```
* VAT indirect effect:
import excel using "$xls_tool", sheet(IO) first clear

/*
/*********** Putting sector names as value label to sector number *************/
tempvar sect_num
egen `sect_num' =  count(sector_name)
qui sum `sect_num'

local i = 0
forval z = 1(1)`r(N)' {
        local ++i
        local lbl = sector_name[`i']
        label define sect_name `i'"`lbl'", modify
}

label list sect_name
label val sector sect_name                                      // assigning value l
> abels
*********************************************************************************
*/

drop sector_name

isid sector
mvencode VAT_rate_PY VAT_exempt_PY sect_*, mv(0) override // make sure that none of
> the coefficient is missing
gen double dp = - VAT_rate_PY
gen fixed = 1 - VAT_exempt_PY // all except exempted sector
        assert dp == 0 if fixed == 0

costpush sect_*, fixed(fixed) price(dp) genptot(VAT_tot_eff_PY) genpind(VAT_ind_eff_
> PY) fix

keep sector VAT_ind_eff_PY
isid sector
tempfile ind_effect_PY
save `ind_effect_PY', replace

* Import rates (for direct effect)
import excel using "$xls_tool", sheet(VAT) first clear
replace VAT_rate_PY = - VAT_rate_PY
keep exp_type sector VAT_rate_PY
isid exp_type
tempfile rates_PY
save `rates_PY', replace


use "${simulationData}\06_${countryName}_${simulationName}_${mkt_inc_PY}.dta", clear
merge 1:1 hhID memberID using "${simulationData}\05_${countryName}_${simulationName}
> _${dem_PY}.dta", nogen assert(match) keepusing(hh_size)
merge 1:1 hhID memberID using "${simulationData}\07_${countryName}_${simulationName}
> _${ssc_dir_tax_PY}.dta", nogen assert(match)
merge 1:1 hhID memberID using "${simulationData}\08_${countryName}_${simulationName}
> _${pens_dir_trans_PY}.dta", nogen assert(match)

egen double disposable_income_orig = rowtotal(net_market_income_orig pens_trans_orig
> )
egen double disposable_income = rowtotal(${market_income} ${SSC} ${direct_taxes} ${p
> ensions} ${direct_transfers})

*su disposable_income_orig net_market_income_orig pens_trans_orig disposable_income
>  ${market_income} ${SSC} ${direct_taxes} ${pensions} ${direct_transfers}
```

```
recast int hh_size

        global exp_consistency_check = 0
        if $exp_consistency_check == 1 {
            assert abs(disposable_income - disposable_income_orig) < 10 ^ (-9)
        }
        assert disposable_income >= 0

collapse (sum) disposable_income_orig disposable_income (mean) hh_size, by(hhID)

isid hhID

merge 1:m hhID using "${simulationData}\04_${countryName}_${simulationName}_${exp_SY
> }.dta" , nogen /* assert(match) */

bysort hhID: egen double total_exp_SY = total(exp_gross_SY)

* total exp adjustment to make consistent with income: 'combined approach' - identif
> y hh, where incomes are lower than some reasonable level of dissaving (i.e. income
> s are less than 50% of expenditures) and normalize (scale down) the expenditures f
> or those. For these hh we assume 1:1 income to expenditure path through, while for
>  the rest of observations, we keep the original ratio between incomes and expendit
> ures we assume the path through from income to expenditures to equal the hh-specif
> ic apc.
gen double exp_net_adj_SY = exp_net_SY
        replace exp_net_adj_SY = exp_net_SY / total_exp_SY * disposable_income_orig
> if total_exp_SY > 2 * disposable_income_orig // We adjust the net expenditures, bu
> t the gross income should be consistent with disapobale
        bysort hhID: egen double total_exp_net_adj_SY = total(exp_net_adj_SY)

gen double exp_net_PY = exp_net_adj_SY / disposable_income_orig * disposable_income
> // normalization to diposable income to adjust for income-exp link
        replace exp_net_PY = 0 if disposable_income_orig == 0
        *assert !mi(exp_net_PY)

        if $exp_consistency_check == 1 {
                replace exp_net_PY = exp_net_SY
        }

merge m:1 exp_type using `rates_PY', nogen /*assert(match)*/
merge m:1 sector using `ind_effect_PY', nogen /*assert(match using) keep(match)*/

gen double exp_gross_PY = exp_net_PY  * (1 - exp_form * VAT_rate_PY) * (1 - VAT_ind_
> eff_PY)

        if $exp_consistency_check == 1 {
                assert abs(exp_gross_PY - exp_gross_SY) < 10 ^ (-10)
        }

gen double VAT = exp_net_PY - exp_gross_PY

* if we would like to separate the direct and indirect effect this can be done:
gen double VAT_dir = exp_net_PY  * exp_form * VAT_rate_PY
gen double VAT_ind = VAT - VAT_dir // the direct and indirect effects are rather cum
> ulative than additive, but for simplicity we can assume the additivity


foreach var in $indirect_taxes {
        replace `var' = `var' / hh_size
}
```

```
*isid hhID exp_type exp_form
collapse (sum) ${indirect_taxes}, by(hhID)
*groupfunction, sum(${indirect_taxes}) by(hhID) norestore
*isid hhID

foreach var in $indirect_taxes {
    assert `var' <= 10 ^ (-11) // they could be marginally positive due to rounding
> error
        replace `var' = 0 if `var' > 0
}


keep hhID ${indirect_taxes}
mvencode ${indirect_taxes}, mv(0) override

*isid hhID
save "${simulationData}\09_${countryName}_${simulationName}_${indir_tax_PY}.dta", re
> place
```