.
 - Both of these are *inflated* weights. This means that for this dataset, the weight inflates to the total population in Guatemala in 2006. You can find the population size by looking at the weighted number of observations. There are also *normalized* weights, **hwgt** and **pwgt**, which can be useful when performing pooled analysis with multiple countries. These will be covered in a later exercise (see Ex. 8 in Part II).
- To get weighted descriptive statistics, you need to tell SPSS to weight by the variable that contains the weighting factor **weight by <varname>** . The variable name must be an existing one, or should have been created in advance. Your final command should look something like this:

weight by <varname> .

or in case of a composed weight :

compute populat = hpopwgt * nhhmem.

weight by populat.

Program

```
get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.  
compute miss_comp = 0.  
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitsa) or  
missing(hitp) or missing(hxit)) miss_comp = 1.  
frequencies variables = miss_comp.  
descriptives variables = dhi factor hitsi hitsa hitp hxit hpopwgt.  
frequencies variables = dhi /statistics=median /format=notable.  
frequencies variables = nhhmem grossnet.  
compute populat = hpopwgt * nhhmem.  
weight by populat.  
descriptives variables = dhi factor hitsi hitsa hitp hxit hpopwgt.  
frequencies variables = dhi /statistics=median /format=notable.  
frequencies variables = nhhmem grossnet.  
select if miss_comp eq 0.  
descriptives variables = dhi factor hitsi hitsa hitp hxit hpopwgt.  
frequencies variables = dhi /statistics=median /format=notable.  
frequencies variables = nhhmem grossnet.
```

Results

miss_comp	number of observations	percent	cumulative percent
0	13,664	99.84	99.84
1	22	0.16	100
Total	13,686	100	

Unweighted, all cases

	number of observations	mean	median	minimum	maximum
hpopwgt	13,686	193.8477	119	2	2,657
dhi	13,664	40670.24	27,092	-186,010	2,729,298
factor	13,664	39,924.08	24,777.50	0	3,866,644
hitsi	13,664	803.47	0	0	134,400
hitsa	13,664	695.49	0	0	62,400
hitp	13,664	1,438.65	0	0	128,020
hxit	13,664	2,191.45	0	0	1,137,346

hhmem	number of observations	percent	cumulative percent
1	646	4.72	4.72
2	1,345	9.83	14.55
3	2,004	14.64	29.19
4	2,376	17.36	46.55
5	2,277	16.64	63.19
6	1,738	12.7	75.89
7	1,218	8.9	84.79
8	856	6.25	91.04
9	538	3.93	94.97
10	284	2.08	97.05
11	183	1.34	98.39
12	93	0.68	99.06
13	50	0.37	99.43
14	44	0.32	99.75
15	20	0.15	99.9
16	6	0.04	99.94
17	3	0.02	99.96
18	1	0.01	99.97
19	2	0.01	99.99
21	2	0.01	100
Total	13,686	100	

grossnet	number of observations	percent	cumulative percent
[120]gross, taxes and contributions imputed	13,686	100	100
Total	13,686	100	

Weighted, all cases

	number of (weighted) observations	mean	median	minimum	maximum
hpopwgt	12,964,197	443.94	295	2	2,657
dhi	12,938,575	49,556.30	33,658	-186,010	2,729,298
factor	12,938,575	49,298.84	30,819	0	3,866,644
hitsi	12,938,575	777.41	0	0	134,400
hitsa	12,938,575	757.02	0	0	62,400
hitp	12,938,575	1,675.62	0	0	128,020
hxit	12,938,575	2,952.59	0	0	1,137,346

Weighted, missing income cases dropped

	number of (weighted) observations	mean	median	minimum	maximum
hpopwgt	12,938,575	443.7055	295	2	2,657
dhi	12,938,575	49,556.3040	33,658	-186,010	2,729,298
factor	12,938,575	49,298.8400	30,819	0	3,866,644
hitsi	12,938,575	777.4066	0	0	134,400
hitsa	12,938,575	757.0184	0	0	62,400
hitp	12,938,575	1,675.6224	0	0	128,020
hxit	12,938,575	2,952.5859	0	0	1,137,346

Question: What currency unit is used for the income variables in this dataset?

- The currency unit in this dataset is the Guatemalan Quetzal. This information can be found on the LIS METIS documentation system at *Our Data* → *METIS*, accessed through this [link](#)
- Each LIS dataset has a variable named "currency". This variable indicates the currency unit used for the income/expenditure variables. This variable is available in both the household and the person data files.

get file = gt06h
frequencies variables = currency.

currency units	number of observations	Percent	Cum.
[320]GTQ - Quetzal	13,664	100	100
Total	13,664	100	

Question: What effect does applying the weights have on the median of disposable household income?

- The median of unweighted **dhi** is 27,092 Quetzals, while the median of weighted **dhi** is 33,658 Quetzals, suggesting that low-income households are over-represented in the sample.

Question: What percentage of cases is being dropped from the dataset?

- There are 22 cases with missing data, or 0.16 percent of the total number of cases in the dataset. Note that you should always be careful if you see a large amount of missing data, as it could bias your estimates.

Question: In subsequent exercises, we will be comparing incomes before and after accounting for different kinds of government transfers. What is the difference between *work-related insurance transfers*, *universal benefits*, and *assistance benefits*?

- These concepts are defined in the [Interactive METadata Information System \(METIS\)](#).

