Statistics/Data analysis

```
use "${simulationData}\01_${countryName}_${simulationName}_${dem_inc_SY}.dta" , clea
> r



*================================================================================
        * 1. Personal Income Tax (PIT)
*================================================================================

/* Adding all incomes that are subjected to PIT */
        global PIT_income_list $market_income


        gen double PIT_income_net = 0

        foreach var in $PIT_income_list {
                replace PIT_income_net = PIT_income_net  + `var'_net            // a
> dding all income components that are subjected to PIT
        }

gen double PIT_income_net_afterExemption = max(0, [PIT_income_net - ${PIT_deduction}
> ])     // income after PIT personal exemption
        gen PIT_exempted = PIT_income_net - PIT_income_net_afterExemption
        assert PIT_exempted >= 0 & PIT_exempted <= ${PIT_deduction}  if !missing(PIT
> _exempted)

/* Recover gross wage from net wage - this part is automoated by global options in d
> irtax.ado */
dirtax PIT_income_net_afterExemption, netinput rates(0 ${PIT_rate_lists}) tholds(0 $
> {PIT_cutoff_lists}) gen(PIT_income_gross_afterExemption)
        assert PIT_income_gross_afterExemption >= PIT_income_net_afterExemption

gen double PIT_0 = -1 * (PIT_income_gross_afterExemption - PIT_income_net_afterExemp
> tion)
gen double PIT_income_gross = PIT_income_gross_afterExemption + PIT_exempted
        assert (PIT_income_gross_afterExemption - PIT_income_net_afterExemption)>=0


/* Restore gross (after PIT) incomes from net proportionally to their contribution t
> o PIT base */
foreach var in $PIT_income_list {
        gen double `var' = `var'_net * PIT_income_gross / PIT_income_net
}

des PIT_0       PIT_income_net PIT_income_gross, full
sum PIT_0       PIT_income_net PIT_income_gross


*================================================================================
        * 2. Social Contributions (SIC): SIC rate in baseline is 20%
*================================================================================
replace wage = wage  / (1 - ${SIC_rate})                        // restore gross wag
> e out of net (of SIC)
gen double SIC_0 = -1*(wage * ${SIC_rate})
```

```
*=============================================================================
        * 3. Calculating net original market incomes (to be used later)
*=============================================================================
* putting the list of net market incomes (wage_net, entr_inc_net, etc) into a macro
> to automate the "egen" command right below it
local net_income_list
foreach z in $market_income {
        local net_income_list  `net_income_list' `z'_net
}
egen double net_market_income_orig = rowtotal(`net_income_list'), missing


*=============================================================================
        * 4. Save it
*=============================================================================
keep hhID memberID ${market_income} net_market_income_orig PIT_0 SIC_0
order hhID memberID ${market_income} net_market_income_orig PIT_0 SIC_0


mvencode ${market_income} net_market_income_orig, mv(0) override        // changing
> missing values to zero
isid hhID memberID

des, full
sum
save "${simulationData}\03_${countryName}_${simulationName}_${mkt_inc_SY}.dta", repl
> ace
```