```


*ALLinOne*

*set (for the first time) the maximum of variables in Stata as the default value might be too low: set maxvar 120000, permanently*


*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. *


foreach a in Croatia Spain Austria Belgium Germany Estonia Finland France Greece Hungary Ireland Italy Lithuania Latvia Netherlands Poland Portugal Slovenia Slovakia {

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

global country = "`a'"
di "$country"


*attributing every HFCS country its respective country code (HFCS variable sa0100); ideally this is checked if an other/updated HFCS wave is used*
scalar Austria = 1
scalar Belgium = 2
scalar Cyprus = 3
scalar Germany = 4
scalar Estonia = 5
scalar Spain = 6
scalar Finland = 7
scalar France = 8
scalar Greece = 9
scalar Croatia = 10
scalar Hungary = 11
scalar Ireland = 12
scalar Italy = 13
scalar Lithuania = 14
scalar Luxembourg = 15
scalar Latvia = 16
scalar Malta = 17
scalar Netherlands = 18
scalar Poland = 19
scalar Portugal = 20
scalar Slovenia = 21
scalar Slovakia = 22

scalar countrycode = `=scalar(`a')'

*file path where you intend to store the original ECB merged data-set*
global hfcsdata "\\int.wsr.at\Nabu\restriktive_Daten\EZB\HFCS\net_wealth_tax_JBNSt\Daten\HFCS_UDB_3_2_STATA"

*file path where you intend to store STATA counrty-data-sets*
global countryfile "\\int.wsr.at\Nabu\restriktive_Daten\EZB\HFCS\net_wealth_tax_JBNSt\Daten\Newly_Created_Data"

*file path where you itend to store EXCEL files*
global excel "\\int.wsr.at\Nabu\Themen\net_wealth_tax_JBNSt\excel_files"

```

```

*importing financial balance sheet data*
import excel "${excel}\\Configuration", sheet("FBS") firstrow clear

keep if country == "${country}"

foreach var of varlist F21 - F100 {
sum `var'
scalar `var'=r(mean)
di `=scalar(`var')'
}

*importing interval in which the optimal adjustment factor "z" for financial wealth is to be found*
import excel "${excel}\\Configuration", sheet("interval_z_wmin") firstrow clear

keep if country == "${country}"

*lower bound interval for optimal "z"*
sum zLB
scalar zLB=r(mean)

*"step", zLB+1*
sum zstep
scalar zstep= r(mean)

*upper bound interval for optimal "z"*
sum zUB
scalar zUB= r(mean)

*lower bound interval for optimal wmin*
sum wminLB
scalar wminLB= r(mean)

*"step", e.g. wminLB+250k or wminLB+500k*
sum wminstep
scalar wminstep= r(mean)

*upper bound interval for optimal wmin
sum wminUB
scalar wminUB= r(mean)

*use the orignial HFCS data-set*
use "$hfcsdata\hfcs.dta", clear

*keeping only data from the respective country*
keep if sa0100==`=scalar(`a')'

keep hd1710 hd1920 da2100 da1140 da1000 hw0010 im0100 dl1000 hd0200 hd1010 sa0100

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

*calculate coverge ratios according to the concepts of Chakraborty et al.(2019)*

**************************NAIVE CONCEPT**************************************
*****************************************************************************

*first, get rid of missing values*
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 ==.)