LIS Database → Select variables → *hitsi/hitsu/hitsa* → click on the information icon ⓘ → a pop-up menu will appear displaying all the information related to these variables (definition and comment)

- Work-related insurance transfers: Monetary transfers stemming from systems where the eligibility is based on the existence and/or the length of an employment relationship; in most cases the benefits are financed by contributions paid by employers, workers or both, and their amount is usually dependent on either the previous earnings or the previous contributions;
- Universal benefits: Monetary transfers stemming from public programmes that provide flat-rate benefits to certain residents or citizens, provided that they are in a certain situation, but without consideration of income, employment or assets; note that in some cases the benefit amount may also depend on the other incomes of the individuals, which at the limit may result on some proportion of the population at the upper end of the income distribution to be excluded from receipt.

- Assistance benefits: Monetary and non-monetary transfers stemming from public programmes that provide benefits especially targeted to needy individuals or households (i.e. with a strict income or assets test); the amount of the benefits is either flat rate or based on the difference between the recipient income and a standard amount representing the minimum subsistence needs as guaranteed by the government.

Question: What specific government programs are included within the “social assistance” variable for Guatemala?

- The LIS interactive documentation tool on the LIS web site (*Our Data* → *METIS*) accessed through this [link](#) , among many functions offered ([see here](#)) , allows checking the institutional documentation for each data set
- From METIS interface → Enter METIS (yellow tab in the left-hand-side of the page) → select LIS → select Country “Guatemala” → select year “2006” → select “SOCIAL SECURITY” (blue tab in the top of the page) → select Main set (from the “Data to display” tab in the left-hand-side of the page) “ , you will see that the overall Social assistance variable for this dataset includes the following programmes:
- School transport allowance (*Bono de transporte escolar*), study grants (*Becas escolares*), scholarships from Child attention programme (*Programa de atención a la niña*), School material kit (*Bolsa de útiles escolares*), Food-related benefits from Social Assistance (*leche en polvo, vaso de leche, vaso de atol, alimentación escolar*), Health programme from Social Assistance and in-kind transfers of goods from public institutions.

Question: How many different values does the variable **grossnet** take in this dataset? How might this variable be useful?

- The variable **grossnet** has the same value for every case ([120] gross, taxes and contributions imputed.) By looking at this variable, you can learn what type of gross or net income information is contained in this variable, even without looking at the documentation. This variable will also be useful when working with multiple datasets that may contain either gross or net income.

3. Working with household income variables: top and bottom coding and equivalence scales

Goal

In order to compare incomes across countries, we need to make sure that our variables are fully comparable. In this exercise, you will apply top- and bottom-codes to remove extreme values. You will then create an *equivalised* income variable that adjusts for household size.

Top and bottom coding - Many inequality measures are sensitive to the values at the bottom and/or top of the income distribution, and some are not defined for non-positive values of income (e.g., any measure that calculates a logarithm). Applying top and bottom-codes (often referred to as 'winsorising') will avoid this problem, as well as ensuring comparability between datasets that may have originally had different top- and bottom-codings.

Equivalence scales - In order to get measures of poverty and/or income inequality in a population, it is necessary to compare income across different types of households. It is not logical to directly compare total household income between households of different sizes and composition.

Suppose you observe three levels of income (A, B, and C), where $A > B > C$. You cannot state that a household earning A is better off than one earning B unless you know the two households are similar in composition. For example, a family of four adult members receiving A is not necessarily better off than a couple with two children who receive B, and the family receiving B may not be better off than the childless couple receiving C.

For this reason, total household income needs to be adjusted to make it comparable across different households. This exercise gives one example of "equalising" households using one specific equivalence scale.

Activity

- (1) Keep the code from your previous exercise that you used to drop cases with missing data.
- (2) Create a new variable, **dhi_tb**, a top- and bottom-coded version of **dhi**. Bottom-code by setting all values less than zero to zero. Top-code by setting all values greater than ten times the median of **dhi** to ten times the median of **dhi**.
- (3) Create another new variable, **edhi_tb**, an equivalised version of top- and

bottom-coded disposable household income. We will correct for household size by applying the "LIS equivalence scale" (i.e., the square root of the number of household members).

- (4) Create a measure of per-capita income, **pcdhi_tb**, by dividing household income (un-equivalised, but top- and bottom-coded) by the number of household members.
- (5) Produce summary statistics showing the mean, median, minimum, and maximum of the four income variables: **dhi**, **dhi_tb**, **pcdhi_tb**, and **edhi_tb**.

Using the results returned from LISSY, fill out the following table:

	Household income (no top or bottom codes)	Household income	Per capita income	Equivalised income
Mean				
Median				
Minimum				
Maximum				

Question: Which of these four versions of the income variable contain *negative* values?

Question: Relative to household income and per capita income, how large are the mean and median of equivalised income?

Question: How does applying the top and bottom code affect the mean and median of household income?

