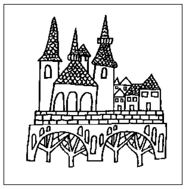
Lab Session Exercises

Summer Workshop 2010

SPSS version

Luxembourg Income Study

June 27th – July 3rd 2010



CONTENTS

INTRODUCTION

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

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

Goal

The Job Submission Interface (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.

The first exercise is an introduction to the JSI to ensure the ability to launch and use it successfully.

Activity

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

Submit a simple program to display the following text in your output (or listing): “Your program has run successfully”.

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.

Guidelines

Once connected to the Job submission Interface, there are three main tasks that may be carried out:

- Submit jobs through the Job Session window.
  - Select a project (LIS, LWS or LES).
  - Select a statistical package (SAS, SPSS or Stata).
  - When submitting a job (Job Session window), always add a subject line.
  - Write your code.
  - Click on the submit button.
  - Note that for security reasons, the output of all job requests will be returned to the email registered in LISSY even if the submission is processed through the JSI. That way, each LIS user will be informed if someone is using his/her userid and password.
- 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.
- 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.

Program

```
title "Your program has run successfully." .
```

Comments

- All listings from LISSY begin with a "Notice to users". The remaining part of the listing is the actual output from SPSS.
- Note that for security reasons, the output of all job requests will be returned to the email registered in LISSY even if the submission is processed through the JSI. That way, each LIS user will be informed if someone is using his/her *userid* and *password*.
- To preserve the confidentiality of information pertaining to individuals and/or households in the micro-databases, LISSY prohibits the use of SPSS commands such as: **`print`** and **`list`**. If those commands are used, LISSY will display a dialog box along with an error message explaining the violation:

  ```
  Your job has been set for manual review
  ```

- In addition to checking for illegal commands, LISSY also filters submissions based on the usage of sequences of commands and/or variables that would end up breaching the rules on data confidentiality. For instance, using commands that display frequencies on continuous variables (e.g., income variables) will be detected by LISSY. LISSY automatically puts such jobs in a security review area to be manually reviewed by the staff. You will be alerted about the result of the manual review.
- If the size of a listing is larger than a given limit for a statistical package, the job is also automatically put in the security review area to ensure that there is no attempt to identify individual-level micro-data. Again, you will be alerted about the result of such review.

BASICS I

Ex. 2: Running Descriptive Statistics: Sample and Population Values

2. Running Descriptive Statistics: Sample and Population Values

Goal

This exercise is an introduction to a few of the variables in the household- and person-level LIS data sets. The exercise concentrates on job syntax, basic descriptive statistics and the use of the weight.

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

For Luxembourg 2004 (LU04), create a household-level dataset containing: the household identifier (***casenum***), household weight (***hweight***), number of earners in the household (***d6***), number of children under 18 (***d27***), whether the head of the household is living in a couple (***married***), age of the household head (***d1***), gender of household head (***d3***), gender of the spouse of the household head (***sexsp***), and the household net disposable income (***dpi)***.

Find the unweighted and weighted number of observations, mean, median, minimum, and maximum for the continuous variables (including ***casenum*** and ***hweight***) and the unweighted and weighted frequencies of the categorical variables.

Use the information from your output to answer the following questions:

1. Why can ***sexsp*** have a value of -1, but ***d3*** can never be -1?

   ____

   ____

   ____

2. Why do the values of ***pgwage*** and ***pgwtime*** differ (check the *Variables Definition List* and the *Lissification Table* for LU04 on line)?

   ____

   ____

   ____

   ____

Guidelines

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

  ```
  get file = lu04h.
  ```

  For more information about the syntax of country/year macros, see the job submission instructions on the LIS web site (*Micro-Databases Access → Job Submission Instructions*). For a list of available data sets and their 2-digit country codes, go to

  Luxembourg Income Study (LIS) → List of Datasets.

The command get file will load the entire dataset in memory. Keeping only a subset of variables avoids unnecessary burden on the machine so that submitted jobs will run faster. Only keep the variables you will be using. The full command will be :

```
get file = lu04h
     /keep = casenum d6 d27 married d1 d3 dpi.
```

- Each SPSS command should be ended by a delimiter, in case of SPSS this is the dot .
- When a command continues over several lines, INDENT each following line with a couple of spaces, and only put the dot at the end of the last line.
- To produce results like the mean, number of cases, minimum , maximum, you can use the command descriptives.
- SPSS reminder:

  - to run descriptives, use `descriptives variables = <variable list>.`

  Unfortunately, unlike the mean, the median will not be produced by the **descriptives** command. Therefore one needs to use the **frequencies** command, while specifying the option **statistics = median**. By default, the frequency command will not only produce the required median, but also the full frequency table. Please note that running frequencies on continuous variables (like wage, weights, identification numbers, etc.) will produce a huge frequency table!! This must be avoided, and can be done by adding the **format = notable** option in order to suppress the table. Your final frequencies command will look like this:

```
frequencies variables = dpi
    / statistics = median
    / format = notable.
```

- If you need help determining which variables are categorical, go to the LIS web site and click on *Luxembourg Income Study (LIS)* → *Luxembourg* → *2004* (column: *Lissification Tables*; row: *Wave VI*). The "Value Labels" column of the *Lissification Table* delineates the values of categorical variables.
- To get weighted results, you need to use the command **weight by** followed by the name of the weighting variable. Each result that follows after this command will give weighted results. The weights stay in effect until another weighting command is encountered. To return to unweighted results, use the command:

```
weight off.
```

- For this dataset, the weight inflates to the total population in Luxembourg in 2004. This means you can find the population size by looking at the **sum** in the weighted summary. Information about sample size and weighted population estimates can be found at *Luxembourg Income Study (LIS)* → *<country>* → *Weighting Procedures.*
- **IMPORTANT: Wait to get your results before sending a new job!**

Program

```
title "** BASICS I – Exercise 2 **" .


get file = lu04h
   / keep = casenum hweight d6 d27 married d1 d3 sexsp dpi .
title " unweighted " .
descriptives variables =  casenum hweight d1 dpi .
* add each time lines for the median only .
frequencies variables =  casenum hweight d1 dpi
        / statistics = median
        / format = notable .
frequencies variables =  d6 d27 married d3 sexsp .
weight by hweight.
title " weighted " .
descriptives variables =  casenum hweight d1 dpi .
frequencies variables =  casenum hweight d1 dpi
        / statistics = median
        / format = notable .
frequencies variables =  d6 d27 married d3 sexsp .

get file = lu04p
   / keep = casenum ppnum pweight page psex pmart prel pgwage pgwtime .
title " unweighted " .
descriptives variables =  casenum ppnum pweight page pgwage pgwtime .
frequencies variables =  casenum ppnum pweight page pgwage pgwtime
        / statistics = median
        / format = notable .
frequencies variables =  psex pmart prel .
weight by pweight.
title " weighted " .
descriptives variables =  casenum ppnum pweight page pgwage pgwtime .
frequencies variables =  casenum ppnum pweight page pgwage pgwtime
        / statistics = median
        / format = notable .
frequencies variables =  psex pmart prel .
```

Results

Continuous household-level variables – unweighted results

	# of obs	*Mean*	*Median*	*Minimum*	*Maximum*
casenum	*3,622*	*1,811.5*	*1,811.5*	*1*	*3,622*
hweight	*3,622*	*49.12*	*28*	*0.105*	*466.04*
d1	*3,622*	*48.95*	*48*	*18*	*100*
dpi	*3,622*	*56,750*	*48,598*	*-34,602*	*686,352*

Continuous household-level variables – weighted results

	# of obs	*Mean*	*Median*	*Minimum*	*Maximum*
casenum	*177,910*	*1,282.1*	*1,317*	*1*	*3,622*
hweight	*177,910*	*113.15*	*96.77*	*0.105*	*466.04*
d1	*177,910*	*51.04*	*49*	*18*	*100*
dpi	*177,910*	*55,371*	*47,373*	*-34,602*	*686,352*

Categorical household-level variables

Variable name	*Codes*	*Labels*	*# of obs in the sample*	*unweighted percent*	*weighted percent*
d6	*0*		*881*	*24.32*	*27.70*
	1		*1,444*	*39.87*	*37.92*
	2		*1,135*	*31.34*	*29.35*
	3		*130*	*3.59*	*3.92*
	4		*27*	*0.75*	*1.04*
	5		*4*	*0.11*	*0.02*
	6		*1*	*0.03*	*0.05*
d27	*0*		*2,291*	*63.25*	*68.96*
	1		*614*	*16.95*	*13.39*
	2		*493*	*13.61*	*12.52*
	3		*177*	*4.89*	*4.30*
	4		*31*	*0.86*	*0.42*
	5		*13*	*0.36*	*0.35*
	6		*2*	*0.06*	*0.01*
	7		*1*	*0.03*	*0.05*
married	*0*	*head not living in couple*	*1,166*	*32.19*	*35.68*
	1	*married couple*	*2,064*	*56.99*	*57.35*
	3	*non-married cohabiting couple*	*382*	*10.55*	*6.48*
	5	*non-married cohabiting couple, both partners same sex*	*10*	*0.28*	*0.49*
d3	*1*	*male*	*2,390*	*65.99*	*64.12*
	2	*female*	*1,232*	*34.01*	*35.88*
sexsp	*-1*		*1,166*	*32.19*	*35.68*
	1	*male*	*550*	*15.18*	*14.75*
	2	*female*	*1,906*	*52.62*	*49.57*

Continuous individual-level variables – unweighted results

	# of obs	*Mean*	*Median*	*Minimum*	*Maximum*
casenum	*9,661*	*1,808.2*	*1,796*	*1*	*3,622*
ppnum	*9,661*	*3.11*	*2*	*1*	*63*
pweight	*9,661*	*46.27*	*24.08*	*0.105*	*466.04*
page	*9,661*	*35.47*	*35*	*0*	*100*
pgwage	*9,661*	*16,763*	*0*	*0*	*430,000*
pgwtime	*9,661*	*1,234.6*	*0*	*0*	*25,000*

Continuous individual-level variables – weighted results

	# of obs	*Mean*	*Median*	*Minimum*	*Maximum*
casenum	*447,006*	*1,250.4*	*1,276*	*1*	*3,622*
ppnum	*447,006*	*2.89*	*2*	*1*	*63*
pweight	*447,006*	*113.01*	*95.80*	*0.105*	*466.04*
page	*447,006*	*37.65*	*38*	*0*	*100*
pgwage	*447,006*	*17,723*	*0*	*0*	*430,000*
pgwtime	*447,006*	*1,314.2*	*0*	*0*	*25,000*

Categorical individual -level variables

Variable name	*Codes*	*Labels*	*# of obs in the sample*	*unweighted percent*	*weighted percent*
psex	*1*	*male*	*4,808*	*49.77*	*49.57*
	2	*female*	*4,853*	*50.23*	*50.43*
pmart	*1*	*never married*	*4,349*	*45.02*	*41.81*
	2	*married*	*4,266*	*44.16*	*46.68*
	3	*separated*	*91*	*0.94*	*0.84*
	4	*widowed*	*425*	*4.40*	*5.58*
	5	*divorced*	*530*	*5.49*	*5.09*
prel	*1*	*head of household*	*3,622*	*37.49*	*39.80*
	2	*husband/wife*	*2,059*	*21.31*	*22.81*
	3	*partner*	*397*	*4.11*	*2.79*
	4	*own/adopted child*	*3,149*	*32.59*	*31.21*
	5	*step child (child of husband/wife)*	*62*	*0.64*	*0.50*
	6	*step child (child of partner)*	*47*	*0.49*	*0.20*
	7	*child in law*	*21*	*0.22*	*0.18*
	8	*foster child*	*15*	*0.16*	*0.24*
	9	*brother or sister*	*46*	*0.48*	*0.33*
	10	*sister/brother in law by marriage*	*9*	*0.09*	*0.08*
	11	*sister/brother in law by partnership*	*1*	*0.01*	*0*
	12	*mother or father*	*78*	*0.81*	*0.85*
	13	*parent-in-law by marriage*	*37*	*0.38*	*0.22*
	14	*parent-in-law by partnership*	*1*	*0.01*	*0.01*
	15	*grandchild*	*66*	*0.68*	*0.48*
	16	*great grandchild*	*1*	*0.01*	*0*
	17	*grandparent*	*2*	*0.02*	*0.01*
	21	*niece or nephew*	*15*	*0.16*	*0.08*
	23	*aunt or uncle*	*6*	*0.06*	*0.07*
	24	*aunt or uncle of spouse*	*1*	*0.01*	*0*
	25	*cousin*	*1*	*0.01*	*0.01*
	27	*other relative of head*	*1*	*0.01*	*0.01*
	29	*other not related person*	*24*	*0.25*	*0.15*

Answers to question 1-2:

1. ***sexsp*** *is -1 when the individual does not belong to the universe of those individuals (the subsample) who are asked for that information. In this case, only households with a couple present are asked about spouse's gender, so* ***sexsp*** *is always -1 for heads not living in a couple, and never -1 for those households with couples.* Since every household must have a head, ***d3*** must always have a value of 1 or 2.
2. *The* Variable Definition List *explains that, with the exception of* ***pgwtime*** *and* ***pnwtime****, all income variables are recorded in annual amounts (see cell H257). It also states that* ***pgwtime*** *contains gross wages for the unit of time (less than a year) that can be most accurately measured in the original data (cell H260). The* Lissification table *tells you that in Luxembourg, this measure is monthly gross income. By looking at the* Contents *(column G) of* ***pgwage*** *and* ***pgwtime****, you can see what constitutes gross income. In the case of LU04, the contents of* ***pgwage*** *include all gross income from dependent work, including wages, 13th and 14th month salaries, special or exceptional bonuses, wages from a secondary professional activity, and income from apprenticeships. The variable* ***pgwtime*** *includes the same information, but is adjusted by the data provider to account for the number of months worked (e.g., if 2 individuals have the same value for* ***pgwage****, the individual with the fewest months worked will have a higher value for* ***pgwtime****.)*

Comments

- File composition

  There are 9,661 observations of the identifier, ***casenum***, which gives us the total sample size (number of persons in this case).

