

Changes to the SPSS Tabulation Syntax (Jan. 17 2007)

Anthropometry

Due to a mistake in the data entry program, the anthropometric Z scores were miscalculated in a few cases in each country. To correct this it is necessary to re-calculate the Z-scores (variables HAP, HAZ, HAM, WAP, WAZ, WAM, WHP, WHZ, WHM, FLAG). To do this there are 4 steps:

- 1) Output selected variables related to anthropometry from the SPSS CH.SAV data file into an ASCII file (AN.DAT). Program: anthro_output.sps
- 2) Run CSPro program to calculate revised Z-scores and output these in ANNEW.DAT. This will also give a listing of cases that are modified. Program: Anthro fix.bch
- 3) Run ANNEW.SPS to read the new data into SPSS (ANNEW.SPS is generated by Anthro fix.bch).
- 4) Import resulting new Z-scores into the CH.SAV file. Program: anthro_import.sps

Place the two SPSS files into the “C:\MICS\SPSS” directory, and place the CSPro programs into “C:\MICS\Anthro fix”.

Steps:

- 1) The first step (anthro_output.sps) should run without any problems as it is, providing all variables exist in your dataset.
- 2) After running the first step, open “Anthro fix.bch” in CSPro, click on the Dict tab in the tree on the left, and then double click “Anthro_check record”. Check that the start location and length of all variables in the CSPro dictionary match EXACTLY the start locations and length printed in the output in SPSS of the previous step. Make changes in the CSPro dictionary to correct the start location and length of each variable, starting from the first variable. Once these are corrected, click on the pencil icon (Edits) to switch back to showing the logic of the application, and then click the traffic light icon (Run) to execute the application. The default names and locations for the files being used should all be correct if the files have been placed in the directories recommended above. The program will print a list of all cases for which the values of the Z-scores are changed (if any), as well as generating a new data file (ANNEW.DAT)
- 3) The previous step generated ANNEW.SPS. Run this program in SPSS to load the ANNEW.DAT files into SPSS.
- 4) Run “Anthro_import.sps” to compare the new Z-scores with the old Z-scores in SPSS, print a list of differences, replace the old Z-scores with the new ones, and save the file CH.SAV with these new Z-scores.

NU.1

Changed working table to add in more detail about cases excluded from analysis

NU.5

changed
recode S11 (7 = 100) (else = 0) into nosalt.
to
recode S11 (**6** = 100) (else = 0) into nosalt.

changed
recode S11 (7=0) (1,2 = 1) (3 = 2) (else = sysmis) into iodized.
to
recode S11 (**6**=0) (1,2 = 1) (3 = 2) (else = sysmis) into iodized.

NU.8

The following changes are made to the program to avoid warnings for division by zero. They do not affect the results in the table, but are important to avoid problems when later transferring results into DevInfo:

changed
compute prop2500 = tot2500/tweighed.
to
if (tweighed > 0) prop2500 = tot2500/tweighed.

CHRECVAC.SPS

Modifications were made to correct for vaccinations reported as DK or missing that were included in the denominator of the calculations but should be left out of the denominator. The list of modifications is too long to show here. Use the revised CHRECVAC.SPS and modify according to the vaccination schedule used in your survey questionnaire.

CH.6

changed
to
op + rf + tr + ot + any + atotal
to
op + rf + **sh** + tr + ot + any + atotal

add to /Statistics:
mean(sh (f5.1) ")

EN.1, EN.2, EN.5, EN.5w, EN.7

Tables run now based on household data file (weighted by total household members and sample weight) and not household member file. This has no effect on the resulting table, but more accurately reflects the fact that the data was collected at the household level. A similar change is made in the sampling error calculation and this change will affect the design effect calculated in the sampling errors.

changed
get file = 'hl.sav'.

to
get file = 'hh.sav'.
select if (HH9 = 1).
compute hhweight = hhweight*HH11.