Guidelines

- As we continue to build up to our final program, some of the code from the previous exercises will no longer be necessary. (For example, the code that produced the summary statistics in the previous exercise). You can choose to delete this code from your program in order to make it shorter. However, if you would like to keep a line code but stop it from being executed, simply place a ***** before it.
- This exercise does not use all of the variables that were used in the previous section. You should continue to keep all of those variables when you open the dataset, however, because they will be needed in future exercises.
- The top coding will be done at ten times the median. So far we merely produced the median in the output, but now we will need to have that amount

stored somehow into our data. This can be achieved with the **aggregate** command. Producing the median as a new variable which is added to the dataset, can be done as follows:

```
aggregate outfile = *  
mode=addvariables  
/break=<varname>  
/newvar=median(oldvar).
```

- You must declare a "break"-variable. This will determine if the median is calculated for instance per region, gender, or by any other subgroups. In this case we want the median for the total sample, so we need to choose a variable without variance. That is why we apply it on **did**, and for the same reason we needed to add did in the keep option.
- Once we have the median available, the top and bottom coding can be done by initializing a new variable **dhi_tb**, and apply the changes to that variable in the following way:

```
compute dhi_tb = dhi  
if dhi gt (10*mediandhi) dhi_tb = 10*mediandhi  
if dhi lt 0 dhi_tb = 0
```

- To equalise income, divide the total household income by the value of the equivalence scale for each observation. To create LIS equivalised income:

```
compute edhi_tb = dhi_tb / (nhhmem**0.5)
```

- Be careful when using weights. Make sure that the weight matches your unit of analysis. Weigh by **hpopwgt** for variables which are intrinsically at the household level (e.g., **dhi**) and by **hpopwgt*nhhmem** (to account for household size) for variables that are conceptually meaningful at the person level (e.g., per capita and equivalised income).

Program

get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.

compute miss_comp = 0.

if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitsa) or missing(hitp) or missing(hxit)) miss_comp = 1.

select if miss_comp eq 0.

weight by hpopwgt.

aggregate outfile= *

mode = addvariables

/break = did

/mediandhi= median(dhi).

***** top and bottomcoding.**

compute dhi_tb = dhi.

if dhi gt (10*mediandhi) dhi_tb = (10*mediandhi).

if dhi lt 0 dhi_tb = 0.

compute edhi_tb = dhi_tb/(nhhmem0.5).**

compute pcdhi_tb = dhi_tb/nhhmem.

descriptives variables = dhi dhi_tb.

frequencies variables = dhi dhi_tb /statistics=median /format=notable.

compute populat = hpopwgt * nhhmem.

weight by populat.

descriptives variables = pcdhi_tb edhi_tb.

frequencies variables = pcdhi_tb edhi_tb /statistics=median /format=notable.

Results:

	Household income (no top or bottom codes) (dhi)	Household income (top or bottom coded) (dhi_tb)	Per capita income (top or bottom coded) (pcdhi_tb)	Equivalised income (edhi_tb)
Mean	47,681.18	46,027.97	9,416.69	20,665.62
Median	31,096	31,096	5,766	14,007
Minimum	-186,010	0	0	0
Maximum	2,729,298	310,960	293,335	293,335

Question: Which of these four versions of the income variable contain *negative* values?

- Only the income variable without top or bottom codes contains negative values, which are removed by applying a bottom-code. This will be important in the next exercise on inequality. The measure of inequality we will be using, the Gini coefficient, does not allow negative values. Removing negative values also allows for the commonly-used logarithmic transformation of income.

Question: Relative to household income and per capita income, how large are the mean and median of equivalised income?

- The mean and median values for equivalised income fall between those for household income and those for per capita income. The equivalising formula of $\mathbf{dhi}/(\mathbf{nhmem}^{0.5})$ is a compromise between assigning all individuals their household income ($\mathbf{dhi}/\mathbf{nhmem}^0$) and assigning them a per capita income ($\mathbf{dhi}/\mathbf{nhmem}^1$).

Question: How does applying the top and bottom code affect the mean and median of equivalised income?

- Applying top and bottom codes makes the mean lower but does not affect the median (**31,096**). Means can be very sensitive to extreme values, so median values are often preferred as a measure of central tendency.

4. Inequality: The Gini Index

Goal

This exercise introduces the Gini index, which is one of the most commonly used income inequality indicators. We will be using the Gini coefficient as our measure of inequality in subsequent exercises, in order to compare inequality across countries and across different concepts of income.

Activity

Calculate the Gini index on total disposable income for Guatemala in 2006, using variables created in the previous exercise.

- (1) Start with your program from the previous exercise, which will drop observations with missing data,
- (2) Apply top and bottom codes,
- (3) Create variables containing equivalised disposable income and disposable income per capita.
- (4) Calculate the Gini coefficient for the winsorised (or bottom- and top-coded) versions of household income, per capita income, and equivalised income, and fill out the following table:

	Household income	Per capita income	Equivalised income
Gini coefficient			

Question: Which shows greater inequality: household income, per capita income or equivalised income? What does this suggest about the possible relationship between income and household size?

Guidelines

- SPSS does not provide a ready-made command that will calculate the Gini directly, and actually this requires some coding. accordingly, the codes for generating gini is provided in the **ginicalc** block, all you need to do is to specify what income variable (`inc_var`) the routine should run on, as well as the weighting factor (`wt_var`) that should be applied. One important note is to make sure that the dataset is weighted by the weight factor as well.

```
compute wt_var = <varname>
compute inc_var = <varname>
ginicalc.
```

- In order to define the ginicalc block and loop it over different income concepts, this leads us to introducing SPSS macros.
- Looping through datasets: introducing SPSS macros

In SPSS, the **define** -- **!enddefine** commands can be used to mark the start and ending of blocks of code. All what is in between these marks can than easily be repeated as many times as needed. If we use the program of the last exercise, and put these marks around it, we can create a macro that performs the same analysis multiple times. Such a macro needs a name. Whenever the macroname is called, the block of code is executed:

```
define <macroname> ( )  
...commands...  
!enddefine  
get file = first_dataset  
<macroname>  
get file = second_dataset  
<macroname>  
get file = third_dataset  
<macroname>  
etc. etc.
```

BE AWARE to write **!enddefine** in the correct spelling, NO space after exclamation mark.