  Without opening the household-level files, we can get the total number of households in the sample by looking at the number of household heads (***prel***=1). In Luxembourg 2004, there are 3,622 household heads. In many cases, you can also find the number of households by looking at the maximum value of ***casenum*** (which here is also 3,622). If these two values differ, then some of the original households have been removed from the main file and have been either included in the shadow file or dropped completely. (Go to *Luxembourg Income Study (LIS)* → *LIS Policy on the Treatment of Missing Information* and *Luxembourg Income Study (LIS)* → *LIS Policy on the Treatment of Shadow Files* for a discussion about the LIS sample composition and shadow files.)

- Remember that the income variables are the nominal value of the national currency.

- Married variable

  As of Wave V, ***pmart*** is always coded as 2 if married. If more detailed marital information is given by the data provider, never married will be coded as 1 and other marital information (e.g., divorced, separated, widowed) are given codes above 2.

  Please be aware that when information about cohabiting status is not available for each person in the original dataset, a head with a cohabiting partner could be coded in ***pmart*** as single (if never civically married). See ***pparsta*** and ***prel*** for more information about cohabiting status.

- For this dataset, unweighted results on age (as well as other variables) are lower than the weighted ones. This means that younger individuals are over-represented in the sample. The sample (person) weight corrected for this by giving those individuals a lower weight. The unweighted result gives the average for the survey sample, not the Luxembourg average in 2004.

- Size of Population

  For this dataset, the weight also inflates to total population. This means you can find the population size by looking at the "Sum of Wgt." in the weighted summary.

  - In some datasets, the average weight is equal to 1. In this case, the "Sum of Wgt." is equal to the number of observations in the sample.
  - In other datasets, the weight "inflates to the population", i.e. the weight for each unit in the sample is equal to the number of units he/she represents in the population (a "unit" could be a household or an individual). In other words, the average weight multiplied by the sample size gives the total population in the country.

- The person-level LIS weight is ***pweight***. In some cases ***pweight*** is given directly by the data provider and is the inverse of the probability of the individual being included in the sample. In cases where ***pweight*** is not provided (e.g., most household surveys), ***pweight*** for each member in the household is equivalent to the household weight, ***hweight***.

BASICS II

Ex. 3: Using Subroutines and Loops to Simplify Submissions Involving Several Datasets

Ex. 4: Combining Datasets

3. Using Subroutines and Loops to Simplify Submissions Involving Several Datasets

Goal

When doing comparative research, it is natural to want to run the same routine across several different countries or across different waves. This approach can be simplified by the use of macros, subprograms/subroutines, and loops. These are easy ways to repeat program commands without having to retype them each time.

Activity

Compare the median (weighted) labour earnings for the United States and Mexico (2000), and the United Kingdom (1999) by looking separately at gross wage earnings (***pgwage***), net wage earnings (***pnwage***) and self-employment earnings (***pself***). Your sample should only include those with positive labour earnings (from either paid or self-employment) who are 16 years of age or older. For this comparison, you should write a subroutine to run for each dataset.

Now repeat the same analysis for the five new Latin American datasets (BR06, CO04, GT06, PE04 and UY04). In order to shorten the code, loop the subroutine created above over the 5 datasets.

Use the information from your output to answer the following questions:

1. Which country had the highest median wages from employment?

   ____

   ____

   ____

2. Are the income variables analysed above expressed in net or gross terms (i.e. before or after deduction of income taxes and mandatory social contributions)?

   ____

   ____

   ____

Guidelines

- For this comparison, proceed according to the following steps:
  - use a subroutine to prepare the data and calculate the estimates;
  - select the age-group;
  - call the subroutine for each of the individual country data sets.
- Writing a subroutine:

```
define dofiles ().         (dofiles is just a name, but you may choose another)
<add all the commands needed, each with a dot at the end >.
!enddefine .
```

- Please be careful when closing a subroutine: if you fail to close it properly your job will block the batch machine on our system.

REMINDER: **a) do not forget the exclamation mark !**
b) no space between the exclamation mark and the command enddefine

c) no misspelling of the command like endefine , eddeffine, etc.

If you accidently misspell enddefine, the command is not recognized.

- The closing brackets at the end of the define command are needed, and can optionally contain further parameters. For our exercises no use of parameters is being made, therefore no space between the brackets. In case you are interested in using the full functionality of macro's we refer to the SPSS manuals.
- Be careful with selecting records: selecting on age-group is harmless, but pay attention when using variables that might be empty throughout the entire dataset; like gross wage or net wage in this example. When you use the select if command, SPSS will delete all records that do not fulfil the condition. Selecting on a variable that is empty will create an empty file, and will therefore result in an SPSS ERROR ! To circumvent this, do not select, but in stead create new labour earnings variables that copy the value from pnwage, pgwage or pself only if positive.
- Calling a subroutine by its name :

```
dofiles.
```

- In the end, your code should look something like:

```
define dofiles () .
<apply the weights>
<select on age-group>
<create the new labour earnings variables>
<find the median of pgwage for adults with positive earnings>
<find the median of pnwage for adults with positive earnings>
 <find the median of pself for adults with positive self-employment
    earnings>
!enddefine .


get file = uk99p
     /keep = <list of variables to use>.
dofiles .

get file = us00p
     /keep = <list of variables to use>.
dofiles .
```

- Do not include too many countries in one and the same job, since the output will grow accordingly. Long listings do end up in our review queue, and have to be examined manually before being released. Please try to avoid long listing since this is a burden for the LIS team.

Program

```
title "** BASICS II – Exercise 3 **" .

title "** Part 1 **" .
define dofiles () .
weight by pweight .
select if page ge 16.
if pgwage gt 0 posgwage = pgwage.
if pnwage gt 0 posnwage = pnwage.
if pself gt 0 posself = pself.
frequencies variables = posgwage posnwage posself
      / statistics = median
      / format = notable .
!enddefine .
get file = uk99p
     / keep = pweight page pgwage pnwage pself  .
dofiles .
get file = us00p
     / keep = pweight page pgwage pnwage pself  .
dofiles .
get file = mx00p
     / keep = pweight page pgwage pnwage pself  .
dofiles .
title "** Part 2 **" .
get file = br06p
     / keep = pweight page pgwage pnwage pself  .
dofiles .

get file = co04p
     / keep = pweight page pgwage pnwage pself  .
dofiles .

get file = gt06p
     / keep = pweight page pgwage pnwage pself  .
dofiles .

get file = pe04p
     / keep = pweight page pgwage pnwage pself  .
dofiles .

get file = uy04p
     / keep = pweight page pgwage pnwage pself  .
dofiles .
```

Results

	Gross wage earnings	Net wage earnings	Self-employment earnings
UK99	*13,513*	*10,628*	*10,400*
US00	*25,000*	*NA*	*12,500*
MX00	*NA*	*27,400*	*14,400*
BR06	*5,865*	*NA*	*6,000*
CO04	*4,320,000*	*NA*	*2,400,000*
GT06	*12,000*	*NA*	*6,000*
PE04	*NA*	*5,986*	*2,980*
UY04	*NA*	*54,042*	*42,577*

Answers to questions 1-2:

1. Country with the highest median wages from employment

   Because incomes are expressed in national currencies, it is not possible to compare values directly.

2. *Whereas it is obvious from the results above that MX00 contains only net incomes and US00 only gross incomes (as only one of the two wage variables is filled), for the UK99 dataset it is necessary to look at the documentation on-line to see that all other income variables are reported gross of taxes and contributions (with, in addition, the net wage variable).*

Comments

- Please note that all the results are given in nominal terms and in national currency. In order to make a direct comparison, convert these values to a common currency with the use of exchange rates (or PPPs) and deflators. LIS leaves it up to the researcher to choose the exchange rates and/or deflators that are best suited to his/her purpose.
- LIS detailed income variables are ideally filled with gross values (before taxes and mandatory social contributions are deducted), so that their overall sum (reported in summary income variable ***gi***) is equal to total gross income, from which taxes and contributions are subtracted to get to the final net disposable income figure (***dpi***). In some instances though, the original datasets only report net incomes: whereas total gross income is then not available (and the summary income variable ***gi*** is thus left empty), total final net disposable income figure is obtainable by aggregating all the net incomes (and this is exactly what ***dpi*** includes). In those cases, each LIS detailed income variable will contain net values instead of gross; only for wages there are two separate variables for those two amounts. For all other variables you need to take care when comparing across datasets that you are not comparing two different concepts of income.

4. Combining Datasets

Goal

There are a number of reasons you may wish to combine LIS files. You may wish to combine files from different countries/years in order to run a regression utilizing country/year indicators. You may wish to use household information in a person-level analysis (or vice versa). You may want to include individuals or households from the shadow files if you are studying a specific subset of the population.

In this exercise, you will be asked to merge household and person level variables from one dataset in the same file.

Activity

Use the data for Belgium 2000. First access the household level dataset and calculate the mean of the number of household members. Then create a new dataset at the individual level containing ***casenum***, ***hweight***, ***d4***, ***d7***, ***d22***, ***ppnum***, ***pweight***, ***page***, ***psex***, ***pnwage*** (by merging the household and person level datasets) and carry out the following tasks:

- create a household-level counter containing the number of persons in each household and compare it with the household-level variable ***d4***; calculate its mean over both individuals and over households;
- compare the mean individual wage of wage earners by region;
- compare the average age of individuals living in households that own their residence outright (i.e., without mortgage) to that of individuals living in households who still have a mortgage on their residence.

Guidelines

➢ In order to merge two datasets, they must necessarily contain at least one "merging variable", i.e. a variable which links the observations of one dataset to those of the other dataset. The variable must have the same name in the two datasets and will link observations with the same value in the two datasets (in case of several merging variables, it will link the observations with the same combination of values). To link household and person LIS datasets, such a variable is the household identifier (***casenum***), which exists in both datasets and takes the same values for the household at the household level and all the individuals belonging to that household in the person-level file.

➢ When merging datasets using a common merging variable, make sure that each of the files is sorted on that merging variable. The person and household files are already sorted on ***casenum***, but when you make certain selections or other data manipulations, you could make it a habit to always sort before merging, just to be on the safe side. In that case, you will need to save some temporary files. When saving temporary files, you can place them in a directory at LIS. In order that your filename differs from others saving files at the same time, save your files using your userid-name. For example, imaginary user Toby would save his temporary file by typing:

```
save outfile= “toby_filename.sav”
```

➢ When merging person-level information to household-level-information, you are combining data of two different units. Inform SPSS of different units being used in the following way : the file with the smaller units (persons) is referred to as file , whereas the file with the larger units (households) is called table. Reminder: you may need to sort on the “by”-variable before merging:

```
match files file = <p-file> /table = <h-file> /by casenum .
```

- In order to calculate household level statistics from a person level file, you can select on household heads only (***ppnum***=1), so that you are sure you include exactly one observation per household.

