README FILE
Simulation of synthetic micro-files and estimation of g-percentiles

Juliette Fournier

June 13, 2016

This directory contains ten programs:

- First, 5 programs that estimate the distribution of incomes (or wealth) from aggregate raw data:
  - the program `gen_pop.m` that generates a synthetic population from tax tabulations;
  - the program `gen_gperc.m` that estimates thresholds and average incomes corresponding to generalized percentiles;
  - the program `gen_gperc_allfiles.m` that generates files containing estimates of the thresholds and average incomes at generalized percentiles corresponding to all the raw data files provided in the subfolder `rawdata/allfiles`;
  - the program `gen_gperc_bysheet.m` that generates files containing estimates of the thresholds and average incomes at generalized percentiles corresponding to all the raw data tabulations provided in each of the worksheet of the file `rawdata_bysheet` which lies in the folder `rawdata`;
  - the program `gen_gperc_byyear.m` that generates files containing estimates of the thresholds and average incomes at generalized percentiles corresponding to all consecutive raw data subtables provided in the file `rawdata_byyear` which lies in the folder `rawdata`.
- Second, 5 programs that estimate the distribution of incomes (or wealth) of *individuals* from aggregate raw data involving *tax units*:
  - the program `gen_pop_indiv.m` that generates a synthetic population of the individuals from tax tabulations regarding the tax units;
  - the program `gen_gperc_indiv.m` that estimates thresholds and average incomes corresponding to generalized percentiles of tax units, individuals, singles, and married people from tax tabulations regarding the tax units;
  - the program `gen_gperc_indiv_allfiles.m` that generates files containing estimates of the thresholds and average incomes at generalized percentiles of tax units, individuals, singles, and married people from tax tabulations regarding tax units provided in all the raw data files of the subfolder `rawdata/indiv_allfiles`;
  - the program `gen_gperc_indiv_bysheet.m` that generates files containing estimates of the thresholds and average incomes at generalized percentiles of tax units, individuals, singles, and married people corresponding to all the raw data tabulations regarding tax units provided in each of the worksheet of the file `rawdata_indiv_bysheet` which lies in the folder `rawdata`;

– the program `gen_gperc_indiv_byyear.m` that generates files containing estimates of the thresholds and average incomes at generalized percentiles of tax units, individuals, singles, and married people corresponding to all consecutive raw data subtables regarding tax units provided in the file `rawdata_indiv_byyear` which lies in the folder `rawdata`.

All these programs make the assumption of an equal sharing of income or wealth between spouses.

These codes import tax tabulations data from an Excel files that lie in the subfolder `rawdata`. Codes `gen_pop.m`, `gen_gperc.m`, `gen_gperc_allfiles.m`, `gen_gperc_byyear.m`, `gen_indiv_pop.m`, `gen_indiv_gperc.m`, `gen_indiv_gperc_allfiles.m`, `gen_indiv_gperc_byyear.m` only read the "RawData" sheet of Excel files and ignore other sheets. In constrast codes `gen_gperc_bysheet.m` and `gen_gperc_indiv_bysheet.m` expect to find consistent data (in RawData format) in all sheets. In case of sheets with inconsistent data the code will move on to the next sheet.

In order to generate well-behaved Pareto curves, we advise that the input data satisfy the conditions for the Pareto curve to be monotone between thresholds (see appendix A).

The outputs are `.csv` files generated in the subfolder `output`.

All the above programs use the following codes that are located in the subfolder `codes`:

- the codes `inputs_check.m` or `inputs_check_indiv.m` that check that there is no inconsistency in raw data inputs;
- the routine `pc.m` that interpolates the Pareto curve using tax tabulation data;
- the routine `neg_part.m` that estimates the part of the population with negative incomes or wealth;
- the routine `quantile.m` that approximates the quantile function using the Pareto curve estimated with `pc.m`;
- the subroutines `bracket.m`, `ranks.m`, `qinterp1.m`, `shift_av.m`, `shift_b.m`, `shift_p.m` and `shift_thr.m`.

The codes described below have been written to run on the software Matlab. They are based on extended and modified version of the results proven in Fournier [2015], available online here.

In the first section, we describe how to use all the programs listed above to generate a synthetic micro-files and to estimate g-percentiles. The second section details the functioning of each of the routines that are successively used by the programs.

Contents

1 How to use the programs

We explain here how to use the various programs, which form the inputs have to take, and where outputs are generated.

1.1 Estimation of the distribution of incomes (or wealth) from aggregate raw data

1.1.1 Program `gen_pop.m`

Taking as an input tabulated income data, the program `gen_pop.m` simulates a synthetic micro-file of a given number N of incomes (or wealth). This micro-file approximates the income distribution (or wealth distribution) of the population.

To generate a file of say 1 million of individuals, the program will run for less than 60 seconds.

`gen_pop.m` imports tax tabulation data from an Excel spreadsheet, calls the Matlab functions `quantile.m` and `neg_part.m` to approximate the quantile function, and generates the synthetic micro-file by applying the inversion principle.

Input An Excel spreadsheet workbook named `rawdata.xlsx` has to be placed in a subfolder, entitled "rawdata", of the folder containing the Matlab program. The tabulated income data needs to have the following form.

In the worksheet named "RawData", the average income (or wealth) of the population must be provided in the second box of the second line. Starting from the third line, a 4-column table has to give:

- the thresholds in a first column "thr";
- the corresponding percentiles in a second column "p";
- and the corresponding Pareto coefficients b in a third column "b", or alternatively the corresponding bracket means in the fourth column "yav".

The first line only contains information about the data. It won't be read by the program.

year	2006		
average	21930		
thr	p	b	yav
9400	0,268757	3,01579	
11250	0,335887	2,68161	
13150	0,407262	2,45805	
15000	0,482853	2,33289	
16900	0,548446	2,2345	
18750	0,602345	2,15837	
23750	0,702708	1,97903	
28750	0,780623	1,8927	
38750	0,877692	1,83887	
48750	0,927604	1,85842	
97500	0,98616	2,08967	

Table 1: Input of the `gen_pop.m` - Worksheet "RawData"

Please notice that you can provide either Pareto coefficient or average value for each bracket, but:

- at least one of these two values has to be provided for each bracket;
- if for some brackets the two columns are filled, the values have to be consistent.

Otherwise, the program will return an error message and won't run.

Also, please note that if the name of the fourth column is not "yav", this column will be ignored.

Similarly, if thresholds or percentiles are not in ascending order, or if bracket means that are given or implicitly defined by Pareto coefficients do not lie within the bounds of the brackets, the program will display an error message.

Including a Dirac at the bottom of the distribution In case you want to include a Dirac (positive mass) at the bottom of the distribution, then include the corresponding threshold twice. E.g. assume you want to simulate a distribution including 10% of the population with income equal to 200 (and 5% with income between 200 and 1000 and b=3 at thr=1000, etc.). Then you should enter data as in table 6. In the special case when there is a positive share of

thr	p	b	yav
200	0		
200	0,1		
1000	0,15	3	

Table 2: Including a Dirac at the bottom of the distribution

the population with zero income and no one has negative income, the Dirac can be included only once (see table 3).

thr	p	b	yav
0	0,1		
1000	0,15	3	

Table 3: Including 0 as a Dirac

In case the lower threshold is nonnegative, then the computer code will not generate any negative observation. But if you include a negative threshold thr0 < 0, then the computer code will automatically generate an exponential distribution for the negative part (the exponential distribution will be truncated at thr0 if the corresponding percentile is 0).

Changing the number of taxpayers to be simulated At the beginning of the Matlab code `gen_pop.m`, a line entitled "Number of taxpayers to be generated" gives the number `Npop` of taxpayers to be simulated. By default, this number is equal to 1000000.