Program

```
define ginicalc ()
sort cases by inc_var (a).
compute cumwgt = cumwgt + wt_var.
leave cumwgt.
aggregate outfile= *
  mode = addvariables
  /break= did
  /meany= mean(inc_var)
  /meanr= mean (cumwgt)
  /n=n.
compute devy= inc_var - meany.
compute rank= cumwgt/n.
compute devr = (rank - 0.5).
compute prod= devy*devr.
aggregate outfile= *
  /break=did
  /sumprod= sum(prod)
  /meany= mean(inc_var)
  /n=n.
compute cov= sumprod/(n-1).
compute gini=cov*2/meany.
formats gini (f10.4).
descriptives var=gini .
!enddefine .

define dataprepare ().
compute miss_comp = 0.
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitsa) or missing(hitp) or
missing(hxit)) miss_comp = 1.
select if miss_comp eq 0.
weight by hpopwgt.
aggregate outfile= *
  mode = addvariables
  /break = did
  /mediandhi= median(dhi).
*** top and bottomcoding.
```

```
compute dhi_tb = dhi.  
if dhi gt (10*mediandhi) dhi_tb = (10*mediandhi).  
if dhi lt 0 dhi_tb = 0.  
compute edhi_tb = dhi_tb/(nhhmem**0.5).  
compute pcdhi_tb = dhi_tb/nhhmem.  
!enddefine .
```

```
get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.  
*run the data prepare part  
dataprep.  
* set two input parameters (weighting and income variable).  
compute wt_var = hpopwgt.  
compute inc_var = dhi_tb.  
*weight the dataset with the weight variable complying with the inequality measure  
weight by hpopwgt .  
*run the gini calculation block  
ginicalc.
```

```
get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.  
* set two input parameters (weighting and income variable).  
dataprep.  
compute wt_var = hpopwgt*nhhmem.  
compute inc_var = pcdhi_tb.  
weight by wt_var.  
ginicalc.
```

```
get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.  
* set two input parameters (weighting and income variable).  
dataprep.  
compute wt_var = hpopwgt*nhhmem.  
compute inc_var = edhi_tb.  
weight by wt_var.  
ginicalc.
```

Results

	Household income	Per capita income	Equivalised income
Gini coefficient	0.479	0.509	0.470

Question: Which shows greater inequality: household income, per capita income or equivalised income? What does this suggest about the possible relationship between income and household size?

- The Gini is lower for equivalised income, than the Gini for household income, which in turn is lower than the Gini for per-capita income. This reflects the fact that poorer households in Guatemala tend to be larger than richer households. Because the LIS equivalence scale assumes some economies of scale in large households, it produces a lower estimate of inequality than a per-capita measure.

5. Relative poverty rates

Goal

In order to get any measure of poverty, it is essential to make some assumptions concerning the criteria based on which to define poverty. The approach used by LIS (and most commonly adopted in the literature), is that of creating a relative poverty line based on the level and distribution of household disposable (equivalised) income in the total population. Households are classified as poor or non-poor on the basis of whether their income is lower or higher than the relative poverty line.

Once poor households are identified, you can create an indicator to help identify the proportion of poor households (or individuals) and to measure the level of poverty. The choice of the indicator used will mainly depend on the purpose of the research. In this exercise, we will calculate the relative poverty rates of households and individuals in the Guatemala 2006 data.

Activity

Add code to your program to produce an indicator for poverty.

Define the poverty line as 50% of the median equivalised income.

Calculate both the percentage of households in poverty and the head count ratio (defined as the percentage of individuals living in poor households), and complete the following table.

	Households	Individuals
Relative poverty rate		

Question: Are there more poor *households* or more poor *individuals*? What can you infer from this?

Guidelines

- From this point forward, we will be working exclusively with equivalised income, so the sections of your code relating to per-capita income can now be commented out or removed. The code for producing the Gini coefficient of equivalised income is not needed for this exercise, but will be required again in the next exercise.
- You can create an indicator variable indicating that an individual is poor (**poor** = 0 or = 1). The mean of this variable will give you the proportion in poverty.
- Whether this is the proportion of *individuals* or *households* in poverty depends

on which weighting you use. Use **hpopwgt** if you want to measure household poverty, and **hpopwgt*nhhmem** if you are interested in individual poverty. If you use the individual-level weighting, you will produce the Head Count Ratio (HCR), which is the percentage of poor individuals in the total population.

- To get the median equivalised income, you can use again the command `aggregate`:

```
aggregate outfile = *  
mode=addvariables  
/break=<varname>  
/newvar=median(oldvar).
```

The principle is the same, fill **oldvar** now with **edhi_tb**.