- To produce results for subsets you can (besides selecting) in stead use the command <split file>. This is a simple way to get results for instance per region. Make sure to sort the file before splitting on the variable by which you want to split up.

```
sort cases by <varlist>.
split file by <varlist> .
```

Program

```
title "** BASICS II – Exercise 4 **" .
get file = be00h
    / keep = casenum hweight d4 .
descriptives variables =  d4 .
weight by hweight.
title "weighted " .
descriptives variables =  d4 .
* merging P and H.
match files    file = be00p
          / table = be00h
          / keep = casenum hweight d4 d7 d22 ppnum pweight page psex pnwage
          / by casenum .
title "unweighted (number of obs) " .
descriptives variables =  d4 .
temporary.
select if ppnum eq 1.
descriptives variables =  d4 .
sort cases by d7.
split file by d7.
if pnwage gt 0 posnwage = pnwage.
descriptives variables = posnwage .
sort cases by d22.
split file by d22.
descriptives variables = page .
split file off.
title "weighted " .
weight by pweight.
descriptives variables =  d4 .
temporary.
select if ppnum eq 1.
descriptives variables =  d4 .
sort cases by d7.
split file by d7.
if pnwage gt 0 posnwage = pnwage.
descriptives variables = posnwage .
sort cases by d22.
split file by d22.
descriptives variables = page .
```

Results

		Number of observations	Mean
d4		*2,697*	*2.44*
Counter of household members	For all observations	*6,935*	*3.19*
	For household heads only	*2,697*	*2.44*
pnwage	Flanders	*1,552*	*670,909*
	Brussels	*234*	*732,045*
	Wallonia	*826*	*648,581*
page	Owners with mortgage	*3,246*	*27.4*
	Owners without mortgage	*2,256*	*54.0*

Comments

- You will have noticed that in this exercise the number of records after merging equals the number of persons. This should always be the case with LIS household and individual level files from the same dataset because all individuals belong to at least one and no more than one household.
- When calculating descriptive statistics, you should always be careful to choose the unit (and hence the weight) that make most sense for the calculation: household level statistics (such as household-level counters) should be calculated over households (using the household-level weight), while person-level statistics should be calculated over persons (using the person-level weight).

DEMOGRAPHICS AND EDUCATION

Ex. 5: Children (Household Level)

Ex. 6: Gender (Person Level)

Ex. 7: Comparing ***educ*** and ***peduc***

Ex. 8: Comparing Educational Outcomes

5. Children (Household Level)

Goal

The standard of living of individuals in single-mother households has been the focus of much research. Nevertheless, there is no clear-cut definition of a single-mother household. We can limit the sample to households composed of a single female adult and her children, or we can allow other adult members to be present (as long as they are not defined as her partner). We may also wish to limit single-mother households to be those with children under a specified age limit.

In this exercise, we will look at the characteristics of households with and without children, limiting the analysis to household heads and partners.

Activity

Use LIS data from Sweden, Germany, and the US in 2000. Compare the percentage of households with and without children. Within these groups, compare the percentage of coupled households, single-women/mother households, and single-men/father households.

Guidelines

- Use ***parstahd*** to identify heads with young children (< 18 years). Refer to the LIS variable definitions to find the standardized values for ***parstahd***. For this exercise, count households with children 18 and over as childless households.
- Use the `crosstab` command to get a cross-tabulation of parenthood status versus the sex of the head. Use the option `/cells = total` to get percentages of all households. Your final command may look something like this:

```
crosstab parstahd by d3
  /cells = total .
```

Program

```
title "** DEMOGRAPHICS AND EDUCATION – Exercise 5 **" .

define dofiles () .
weight by hweight .
crosstab parstahd by d3
  /cells = total .
!enddefine .

get file = se00h
     / keep = hweight d3 parstahd .
dofiles .

get file = de00h
     / keep = hweight d3 parstahd  .
dofiles .

get file = us00h
     / keep = hweight d3 parstahd  .
dofiles .
```

Results

	SE00	**DE00**	**US00**
Percentage of households			
Couples	*45.87 %* *(23.46 + 19.17 + 3.24)*	*55.91%*	*56.19%*
Single women	*30.48 %* *(24.78 + 4.82+ +0.88)*	*27.63%*	*27.63%*
Single men	*23.66 %* *(22.35 + 0.95 + 0.36)*	*16.55%*	*16.19%*
TOTAL	*100%*	*100%*	*100%*
Percentage of households with children <18			
Coupled parents	*19.17%*	*19.95%*	*25.23%*
Single mothers	*4.82%*	*3.1%*	*6.13%*
Single fathers	*0.95%*	*0.3%*	*1.14%*

Comments

- Note that the proportion of coupled households is much higher in Germany and the US with respect to Sweden. This is most likely because an individual is only coded as a partner in Sweden if they are married or have registered for partnership status, whereas couples in Germany and the US include cohabiting partners. Nevertheless, most households with children are in coupled households, which may mean that cohabiting partners in Sweden marry or register their partnership status when a child is imminent. Be sure to check the documentation for each country to make sure you are clear about the information provided in the LIS variables.
- Beware that ***parstahd*** assumes that the children of the head are the children of the spouse (and vice versa). While this assumption may be valid for some analyses, you may need dig deeper if, for example, you are trying to connect motherhood to labour force status. (See ***pclfs*** and ***pcare*** for employment and leave status).

6. Gender (Person Level)

Goal

In doing any estimation, it is important to be careful about the unit of analysis. In research focusing on women, you must take into account whether the data you are using are individual- or household-specific.

Using individual-level data allows you to identify individual-specific income, but problems may arise in estimation depending on your research question. Some of these issues are general, but others are specific to the LIS database. First, some income sources are common to the household (such as child benefits or housing allowances) and are not available at the individual level. In LIS, certain individual income sources (invalidity and work accident pensions, sickness and maternity allowances, means-tested benefits, social transfers) are reported in detail only in the household file. The information is present in the person-level file in an aggregated form.

In this exercise, we introduce income analysis by gender. Using the person-level file, we will focus only on earned income amounts, not considering social transfers.

Activity

Examine the working-aged population (25 to 60, inclusive) in the UK in 1999 and the US in 2000. Compare the percentage of working men to that of working women, defined as those with positive earnings from any employment. Calculate the average total income by gender of both the total working-aged population and the working population. Estimate the gender earnings gap for both the working-aged population and for those who work.

Guidelines

- For this exercise, define the "working-aged" population as those aged 25 to 60, inclusive, and the "working" population as those with positive earnings from paid and/or self-employment (***pgwage*** + ***pself***).
- The gender income gap is defined as the ratio of average total earnings (***pgwage***+***pself***) of males to females.
- To simplify the analysis in this exercise, set negative values of ***pself*** to missing before calculating the "working" population. Failure to do so may result in self-employed with negative incomes being counted as not working, or negative incomes being considered in the average for the population.
- You may also find the *SPSS* commands `temporary` in combination with `select if` useful to get your results. To obtain descriptives for subsets, you can either run the program twice, or use the `temporary` to apply a filter which serves the same purpose as the selection. Please remind that the temporary only applies to the command that immediately follows.
- For most of the next exercises, when percentages are asked, simply use the calculator.

Program

```
title "** DEMOGRAPHICS AND EDUCATION – Exercise 6 **" .

define dofiles () .
weight by pweight .
if pself lt 0 pself = $sysmis .
compute ttearn = pgwage + pself .
temporary .
select if psex eq 1 and (page ge 25 and page le 60) .
frequencies variables = ttearn
     / statistics = mean
     / format = notable .
temporary .
select if psex eq 2 and (page ge 25 and page le 60) .
frequencies variables = ttearn
     / statistics = mean
     / format = notable .
temporary .
select if psex eq 1 and (page ge 25 and page le 60) and ttearn gt 0 .
frequencies variables = ttearn
     / statistics = mean
     / format = notable .
temporary .
select if psex eq 2 and (page ge 25 and page le 60) and ttearn gt 0 .
frequencies variables = ttearn
     / statistics = mean
     / format = notable .
!enddefine .

get file = uk99p
    / keep = pweight page psex pgwage pself  .
dofiles .

get file = us00p
    / keep = pweight page psex pgwage pself  .
dofiles .
```

Results

	UK99		*US00*	
	Males	**Females**	**Males**	**Females**
Percentage of individuals with positive earnings in working-aged population	*11,359,337 / 13,906,081 = **81.7%***	*9,429,261 / 13,906,112 = **67.8%***	*60,066,557 / 66,098,374 = **90.9%***	*53,819,001 / 68,821,548 = **78.2%***
Average total earnings (working-aged population)	***£ 19,101***	***£ 8,543***	***$ 44,133***	***$ 22,062***
Average total earnings (working population)	***£ 23,384***	***£ 12,599***	***$ 48,565***	***$ 28,212***
Gender earnings gap (working-aged population)	*19,101 / 8,543 = **2.24***		*44,133 / 22,062 = **2.00***	
Gender earnings gap (working population)	*23,384 / 12,599 = **1.86***		*48,565 / 28,212 = **1.72***	

Comments

- Please note that this exercise examines only individual earnings. Allocation of earnings (and other income) among household members is not considered. Income gender analysis becomes much more demanding and requires many more assumptions about the allocation of total household income when other household members are present.
- It is interesting to note that the earnings gap is lower when the employment rate is higher. While a two-country statistical snapshot does not provide enough information to draw conclusions, these types of summary statistics often provide researchers with new questions to investigate.

7. Comparing *educ* and *peduc*

Goal

Education information can vary substantially between datasets. This results mainly from the different educational systems in the various countries, but can also be the result of different ways of surveying the topic. (Some surveys allow for very detailed answers, while others ask only for categories of education.). LIS harmonises, but does not standardise this information; that is, the original survey information is coded into the same variable for every country, but as much country-specific detail as possible remains.

Nevertheless, since measures of comparable education are used in many areas of research, LIS has created a standardisation routine for the education variables that transforms each country-specific educational label into a new standardised variable based on the International Standard Classification of Education (ISCED). This exercise compares the country-specific education information to the ISCED recoding using that routine.

Activity

Using data from the United States and Luxembourg in 2000, run the education recode program to create the standardized education variable (***educ***). Tabulate the country-specific education variable (***peduc***) with the LIS standardized education variable (***educ***).

Guidelines

- When running the education recode file, remember to include the variables ***country***, ***ptocc***, and ***peduc*** when calling your country file.
- To run the education standardization program, include the following line in your program:

```
include file edurecop.
```

- For more information about how education levels are recoded in each country, see http://www.lisproject.org/techdoc/education-level/education-level.htm.
- To compare ***peduc*** and ***educ***, use the `crosstab` command.

Program

```
title "** DEMOGRAPHICS AND EDUCATION – Exercise 7 **" .

define dofiles () .
include file edurecop .
crosstab peduc by educ.
!enddefine .

get file = us00p
    / keep = country peduc ptocc penrol  .
dofiles .

get file = lu00p
    / keep = country peduc ptocc penrol  .
dofiles .
```

Results

US00

code	Label	Low	medium	high	missing
-1	not applicable				*X*
1	less than 1st grade	*X*			
2	1st , 2nd, 3rd or 4th grade	*X*			
3	5th or 6th grade	*X*			
4	7th or 8th grade	*X*			
5	9th grade	*X*			
6	10th grade	*X*			
7	11th grade	*X*			
8	12th grade, no diploma	*X*			
9	high school graduate (high school diploma or equivalent)		*X*		
10	some college, no degree		*X*		
11	associate degree, vocational program			*X*	
12	associate degree, academic program			*X*	
13	bachelor's degree			*X*	
14	master's degree			*X*	
15	professional school degree (md, dds, dvm, llb, jd)			*X*	
16	doctorate degree			*X*	

LU00

code	label	low	medium	high	missing
-1	not applicable				*X*
0	no education	*X*			
1	primary school	*X*			
2	first stage lower technical secondary education	*X*			
3	complementary education	*X*			
4	second stage lower technical secondary education	*X*			
5	medium technical training		*X*		
6	higher technical education			*X*	
7	professional certificate		*X*		
8	lower secondary general education	*X*			
9	higher secondary general education		*X*		
10	handicraft certificate		*X*		
11	first stage university education			*X*	
12	university education (3 years)			*X*	
13	university education (4 years)			*X*	
14	post-university education			*X*	
.	Missing				*X*

8. Comparing Educational Outcomes

Goal

When comparing educational levels across countries, it is necessary to carefully look at the labels of those variables for each country, and recode them to make them comparable across the countries you are investigating. Or you may wish to use the education routine created by LIS (as seen in the previous exercise). In this exercise, the educational population structure of different countries is compared with the use of the LIS educational routine.

Activity

Compare the educational composition of the total adult populations (16+) of the US, Luxembourg, and Italy in 2000 by gender. Repeat the exercise for the wage-earning population. Calculate average wages of wage earners for each country by education level.