*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
}
```

```

*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)*

forvalues i=1(1)5{
gen FAadC`i'=(da2100-hd1710-hd1920+da1140)*hw0010 if im0100==`i'
}

*find out the "adjusted" coverage ratio for all 5 implicates*
foreach var of varlist FAadC1 - FAadC5 {
sum `var'
scalar s_`var'=r(sum)
scalar Ratio`var'=s_`var'/(`=scalar(FAadC_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!*

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 wFAadC`i'`x'`y'=(da2100-hd1710-hd1920+da1140)/RatioFAadC`i'*(`x'.`y') if im0100==`i'
}
}
}

*export naive  and adjusted coverage ratios to excel table*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify


putexcel D2="RatioFAnaiveConcept"
sleep 1000
putexcel D3=`=scalar(RatioFAnaC1)'
sleep 1000
putexcel D4=`=scalar(RatioFAnaC2)'
sleep 1000
putexcel D5=`=scalar(RatioFAnaC3)'
sleep 1000
putexcel D6=`=scalar(RatioFAnaC4)'
sleep 1000
putexcel D7=`=scalar(RatioFAnaC5)'
sleep 1000

putexcel E2="RatioFAadjustedConcept"
sleep 1000
putexcel E3=`=scalar(RatioFAadC1)'
sleep 1000
putexcel E4=`=scalar(RatioFAadC2)'
sleep 1000
putexcel E5=`=scalar(RatioFAadC3)'
sleep 1000
putexcel E6=`=scalar(RatioFAadC4)'
sleep 1000
putexcel E7=`=scalar(RatioFAadC5)'
sleep 1000


*export HFCS financial wealth (adjusted Concept) to excel table*
putexcel F2="HFCS_FAadjustedConcept"
sleep 1000
putexcel F3=`=scalar(s_FAadC1)'
sleep 1000
putexcel F4=`=scalar(s_FAadC2)'
sleep 1000
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'
gen SWwFAadC`i'`z'`y'`x'=sum(hw0010) if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
}
}
}
}

*total sum of HH 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'
egen nWwFAadC`i'`z'`y'`x'=total(hw0010) if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
}
}
}
}

*ln(cumulative sum of HH weights/total sum HH 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'
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'
}
}
}
}

*ln(wealth/lower bound=xmin)*(-1)*
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'
gen xWwFAadC`i'`z'`y'`x'=ln(WwFAadC`i'`z'`y'/`x')*(-1) if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
}
}
}
}

*OLS; no constant*
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'
reg yWwFAadC`i'`z'`y'`x' xWwFAadC`i'`z'`y'`x' if im0100==`i', nocons
gen alpha`i'`z'`y'`x'=_b[x]
}
}
}
}

*generate fitted distribution*
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'
gen Ffit`i'`z'`y'`x'=(WwFAadC`i'`z'`y'/`x')^(-alpha`i'`z'`y'`x') if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
}
}
}
}
```

```

*generate empirical distribution*
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'
gen Femp`i'`z'`y'`x'=(SWwFAadC`i'`z'`y'`x'/nWwFAadC`i'`z'`y'`x') if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
}
}
}
}

*finding the (maximum) distance between fitted and empirical distribution=Kolmogorv-Smirnov; in STATA: ksmirnov=ks*
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'
ksmirnov Femp`i'`z'`y'`x'=Ffit`i'`z'`y'`x' if WwFAadC`i'`z'`y'>=`x' & im0100==`i'
gen ksWwFAadC`i'`z'`y'`x'=r(D)
scalar ksWwFAadC`i'`z'`y'`x'=r(D)
}
}
}
}

*optimal lowerbound=lb is the lower bound for which the fitted distribution has the smallest distance to the empirical distribution*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
egen lbWwFAadC`i'`z'`y'=rowmin( ksWwFAadC`i'`z'`y'`=scalar(wminLB)' - ksWwFAadC`i'`z'`y'`=scalar(wminUB)') if im0100==`i'
}
}
}

*creating the xmin variable*
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)' {
gen xminWwFAadC`i'`z'`y'`x'=`x' if lbWwFAadC`i'`z'`y'==ksWwFAadC`i'`z'`y'`x'
}
}
}
}

******housekeeping, getting rid of  variables with missing entries that were created in the procedure*****
dropmiss, force

