Statistics/Data analysis

```
//  Program:    05-01-dataPrep-expVars-foodAndNonfood.do
//  Task:       cleans household (food and non-food) consumption expenditures using
> Kenyan survey data (2015/16 KIHBS)
//  Project:    Kenya Fical MicroSim
//  Author:     Yared Seid - 2023Jan18


// #1: Food expenditure
// Loading the data
use "${dataSources}\Kenya_2015_16_KIHBS\food.dta", clear

// Generating vars

* Generating IDs
codebook clid hhid

qui tostring clid,      gen(clusterID) format(%04.0f)                    // cluster i
> d
qui tostring hhid,      gen(hhid_STR) format(%02.0f)                     // houdehold
>  id, within a cluster

label var clusterID "Cluster ID"

gen hhID = clusterID + hhid_STR
label var hhID "Household ID"

* dummy for expenditure catagory
gen exp_category = 1                                                 // 1= food expenditu
> re
label var exp_category "Expendidure category: 1 - food, 2 - non-food"
label define exp_category 1 "Food expenditure" 2 "Non-food expenditure"
label val exp_category exp_category

* Food expenditure
clonevar exp_type = item_code
label copy item_code exp_type                                        // copying value lab
> el to under the name "exp_type"
label val exp_type exp_type

clonevar exp_value = t04_amt                                                    // w
> eekly value

gen     place_purchase =.
replace place_purchase = 1 if (t05 ==1 | t05 ==2 | t05 ==3 | t05 ==4 | t05 ==5 | t05
>  ==6 | t05 ==7 | t05 ==8)
replace place_purchase = 2 if (t05 ==9 | t05 ==10)
replace place_purchase = 3 if t05 ==11
replace place_purchase = 4 if t05 ==12

label define place_purchase 1 "store, supermarket, kiosk, etc" 2 "hh sales, roadside
>  sales, etc" 3 "outside Kenya (not online)" 4 "others"
label val place_purchase place_purchase


*************************************************************************
/* Final touch of key variables - order matters! First annualize, then cap upper out
> liers */
local finalTouchVars exp_value

foreach z of local finalTouchVars {
ys_annualizing_values   `z', recall_in_days(7)          // annualizing weekly values
>
}
```

```
sum `finalTouchVars'
ys_upper_outliers              `finalTouchVars'                                     /
> 7 setting upper outliers to MEAN+3SD
sum `finalTouchVars'


        /* Graph it */
        foreach z of local finalTouchVars{
        sum `z'
        local `z'_Mean = round(`r(mean)')
        histogram `z', fraction normal ti("`z' - Annual (Mean = ``z'_Mean')")  xline
> (``z'_Mean')
        }
*******************************************************************************


* Saving the food data
local keepVars clid clusterID hhID  ///
                          exp_type exp_value place_purchase ///
                          exp_category
order `keepVars'
keep `keepVars'

sort  hhID

tempfile Exp_food
save `Exp_food', replace


                        ***********************************************
// #2: Non-food expenditure
// Loading the data
use "${dataSources}\Kenya_2015_16_KIHBS\nonfood.dta", clear

// Generating vars

* Generating IDs
codebook clid hhid

qui tostring clid,      gen(clusterID) format(%04.0f)                    // cluster i
> d
qui tostring hhid,      gen(hhid_STR) format(%02.0f)                     // houdehold
>  id, within a cluster

label var clusterID "Cluster ID"

gen hhID = clusterID + hhid_STR
label var hhID "Household ID"

* dummy for expenditure catagory
gen exp_category = 2                                    // 2= non-food expenditure
label var exp_category "Expendidure category: 1 - food, 2 - non-food"
label define exp_category 1 "Food expenditure" 2 "Non-food expenditure"
label val exp_category exp_category

* Non-food expenditure
clonevar exp_type = nf01
label copy nf2 exp_type                                                  // copying v
> alue label to under the name "exp_type"
label val exp_type exp_type
```

```
clonevar exp_value = nf04_amt                                                   // w
> eekly, monthly, quarterly, and annual values
label var exp_value "Amout paid in KSHS"

gen     place_purchase =.