- Create a variable `povline` that is equal to half the median of equivalised household income:

```
compute povline = medianedhi_tb * 0.5 .
```

- To determine the poverty rate, we create a dummy which is initialized at zero, and takes the value of 1 if a certain condition is satisfied (income below poverty line). The frequency table of that dummy will display the poverty rate.

```
compute poor = 0  
if edhi_tb lt povline poor = 1
```

Program

```
get file = gt06h /keep=did dhi factor hitsi hitp hxit hpopwgt nhhmem grossnet.
compute miss_comp = 0.
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitp) or missing(hxit)) miss_comp
= 1.
select if miss_comp eq 0.
weight by hpopwgt.
aggregate outfile= *
  mode = addvariables
  /break = did
  /mediandhi= median(dhi).
*** top and bottomcoding.
compute dhi_tb = dhi.
if dhi gt (10*mediandhi) dhi_tb = (10*mediandhi).
if dhi lt 0 dhi_tb = 0.
compute edhi_tb = dhi_tb/(nhhmem**0.5).

compute populat = hpopwgt * nhhmem.
weight by populat.

aggregate outfile = *
  mode = addvariables
  /break = did
  /medianedhi_tb = median(edhi_tb) .

compute povline = medianedhi_tb * 0.5 .
descriptives variables = povline .

compute poor = 0 .
if edhi_tb lt povline poor = 1 .
weight by hpopwgt.
frequencies variables = poor .
weight by populat.
frequencies variables = poor .
```

Results

	Households	Individuals
Relative poverty rate	22.1%	22.4%

Question: Are there more poor *households* or more poor *individuals*? What can you infer from this?

- There is a slightly greater proportion of poor individuals than poor households. This is because poor households are larger on average than non-poor households. This is another reason why the use of the equivalence scale is important.

Comments

- The head count ratio (HCR) measures poverty incidence (i.e., the number or proportion of poor people), but gives every person equal weight no matter how far they fall from the poverty line.
- Another measure, the Income Gap Ratio (IGR) measures poverty intensity or depth (how poor are the poor), but one poor person with an income of an amount x counts the same as two poor people each with an income of $x/2$. That is, the IGR measures the average income gap, but not its distribution among the poor).
- There are many other indicators of poverty that may be useful for different purposes. These include, among the most common, the whole family of Foster-Greer-Thorbecke indicators (of which the HCR is only one), the Sen index, the Takayama index, the Clark index, and the Thon index. It is important to note that a country may score better in comparison to a second country when using a particular index, but could score worse if another index was used instead.

6. Comparing income concepts

Goal

Now that we have calculated Gini coefficients and poverty rates based on disposable income, we can easily apply this same code to two other income concepts: income before any taxes and government transfers, and income after taxes, social insurance, and universal benefit transfers (but before social assistance transfers).

Different income concepts - The income variable we have been working with, **dhi**, combines multiple income and expenditure flows. It is the sum of labour and capital income, private transfers, work-related insurance transfers, universal benefits, and social assistance benefits, minus income taxes and social insurance contributions paid. We will now define two new concepts of income. One of them is income before *any* government redistribution. The second is income *after* taxes, social insurance, and universal benefits, but *before* social assistance is included.

By calculating the Gini coefficient and the poverty rate using each of these three income concepts (income before government intervention, income after non-assistance government redistribution, and income after all government redistribution, i.e. our original disposable household income variable, **dhi**), we gain some insight into the effect of government programmes on inequality and poverty.

Activity

As always, begin with the program you developed for the last exercise. Modify it to create two new income variables. The first, **mi**, is the sum of factor income (**factor**) and private transfers (**hitp**). Because we are specifically interested in the role of *government* transfers, we add private transfers to our measure of “market income” from labor and capital.

The second, **siti**, adds **mi** together with social insurance transfers (**hitsi**) and universal benefits (**hitsu**), while subtracting taxes and social contributions paid (**hxit**).

The income variable we have been using up to now, disposable household income, adds together the variables contained in **siti** along with social assistance transfers (which are also contained in the variable **hitsa**).

Make sure you apply top codes, bottom codes, and the equivalence scale to the new variables, producing the final variables **emi_tb**, **esiti_tb**, and **edhi_tb**. You

should apply the same topcode value of ten times median **dhi** to both of the other two variables.

Calculate the Gini coefficient and the poverty rate for all three income variables, based on the code from the previous two exercises. Use it to fill out the table below. Make sure you use the *same* poverty line for all three income variables. That is, the poverty line should be defined as 50 percent of the median equivalised disposable household income, and that same poverty definition should be applied to the other two income variables.

	Before taxes and government transfers	After taxes, social insurance, and universal benefits	After taxes and all transfers
Gini coefficient			
Poverty rate			

Question: What has the greater impact on inequality and poverty in Guatemala: taxes/social insurance/universal benefits, or social assistance?