*renaming the xmin variable, thus loosing the `x' in the name*
forvalues i=1(1)5{
forvalues y=0(1)9{
forvalues z=`=scalar(zLB)'(1)`=scalar(zUB)'{
rename xminWwFAadC`i'`z'`y' xminWwFAadC`i'`z'`y'
}
}
}

*creating the scalar xmin for future calculations*
forvalues i=1(1)5{
forvalues y=0(1)9{
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)
}
}

*lowerbound=lb*
forvalues i=1(1)5{
egen lbthechosenOne`i'=rowmin( ksthechosenOne`i'`=scalar(wminLB)' - ksthechosenOne`i'`=scalar(wminUB)') if im0100==`i'
}


*choosing xmin*
forvalues i=1(1)5{
foreach x of numlist `=scalar(wminLB)' `=scalar(wminstep)' to `=scalar(wminUB)' {
gen xminthechosenOne`i'`x'=`x' if lbthechosenOne`i'==ksthechosenOne`i'`x'
}
}

*housekeeping*
dropmiss, force

*renaming, thus loosing the `x' in the name*
forvalues i=1(1)5{
rename xminthechosenOne`i' xminthechosenOne`i'
}

*creating scalars for later calcualtios*
forvalues i=1(1)5{
global xminthechosenOne`i'=xminthechosenOne`i'
scalar xminthechosenOne`i'=xminthechosenOne`i'
di xminthechosenOne`i'
}

*estimating the pareto coefficient via OLS with no-constant as in Vermeulen(2014)*
forvalues i=1(1)5{
sum xminthechosenOne`i' if im0100==`i'
di r(mean)
scalar XMIN`i'=r(mean)
di XMIN`i'
gsort -thechosenOne`i'
gen SthechosenOne`i'=sum(hw0010) if thechosenOne`i'>=`=scalar(XMIN`i')' & im0100==`i'
egen nthechosenOne`i'=total(hw0010) if thechosenOne`i'>=`=scalar(XMIN`i')' & im0100==`i'
gen ythechosenOne`i'=ln(SthechosenOne`i'/nthechosenOne`i') if thechosenOne`i'>=`=scalar(XMIN`i')' & im0100==`i'
gen xthechosenOne`i'=ln(thechosenOne`i'/`=scalar(XMIN`i')')*(-1) if thechosenOne`i'>=`=scalar(XMIN`i')' & im0100==`i'
reg ythechosenOne`i' xthechosenOne`i' if im0100==`i' , nocons
scalar alphathechosenOne`i'=_b[x]
di alphathechosenOne`i'
gen alphathechosenOne`i'=alphathechosenOne`i'
}

*number HH above xmin*
forvalues i=1(1)5{
egen HHxminthechosenOne`i'=total(hw0010) if thechosenOne`i'>XMIN`i' & thechosenOne`i'<1000000000 & im0100==`i'
sum HHxminthechosenOne`i'
scalar HHxminthechosenOne`i'=r(mean)
di HHxminthechosenOne`i'
di `=scalar(HHxminthechosenOne`i')'
}


*Wealth in Tail based on "thechosenOne"*
forvalues i=1(1)5{
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
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')'
```

```
}

forvalues i=1(1)5{
forvalues y=1(1)10{
scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
di `=scalar(HH`y'dec`i')'
}
}

forvalues i=1(1)5{
sort thechosenOne`i'
gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
}

forvalues i=1(1)5{
gen dec`i'=1 if im0100==`i'
}

*defining deciles of the gross wealth distribution*
forvalues i=1(1)5{
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'
}

forvalues i=1(1)5{
forvalues y=1(1)10{
egen sumliabHW`i'dec`y'=total(liabHW`i') if im0100==`i' & dec`i'==`y'
sum sumliabHW`i'dec`y'
di r(mean)
scalar totalliab`i'dec`y'=r(mean)
}
}

forvalues i=1(1)5{
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')'
di `=scalar(check`i')'
}

forvalues i=1(1)5{
forvalues y=1(1)10{
scalar percentdec`y'liab`i'=`=scalar(totalliab`i'dec`y')'/(`=scalar(totalliab`i')'/100)
di `=scalar(percentdec`y'liab`i')'
}
}

*difference financial balace sheets and HFCS*
forvalues i=1(1)5{
scalar DiffLiab`i'=`=scalar(LIABadC_Eurostat)'-`=scalar(totalliab`i')'
}
```

```

*liability gap to be divided*
forvalues i=1(1)5{
forvalues y=1(1)10{
scalar LiabTBdivideddec`y'imp`i'=(`=scalar(DiffLiab`i')'/100)*percentdec`y'liab`i'
di `=scalar(LiabTBdivideddec`y'imp`i')'
}
}


*saving alpha, xmin, HH above xmin*
forvalues i=1(1)5{
sum alphathechosenOne`i'
di r(mean)
scalar alphathechosenOne`i'=r(mean)
sum HHxminthechosenOne`i'
di r(mean)