replace place_purchase = 1 if (nf05 ==1 | nf05 ==2 | nf05 ==3 | nf05 ==4 | nf05 ==5
> | nf05 ==6 | nf05 ==7 | nf05 ==8)
replace place_purchase = 2 if (nf05 ==9 | nf05 ==10)
replace place_purchase = 3 if nf05 ==11
replace place_purchase = 4 if (nf05 ==12 |nf05 == 96)

label define place_purchase 1 "store, supermarket, kiosk, etc" 2 "hh sales, roadside
>  sales, etc" 3 "outside Kenya (not online)" 4 "others"
label val place_purchase place_purchase


*************************************************************************
/* Final touch of key variables - order matters! First annualize, then cap upper out
> liers */
local finalTouchVars exp_value

        foreach z of local finalTouchVars {
        ys_annualizing_values   `z' if recall ==1, recall_in_days(7)            // a
> nnualizing weekly values
        ys_annualizing_values   `z' if recall ==2, recall_in_days(30)           // a
> nnualizing monthly values
        ys_annualizing_values   `z' if recall ==3, recall_in_days(90)           // a
> nnualizing quarterly values
        }
*

sum `finalTouchVars'
ys_upper_outliers               `finalTouchVars'                                /
> / setting upper outliers to MEAN*3SD
sum `finalTouchVars'

        /* Graph it */
        foreach z of local finalTouchVars{
        sum `z'
        local `z'_Mean = round(`r(mean)')
        histogram `z', fraction normal ti("`z' - Annual (Mean = ``z'_Mean')")  xline
> (``z'_Mean')
        }
*************************************************************************

label define exp_type 1906 "Mobile phone airtime", modify               // this is f
> rom the questionnair since "1906" is not defined in the micro data

* Saving the food data
local keepVars clid clusterID hhID  ///
                        exp_type exp_value place_purchase ///
                        exp_category
order `keepVars'
keep `keepVars'

sort  hhID

tempfile Exp_nonfood
save `Exp_nonfood', replace
```

```
* Coping value labels from nonfood data for later use (when I'll append the non-food
>  data to food data)
local ys_label exp_type place_purchase exp_category
foreach z of local ys_label {
        tempfile `z'
        label save `z' using ``z''              // save value labels (whose names ar
> e saved in "ys_label" macro) in a temporary do file
}


// Appending expenditure datasets (i.e., food and non-food datasets)
use `Exp_food', clear
append using `Exp_nonfood'

* Updating value labels
local ys_label exp_type place_purchase exp_category
foreach z of local ys_label {
        qui do "``z''"                  // first call for the saved do file which ha
> s value labels saved in "ys_label" macro above; then, modify the existing value la
> bels saved in "ys_label" - i.e., adding the new value labels from non-food data to
>  the existing value labels from food data
}

* Capitalizing all the value labels
qui levelsof exp_type, local(val_exp_type)              // extracting the numbers of
>  a value label called "exp_type"

foreach z of local val_exp_type {
    local lbl: label exp_type `z'                       //      keeping in macro of
> each value lebel definition for each number of the value label through looping, e.
> g., lbl = one if label definition was 1 "one"
    local lbl = upper(`"`lbl'"')                        // making all letters in val
> ue definition in upper case
*       local lbl = upper(substr(`"`lbl'"', 1, 1)) + substr(`"`lbl'"', 2, .)    // m
> aking only the first letter in value definition in upper case (aka making value de
> finitions a "Title Case") --> this code assumes letters starting from the second l
> etter are in lower case in the original definition of values
    label define exp_type    `z'    `"`lbl'"', modify           // modifying the val
> ue label as defined in the immediate above codes
}


// #3
// Saving the data
*******************************************************************************
/* Dummy indicator for this specific data since this data has multiple expenditure i
> tems per hh --> Hence, "assert hh_size == hh_size_test" fails. */
gen data_multiple_exp   = 1
label var data_multiple_exp "Dummy for data with multiple exp info per hh: food and
> non-food exp"
*******************************************************************************

format exp_type %35.0f                  // to narrow down "exp_type" in data browse
> to only the first 35 carachters

local keepVars /*county*/ clid clusterID hhID  ///
                          exp_type exp_value place_purchase ///
                          exp_category data_multiple_exp

/* To assign Yes/No value lable to dummy variables */
local yesnoVars  data_multiple_exp
```

```
order `keepVars'
keep `keepVars'


sort  hhID exp_type
codebook hhID

* saving details
local dta_name           05-01-dataPrep-expVars-foodAndNonfood
local dta_note          "Expenditure: food and non-food variables"

include "${DataCleaningPath}\i-dta-savingDetails.do"    // saving the data with addi
> tional info on notes, labels, etc
*************************


*******************************************************************************
// #4
// Post-data-saving checking

/*
/* Checking for the hh size (we count only those who are present) */
merge m:1 hhID using 01-dataPrep-hhDemogVars, keepusing(hh_size)
>     // import hh size
drop _merge
merge m:m hhID using 02-dataPrep-hhMemberDemogVars, keepusing(hhID)             // i
> mport hh member id
drop _merge

bysort hhID: egen double hh_size_test = count(hhID)
        assert hh_size == hh_size_test // if !mi(ind_weight)            // FALSE, b/
> c of multiple expenditure types and associated values per hh

order hhID memberID hh_size hh_size_test
*/

*******************************************************************************
```