Guidelines

- Please note that for some datasets, including Italy and Luxembourg, income is reported net of taxes and social contributions. (For more information, see http://www.lisproject.org/techdoc/netdatasets.htm). For those datasets, use net wages instead of gross wages. Net wage is reported in a different variable (***pnwage*** rather than ***pgwage***).
- One way to ensure that you choose the correct wage is by creating a new variable with the wage we want to use. In this case, we want ***pgwage*** if it is available and ***pnwage*** otherwise The order of the two will simply determine which one has priority over the other :

  ```
  compute pwage = 0 .
  if pnwage gt 0 pwage = pnwage .
  if pgwage gt 0 pwage = pgwage .
  ```

  Keep in mind that in some countries, both ***pgwage*** and ***pnwage*** exist. For such countries, this code will always select ***pgwage*** over ***pnwage***. If you prefer to prioritize ***pnwage*** over ***pgwage***, simply interchange the two lines accordingly.
- Use the education standardisation subroutine:

  ```
  include file edurecop .
  ```
- Remember to include the variables necessary for the standardisation routine in your `keep` statement in addition to ***pweight*** and the applicable wage variable.
- Use the `crosstab` command to find all the combinations of education level and gender. Since you want results by gender, you will need to get the `/cells = columns` option to produce all the required percentages.
- To get the mean of the wage by level of education for wage earners, you can use:

  ```
  split file by psex educ.
  ```

Program

```
title "** DEMOGRAPHICS AND EDUCATION – Exercise 8 **" .

define dofiles () .
select if page ge 16.
include file edurecop .
weight by pweight .
compute pwage = 0 .
if pnwage gt 0 pwage = pnwage .
if pgwage gt 0 pwage = pgwage .
crosstabs table = educ by psex
   / cells = column .
select if pwage gt 0 .
crosstabs table = educ by psex
   / cells = column .
sort cases by psex educ .
split file by psex educ .
descriptives variables = pwage
     / statistics = mean .
!enddefine .

get file = us00p
    / keep = country pweight page psex peduc ptocc pgwage pnwage .
dofiles .
get file = lu00p
    / keep = country pweight page psex peduc ptocc pgwage pnwage .
dofiles .
get file = it00p
    / keep = country pweight page psex peduc ptocc pgwage pnwage .
dofiles .
```

Results

	Educ Level	**US00**		**LU00**		**IT00**	
		Males	Females	Males	Females	Males	Females
Percent of total population	low	*20.3*	*19.2*	*30.4*	*45.2*	*57.9*	*63.3*
	medium	*49.0*	*51.0*	*36.4*	*28.2*	*33.6*	*29.6*
	high	*30.7*	*29.8*	*26.1*	*19.0*	*8.5*	*7.1*
Percent of wage earning population	low	*14.8*	*11.9*	*26.0*	*32.8*	*47.7*	*33.0*
	medium	*50.6*	*52.1*	*39.1*	*34.9*	*41.3*	*50.4*
	high	*34.6*	*36.0*	*34.4*	*31.8*	*11.0*	*16.7*
Average wage of wage earners	low	*17,744*	*10,364*	*883,626*	*514,064*	*23,334*	*17,142*
	medium	*33,026*	*20,215*	*1,204,372*	*681,842*	*28,717*	*22,112*
	high	*64,344*	*35,711*	*1,747,091*	*1,107,552*	*40,286*	*27,220*
Returns to education	low →med	*86%*	*95%*	*36%*	*33%*	*23%*	*29%*
	med →high	*95%*	*77%*	*45%*	*62%*	*40%*	*23%*
Gender wage gap	low	*1.7*		*1.7*		*1.4*	
	medium	*1.6*		*1.8*		*1.3*	
	high	*1.8*		*1.6*		*1.5*	

Comments

- Please note that education was not recoded for some countries in certain years. Check documentation on-line (lissification tables and descriptives) for more precise information about education levels.
- Please note that in the results above, the percentage of population by level of education may not add up to 100% because the category ***educ*** =9 (missing or not defined) was not included in the table, but was included in the calculations.
- The education composition across countries varies considerably. (Italy has the least-educated population of the three countries chosen for this comparison.)
- In all countries, wage earners are more educated than the total population.
- As expected, wages increase with the level of education, but to a different extent in each country. In the US, returns to education are substantially higher than in Luxembourg or Italy.

- Net versus Gross wages

  Please note that, even when considering exchange rates (or PPPs), it is not possible to directly compare the level of wages between countries that report either net or gross wages. In these cases, it is only possible to compare ratios. Even then, a progressive taxation system might affect the ratios. If high wage earners (i.e., the most-educated) face higher tax rates than low earners, the returns to higher education will be lower than if the returns had been measured using gross wages. The higher gross returns, therefore, are partly offset by higher taxes.

INCOME DISTRIBUTION I

Ex. 9: Equivalence Scales

Ex. 10: Poverty Lines and Poverty Rates

Ex. 11: Elderly and Child Poverty Rates

9. Equivalence Scales

Goal

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 4 adult members receiving A is not necessarily better off than a couple with 2 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 "equalizing" households using one specific equivalence scale.

Activity

Summarise total disposable income, per capita disposable income, and equivalised disposable income using the "LIS equivalence scale" (i.e., the square root of the number of household members) in Finland in 2000. First calculate the averages for the total population. Then recalculate the same averages by the number of household members. Print your results only for households with 7 or fewer household members. Be sure to eliminate observations with zero or missing ***dpi*** and to use the appropriate weights.

Guidelines

- Do not forget to "clean" the data. As always, it is important to be vigilant about missing values. Prior to Wave V, no distinction was made between 0 and missing values. Starting from Wave V, the lissification process consistently coded missing values with a "dot" and genuine 0 values by 0. Nevertheless, to be able to cover all the waves consistently we advise you to drop both missing and 0 values of ***dpi***.

  Warning! When you start working with smaller sub-samples, dropping observations may significantly affect your results if dropped observations all belong to one group that is central to your analysis (e.g., older immigrants, or low-educated blue-collar workers). Be careful about what you are doing. Understand your data.

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

  ```
  compute ey = dpi/(d4**0.5)
  ```

- Be careful when using weights. Make sure that the weight matches your unit of analysis. Weigh by ***hweight*** for variables which are intrinsically at the household level (e.g., ***dpi***) and by ***hweight*d4*** (to account for household size) for variables that are conceptually meaningful at the person level (e.g., per capita and equivalised income).

Program

```
title "** INCOME DISTRIBUTION I – Exercise 9 **" .

get file = fi00h /keep = hweight d4 dpi .

select if dpi ne 0 .
compute ypc = dpi/d4 .
compute ey = dpi/(d4**0.5) .
compute wt = hweight*d4 .

weight by hweight .
descriptives variables = dpi .
weight by wt .
descriptives variables = ypc ey .
sort cases by d4 .
temporary.
select if d4 le 7.
split file by d4 .
weight by hweight .
descriptives variables = dpi / statistics = mean .
weight by wt .
temporary.
select if d4 le 7.
split file by d4 .
descriptives variables = ypc ey / statistics = mean .
* optional : additional line to see number of
   observations per household size.
weight off .
frequencies variables = d4 .
```

Results

	Total income	**Per capita income**	**Equivalised income**
Average income for all households	*144 891*	*67 338*	*105 377*
Average income for households:			
- with 1 member	*77 475*	*77 475*	*77 475*
- with 2 members	*160 425*	*80 212*	*113 438*
- with 3 members	*198 849*	*66 283*	*114 806*
- with 7 members	*234 376*	*33 482*	*88 586*
What is the relationship between income and household size?	*Positive*	*Negative*	*No clear pattern*

Comments

- Total household income obviously increases with household size, whereas per capita household income generally decreases. Neither of these two measures is appropriate to compare the well-being of households of different sizes.
- We use an equivalence scale because we believe that there are economies of scale in a household. Therefore, the marginal income needed decreases as the household size grows. As a result, equivalised income becomes independent from the household size, and we can compare different households.
- Note that when calculating statistics, it is always important to check for the cell size: the average income measures by number of household members, may be based on very few observations when the household size increases (in this specific case, the number of households with more than 8 members drops to less than 30 observations, so that no sound conclusion can be taken for that group of households).

10. Poverty Lines and 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 indicator used will mainly depend on the purpose of the research. In this exercise, we will calculate the main indicator of poverty incidence, the head count ratio, and the income gap ratio (an important indicator of poverty intensity).

Activity

Using the 2000 Finnish data, run the same data cleaning procedures and create the equivalence scale introduced in the previous exercise. Define the poverty line as 50% of the median equivalised income. Calculate the head count ratio (defined as the percentage of individuals living in poor households) and the income gap ratio (as explained in the guidelines).

Guidelines

- To know if a household is below (a certain percentage of) the median equivalised disposable income, we need to calculate this median, and match it back to each household record. This is complex to achieve in SPSS, and therefore LIS has prepared the steps for you in this excercise (see block of code in program).

  For the poverty line, create a new variable ***povline*** equal to 50% of the median equivalised income:

  ```
  compute povline = 0.5 * <median_equivalised_income> .
  ```

- Again, be careful when choosing weights: use ***hweight*** if you want to measure household poverty, and ***hweight**d4*** if you are interested in individual poverty.
- The Head Count Ratio (HCR) is the percentage of poor individuals in the total population. When you create a dummy variable indicating that an individual is poor (**poor** = 0 or =1), then the mean of the indicator variable (properly weighted) will be the percentage of poor individuals.
- The Income Gap is the difference between income and the poverty line. The Income Gap Ratio (IGR) is the average income gap as a percentage of the poverty line. In this exercise it is probably the simplest to determine the ICR by using a calculator while using the results from IG and poverty line.

Program

```
title "** INCOME DISTRIBUTION I – Exercise 10 **" .


get file = fi00h /keep = hweight d4 d5 dpi .
select if dpi ne 0 .
select if d5 ne 3 .
compute ey = dpi/(d4**0.5) .
compute wt = hweight*d4 .
weight by wt .
frequencies variables = ey
   / statistics = median
   / format = notable.

*** code to use median within same session.
compute group = 1 .
rank ey by group /ties = high  /rfraction into rankey .
compute overmid = 0 .
if rankey gt 0.5 overmid = 1.
temporary .
select if  overmid eq 1 .
aggregate outfile = outmedey
  / presorted
  / break = group
  / medianey = min(ey) .
match files
  / table = outmedey
  / file = *
      /drop = rankey
  / by group .

compute povline = medianey * 0.5 .
descriptives variables = povline .
compute poor = 0 .
if ey lt povline poor = 1 .
if ey lt povline gap = povline-ey .
frequencies variables = poor .
descriptives variables = gap .
weight OFF .
frequencies variables = poor .
```

Results

Median equivalised income	*96 332*
Poverty line	*96,332 / 2 = 48,166*
How many poor households are in the sample?	*588*
How many poor individuals are there in the total population?	*277 263*
Head Count Ratio	*5.43%*
What is the average income gap among poor individuals?	*12 317*
Income Gap Ratio	*12,317 / 48,166 = 25.57%*

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.
- 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).
- Only two indicators of poverty are mentioned here, but there are several others. 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.

11. Elderly and Child Poverty Rates

Goal

Households with children and/or the elderly are usually at higher risk of poverty. The rising interest in analysing the incomes of these groups has increased the use of child and elderly poverty rates. (See, for example, the LIS key figures.)

Activity

Use data for Finland and the US in 2000. Calculate the Head Count Ratio and the Income Gap Ratio for the total population, the elderly, and for children.

Guidelines

- Prepare the data as you did in the previous exercise (drop observations with missing or zero ***dpi***).
- All surveyed households and their members must be included in the estimates of the poverty line. After the (unique) poverty line has been calculated, only those households with either members under the age of 18 (for the child poverty figures) or over 64 (elderly figures) are included when computing the proportion of the population (subgroup) living in poverty.
- One way to consider a subgroup in your calculations is to change the weights. In this case, you can create two additional weights: one for households with children and one for households with elderly. These weights will be equal to 0 if there are no children/elderly in the household, but will be equal to the normal weight multiplied by the number of children/elderly (and not total number of household members!):

  ```
  compute cwt = hweight * d27.
  compute ewt = hweight * (num6574+numge75).
  ```

- For this exercise, construct the indicator three times: once for each group you want to examine. For each indicator, you will use the corrected weight.

Program

```
title "** INCOME DISTRIBUTION I – Exercise 11 **"


define keepvars () .
  hweight d4 d5 d27 num6574 numge75 dpi .
!enddefine .

define pov () .
select if dpi ne 0 .
select if d5 ne 3 .

* run unweighted number of obs .
descriptives variables = hweight.
temporary .
select if d27 gt 0.
descriptives variables = hweight.
temporary .
select if num6574 gt 0 or numge75 gt 0.
descriptives variables = hweight.

compute ey = dpi/(d4**0.5) .
compute wt = hweight*d4 .
compute cwt = hweight*d27 .
compute ewt = hweight*(num6574+numge75) .
weight by wt .

*** code to use median within same session.
compute group = 1 .
rank ey by group /ties = high  /rfraction into rankey .
compute overmid = 0 .
if rankey gt 0.5 overmid = 1.
temporary .
select if  overmid eq 1 .
aggregate outfile = outmedey
   / presorted
   / break = group
   / medianey = min(ey) .
match files
   / table = outmedey
   / file = *
   /drop = rankey
   / by group .

compute povline = medianey * 0.5 .
compute poor = 0 .
if ey lt povline poor = 1 .
if ey lt povline gap = povline-ey .
```

```
* poverty line, hcr and gap .
descriptives variables = povline .
frequencies variables = poor .
descriptives variables = gap .
weight by cwt .
descriptives variables = povline .
frequencies variables = poor .
descriptives variables = gap .
weight by ewt .
descriptives variables = povline .
frequencies variables = poor .
descriptives variables = gap .
!enddefine .

get file = fi00h / keep = keepvars .
pov .
get file = us00h / keep = keepvars .
pov .
```

Results

	# Observations	Poverty line	HCR	IGR
Finland 2000				
Total	*10 421*	*48,166.15*	*5.43%*	*25.57%*
with children	*3 893*	*48,166.15*	*2.97%*	*44.70%*
with elderly	*1 837*	*48,166.15*	*8.47%*	*12.20%*
US 2000				
Total	*49 351*	*12,046.99*	*17.05%*	*33.45%*
with children	*18 030*	*12,046.99*	*21.93%*	*34.13%*
with elderly	*11 634*	*12,046.99*	*24.72%*	*29.60%*

Comments

- As already mentioned above, the poverty line remains the same whether one examines the entire population, children, or the elderly but the households included in calculating the percentage below the threshold change with each subgroup. Therefore, keep in mind that the sample sizes of these subgroup rates are based on fractions of the entire sample and treat these figures with care when interpreting your results.