scalar HHxminthechosenOne`i'=r(mean)
sum xminthechosenOne`i'
di r(mean)
scalar xminthechosenOne`i'=r(mean)
}

*export alpha, xmin, HH above xmin to excel table*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify

putexcel A2=("alpha")
sleep 1000
putexcel A3=`=scalar(alphathechosenOne1)'
sleep 1000
putexcel A4=`=scalar(alphathechosenOne2)'
sleep 1000
putexcel A5=`=scalar(alphathechosenOne3)'
sleep 1000
putexcel A6=`=scalar(alphathechosenOne4)'
sleep 1000
putexcel A7=`=scalar(alphathechosenOne5)'
sleep 1000

putexcel B2=("w_min")
sleep 1000
putexcel B3=`=scalar(xminthechosenOne1)'
sleep 1000
putexcel B4=`=scalar(xminthechosenOne2)'
sleep 1000
putexcel B5=`=scalar(xminthechosenOne3)'
sleep 1000
putexcel B6=`=scalar(xminthechosenOne4)'
sleep 1000
putexcel B7=`=scalar(xminthechosenOne5)'
sleep 1000

putexcel C2=("HH_above_wmin")
sleep 1000
putexcel C3=`=scalar(HHxminthechosenOne1)'
sleep 1000
putexcel C4=`=scalar(HHxminthechosenOne2)'
sleep 1000
putexcel C5=`=scalar(HHxminthechosenOne3)'
sleep 1000
putexcel C6=`=scalar(HHxminthechosenOne4)'
sleep 1000
putexcel C7=`=scalar(HHxminthechosenOne5)'
sleep 1000

putexcel J2=("realassets_above_wmin")
sleep 1000
putexcel J3=`=scalar(totalRealadC1aboveXmin)'
```

```
sleep 1000
putexcel J4=`=scalar(totalRealadC2aboveXmin)'
sleep 1000
putexcel J5=`=scalar(totalRealadC3aboveXmin)'
sleep 1000
putexcel J6=`=scalar(totalRealadC4aboveXmin)'
sleep 1000
putexcel J7=`=scalar(totalRealadC5aboveXmin)'
sleep 1000

putexcel K2=("wealth_in_tail")
sleep 1000
putexcel K3=`=scalar(WTail1)'
sleep 1000
putexcel K4=`=scalar(WTail2)'
sleep 1000
putexcel K5=`=scalar(WTail3)'
sleep 1000
putexcel K6=`=scalar(WTail4)'
sleep 1000
putexcel K7=`=scalar(WTail5)'
sleep 1000

putexcel L2=("total_forbes_wealth")
sleep 1000
putexcel L3=`=scalar(totalWForbes1)'
sleep 1000
putexcel L4=`=scalar(totalWForbes2)'
sleep 1000
putexcel L5=`=scalar(totalWForbes3)'
sleep 1000
putexcel L6=`=scalar(totalWForbes4)'
sleep 1000
putexcel L7=`=scalar(totalWForbes5)'
sleep 1000

putexcel M2=("investment_in_nonselfemployedbusiness_below_xmin")
sleep 1000
putexcel M3=`=scalar(totalInvestNSE1)'
sleep 1000
putexcel M4=`=scalar(totalInvestNSE2)'
sleep 1000
putexcel M5=`=scalar(totalInvestNSE3)'
sleep 1000
putexcel M6=`=scalar(totalInvestNSE4)'
sleep 1000
putexcel M7=`=scalar(totalInvestNSE5)'
sleep 1000

putexcel N2=("investment_in_selfemployedbusiness_below_xmin")
sleep 1000
putexcel N3=`=scalar(totalInvestSE1)'
sleep 1000
putexcel N4=`=scalar(totalInvestSE2)'
sleep 1000
putexcel N5=`=scalar(totalInvestSE3)'
sleep 1000
putexcel N6=`=scalar(totalInvestSE4)'
sleep 1000
putexcel N7=`=scalar(totalInvestSE5)'
sleep 1000

putexcel O2=("total_realassets")
sleep 1000
putexcel O3=`=scalar(allRealadC1)'
sleep 1000
putexcel O4=`=scalar(allRealadC2)'
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
```

```


*implicate4*
putexcel W1=("percent_of_liab_in_grosswealth_decile")
sleep 1000
putexcel W2=("implicate4")
sleep 1000
putexcel W3=`=scalar(percentdec1liab4)'
sleep 1000
putexcel W4=`=scalar(percentdec2liab4)'
sleep 1000
putexcel W5=`=scalar(percentdec3liab4)'
sleep 1000
putexcel W6=`=scalar(percentdec4liab4)'
sleep 1000
putexcel W7=`=scalar(percentdec5liab4)'
sleep 1000
putexcel W8=`=scalar(percentdec6liab4)'
sleep 1000
putexcel W9=`=scalar(percentdec7liab4)'
sleep 1000
putexcel W10=`=scalar(percentdec8liab4)'
sleep 1000
putexcel W11=`=scalar(percentdec9liab4)'
sleep 1000
putexcel W12=`=scalar(percentdec10liab4)'
sleep 1000

putexcel W14=("missing_liabilities_€")
sleep 1000
putexcel W15=`=scalar(DiffLiab4)'
sleep 1000


*implicate5*
putexcel X1=("percent_of_liab_in_grosswealth_decile")
sleep 1000
putexcel X2=("implicate5")
sleep 1000
putexcel X3=`=scalar(percentdec1liab1)'
