```
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
* CREATING an MPI IN STATA
* OPHI Training 2018
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------


* Clear information in the memory
clear all
set more off

* Change working folder
cd "C:\Dropbox\Maldives\Presentations\Example Stata"


* Save a record of everything
capture log close
log using log_stata.log, replace

* Open a dataset
use "pak_dhs12-13_sample.dta", clear


* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
* BUILDING THE DEPRIVATION MATRIX
* Each vector provides information about the condition of deprivation of each
* individual in an specific indicator. The mean of this vector shows the
* incidence of each deprivation on the total population
* (Uncensored Headcount Ratios)
```

```
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------


* ------------------------------------------------------------------------
* WATER
* ------------------------------------------------------------------------
/*
A household is non deprived if: it has piped water, public tap, borehole,
hand pump, protected well, protected spring, rainwater, filtration plant,
or bottled water

A household is deprived if: it gets water from an unprotected well,
unprotected spring; river/dam/stream/pond/canal; tanker truck,
cart with small tank, other
*/

lookfor  water
codebook water, tab(20)
recode   water (11/31=0)(32=1)(41=0)(42/43=1)(51=0)(61/62=1)(63/73=0)(96=1)(99=.),
gen(hh_d_water)
lab var  hh_d_water "Household deprived in Access to Safe Water"


* ------------------------------------------------------------------------
* ASSETS
* ------------------------------------------------------------------------
* A household is deprived if it has less than 2 small assets and no car

egen n_assets = rowtotal(radio television refrigerator bicycle motorbike), missing
tab n_assets [aw = weight], miss
```

```
gen hh_d_assets = (n_assets<2) if n_assets!=.
tab n_assets hh_d_assets [aw = weight]

* Using the car as veto
replace hh_d_assets = 0 if car==1
lab var hh_d_assets "Household deprived in Assets"
table n_assets car hh_d_assets [aw = weight]



* ------------------------------------------------------------------------
* SCHOOLING
* ------------------------------------------------------------------------
* A household is deprived if no member older than 15 has completed 5+ years of schooling

gen d_scho = (eduyears<5) if age>15 & age!=. & eduyears!=.
bys hh_id: egen hh_d_school = min(d_scho)
lab var hh_d_school "Household deprived in Years of Schooling"



* ------------------------------------------------------------------------
* NUTRITION
* ------------------------------------------------------------------------
* A household is deprived if any child under 5 with nutritional information is undernourished

sum z_scorewa [aw = weight]
gen d_nutrition = (z_scorewa<-2) if z_scorewa!=.
tab d_nutrition [aw = weight], miss
bys hh_id: egen hh_d_nutri = max(d_nutrition)
replace hh_d_nutri = 0 if no_child_eligible==1
lab var hh_d_nutri "Household deprived in Nutrition"
```

```
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
* RELEVANT SAMPLE
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------

* We construct a filter variable that identifies the observations with info for all relevant indicators
gen sample = (hh_d_water~=. & hh_d_assets~=. & hh_d_school~=. & hh_d_nutri~=.)
sum hh_d_water hh_d_assets hh_d_school hh_d_nutri [aw = weight] if sample==1



* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
* ANALYSIS OF THE MPI INDICATORS
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------


*** 1. MISSING VALUES


* Final check to see the total number of missing values we have for each variable
* Variables should not have high proportion of missing values at this stage
* The command might need to be installed: write "findit mdesc" in the command window, and install it

mdesc hh_d_school hh_d_nutri hh_d_water hh_d_assets
```

*** We keep only those observations with information for all relevant indicators and that are usual members of the household

keep if sample==1 & hv102==1

* --

* --

* UNCENSORED HEADCOUNT RATIOS

* --

* --

sum hh_d_school [aw = weight]

gen new_uncen_H_temp = r(mean)*100

foreach var in hh_d_school hh_d_nutri hh_d_water hh_d_assets {

sum `var' [aw = weight]

gen uncen_H_`var' = r(mean)*100

lab var uncen_H_`var' "Uncensored Headcount Ratio: Percentage of people who are deprived in …"

