Statistics/Data analysis

```
*******************************************************************************
* INCOME CONCEPTS: merging together
*******************************************************************************
clear
use "${simulationData}\05_${countryName}_${simulationName}_${dem_PY}.dta"

merge 1:1 hhID memberID using "${simulationData}\06_${countryName}_${simulationName}
> _${mkt_inc_PY}.dta", nogen

merge 1:1 hhID memberID using "${simulationData}\07_${countryName}_${simulationName}
> _${ssc_dir_tax_PY}.dta", nogen

merge 1:1 hhID memberID using "${simulationData}\08_${countryName}_${simulationName}
> _${pens_dir_trans_PY}.dta", nogen

merge m:1 hhID using "${simulationData}\09_${countryName}_${simulationName}_${indir_
> tax_PY}.dta", nogen

merge 1:1 hhID memberID using "${simulationData}\10_${countryName}_${simulationName}
> _${indir_subcidy_PY}.dta", nogen

merge 1:1 hhID memberID using "${simulationData}\11_${countryName}_${simulationName}
> _${inkind_trans_PY}.dta", nogen



global program_list
foreach aggregate in market_income $comp_list {
        foreach var in $`aggregate' {
                mvencode  `var', mv(0) override
                global program_list ${program_list} `var'
        }
        egen double `aggregate' = rowtotal(${`aggregate'})
}


label variable market_income "MARKET INCOME"


label variable SSC "SOCIAL CONTRIBUTIONS"
label variable direct_taxes "ALL DIRECT TAXES OTHER THAN SOCIAL CONTRIBUTIONS"

label variable pensions "CONTRIBUTORY PENSIONS"
label variable direct_transfers "DIRECT TRANSFERS AND NON-CONTRIBUTORY PENSIONS"

label variable indirect_taxes "INDIRECT TAXES"
label variable indirect_subsidies "INDIRECT SUBSIDIES"

*label variable health "INKIND HEALTH"
*label variable education "INKIND EDUCATION"

gen double net_market_income = market_income + direct_taxes + SSC
label variable net_market_income "NET MARKET INCOME"

gen double market_pens_income = market_income + pensions + SSC
label variable market_pens_income "MARKET INCOME PLUS PENSIONS"

gen double gross_income = market_income + direct_transfers + pensions
label variable gross_income "GROSS INCOME"
```

```
gen double disposable_income = market_income + direct_transfers + pensions + direct_
> taxes + SSC
label variable disposable_income "DISPOSABLE INCOME"

assert abs(disposable_income - (market_pens_income + direct_transfers + direct_taxes
> ))  < 10^(-9)
assert abs(disposable_income - (net_market_income + direct_transfers + pensions)) <
> 10^(-9)
assert disposable_income == gross_income + direct_taxes + SSC

gen double consumable_income = disposable_income + indirect_subsidies + indirect_tax
> es
label variable consumable_income "CONSUMABLE INCOME"

gen double final_income = consumable_income /*+ health + education*/
label variable final_income "FINAL INCOME"

*converting individual-level data to average per capita values
foreach var in $income_list {
    rename `var' `var'_ind
        bysort hhID: egen `var'_hh=total(`var'_ind)
        gen `var' = `var'_hh / hh_size
        assert  `var' >= 0
        drop `var'_ind `var'_hh
}

/* Delete the immediate 2 line codes below under "drop" since they are Yared's tempo
>  addition */
drop if missing(hhID)
drop if missing(memberID)

isid  hhID memberID
keep  hhID memberID ${dem_list} ${income_list} ${comp_list} ${program_list}
order hhID memberID ${dem_list} ${income_list} ${comp_list} ${program_list}

save "${postSimulationData}\12_${countryName}_${simulationName}_${output_inc_concept
> s}.dta", replace
```