Output The synthetic micro-file is a file named `sim_pop.csv` that is created in a subfolder `output`. Each line corresponds to a taxpayer. To import it in Stata, click `File`, `Import`, `Text data (delimited, *.csv, ...)`.

1.1.2 Program `gen_gperc.m`

Taking as an input tabulated income data, the program `gen_gperc.m` estimates thresholds and average incomes corresponding to generalized percentiles.

The program will run for less than 60 seconds.

`gen_gperc.m` imports tax tabulation data from an Excel spreadsheet, calls the Matlab functions `quantile.m` and `neg_part.m` to approximate the quantile function, and estimates the thresholds and average values at g-percentiles by applying the inversion principle.

Input `gen_gperc.m` uses the same Excel spreadsheet workbook as `gen_pop.m`. It is named `rawdata.xlsx` and has to be placed in a subfolder, entitled "rawdata", of the folder containing the Matlab program. The tabulated income data needs to have exactly the same form as input data described in section 1.1.1.

Output The table lies in a file named `gperc.csv` that is created in a subfolder `output`. The column "thr" gives the thresholds corresponding to g-percentiles "p", the column "ytop" gives the averages above the g-percentiles, and the column "b" contains the corresponding (inverted) Pareto coefficients. To convert it to `.xls`, click `Data` and `Convert...`.

1.1.3 Program `gen_gperc_allfiles.m`

The program `gen_gperc_allfiles.m` generates files containing estimates of the thresholds and average incomes at generalized percentiles corresponding to all the raw data files provided in the subfolder `rawdata/allfiles`.

Inputs Raw data files should be placed in Excel spreadsheet workbooks all in the subfolder `allfiles` of the folder `rawdata`. In each file, the data should take the form described in section 1.1.1.

Outputs The name of the output file corresponding to an input file *filename*`.xls` will be `gperc_`*filename*`.csv`. It will be in the subfolder `allfiles` of the folder `output`.

1.1.4 Program `gen_gperc_bysheet.m`

The program `gen_gperc_bysheet.m` generates files containing estimates of the thresholds and average incomes at generalized percentiles corresponding to all the raw data tabulations provided in each of the worksheet of the file `rawdata_bysheet` which lies in the folder `rawdata`.

Inputs Raw data files should be placed in the worksheets of the file `rawdata_bysheet` which is in the folder `rawdata`. In each worksheet, the data should take the form described in section 1.1.1.

Outputs The name of the output file corresponding to the worksheet *sheetname* will be `gperc_`*sheetname*`.csv`. It will be in the subfolder `bysheet` of the folder `output`.

1.1.5 Program `gen_gperc_byyear.m`

The program `gen_gperc_byyear.m` generates files containing estimates of the thresholds and average incomes at generalized percentiles corresponding to all consecutive raw data subtables provided in the file `rawdata_byyear` which lies in the folder `rawdata`.

year	1959		year	1960	
average	1084303,522		average	11024,82199	
thr	p	b	thr	p	b
172404,26	0,5	12,07861635	1841,145273	0,5	11,4760479
1000000	0,813383844	4,762353207	10000	0,816845393	4,944672853
2000000	0,89939459	3,785989569	20000	0,893940543	3,72510883
5000000	0,963094	3,011796277	50000	0,95999992	2,881600613
10000000	0,985506482	2,72794417	100000	0,984347262	2,576777604
20000000	0,994715071	2,495035885	200000	0,994492328	2,366450297
50000000	0,998826274	2,385314183	500000	0,998684369	2,049447205
100000000	0,99961227	2,176467391	1000000	0,999591419	1,7789

Table 4: Input of the program `gen_gperc_byyear.m`

Inputs Raw data tables should be placed in the worksheet `RawData` of the file `rawdata_byyear` which is in the folder `rawdata`. The subtables should be positioned side by side in the `RawData` worksheet, as in the figure 4 below. This template doesn't allow to insert an "yav" column. Therefore, the user should provide all the Pareto coefficients.

Apart from that, the data in each subtable should take the form described in section 1.1.1.

Outputs The name of the output file corresponding to the subtable *year* will be `gperc_year.csv`. It will be in the subfolder `byyear` of the folder `output`.

1.2 Individualization programs

The programs `gen_pop_indiv.m`, `gen_gperc_indiv.m`, `gen_gperc_indiv_allfiles.m`, `gen_gperc_indiv_bysh` and `gen_gperc_indiv_byyear.m` estimate the distribution of incomes (or wealth) of *individuals* from aggregate raw data involving *tax units*.

All these programs make the assumption of an equal sharing of income or wealth between spouses.

1.2.1 Program `gen_pop_indiv.m`

Taking as an input tabulated income data about taw units, the program `gen_pop_indiv.m` simulates a synthetic micro-file of a given number N of incomes (or wealth). This micro-file approximates the income distribution (or wealth distribution) of the individuals in the population.

To generate a file of say 1 million of individuals, the program will run for about than 2 or 3 minutes.

`gen_pop_indiv.m` imports tax tabulation data from an Excel spreadsheet, calls the Matlab function `quantile.m` to approximate the quantile function of the singles and of the couples, and generates the synthetic micro-file by applying the inversion principle.

Input An Excel spreadsheet workbook named `rawdata_indiv.xlsx` has to be placed in a subfolder, entitled "rawdata", of the folder containing the Matlab program. The tabulated income data needs to have the following form.

In the worksheet named "RawData", the average income (or wealth) of the population must be provided in the second box of the second line. Starting from the third line, a 5-column table (see table 5) has to give:

- the thresholds in a first column "thr";
- the corresponding percentiles in a second column "p";
- the corresponding Pareto coefficients b in a third column "b";
- the corresponding share of singles among tax units "s";
- optionally, the corresponding ratio of average incomes of the singles to average incomes of the couples "ys" for each bracket.

The first line only contains information about the data. It won't be read by the program. Please note that if the name of the fifth column is not "ys", this column will be ignored. In case there is no data in the "ys" column or if this column is missing, the ratios are assumed to be equal to 1. That is, it will be assumed by default that singles have the same mean income as couples within each bracket.

If thresholds or percentiles are not in ascending order, or if bracket means that are given or implicitly defined by Pareto coefficients do not lie within the bounds of the brackets, or if the shares of singles are not between 0 and 1, the program will display an error message. The program does not allow for negative incomes or wealth.

Notice that in this program, the lowest bracket between 0 and the first tax threshold should *always* be included in order to provide the share of singles in this bracket.

year	2006			
average	21 930			
thr	p	b	s	ys
0	0,00000000		0,85463993	
9 400	0,26875663	3,01579213	0,90324554	
11 250	0,33588719	2,68160653	0,80107526	
13 150	0,40726215	2,45804620	0,80107526	
15 000	0,48285258	2,33288574	0,70159174	
16 900	0,54844570	2,23450232	0,70159174	
18 750	0,60234475	2,15836740	0,50765239	
23 750	0,70270777	1,97902656	0,50765239	
28 750	0,78062350	1,89269888	0,26236481	
38 750	0,87769240	1,83887136	0,15389074	
48 750	0,92760420	1,85841691	0,15389074	
97 500	0,98616028	2,08967328	0,15389074	

Table 5: Input of `gen_pop_indiv.m` - Worksheet "RawData"

Including a Dirac at the bottom of the distribution In case you want to include a Dirac (positive mass) at the bottom of the distribution, then include the corresponding threshold twice. E.g. assume you want to simulate a distribution including 10% of the population with income equal to 0 (and 5% with income between 0 and 1000 and b=3 at thr=1000, etc.). Then you should enter data as in table 6.

thr	p	b	s	ys
0	0		0,85	
0	0,1		0,9	
1000	0,15	3	0,8	

Table 6: Including a Dirac at the bottom of the distribution

Changing the number of taxpayers to be simulated At the beginning of the Matlab code `gen_pop.m`, a line entitled "Number of taxpayers to be generated" gives the number `Npop` of taxpayers to be simulated. By default, this number is equal to 1000000.