sleep 1000
putexcel X4=`=scalar(percentdec2liab1)'
sleep 1000
putexcel X5=`=scalar(percentdec3liab1)'
sleep 1000
putexcel X6=`=scalar(percentdec4liab1)'
sleep 1000
putexcel X7=`=scalar(percentdec5liab1)'
sleep 1000
putexcel X8=`=scalar(percentdec6liab1)'
sleep 1000
putexcel X9=`=scalar(percentdec7liab1)'
sleep 1000
putexcel X10=`=scalar(percentdec8liab1)'
sleep 1000
putexcel X11=`=scalar(percentdec9liab1)'
sleep 1000
putexcel X12=`=scalar(percentdec10liab1)'
sleep 1000

putexcel X14=("missing_liabilities_€")
sleep 1000
putexcel X15=`=scalar(DiffLiab5)'
sleep 1000

*average wealth in implicate-specific pareto tail*
putexcel Y2=("average_wealth_in_tail")
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'
di `=scalar(HH`y'dec`i')'
}
}
forvalues i=1(1)1{
sort thechosenOne`i'
gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
}
forvalues i=1(1)1{
gen dec`i'=1 if im0100==`i'
}
*defining deciles of the gross wealth distribution*
forvalues i=1(1)1{
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(percentdec1liab1)' / (`=scalar(HH1)'/10) if dec1 == 1

replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab1)'/(`=scalar(HH1)'/10) if dec1 == 2
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab1)'/(`=scalar(HH1)'/10) if dec1 == 3
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab1)'/(`=scalar(HH1)'/10) if dec1 == 4
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab1)'/(`=scalar(HH1)'/10) if dec1 == 5
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab1)'/(`=scalar(HH1)'/10) if dec1 == 6
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab1)'/(`=scalar(HH1)'/10) if dec1 == 7
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab1)'/(`=scalar(HH1)'/10) if dec1 == 8
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab1)'/(`=scalar(HH1)'/10) if dec1 == 9
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab1)'/(`=scalar(HH1)'/10) if dec1 == 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 = thechosenOne1 - 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(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
```

```
putexcel AB6=`=scalar(HHForbes)'
sleep 1000

clear


***************************************************************************
***** tax revenues implicate 2*******
use "${countryfile}\\${country}Original.dta"

*original*

keep if im0100==2
keep if thechosenOne2<`=scalar(xminthechosenOne2)' | thechosenOne2>=`=scalar(least_rich_observation)'
gsort - thechosenOne2

*finding out the number of rich list observations*

egen HHForbes=total(hw0010) if thechosenOne2>=`=scalar(least_rich_observation)'
sum HHForbes
scalar HHForbes=r(mean)
di HHForbes
di `=scalar(HHForbes)'

keep thechosenOne2 hw0010 im0100
save "${countryfile}\\${country}Experiment.dta", replace
clear

*creating synt HH*

scalar HHxminthechosenOne2round = round(`=scalar(HHxminthechosenOne2)')
set obs `=scalar(HHxminthechosenOne2round)'
egen double thechosenOne2 = rndraw() , pareto(`=scalar(xminthechosenOne2)' `=scalar(alphathechosenOne2)')
gen hw0010=1
gen im0100=2
gsort -thechosenOne2
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 -thechosenOne2

*finding out the number of all HH*
egen HH2=total(hw0010)
sum HH2
scalar HH2=r(mean)
di HH2
di `=scalar(HH2)'


forvalues i=2(2)2{
forvalues y=1(1)10{
scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
di `=scalar(HH`y'dec`i')'
}
}
forvalues i=2(2)2{
sort thechosenOne`i'
gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
}
forvalues i=2(2)2{
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')'
}
}
forvalues i=3(3)3{
sort thechosenOne`i'
gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
}
forvalues i=3(3)3{
gen dec`i'=1 if im0100==`i'