INCOME DISTRIBUTION II

Ex. 12: Dealing with Extreme Values: Trimming & Bottom- /Top-coding

Ex. 13: Inequality: the Gini Index

Ex. 14: Sensitivity Analysis Using Different Concepts of Income

Ex. 15: Sensitivity Analysis Using Different Equivalence Scales

12. Dealing with Extreme Values: Trimming and Bottom- / Top-coding

Goal

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). Therefore, comparative researchers sometimes ‘trim’ the distribution (by deleting the top and bottom 1% for example) or impose ‘bottom codes’ and ‘top codes’ to provide a common calculation of lower and upper limits, method often referred to as ‘winsorising’.

Activity

Use the data for Sweden 2005. Remove all missing and zero values of household disposable income. Using both the trimming and winsorising methods, create the following two new variables:

- variable ***trim***, where the top 1% and bottom 1% of weighted household disposable income (***dpi***) is set to missing (trimming);
- variable ***wins*** where the top 1% and bottom 1% of weighted household disposable income (***dpi***) are set respectively to the value of the 1st and 99th percentile (winsorising).

Compare the mean, median, and the first four and last four observations of the household income before the changes, after trimming, and after winsorising.

Guidelines

- You can easily find the values of the 1st and 99th percentiles of disposable income by using the **`frequencies`** command with the option **`percentiles`**. However, these values are only displayed, and cannot be used for further calculations. The first way of solving this problem is to run the program in two sessions; the first to display the value of the percentile, and a second session with the manually typed values in the program. This way of working is both cumbersome, and error-prone!
- In order to facilitate things, LIS has prepared a block of code that matches any percentile (therefore also valid for the median which is nothing else than the 50th percentile) to your existing data. The block of code is to be found in the program below. In case you need to repeat the calculation more than once, you can consider treating the block of code like it were a routine.
- This code requires two parameters to be assigned before calling it : which percentile, and for which income variable you want your calculations. For instance , for the 25th percentile of DPI, use the following two lines :

```
compute inc_var = dpi .

compute pctl = 25 .
```

- The routine will create a new variables for the chosen percentile; its name being ***pctli*** .Since in this exercise we need to trim both ends of the distribution, we will need to run the routine twice, once for the first percentile, and a second time for the 99th percentile. As the name of the new variable as produced within the routine remains constant, be aware to copy the contents of pctli into a new variable, otherwise the contents gets lost with the second run, for instance like :

```
compute pctl99 = pctli .
```

- To see the smallest and largest observation, you can use the minimum and maximum from:

```
descriptives variables = dpi trim wins.
```

- The median will not be produced by the **`descriptives`** command. Therefore one needs to use the **`frequencies`** command, while specifying the option **`statistics`**. Remind that running frequencies on continuous variables (like wage) will produce a huge listing!! This must be avoided, and can be done by adding the **`format`** option, like this:

```
frequencies variables = dpi
    /  statistics = median
    /  format = notable .
```

Program

```
title "** INCOME DISTRIBUTION II – Exercise 12 **" .

get file = se05h /keep = hweight dpi .

select if dpi ne 0 .
select if not missing(dpi) .
weight by hweight .
compute wins = dpi.
compute trim = dpi.
compute inc_var = dpi.
compute pctl = 99.

*** code to use percentile within same session.
compute group = 1 .
rank inc_var by group /ties = high  /rfraction into rankinc .
compute overmid = 0 .
compute pctld = pctl/100 .
if rankinc gt pctld overmid = 1.
temporary .
select if  overmid eq 1 .
aggregate outfile = outpctli
  / presorted
  / break = group
  / pctli = min(inc_var) .
match files
  / table = outpctli
  / file = *
      /drop = rankinc
  / by group .


***  topcoding, winsorizing.
compute pctl99 = pctli.
if dpi gt pctl99 wins = pctl99 .
frequencies variables = dpi wins trim
   / statistics = median default
   / format = notable .
save outfile = id_ex12 .
get file = id_ex12 .
compute pctl = 1.


*** code to use percentile within same session.
compute group = 1 .
rank inc_var by group /ties = high  /rfraction into rankinc .
compute overmid = 0 .
compute pctld = pctl/100 .
if rankinc gt pctld overmid = 1.
```

```
temporary .
select if  overmid eq 1 .
aggregate outfile = outpctli
  / presorted
  / break = group
  / pctli = min(inc_var) .
match files
  / table = outpctli
  / file = *
      /drop = rankinc
  / by group .


***  bottomcoding, winsorizing.
compute pctl1 = pctli.
if dpi lt pctl1 wins = pctl1 .
frequencies variables = dpi wins trim
   / statistics = median default
   / format = notable .

***  topcoding, trimming.
if dpi gt pctl99 trim = $sysmis .
frequencies variables = dpi wins trim
   / statistics = median default
   / format = notable .

***  bottomcoding, trimming.
if dpi lt pctl1 trim = $sysmis .
frequencies variables = dpi wins trim
   / statistics = median default
   / format = notable .

weight off .
frequencies variables = dpi wins trim
   / statistics = min max
   / format = notable .
```

Results

		Original values	**After trimming**	**After winsorising**
Number of valid observations		*16,268*	*15,918*	*16,268*
Average income		*269,551*	*262,484*	*265,713*
Median income		*223,861*	*223,861*	*223,861*
Income level of the first four observations **(smallest incomes)**	Smallest:	*-1,053,732*	*43,066*	*43,066*
	2nd smallest:	*-813,940*	*43,487*	*43,066*
	3rd smallest:	*-270,365*	*43,644*	*43,066*
	4th smallest:	*-239,543*	*43,671*	*43,066*
Income level of the last four observations **(highest incomes)**	4th largest:	*6,542,836*	*803,085*	*806,076*
	3rd largest:	*6,746,146*	*803,952*	*806,076*
	2nd largest:	*7,609,412*	*804,307*	*806,076*
	Largest:	*1,072,029,135*	*806,076*	*806,076*

13. Inequality: the Gini Index

Goal

This exercise introduces the Gini index, which is one of the most commonly used income inequality indicators.

Activity

Calculate the Gini index on total disposable income for Finland and the US in 2000, after bottom-coding disposable income at 1 percent of its equivalised mean and top-coding at 10 times the unequivalised median.

Guidelines

- Prepare the data as you did in the previous exercise (drop observations with missing or zero ***dpi***).
- In the previous exercise you have seen two different methods of dealing with extreme values, trimming and winsorising (or bottom-/top-coding). The LIS key figures are calculated using a particular type of bottom-/top-coding, which we will to replicate in this exercise. The bottom-coding is carried out after the equivalisation of income (on the equivalised income distribution), while the top-coding is carried out before (on the unequivalised distribution) in the following way:
  - at the bottom of the distribution, equivalised income is bottom-coded at 1 percent of its equivalised mean, i.e. all observations for which equivalised income is lower than 1% of the average equivalised income are set to that value.
  - at the top of the distribution, income is top-coded at 10 times the unequivalised median, i.e. all observations for which unequivalised income (or dpi) is higher than 10 times the median unequivalised income are set to that value.

- Producing the Gini itself is complex: graphically it represents the surface between the Lorenz curve and the line of ideal equality. The translation into SPSS syntax requires many lines of code, which you again will find as a block of pre-prepared code in the program below, this time in the form of a routine.

Program

```
title "** INCOME DISTRIBUTION II – Exercise 13 **" .


define keepvars ().
  hweight d4 d5 dpi.
!enddefine.

define prepare ().
select if dpi ne 0 .
select if d5 ne 3 .
compute wt = hweight*d4 .
weight by wt.
!enddefine .



*** code to produce gini coefficient .
define gincalc ().
compute ey = inc_var/(d4**0.5).
compute group = 1 .
aggregate outfile = outmean
  / presorted
  / break = group
  / meaney = mean(ey) .
rank inc_var by group /ties = high  /rfraction into ranky .
compute overmid = 0 .
if ranky gt 0.5 overmid = 1.
temporary .
select if  overmid eq 1 .
aggregate outfile = outmed
  / presorted
  / break = group
  / medinc = min(inc_var) .
match files
  / file = *
  / table = outmean
  / by group .
match files
  / file = *
  / table = outmed
  / by group .
*** top and bottomcoding.
if inc_var gt (10*medinc) ey = (10*medinc)/d4**0.5 .
if ey lt (0.01*meaney) ey = 0.01*meaney .
*** gini procedure.
sort cases by ey (a).
compute cumwgt=cumwgt + wt.
leave cumwgt.
aggregate outfile= gini
    /presorted
    /break= group
    /meany= mean(ey)
    /meanr= mean (cumwgt)
    /n=n.
match files /file= * /table= gini /by group.
compute devy= ey - meany.
compute rank= cumwgt/n.
compute devr = (rank - 0.5).
compute prod= devy*devr.
aggregate outfile= *
    /presorted
    /break=group
    /sumprod= sum(prod)
    /meany= mean(ey)
    /n=n.
compute cov= sumprod/(n-1).
compute gini=cov*2/meany.
formats gini (f10.4).
list var=gini .
!enddefine .


get file = us00h /keep = keepvars .
prepare .
compute inc_var = dpi.
gincalc .

get file = fi00h /keep = keepvars .
prepare .
compute inc_var = dpi.
gincalc .
```

Results

	Gini
US 2000	*0.36823*
Finland 2000	*0.24621*

Comments

- The Gini index ranges between 0 and 1, with inequality increasing with an increasing index. A value of 0 means there is a completely equal distribution of income; a 1 refers to the extreme situation of one household holding the total population income, and all the rest having no income at all.
- As expected, inequality is much larger in the US than in Finland.
- To see the Ginis for all LIS datasets online, go to http://www.lisproject.org/key-figures/key-figures.htm.

14. Sensitivity Analysis Using Different Concepts of Income

Goal

When analysing inequality, you should always utilise multiple methods when comparing different countries. Inequality measures can be affected by the indicator used or by the measure of income.

The Gini index may change enormously when it is calculated on income concepts other than disposable income (***dpi***), which is calculated after transfers and taxes. Instead, you may want to look at market income (***mi***), which is calculated before taxes and transfers. By computing a Gini index on both income concepts, you can analyse the effect of government influence on the income distribution.

Activity

Calculate the Gini index for Finland and the US in 2000 for both market income and total disposable income, after bottom and top coding as in the previous exercise.

Guidelines

- Prepare the data as you did in the previous exercise. Be careful when choosing your sample. For each of the two income concepts, there will be a different sample of "valid" values (i.e., not missing and not zero). Since we are comparing the two final measures, we want to use the same sample in the two cases. In this exercise, use the sample that drops observations for which ***dpi*** is zero or missing. In this case, observations with zero mi but "valid" ***dpi*** are kept.
- In this exercise, you will need to create two measures of income: an equivalised disposable income (***dpi***) as well as an equivalised market income (***mi***).
- Use the subroutine for the gini twice and make sure to define the respective income variable (***inc_var***) each time before calling the subroutine.