Program

```
define dataprep ( ).
compute miss_comp = 0.
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitsa) or missing(hitp) or
missing(hxit)) miss_comp = 1.
select if miss_comp eq 0.
weight by hpopwgt.
aggregate outfile= *
  mode = addvariables
  /break = did
  /mediandhi= median(dhi).
compute dhi_tb = dhi.
compute mi_tb = factor + hitp.
compute siti_tb = factor + hitp + hitsi - hxit.
*** top and bottomcoding.
if dhi gt (10*mediandhi) dhi_tb = (10*mediandhi).
if mi_tb gt (10*mediandhi) mi_tb = (10*mediandhi).
if siti_tb gt (10*mediandhi) siti_tb = (10*mediandhi).
if dhi lt 0 dhi_tb = 0.
if mi_tb lt 0 mi_tb = 0.
if siti_tb lt 0 siti_tb = 0.
compute edhi_tb = dhi_tb/(nhhmem**0.5).
compute emi_tb = mi_tb/(nhhmem**0.5).
compute esiti_tb = siti_tb/(nhhmem**0.5).
!enddefine .

define ginicalc ( )
sort cases by inc_var (a).
compute cumwgt = cumwgt + wt_var.
leave cumwgt.
aggregate outfile= *
  mode = addvariables
  /break= did
  /meany= mean(inc_var)
  /meanr= mean (cumwgt)
```

```

/n=n.
compute devy= inc_var - meany.
compute rank= cumwgt/n.
compute devr = (rank - 0.5).
compute prod= devy*devr.
aggregate outfile= *
  /break=did
  /sumprod= sum(prod)
  /meany= mean(inc_var)
  /n=n.
compute cov= sumprod/(n-1).
compute gini=cov*2/meany.
formats gini (f10.4).
descriptives var=gini .
!enddefine .

```

```

get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.
dataprepere.
* set two input parameters (weighting and income variable).
compute wt_var = hpopwgt * nhhmem.
compute inc_var = emi_tb.
weight by wt_var.
ginicalc.

```

```

get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.
dataprepere.
* set two input parameters (weighting and income variable).
compute wt_var = hpopwgt * nhhmem.
compute inc_var = esiti_tb.
weight by wt_var.
ginicalc.

```

```

get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.
dataprepere.
* set two input parameters (weighting and income variable).

```

```

compute wt_var = hpopwgt * nhhmem.
compute inc_var = edhi_tb.
weight by wt_var.
ginicalc.
get file = gt06h /keep=did dhi factor hitsi hitsa hitp hxit hpopwgt nhhmem grossnet.
dataprepate.
compute populat = hpopwgt * nhhmem.
weight by populat.
aggregate outfile = *
  mode = addvariables
  /break = did
  /medianedhi_tb = median(edhi_tb) .
compute povline = medianedhi_tb * 0.5 .
compute poor = 0 .
if emi_tb lt povline poor = 1 .
frequencies variables = poor .
compute poor = 0 .
if esiti_tb lt povline poor = 1 .
frequencies variables = poor .
compute poor = 0 .
if edhi_tb lt povline poor = 1 .
frequencies variables = poor .

```

Results

	Before taxes and government transfers "mi"	After taxes, social insurance, and universal benefits "siti"	After taxes and all transfers "dhi"
Gini coefficient	0.489	0.478 *	0.470
Poverty rate	24.5%	24.0% *	22.5%

* Calculation excludes universal benefits since it is not filled in gt06

Question: What has the greater impact on inequality and poverty in Guatemala: taxes/social insurance/universal benefits, or social assistance?

- For both inequality and poverty, assistance benefits play a larger role than social insurance and universal benefits.

Comments

- When comparing incomes before and after taxes and transfers, take care in interpreting the meaning of the pre-tax and transfer figure. It is tempting to interpret this number as a representation of the income distribution that would exist in the absence of government programs. However, since outcomes in the private sector are conditioned by the presence or absence of government programs, this is not generally a reasonable inference.

7. Comparing multiple countries

Goal

Now that we have written code to compute all of our statistics of interest, it is time to calculate these quantities for multiple countries. This exercise also shows the use of macros to loop through multiple datasets.

Adding more countries – Before adding a new country/year to an analysis, it is important to check that the dataset in question has all information necessary for the analysis you are performing. In this case, one should carefully check that the income sub-components variables used in the previous exercise are filled for all the new datasets to be introduced (and if not, whether the analysis can be slightly modified to take into account a different situation).

Use of macros to loop – In this exercise we will introduce the use of macros to call the same lines of codes several times from the main program. This is a technique that allows you to better organise your code, and it is also helpful when you want to repeat a series of commands several times.

Activity

Take the code from the previous exercise and modify it so that it loops through six datasets: Guatemala 2006 (**gt06**), United States 2004 (**us04**), Denmark 2004 (**dk04**), Poland 2004 (**pl04**), Hungary 2005 (**hu05**) and Israel 2005 (**il05**).

One of these countries is a net income dataset. This means that the tax contribution variable **hxit** is missing for all cases. Because our current program drops cases that are missing any income subcomponent variable, all the cases will be dropped for net income datasets. Using the **grossnet** variable, you can modify your code to prevent this from happening.

Use your results to fill in the following tables:

Gini Coefficient

Dataset	Before taxes and government transfers	After taxes, social insurance, and universal benefits	After taxes and all transfers
GT06			
US04			
DK04			
PL04			
HU05			
IL05			

Poverty Rate

Dataset	Before taxes and government transfers	After taxes, social insurance, and universal benefits	After taxes and all transfers
GT06			
US04			
DK04			
PL04			
HU05			
IL05			

Keep in mind that even if all cells can technically be constructed, the result may not necessarily be comparable conceptually! Think carefully about whether the dataset you are looking at contains the necessary information to calculate the quantity in each column.

Question: In what cells does the figure you produced not match the income concept described in the column header?

Question: Comparing Guatemala and Poland, which country has higher inequality before taxes and transfers? After?

Question: Which country has the highest poverty rate before taxes and transfers? After?

Question: In which country do government programmes do the most to reduce inequality and poverty, in percentage terms? In which country do they do the least?

Question: In which countries do social assistance benefits do more to reduce poverty than social insurance plus universal benefits and taxes?

