```
1
2
3   *ALLinOne*
4
5   *set (for the first time) the maximum of variables in Stata as the default value might be too
    low: set maxvar 120000, permanently*
6
7
8   *This do-file adjusts the HFCS data of the countries specified in the first line of code. One can
    simply add or remove countries. Calculating all countries at once will take up to 24 hours.
    Please also make sure that the additional input data, i.e. rich-list observations, financial
    balance sheets and intervals for the optimal financial coverage ratio as well as for the optimal
    lower bound of the pareto tail are made available to the program in the respective excel file
    (Configuration.xsls). The rich-list observations ought to be stored in the sheet named
    "Forbes_HFCS_countries_2017", the financial balance sheets ought to be stored in the sheet "FBS",
    the intervals fo both the optimal coverage ratio as well as for the optimal lower bound (w_min)
    have to be entered in the sheet "intervals_z_wmin". Please also make sure that in
    "Configuration.xsls" there are country_output sheets available for every HFCS country you want to
    analyse: output_Italy output_Germany etc. *
9
10
11  foreach a in Croatia Spain Austria Belgium Germany Estonia Finland France Greece Hungary Ireland
    Italy Lithuania Latvia Netherlands Poland Portugal Slovenia Slovakia {
12
13  ********************************************************************************
14
15  global country = "`a'"
16  di "$country"
17
18
19  *attributing every HFCS country its respective country code (HFCS variable sa0100); ideally this
    is checked if an other/updated HFCS wave is used*
20  scalar Austria = 1
21  scalar Belgium = 2
22  scalar Cyprus = 3
23  scalar Germany = 4
24  scalar Estonia = 5
25  scalar Spain = 6
26  scalar Finland = 7
27  scalar France = 8
28  scalar Greece = 9
29  scalar Croatia = 10
30  scalar Hungary = 11
31  scalar Ireland = 12
32  scalar Italy = 13
33  scalar Lithuania = 14
34  scalar Luxembourg = 15
35  scalar Latvia = 16
36  scalar Malta = 17
37  scalar Netherlands = 18
38  scalar Poland = 19
39  scalar Portugal = 20
40  scalar Slovenia = 21
41  scalar Slovakia = 22
42
43  scalar countrycode = `=scalar(`a')'
44
45  *file path where you intend to store the original ECB merged data-set*
46  global hfcsdata
    "\\int.wsr.at\Nabu\restriktive_Daten\EZB\HFCS\net_wealth_tax_JBNSt\Daten\HFCS_UDB_3_2_STATA"
47
48  *file path where you intend to store STATA counrty-data-sets*
49  global countryfile
    "\\int.wsr.at\Nabu\restriktive_Daten\EZB\HFCS\net_wealth_tax_JBNSt\Daten\Newly_Created_Data"
50
51  *file path where you itend to store EXCEL files*
52  global excel "\\int.wsr.at\Nabu\Themen\net_wealth_tax_JBNSt\excel_files"
53
```

```
54
55  *importing financial balance sheet data*
56  import excel "${excel}\\Configuration", sheet("FBS") firstrow clear
57
58  keep if country == "${country}"
59
60  foreach var of varlist F21 - F100 {
61  sum `var'
62  scalar `var'=r(mean)
63  di `=scalar(`var')'
64  }
65
66  *importing interval in which the optimal adjustment factor "z" for financial wealth is to be found*
67  import excel "${excel}\\Configuration", sheet("interval_z_wmin") firstrow clear
68
69  keep if country == "${country}"
70
71  *lower bound interval for optimal "z"*
72  sum zLB
73  scalar zLB=r(mean)
74
75  *"step", zLB+1*
76  sum zstep
77  scalar zstep= r(mean)
78
79  *upper bound interval for optimal "z"*
80  sum zUB
81  scalar zUB= r(mean)
82
83  *lower bound interval for optimal wmin*
84  sum wminLB
85  scalar wminLB= r(mean)
86
87  *"step", e.g. wminLB+250k or wminLB+500k*
88  sum wminstep
89  scalar wminstep= r(mean)
90
91  *upper bound interval for optimal wmin
92  sum wminUB
93  scalar wminUB= r(mean)
94
95  *use the orignial HFCS data-set*
96  use "$hfcsdata\hfcs.dta", clear
97
98  *keeping only data from the respective country*
99  keep if sa0100==`=scalar(`a')'
100
101 keep hd1710 hd1920 da2100 da1140 da1000 hw0010 im0100 dl1000 hd0200 hd1010 sa0100
102
103 *create the country data-file*
104 save "${countryfile}\\${country}Original.dta", replace
105
106 *calculate coverge ratios according to the concepts of Chakraborty et al.(2019)*
107
108 **************************NAIVE CONCEPT*************************************
109 ****************************************************************************
110
111 *first, get rid of missing values*
112 replace hd1710 = 0 if (hd1710 ==.)
113 replace hd1920 = 0 if (hd1920 ==.)
114 replace da2100 = 0 if (da2100 ==.)
115 replace da1140 = 0 if (da1140 ==.)
116 replace da1000 = 0 if (da1000 ==.)
117 replace dl1000 = 0 if (dl1000 ==.)
118 replace hd0200 = 0 if (hd0200 ==.)
119 replace hd1010 = 0 if (hd1010 ==.)
120
121 *specify financial wealth from the national accounts according to the definitions in Chakraborty
```

```
et al.(2019, p.41) - watch out to select the correct year - the third wave was collected in 2017
- ergo choose the finacial balance sheetes of 2017*
scalar FAnaC_Eurostat= `=scalar(F21)'+`=scalar(F22)'+`=scalar(F29)'+`=scalar(F3)'+
`=scalar(F4assets)'+`=scalar(F5)'+`=scalar(F6)'+`=scalar(F7)'+`=scalar(F8)'
di `=scalar(FAnaC_Eurostat)'

*generate a new financial wealth variable by multiplying with the respective household weight for
all 5 implicates; the "naive" refers to the definition of Chackraborty et al. (2019); FA stands
for FinancialAssets; naC stands for naive concept*
forvalues i=1(1)5{
gen FAnaC`i'=da2100*hw0010 if im0100==`i'
}

*find out the "naive" coverage ratio for al 5 implicates*
foreach var of varlist FAnaC1 - FAnaC5 {
sum `var'
scalar s_`var'=r(sum)
scalar Ratio`var'=s_`var'/(FAnaC_Eurostat/100)
}

*creating newly weighted HFCS financial wealth variables; here: so that it equals 1%, 2%, 3%,...,
100% of the financial wealth in the national balance sheets; we dont aoutomatically set it to
100% of the financial balance sheets because the pareto tail will add to the financial wealth
later on!*

*note: the loop from 0 to 9 is necessary for the decimal places as STATA does not allow decimal
places in names of variables*

forvalues i=1(1)5{
foreach x of numlist `=scalar(zLB)' `=scalar(zstep)' to `=scalar(zUB)' {
foreach y of numlist 0 1 to 9 {
gen wFAnaC`i'`x'`y'=da2100/RatioFAnaC`i'*(`x'.`y') if im0100==`i'
}
}
}

**********************************ADJUSTED CONCEPT*****************************************
*******************************************************************************************

*Financial wealth according to the "adjusted (1) concept" of Chakraborty et al. (2019, p. 43)*
scalar FAadC_Eurostat= `=scalar(FAnaC_Eurostat)'- `=scalar(F21)'-`=scalar(F4assets)'-
`=scalar(F61)'-`=scalar(F63)'-`=scalar(F64)'-`=scalar(F65)'-`=scalar(F66)'-`=scalar(F7)'-
`=scalar(F8)'+`=scalar(F100)'
di `=scalar(FAadC_Eurostat)'

*Liabilities as stated in the national accounts..."F4"; short- and long-term loans*
scalar LIABadC_Eurostat=`=scalar(F4liabilities)'
di `=scalar(F4liabilities)'


*HFCS liabilities*
forvalues i=1(1)5{
gen liabHW`i'=dl1000*hw0010 if im0100==`i'
egen sumliabHW`i'=total(liabHW`i') if im0100==`i'
sum sumliabHW`i'
di r(mean)
scalar totalliab`i'=r(mean)
}

*create coverage rations for liabilites*
forvalues i=1(1)5{
scalar RatioLiab`i'=totalliab`i'/(`=scalar(LIABadC_Eurostat)'/100)
}

*create adjusted liability-variables*
forvalues i=1(1)5{
gen LiabManip`i'=dl1000/RatioLiab`i'*100
}
```

```
179
180   *generate a new HFCS financial wealth variable by multiplying with the respective household
      weight for all 5 implicates; the "adjusted" refers to the definition of Chackraborty et al.
      (2019, p.43)*
181
182   forvalues i=1(1)5{
183   gen FAadC`i'=(da2100-hd1710-hd1920+da1140)*hw0010 if im0100==`i'
184   }
185
186   *find out the "adjusted" coverage ratio for all 5 implicates*
187   foreach var of varlist FAadC1 - FAadC5 {
188   sum `var'
189   scalar s_`var'=r(sum)
190   scalar Ratio`var'=s_`var'/(`=scalar(FAadC_Eurostat)'/100)
191   }
192
193   *creating newly weighted HFCS financial wealth variables; here: so that it equals 1%, 2%, 3%,...,
      100% of the financial wealth in the national balance sheets; we dont aoutomatically set it to
      100% of the financial balance sheets because the pareto tail will add to the financial wealth
      later on!*
194
195   forvalues i=1(1)5{
196   foreach x of numlist `=scalar(zLB)' `=scalar(zstep)' to `=scalar(zUB)' {
197   foreach y of numlist 0 1 to 9 {
198   gen wFAadC`i'`x'`y'=(da2100-hd1710-hd1920+da1140)/RatioFAadC`i'*(`x'.`y') if im0100==`i'
199   }
200   }
201   }
202
203   *export naive  and adjusted coverage ratios to excel table*
204   putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
205
206
207   putexcel D2="RatioFAnaiveConcept"
208   sleep 1000
209   putexcel D3=`=scalar(RatioFAnaC1)'
210   sleep 1000
211   putexcel D4=`=scalar(RatioFAnaC2)'
212   sleep 1000
213   putexcel D5=`=scalar(RatioFAnaC3)'
214   sleep 1000
215   putexcel D6=`=scalar(RatioFAnaC4)'
216   sleep 1000
217   putexcel D7=`=scalar(RatioFAnaC5)'
218   sleep 1000
219
220   putexcel E2="RatioFAadjustedConcept"
221   sleep 1000
222   putexcel E3=`=scalar(RatioFAadC1)'
223   sleep 1000
224   putexcel E4=`=scalar(RatioFAadC2)'
225   sleep 1000
226   putexcel E5=`=scalar(RatioFAadC3)'
227   sleep 1000
228   putexcel E6=`=scalar(RatioFAadC4)'
229   sleep 1000
230   putexcel E7=`=scalar(RatioFAadC5)'
231   sleep 1000
232
233
234   *export HFCS financial wealth (adjusted Concept) to excel table*
235   putexcel F2="HFCS_FAadjustedConcept"
236   sleep 1000
237   putexcel F3=`=scalar(s_FAadC1)'
238   sleep 1000
239   putexcel F4=`=scalar(s_FAadC2)'
240   sleep 1000
241   putexcel F5=`=scalar(s_FAadC3)'
```

```
sleep 1000
putexcel F6=`=scalar(s_FAadC4)'
sleep 1000
putexcel F7=`=scalar(s_FAadC5)'
sleep 1000

*export national balance sheet financial wealth to excel table; naive and adjusted Concept*
putexcel G2="Eurostat_FAnaiveConcept"
sleep 1000
putexcel G3=`=scalar(FAnaC_Eurostat)'
sleep 1000
putexcel H2="Eurostat_FAadjustedConcept"
sleep 1000
putexcel H3=`=scalar(FAadC_Eurostat)'
sleep 1000


*export coverage ratios liabilities*
putexcel I2="Ratio_liabilities"
sleep 1000
putexcel I3=`=scalar(RatioLiab1)'
sleep 1000
putexcel I4=`=scalar(RatioLiab2)'
sleep 1000
putexcel I5=`=scalar(RatioLiab3)'
sleep 1000
putexcel I6=`=scalar(RatioLiab4)'
sleep 1000
putexcel I7=`=scalar(RatioLiab5)'
sleep 1000

*save the country data-file*
save "${countryfile}\\${country}Original.dta", replace

*********import Forbes Data (and the respective exchange rate)***************
**********************************************************************
import excel "${excel}\\Configuration", sheet("exchange") firstrow clear
sum exchange
scalar exchange=r(mean)

import excel "${excel}\\Configuration", sheet("Forbes_HFCS_countries_2017") firstrow clear
keep if country == "${country}"

*simply creating (identical) 5 implicates*
gen var2=2
expand var2, gen (newvar)
replace newvar = 2 if (newvar ==0)
rename newvar im
expand var2 if im==2 , gen (newvar)
replace im = 3 if (newvar ==1)
drop newvar
expand var2 if im==2 , gen (newvar)
replace im = 4 if (newvar ==1)
drop newvar
expand var2 if im==2 , gen (newvar)
replace im = 5 if (newvar ==1)
gen im0100 =im
drop im
drop newvar
drop var2


*convert Dollar into Euro using average excahange rate of 2017, which is the year of observation,
source ECB hompage*
*creating relevant variables*
gen WForbes=dn3001/`=scalar(exchange)'
gen ForbesName=name
gen ForbesID=rank
```

```
gen hw0010=1

*specify country*
gen sa0100=`=scalar(`a')'

*specify the least rich observation of the rich list*
sum WForbes
scalar least_rich_observation=r(min)

*export F4liailities, F100, least-rich observation*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify

putexcel Z2="F4liabilities"
putexcel Z3=`=scalar(F4liabilities)'

putexcel AA2="F100"
putexcel AA3=`=scalar(F100)'

putexcel AB2="least_rich_observation"
putexcel AB3=`=scalar(least_rich_observation)'

*save Forbes as STATA.dta file*
save "${countryfile}\\${country}Forbes.dta", replace

*appending*
append using "${countryfile}\\${country}Original.dta"

* get rid of missing values again*
replace hd1710 = 0 if (hd1710 ==.)