Program

```
title "** INCOME DISTRIBUTION II – Exercise 14 **" .


define keepvars ().
  hweight d4 d5 mi gi dpi.
!enddefine.

define prepare ().
select if dpi ne 0 .
select if d5 ne 3 .
compute wt = hweight*d4 .
weight by wt.
!enddefine .

*** code to produce gini coefficient .
define gincalc ().
compute ey = inc_var/(d4**0.5).
compute group = 1 .
aggregate outfile = outmean
  / presorted
  / break = group
  / meaney = mean(ey) .
rank inc_var by group /ties = high  /rfraction into ranky .
compute overmid = 0 .
if ranky gt 0.5 overmid = 1.
temporary .
select if  overmid eq 1 .
aggregate outfile = outmed
  / presorted
  / break = group
  / medinc = min(inc_var) .
match files
  / file = *
  / table = outmean
  / by group .
match files
  / file = *
  / table = outmed
  / by group .
*** top and bottomcoding.
if inc_var gt (10*medinc) ey = (10*medinc)/d4**0.5 .
if ey lt (0.01*meaney) ey = 0.01*meaney .
*** gini procedure.
sort cases by ey (a).
compute cumwgt=cumwgt + wt.
leave cumwgt.
aggregate outfile= gini
    /presorted
    /break= group
    /meany= mean(ey)
```

```
   /meanr= mean (cumwgt)
   /n=n.
match files /file= * /table= gini /by group.
compute devy= ey - meany.
compute rank= cumwgt/n.
compute devr = (rank - 0.5).
compute prod= devy*devr.
aggregate outfile= *
   /presorted
   /break=group
   /sumprod= sum(prod)
   /meany= mean(ey)
   /n=n.
compute cov= sumprod/(n-1).
compute gini=cov*2/meany.
formats gini (f10.4).
list var=gini .
!enddefine .



get file = us00h /keep = keepvars .
prepare .
compute inc_var = mi.
gincalc .

get file = us00h /keep = keepvars .
prepare .
compute inc_var = dpi.
gincalc .

get file = fi00h /keep = keepvars .
prepare .
compute inc_var = mi.
gincalc .

get file = fi00h /keep = keepvars .
prepare .
compute inc_var = dpi.
gincalc .
```

Results

Gini	***mi***	***dpi***
US 2000	*0.478*	*0.368*
Finland 2000	*0.463*	*0.246*

Comments

- The Gini index is obviously much larger when calculated on the market income than on net income, since both transfers and taxes have a redistributive purpose.
- Even though Finland remains the more equal of the two countries even if looking at the ***mi*** measure, the difference becomes much less evident; the much larger decrease in Gini (when going from the pre-tax and transfers to the post-tax and transfer income concept) in Finland with respect to the US, points to two completely different social systems in terms of income redistribution.
- As already mentioned, there are quite a few LIS datasets that contain net income instead of gross income data, including some large countries like France and Italy. The income data from these countries are net of taxes and thus not readily comparable to the gross income data from other countries.

15. Sensitivity Analysis Using Different Equivalence Scales

Goal

The LIS equivalence scale (square root of the number of household members) is just one among many possible equivalence scales. Obviously, the choice of the scale will have an impact on the measure calculated. The difference is all the more important when you are considering specific subgroups of the population that are treated differently by the different equivalence scales. In this exercise we will see how the head count ratio changes when using the OECD equivalence scales (modified and original) with respect to the LIS equivalence scale.

Activity

Calculate the head count ratio (HCR) for the total population, the child population, and the elderly population in Finland in 2000. Use the LIS equivalence scale, the OECD modified scale, and the OECD original scale to calculate three different measures of equivalised income.

Guidelines

- This exercise utilizes three different equivalence scales:
  - LIS scale = square root of the number of persons;
  - OECD modified scale = 1 + 0.5*number of other adult members + 0.3 * number of children below 14;
  - OECD original scale = 1 + 0.7*number of other adult members + 0.5 * number of children below 14.
- To calculate the head count ratio with the three equivalence scales for the three population groups, calculate three different equivalised incomes:

```
compute escale = d4**0.5 .
compute escale = 1 + 0.5*(d4-1-d29) + 0.3*d29).
compute escale = 1 + 0.7*(d4-1-d29) + 0.5*d29).
```

Program

```
title "** INCOME DISTRIBUTION II – Exercise 15 **" .


get file = fi00h /keep = d4 d5 d27 d29 dpi num6574 numge75 hweight .
* prepare file.
select if dpi ne 0 .
select if d5 ne 3 .
compute escale1 = d4**0.5 .
compute escale2 = 1 + 0.5*(d4-1-d29) + 0.3*d29.
compute escale3 = 1 + 0.7*(d4-1-d29) + 0.5*d29.
compute wt = hweight*d4 .
compute cwt = hweight*d29 .
compute ewt = hweight*(num6574+numge75) .

*** code to use median within same session.
define medcalc () .
compute group = 1 .
rank ey by group /ties = high  /rfraction into rankey .
compute overmid = 0 .
if rankey gt 0.5 overmid = 1.
temporary .
select if  overmid eq 1 .
aggregate outfile = outmedey
  / presorted
  / break = group
  / medianey = min(ey) .
match files
  / table = outmedey
  / file = *
      /drop = rankey
  / by group .
!enddefine .

define pov () .
medcalc .
compute povline = medianey * 0.5 .
compute poor = 0 .
if ey lt povline poor = 1 .
!enddefine .

weight by wt .
compute ey = dpi/escale1 .
pov .
frequencies variables = poor .
weight by cwt .
frequencies variables = poor .
weight by ewt .
frequencies variables = poor .
```

```
save outfile = 'id_ex15' .
get     file = 'id_ex15' .

weight by wt .
compute ey = dpi/escale2 .
pov .
frequencies variables = poor .
weight by cwt .
frequencies variables = poor .
weight by ewt .
frequencies variables = poor .

save outfile = 'id_ex15' .
get     file = 'id_ex15' .

weight by wt .
compute ey = dpi/escale3 .
pov .
frequencies variables = poor .
weight by cwt .
frequencies variables = poor .
weight by ewt .
frequencies variables = poor .
```

Results

HCR	**LIS equivalence scale**	**OECD modified scale**	**OECD original scale**
Total population	5.432	4.324	3.872
Children (< 14)	3.257	3.104	5.591
Elderly	8.469	4.736	1.179

Comments

- As expected, the HCR changes when using different equivalence scales. This is especially true for child and elderly poverty rates.
- The poverty rate changes substantially between different equivalence scales. Equivalised income *ey* is calculated as:

  $$ey = y / d^{\varepsilon}$$

  where *y* is the unequivalised disposable income (***dpi*** in LIS terms), *d* is the size of the family (***d4*** in LIS terms) and ε is the equivalence elasticity. ε varies between 0 (indicating full economies of scale and no need to adjust household income with size) and 1 (no economies of scale, which indicates the need to use per capita income). Obviously, the larger the elasticity, the smaller the economies of scale assumed by the equivalence scale. HCR increases with ε for large households, whereas it decreases for small households. This is because with a higher ε (i.e., smaller economies of scale), large households will need more income to get the same standards of living, whereas smaller households need less income. The LIS equivalence scale uses ε= 0.5; the original OECD scale corresponds to a value of ε≈0.7; and the modified OECD scale stands somewhere in the middle. As a result, when using the OECD scales, poverty will generally be higher in households with children (which tend to be larger than the average household), and much lower in households with elderly (which tend to be much smaller).

LABOUR MARKET

Ex. 16: Calculating Employment Rates

Ex. 17: Calculating Unemployment Rates

Ex. 18: Person- and Job-specific Characteristics of Employment

16. Calculating Employment Rates

Goal

The employed population can be defined differently depending on the research purpose. In labour market analysis, you may wish to calculate employment rates or run wage regressions. In this case, you may want to identify persons working at a certain moment in time. Within the framework of this strict reference period, a person may be considered as employed as soon as he/she has carried out any work (i.e., the International Labour Organisation (ILO) definition of employment, LIS variable ***pclfs***). Those more interested in the primary employment status at a certain moment in time should not use the ILO definition, but rather look at those whose main activity is employment (LIS variable ***pcmas***).

Those wishing to perform income distribution analyses according to the activity status of individuals (e.g., calculation of poverty rates for the working poor), may choose to focus on employment over a longer reference period. In this case, you need to be able to identify an individual's primary activity over the income reference period (which in LIS is normally one year, LIS variable ***pumas***).

In this exercise, we identify the employed population according to these three different concepts of employment.

Activity

Calculate employment rates in the US and Germany in 2000 using the three different employment measures. Use the sample of individuals 18 years of age and older. Recode all of those with unclear employment attachment (coded in 900s) to missing.

Guidelines

- The employment rate is the percentage of employed persons in the adult population, which we define here as 18 years of age or older.
- There are three different LIS variables that coincide with the definitions of employment described above:
  - persons carrying out any work at present can be identified using ***pclfs***;
  - persons for whom work is the main activity at present are in ***pcmas***;
  - persons who are mainly employed during a longer reference period can be found in ***pumas***.
- By definition, those coded in the 900s may have some employment attachment, but the information is too ambiguous to either define them either as "employed" or "not employed". These individuals were kept in separate detailed categories to allow users to redefine them based on the specific research project. Always check the country-level documentation to determine how you wish to recode those assigned to the 900 (and 400, in ***pclfs***) category.
  - In your *SPSS* code, you should always try to be concise as possible. This will minimize the possibility that LISSY will reject your program for overly long output. To create dummy variables, an efficient method of programming is to use:

    ```
    recode <original variable names> (old_val1 = new_val1) into <new
    variable names>.
    ```
- Tabulate the 3 variables to get the percentage of employed (remember to weigh the results).

Program

```
title "** LABOUR MARKET – Exercise 16 **" .

define dofiles () .
select if page ge 18.
recode pclfs (100 thru 199 = 1) (200 thru 499 = 0)
       (900 thru 999 = sysmis) (-1  = sysmis) into emp1.

recode pcmas (100 thru 199 = 1) (200 thru 299 = 0)
       (900 thru 999 = sysmis) (-1  = sysmis) into emp2.

recode pumas (100 thru 199 = 1) (200 thru 299 = 0)
       (900 thru 999 = sysmis) (-1  = sysmis) into emp3.

weight by pweight .
frequencies variables = emp1 to emp3.
!enddefine .

get file = us00p
    / keep = country pweight page pclfs pcmas pumas .
dofiles .

get file = de00p
    / keep = country pweight page pclfs pcmas pumas .
dofiles .
```

Results

	US00	**DE00**
At present (any work)	*65.5*	*59.6*
At present (main activity)	*59.3*	*45.8*
During the calendar year (main activity)	*63.2*	*54.4*

Comments

- As expected, the employment rate is much higher if employment is defined as any work (***pclfs***) than if it is defined as the main activity held (***pcmas*** or ***pumas***).
- Be sure to look at the Value Labels and read the Comments/Warnings in the Lissification tables! This is where you will find information about the 900 category.
  - In most cases, the difference between including the 900s or setting them to missing in ***pclfs*** is minor or non-existent. Determining whether an individual held a job in a specified period is fairly straightforward and the 900 category is reserved for truly marginal cases.
  - More ambiguity arises when defining primary activity (***pcmas*** and ***pumas***). This happens most often when questions about primary activity focus on employment. In some surveys, questions about non-employment are only asked of those who were not employed in ***pclfs***, so those who appear to have marginal attachment to the labour force (e.g., part-time or irregular workers), but who have specified no other activity are coded in the 900s.

17. Calculating Unemployment Rates

Goal

The definition of the unemployment rate is quite standard: the percentage of the unemployed population over the active population at a given point in time. The active population includes both the employed and the unemployed, but not the non-employed. The definition of the unemployed population, however, varies considerably. The main difficulty in calculating unemployment rates lies in determining the identity of the unemployed.

The most widely recognised measure of unemployment is that of International Labour Organisation (ILO). According to the ILO, the employed are those who had at least one paid hour of work (or were temporarily absent from work) during a given reference week. The unemployed are those who were not employed during the reference week, were actively searching for a job during the previous 4 weeks, and were available to start a new job in the following 2 weeks.

However, unemployment can be measured in a variety of ways, and this is reflected in the surveys used by LIS. Criteria range from registration at the unemployment office, receipt of unemployment benefits, self-declared unemployment, or job search. Unfortunately not all concepts are available for all datasets, so it is important to read the documentation very carefully when attempting to compare the unemployment rates in two different datasets.

In this exercise, we will use the ILO definition of unemployment to calculate unemployment rates.

Activity

For the UK in 1999 and Germany 2000 calculate unemployment rates (percentage of unemployed persons over the active population) with the ILO definition of unemployment (***pclfs*** **210-219**).

Consider those individuals 25 to 60 years of age to be in the working-aged population. Exclude those currently in the military.

Guidelines

- Identify the LIS variable to be used for the calculation of the unemployment rates. All of the different concepts of unemployment are all included in one LIS variable only.
- Create a dummy variable for unemployed following the definition given above. When defining the dummy, be very careful how you define the unemployed as well as how you define the "active population". (The universe depends on your definition!)
- Please note that all ILO unemployed are grouped in the semi-standardised codes 210-219. The registered unemployed, who are not ILO unemployed, are coded as 220-229. (Note, however, that it is possible to be both registered and ILO unemployed (or unknown ILO), so be careful when coding if you are only interested in the registered unemployed.)
- In the UK:
  - There are many programs that combine unemployment benefits with work and/or training (values 910s). For this exercise, they should be considered registered unemployed (as well as included in the "all unemployed" categories), but not ILO unemployed.
  - The potentially unemployed are those in the 400 category.
- Military personnel (whether at work or on leave) are always coded from 180 to 189 in ***pclfs***.
- Tabulate the variables to get the unemployment rate (remember to weight the results).