Output The synthetic micro-file is a file named `sim_pop_indiv.csv` that is created in a subfolder `output`. Each line corresponds to a taxpayer. To import it in Stata, click `File`, `Import`, `Text data (delimited, *.csv, ...)`.

1.2.2 Program `gen_gperc_indiv.m`

Taking as an input tabulated income data, the program `gen_gperc.m` estimates thresholds and average incomes corresponding to generalized percentiles of tax units, individuals, singles, and married people from tax tabulations regarding the tax units.

The program will run for about 2 or 3 minutes.

`gen_gperc_indiv.m` imports tax tabulation data from an Excel spreadsheet, calls the Matlab function `quantile.m` to approximate the quantile function, and estimates the thresholds and average values at g-percentiles by applying the inversion principle.

Input `gen_gperc_indiv.m` uses the same Excel spreadsheet workbook as `gen_pop_indiv.m`. It is named `rawdata.xlsx` and has to be placed in a subfolder, entitled "rawdata", of the folder containing the Matlab program. The tabulated income data needs to have exactly the same form as input data described in section 1.2.1.

Output The table lies in a file named `gperc_indiv.csv` that is created in a subfolder `output`. The column "thr_unit" gives the thresholds corresponding to g-percentiles "p" for tax units, the column "ytop_unit" gives the averages above the g-percentiles for tax units, and the column "b_unit" contains the corresponding (inverted) Pareto coefficients for tax units. Columns with the subscript "indiv" give the same information for individuals, columns with the subscript "singles" give this information for singles, and columns with the subscript "couples" give this information for individuals that are not single.

To convert the file to `.xls`, click `Data` and `Convert...`.

1.2.3 Program `gen_gperc_indiv_allfiles.m`

The program `gen_gperc_indiv_allfiles.m` generates files containing estimates of the thresholds and average incomes at generalized percentiles of tax units, individuals, singles, and married people from tax tabulations regarding tax units provided in all the raw data files of the subfolder `rawdata/indiv_allfiles`.

Inputs Raw data files should be placed in Excel spreadsheet workbooks all in the subfolder `indiv_allfiles` of the folder `rawdata`. In each file, the data should take the form described in section 1.2.1.

Outputs The name of the output file corresponding to an input file *filename*`.xls` will be `gperc_indiv_`*filename*`.csv`. It will be in the subfolder `indiv_allfiles` of the folder `output`.

1.2.4 Program `gen_gperc_indiv_bysheet.m`

The program `gen_gperc_indiv_bysheet.m` generates files containing estimates of the thresholds and average incomes at generalized percentiles of tax units, individuals, singles, and married people corresponding to all the raw data tabulations regarding tax units provided in each of the worksheet of the file `rawdata_indiv_bysheet` which lies in the folder `rawdata`.

Inputs Raw data files should be placed in the worksheets of the file `rawdata_indiv_bysheet` which is in the folder `rawdata`. In each worksheet, the data should take the form described in section 1.2.1.

Outputs The name of the output file corresponding to the worksheet *sheetname* will be `gperc_indiv_`*sheetname*`.csv`. It will be in the subfolder `indiv_bysheet` of the folder `output`.

1.2.5 Program `gen_gperc_indiv_byyear.m`

The program `gen_gperc_indiv_byyear.m` generates files containing estimates of the thresholds and average incomes at generalized percentiles of tax units, individuals, singles, and married people corresponding to all consecutive raw data subtables regarding tax units provided in the file `rawdata_indiv_byyear` which lies in the folder `rawdata`.

year	1985				year	2000			
average	10 326				average	14 512			
thr	p	b	s	ys	thr	p	b	s	ys
0	0,00000000		0,56858222		0	0,00000000		0,82523079	
6 099	0,37550981	2,39399932	0,43357489		6 099	0,27125649	3,08481803	0,85095334	
7 624	0,49006224	2,14311800	0,35846334		7 624	0,35338493	2,66702403	0,78487458	
9 148	0,58791867	1,99321778	0,28797580		9 148	0,43162635	2,40246250	0,69904201	
10 673	0,66060266	1,87645623	0,20737455		10 673	0,51125286	2,24379601	0,63292594	
12 197	0,71692600	1,78175266	0,16081297		12 197	0,57827064	2,12668010	0,57103733	
13 722	0,76850694	1,72660618	0,12703254		13 722	0,63499450	2,03764488	0,55401073	
15 246	0,81115891	1,69072122	0,09568303		15 246	0,67693664	1,94917381	0,40560633	
19 057	0,88527668	1,65099691	0,07218490		19 057	0,76833821	1,82105601	0,29408263	
22 868	0,92754248	1,64827413	0,06357285		22 868	0,83358028	1,75518697	0,19660140	
30 491	0,96594849	1,66834503	0,06338745		30 491	0,90995009	1,70291795	0,14145051	
38 113	0,98067898	1,67643251	0,06271843		38 113	0,94788444	1,70716682	0,11686917	
76 224	0,99657898	1,68530289	0,07181455		76 224	0,99106573	1,82826587	0,12387915	

Table 7: Input of the program `gen_gperc_indiv_byyear.m`

Inputs Raw data tables should be placed in the worksheet `RawData` of the file `rawdata_indiv_byyear` which is in the folder `rawdata`. The subtables should be positioned side by side in the `RawData` worksheet, as in the figure 7 below. The data in each subtable should take the form described in section 1.2.1.

Outputs The name of the output file corresponding to the subtable *year* will be `gperc_indiv_`*year*`.csv`. It will be in the subfolder `indiv_byyear` of the folder `output`.

2 Matlab codes

The codes to generate micro-files of incomes distributed as the taxpaying population and generalized percentiles breaks down into five programs:

- the routine `pc.m` that interpolates the Pareto curve using tax tabulation data;
- the routine `quantile.m` that approximates the quantile function using the Pareto curve estimated with `pc.m`;
- the routine `neg_part.m` that estimates the part of the population with negative incomes or wealth;
- the main part of the code (`gen_pop.m`, `gen_gperc.m`...) that uses the approximation of the quantile function provided by `quantile.m` and the negative part estimated by `neg_part.m` to generate the population or to estimate g-percentiles.

2.1 Main programs

2.1.1 Program `gen_pop.m`

Program `gen_pop.m` is based on the inversion principle. It generates a vector u of percentile ranks in the population and returns the estimates of the quantile function at these points.

Steps of the program

1. After clearing the workspace, the path to access tabulated data is indicated to the program. The user has also to specify `Npop`, the size of the simulated population.

2. Then, the program loads the tabulated data from the worksheet `RawData` of the file `rawdata.xlsx`. The shortcut `loadpath` gives the emplacement of this spreadsheet. The code cuts the first line of the data tabulation (which contains descriptive information about the data) and removes useless data in the forth column if the column name is not `yav`.

   The program calls the function `inputs_check.m`, which is located in the subfolder `codes`, to validate the inputs. This function returns the thresholds of the tax tabulation in the vector `raw.thr`, the corresponding percentiles in the vector `raw.p`, the Pareto coefficients in the vector `raw.b`, the average income (wealth) in `y_av`, information about the potential Dirac is temporarily stored in `dirac`, and information about the negative part in `neg`.

   If a Dirac does appear in the population, its location is stored in `dirac`, and the percentile of the population at this point is given in `p0`. Otherwise, `dirac` is set equal to 0 and `p0` is `NaN`.

   Similarly, if there are negative values in raw data, the negative threshold is `thr_neg`, the percentile of the population with negative income (wealth) is `p_neg` and their average income (wealth) is `av_neg`. Otherwise, these variables are all set equal to 0.