replace hd1920 = 0 if (hd1920 ==.)
replace da2100 = 0 if (da2100 ==.)
replace da1140 = 0 if (da1140 ==.)
replace da1000 = 0 if (da1000 ==.)
replace dl1000 = 0 if (dl1000 ==.)
replace hd0200 = 0 if (hd0200 ==.)
replace hd1010 = 0 if (hd1010 ==.)
replace WForbes = 0 if (WForbes ==.)
replace ForbesID = 0 if (ForbesID ==.)


* From here on only the "Adjusted Concept" is used*

*New "Gross (i.e. without liabilities) total wealth distributions (i.e, variables) for each implicate; adjusted concept: subtracting self-employed business from real assets in  first step!!!*
forvalues i=1(1)5{
gen RealadC`i'=da1000 - da1140 if im0100==`i'
}

forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
egen WwFAadC`i'`z'`y'=rowtotal(RealadC`i' wFAadC`i'`z'`y' WForbes) if im0100==`i'
}
}
}

*estimating alpha coeficient and procedure to find the optimal lower bound: Kolmogorv-Smirnov Test*
set more off

*generating all the relevant variables for OLS, accroding to Vermeulen (2016) with different lower bounds(=xmin)*

*cumulative sum of HH(=household) weights*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -WwFAadC`i'`z'`y'
```

```
375   gen SWwFAadC`i'`z'`y'`x'=sum(hw0010) if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
376   }
377   }
378   }
379   }
380
381   *total sum of HH weights*
382   forvalues i=1(1)5{
383   forvalues y=0(1)9{
384   forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
385   foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
386   gsort -WwFAadC`i'`z'`y'
387   egen nWwFAadC`i'`z'`y'`x'=total(hw0010) if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
388   }
389   }
390   }
391   }
392
393   *ln(cumulative sum of HH weights/total sum HH weights)*
394   forvalues i=1(1)5{
395   forvalues y=0(1)9{
396   forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
397   foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
398   gsort -WwFAadC`i'`z'`y'
399   gen yWwFAadC`i'`z'`y'`x'=ln(SWwFAadC`i'`z'`y'`x'/nWwFAadC`i'`z'`y'`x') if WwFAadC`i'`z'`y'>=`x' &
      im0100==`i'
400   }
401   }
402   }
403   }
404
405   *ln(wealth/lower bound=xmin)*(-1)*
406   forvalues i=1(1)5{
407   forvalues y=0(1)9{
408   forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
409   foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
410   gsort -WwFAadC`i'`z'`y'
411   gen xWwFAadC`i'`z'`y'`x'=ln(WwFAadC`i'`z'`y'/`x')*(-1) if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
412   }
413   }
414   }
415   }
416
417   *OLS; no constant*
418   forvalues i=1(1)5{
419   forvalues y=0(1)9{
420   forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
421   foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
422   gsort -WwFAadC`i'`z'`y'
423   reg yWwFAadC`i'`z'`y'`x' xWwFAadC`i'`z'`y'`x' if im0100==`i', nocons
424   gen alpha`i'`z'`y'`x'=_b[x]
425   }
426   }
427   }
428   }
429
430   *generate fitted distribution*
431   forvalues i=1(1)5{
432   forvalues y=0(1)9{
433   forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
434   foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
435   gsort -WwFAadC`i'`z'`y'
436   gen Ffit`i'`z'`y'`x'=(WwFAadC`i'`z'`y'/`x')^(-alpha`i'`z'`y'`x') if WwFAadC`i'`z'`y'>=`x' & im0100
      ==`i'
437   }
438   }
439   }
440   }
```

```
441
442  *generate empirical distribution*
443  forvalues i=1(1)5{
444  forvalues y=0(1)9{
445  forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
446  foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
447  gsort -WwFAadC`i'`z'`y'
448  gen Femp`i'`z'`y'`x'=(SWwFAadC`i'`z'`y'`x'/nWwFAadC`i'`z'`y'`x') if WwFAadC`i'`z'`y'>=`x' & im0100
     ==`i'
449  }
450  }
451  }
452  }
453
454  *finding the (maximum) distance between fitted and empirical distribution=Kolmogorv-Smirnov; in
     STATA: ksmirnov=ks*
455  forvalues i=1(1)5{
456  forvalues y=0(1)9{
457  forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
458  foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
459  gsort -WwFAadC`i'`z'`y'
460  ksmirnov Femp`i'`z'`y'`x'=Ffit`i'`z'`y'`x' if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
461  gen ksWwFAadC`i'`z'`y'`x'=r(D)
462  scalar ksWwFAadC`i'`z'`y'`x'=r(D)
463  }
464  }
465  }
466  }
467
468  *optimal lowerbound=lb is the lower bound for which the fitted distribution has the smallest
     distance to the empirical distribution*
469  forvalues i=1(1)5{
470  forvalues y=0(1)9{
471  forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
472  egen lbWwFAadC`i'`z'`y'=rowmin( ksWwFAadC`i'`z'`y'`=scalar(wminLB)' - ksWwFAadC
     `i'`z'`y'`=scalar(wminUB)') if im0100==`i'
473  }
474  }
475  }
476
477  *creating the xmin variable*
478  forvalues i=1(1)5{
479  forvalues y=0(1)9{
480  forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
481  foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
482  gen xminWwFAadC`i'`z'`y'`x'=`x' if lbWwFAadC`i'`z'`y'==ksWwFAadC`i'`z'`y'`x'
483  }
484  }
485  }
486  }
487
488  ******housekeeping, getting rid of  variables with missing entries that were created in the
     procedure*****
489  dropmiss, force
490
491  *renaming the xmin variable, thus loosing the `x' in the name*
492  forvalues i=1(1)5{
493  forvalues y=0(1)9{
494  forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
495  rename xminWwFAadC`i'`z'`y' xminWwFAadC`i'`z'`y'
496  }
497  }
498  }
499
500  *creating the scalar xmin for future calculations*
501  forvalues i=1(1)5{
502  forvalues y=0(1)9{
503  forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
```

```
scalar xminWwFAadC`i'`z'`y'=xminWwFAadC`i'`z'`y'
di xminWwFAadC`i'`z'`y'
}
}
}

*Knowing xmin, the ALPHA coefficients for all 5 implicates are estimated*
*estimating the pareto coefficient via OLS without no-constant as in Vermeulen(2014 and 2016)*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
sum xminWwFAadC`i'`z'`y' if im0100==`i'
di r(mean)
scalar XMIN`i'`z'`y'=r(mean)
di XMIN`i'`z'`y'
gsort -WwFAadC`i'`z'`y'
gen SWwFAadC`i'`z'`y'=sum(hw0010) if WwFAadC`i'`z'`y'>=`=scalar(XMIN`i'`z'`y')' & im0100==`i'
egen nWwFAadC`i'`z'`y'=total(hw0010) if WwFAadC`i'`z'`y'>=`=scalar(XMIN`i'`z'`y')' & im0100==`i'
gen yWwFAadC`i'`z'`y'=ln(SWwFAadC`i'`z'`y'/nWwFAadC`i'`z'`y') if WwFAadC`i'`z'`y'>=
`=scalar(XMIN`i'`z'`y')' & im0100==`i'
gen xWwFAadC`i'`z'`y'=ln(WwFAadC`i'`z'`y'/`=scalar(XMIN`i'`z'`y')')*(-1) if WwFAadC`i'`z'`y'>=
`=scalar(XMIN`i'`z'`y')' & im0100==`i'
reg yWwFAadC`i'`z'`y' xWwFAadC`i'`z'`y' if im0100==`i' , nocons
scalar alphaWwFAadC`i'`z'`y'=_b[x]
di alphaWwFAadC`i'`z'`y'
}
}
}

*defining upper bound of gap betwen HFCS and rich list, i.e. the least-rich entry of the rich
list *
scalar upper=`=scalar(least_rich_observation)'

*Calculating number of HH above xmin*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
egen HHxminWwFAadC`i'`z'`y'=total(hw0010) if WwFAadC`i'`z'`y'>XMIN`i'`z'`y' & WwFAadC`i'`z'`y'<
`=scalar(upper)' & im0100==`i'
sum HHxminWwFAadC`i'`z'`y'
scalar HHxminWwFAadC`i'`z'`y'=r(mean)
di HHxminWwFAadC`i'`z'`y'
di `=scalar(HHxminWwFAadC`i'`z'`y')'
}
}
}

*calculating wealth in tail (=Wtail) given a specific pair of alpha and xmin, WTail is the wealth
in the pareto tail*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
integrate, f(x*(`=scalar(alphaWwFAadC`i'`z'`y')'*`=scalar(XMIN`i'`z'`y')'^
`=scalar(alphaWwFAadC`i'`z'`y')')/(x)^(1+`=scalar(alphaWwFAadC`i'`z'`y')')) l(
`=scalar(XMIN`i'`z'`y')') u(`=scalar(upper)') quadpts(1000) vectorise
di r(integral)
scalar intXMIN`i'`z'`y'=r(integral)
di intXMIN`i'`z'`y'
scalar WTail`i'`z'`y'=`=scalar(HHxminWwFAadC`i'`z'`y')'*`=scalar(intXMIN`i'`z'`y')'
di WTail`i'`z'`y'
gen WTail`i'`z'`y'=`=scalar(WTail`i'`z'`y')' if im0100==`i'
}
}
}

*finding real assets above XMIN, dont forget multiplication with HH weights*
*hw in the variable name indicates that it is multiplied with HH weights*
forvalues i=1(1)5{
```

```
gen RealadC`i'hw=RealadC`i'*hw0010 if im0100==`i'
}


*real assets above XMIN*
*(change variable names in the final version of the report-could be misleading)*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
egen totalRealadC`i'`z'`y'=total(RealadC`i'hw) if WwFAadC`i'`z'`y'>=`=scalar(XMIN`i'`z'`y')' & im0100==`i'
sum totalRealadC`i'`z'`y'
scalar totalRealadC`i'`z'`y'=r(mean)
}
}
}

*finding financial assets below XMIN*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
gen wFAadC`i'`z'`y'hw=wFAadC`i'`z'`y'*hw0010 if im0100==`i'
egen totalFAadC`i'`z'`y'=total(wFAadC`i'`z'`y'hw) if WwFAadC`i'`z'`y'<`=scalar(XMIN`i'`z'`y')' & im0100==`i'
sum totalFAadC`i'`z'`y'
scalar totalFAadC`i'`z'`y'=r(mean)
}
}
}

*putting together total (new) financial wealth;WTail-totalReal+totalDadC AND subtracting FA_EUrostat in order to get the absolute difference betwen HFCS and national accounts*
*subtract also total wealth of Forbes observations from the FAadC_Eurostat*

forvalues i=1(1)5{
egen totalWForbes`i'=total(WForbes) if im0100==`i'
sum totalWForbes`i'
scalar totalWForbes`i'=r(mean)
}

forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
gen DIFF`i'`z'`y'=abs([`=scalar(WTail`i'`z'`y')'-`=scalar(totalRealadC`i'`z'`y')'+`=scalar(totalFAadC`i'`z'`y')'-(`=scalar(FAadC_Eurostat)'-`=scalar(totalWForbes`i')')]) if im0100==`i'
}
}