}

*

save "pak_dhs2012-13_cleaned.dta", replace

* --

```
* ------------------------------------------------------------------------
* SETTING WEIGHTS
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------

* Define vector 'w' of weights
* Change according to your specification. Remember the sum of weights MUST be
* equal to 1 or 100%

foreach var in hh_d_school hh_d_nutri  {

        gen       w_`var' = 1/3
        lab var w_`var' "Weight `var'"
        }
*

foreach var in hh_d_water hh_d_assets {

        gen       w_`var' = 1/6
        lab var w_`var' "Weight `var'"
        }
*



* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
* WEIGTHED DEPRIVATION MATRIX
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------

* The following commands multiply the deprivation matrix by the weight of each
```

```
* indicator.

foreach var in hh_d_school hh_d_nutri hh_d_water hh_d_assets {

        gen     g0_w_`var' = `var' * w_`var'
        lab var g0_w_`var' "Weigthed Deprivation of `var'"
        }
*



* ----------------------------------------------------------------------
* ----------------------------------------------------------------------
* COUNTING VECTOR
* ----------------------------------------------------------------------
* ----------------------------------------------------------------------

* Generate the vector of individual weighted deprivation score, 'c'

egen    c_vector = rowtotal(g0_w_*)
lab var c_vector "Counting Vector"
tab     c_vector [aw = weight], m



* ----------------------------------------------------------------------
* ----------------------------------------------------------------------
* INDENTIFICATION
* ----------------------------------------------------------------------
* ----------------------------------------------------------------------

* Using different poverty cut-offs (i.e. different k)
```

```
forvalue k = 10(10)100 {

        gen       multid_poor_`k' = (c_vector >= `k'/100)
        lab var multid_poor_`k' "Poverty Identification with k=`k'%"
        }
*



* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
* CENSORED COUNTING VECTOR
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------

* Generate the censored vector of individual weighted deprivation score, 'c(k)',
* providing a score of zero if a person is not poor

forvalue k = 10(10)100 {

        gen       cens_c_vector_`k' = c_vector
        replace cens_c_vector_`k' = 0 if multid_poor_`k'==0
        }
*



* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
* M0, H and A for all the possible cutoffs so far
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
```

```
* By sumarizing (obtaining the mean) of the identification vector, the individual deprivation share,

* and the individual censored c vector at any level of k we will obtain the Multidimensional
Headcount

* Ratio (H), the Intensity of Poverty among the Poor (A), and the Adjusted Headcount Ratio (M0),
respectively.


*** H ***

sum multid_poor_* [aw = weight], sep(15)


*** A ***

forvalue k = 10(10)100 {

        sum cens_c_vector_`k' if multid_poor_`k'==1 [aw = weight], sep(15)

        }


*** MPI ***

forvalue k = 10(10)100 {

        sum cens_c_vector_`k' [aw = weight], sep(15)

        }




* ----------------------------------------------------------------------------

* ----------------------------------------------------------------------------

* M0, H and A for k = 40%

* ----------------------------------------------------------------------------

* ----------------------------------------------------------------------------


/*( NOW WE CHOOSE A VALUE OF k )*/

local k = 40
```

```
* ------------------------------------------------------------------------
* CENSORED DEPRIVATION MATRIX
* ------------------------------------------------------------------------

foreach var in hh_d_school hh_d_nutri hh_d_water hh_d_assets {

        gen     g0_`k'_`var' = `var'
        replace g0_`k'_`var' = 0 if multid_poor_`k'==0
        }


* ------------------------------------------------------------------------
* HEADCOUNT/INCIDENCE OF MULTIDIMENSIONAL POVERTY FOR k = 40% (H)
* ------------------------------------------------------------------------

sum     multid_poor_`k' [aw = weight]
gen     H = r(mean)*100
lab var H "Headcount Ratio (H): % Population in multidimensional poverty"


* ------------------------------------------------------------------------
* INTENSITY OF POVERTY AMONG THE POOR FOR k = 40% (A)
* ------------------------------------------------------------------------

sum     cens_c_vector_`k' [aw = weight] if multid_poor_`k'==1
gen     A = r(mean)*100
lab var A  "Intensity of deprivation among the poor (A): Average % of weighted deprivations"


* ------------------------------------------------------------------------
* ADJUSTED HEADCOUNT RATIO (M0) FOR k = 40%
```

```
* ------------------------------------------------------------------------

sum      cens_c_vector_`k' [aw = weight]

gen      M0 = r(mean)

lab var M0 "Adjusted Headcount Ratio (M0 = H*A): Range 0 to 1"




* ------------------------------------------------------------------------

* ------------------------------------------------------------------------

* CENSORED HEADCOUNT RATIOS

* ------------------------------------------------------------------------

* ------------------------------------------------------------------------


* The Censored Headcount Ratio of an indicator is the proportion of the population

* that are poor AND deprived in that indicator.

* They can be obtained as the mean of each column of the censored deprivation matrix

local k = 40

foreach var in hh_d_school hh_d_nutri hh_d_water hh_d_assets {


        sum      g0_`k'_`var' [aw = weight]