3. A vector `u` of `Npop` numbers evenly distributed on the interval $[0, 1]$ is defined. Its elements correspond to the percentile ranks in the population.

   Alternatively, a vector `u` of `Npop` numbers uniformly distributed on the interval $[0, 1]$ can be generated. The incomes stored in `u` are sorted in an increasing order.

4. The main program calls the routine `quantile.m` in the emplacement `codespath` to recover the quantile function. To do so, the main program provides the vector `u`, as well as tabulated thresholds, percentiles, Pareto coefficients, average income and Dirac to the quantile function, possibly shifting them using `shift_thr.m`, `shift_p.m`, `shift_b.m` and `shift_av.m` if there are negative observations. The function `quantile.m` interpolates the quantile function at the points of vector `u` corresponding to nonnegative observations. It returns the vector `sim_pop_pos`.

   If there are negative observations, the main program calls the function `neg_part.m` to simulate them and stores the results in the vector `sim_pop_neg`.

   The two vectors `sim_pop_pos` and `sim_pop_neg` are then merged into `sim_pop` which is a vector of `Npop` random numbers which are distributed as the incomes of the population.

5. The last step is the writing of the output file `sim_pop.csv` in the subfolder `output`.

2.1.2 Program `gen_gperc.m`

The steps of the program `gen_gperc.m` are similar to those of program `gen_pop.m` that are described in section 2.1.1. The vector `u` of percentile ranks is replaced with the vector `gperc` of generalized percentiles. In step 4, the vector `B_pos` containing the values of the Pareto curve is the second output of routine `quantile`. It is adjusted in case there is a Dirac. In step 5, the average income above each generalized percentile is computed in vector `av`. It is defined as the product of the thresholds and the Pareto coefficients in the positive part of the population, and below it is estimated by integrating the thresholds. The vector `B` of Pareto coefficients for the whole population is finally defined.

2.1.3 Program `gen_gperc_allfiles.m`

First, the program loads all the Excel files of the folder defined in `loadpath` (by default, the subfolder `allfiles` of the folder `rawdata`). `inputsNames` contains the names of these files. For

each of the data files, the program `gen_gperc_allfiles.m` follows exactly the same steps as the program `gen_gperc.m`. At each loop, if there is an error at any step, the program go on to the next data file (which is coded with the `try`, `catch` and `continue` commands).

2.1.4 Program `gen_gperc_bysheet.m`

First, the program loads all the sheets of the data file `rawdata_bysheet.xlsx` which lies in the folder `rawdata`. The names of the sheets are stored in `sheets`. For each of these worksheets, the program `gen_gperc_bysheet.m` follows exactly the same steps as the program `gen_gperc.m`. At each loop, if there is an error at any step, the program go on to the next data sheet.

2.1.5 Program `gen_gperc_byyear.m`

First, the program loads the whole table of the worksheet `RawData` of the file `rawdata_byyear.xlsx` which lies in the folder `rawdata`. At each loop, it keeps the `ii`th three-column subtable. It keeps the corresponding year in variable `year`. For each of these subtables, the program `gen_gperc_byyear.m` follows exactly the same steps as the program `gen_gperc.m`.

At each loop, if there is an error at any step, the program go on to the next data subtable.

2.1.6 Program `gen_pop_indiv.m`

Program `gen_pop_indiv.m` is also based on the inversion principle. It generates two vectors, u_s and u_c, of percentile ranks. The first corresponds to singles, the second to couples. It computes the values of the quantile functions estimated with raw data at these points.

It is assumed that the income is equally shared between spouses.

Steps of the program

1. After clearing the workspace, the path to access tabulated data is indicated to the program. The user has also to specify `Npop`, the size of the simulated population.

2. Then, the program loads the tabulated data from the worksheet `RawData` of the file `rawdata.xlsx`. The shortcut `loadpath` gives the emplacement of this spreadsheet. The code cuts the first line of the data tabulation (which contains descriptive information about the data) and removes useless data in the forth column if the column name is not `ys`.

   The program calls the function `inputs_check_indiv.m`, which is located in the subfolder `codes`, to validate the inputs. This function returns the thresholds of the tax tabulation in the vector `raw.thr`, the corresponding percentiles for singles (resp. couples) in the vector `raw.p_s` (resp. `raw.p_c`), the Pareto coefficients for singles (resp. couples) in the vector `raw.b_s` (resp. `raw.b_c`), the average income (wealth) of singles (resp. couples) in `y_av_s` (resp. `y_av_s`), and information about the potential Dirac for singles is temporarily stored in `dirac_s` (resp. `dirac_c`). `share_s` denotes the share of singles among the households.

   If a Dirac does appear in the population of singles (resp. couples), its location is stored in `dirac_s` (resp. `dirac_c`), and the percentile of the population at this point is given in `p0_s` (resp. `p0_c`). Otherwise, `dirac_s` (resp. `dirac_c`) is set equal to 0 and `p0_s` (resp. `p0_c`) is `NaN`.

3. First, the program computes the number `N_c` of couples and the number `N_s` of singles to be simulated to have a total of`Npop` individuals (and to respect the share `share_s` of singles among households).

Then, a vector `u_s` of `N_s` numbers (resp. a vector `u_c` of `N_c` numbers) evenly distributed on the interval $[0, 1]$ is defined. Its elements correspond to the percentile ranks in the population of singles (resp. couples).

Alternatively, vectors `u_s` and `u_c` can be generated from a uniformly distribution on the interval $[0, 1]$. The incomes stored in `u_s` and `u_c` are sorted in an increasing order.

4. The main program calls the routine `quantile.m` in the emplacement `codespath` to recover the quantile function. To do so, the main program provides the vector `u_s` (resp. `u_c`), as well as tabulated thresholds, percentiles, Pareto coefficients, average income and Dirac corresponding to singles (resp. couples) to the quantile function, possibly shifting them using `shift_thr.m`, `shift_p.m`, `shift_b.m` and `shift_av.m` if there is a Dirac. It returns the vector `sim_pop_indiv` of `Npop` individuals where the population of singles is merged with the population of individuals living in couple. It is assumed that the income is equally shared between spouses.

5. The last step is the writing of the output file `sim_pop_indiv.csv` in the subfolder `output`.

2.1.7 Program `gen_gperc_indiv.m`

The steps of the program `gen_gperc_indiv.m` are similar to those of program `gen_pop_indiv.m` that are described in section 2.1.6.

The program first estimate the quantile functions of singles and of couples at a large number of points (by default, `N` is equal to 50000000) displayed on vectors `u_s` and `u_c` respectively. It stores the estimates of the quantile functions in vectors `thr0_s` and `thr0_c` respectively, and estimates of the Pareto coefficients in vectors `B0_s` and `B0_c`.

Then, it evaluates successively the values of the thresholds, Pareto coefficients and top averages at g-percentiles for singles, couples, households (after merging vectors `thr0_s` and `thr0_c`) and individuals (assuming an equal sharing of the income between spouses). As in program `gen_gperc.m`, the top average incomes are computed as the product of the thresholds and the Pareto coefficients above p_0, and by integrating thresholds below p_0. The Pareto curves corresponding to households and to individuals are also computed by integrating respectively the vectors of thresholds `thr0_unit` and `thr0_indiv`, using the trapezoidal rule.

2.1.8 Program `gen_gperc_indiv_allfiles.m`

First, the program loads all the Excel files of the folder defined in `loadpath` (by default, the subfolder `indiv_allfiles` of the folder `rawdata`). `inputsNames` contains the names of these files. For each of the data files, the program `gen_gperc_indiv_allfiles.m` follows exactly the same steps as the program `gen_gperc_indiv.m`. At each loop, if there is an error at any step, the program go on to the next data file (which is coded with the `try`, `catch` and `continue` commands).