}

*finding the minimum difference through "rowmin"*
forvalues i=1(1)5{
egen minDiff`i'=rowmin(DIFF`i'`=scalar(zLB)'0 - DIFF`i'`=scalar(zUB)'9) if im0100==`i'
}

*choosing the wealth distribution, thus the name: "the chosen one"*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
gen  thechosenOne`i'`z'`y' = WwFAadC`i'`z'`y' if DIFF`i'`z'`y'==minDiff`i'
}
}
}

*housekeeping, getting rid of variables with empty entries that were created in the process*
dropmiss, force
```

```
*renaming "thechosenOne" in order to lose the `z''y'*
forvalues i=1(1)5{
rename thechosenOne`i' thechosenOne`i'
}

*************************************************************************
*************************************************************************

*repeating the process but only with the 5 "chosen" distributions, KS is performed as well but is only in place as a double check (some calculation time could be saved here, if required)*
forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -thechosenOne`i'
gen SthechosenOne`i''x'=sum(hw0010) if thechosenOne`i'>=`x' & im0100==`i'
}
}

forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -thechosenOne`i'
egen nthechosenOne`i''x'=total(hw0010) if thechosenOne`i'>=`x' & im0100==`i'
}
}

forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -thechosenOne`i'
gen ythechosenOne`i''x'=ln(SthechosenOne`i''x'/nthechosenOne`i''x') if thechosenOne`i'>=`x' & im0100==`i'
}
}

forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -thechosenOne`i'
gen xthechosenOne`i''x'=ln(thechosenOne`i'/`x')*(-1) if thechosenOne`i'>=`x' & im0100==`i'
}
}

forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -thechosenOne`i'
reg ythechosenOne`i''x' xthechosenOne`i''x' if im0100==`i', nocons
gen alpha`i''x'=_b[x]
}
}

forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -thechosenOne`i'
gen Ffit`i''x'=(thechosenOne`i'/`x')^(-alpha`i''x') if thechosenOne`i'>=`x' & im0100==`i'
}
}

forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -thechosenOne`i'
gen Femp`i''x'=(SthechosenOne`i''x'/nthechosenOne`i''x') if thechosenOne`i'>=`x' & im0100==`i'
}
}

*ksmirnov=ks*
forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gsort -thechosenOne`i'
ksmirnov Femp`i''x'=Ffit`i''x' if thechosenOne`i'>=`x' & im0100==`i'
gen ksthechosenOne`i''x'=r(D)
scalar ksthechosenOne`i''x'=r(D)
```

```
694   }
695   }
696
697   *lowerbound=lb*
698   forvalues i=1(1)5{
699   egen lbthechosenOne`i'=rowmin( ksthechosenOne`i'`=scalar(wminLB)' - ksthechosenOne
      `i'`=scalar(wminUB)') if im0100==`i'
700   }
701
702
703   *choosing xmin*
704   forvalues i=1(1)5{
705   foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
706   gen xminthechosenOne`i'`x'=`x' if lbthechosenOne`i'==ksthechosenOne`i'`x'
707   }
708   }
709
710   *housekeeping*
711   dropmiss, force
712
713   *renaming, thus loosing the `x' in the name*
714   forvalues i=1(1)5{
715   rename xminthechosenOne`i' xminthechosenOne`i'
716   }
717
718   *creating scalars for later calcualtios*
719   forvalues i=1(1)5{
720   global xminthechosenOne`i'=xminthechosenOne`i'
721   scalar xminthechosenOne`i'=xminthechosenOne`i'
722   di xminthechosenOne`i'
723   }
724
725   *estimating the pareto coefficient via OLS with no-constant as in Vermeulen(2014)*
726   forvalues i=1(1)5{
727   sum xminthechosenOne`i' if im0100==`i'
728   di r(mean)
729   scalar XMIN`i'=r(mean)
730   di XMIN`i'
731   gsort -thechosenOne`i'
732   gen SthechosenOne`i'=sum(hw0010) if thechosenOne`i'>=`=scalar(XMIN`i')' & im0100==`i'
733   egen nthechosenOne`i'=total(hw0010) if thechosenOne`i'>=`=scalar(XMIN`i')' & im0100==`i'
734   gen ythechosenOne`i'=ln(SthechosenOne`i'/nthechosenOne`i') if thechosenOne`i'>=`=scalar(XMIN`i')'
      & im0100==`i'
735   gen xthechosenOne`i'=ln(thechosenOne`i'/`=scalar(XMIN`i')')*(-1) if thechosenOne`i'>=
      `=scalar(XMIN`i')' & im0100==`i'
736   reg ythechosenOne`i' xthechosenOne`i' if im0100==`i' , nocons
737   scalar alphathechosenOne`i'=_b[x]
738   di alphathechosenOne`i'
739   gen alphathechosenOne`i'=alphathechosenOne`i'
740   }
741
742   *number HH above xmin*
743   forvalues i=1(1)5{
744   egen HHxminthechosenOne`i'=total(hw0010) if thechosenOne`i'>XMIN`i' & thechosenOne`i'<1000000000 &
       im0100==`i'
745   sum HHxminthechosenOne`i'
746   scalar HHxminthechosenOne`i'=r(mean)
747   di HHxminthechosenOne`i'
748   di `=scalar(HHxminthechosenOne`i')'
749   }
750
751
752   *Wealth in Tail based on "thechosenOne"*
753   forvalues i=1(1)5{
754   integrate, f(x*(`=scalar(alphathechosenOne`i')'*`=scalar(XMIN`i')'^`=scalar(alphathechosenOne`i')'
      )/(x)^(1+`=scalar(alphathechosenOne`i')')) l(`=scalar(XMIN`i')') u(`=scalar(upper)') quadpts(1000)
       vectorise
755   di r(integral)
```

```
scalar intXMIN`i'=r(integral)
di intXMIN`i'
scalar WTail`i'=`=scalar(HHxminthechosenOne`i')'*`=scalar(intXMIN`i')'
di WTail`i'
}

*Realassets above wmin based on "thechosenOne"*
forvalues i=1(1)5{
egen totalRealadC`i'aboveXmin=total(RealadC`i'hw) if thechosenOne`i'>=`=scalar(XMIN`i')' & im0100
==`i'
sum totalRealadC`i'aboveXmin
scalar totalRealadC`i'aboveXmin=r(mean)
}

*all Realassets*
forvalues i=1(1)5{
egen allRealadC`i'=total(RealadC`i'hw) if im0100==`i'
sum allRealadC`i'
scalar allRealadC`i'=r(mean)
}

*z*
forvalues i=1(1)5{
gen sumthechosenOne`i'=thechosenOne`i'*hw0010 if thechosenOne`i'<`=scalar(upper)' & im0100==`i'
egen allthechosenOne`i'=total(sumthechosenOne`i') if im0100==`i'
sum allthechosenOne`i'
scalar allthechosenOne`i'=r(mean)
}

forvalues i=1(1)5{
scalar newRatio`i'=(`=scalar(allthechosenOne`i')'-`=scalar(allRealadC`i')')/(
`=scalar(FAadC_Eurostat)'/100)
}

forvalues i=1(1)5{
scalar z`i'=`=scalar(newRatio`i')'/`=scalar(RatioFAadC`i')'
di `=scalar(z`i')'
}

*total Investment in non-self-employed business below xmin based on "the consenOne"*
forvalues i=1(1)5{
gen InvestNSE`i'=hd1010*hw0010 if im0100==`i'
}

forvalues i=1(1)5{
egen totalInvestNSE`i'=total(InvestNSE`i') if thechosenOne`i'<`=scalar(XMIN`i')' & im0100==`i'
sum totalInvestNSE`i'
scalar totalInvestNSE`i'=r(mean)
}

*total Investment in self-employed business below xmin based on "the consenOne*
forvalues i=1(1)5{
gen InvestSE`i'=hd0200*hw0010 if im0100==`i'
}

forvalues i=1(1)5{
egen totalInvestSE`i'=total(InvestSE`i') if thechosenOne`i'<`=scalar(XMIN`i')' & im0100==`i'
sum totalInvestSE`i'
scalar totalInvestSE`i'=r(mean)
}

* assessing the "distribution" of liabilities based on the "chosen" gross wealth distribution*
forvalues i=1(1)5{
egen HH`i'=total(hw0010) if im0100==`i'
sum HH`i'
scalar HH`i'=r(mean)
di HH`i'
di `=scalar(HH`i')'
```

```
822  }
823
824  forvalues i=1(1)5{
825  forvalues y=1(1)10{
826  scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
827  di `=scalar(HH`y'dec`i')'
828  }
829  }
830
831  forvalues i=1(1)5{
832  sort thechosenOne`i'
833  gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
834  }
835
836  forvalues i=1(1)5{
837  gen dec`i'=1 if im0100==`i'
838  }
839
840  *defining deciles of the gross wealth distribution*
841  forvalues i=1(1)5{
842  replace dec`i'=2 if SHHthechosenOne`i'>`=scalar(HH1dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH2dec`i')' & im0100==`i'
843  replace dec`i'=3 if SHHthechosenOne`i'>`=scalar(HH2dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH3dec`i')' & im0100==`i'
844  replace dec`i'=4 if SHHthechosenOne`i'>`=scalar(HH3dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH4dec`i')' & im0100==`i'
845  replace dec`i'=5 if SHHthechosenOne`i'>`=scalar(HH4dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH5dec`i')' & im0100==`i'
846  replace dec`i'=6 if SHHthechosenOne`i'>`=scalar(HH5dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH6dec`i')' & im0100==`i'
847  replace dec`i'=7 if SHHthechosenOne`i'>`=scalar(HH6dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH7dec`i')' & im0100==`i'
848  replace dec`i'=8 if SHHthechosenOne`i'>`=scalar(HH7dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH8dec`i')' & im0100==`i'
849  replace dec`i'=9 if SHHthechosenOne`i'>`=scalar(HH8dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH9dec`i')' & im0100==`i'
850  replace dec`i'=10 if SHHthechosenOne`i'>`=scalar(HH9dec`i')' & SHHthechosenOne`i'<=
     `=scalar(HH10dec`i')' & im0100==`i'
851  }
852
853  forvalues i=1(1)5{
854  forvalues y=1(1)10{
855  egen sumliabHW`i'dec`y'=total(liabHW`i') if im0100==`i' & dec`i'==`y'
856  sum sumliabHW`i'dec`y'
857  di r(mean)
858  scalar totalliab`i'dec`y'=r(mean)
859  }
860  }
861
862  forvalues i=1(1)5{
863  scalar check`i'= `=scalar(totalliab`i'dec1)'+`=scalar(totalliab`i'dec2)'+
     `=scalar(totalliab`i'dec3)'+`=scalar(totalliab`i'dec4)'+`=scalar(totalliab`i'dec5)'+
     `=scalar(totalliab`i'dec6)'+`=scalar(totalliab`i'dec7)'+`=scalar(totalliab`i'dec8)'+
     `=scalar(totalliab`i'dec9)'+`=scalar(totalliab`i'dec10)'-`=scalar(totalliab`i')'
864  di `=scalar(check`i')'
865  }
866
867  forvalues i=1(1)5{
868  forvalues y=1(1)10{
869  scalar percentdec`y'liab`i'=`=scalar(totalliab`i'dec`y')'/(`=scalar(totalliab`i')'/100)
870  di `=scalar(percentdec`y'liab`i')'
871  }
872  }
873
874  *difference financial balace sheets and HFCS*