EN.2 changed “driking” to “**drinking**”

EN.9, EN.10

changed
if ((fnatur = 100 and poor = 100) or narrow = 100 or fourhaz = 100)
to
if ((fnatur = 100 and **tworep** = 100) or narrow = 100 or fourhaz = 100) .

changed in EN.9 (four times)
/spoor = means(poor)
to
/spoor = means(**tworep**)

RH.6

The following changes are made to the program to avoid warnings for division by zero. They do not affect the results in the table, but are important to avoid problems when later transferring results into DevInfo:

changed
compute liferisk = matdeath / unitrisk.
to
if (unitrisk > 0) liferisk = matdeath / unitrisk.

changed
compute pcomatdth = 100 * matdeath / deadsis.

to
if (deadsis > 0) pcmatdth = 100 * matdeath / deadsis.

ED.5

The following changes are made to the program to avoid warnings for division by zero. They do not affect the results in the table, but are important to avoid problems when later transferring results into DevInfo:

changed
compute y1 = (z12/(z12+z10))*100.
compute y2 = (z23/(z23+z20))*100.
compute y3 = (z34/(z34+z30))*100.
compute y4 = (z45/(z45+z40))*100.
to
if (z12+z10 > 0) y1 = (z12/(z12+z10))*100.
if (z23+z20 > 0) y2 = (z23/(z23+z20))*100.
if (z34+z30 > 0) y3 = (z34/(z34+z30))*100.
if (z45+z40 > 0) y4 = (z45/(z45+z40))*100.

ED.6

The following changes are made to the program to avoid warnings for division by zero. They do not affect the results in the table, but are important to avoid problems when later transferring results into DevInfo:

changed
compute npscr = (nplgrade/npageg)*100.
to
if (npageg > 0) npscr = (nplgrade/npageg)*100.

changed
compute trse = (ncass/nlgps)*100.
to
if (nlgps > 0) trse = (ncass/nlgps)*100.

ED.7

The following changes are made to the program to avoid warnings for division by zero. They do not affect the results in the table, but are important to avoid problems when later transferring results into DevInfo:

Changed
compute pgirls = (nnarf/ngirlsp)*100.

variable label pgirls "Primary school net attendance ratio (NAR), girls".

compute pboys = (nnarbp/nboysp)*100.

variable label pboys "Primary school net attendance ratio (NAR), boys".

compute ratio1 = (pgirls/pboys).

variable label ratio1 "Gender parity index (GPI) for primary school NAR**".

compute sgirls = (nnarfs/ngirlss)*100.

variable label sgirls "Secondary school net attendance ratio (NAR), girls".

compute sboys = (nnarbs/nboys)*100.

variable label sboys "Secondary school net attendance ratio (NAR), boys".

compute ratio2 = (sgirls/sboys).

variable label ratio2 "Gender parity index (GPI) for secondary school NAR**".

To

if (ngirlsp > 0) pgirls = (nnarfp/ngirlsp)*100.

variable label pgirls "Primary school net attendance ratio (NAR), girls".

if (nboysp > 0) pboys = (nnarbp/nboysp)*100.

variable label pboys "Primary school net attendance ratio (NAR), boys".

if (pboys > 0) ratio1 = (pgirls/pboys).

variable label ratio1 "Gender parity index (GPI) for primary school NAR**".

if (ngirlss > 0) sgirls = (nnarfs/ngirlss)*100.

variable label sgirls "Secondary school net attendance ratio (NAR), girls".

if (nboys > 0) sboys = (nnarbs/nboys)*100.

variable label sboys "Secondary school net attendance ratio (NAR), boys".

if (sboys > 0) ratio2 = (sgirls/sboys).

variable label ratio2 "Gender parity index (GPI) for secondary school NAR**".

HA.12

changed

/table = HL4 + HH7 + HH6 + wlthind5 +HC1B+ tot1 by
tot2 + tot3 + tot4 + tot5 + ratio1 + tot6 + tot7 + tot8 + tot9 + ratio2 + tot10
to

/table = HL4 + HH7 + HH6 + wlthind5 +HC1B+ **total** by
tot2 + tot3 + tot4 + tot5 + ratio1 + tot6 + tot7 + tot8 + tot9 + ratio2 + tot10