2.1.9 Program `gen_gperc_indiv_bysheet.m`

First, the program loads all the sheets of the data file `rawdata_indiv_bysheet.xlsx` which lies in the folder `rawdata`. The names of the sheets are stored in `sheets`. For each of these worksheets, the program `gen_gperc_indiv_bysheet.m` follows exactly the same steps as the program `gen_gperc_indiv.m`. At each loop, if there is an error at any step, the program go on to the next data sheet.

2.1.10 Program `gen_gperc_indiv_byyear.m`

First, the program loads the whole table of the worksheet `RawData` of the file `rawdata_indiv_byyear.xlsx` which lies in the folder `rawdata`. At each loop, it keeps the `iith` three-column subtable. It keeps the corresponding year in variable `year`. For each of these subtables, the program `gen_gperc_indiv_byyear.m` follows exactly the same steps as the program `gen_gperc_indiv.m`.

At each loop, if there is an error at any step, the program go on to the next data subtable.

2.2 Routine `quantile.m`

The function `quantile.m` approximates the quantile function of a distribution corresponding to tax tabulation data received as an input. To do so, it calls the subroutine `pc.m` that provides an estimation of the Pareto curve.

The method used is based on the formula:

$$Q(p) = \begin{cases} 0 \quad \text{if } 0 \leqslant p \leqslant p_{min}, \\ y^* \frac{(1-p^*)b(p^*)}{(1-p)b(p)} \exp\left(-\int_{p^*}^{p} \frac{1}{(1-q)b(q)} dq\right) \text{ if } p > p_{min} \end{cases} \tag{1}$$

where $y^* = Q(p^*)$. b denotes the Pareto curve.

Description of the code

Inputs The following inputs have to be specified to the Matlab function `quantile.m`.

- a vector `N` of percentiles in $[0, 1]$where the quantile function has to be estimated;
- a vector `thr` of incomes corresponding to the thresholds in the tax tabulation;
- a vector `pp` of percentiles (in $[0, 1]$) corresponding to the thresholds in the tax tabulation;
- a vector `bb` of (inverted) Pareto coefficients corresponding to the thresholds in the tax tabulation;
- the average income of the taxpaying population `y_av`.
- possibly the size of a Dirac in 0 `p0`.

Outputs

- A vector `Q` of approximative values taken by the quantile function Q at the points of `N`, that is, `Q`=Q(`N`);
- a vector `B` of values of the Pareto curve b at the points of table `N`, that is, `B`=b(`N`).

Steps of the program

1. Definition of the Matlab function `quantile`. `T` is the number of thresholds in the tax tabulation. `Xn` is the number of points in the input vector `N`.

   `Ym` is the number of points where the integrand of the integral appearing in the expression (1) of Q will be calculated. It determines the precision of the results.

   `a` and `b` are the two extremities of the interval where the integrand will be calculated.

   `nodes` is a mesh of `Ym`+1 points of the interval $[a, b]$, where $a = \min(1/Ym, \min(N)/2, \min(pp)/2)$ and $b = \max(1 - 1/Ym, 1 - (1 - \max(N))/2, 1 - (1 - \max(pp))/2)$.

2. `quantile.m` calls the function `pc.m` to obtain the approximation of the Pareto curve at the points of vector `nodes`. These values are stored in vector `B0`.

   Then, it interpolates the values of the Pareto curve at the points of `N` using `qinterp1`. These values are stored in vector `B`. If there is a Dirac, the Pareto curve is equal to `Inf` below `p0`.

3. Numerical integration.

   The vector `integrand` contains the values at the points of `nodes` of the application $p \mapsto \frac{1}{(1-p)b(p)}$ that is integrated in the expression (1) of Q.
   The next step is to numerically integrate the function defined by `integrand` using the trapezoidal rule. The vector `trapz` which contains the approximated values of the trapezoids under the curve `integrand` that are delimited by the points of `nodes`. It is computed with the Matlab function `conv`, using the vector `h`.

   The approximation of the integral is stored in vector `int`. It is calculated as the cumulative sum of the elements of `trapz`, using Matlab function `cumsum`.

4. The vector `temp0` gives, up to scalar multiplication, the values of the Q at the points of vector `nodes`. The code interpolates the values of `temp0` at the points of `N` using the function `qinterp1`. The resulting vector is called `temp`.

   `q0` is the vector of values taken by `temp` at the percentiles of the tax scale (which are stored in `pp`). They are also obtained by interpolating with the Matlab routine `qinterp1`. `Q_ast` is the vector of normalization coefficients that have to be applied at the points of `pp` to vector `temp` to find the thresholds `thr`.

5. The last step is the approximation of the quantile function Q at the points of `N`. The values calculated will be stored in the table `Q`.

   `quantile.m` calls the function `bracket.m` to compute estimate in which bracket lie the different points in vector `N`.

   First, observations lying in bracket 0, that is, those who are below the lower threshold of the tax tabulation, are approximated by normalizing `temp` with factor `Q_ast(1)`. If there is a Dirac, the values of `Q` below `p0` are set to 0.

   Then, observations lying in bracket T, that is, those who are above the highest threshold of the tax tabulation, are approximated by interpolating `Q` and using the normalization coefficient `Q_ast(T)`.

   We define two vectors `Q1` and `Q2` as the vectors of normalization coefficients lying right below and right above the points of `N`. The two vectors `w1` and `w2` are the weights giving the position of the points of `N` in their respective brackets.

   Finally, we compute the vector `Q` at the remaining points of `N`, as the weighted sum of the two approximations of Q passing through the thresholds bounding the bracket where each point lies. To do so, we multiply normalization coefficients by the corresponding weights, and multiply the resulting vector with vector `temp`.

2.3 Subroutine `pc.m`

The Matlab function `pc.m` approximates the Pareto curve of a distribution corresponding to tax tabulation data received as an input.

This program is based on PCHIP interpolation. The estimated Pareto curve b goes through the data points, that is:

$$\forall i, \qquad b(\texttt{pp(i)}) = \texttt{bb(i)}.$$

There is an additional constraint that has to be satisfied by the estimated Pareto curve. Indeed, according to formula (1), we have for all i:

$$\texttt{thr(i+1)} = \texttt{thr(i)} \frac{(1-\texttt{pp(i)})\texttt{bb(i)}}{(1-\texttt{pp(i+1)})\texttt{bb(i+1)}} \exp\left(-\int_{\texttt{pp(i)}}^{\texttt{pp(i+1)}} \frac{1}{(1-q)b(q)} dq\right).$$

Reformulating:

$$\int_{\texttt{pp(i)}}^{\texttt{pp(i+1)}} \frac{1}{(1-q)b(q)} dq = \ln\left(\frac{(1-\texttt{pp(i)})\texttt{thr(i)bb(i)}}{(1-\texttt{pp(i+1)})\texttt{thr(i+1)bb(i+1)}}\right) = \beta_i.$$

Also, we have a similar constraint for the lower part of the distribution:

$$\int_{\texttt{p0}}^{\texttt{pp(1)}} \frac{1}{(1-q)b(q)} dq = \ln\left(\frac{\texttt{y\_av}}{(1-\texttt{pp(1)})\texttt{thr(1)bb(1)}}\right) = \beta_0.$$

To comply with these additional constraints, mobile points are defined within the consecutive intervals $[\texttt{pp(i)}, \texttt{pp(i+1)}]$. The idea is to adjust these points along an axis orthogonal to the line passing through `bb(i)` and `bb(i+1)` until the integral $\int_{\texttt{pp(i)}}^{\texttt{pp(i+1)}} \frac{1}{(1-q)b(q)} dq$ is equal to β_i.

Description of the code

Inputs The following inputs have to be specified to the Matlab function `pc.m`.