875  forvalues i=1(1)5{
876  scalar DiffLiab`i'=`=scalar(LIABadC_Eurostat)'-`=scalar(totalliab`i')'
877  }
```

```
878
879   *liability gap to be divided*
880   forvalues i=1(1)5{
881   forvalues y=1(1)10{
882   scalar LiabTBdivideddec`y'imp`i'=(`=scalar(DiffLiab`i')'/100)*percentdec`y'liab`i'
883   di `=scalar(LiabTBdivideddec`y'imp`i')'
884   }
885   }
886
887
888   *saving alpha, xmin, HH above xmin*
889   forvalues i=1(1)5{
890   sum alphathechosenOne`i'
891   di r(mean)
892   scalar alphathechosenOne`i'=r(mean)
893   sum HHxminthechosenOne`i'
894   di r(mean)
895   scalar HHxminthechosenOne`i'=r(mean)
896   sum xminthechosenOne`i'
897   di r(mean)
898   scalar xminthechosenOne`i'=r(mean)
899   }
900
901   *export alpha, xmin, HH above xmin to excel table*
902   putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
903
904   putexcel A2=("alpha")
905   sleep 1000
906   putexcel A3=`=scalar(alphathechosenOne1)'
907   sleep 1000
908   putexcel A4=`=scalar(alphathechosenOne2)'
909   sleep 1000
910   putexcel A5=`=scalar(alphathechosenOne3)'
911   sleep 1000
912   putexcel A6=`=scalar(alphathechosenOne4)'
913   sleep 1000
914   putexcel A7=`=scalar(alphathechosenOne5)'
915   sleep 1000
916
917   putexcel B2=("w_min")
918   sleep 1000
919   putexcel B3=`=scalar(xminthechosenOne1)'
920   sleep 1000
921   putexcel B4=`=scalar(xminthechosenOne2)'
922   sleep 1000
923   putexcel B5=`=scalar(xminthechosenOne3)'
924   sleep 1000
925   putexcel B6=`=scalar(xminthechosenOne4)'
926   sleep 1000
927   putexcel B7=`=scalar(xminthechosenOne5)'
928   sleep 1000
929
930   putexcel C2=("HH_above_wmin")
931   sleep 1000
932   putexcel C3=`=scalar(HHxminthechosenOne1)'
933   sleep 1000
934   putexcel C4=`=scalar(HHxminthechosenOne2)'
935   sleep 1000
936   putexcel C5=`=scalar(HHxminthechosenOne3)'
937   sleep 1000
938   putexcel C6=`=scalar(HHxminthechosenOne4)'
939   sleep 1000
940   putexcel C7=`=scalar(HHxminthechosenOne5)'
941   sleep 1000
942
943   putexcel J2=("realassets_above_wmin")
944   sleep 1000
945   putexcel J3=`=scalar(totalRealadC1aboveXmin)'
```

```
946   sleep 1000
947   putexcel J4=`=scalar(totalRealadC2aboveXmin)'
948   sleep 1000
949   putexcel J5=`=scalar(totalRealadC3aboveXmin)'
950   sleep 1000
951   putexcel J6=`=scalar(totalRealadC4aboveXmin)'
952   sleep 1000
953   putexcel J7=`=scalar(totalRealadC5aboveXmin)'
954   sleep 1000
955
956   putexcel K2=("wealth_in_tail")
957   sleep 1000
958   putexcel K3=`=scalar(WTail1)'
959   sleep 1000
960   putexcel K4=`=scalar(WTail2)'
961   sleep 1000
962   putexcel K5=`=scalar(WTail3)'
963   sleep 1000
964   putexcel K6=`=scalar(WTail4)'
965   sleep 1000
966   putexcel K7=`=scalar(WTail5)'
967   sleep 1000
968
969   putexcel L2=("total_forbes_wealth")
970   sleep 1000
971   putexcel L3=`=scalar(totalWForbes1)'
972   sleep 1000
973   putexcel L4=`=scalar(totalWForbes2)'
974   sleep 1000
975   putexcel L5=`=scalar(totalWForbes3)'
976   sleep 1000
977   putexcel L6=`=scalar(totalWForbes4)'
978   sleep 1000
979   putexcel L7=`=scalar(totalWForbes5)'
980   sleep 1000
981
982   putexcel M2=("investment_in_nonselfemployedbusiness_below_xmin")
983   sleep 1000
984   putexcel M3=`=scalar(totalInvestNSE1)'
985   sleep 1000
986   putexcel M4=`=scalar(totalInvestNSE2)'
987   sleep 1000
988   putexcel M5=`=scalar(totalInvestNSE3)'
989   sleep 1000
990   putexcel M6=`=scalar(totalInvestNSE4)'
991   sleep 1000
992   putexcel M7=`=scalar(totalInvestNSE5)'
993   sleep 1000
994
995   putexcel N2=("investment_in_selfemployedbusiness_below_xmin")
996   sleep 1000
997   putexcel N3=`=scalar(totalInvestSE1)'
998   sleep 1000
999   putexcel N4=`=scalar(totalInvestSE2)'
1000  sleep 1000
1001  putexcel N5=`=scalar(totalInvestSE3)'
1002  sleep 1000
1003  putexcel N6=`=scalar(totalInvestSE4)'
1004  sleep 1000
1005  putexcel N7=`=scalar(totalInvestSE5)'
1006  sleep 1000
1007
1008  putexcel O2=("total_realassets")
1009  sleep 1000
1010  putexcel O3=`=scalar(allRealadC1)'
1011  sleep 1000
1012  putexcel O4=`=scalar(allRealadC2)'
1013  sleep 1000
```

```
putexcel O5=`=scalar(allRealadC3)'
sleep 1000
putexcel O6=`=scalar(allRealadC4)'
sleep 1000
putexcel O7=`=scalar(allRealadC5)'
sleep 1000

putexcel P2=("FA_coverageratio_after_adjustment")
sleep 1000
putexcel P3=`=scalar(newRatio1)'
sleep 1000
putexcel P4=`=scalar(newRatio2)'
sleep 1000
putexcel P5=`=scalar(newRatio3)'
sleep 1000
putexcel P6=`=scalar(newRatio4)'
sleep 1000
putexcel P7=`=scalar(newRatio5)'
sleep 1000

putexcel Q2=("z")
sleep 1000
putexcel Q3=`=scalar(z1)'
sleep 1000
putexcel Q4=`=scalar(z2)'
sleep 1000
putexcel Q5=`=scalar(z3)'
sleep 1000
putexcel Q6=`=scalar(z4)'
sleep 1000
putexcel Q7=`=scalar(z5)'
sleep 1000

putexcel R2=("F512")
sleep 1000
putexcel R3=`=scalar(F512)'
sleep 1000
putexcel S2=("F519 ")
sleep 1000
putexcel S3=`=scalar(F519)'
sleep 1000

*implicate1*
putexcel T1=("percent_of_liab_in_grosswealth_decile")
sleep 1000
putexcel T2=("implicate1")
sleep 1000
putexcel T3=`=scalar(percentdec1liab1)'
sleep 1000
putexcel T4=`=scalar(percentdec2liab1)'
sleep 1000
putexcel T5=`=scalar(percentdec3liab1)'
sleep 1000
putexcel T6=`=scalar(percentdec4liab1)'
sleep 1000
putexcel T7=`=scalar(percentdec5liab1)'
sleep 1000
putexcel T8=`=scalar(percentdec6liab1)'
sleep 1000
putexcel T9=`=scalar(percentdec7liab1)'
sleep 1000
putexcel T10=`=scalar(percentdec8liab1)'
sleep 1000
putexcel T11=`=scalar(percentdec9liab1)'
sleep 1000
putexcel T12=`=scalar(percentdec10liab1)'
sleep 1000

```

```
putexcel T14=("missing_liabilities_€")
sleep 1000
putexcel T15=`=scalar(DiffLiab1)'
sleep 1000


*implicate2*
putexcel U1=("percent_of_liab_in_grosswealth_decile")
sleep 1000
putexcel U2=("implicate2")
sleep 1000
putexcel U3=`=scalar(percentdec1liab2)'
sleep 1000
putexcel U4=`=scalar(percentdec2liab2)'
sleep 1000
putexcel U5=`=scalar(percentdec3liab2)'
sleep 1000
putexcel U6=`=scalar(percentdec4liab2)'
sleep 1000
putexcel U7=`=scalar(percentdec5liab2)'
sleep 1000
putexcel U8=`=scalar(percentdec6liab2)'
sleep 1000
putexcel U9=`=scalar(percentdec7liab2)'
sleep 1000
putexcel U10=`=scalar(percentdec8liab2)'
sleep 1000
putexcel U11=`=scalar(percentdec9liab2)'
sleep 1000
putexcel U12=`=scalar(percentdec10liab2)'
sleep 1000

putexcel U14=("missing_liabilities_€")
sleep 1000
putexcel U15=`=scalar(DiffLiab2)'
sleep 1000


*implicate3*
putexcel V1=("percent_of_liab_in_grosswealth_decile")
sleep 1000
putexcel V2=("implicate3")
sleep 1000
putexcel V3=`=scalar(percentdec1liab3)'
sleep 1000
putexcel V4=`=scalar(percentdec2liab3)'
sleep 1000
putexcel V5=`=scalar(percentdec3liab3)'
sleep 1000
putexcel V6=`=scalar(percentdec4liab3)'
sleep 1000
putexcel V7=`=scalar(percentdec5liab3)'
sleep 1000
putexcel V8=`=scalar(percentdec6liab3)'
sleep 1000
putexcel V9=`=scalar(percentdec7liab3)'
sleep 1000
putexcel V10=`=scalar(percentdec8liab3)'
sleep 1000
putexcel V11=`=scalar(percentdec9liab3)'
sleep 1000
putexcel V12=`=scalar(percentdec10liab3)'
sleep 1000

putexcel V14=("missing_liabilities_€")
sleep 1000
putexcel V15=`=scalar(DiffLiab3)'
sleep 1000
```

```
1150
1151
1152    *implicate4*
1153    putexcel W1=("percent_of_liab_in_grosswealth_decile")
1154    sleep 1000
1155    putexcel W2=("implicate4")
1156    sleep 1000
1157    putexcel W3=`=scalar(percentdec1liab4)'
1158    sleep 1000
1159    putexcel W4=`=scalar(percentdec2liab4)'
1160    sleep 1000
1161    putexcel W5=`=scalar(percentdec3liab4)'
1162    sleep 1000
1163    putexcel W6=`=scalar(percentdec4liab4)'
1164    sleep 1000
1165    putexcel W7=`=scalar(percentdec5liab4)'
1166    sleep 1000
1167    putexcel W8=`=scalar(percentdec6liab4)'
1168    sleep 1000
1169    putexcel W9=`=scalar(percentdec7liab4)'
1170    sleep 1000
1171    putexcel W10=`=scalar(percentdec8liab4)'
1172    sleep 1000
1173    putexcel W11=`=scalar(percentdec9liab4)'
1174    sleep 1000
1175    putexcel W12=`=scalar(percentdec10liab4)'
1176    sleep 1000
1177
1178    putexcel W14=("missing_liabilities_€")
1179    sleep 1000
1180    putexcel W15=`=scalar(DiffLiab4)'
1181    sleep 1000
1182
1183
1184    *implicate5*
1185    putexcel X1=("percent_of_liab_in_grosswealth_decile")
1186    sleep 1000
1187    putexcel X2=("implicate5")
1188    sleep 1000
1189    putexcel X3=`=scalar(percentdec1liab1)'
1190    sleep 1000
1191    putexcel X4=`=scalar(percentdec2liab1)'
1192    sleep 1000
1193    putexcel X5=`=scalar(percentdec3liab1)'
1194    sleep 1000
1195    putexcel X6=`=scalar(percentdec4liab1)'
1196    sleep 1000
1197    putexcel X7=`=scalar(percentdec5liab1)'
1198    sleep 1000
1199    putexcel X8=`=scalar(percentdec6liab1)'
1200    sleep 1000
1201    putexcel X9=`=scalar(percentdec7liab1)'
1202    sleep 1000
1203    putexcel X10=`=scalar(percentdec8liab1)'
1204    sleep 1000
1205    putexcel X11=`=scalar(percentdec9liab1)'
1206    sleep 1000
1207    putexcel X12=`=scalar(percentdec10liab1)'
1208    sleep 1000
1209
1210    putexcel X14=("missing_liabilities_€")
1211    sleep 1000
1212    putexcel X15=`=scalar(DiffLiab5)'
1213    sleep 1000
1214
1215    *average wealth in implicate-specific pareto tail*
1216    putexcel Y2=("average_wealth_in_tail")
1217    sleep 1000
```

```