Guidelines

➤ Looping through datasets: defining a macro with arguments

If you want to perform a loop through many countries, you could go even one step further, and extend the macro with an argument (in our example number of repetitions). You have to specify the number of repetitions in the start, and list the names of the datasets when you call the macroname. BE AWARE : make sure that the number # matches the number of dataset that you list when calling the macroname. We only recommend this further step to those who feel comfortable with macro language:

```

define <macroname> (!positional !tokens(#)
!do !i !in (!1)
get file = !i /keep= etc....
...commands...
!doend
!enddefine
<macroname> first_dataset second_dataset etc. etc.

```

- The variable **grossnet** reports whether the incomes in a dataset are *gross* income, before taxes, or whether the only report post-tax values. Within a single dataset, all cases will have the same value for this variable. In a purely net income dataset, **grossnet** will be between 200 and 299. For more information about whether the LIS datasets report gross or net values, This information can be found on the LIS METIS documentation system at *Our Data* → *METIS* → *LIS database* → *select "ccyy"* → *Dataset information*. *METIS* is accessed through this [link](#)

Use **grossnet** to create a new indicator variable **netincome**. Adjust the value of **netincome** to perform different calculations for net income datasets. For example, this code will prevent our program from dropping all observations in net income datasets:

```

compute miss_comp = 0.
frequencies variables = grossnet.
compute netincome = 0
if grossnet ge 200 and grossnet le 299 netincome = 1.
do if netincome eq 0.
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitsu) or
missing(hitp) or missing(hxit)) miss_comp = 1.
else.
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitsu) or
missing(hitp)) miss_comp = 1.
end if.

```

The variable «hitsu» is not filled in gt06. therefore, we want to omit hitsu from the creation of miss_comp. One way to prevent the creation from taking hitsu into account, is to create a new variable as the mean of "hitsu" (using the aggregate command and then break the variable by the **did**), if this **mean variable** equals **missing**, then hitsu should be omitted from the calculation of miss_comp. using this code will prevent the program from dropping all the cases in gt06.

Program

```
define gt06data ()  
get file = gt06h / keep = did dhi factor hitsi hitsu hitp hxit hpopwgt nhhmem  
grossnet.  
!enddefine  
define us04data ()  
get file = us04h / keep = did dhi factor hitsi hitsu hitp hxit hpopwgt nhhmem  
grossnet.  
!enddefine  
define dk04data ()  
get file = dk04h / keep = did dhi factor hitsi hitsu hitp hxit hpopwgt nhhmem  
grossnet.  
!enddefine  
define pl04data ()  
get file = pl04h / keep = did dhi factor hitsi hitsu hitp hxit hpopwgt nhhmem  
grossnet.  
!enddefine  
define hu05data ()  
get file = hu05h / keep = did dhi factor hitsi hitsu hitp hxit hpopwgt nhhmem  
grossnet.  
!enddefine  
define il05data ()  
get file = il05h / keep = did dhi factor hitsi hitsu hitp hxit hpopwgt nhhmem  
grossnet.  
!enddefine
```

```
define ginicalc ()  
sort cases by inc_var (a).  
compute cumwgt = cumwgt + wt_var.  
leave cumwgt.  
aggregate outfile= *  
  mode = addvariables  
  /break= did  
  /meany= mean(inc_var)  
  /meanr= mean (cumwgt)  
  /n=n.  
compute devy= inc_var - meany.  
compute rank= cumwgt/n.  
compute devr = (rank - 0.5).  
compute prod= devy*devr.  
aggregate outfile= *  
  /break=did  
  /sumprod= sum(prod)  
  /meany= mean(inc_var)  
  /n=n.  
compute cov= sumprod/(n-1).  
compute gini=cov*2/meany.  
formats gini (f10.4).  
descriptives var=gini .  
!enddefine .
```

```
define run ()  
compute netincome = 0.  
if grossnet eq 200 netincome = 1.  
compute miss_comp = 0.  
aggregate outfile= *
```

```

mode = addvariables
/break = did
/meanhitsu= mean(hitsu).
if missing(meanhitsu) miss_hitsu =1.
do if netincome eq 0.
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitsu) or
missing(hitp) or missing(hxit) ) & (miss_hitsu =0) miss_comp = 1.
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitp) or missing(hxit)
) & (miss_hitsu =1) miss_comp = 1.
else.
if (missing(dhi) or missing(factor) or missing(hitsi) or missing(hitsu) or
missing(hitp)) miss_comp = 1.
end if.
select if miss_comp eq 0.
weight by hpopwgt.
aggregate outfile= *
mode = addvariables
/break = did
/mediandhi= median(dhi).
compute dhi_tb = dhi.
compute mi_tb = factor + hitp.

compute siti_tb= factor + hitp + hitsi + hitsu - hxit.
if ( miss_hitsu eq 1 & netincome eq 0) siti_tb = factor + hitp + hitsi - hxit.
if (netincome eq 1) siti_tb = factor + hitp + hitsi + hitsu.

*** top and bottomcoding.
if dhi gt (10*mediandhi) dhi_tb = (10*mediandhi).
if mi_tb gt (10*mediandhi) mi_tb = (10*mediandhi).
if siti_tb gt (10*mediandhi) siti_tb = (10*mediandhi).
if dhi lt 0 dhi_tb = 0.
if mi_tb lt 0 mi_tb = 0.
if siti_tb lt 0 siti_tb = 0.

compute edhi_tb = dhi_tb/(nhhmem**0.5).
compute emi_tb = mi_tb/(nhhmem**0.5).
compute esiti_tb = siti_tb/(nhhmem**0.5).

compute populat = hpopwgt * nhhmem.
weight by populat.
aggregate outfile = *
mode = addvariables
/break = did
/medianedhi_tb = median(edhi_tb) .
compute povline = medianedhi_tb * 0.5 .

compute poor_emi = 0 .
if emi_tb lt povline poor_emi = 1 .
compute poor_esiti = 0 .
if esiti_tb lt povline poor_esiti = 1 .
compute poor_edhi = 0 .
if edhi_tb lt povline poor_edhi = 1 .
!enddefine

define poverty ()
run.
frequencies variables = poor_emi.