- a vector `PP` of points in $[0, 1]$ where the Pareto curve has to be estimated;
- a vector `thr` of thresholds of the tax tabulation;
- a vector `pp` of percentiles (in $[0, 1]$) corresponding to the thresholds;
- a vector `bb` of (inverted) Pareto coefficients corresponding to the thresholds;
- the average income (wealth) `y_av`;
- possibly a Dirac in 0 denoted `p0`.

Outputs

- A vector `B` of approximative values taken by the Pareto curve b at the points of `PP`, that is, `B`$=b($`PP`$)$.

Steps of the program

1. Definition of function `pc`.
   (a) First of all, the output vector `BB` that will contain the Pareto coefficients corresponding to vector `PP` is defined. `T` is the number of tax thresholds.
   Then, the vector `beta` containing the values of the integrals $\beta_i = \int_{p_i}^{p_{i+1}} \frac{1}{(1-q)b(q)} dq = \ln\left(\frac{(1-\texttt{pp(i)})\texttt{thr(i)bb(i)}}{(1-\texttt{pp(i+1)})\texttt{thr(i+1)bb(i+1)}}\right)$ for all the brackets defined by raw data is computed. The equivalent for the lower part `beta0` is also defined by $\beta_0 = \ln\left(\frac{\texttt{y\_av}}{(1-\texttt{pp(1)})\texttt{thr(1)bb(1)}}\right)$.

`thr_min` is the minimal positive income (wealth) in the simulated population.

`eps` determines the precision of the approximations of the β_is by the integrals built on estimated Pareto curve b.

A time limit `timelimit` is set for loops.

(b) Next, the program approximates the upper part of the distribution, that is, the part above the first threshold of tax data.

We define the vectors `Lp` (percentiles) and `Lb` (corresponding Pareto coefficients) of all points that have been estimated so far. At each iteration i of the loop, the percentiles and mobile points within the bracket [`pp(i)`, `pp(i+1)`] will be appended to these vectors.

The code then browses through the raw brackets to set the intermediary points so that the successive integrals $\int_{p_i}^{p_{i+1}} \frac{1}{(1-q)b(q)}dq$ are equals to the β_is.
When it focuses on bracket i, it first defines `p2` = `pp(i)`, `p3` = `pp(i+1)`, `b2` = `bb(i)` and `b3` = `bb(i+1)`. `p1`, `b1`, `p4` and `b4` are the percentiles and Pareto coefficients right below and right above the bracket of interest respectively. They are used to summarize the local shape of the Pareto curve.

The program calls function `f_beta` to estimate the Pareto curve on bracket i. It uses its output to fill in vectors `Lp` and `Lb`.

(c) Above the last threshold of raw data, the estimation extends linearly the Pareto curve below `pp(T)`.

(d) Finally, the program estimates the Pareto curve below the first tax threshold.

`p1` is set equal to 0, except if there is a Dirac when it is equal to `p0`. `b1` is set so that the minimal positive income in the population is `thr_min`. `p2` and `b2` are equal to `pp(1)` and `bb(1)`, corresponding to the first threshold of the tax tabulation.

There are two mobile points for the lower part of the curve, in percentiles `q1` and `q2`. These two mobile points will be adjusted vertically to fit β_0.

`b_low(p,bq1,bq2)` gives the value of the Pareto curve at percentile p conditional on the Pareto curve being equal to `bq1` and `bq2` in percentiles `q1` and `q2` respectively. `int_low(bq1,bq2)` gives $\int_{\texttt{p1}}^{\texttt{p2}} \frac{1}{(1-q)b(q)}dq$ conditional on the Pareto curve being equal to `bq1` and `bq2` in percentiles `q1` and `q2` respectively.

`bq1_init` and `bq2_init` are initial guesses for `bq1` and `bq2`. They are the values at `q1` and `q2` of the extension of the line passing through the two first points in the raw data. `temp_low` is the corresponding initial value of the integral of interest.

- If `temp_low`>`beta_0`, the initial guess of the integral is too high. This means that `bq1_init` and `bq2_init` are too low.
  We set `bq1_l` and `bq2_l` equal to `bq1_init` and `bq2_init`. We define `bq1_h` and `bq2_h` high enough to have `int_low(bq1_h,bq2_h)`>`beta0`.
  In the "while" loop, we adjust `bq1_l`, `bq1_h`, `bq2_l` and `bq2_h` as long as the integral `temp_low` is not close enough to `beta0`. This loop ends up if it longs more than `timelimit` seconds.
  To do so, we define `bq1_temp` and `bq2_temp`, the midpoints of segments [`bq1_l`, `bq1_h`] and [`bq2_l`, `bq2_h`] respectively. If `int_low(bq1_temp,bq2_temp)`>`beta0`, then we higher the lower bounds `bq1_l` and `bq2_l` to temp values `bq1_temp` and `bq2_temp` and focus next on the intervals [`bq1_temp`, `bq1_h`] and [`bq2_temp`, `bq2_h`]. Otherwise, we lower the upper bounds `bq1_h` and `bq2_h` to temp values `bq1_temp` and `bq2_temp` and focus next on the intervals [`bq1_l`, `bq1_temp`] and [`bq2_l`, `bq2_temp`].
  We iterate with this time `temp_low = int_low(bq1_temp,bq2_temp)`.

When the integral is close enough to `beta0`, we estimate the values of `BB` below `pp(1)` with `b_low`.

- Otherwise, the initial guess of the integral is too low. This means that `bq1_init` and `bq2_init` are too high. In this case, we will try to adjust only one mobile point `q1`.
  If it is not possible while satisfying the condition $1 - b(p) + (1 - p)b'(p) < 0$ throughout the interval (that guarantees that the quantile function is increasing), we define `b_low_eta` so that$1 - b(p) + (1 - p)b'(p) = -\eta$, and adjust parameter `eta`.

2. Function `f_beta` adjusts two mobile points on each interval. These mobile points move along a line orthogonal to the line defined by bracketing data Pareto coefficients, unless they are too close of the extremities of the intervals. In this special case, they start to move vertically.

2.4 Subroutine `neg_part.m`

Routine `neg_part.m` using a q-exponential distribution. First, it computes the values of the parameters in order to fit the desired average and lowest threshold. Then, it interpolates the values of the negative incomes (wealth) using the quantile function.

2.5 Subroutines `inputs_check.m` and `inputs_check_indiv.m`

These two routines check that raw data received as an input satisfies a range of consistency conditions: that the thresholds and the percentiles are in an ascending order, that the bracket means implicitly defined by Pareto coefficients are lying between the bracketing thresholds, that the average income in the population is consistent with the other data...

2.6 Codes `bracket.m` and `ranks.m`

`ranks(thr,N)` returns the vector of ranks that correspond to the first elements of `N` to be higher than the thresholds in `thr`. It browses through the vector `thr` and for each `thr(i)` finds the rank of the first element of `N` to be higher than `thr(i)`. If there is no such element, the rank is by definition `Inf`.

`bracket(thr,N)` returns the vector output that gives the number of the bracket defined by elements of `thr` in which elements of `N` lie. That is, if `thr(i)`$\leqslant$`N(k)`$<$`thr(i+1)`, `output(k)`$=$`i`. If `N(k)`$<$`thr(1)`, then `output(k)`$= 0$. If `N(k)`$>$`thr(T)`, then `output(k)`$=$`T`, where `T` is the length of vector `thr`. `bracket(thr,N)` also returns the vector `rk` given by `rk`$=$`ranks(thr,N)`.

2.7 Code `qinterp1.m`

`qinterp1.m` is a program written by N. Brahms and available here. It performs fast interpolation. `qinterp1(x,Y,xi)` interpolates linearly the values at the points of `xi` of the function implicitly defined by vectors `x` and `Y`. Vector `x` must be monotonically and evenly increasing.