}
*defining deciles of the gross wealth distribution*
forvalues i=3(3)3{
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(percentdec1liab3)' / (`=scalar(HH2)'/10) if dec3 == 1

replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab3)'/(`=scalar(HH2)'/10) if dec3 == 2
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab3)'/(`=scalar(HH2)'/10) if dec3 == 3
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab3)'/(`=scalar(HH2)'/10) if dec3 == 4
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab3)'/(`=scalar(HH2)'/10) if dec3 == 5
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab3)'/(`=scalar(HH2)'/10) if dec3 == 6
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab3)'/(`=scalar(HH2)'/10) if dec3 == 7
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab3)'/(`=scalar(HH2)'/10) if dec3 == 8
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab3)'/(`=scalar(HH2)'/10) if dec3 == 9
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab3)'/(`=scalar(HH2)'/10) if dec3 == 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 = thechosenOne3 - 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 AK5=("implicate 3")
sleep 1000
putexcel AL5= matrix(a)
sleep 1000
putexcel AX5=("implicate 3")
sleep 1000
putexcel AY5= matrix(b)
sleep 1000
```

```


*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
putexcel AD5=`=scalar(sumrev1hw0010)'
sleep 1000
putexcel AE5=`=scalar(sumrev2hw0010)'
sleep 1000
putexcel AF5=`=scalar(sumrev3hw0010)'
sleep 1000
putexcel AG5=`=scalar(totalnetwealthhw0010)'
sleep 1000
putexcel AH5=`=scalar(HH3)'
sleep 1000
putexcel AI5=`=scalar(HHabove1mio)'
sleep 1000
putexcel AJ5=`=scalar(HHabove1p5mio)'
sleep 1000

clear


***************************************************************************
***** tax revenues implicate 4*******
use "${countryfile}\\${country}Original.dta"

*original*

keep if im0100==4
keep if thechosenOne4<`=scalar(xminthechosenOne4)' | thechosenOne4>=`=scalar(least_rich_observation)'
gsort - thechosenOne4

*finding out the number of rich list observations*

egen HHForbes=total(hw0010) if thechosenOne4>=`=scalar(least_rich_observation)'
sum HHForbes
scalar HHForbes=r(mean)
di HHForbes
di `=scalar(HHForbes)'

keep thechosenOne4 hw0010 im0100
save "${countryfile}\\${country}Experiment.dta", replace
clear

*creating synt HH*

scalar HHxminthechosenOne4round = round(`=scalar(HHxminthechosenOne4)')
set obs `=scalar(HHxminthechosenOne4round)'
egen double thechosenOne4 = rndraw() , pareto(`=scalar(xminthechosenOne4)' `=scalar(alphathechosenOne4)')
gen hw0010=1
gen im0100=4
gsort -thechosenOne4
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 -thechosenOne4

*finding out the number of all HH*
egen HH4=total(hw0010)
sum HH4
scalar HH4=r(mean)
```

```
di HH4
di `=scalar(HH4)'


forvalues i=4(4)4{
forvalues y=1(1)10{
scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
di `=scalar(HH`y'dec`i')'
}
}
forvalues i=4(4)4{
sort thechosenOne`i'
gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
}
forvalues i=4(4)4{
gen dec`i'=1 if im0100==`i'
}
*defining deciles of the gross wealth distribution*
forvalues i=4(4)4{
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(percentdec1liab4)' / (`=scalar(HH4)'/10) if dec4 == 1

replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab4)'/(`=scalar(HH2)'/10) if dec4 == 2
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab4)'/(`=scalar(HH2)'/10) if dec4 == 3
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab4)'/(`=scalar(HH2)'/10) if dec4 == 4
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab4)'/(`=scalar(HH2)'/10) if dec4 == 5
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab4)'/(`=scalar(HH2)'/10) if dec4 == 6