```

```

frequencies variables = poor_esiti.
frequencies variables = poor_edhi.
!enddefine

define emi ()
run.
compute wt_var = hpopwgt * nhhmem.
compute inc_var = emi_tb.
weight by wt_var.
ginicalc
!enddefine

define esiti ()
run.
compute wt_var = hpopwgt * nhhmem.
compute inc_var = esiti_tb.
weight by wt_var.
ginicalc
!enddefine

define edhi ()
run.
compute wt_var = hpopwgt * nhhmem.
compute inc_var = edhi_tb.
weight by wt_var.
ginicalc
!enddefine

```

gt06data.
emi.
gt06data.
esiti.
gt06data.
edhi.
gt06data.
poverty.

us04data.
emi.
us04data.
esiti.
us04data.
edhi.
us04data.
poverty.

dk04data.
emi.
dk04data.
esiti.
dk04data.
edhi.
dk04data.
poverty.

**pl04data.
emi.
pl04data.
esiti.
pl04data.
edhi.
pl04data.
poverty.**

**hu05data.
emi.
hu05data.
esiti.
hu05data.
edhi.
hu05data.
poverty.**

**il05data.
emi.
il05data.
esiti.
il05data.
edhi.
il05data.
poverty.**

Results

Gini Coefficient

Dataset	Before taxes and government transfers	After taxes, social insurance, and universal benefits	After taxes and all transfers
GT06	0.489	<i>0.478**</i>	0.470
US04	0.487	0.389	0.364
DK04	0.447	0.264	0.228
PL04	0.531	0.336	0.315
HU05	<i>0.535*</i>	0.316	0.289
IL05	0.518	0.394	0.369

Poverty Rate

Dataset	Before taxes and government transfers (mi)	After taxes, social insurance, and universal benefits (siti)	After taxes and all transfers (dhi)
GT06	24.5	<i>24.0**</i>	22.5
US04	28	21.3	17.3
DK04	29.2	13.4	5.6
PL04	42.0	14.7	10.8
HU05	<i>44.1*</i>	12.2	7.4
IL05	32.8	22.5	19.4

*Calculation based on post-tax income

**Calculation excludes universal benefits

Question: In what cells does the figure you produced not match the income concept described in the column header?

- Because Hungary 2005 is a net income dataset, the Gini before taxes and transfers cannot be included. While the example program below does produce a result for Hungary, it is not actually comparable to the other two countries because it is post-tax. That cell should therefore be left blank.

Question: Comparing Guatemala and Poland, which country has higher inequality before taxes and transfers? After?

- Poland has higher inequality before taxes and transfers are accounted for. After taxes and transfers, however, inequality is substantially higher in the Guatemala.

Question: Which country has the highest poverty rate *before* taxes and transfers? After?

- Poland has the highest poverty rate before taxes and transfers, but Guatemala has the highest poverty rate after taxes and transfers.

Question: In which country do government programmes do the most to reduce inequality and poverty, in percentage terms? In which country do they do the least?

- Government programmes have the largest impact on reducing inequality and poverty in Denmark, where they reduce the Gini coefficient by 49 percent, and poverty by 80 percent.
- Government programs have the lowest impact on reducing inequality and poverty in Guatemala, where they reduce the Gini coefficient by 4 percent and poverty by 8 percent.

Question: In which countries do social assistance benefits do more to reduce poverty than social insurance plus universal benefits and taxes?

- In Guatemala, social assistance reduces poverty more than social insurance plus universal benefits and taxes.

Comments

- The datasets in these exercises were chosen because they allow social assistance to be separated from other kinds of government transfers. In many datasets, unfortunately, this separation is not possible due to the limitations of the original data. In such cases, the total amount of transfer income will be contained in a higher-level variable such as **hits**, and lower level variables such as **hitsi** and **hitsu** will be unfilled. You can consult the **Cross-compare** function in the METIS documentation tool, to determine which variables are available in each dataset. This is the **main functionality** of METIS. Upon entering the tool ([METIS](#)), it is possible to view the availability of the selected variable(s) in the selected dataset(s) and to **compare information about this selection**
- In datasets containing gross incomes, the **grossnet** variable can take values of 100, 110, or 120. In datasets containing only net income, **grossnet** will always take a value of 200. Note, however, that a few datasets have **grossnet** codes of 300, 310, or 320, because they contain a mixture of gross and net incomes. See the variable definition and its content using the METIS function **variable information** that allows you to check the content and the possible values of selected variable(s). while the **Cross-compare** function allows you to check the variable value taken for the selected dataset(s).