        gen      cen_H_`var' = r(mean)*100

        lab var cen_H_`var'  "Censored Headcount Ratio: % of people who are poor and deprived in
…"

        }


fsum uncen_H_* cen_H_* [aw = weight]


sum cen_H_*


* ------------------------------------------------------------------------
```

```
* ------------------------------------------------------------------------
* DIMENSIONAL BREAKDDOWN: PERCENTAGE CONRIBUTIONS
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------

foreach var in hh_d_school hh_d_nutri hh_d_water hh_d_assets {

        gen       perc_cont_`var' = (cen_H_`var' * w_`var') / M0
        lab var perc_cont_`var' "Percentage contribution to M0"
        }

sum perc_cont_* [aw = weight], sep(15)




* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
* SUBGROUP DECOMPOSITION
* ------------------------------------------------------------------------
* ------------------------------------------------------------------------
local k = 40

* Uncensored Headcount Ratios by region
tabstat hh_d_* [aw = weight], by(region)
local k = 40

* Incidence of Poverty (H) by region
tabstat multid_poor_`k' [aw = weight], by(region)

* Intensity of Poverty (A) by region
tabstat cens_c_vector_`k' [aw = weight] if multid_poor_`k'==1, by(region)
```

```
* Adjusted Headcount Ratio (M0) by region
tabstat cens_c_vector_`k' [aw = weight], by(region)


local k=40
* Censored Headcount Ratios by region
foreach var in hh_d_school hh_d_nutri hh_d_water hh_d_assets {
	forvalue r = 1/6 {
		sum	g0_`k'_`var' [aw = weight] if region==`r'
		gen	cen_H_r`r'_`var' = r(mean)*100
		lab var cen_H_r`r'_`var'  "Censored Headcount Ratio - region `r'"
		}
	}

sum cen_H_r* [aw = weight], sep(6)


* Contributions by region

forvalue r = 1/6 {
	foreach var in hh_d_school hh_d_nutri hh_d_water hh_d_assets {
		sum	cens_c_vector_`k' [aw = weight] if region==`r'
		loc	M0_r`r' = r(mean)
		gen	perc_cont_r`r'_`var' = (cen_H_r`r'_`var' * w_`var') / `M0_r`r''
		lab var perc_cont_r`r'_`var' "Percentage contribution to M0 - region `r'"
		}
	}

sum perc_cont_r* [aw = weight], sep(7)
```

```
save "MyFirstMPI.dta", replace



* ----------------------------------------------------------------------------

* ----------------------------------------------------------------------------

* COLLAPSE RESULTS

* ----------------------------------------------------------------------------

* ----------------------------------------------------------------------------

forvalues k=10(10)100{

gen A_`k' = cens_c_vector_`k' if multid_poor_`k'==1

}


collapse A_* uncen_H_* w_* c_vector multid_poor_* cens_c_vector_* cen_H_* perc_cont_* [aw = weight], by(region)


save "Collapsed_results by region.dta", replace



* ----------------------------------------------------------------------------

* ----------------------------------------------------------------------------

* POVERTY MAPS

* ----------------------------------------------------------------------------

* ----------------------------------------------------------------------------


/*

1. To obtain the shapefiles go to http:*www.diva-gis.org/gdata and download the data for your country.

Then, se the following command to transform the shapefile in a stata file:

shp2dta using XXX_adm1, database(region) coordinates(map) genid(id) gencentroids(center)
```

```
In the new dataset called "region" check the id for each region (br id NAME_1 VARNAME_1).
In the collapsed results dataset generate a new variable "id" following the structure of the dataset
"regions.dta"
*/


cd "C:\TanzaniaHBS2012\sdr_subnational_boundaries_2018-05-11\shps"

ssc install shp2dta

ssc install spmap

shp2dta using sdr_subnational_boundaries, database(tz_database) coordinates(tz_coord)

* ------------------------------------------------------------------------------
* INCIDENCE OF POVERTY (H)
* ------------------------------------------------------------------------------
ta multid_poor_40

cd "C:\Dropbox\Maldives\Presentations\Example Stata"

spmap multid_poor_40 using Pakistan_coord.dta, id(region) clmethod(unique) fcolor(Reds)
        ///
legend(pos(5) subtitle("Headcount Ratio", size(vsmall))) ///
title("Incidence of Multidimensional Poverty H" "in Pakistan, k=40%")

gr export Incidence_Pakistan.emf, replace


* ------------------------------------------------------------------------------
* MULTIDIMENSIONAL POVERTY INDEX (M0)
```

```
* ------------------------------------------------------------------------------
ta cens_c_vector_40

spmap cens_c_vector_40 using pak_c.dta, id(region) clmethod(unique) fcolor(Reds)        ///
legend(pos(5) subtitle("M0", size(vsmall))) line(data("pak_c.dta"))                     ///
title("Multidimensional Poverty in Pakistan (MPI), k = 40%")

gr export M0_Pakistan.emf, replace

*/

* ------------------------------------------------------------------------------
* ------------------------------------------------------------------------------
* ROBUSTNESS, SENSITIVITY AND STANDARD ERRORS
* ------------------------------------------------------------------------------
* ------------------------------------------------------------------------------

* We open the clean dataset

clear
use "MyFirstMPI.dta"


* ------------------------------------------------------------------------------
* RANK ROBUSTNESS COMPARISONS
* ------------------------------------------------------------------------------

* Robustness tests are based on the coefficient of rank correlations Kendall tau-b, which
* measures the association between pairs, given the position that each takes when results are
* sorted using different poverty indices.
* These different poverty indices can be obtained changing the weights of indicators or the
```