putexcel Y3=`=scalar(intXMIN1)'
sleep 1000
putexcel Y4=`=scalar(intXMIN2)'
sleep 1000
putexcel Y5=`=scalar(intXMIN3)'
sleep 1000
putexcel Y6=`=scalar(intXMIN4)'
sleep 1000
putexcel Y7=`=scalar(intXMIN5)'
sleep 1000



*save and exit*
save "${countryfile}\\${country}Original.dta", replace


*************************************************************************
***** tax revenues implicate 1*******
use "${countryfile}\\${country}Original.dta"

*original*

keep if im0100==1
keep if thechosenOne1<`=scalar(xminthechosenOne1)' | thechosenOne1>=
`=scalar(least_rich_observation)'
gsort - thechosenOne1

*finding out the number of rich list observations*

egen HHForbes=total(hw0010) if thechosenOne1>=`=scalar(least_rich_observation)'
sum HHForbes
scalar HHForbes=r(mean)
di HHForbes
di `=scalar(HHForbes)'

keep thechosenOne1 hw0010 im0100
save "${countryfile}\\${country}Experiment.dta", replace
clear
*creating synt HH*
scalar HHxminthechosenOne1round = round(`=scalar(HHxminthechosenOne1)')
set obs `=scalar(HHxminthechosenOne1round)'
egen double thechosenOne1 = rndraw() , pareto(`=scalar(xminthechosenOne1)'
`=scalar(alphathechosenOne1)')
gen hw0010=1
gen im0100=1
gsort -thechosenOne1
gen n= _n


*subtracting the #number of richest individuals = #number of forbes observations, which are to be
replaced with the observations of the Forbes rich list*

keep if _n > `=scalar(HHForbes)'
append using "${countryfile}\\${country}Experiment.dta"
gsort -thechosenOne1

*finding out the number of all HH*
egen HH1=total(hw0010)
sum HH1
scalar HH1=r(mean)
di HH1
di `=scalar(HH1)'


forvalues i=1(1)1{
forvalues y=1(1)10{
scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
```

```
1283  di `=scalar(HH`y'dec`i')'
1284  }
1285  }
1286  forvalues i=1(1)1{
1287  sort thechosenOne`i'
1288  gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
1289  }
1290  forvalues i=1(1)1{
1291  gen dec`i'=1 if im0100==`i'
1292  }
1293  *defining deciles of the gross wealth distribution*
1294  forvalues i=1(1)1{
1295  replace dec`i'=2 if SHHthechosenOne`i'>`=scalar(HH1dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH2dec`i')' & im0100==`i'
1296  replace dec`i'=3 if SHHthechosenOne`i'>`=scalar(HH2dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH3dec`i')' & im0100==`i'
1297  replace dec`i'=4 if SHHthechosenOne`i'>`=scalar(HH3dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH4dec`i')' & im0100==`i'
1298  replace dec`i'=5 if SHHthechosenOne`i'>`=scalar(HH4dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH5dec`i')' & im0100==`i'
1299  replace dec`i'=6 if SHHthechosenOne`i'>`=scalar(HH5dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH6dec`i')' & im0100==`i'
1300  replace dec`i'=7 if SHHthechosenOne`i'>`=scalar(HH6dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH7dec`i')' & im0100==`i'
1301  replace dec`i'=8 if SHHthechosenOne`i'>`=scalar(HH7dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH8dec`i')' & im0100==`i'
1302  replace dec`i'=9 if SHHthechosenOne`i'>`=scalar(HH8dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH9dec`i')' & im0100==`i'
1303  replace dec`i'=10 if SHHthechosenOne`i'>`=scalar(HH9dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH10dec`i')' & im0100==`i'
1304  }
1305
1306  gen lia = `=scalar(LIABadC_Eurostat)' /100* `=scalar(percentdec1liab1)' / (`=scalar(HH1)'/10) if
      dec1 == 1
1307
1308  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 2
1309  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 3
1310  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 4
1311  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 5
1312  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 6
1313  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 7
1314  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 8
1315  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 9
1316  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab1)'/(`=scalar(HH1)'/10) if
      dec1 == 10
1317
1318  *litlle check if liabs add up to correct total*
1319  gen liahw0010= lia*hw0010
1320  egen totalliahw0010 = total(liahw0010)
1321  sum totalliahw0010
1322  di `=scalar(LIABadC_Eurostat)'
1323
1324  *creating the netw wealth variable and applying the 3 elasticities as weöö as tje simple tax
      schedule of 1% for weatlh above 1Mio EUR and 1.5% for wealth above 1.5 Mio EUR*
1325
1326  *gen netwealth*
1327  gen netwealth = thechosenOne1 - lia
1328  gen netwealthhw0010= netwealth*hw0010
1329  egen totalnetwealthhw0010 = total(netwealthhw0010)
1330  sum totalnetwealthhw0010
```

```
scalar totalnetwealthhw0010=r(mean)

di `=scalar(totalnetwealthhw0010)'

*finding out the number of all HH above the tax threshold of 1 Mio EUR*
egen HHabove1mio=total(hw0010) if netwealth>=1000000
sum HHabove1mio
scalar HHabove1mio=r(mean)
di HHabove1mio
di `=scalar(HHabove1mio)'

*finding out the number of all HH above the tax threshold of 5 Mio EUR*
egen HHabove1p5mio=total(hw0010) if netwealth>=5000000
sum HHabove1p5mio
scalar HHabove1p5mio=r(mean)
di HHabove1p5mio
di `=scalar(HHabove1p5mio)'

*tax base elasticity 1 = 6.9*
gen rev1 = (netwealth-1000000)/100*(100-6.9)*0.01 if netwealth>=1000000
replace rev1 = (netwealth-1000000)/100*(100-6.9-6.9/2)*0.015 if netwealth>=5000000
gen rev1hw0010 = rev1*hw0010

egen sumrev1hw0010=total(rev1hw0010)
sum sumrev1hw0010
scalar sumrev1hw0010=r(mean)

di `=scalar(sumrev1hw0010)'


*elasticity 2 = 11.3*
gen rev2 = (netwealth-1000000)/100*(100-11.3)*0.01 if netwealth>=1000000
replace rev2 = (netwealth-1000000)/100*(100-11.3-11.3/2)*0.015 if netwealth>=5000000
gen rev2hw0010 = rev2*hw0010

egen sumrev2hw0010=total(rev2hw0010)
sum sumrev2hw0010
scalar sumrev2hw0010=r(mean)

di `=scalar(sumrev2hw0010)'

*elasticity 3, 4, 5 ... just copy and paste*

*elasticity 3 = 35*
gen rev3 = netwealth/100*(100-35)*0.01 if netwealth>=1000000
replace rev3 = netwealth/100*(100-35-35/2)*0.015 if netwealth>=5000000
gen rev3hw0010 = rev3*hw0010

egen sumrev3hw0010=total(rev3hw0010)
sum sumrev3hw0010
scalar sumrev3hw0010=r(mean)

di `=scalar(sumrev2hw0010)'

*Ungleichheitsmaße*
pshare netwealth [pw=hw0010], p(10(10)90 95 99) gini
ereturn list
matrix a = e(b)
matrix b =e(G)
*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify


putexcel AL1=("Percentile shares adjusted net wealth")
sleep 1000
putexcel AL2=("0-10")
sleep 1000
putexcel AM2=("10-20")
```

```
sleep 1000
putexcel AN2=("20-30")
sleep 1000
putexcel AO2=("30-40")
sleep 1000
putexcel AP2=("40-50")
sleep 1000
putexcel AQ2=("50-60")
sleep 1000
putexcel AR2=("60-70")
sleep 1000
putexcel AS2=("70-80")
sleep 1000
putexcel AT2=("80-90")
sleep 1000
putexcel AU2=("90-95")
sleep 1000
putexcel AV2=("95-99")
sleep 1000
putexcel AW2=("99")
sleep 1000
putexcel AY2=("gini")
sleep 1000


putexcel AK3=("implicate 1")
sleep 1000
putexcel AL3= matrix(a)
sleep 1000
putexcel AX3=("implicate 1")
sleep 1000
putexcel AY3= matrix(b)
sleep 1000


*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
putexcel AD2=("revenues, elasticity = 6.9%")
sleep 1000
putexcel AD3=`=scalar(sumrev1hw0010)'
sleep 1000
putexcel AE2=("revenues, elasticity = 11.3%")
sleep 1000
putexcel AE3=`=scalar(sumrev2hw0010)'
sleep 1000
putexcel AF2=("revenues, elasticity = 35%")
sleep 1000
putexcel AF3=`=scalar(sumrev3hw0010)'
sleep 1000
putexcel AG2=("total net wealth")
sleep 1000
putexcel AG3=`=scalar(totalnetwealthhw0010)'
sleep 1000
putexcel AH2=("#HHimplicate1")
sleep 1000
putexcel AH3=`=scalar(HH1)'
sleep 1000
putexcel AI2=("#HH above 1 Mio €")
sleep 1000
putexcel AI3=`=scalar(HHabove1mio)'
sleep 1000
putexcel AJ2=("#HH above 5 Mio €")
sleep 1000
putexcel AJ3=`=scalar(HHabove1p5mio)'
sleep 1000

putexcel AB5=("# richlist observations")
sleep 1000
```

```
1467  putexcel AB6=`=scalar(HHForbes)'
1468  sleep 1000
1469
1470  clear
1471
1472
1473  ********************************************************************************
1474  ***** tax revenues implicate 2*******
1475  use "${countryfile}\\${country}Original.dta"
1476
1477  *original*
1478
1479  keep if im0100==2
1480  keep if thechosenOne2<`=scalar(xminthechosenOne2)' | thechosenOne2>=
      `=scalar(least_rich_observation)'
1481  gsort - thechosenOne2
1482
1483  *finding out the number of rich list observations*
1484
1485  egen HHForbes=total(hw0010) if thechosenOne2>=`=scalar(least_rich_observation)'
1486  sum HHForbes
1487  scalar HHForbes=r(mean)
1488  di HHForbes
1489  di `=scalar(HHForbes)'
1490
1491  keep thechosenOne2 hw0010 im0100
1492  save "${countryfile}\\${country}Experiment.dta", replace
1493  clear
1494
1495  *creating synt HH*
1496
1497  scalar HHxminthechosenOne2round = round(`=scalar(HHxminthechosenOne2)')
1498  set obs `=scalar(HHxminthechosenOne2round)'
1499  egen double thechosenOne2 = rndraw() , pareto(`=scalar(xminthechosenOne2)'
      `=scalar(alphathechosenOne2)')
1500  gen hw0010=1
1501  gen im0100=2
1502  gsort -thechosenOne2
1503  gen n= _n
1504
1505
1506  *subtracting the #number of richest individuals = #number of forbes observations, which are to be
      replaced with the observations of the Forbes rich list*
1507
1508  keep if _n > `=scalar(HHForbes)'