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab4)'/(`=scalar(HH2)'/10) if dec4 == 7
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab4)'/(`=scalar(HH2)'/10) if dec4 == 8
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab4)'/(`=scalar(HH2)'/10) if dec4 == 9
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab4)'/(`=scalar(HH2)'/10) if dec4 == 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 = thechosenOne4 - 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 AK6=("implicate 4")
sleep 1000
putexcel AL6= matrix(a)
sleep 1000
putexcel AX6=("implicate 4")
sleep 1000
putexcel AY6= matrix(b)
sleep 1000


*saving reveues in excel*
putexcel set "${excel}\Configuration.xlsx", sheet("output_${country}") modify
putexcel AD6=`=scalar(sumrev1hw0010)'
sleep 1000
putexcel AE6=`=scalar(sumrev2hw0010)'
sleep 1000
putexcel AF6=`=scalar(sumrev3hw0010)'
sleep 1000
putexcel AG6=`=scalar(totalnetwealthhw0010)'
sleep 1000
putexcel AH6=`=scalar(HH4)'
sleep 1000
putexcel AI6=`=scalar(HHabove1mio)'
sleep 1000
putexcel AJ6=`=scalar(HHabove1p5mio)'
sleep 1000

clear


***************************************************************************
***** tax revenues implicate 5*******
use "${countryfile}\\${country}Original.dta"

*original*

keep if im0100==5
keep if thechosenOne5<`=scalar(xminthechosenOne5)' | thechosenOne5>= `=scalar(least_rich_observation)'
gsort - thechosenOne5

*finding out the number of rich list observations*

egen HHForbes=total(hw0010) if thechosenOne5>=`=scalar(least_rich_observation)'
sum HHForbes
scalar HHForbes=r(mean)
di HHForbes
di `=scalar(HHForbes)'

keep thechosenOne5 hw0010 im0100
save "${countryfile}\\${country}Experiment.dta", replace
clear

*creating synt HH*

scalar HHxminthechosenOne5round = round(`=scalar(HHxminthechosenOne5)')
set obs `=scalar(HHxminthechosenOne5round)'
egen double thechosenOne5 = rndraw() , pareto(`=scalar(xminthechosenOne5)' `=scalar(alphathechosenOne5)')
gen hw0010=1
gen im0100=5
gsort -thechosenOne5
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 -thechosenOne5

*finding out the number of all HH*
egen HH5=total(hw0010)
sum HH5
scalar HH5=r(mean)
di HH5
di `=scalar(HH5)'


forvalues i=5(5)5{
forvalues y=1(1)10{
scalar HH`y'dec`i'=`=scalar(HH`i')'/10*`y'
di `=scalar(HH`y'dec`i')'
}
}
forvalues i=5(5)5{
sort thechosenOne`i'
gen SHHthechosenOne`i'=sum(hw0010) if im0100==`i'
}
forvalues i=5(5)5{
gen dec`i'=1 if im0100==`i'
}
*defining deciles of the gross wealth distribution*
forvalues i=5(5)5{
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(percentdec1liab5)' / (`=scalar(HH4)'/10) if
dec5 == 1

replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec2liab5)'/(`=scalar(HH2)'/10) if
dec5 == 2
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec3liab5)'/(`=scalar(HH2)'/10) if
dec5 == 3
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec4liab5)'/(`=scalar(HH2)'/10) if
dec5 == 4
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec5liab5)'/(`=scalar(HH2)'/10) if
dec5 == 5
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec6liab5)'/(`=scalar(HH2)'/10) if
dec5 == 6
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec7liab5)'/(`=scalar(HH2)'/10) if
dec5 == 7
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec8liab5)'/(`=scalar(HH2)'/10) if
dec5 == 8
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec9liab5)'/(`=scalar(HH2)'/10) if
dec5 == 9
replace lia = `=scalar(LIABadC_Eurostat)'/100*`=scalar(percentdec10liab5)'/(`=scalar(HH2)'/10) if
```

```
dec5 == 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 = thechosenOne5 - 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 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
}
```