Program

```
title "** LABOUR MARKET – Exercise 17 **" .

define dofiles () .
select if (page ge 25 and page le 60).
recode pclfs (100 thru 179 = 0) (190 thru 199 = 0)
       (210 thru 219 = 1) into unemp.
variable labels
  unemp "ILO unemployment" .
weight by pweight .
frequencies variables = unemp.
!enddefine .

get file = uk99p
     / keep = country pweight page pclfs .
dofiles .
get file = de00p
     / keep = country pweight page pclfs .
dofiles .
```

Results

Unemployment rate, at present	***UK99***	***DE00***
ILO unemployed	*4.6*	*4.5*

Comments

- The broadest unemployment definition (including those who may potentially be unemployed) may vary considerably from dataset to dataset. In Germany, it includes discouraged workers and those with some vague intent of working in the future, as well as those who were previously unemployed, but were receiving sickness or maternity benefits during the reference week. In the UK, it includes discouraged workers and those in work or training through the employment service.
- When comparing LM variables across datasets with the use of the semi-standardised codes, look very carefully at the country-specific documentation in order to interpret the results.

18. Person and Job specific Characteristics of Employment

Goal

There are a number of employment characteristics that may interest researchers. This exercise introduces you to a few of the person- and job-specific variables that are available from LIS.

Activity

For Spain in 2000, create a SPSS dataset containing ***page***, ***pweight***, ***pclfs***, and the following:

- Characteristics of the Individual
  - continuous: ***phoursu***, ***pweektl***
  - categorical: ***psecjob***, ***psearch***, ***pcare***
- Characteristics of the Primary Job
  - categorical: ***pactiv***, ***pind***, ***ptypewk***, ***pfulpar***

This exercise uses a sample of adults (16 and over). From ***pclfs*** create a dummy variable (**emp**) that will be coded 0 for individuals that are not employed and 1 for individuals that are employed. Using the characteristics of the individual (continuous variables ***phoursu***, ***pweektl*** as well as the newly created employment dummy variable, ***emp***), find the weighted mean, minimum, and maximum. Repeat the same calculations (for the same variables) after you "clean" the data (see the **Guidelines** for instructions).

Finally, tabulate the dummy variable (***emp***) with each categorical variable. Run this tabulation twice: (1) without weights and with those individuals who are coded -1; (2) with weights and excluding those who are coded -1.

Use the information from your output and/or the documentation to answer the following questions:

1. What are the minimum and maximum numbers of usual weekly hours for the employed?

   ____

   Explain these results.

   ____

   ____

   What are the implications for interpreting the average hours worked reported above?

   ____

   ____

   ____

2. Why does ***phoursu*** equal 0 for all observations of those not employed? When is it possible for ***phoursu*** to equal to 0 for some of those employed?

   ____

   ____

   ____

3. How is it possible that some of those who are not employed reported 52 weeks of employment?

____

____

____

4. Which variables have no observations equal to -1 among the employed and all observations equal to -1 among those not employed?

____

What does this tell us about the universe of those variables?

____

____

5. Why does ***ptypewk*** have some observations equal to -1 among the employed?

____

____

____

6. Which variables have no observations equal to -1 for either group?

____

What does this tell us about the universe of those variables?

____

____

Guidelines

- When creating your employment dummy variable, be sure to include only those you can clearly identify as being either employed or not employed. In other words, be careful what you do with the 900s codes in ***pclfs***.
- The variables ***phoursu*** and ***pweektl*** (as well as all other time variables), are peculiar in that they may be continuous and discrete at the same time: all the values between 0 and 999 are normal continuous values, representing the duration of time (either hours of weeks), but on top of those there may be some discrete values, notably:
  - the value -9 for ***phoursu*** and ***phoursa*** stands for "hours vary";
  - all the values greater or equal to 1000 indicate partial or incomplete information, where the 1000 is a flag and stands for "at least xx number of hours" (see Section D of the introduction of the Labour Market Variables Guidelines).

  As a consequence, none of these values should be used as such (as they do not represent actual numbers of hours / weeks), and they should be excluded from the sample unless recoded to a valid value.
- For the continuous variables, you will need two different commands for each variable: (1) for the entire sample; and (2) for the restricted (or cleaned) sample. In both cases, use the **`split files`** command to have the values for the employed and the not employed separately.

- For the categorical variables, you will need two different commands for each variable. In order to get the sample frequencies of a specific value of a variable, remember not to weight the results. To get the correct estimated percentages among the valid observations, you will need to carry out a weighted tabulation excluding -1 from the valid observations. In both case, use the `split files` command to have the values for the employed and the not employed separately.

Program

```
title "** LABOUR MARKET – Exercise 18 **" .


get file = es00p
    / keep = pweight page pclfs pactiv pind ptypewk pfulpar
               phoursu pweektl psecjob psearch pcare .
select if page ge 16 .
recode pclfs (100 thru 199 = 1) (200 thru 999 = 2)
     (-1 = sysmis) into emp .
sort cases by emp .
split file by emp .
*** run unweighted number of observations / -1 / missing .
descriptives variables = phoursu pweektl .
frequencies variables = pactiv pind ptypewk pfulpar psecjob psearch pcare .
temporary.
select if phoursu ge 0 and phoursu le 999 .
descriptives variables = phoursu .
temporary.
select if pweektl ge 0 and pweektl le 52 .
descriptives variables = pweektl .
*** introduce the weight.
weight by pweight .
descriptives variables = phoursu pweektl .
*** weighted percentages without -1.
temporary.
select if ((pactiv ne -1) or missing(pactiv)) .
frequencies variables = pactiv.
temporary.
select if ((pind ne -1) or missing(pind)) .
frequencies variables = pind.
temporary.
select if ((ptypewk ne -1) or missing(ptypewk)) .
frequencies variables = ptypewk.
temporary.
select if ((pfulpar ne -1) or missing(pfulpar)) .
frequencies variables = pfulpar.
temporary.
select if ((psecjob ne -1) or missing(psecjob)) .
frequencies variables = psecjob.
temporary.
select if ((psearch ne -1) or missing(psearch)) .
frequencies variables = psearch.
temporary.
select if ((pcare ne -1) or missing(pcare)) .
frequencies variables = pcare.
```

Results

Table 1

Continuous variables		# observations	Mean	Min	Max
All values					
Employed					
	phoursu	*5,599*	*66.2*	*-9*	*1,072*
	pweektl	*5,350*	*51.8*	*0*	*1,048*
Not Employed					
	phoursu	*6,489*	*0*	*0*	*0*
	pweektl	*6,446*	*6.8*	*0*	*1,048*
“Clean” values only					
Employed					
	phoursu	*5,410*	*41.3*	*2*	*96*
	pweektl	*5,317*	*46.1*	*0*	*52*
Not Employed					
	phoursu	*6,489*	*0*	*0*	*0*
	pweektl	*6,421*	*3.1*	*0*	*52*

Table 2

Categorical Variables		Employed	Not Employed
Characteristics of the Individual			
psecjob	# obs = -1	*202*	*6,492*
	# missing obs	*154*	*50*
	% with multiple jobs	*3.14*	*NA*
psearch	# obs = -1	*0*	*0*
	# missing obs	*155*	*51*
	% looking for a job	*8.78*	*11.6*
pcare	# obs = -1	*0*	*0*
	# missing obs	*155*	*71*
	% caregivers	*22.61*	*21.5*
Characteristics of the Job			
pactiv	# obs = -1	*0*	*6,492*
	# missing obs	*134*	*50*
	% own-account workers	*11.52*	*NA*
pind	# obs = -1	*0*	*6,492*
	# missing obs	*134*	*50*
	% in construction	*11.33*	*NA*
ptypewk	# obs = -1	*202*	*6,492*
	# missing obs	*154*	*50*
	% in public sector	*16.46*	*NA*
pfulpar	# obs = -1	*0*	*6,492*
	# missing obs	*6*	*50*
	% in full-time employment	*87.2*	*NA*

Answers to questions 1 to 6:

1. The minimum number of usual weekly hours is -9 and the maximum one is 1072.

   -9 is the standard LIS value indicating the employee works irregular hours, while the 1000 is a flag for partial information on hours (10**xx** stands for "at least **xx** hours").

   The average hours worked by the employed are underestimated if the -9 values are included in the averages, while they are significantly overestimated if the values above 1000 are included. Negative values and values greater than or equal to 1000 should be excluded (or recoded) when calculating averages.

2. The employment dummy created in this exercise counts even those on leave and in marginal jobs as employed. Therefore, we would expect "usual" hours for the employed to include 0 hours (e.g., those currently on leave), but no one counted as not employed should have worked even one hour.

3. This is due to the different reference periods of the variables. The variable ***pclfs*** reports the labour force status (LFS) at the time of the interview, and ***pweektl*** reports total weeks worked during 2000. Looking at the country-specific survey documentation shows that the interview takes place between October and December of 2001, so a person could have worked the entire previous year, but stopped in between the end of 2000 and the interview.

4. Variables ***pactiv***, ***pind*** and ***pfulpar*** have no observations equal to -1 among the employed and all observations equal to -1 among those not employed.

   This means that the universe of those variables (i.e., the individuals to whom this question is relevant) is exactly all employed persons as reported in **pclfs**.

5. Different from the other variables about primary job characteristics, the universe of ***ptypewk*** is smaller than all employed persons. The documentation tells us that the universe contains only persons who currently work at least 15 hours per week, not all employed individuals.

6. Variables ***psearch*** and ***pcare*** have no observations equal to -1 for either group.

   This means that the universe of those variables is at least all adults with known LFS, independently from whether they are employed or not.

STACKED DATA
(OPTION 1)

Ex. 19: A Cross-national Comparison Using Stacked Data

19. A Cross-national Comparison Using Stacked Data

Goal

In this exercise, we combine household- and person-level files across countries to run a regression estimating the usual hours of the working-aged civilian population, utilizing a number of the techniques learned in earlier exercises. In addition, we show the importance of weighting correctly when combining files with very diverse sample sizes and weighting structures.

Activity

For Ireland and Austria in 2000, do the following:

1. Using the household-level file:
   a. Create a sample containing ***casenum***, ***hweight***, ***d4***, ***d5***, and ***dpi***.
   b. Create a dummy variable (***poorhh***) indicating that a household is poor (as defined by having less than 50 percent of median equivalised disposable income).
   c. Create a variable (***count***) containing the number of households in the sample (that is the same for all observations within the country).
   d. Create a new weight that normalizes the household weight to 1.
   e. Save your file in the temporary LIS directory for later use. (See Guidelines below for details.)
2. Using the person-level file, create a sample of non-military working-aged adults (25 through 60).
   a. Include the variables ***country***, ***casenum***, ***pweight***, ***page***, ***psex***, ***peduc***, ***ptocc***, ***phoursu***, ***pclfs***, and ***pcare***.
   b. Using ***pclfs***, drop any individuals who are in the military (both regular armed forces and conscripts).
   c. Run the LIS *include* program to recode education. Recode ***educ*** to missing if education is not in one of the major classifications (1 to 3).
   d. Using ***pcare***, create a dummy variable (***ccare***) that indicates care of children.
   e. Recode ***phoursu*** to missing if the usual hours worked vary.
   f. Create dummy variables for gender (***female***) and country (***at00***).
3. For each country, merge the person- and household-level files keeping only working-age civilian adults in households without missing or zero dpi.
   a. Note: Your final file will be at the person level. You should only retain observations that include information from both of the two samples you created above.)
   b. Save the merged file with the same temporary file name as you used when saving the household data.
4. Create your final sample by appending all of your country-level merged files.
   a. Create a variable that defines each household in your combined sample (i.e., every combination of ***country*** and ***casenum*** will have a unique value).
   b. Save your final combined file to the temporary directory.

5. For this exercise, assume that the correct population model estimating the number of hours worked is as follows:

   ***phoursu = f(page educ poorhh female ccare* <country dummies>)**

   where <***country dummies***> are included for each country in your analysis (excluding, of course, the base country).

   a. Run a regression of your model for each country using the household weight. Since the sampling unit is the household, you will need to correct for the clustering of households.

   b. Run the same regression combining countries and including the dummy variable for Austria.

   c. Instead of using the household weight, use the normalized weight you created when you set up the household file running the regression using both countries.