1509  append using "${countryfile}\\${country}Experiment.dta"
1510  gsort -thechosenOne2
1511
1512  *finding out the number of all HH*
1513  egen HH2=total(hw0010)
1514  sum HH2
1515  scalar HH2=r(mean)
1516  di HH2
1517  di `=scalar(HH2)'
1518
1519
1520  forvalues i=2(2)2{
1521  forvalues y=1(1)10{
1522  scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
1523  di `=scalar(HH`y'dec`i')'
1524  }
1525  }
1526  forvalues i=2(2)2{
1527  sort thechosenOne`i'
1528  gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
1529  }
1530  forvalues i=2(2)2{
1531  gen dec`i'=1 if im0100==`i'
```

```
}
*defining deciles of the gross wealth distribution*
forvalues i=2(2)2{
replace dec`i'=2 if SHHthechosenOne`i'>`=scalar(HH1dec`i')' & SHHthechosenOne`i'<=`=scalar(HH2dec`i')' & im0100==`i'
replace dec`i'=3 if SHHthechosenOne`i'>`=scalar(HH2dec`i')' & SHHthechosenOne`i'<=`=scalar(HH3dec`i')' & im0100==`i'
replace dec`i'=4 if SHHthechosenOne`i'>`=scalar(HH3dec`i')' & SHHthechosenOne`i'<=`=scalar(HH4dec`i')' & im0100==`i'
replace dec`i'=5 if SHHthechosenOne`i'>`=scalar(HH4dec`i')' & SHHthechosenOne`i'<=`=scalar(HH5dec`i')' & im0100==`i'
replace dec`i'=6 if SHHthechosenOne`i'>`=scalar(HH5dec`i')' & SHHthechosenOne`i'<=`=scalar(HH6dec`i')' & im0100==`i'
replace dec`i'=7 if SHHthechosenOne`i'>`=scalar(HH6dec`i')' & SHHthechosenOne`i'<=`=scalar(HH7dec`i')' & im0100==`i'
replace dec`i'=8 if SHHthechosenOne`i'>`=scalar(HH7dec`i')' & SHHthechosenOne`i'<=`=scalar(HH8dec`i')' & im0100==`i'
replace dec`i'=9 if SHHthechosenOne`i'>`=scalar(HH8dec`i')' & SHHthechosenOne`i'<=`=scalar(HH9dec`i')' & im0100==`i'
replace dec`i'=10 if SHHthechosenOne`i'>`=scalar(HH9dec`i')' & SHHthechosenOne`i'<=`=scalar(HH10dec`i')' & im0100==`i'
}

gen lia = `=scalar(LIABadC_Eurostat)' /100* `=scalar(percentdec1liab2)' / (`=scalar(HH2)'/10) if dec2 == 1

replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab2)'/(`=scalar(HH2)'/10) if dec2 == 2
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab2)'/(`=scalar(HH2)'/10) if dec2 == 3
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab2)'/(`=scalar(HH2)'/10) if dec2 == 4
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab2)'/(`=scalar(HH2)'/10) if dec2 == 5
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab2)'/(`=scalar(HH2)'/10) if dec2 == 6
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab2)'/(`=scalar(HH2)'/10) if dec2 == 7
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab2)'/(`=scalar(HH2)'/10) if dec2 == 8
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab2)'/(`=scalar(HH2)'/10) if dec2 == 9
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab2)'/(`=scalar(HH2)'/10) if dec2 == 10

*litlle check if liabs add up to correct total*
gen liahw0010= lia*hw0010
egen totalliahw0010 = total(liahw0010)
sum totalliahw0010
di `=scalar(LIABadC_Eurostat)'

*creating the netw wealth variable and applying the 3 elasticities as weöö as tje simple tax schedule of 1% for weatlh above 1Mio EUR and 1.5% for wealth above 1.5 Mio EUR*

*gen netwealth*
gen netwealth = thechosenOne2 - lia
gen netwealthhw0010= netwealth*hw0010
egen totalnetwealthhw0010 = total(netwealthhw0010)
sum totalnetwealthhw0010
scalar totalnetwealthhw0010=r(mean)

di `=scalar(totalnetwealthhw0010)'

*finding out the number of all HH above the tax threshold of 1 Mio EUR*
egen HHabove1mio=total(hw0010) if netwealth>=1000000
sum HHabove1mio
scalar HHabove1mio=r(mean)
di HHabove1mio
```

```
di `=scalar(HHabove1mio)'

*finding out the number of all HH above the tax threshold of 5 Mio EUR*
egen HHabove1p5mio=total(hw0010) if netwealth>=5000000
sum HHabove1p5mio
scalar HHabove1p5mio=r(mean)
di HHabove1p5mio
di `=scalar(HHabove1p5mio)'

*tax base elasticity 1 = 6.9*
gen rev1 = (netwealth-1000000)/100*(100-6.9)*0.01 if netwealth>=1000000
replace rev1 = (netwealth-1000000)/100*(100-6.9-6.9/2)*0.015 if netwealth>=5000000
gen rev1hw0010 = rev1*hw0010

egen sumrev1hw0010=total(rev1hw0010)
sum sumrev1hw0010
scalar sumrev1hw0010=r(mean)

di `=scalar(sumrev1hw0010)'


*elasticity 2 = 11.3*
gen rev2 = (netwealth-1000000)/100*(100-11.3)*0.01 if netwealth>=1000000
replace rev2 = (netwealth-1000000)/100*(100-11.3-11.3/2)*0.015 if netwealth>=5000000
gen rev2hw0010 = rev2*hw0010

egen sumrev2hw0010=total(rev2hw0010)
sum sumrev2hw0010
scalar sumrev2hw0010=r(mean)

di `=scalar(sumrev2hw0010)'

*elasticity 3, 4, 5 ... just copy and paste*

*elasticity 3 = 35*
gen rev3 = netwealth/100*(100-35)*0.01 if netwealth>=1000000
replace rev3 = netwealth/100*(100-35-35/2)*0.015 if netwealth>=5000000
gen rev3hw0010 = rev3*hw0010

egen sumrev3hw0010=total(rev3hw0010)
sum sumrev3hw0010
scalar sumrev3hw0010=r(mean)

di `=scalar(sumrev3hw0010)'

*Ungleichheitsmaße*
pshare netwealth [pw=hw0010], p(10(10)90 95 99) gini
ereturn list
matrix a = e(b)
matrix b =e(G)
*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify

putexcel AK4=("implicate 2")
sleep 1000
putexcel AL4= matrix(a)
sleep 1000
putexcel AX4=("implicate 2")
sleep 1000
putexcel AY4= matrix(b)
sleep 1000


*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
putexcel AD4=`=scalar(sumrev1hw0010)'
sleep 1000
putexcel AE4=`=scalar(sumrev2hw0010)'
```

```
sleep 1000
putexcel AF4=`=scalar(sumrev3hw0010)'
sleep 1000
putexcel AG4=`=scalar(totalnetwealthhw0010)'
sleep 1000
putexcel AH4=`=scalar(HH2)'
sleep 1000
putexcel AI4=`=scalar(HHabove1mio)'
sleep 1000
putexcel AJ4=`=scalar(HHabove1p5mio)'
sleep 1000

clear
*************************************************************************
***** tax revenues implicate 3*******
use "${countryfile}\\${country}Original.dta"

*original*

keep if im0100==3
keep if thechosenOne3<`=scalar(xminthechosenOne3)' | thechosenOne3>=
`=scalar(least_rich_observation)'
gsort - thechosenOne3

*finding out the number of rich list observations*

egen HHForbes=total(hw0010) if thechosenOne3>=`=scalar(least_rich_observation)'
sum HHForbes
scalar HHForbes=r(mean)
di HHForbes
di `=scalar(HHForbes)'

keep thechosenOne3 hw0010 im0100
save "${countryfile}\\${country}Experiment.dta", replace
clear

*creating synt HH*

scalar HHxminthechosenOne3round = round(`=scalar(HHxminthechosenOne3)')
set obs `=scalar(HHxminthechosenOne3round)'
egen double thechosenOne3 = rndraw() , pareto(`=scalar(xminthechosenOne3)'
`=scalar(alphathechosenOne3)')
gen hw0010=1
gen im0100=3
gsort -thechosenOne3
gen n= _n


*subtracting the #number of richest individuals = #number of forbes observations, which are to be
replaced with the observations of the Forbes rich list*

keep if _n > `=scalar(HHForbes)'
append using "${countryfile}\\${country}Experiment.dta"
gsort -thechosenOne3

*finding out the number of all HH*
egen HH3=total(hw0010)
sum HH3
scalar HH3=r(mean)
di HH3
di `=scalar(HH3)'


forvalues i=3(3)3{
forvalues y=1(1)10{
scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
di `=scalar(HH`y'dec`i')'
}
```

```
1713  }
1714  forvalues i=3(3)3{
1715  sort thechosenOne`i'
1716  gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
1717  }
1718  forvalues i=3(3)3{
1719  gen dec`i'=1 if im0100==`i'
1720  }
1721  *defining deciles of the gross wealth distribution*
1722  forvalues i=3(3)3{
1723  replace dec`i'=2 if SHHthechosenOne`i'>`=scalar(HH1dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH2dec`i')' & im0100==`i'
1724  replace dec`i'=3 if SHHthechosenOne`i'>`=scalar(HH2dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH3dec`i')' & im0100==`i'
1725  replace dec`i'=4 if SHHthechosenOne`i'>`=scalar(HH3dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH4dec`i')' & im0100==`i'
1726  replace dec`i'=5 if SHHthechosenOne`i'>`=scalar(HH4dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH5dec`i')' & im0100==`i'
1727  replace dec`i'=6 if SHHthechosenOne`i'>`=scalar(HH5dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH6dec`i')' & im0100==`i'
1728  replace dec`i'=7 if SHHthechosenOne`i'>`=scalar(HH6dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH7dec`i')' & im0100==`i'
1729  replace dec`i'=8 if SHHthechosenOne`i'>`=scalar(HH7dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH8dec`i')' & im0100==`i'
1730  replace dec`i'=9 if SHHthechosenOne`i'>`=scalar(HH8dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH9dec`i')' & im0100==`i'
1731  replace dec`i'=10 if SHHthechosenOne`i'>`=scalar(HH9dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH10dec`i')' & im0100==`i'
1732  }
1733
1734  gen lia = `=scalar(LIABadC_Eurostat)' /100* `=scalar(percentdec1liab3)' / (`=scalar(HH2)'/10) if
      dec3 == 1
1735
1736  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 2
1737  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 3
1738  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 4
1739  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 5
1740  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 6
1741  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 7
1742  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 8
1743  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 9
1744  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab3)'/(`=scalar(HH2)'/10) if
      dec3 == 10
1745
1746  *litlle check if liabs add up to correct total*