2.8 Codes `shift_av.m`, `shift_b.m`, `shift_p.m` and `shift_thr.m`

- `shift_av(y_av,dirac,p_neg,av_neg)` computes the average income (wealth) of the non-negative part of the population if there is a share `p_neg` of the population with average negative income (wealth) `av_neg`. If there is a Dirac at `dirac`, the resulting average is shifted by `-dirac`.

- `shift_b(thr,bb,dirac)` computes the Pareto coefficients of the population if there is a Dirac at `dirac` by shifting all the thresholds and averages by `-dirac`.
- `shift_p(pp,p_neg)` computes the percentiles ranks of the nonnegative part of the population if there is a share `p_neg` of the population with negative income (wealth).
- `shift_thr(thr,dirac)` shifts the thresholds by `-dirac`.

A Monotony of the Pareto curve between thresholds

Suppose that we know percentiles p_1 and p_2 and corresponding top shares s_1 and s_2:

$$0 < p_1 < p_2 < 1, \quad 0 < s_2 < s_1 < 1 \quad \text{and} \quad \frac{s_1}{1-p_1} < \frac{s_2}{1-p_2}.$$

We are looking for the values of the income thresholds ratio $\frac{q_2}{q_1}$ if we assume b to be monotonous between p_1 and p_2?

Conditions

1. If b is assumed to be decreasing, any values of b_1 and b_2 can be chosen as soon as $b_1, b_2 > 1$ and $b_1 > \beta > b_2$ where :
$$\beta = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\ln\left(\frac{s_1}{s_2}\right)}.$$
The corresponding ratio $\frac{q_2}{q_1}$ can take any value strictly larger than 1.

2. Assume that b is increasing. As before, define $\beta = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\ln\left(\frac{s_1}{s_2}\right)}$. Any values of b_1 and b_2 such that $b_1, b_2 > 1$ that satisfy the four conditions below can be chosen:

   (i) $b_1 < \beta < b_2$;

   (ii) $(1-p_1)(b_1-1) > (1-p_2)(b_2-1)$;

   (iii) $\left(1-\frac{1}{b_1}\right)\ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_1}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right) < \ln\left(\frac{s_1}{s_2}\right)$;

   (iv) $\left(1-\frac{1}{b_2}\right)\ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_2}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right) > \ln\left(\frac{s_1}{s_2}\right)$.

3. Assume that b is constant. Then $b \equiv \beta$ and $\frac{q_2}{q_1} = \left(\frac{1-p_1}{1-p_2}\right)^{1-1/\beta}$.

Remark In particular, inequalities (ii) and (iv) imply that:

$$\frac{s_1}{(1-p_1)b_1} < \frac{s_2}{(1-p_2)b_2},$$

which is equivalent to $q_1 < q_2$ since $s_i = (1-p_i)b_i q_i$, $i = 1, 2$.

Indeed, by (i),

$$\frac{(1-p_1)b_1}{(1-p_2)b_2} > \frac{b_1(b_2-1)}{b_2(b_1-1)} = \frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}.$$

Therefore, by (iv) :

$$\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right) > \left(1-\frac{1}{b_2}\right)\ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_2}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right) > \ln\left(\frac{s_1}{s_2}\right).$$

PROOF:
Assume that b is a solution to our problem. We have:

$$\frac{s_2}{s_1} = \frac{(1-p_2)q_2 b_2}{(1-p_1)q_1 b_1} = \exp\left(-\int_{p_1}^{p_2} \frac{1}{(1-p)b(p)}dp\right).$$

We set:

$$\beta = \frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b(p)} dp} = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\ln\left(\frac{s_1}{s_2}\right)}.$$

β is a weighted average of b on the interval $[p_1, p_2]$. In particular, we have necessarily $\min\{b_1, b_2\} < \beta < \max\{b_1, b_2\}$.

Reciprocally, if b is such that $\frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b(p)} dp} = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\ln\left(\frac{s_1}{s_2}\right)}$ and if we set for all $p \in [0,1]$,

$$Q(p) = \frac{s_1}{(1-p)b(p)} \exp\left(-\int_{p_1}^{p} \frac{1}{(1-r)b(r)} dr\right),$$

we have $(1-p_2)Q(p_2)b(p_2) = s_2$.

We have to find conditions on b_1 and b_2 so that there exists a monotonous Pareto curve monotone between p_1 and p_2 which satisfies this equality.

Lemme

1. *If b_1 and b_2 are such that $b_1 > \beta > b_2$, then there exists a continuous decreasing function $b : [p_1, p_2] \to \mathbb{R}$ such that $b(p_1) = b_1$, $b(p_2) = b_2$ and:*

$$\beta = \frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b(p)} dp}$$

under the additional condition:

$$\forall p \in [p_1, p_2], \quad 1 - b(p) + (1-p)b'(p) < 0.$$

2. *If b_1 and b_2 satisfy the conditions (i), (ii), (iii) and (iv) below,*

   (i) $b_1 < \beta < b_2$

   (ii) $(1-p_1)(b_1 - 1) > (1-p_2)(b_2 - 1)$

   (iii) $\beta > I_{min} = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\left(1-\frac{1}{b_1}\right)\ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_1}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right)}$

   (iv) $\beta < I_{max} = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\left(1-\frac{1}{b_2}\right)\ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_2}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right)}$

   then there exists a continuous increasing function $b : [p_1, p_2] \to \mathbb{R}$ such that $b(p_1) = b_1$, $b(p_2) = b_2$ and:

$$\beta = \frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b(p)} dp}$$

under the additional assumption:

$$\forall p \in [p_1, p_2], \quad 1 - b(p) + (1-p)b'(p) < 0.$$

Proof of the lemma:

1. • Consider the piecewise linear Pareto curve $b^{\varepsilon}_{\min}$ defined by the points:

$$b^{\varepsilon}_{\min}(p_1) = b_1, \quad b^{\varepsilon}_{\min}(p_1 + \varepsilon) = b_2 \quad \text{and} \quad b^{\varepsilon}_{\min}(p_2) = b_2,$$

for a small $\varepsilon > 0$. By monotonous convergence, we have, when ε goes to 0 :

$$I^{\varepsilon}_{\min} = \frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b^{\varepsilon}_{\min}(p)} dp} \longrightarrow b_2.$$

- Similarly, we define the piecewise linear Pareto curve $b^{\varepsilon}_{\max}$ by the points:

$$b^{\varepsilon}_{\max}(p_1) = b_1, \quad b^{\varepsilon}_{\min}(p_2 - \varepsilon) = b_1 \quad \text{and} \quad b^{\varepsilon}_{\min}(p_2) = b_2,$$

for a small $\varepsilon > 0$. From monotone convergence theorem, we have when ε goes to 0:

$$I^{\varepsilon}_{\max} = \frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b^{\varepsilon}_{\max}(p)} dp} \longrightarrow b_1.$$