* povetry cut-off (k).

* a. Variations in weights: several MPI are computed keeping dimensions/indicators and deprivations

* cut-offs unchanged; only the weights are modified. Once all the MPI have been computed, figures by

* subnational regions can be obtained and regions ranked. The Kendall tau-b coefficient can then be

* computed over the rankings.

* b. Variations in the poverty cut-offs (k): several MPI are computed keeping the structure unchanged

* and also adjusting the k-value. Once all the MPI have been computed, figures by subnational regions

* can be obtained and regions ranked. The Kendall tau-b coefficient can then be

* computed over the rankings.

```
forvalues k = 10(10)100 {

        gen H_`k' = .

        forvalues r = 1/7 {

                sum multid_poor_`k' [aw = weight] if region==`r'
                replace H_`k' = r(mean)*100 if region==`r'
                }
        }

ktau H_10 H_20 H_30 H_40 H_50 H_60 H_70 H_80 H_90 H_100, stats(taub score se p)
```

```
* ------------------------------------------------------------------------
* STANDARD ERRORS
* ------------------------------------------------------------------------

* We open the clean dataset
clear
use "MyFirstMPI.dta"

* Set the characteristics of the survey
svyset psu [pw = weight], strata(strata)



* Incidence of Poverty (H)
* For details and discussions see equations (8.13) and (8.31), chapter 8 of OPHI book

forvalue k = 10(10)100 {

        svy: mean multid_poor_`k'
        gen se_H_`k' = (_se[multid_poor_`k'])
        gen lb_H_`k' = _b[multid_poor_`k'] - 1.96 * se_H_`k'
        gen ub_H_`k' = _b[multid_poor_`k'] + 1.96 * se_H_`k'
        }

sum multid_poor_* lb_H_* ub_H_* [aw = weight]



* Adjusted Headcount Ratio (M0)
* For details and discussions see equations (8.11) and (8.30), chapter 8 of OPHI book

forvalue k = 10(10)100 {
```

```
        svy: mean cens_c_vector_`k'
        gen se_M0_`k' = (_se[cens_c_vector_`k'])
        gen lb_M0_`k' = _b[cens_c_vector_`k'] - 1.96 * se_M0_`k'
        gen ub_M0_`k' = _b[cens_c_vector_`k'] + 1.96 * se_M0_`k'
        }

sum cens_c_vector_* lb_M0_* ub_M0_* [aw = weight]



* Average Deprivation among the Poor (A)
* For details and discussions see equations (8.19), (8.35) and (8.36), chapter 8 of OPHI book

forvalue k = 10(10)100 {

        svy: mean multid_poor_`k' cens_c_vector_`k'
        mat cov = e(V)
        loc cov = cov[2,1]
        loc var_H = cov[1,1]
        loc var_MPI = cov[2,2]

        gen se_A_`k' = ((`var_MPI'/_b[multid_poor_`k']^2) +
(((_b[cens_c_vector_`k']/_b[multid_poor_`k']^2)^2)*(`var_H')) ///
        - 2*((_b[cens_c_vector_`k']/_b[multid_poor_`k']^3)*`cov'))^0.5

        gen lb_A_`k' = (_b[cens_c_vector_`k']/_b[multid_poor_`k']) - 1.96 * se_A_`k'
        gen ub_A_`k' = (_b[cens_c_vector_`k']/_b[multid_poor_`k']) + 1.96 * se_A_`k'
        }



* ------------------------------------------------------------------------
```

```