1747  gen liahw0010= lia*hw0010
1748  egen totalliahw0010 = total(liahw0010)
1749  sum totalliahw0010
1750  di `=scalar(LIABadC_Eurostat)'
1751
1752  *creating the netw wealth variable and applying the 3 elasticities as weöö as tje simple tax
      schedule of 1% for weatlh above 1Mio EUR and 1.5% for wealth above 1.5 Mio EUR*
1753
1754  *gen netwealth*
1755  gen netwealth = thechosenOne3 - lia
1756  gen netwealthhw0010= netwealth*hw0010
1757  egen totalnetwealthhw0010 = total(netwealthhw0010)
1758  sum totalnetwealthhw0010
1759  scalar totalnetwealthhw0010=r(mean)
1760
```

```
di `=scalar(totalnetwealthhw0010)'

*finding out the number of all HH above the tax threshold of 1 Mio EUR*
egen HHabove1mio=total(hw0010) if netwealth>=1000000
sum HHabove1mio
scalar HHabove1mio=r(mean)
di HHabove1mio
di `=scalar(HHabove1mio)'

*finding out the number of all HH above the tax threshold of 5 Mio EUR*
egen HHabove1p5mio=total(hw0010) if netwealth>=5000000
sum HHabove1p5mio
scalar HHabove1p5mio=r(mean)
di HHabove1p5mio
di `=scalar(HHabove1p5mio)'

*tax base elasticity 1 = 6.9*
gen rev1 = (netwealth-1000000)/100*(100-6.9)*0.01 if netwealth>=1000000
replace rev1 = (netwealth-1000000)/100*(100-6.9-6.9/2)*0.015 if netwealth>=5000000
gen rev1hw0010 = rev1*hw0010

egen sumrev1hw0010=total(rev1hw0010)
sum sumrev1hw0010
scalar sumrev1hw0010=r(mean)

di `=scalar(sumrev1hw0010)'


*elasticity 2 = 11.3*
gen rev2 = (netwealth-1000000)/100*(100-11.3)*0.01 if netwealth>=1000000
replace rev2 = (netwealth-1000000)/100*(100-11.3-11.3/2)*0.015 if netwealth>=5000000
gen rev2hw0010 = rev2*hw0010

egen sumrev2hw0010=total(rev2hw0010)
sum sumrev2hw0010
scalar sumrev2hw0010=r(mean)

di `=scalar(sumrev2hw0010)'

*elasticity 3, 4, 5 ... just copy and paste*

*elasticity 3 = 35*
gen rev3 = netwealth/100*(100-35)*0.01 if netwealth>=1000000
replace rev3 = netwealth/100*(100-35-35/2)*0.015 if netwealth>=5000000
gen rev3hw0010 = rev3*hw0010

egen sumrev3hw0010=total(rev3hw0010)
sum sumrev3hw0010
scalar sumrev3hw0010=r(mean)

di `=scalar(sumrev3hw0010)'

*Ungleichheitsmaße*
pshare netwealth [pw=hw0010], p(10(10)90 95 99) gini
ereturn list
matrix a = e(b)
matrix b =e(G)
*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify

putexcel AK5=("implicate 3")
sleep 1000
putexcel AL5= matrix(a)
sleep 1000
putexcel AX5=("implicate 3")
sleep 1000
putexcel AY5= matrix(b)
sleep 1000
```

```
1829
1830
1831    *saving reveues in excel*
1832    putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
1833    putexcel AD5=`=scalar(sumrev1hw0010)'
1834    sleep 1000
1835    putexcel AE5=`=scalar(sumrev2hw0010)'
1836    sleep 1000
1837    putexcel AF5=`=scalar(sumrev3hw0010)'
1838    sleep 1000
1839    putexcel AG5=`=scalar(totalnetwealthhw0010)'
1840    sleep 1000
1841    putexcel AH5=`=scalar(HH3)'
1842    sleep 1000
1843    putexcel AI5=`=scalar(HHabove1mio)'
1844    sleep 1000
1845    putexcel AJ5=`=scalar(HHabove1p5mio)'
1846    sleep 1000
1847
1848    clear
1849
1850
1851    ********************************************************************************
1852    ***** tax revenues implicate 4*******
1853    use "${countryfile}\\${country}Original.dta"
1854
1855    *original*
1856
1857    keep if im0100==4
1858    keep if thechosenOne4<`=scalar(xminthechosenOne4)' | thechosenOne4>=
        `=scalar(least_rich_observation)'
1859    gsort - thechosenOne4
1860
1861    *finding out the number of rich list observations*
1862
1863    egen HHForbes=total(hw0010) if thechosenOne4>=`=scalar(least_rich_observation)'
1864    sum HHForbes
1865    scalar HHForbes=r(mean)
1866    di HHForbes
1867    di `=scalar(HHForbes)'
1868
1869    keep thechosenOne4 hw0010 im0100
1870    save "${countryfile}\\${country}Experiment.dta", replace
1871    clear
1872
1873    *creating synt HH*
1874
1875    scalar HHxminthechosenOne4round = round(`=scalar(HHxminthechosenOne4)')
1876    set obs `=scalar(HHxminthechosenOne4round)'
1877    egen double thechosenOne4 = rndraw() , pareto(`=scalar(xminthechosenOne4)'
        `=scalar(alphathechosenOne4)')
1878    gen hw0010=1
1879    gen im0100=4
1880    gsort -thechosenOne4
1881    gen n= _n
1882
1883
1884    *subtracting the #number of richest individuals = #number of forbes observations, which are to be
        replaced with the observations of the Forbes rich list*
1885
1886    keep if _n > `=scalar(HHForbes)'
1887    append using "${countryfile}\\${country}Experiment.dta"
1888    gsort -thechosenOne4
1889
1890    *finding out the number of all HH*
1891    egen HH4=total(hw0010)
1892    sum HH4
1893    scalar HH4=r(mean)
```

```
1894  di HH4
1895  di `=scalar(HH4)'
1896
1897
1898  forvalues i=4(4)4{
1899  forvalues y=1(1)10{
1900  scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
1901  di `=scalar(HH`y'dec`i')'
1902  }
1903  }
1904  forvalues i=4(4)4{
1905  sort thechosenOne`i'
1906  gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
1907  }
1908  forvalues i=4(4)4{
1909  gen dec`i'=1 if im0100==`i'
1910  }
1911  *defining deciles of the gross wealth distribution*
1912  forvalues i=4(4)4{
1913  replace dec`i'=2 if SHHthechosenOne`i'>`=scalar(HH1dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH2dec`i')' & im0100==`i'
1914  replace dec`i'=3 if SHHthechosenOne`i'>`=scalar(HH2dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH3dec`i')' & im0100==`i'
1915  replace dec`i'=4 if SHHthechosenOne`i'>`=scalar(HH3dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH4dec`i')' & im0100==`i'
1916  replace dec`i'=5 if SHHthechosenOne`i'>`=scalar(HH4dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH5dec`i')' & im0100==`i'
1917  replace dec`i'=6 if SHHthechosenOne`i'>`=scalar(HH5dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH6dec`i')' & im0100==`i'
1918  replace dec`i'=7 if SHHthechosenOne`i'>`=scalar(HH6dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH7dec`i')' & im0100==`i'
1919  replace dec`i'=8 if SHHthechosenOne`i'>`=scalar(HH7dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH8dec`i')' & im0100==`i'
1920  replace dec`i'=9 if SHHthechosenOne`i'>`=scalar(HH8dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH9dec`i')' & im0100==`i'
1921  replace dec`i'=10 if SHHthechosenOne`i'>`=scalar(HH9dec`i')' & SHHthechosenOne`i'<=
      `=scalar(HH10dec`i')' & im0100==`i'
1922  }
1923
1924  gen lia = `=scalar(LIABadC_Eurostat)' /100* `=scalar(percentdec1liab4)' / (`=scalar(HH4)'/10) if
      dec4 == 1
1925
1926  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 2
1927  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 3
1928  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 4
1929  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 5
1930  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 6
1931  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 7
1932  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 8
1933  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 9
1934  replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab4)'/(`=scalar(HH2)'/10) if
      dec4 == 10
1935
1936  *litlle check if liabs add up to correct total*
1937  gen liahw0010= lia*hw0010
1938  egen totalliahw0010 = total(liahw0010)
1939  sum totalliahw0010
1940  di `=scalar(LIABadC_Eurostat)'
1941
1942  *creating the netw wealth variable and applying the 3 elasticities as weöö as tje simple tax
```

```
    schedule of 1% for weatlh above 1Mio EUR and 1.5% for wealth above 1.5 Mio EUR*
1943
1944    *gen netwealth*
1945    gen netwealth = thechosenOne4 - lia
1946    gen netwealthhw0010= netwealth*hw0010
1947    egen totalnetwealthhw0010 = total(netwealthhw0010)
1948    sum totalnetwealthhw0010
1949    scalar totalnetwealthhw0010=r(mean)
1950
1951    di `=scalar(totalnetwealthhw0010)'
1952
1953    *finding out the number of all HH above the tax threshold of 1 Mio EUR*
1954    egen HHabove1mio=total(hw0010) if netwealth>=1000000
1955    sum HHabove1mio
1956    scalar HHabove1mio=r(mean)
1957    di HHabove1mio
1958    di `=scalar(HHabove1mio)'
1959
1960    *finding out the number of all HH above the tax threshold of 5 Mio EUR*
1961    egen HHabove1p5mio=total(hw0010) if netwealth>=5000000
1962    sum HHabove1p5mio
1963    scalar HHabove1p5mio=r(mean)
1964    di HHabove1p5mio
1965    di `=scalar(HHabove1p5mio)'
1966
1967    *tax base elasticity 1 = 6.9*
1968    gen rev1 = (netwealth-1000000)/100*(100-6.9)*0.01 if netwealth>=1000000
1969    replace rev1 = (netwealth-1000000)/100*(100-6.9-6.9/2)*0.015 if netwealth>=5000000
1970    gen rev1hw0010 = rev1*hw0010
1971
1972    egen sumrev1hw0010=total(rev1hw0010)
1973    sum sumrev1hw0010
1974    scalar sumrev1hw0010=r(mean)
1975
1976    di `=scalar(sumrev1hw0010)'
1977
1978
1979    *elasticity 2 = 11.3*
1980    gen rev2 = (netwealth-1000000)/100*(100-11.3)*0.01 if netwealth>=1000000
1981    replace rev2 = (netwealth-1000000)/100*(100-11.3-11.3/2)*0.015 if netwealth>=5000000
1982    gen rev2hw0010 = rev2*hw0010
1983
1984    egen sumrev2hw0010=total(rev2hw0010)
1985    sum sumrev2hw0010
1986    scalar sumrev2hw0010=r(mean)
1987
1988    di `=scalar(sumrev2hw0010)'
1989
1990    *elasticity 3, 4, 5 ... just copy and paste*
1991