Guidelines

- For household files, do not forget the standard data cleaning procedures (drop missing or 0 ***dpi***).
- When calculating equivalised household income, use the LIS equivalence scale (***dpi***/sqrt(***d4***)).
- For this exercise, no bottom- or top-coding is necessary.
- As already mentioned in Exercise 5, when saving temporary files, you can place them in a directory at LIS. In this exercise, you will create a file for each country. In order that your filename differs from others saving files at the same time, save your files using your name as suffix. For example, Toby would save Austria by typing:

  ```
  save outfile = “at_toby” .
  ```

  and Ireland using:

  ```
  save outfile = “ie_toby” .
  ```

- Remember that the education routine to standardize education levels across countries can be called by:

  ```
  Include file edurecop .
  ```

- The education recode program creates the variable ***educ*** that takes on the values of 1 (low) through 3 (high). (See Exercise 8 for more detail.) The value 9 is used for education levels that cannot be categorized.
- In some cases, surveys provide information that usual hours of work vary. Rather than coding these as missing and losing information, LIS codes variable usual hours in ***phoursu*** as -9. Users need to correct for this when estimating hours worked.
- Like in Exercise 5, merge person-level information to the household sample.

  ```
  match files
    / file = "atp_<your_id>"
    / table = "ath_<your_id>"
    / by casenum .
  ```

  Then save the combined file for later use:

  ```
  save outfile “att_<your_id>” .
  ```

- To put all of your country files together, you will need to append the data sets:

```
add files file =  "iet_<your_id>"
        / file = "att_<your_id>".
```

- Your regression could look something like:

```
regression
   / regwgt = hweight
   / dependent = phoursu
   / method = enter page educ female ccare poorhh .
```

Program

```
title "** STACKED DATA– Exercise 19 **" .


*** code to use median within same session.
define medcalc () .
compute group = 1 .
rank ey by group /ties = high  /rfraction into rankey .
compute overmid = 0 .
if rankey gt 0.5 overmid = 1.
temporary .
select if  overmid eq 1 .
aggregate outfile = outmedey
  / presorted
  / break = group
  / medianey = min(ey) .
match files
  / table = outmedey
  / file = *
      /drop = rankey
  / by group .
!enddefine .


get file = ie00p /keep = country casenum
                 page psex peduc ptocc phoursu pclfs
                 pcare.
select if page ge 25 and page le 60 .
select if pclfs le 179 or pclfs ge 190 .
include file edurecop .
if educ eq 9 educ = $sysmis .
if phoursu eq -9 phoursu = $sysmis .
recode pcare (100 thru 119 = 1)
   (120 thru 299 , 901 = 0) (-1 , 902 = sysmis) into ccare .
recode psex (1=0) (2=1) into female.
recode country (139 = 1) (137 = 0) into at .
save outfile = 'iep_id'   .


get file = ie00h
   /keep = casenum hweight d4 d5 dpi.
select if dpi ne 0 .
select if not missing(dpi) .
compute group = 1 .
aggregate outfile = "sumhwgt"
   / presorted
   / break = group
   / sumw = sum(hweight) .
match files
   / file = *
   / table = "sumhwgt"
   / by group .
compute normw = hweight / sumw .
weight by hweight.
compute ey = dpi/(d4** 0.5) .
medcalc .
compute povline = medianey * 0.5 .
compute poorhh = 0 .
if ey lt povline poorhh = 1 .
save outfile = 'ieh_id'   .


match files
   / file = "iep_id"
   / table = "ieh_id"
   / by casenum .
descriptives hweight normw
  /statistics = default sum .
weight off.
regression
      / regwgt = hweight
      / dependent = phoursu
      / method = enter page educ female ccare poorhh .
save outfile = "iet_id" .

get file = at00p /keep = country casenum
                 page psex peduc ptocc phoursu pclfs
                 pcare.
select if page ge 25 and page le 60 .
select if pclfs le 179 or pclfs ge 190 .
include file edurecop .
if educ eq 9 educ = $sysmis .
if phoursu eq -9 phoursu = $sysmis .
recode pcare (100 thru 119 = 1)
        (120 thru 299 , 901 = 0) (-1 , 902 = sysmis) into ccare .
recode psex (1=0) (2=1) into female.
recode country (139 = 1) (137 = 0) into at .
```

```
save outfile = 'atp_id'   .

get file = at00h
   /keep = casenum hweight d4 d5 dpi.
select if dpi ne 0 .
select if not missing(dpi) .
compute group = 1 .
aggregate outfile = "sumhwgt"
   / presorted
   / break = group
   / sumw = sum(hweight) .
match files
   / file = *
   / table = "sumhwgt"
   / by group .
compute normw = hweight / sumw .
weight by hweight.
compute ey = dpi/(d4** 0.5) .
medcalc .
compute povline = medianey * 0.5 .
compute poorhh = 0 .
if ey lt povline poorhh = 1 .
save outfile = 'ath_id'   .

match files
   / file = "atp_id"
   / table = "ath_id"
   / by casenum .
descriptives hweight normw
  /statistics = default sum .
weight off .
regression
      / regwgt = hweight
      / dependent = phoursu
      / method = enter page educ female ccare poorhh .
save outfile = "att_id"



add files file =  "iet_id"
```

```
        / file = "att_id" .
descriptives hweight normw
   / statistics = mean sum .
weight off .
regression
      / regwgt = hweight
      / dependent = phoursu
      / method = enter page educ female ccare poorhh at .
regression
      / regwgt = normw
      / dependent = phoursu
      / method = enter page educ female ccare poorhh at .
```

Results

		Weighted by ***hweight***			Normalised weight
		Ireland	Austria	Both	
Sum of weights		*$1.4*10^6$*	*$3.0*10^3$*	*$1.4*10^6$*	*2.5*
Coefficient P>|t|	Age	*-.26*	*-.41*	*-.26*	*-.35*
		0.000	*0.000*	*0.000*	*0.000*
	Education	*2.66*	*3.38*	*2.66*	*2.75*
		0.000	*0.000*	*0.000*	*0.000*
	Female	*-19.64*	*-12.24*	*-19.62*	*-16.07*
		0.000	*0.000*	*0.000*	*0.000*
	Care for Child	*-2.91*	*-5.49*	*-2.91*	*-4.09*
		0.002	*0.000*	*0.002*	*0.000*
	Poor household	*-18.52*	*-2.97*	*-18.51*	*-13.18*
		0.000	*0.307*	*0.000*	*0.000*
	Austria	---	---	*1.51*	*1.80*
		---	---	*0.015*	*0.004*
	Constant	*47.25*	*49.95*	*47.26*	*48.65*
		0.000	*0.000*	*0.000*	*0.000*

Comments

- Weights in Ireland inflate to the population, while weights in Austria sum to 1 and inflate to the sample size. The results you see when running the two countries together are, therefore, being driven completely off of the results from Ireland.
- Be sure you understand the weighting procedures in each country. If you have one country with inflating weights and another without, it is vital that you normalize the weights across countries. If all countries in your analysis inflate to population weights, you can keep the weights as they are if you are interested in weighting all households equally, but you should normalize if you want to give each country equal importance. **Make sure you know what you are estimating!**

LWS BASICS
(OPTION 2)

Ex. 20: Differences in Concepts of Net Worth

Ex. 21: Asset Participation

20. Differences in Concepts of Net Worth

Goal

Estimates of net worth can differ substantially depending on the wealth measure you use. In this exercise, you will begin to familiarize yourself with the summary measures in LWS and determine the differences in portfolio compositions for the whole population using two different definitions of net worth.

Activity

Calculate summary statistics (mean and median) for two LWS net worth concepts, ***nw1*** and ***nw2***, as defined in the *LWS Quick Reference Guide*. Determine the differences in portfolio compositions for these two measures in Italy and Sweden in 2002.

Guidelines

- Don’t forget to change the project in your job submission panel to LWS.
- Use the *LWS Quick Reference Guide* to identify the components of ***nw1*** and ***nw2***.
- For the first part of the exercise calculate the means and medians for the two net worth measures.
- For the second part of the exercise calculate the means of the components and take the appropriate ratios to find the shares.
- For business holdings, use the measure for business equity, if available. Otherwise, use business assets. In order to do this, you will need to check the country-specific documentation for the availability of business assets, business debt, and business equity.
- Use the country documentation to check for differences in variable construction.

Program

```
title "** LWS BASICS – Exercise 20 **" .

get file = it02w
     / keep = wgt nw1 nw2 tfa1 tnf1 td be ba .
compute ta = tfa1 + tnf1 + be.
weight by wgt.
frequencies variables = nw1 nw2
      /format = notable
      /statistics = median.
descriptives variables =  nw1 nw2 tfa1 tnf1 ta td be ba.

get file = se02w
     / keep = wgt nw1 nw2 tfa1 tnf1 td be ba .
compute ta = tfa1 + tnf1 + be.
weight by wgt.
frequencies variables = nw1 nw2
      /format = notable
      /statistics = median.
descriptives variables =  nw1 nw2 tfa1 tnf1 ta td be ba.
```

Results

	Italy 2002	**Sweden 2002**
Net worth (definition 1)		
Mean	*154,237*	*537,838*
Median	*98,000*	*165,120*
Net worth (definition 2)		
Mean	*177,766*	*617,798*
Median	*101,500*	*178,145*

	Italy 2002		**Sweden 2002**	
	nw1	***nw2***	***nw1***	***nw2***
Total Financial Assets (definition 1)	*14.9%* *(23,678)*	*13.0%* *(23,678)*	*28.1%* *(232,672)*	*25.6%* *(232,672)*
Total Non-financial Assets (definition 1)	*85.1%* *(134,955)*	*74.1%* *(134,955)*	*71.9%* *(595,631)*	*65.6%* *(595,631)*
Business Equity/Assets	----	*12.9%* *(23,526)*	---	*8.8%* *(79,955)*
Total Assets (sum of the 2 or 3 lines above)	*100%* *(158,633)*	*100%* *(182,159)*	*100%* *(828,303)*	*100%* *(908,258)*
Debt	*2.8%* *(4,398)*	*2.4%* *(4,398)*	*35.1%* *(290,513)*	*32%* *(290,513)*
Net worth (total assets – debt)	*97.2%* *(154,237)*	*97.6%* *(177,766)*	*64.9%* *(537,838)*	*68%* *(617,798)*

21. Asset Participation

Goal

The goal of this exercise is to familiarize yourself with different types of assets in the LWS data and to compare asset participation of the elderly with the population as a whole.

Activity

Calculate participation in the three assets (deposit accounts, stocks, investment real estate, business assets/equity) for the total population, and the elderly population in Finland in 1998, Italy 2002 and Sweden 2002.

Guidelines

- Use the *LWS Quick Reference Guide* to help you with the job submission.
- Identify the wealth variables needed to calculate the participation rates using the documentation to check whether each of these components exists in each of these countries.
- Create dummy variables for each of the wealth components to indicate that a household is holding a particular asset:

  ```
  compute dst = 0.
  if st gt 0 dst = 1.
  ```

- For business holdings, use the measure for business equity, if available. Otherwise, use business assets. In order to do this, you will need to check the country-specific documentation for the availability of business assets, business debt, and business equity.
- When measuring assets of the elderly population, define elderly households as those with a head or spouse 65 years of age or older.

Program

```
title "** LWS BASICS – Exercise 21 **" .

get file = fi98w /keep = wgt ageh ages da st ir ba be .
weight by wgt.
compute dda = 0.
if da gt 0 dda = 1.
compute dst = 0.
if st gt 0 dst = 1.
compute dir = 0.
if ir gt 0 dir = 1.
compute dbe = 0.
if ba gt 0 dbe = 1.
if be gt 0 dbe = 1.
compute eld = 0.
if ((ageh ge 65) or (ages ge 65)) eld = 1.
frequencies dda dst dir dbe .
temporary.
select if eld eq 1.
frequencies dda dst dir dbe .

get file = it02w /keep = wgt ageh ages da st ir ba be .
weight by wgt.
compute dda = 0.
if da gt 0 dda = 1.
compute dst = 0.
if st gt 0 dst = 1.
compute dir = 0.
if ir gt 0 dir = 1.
compute dbe = 0.
if ba gt 0 dbe = 1.
if be gt 0 dbe = 1.
compute eld = 0.
if ((ageh ge 65) or (ages ge 65)) eld = 1.
frequencies dda dst dir dbe .
temporary.
select if eld eq 1.
frequencies dda dst dir dbe .
```

```
get file = se02w /keep = wgt ageh ages da st ir ba be .
weight by wgt.
compute dda = 0.
if da gt 0 dda = 1.
compute dst = 0.
if st gt 0 dst = 1.
compute dir = 0.
if ir gt 0 dir = 1.
compute dbe = 0.
if ba gt 0 dbe = 1.
if be gt 0 dbe = 1.
compute eld = 0.
if ((ageh ge 65) or (ages ge 65)) eld = 1.
frequencies dda dst dir dbe .
temporary.
select if eld eq 1.
frequencies dda dst dir dbe .
```

Results

Total population	**Finland 1998**	**Italy 2002**	**Sweden 2002**
Deposit Accounts	*90.7*	*80.7*	*58.5*
Stocks	*32.9*	*10.1*	*36.3*
Investment Real Estate	*26.9*	*21.8*	*13.6*
Business Assets/Equity	---	*15.5*	*7.5*

Elderly Population 65+	**Finland 1998**	**Italy 2002**	**Sweden 2002**
Deposit Accounts	*88.4*	*72.9*	*70.2*
Stocks	*28.9*	*6.1*	*35.8*
Investment Real Estate	*29.9*	*19.5*	*14.6*
Business Assets/Equity	---	*4.7*	*8.3*

Comments

- Finland has a higher proportion of investments in stocks and real estate, but they also have a high investment in deposit accounts. In Sweden, investment is also high, but deposit accounts are lower, which suggests a portfolio with a riskier balance.
- Except in Sweden, deposit accounts are lower, suggesting a spending of funds as individuals age. In Sweden, however, deposits rise after 65, which may mean healthy retirement programs and/or a decrease in spending in later years.