* DOMINANCE AMONG SUBNATIONAL REGIONS
* ------------------------------------------------------------------------

* For M0
collapse mean cens_c_vector_10 cens_c_vector_20 cens_c_vector_30 cens_c_vector_40          ///
                cens_c_vector_50 cens_c_vector_60 cens_c_vector_70 cens_c_vector_80     ///
                cens_c_vector_90 cens_c_vector_100 [aw = weight], by(region)

reshape long cens_c_vector_, i(region) j(k)

gen cens_c_vector_1 = cens_c_vector_ if region==1
label var cens_c_vector_1 "Balochistan"
gen cens_c_vector_2 = cens_c_vector_ if region==2
label var cens_c_vector_2 "Islamabad (ICT)"
gen cens_c_vector_3 = cens_c_vector_ if region==3
label var cens_c_vector_3 "Khyber Pakhtunkhawa"
gen cens_c_vector_4 = cens_c_vector_ if region==4
label var cens_c_vector_4 "Gilgit Baltistan"
gen cens_c_vector_5 = cens_c_vector_ if region==5
label var cens_c_vector_5 "Punjab"
gen cens_c_vector_6 = cens_c_vector_ if region==6
label var cens_c_vector_6 "Sindh"

graph twoway line cens_c_vector_1 k || line cens_c_vector_2 k || line cens_c_vector_3 k || line
cens_c_vector_4 k || line cens_c_vector_5 k || line cens_c_vector_6 k


* For H
clear
use "MyFirstMPI.dta"
```

```
collapse mean multid_poor_10 multid_poor_20 multid_poor_30 multid_poor_40 multid_poor_50 multid_poor_60 ///
        multid_poor_70 multid_poor_80 multid_poor_90 multid_poor_100 [aw = weight], by(region)

reshape long multid_poor_, i(region) j(k)

gen multid_poor_1 = multid_poor_ if region==1
label var multid_poor_1 "Balochistan"
gen multid_poor_2 = multid_poor_ if region==2
label var multid_poor_2 "Islamabad (ICT)"
gen multid_poor_3 = multid_poor_ if region==3
label var multid_poor_3 "Khyber Pakhtunkhawa"
gen multid_poor_4 = multid_poor_ if region==4
label var multid_poor_4 "Gilgit Baltistan"
gen multid_poor_5 = multid_poor_ if region==5
label var multid_poor_5 "Punjab"
gen multid_poor_6 = multid_poor_ if region==6
label var multid_poor_6 "Sindh"

graph twoway line multid_poor_1 k || line multid_poor_2 k || line multid_poor_3 k || line multid_poor_4 k || line multid_poor_5 k || line multid_poor_6 k

* ------------------------------------------------------------------------
* TEST OF DIFFERENCE BETWEEN REGIONS (e.g. Balochistan and Sindh, M0)
* ------------------------------------------------------------------------

clear
use "MyFirstMPI.dta"
```

```
svyset psu [pw = weight], strata(strata)


svy: mean cens_c_vector_30, over(region)
test _b[Balochistan] = _b[Sindh]




* ------------------------------------------------------------------------------
* NOTES
* ------------------------------------------------------------------------------

* For discussions regarding statistical tests, please see Sections 8.2.2 and 8.2.3 of OPHI Book
* For Robustness and dominance analysis with Statistical inference, see Section 8.3 of OPHI Book.
* For those who are interested in bootstrap, see the Appendix of Chapter 8 of the OPHI Book.
```