1992    *elasticity 3 = 35*
1993    gen rev3 = netwealth/100*(100-35)*0.01 if netwealth>=1000000
1994    replace rev3 = netwealth/100*(100-35-35/2)*0.015 if netwealth>=5000000
1995    gen rev3hw0010 = rev3*hw0010
1996
1997    egen sumrev3hw0010=total(rev3hw0010)
1998    sum sumrev3hw0010
1999    scalar sumrev3hw0010=r(mean)
2000
2001    di `=scalar(sumrev3hw0010)'
2002
2003    *Ungleichheitsmaße*
2004    pshare netwealth [pw=hw0010], p(10(10)90 95 99) gini
2005    ereturn list
2006    matrix a = e(b)
2007    matrix b =e(G)
2008    *saving reveues in excel*
2009    putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
```

```
2010
2011    putexcel AK6=("implicate 4")
2012    sleep 1000
2013    putexcel AL6= matrix(a)
2014    sleep 1000
2015    putexcel AX6=("implicate 4")
2016    sleep 1000
2017    putexcel AY6= matrix(b)
2018    sleep 1000
2019
2020
2021    *saving reveues in excel*
2022    putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
2023    putexcel AD6=`=scalar(sumrev1hw0010)'
2024    sleep 1000
2025    putexcel AE6=`=scalar(sumrev2hw0010)'
2026    sleep 1000
2027    putexcel AF6=`=scalar(sumrev3hw0010)'
2028    sleep 1000
2029    putexcel AG6=`=scalar(totalnetwealthhw0010)'
2030    sleep 1000
2031    putexcel AH6=`=scalar(HH4)'
2032    sleep 1000
2033    putexcel AI6=`=scalar(HHabove1mio)'
2034    sleep 1000
2035    putexcel AJ6=`=scalar(HHabove1p5mio)'
2036    sleep 1000
2037
2038    clear
2039
2040
2041    ********************************************************************************
2042    ***** tax revenues implicate 5*******
2043    use "${countryfile}\\${country}Original.dta"
2044
2045    *original*
2046
2047    keep if im0100==5
2048    keep if thechosenOne5<`=scalar(xminthechosenOne5)' | thechosenOne5>=
       `=scalar(least_rich_observation)'
2049    gsort - thechosenOne5
2050
2051    *finding out the number of rich list observations*
2052
2053    egen HHForbes=total(hw0010) if thechosenOne5>=`=scalar(least_rich_observation)'
2054    sum HHForbes
2055    scalar HHForbes=r(mean)
2056    di HHForbes
2057    di `=scalar(HHForbes)'
2058
2059    keep thechosenOne5 hw0010 im0100
2060    save "${countryfile}\\${country}Experiment.dta", replace
2061    clear
2062
2063    *creating synt HH*
2064
2065    scalar HHxminthechosenOne5round = round(`=scalar(HHxminthechosenOne5)')
2066    set obs `=scalar(HHxminthechosenOne5round)'
2067    egen double thechosenOne5 = rndraw() , pareto(`=scalar(xminthechosenOne5)'
       `=scalar(alphathechosenOne5)')
2068    gen hw0010=1
2069    gen im0100=5
2070    gsort -thechosenOne5
2071    gen n= _n
2072
2073
2074    *subtracting the #number of richest individuals = #number of forbes observations, which are to be
       replaced with the observations of the Forbes rich list*
```

```
2075
2076    keep if _n > `=scalar(HHForbes)'
2077    append using "${countryfile}\\${country}Experiment.dta"
2078    gsort -thechosenOne5
2079
2080    *finding out the number of all HH*
2081    egen HH5=total(hw0010)
2082    sum HH5
2083    scalar HH5=r(mean)
2084    di HH5
2085    di `=scalar(HH5)'
2086
2087
2088    forvalues i=5(5)5{
2089    forvalues y=1(1)10{
2090    scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
2091    di `=scalar(HH`y'dec`i')'
2092    }
2093    }
2094    forvalues i=5(5)5{
2095    sort thechosenOne`i'
2096    gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
2097    }
2098    forvalues i=5(5)5{
2099    gen dec`i'=1 if im0100==`i'
2100    }
2101    *defining deciles of the gross wealth distribution*
2102    forvalues i=5(5)5{
2103    replace dec`i'=2 if SHHthechosenOne`i'>`=scalar(HH1dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH2dec`i')' & im0100==`i'
2104    replace dec`i'=3 if SHHthechosenOne`i'>`=scalar(HH2dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH3dec`i')' & im0100==`i'
2105    replace dec`i'=4 if SHHthechosenOne`i'>`=scalar(HH3dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH4dec`i')' & im0100==`i'
2106    replace dec`i'=5 if SHHthechosenOne`i'>`=scalar(HH4dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH5dec`i')' & im0100==`i'
2107    replace dec`i'=6 if SHHthechosenOne`i'>`=scalar(HH5dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH6dec`i')' & im0100==`i'
2108    replace dec`i'=7 if SHHthechosenOne`i'>`=scalar(HH6dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH7dec`i')' & im0100==`i'
2109    replace dec`i'=8 if SHHthechosenOne`i'>`=scalar(HH7dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH8dec`i')' & im0100==`i'
2110    replace dec`i'=9 if SHHthechosenOne`i'>`=scalar(HH8dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH9dec`i')' & im0100==`i'
2111    replace dec`i'=10 if SHHthechosenOne`i'>`=scalar(HH9dec`i')' & SHHthechosenOne`i'<=
        `=scalar(HH10dec`i')' & im0100==`i'
2112    }
2113
2114    gen lia = `=scalar(LIABadC_Eurostat)' /100* `=scalar(percentdec1liab5)' / (`=scalar(HH4)'/10) if
        dec5 == 1
2115
2116    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab5)'/(`=scalar(HH2)'/10) if
        dec5 == 2
2117    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab5)'/(`=scalar(HH2)'/10) if
        dec5 == 3
2118    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab5)'/(`=scalar(HH2)'/10) if
        dec5 == 4
2119    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab5)'/(`=scalar(HH2)'/10) if
        dec5 == 5
2120    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab5)'/(`=scalar(HH2)'/10) if
        dec5 == 6
2121    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab5)'/(`=scalar(HH2)'/10) if
        dec5 == 7
2122    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab5)'/(`=scalar(HH2)'/10) if
        dec5 == 8
2123    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab5)'/(`=scalar(HH2)'/10) if
        dec5 == 9
2124    replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab5)'/(`=scalar(HH2)'/10) if
```

```
    dec5 == 10
2125
2126    *litlle check if liabs add up to correct total*
2127    gen liahw0010= lia*hw0010
2128    egen totalliahw0010 = total(liahw0010)
2129    sum totalliahw0010
2130    di `=scalar(LIABadC_Eurostat)'
2131
2132    *creating the netw wealth variable and applying the 3 elasticities as weöö as tje simple tax
    schedule of 1% for weatlh above 1Mio EUR and 1.5% for wealth above 1.5 Mio EUR*
2133
2134    *gen netwealth*
2135    gen netwealth = thechosenOne5 - lia
2136    gen netwealthhw0010= netwealth*hw0010
2137    egen totalnetwealthhw0010 = total(netwealthhw0010)
2138    sum totalnetwealthhw0010
2139    scalar totalnetwealthhw0010=r(mean)
2140
2141    di `=scalar(totalnetwealthhw0010)'
2142
2143    *finding out the number of all HH above the tax threshold of 1 Mio EUR*
2144    egen HHabove1mio=total(hw0010) if netwealth>=1000000
2145    sum HHabove1mio
2146    scalar HHabove1mio=r(mean)
2147    di HHabove1mio
2148    di `=scalar(HHabove1mio)'
2149
2150    *finding out the number of all HH above the tax threshold of 5 Mio EUR*
2151    egen HHabove1p5mio=total(hw0010) if netwealth>=5000000
2152    sum HHabove1p5mio
2153    scalar HHabove1p5mio=r(mean)
2154    di HHabove1p5mio
2155    di `=scalar(HHabove1p5mio)'
2156
2157    *tax base elasticity 1 = 6.9*
2158    gen rev1 = (netwealth-1000000)/100*(100-6.9)*0.01 if netwealth>=1000000
2159    replace rev1 = (netwealth-1000000)/100*(100-6.9-6.9/2)*0.015 if netwealth>=5000000
2160    gen rev1hw0010 = rev1*hw0010
2161
2162    egen sumrev1hw0010=total(rev1hw0010)
2163    sum sumrev1hw0010
2164    scalar sumrev1hw0010=r(mean)
2165
2166    di `=scalar(sumrev1hw0010)'
2167
2168
2169    *elasticity 2 = 11.3*
2170    gen rev2 = (netwealth-1000000)/100*(100-11.3)*0.01 if netwealth>=1000000
2171    replace rev2 = (netwealth-1000000)/100*(100-11.3-11.3/2)*0.015 if netwealth>=5000000
2172    gen rev2hw0010 = rev2*hw0010
2173
2174    egen sumrev2hw0010=total(rev2hw0010)
2175    sum sumrev2hw0010
2176    scalar sumrev2hw0010=r(mean)
2177
2178    di `=scalar(sumrev2hw0010)'
2179
2180    *elasticity 3, 4, 5 ... just copy and paste*
2181
2182    *elasticity 3 = 35*
2183    gen rev3 = netwealth/100*(100-35)*0.01 if netwealth>=1000000
2184    replace rev3 = netwealth/100*(100-35-35/2)*0.015 if netwealth>=5000000
2185    gen rev3hw0010 = rev3*hw0010
2186
2187    egen sumrev3hw0010=total(rev3hw0010)
2188    sum sumrev3hw0010
2189    scalar sumrev3hw0010=r(mean)
2190
```

```
di `=scalar(sumrev3hw0010)'

*Ungleichheitsmaße*
pshare netwealth [pw=hw0010], p(10(10)90 95 99) gini
ereturn list
matrix a = e(b)
matrix b =e(G)
*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify

putexcel AK7=("implicate 5")
sleep 1000
putexcel AL7= matrix(a)
sleep 1000
putexcel AX7=("implicate 5")
sleep 1000
putexcel AY7= matrix(b)
sleep 1000


*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
putexcel AD7=`=scalar(sumrev1hw0010)'
sleep 1000
putexcel AE7=`=scalar(sumrev2hw0010)'
sleep 1000
putexcel AF7=`=scalar(sumrev3hw0010)'
sleep 1000
putexcel AG7=`=scalar(totalnetwealthhw0010)'
sleep 1000
putexcel AH7=`=scalar(HH5)'
sleep 1000
putexcel AI7=`=scalar(HHabove1mio)'
sleep 1000
putexcel AJ7=`=scalar(HHabove1p5mio)'
sleep 1000

clear
}
```