- Let $\varepsilon > 0$ be such that $I^{\varepsilon}_{\min} < \beta$ and $I^{\varepsilon}_{\max} > \beta$. For all $x \in [0, p_2 - p_1 - \varepsilon]$, we define the piecewise linear Pareto curve b^x by:

$$b^x(p_1) = b_1, \quad b^x(p_1 + x) = b_1, \quad b^x(p_1 + x + \varepsilon) = b_2 \quad \text{et} \quad b^x(p_2) = b_2,$$

and we set for all x :

$$I(x) = \frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b^x(p)} dp}.$$

$I(0) < \beta$, $I(p_2 - p_1 - \varepsilon) > \beta$, and $x \mapsto I(x)$ is continuous from dominated convergence theorem. Therefore, there exists $x_0 \in [0, p_2 - p_1 - \varepsilon]$ such that $I(x_0) = \beta$.
Moreover, b^{x_0} is a Pareto curve. Indeed, as this function is decreasing, it satisfies the inequality:

$$\forall p \in [p_1, p_2], \quad 1 - b^{x_0}(p) + (1-p)b^{x_0\prime}(p) < 0.$$

2. - We are looking for an increasing Pareto curve $b_{\max}$ such that $b_{\max}(p_1) = b_1$ and $b_{\max}(p_2) = b_2$ which maximizes the functional $b \mapsto \frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b(p)} dp}$ while satisfying the inequality:

$$\forall p \in [p_1, p_2], \quad 1 - b_{\max}(p) + (1-p)b'_{\max}(p) < 0.$$

Let $\varepsilon > 0$ be a small positive real. We define $b^{\varepsilon}_{\max}$ by:

$$\forall p \in [p_1, \tilde{p}], \quad b^{\varepsilon}_{\max}(p) = \frac{1}{1-p}((1-p_1)b_1 - (1+\varepsilon)(p - p_1))$$

and

$$\forall p \in [\tilde{p}, p_2], \quad b^{\varepsilon}_{\max}(p) = b_2$$

with $\tilde{p} = 1 - (1-p_1)\frac{b_1 - (1+\varepsilon)}{b_2 - (1+\varepsilon)}$. We can check that:

$$\forall p \in [p_1, \tilde{p}], \quad 1 - b^{\varepsilon}_{\max}(p) + (1-p)b^{\varepsilon\prime}_{\max}(p) = -\varepsilon.$$

Under the condition $(1-p_1)(b_1 - 1) > (1-p_2)(b_2 - 1)$, we do have $\tilde{p} < p_2$ for ε small enough. If this condition wasn't verified, there would be no Pareto curve equal to b_1 at p_1 and b_2 at p_2 (indeed, any Pareto curve equal to b_1 at p_1 is bounded above by $b^0_{\max}$ - see below). We find condition (ii).
This proves that $b^{\varepsilon}_{\max}$ is a Pareto curve.
As before, we have when ε goes to 0 by monotone convergence:

$$I^{\varepsilon}_{\max} = \frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b^{\varepsilon}_{\max}(p)} dp} \longrightarrow I_{\max}$$

where $I_{\max} = I^0_{\max}$.
Let's check that for any continuous and increasing Pareto curve b such that $b(p_1) = b_1$, $b(p_2) = b_2$, we have:

$$\frac{\int_{p_1}^{p_2} \frac{1}{1-p} dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b(p)} dp} < I_{\max}.$$

If $\tilde{p}_0 = 1 - (1-p_1)\frac{b_1-1}{b_2-1}$, we have:

$$\forall p \in [\tilde{p}_0, p_2], \quad b(p) \leqslant b_2 = b^0_{\max}(p)$$

and

$$\forall p \in [p_1, \tilde{p}_0], \quad b(p) < b^0_{\max}(p) = \frac{1}{1-p}((1-p_1)b_1 - (p-p_1)).$$

Indeed, as b is a Pareto curve, $p \mapsto (1-p)b(p)$ is strictly decreasing. So for all $p \in [p_1, p_2]$:

$$(1-p)(b(p)-1) < (1-p_1)(b_1-1)$$

which leads to:

$$b(p) < \frac{1}{1-p}((1-p_1)b_1 - (p-p_1)).$$

Therefore, we have the desired inequality.
Let's compute $I_{\max}$.

$$\begin{aligned}
\int_{p_1}^{p_2} \frac{1}{(1-p)b^\varepsilon_{\max}(p)}dp &= \int_{p_1}^{\tilde{p}_0} \frac{1}{(1-p_1)b_1 - (p-p_1)}dp + \int_{\tilde{p}_0}^{p_2} \frac{1}{(1-p)b_2}dp \\
&= \ln\left(\frac{(1-p_1)b_1}{(1-\tilde{p}_0)b_2}\right) + \frac{1}{b_2}\ln\left(\frac{1-\tilde{p}_0}{1-p_2}\right) \\
&= \ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_2}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\frac{1-\frac{1}{b_1}}{1-\frac{1}{b_2}}\right) \\
&= \left(1-\frac{1}{b_1}\right)\ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_1}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right)
\end{aligned}$$

We find the condition :

$$\beta < I_{\max} = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\left(1-\frac{1}{b_2}\right)\ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_2}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right)}$$

- Similarly, for $\varepsilon > 0$ small enough, we define $b^\varepsilon_{\min}$ by:

$$\forall p \in [p_1, \tilde{p}], \quad b^\varepsilon_{\min}(p) = b_1$$

and

$$\forall p \in [\tilde{p}, p_2], \quad b^\varepsilon_{\min}(p) = \frac{1}{1-p}((1-p_2)b_2 - (1+\varepsilon)(p_2-p))$$

with $\tilde{p} = 1 - (1-p_2)\frac{b_2-(1+\varepsilon)}{b_1-(1+\varepsilon)}$. Sous $(1-p_1)(b_1-1) > (1-p_2)(b_2-1)$, we have $\tilde{p} > p_1$.
Let's check that:

$$\forall p \in [\tilde{p}, p_2], \quad 1 - b^\varepsilon_{\min}(p) + (1-p)b^{\varepsilon\prime}_{\min}(p) = -\varepsilon.$$

$b^\varepsilon_{\min}$ est donc bien une Pareto curve.
We define as before $I^\varepsilon_{\min}$ and $I_{\min}$. We have:

$$I^\varepsilon_{\min} \to I_{\min}$$

when ε goes to 0.
We also check that for any continuous and increasing Pareto curve b such that $b(p_1) = b_1$ and $b(p_2) = b_2$:

$$\frac{\int_{p_1}^{p_2} \frac{1}{1-p}dp}{\int_{p_1}^{p_2} \frac{1}{(1-p)b(p)}dp} > I_{\min}.$$

Finally, we obtain the condition:

$$\beta > I_{\min} = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\left(1-\frac{1}{b_1}\right)\ln\left(\frac{1-\frac{1}{b_2}}{1-\frac{1}{b_1}}\right) + \frac{1}{b_1}\ln\left(\frac{(1-p_1)b_1}{(1-p_2)b_2}\right)}.$$

- Let $\varepsilon > 0$ be such that $I^{\varepsilon}_{\min} < \beta$ and $I^{\varepsilon}_{\max} > \beta$. We define for $x \in [0, p_2 - p_1 - (1-p_2)\frac{b_2-b_1}{b_1-(1+\varepsilon)}]$. We define the Pareto curve b^x by:

$$\forall p \in [p_1, p_1 + x], \quad b^x(p) = b_1,$$

$$\forall p \in [p_1 + x, \frac{b_2 - (p_1 + x) - (1 - (p_1 + x))b_1}{b_2 - 1}], \quad b^x(p) = \frac{1}{1-p}((1-(p_1+x))b_1 - (p-(p_1+x))),$$

$$\forall p \in [\frac{b_2 - (p_1 + x) - (1 - (p_1 + x))b_1}{b_2 - 1}, 1 - (1 - p_2)\frac{b_2 - (1 + \varepsilon)}{b_1 - (1 + \varepsilon)}], \quad b^x(p) = b_2.$$

  As in 1., we define $I(x)$. $I(0) < \beta$, $I\left(1 - (1 - p_2)\frac{b_2-(1+\varepsilon)}{b_1-(1+\varepsilon)}\right) > \beta$ and $x \mapsto I(x)$ is continuous from dominated convergence theorem. So there exists x_0 such that $I(x_0) = \beta$.
  Moreover, b^{x_0} is a Pareto curve.

□

□

References

Fournier, J. (2015). Generalized Pareto curves: theory and application using income and inheritance tabulations for France 1901-2012. Master's thesis, Paris School of Economics.