
Subject: Using igrowup SPSS syntax
Posted by [BLM](#) on Fri, 22 Mar 2019 23:14:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Please help! I merged DHS PR and KR data and have been trying to get the prevalence for stunting, underweight, and wasting in children using the 2015-2016 Malawi data. I keep getting error whenever I run the igrowup syntax provided by WHO. Once ran, I do not get any data in the variables assessing stunting, wasting etc. What am I doing wrong? Please see the syntax below. Thank you!!!

```
set header on.  
set printback=on length=None Width=132.
```

```
get file="C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav".
```

```
title "ZPR_KIDS using WHO Child Growth Standards".
```

```
* Typical DHS Household variables.
```

```
compute casenum=$casenum.
```

```
compute weight2=hc2/10.
```

```
compute sex=hc27.
```

```
compute agemos=hc1.
```

```
compute length=hc3/10.
```

```
compute lorh=-99.
```

```
missing values lorh (-99).
```

```
* Note that although DHS has a variable (hc15) to measure lying or standing height(lorh).
```

```
* DHS regards it as unreliable, so we will use age, or else length, to determine this.
```

```
string oedema (A1).
```

```
Compute oedema='n'.
```

```
* delete the above if oedema is in dataset.
```

```
* oedema is not in DHS datasets as of this writing.
```

```
COMPUTE wgting = hv005/1000000.
```

```
COMMENT * using RENAME allows value labels to transfer.
```

```
RENAME VARIABLES (hv024=REGION).
```

```
RENAME VARIABLES (hv025=URBANR).
```

```
*
```

```
missing values hc16 (98,99).
```

```
* if day of birth is missing, assumes the 15th.
```

```
if (missing(hc16)) hc16=15.
```

```
* compute days from day interviewed - day born.
```

```
compute agedays3=(yrmoda(hc19,hc18,hc17)-yrmoda(hc31,hc30,hc16)).
```

```
* if negative age, impute it from months old.
```

```
if((yrmoda(hc19,hc18,hc17)-yrmoda(hc31,hc30,hc16))<0) agedays3=agemos*30.4375.
```

```
*
```

```
* if missing daysold, impute from months old.
```

```
if(missing(agedays3) )agedays3=agemos*30.4375.
```

```
* agedays3 will be used later to compute exact month cutoffs, and should not be rounded.
```

```

*
.
compute agedays2=rnd(agedays3).
compute dateborn=(date.dmy(hc16,hc30,hc31)).
compute datevis=(date.dmy(hc17,hc18,hc19)).
formats dateborn, datevis(adate12).
*
.
* HV103 is "slept last night at house" 1=yes.
* HV103 adjusts for orphans.
* hc13 is "reason not measured" 0=measured.
*
.
select if hv103=1.
select if hc13=0.

* ===== Length & Height adjustments=====
* take what they call length or height and create one lenhei2 variable _____.
* lenhei2 is the proper length/height measurement adjusted for.

* set all lengths as ok and then change those that are not ok.
compute lenhei2=length.
compute useLngth=-99.
missing values useLngth (-99).
if (missing(lenhei2)) useLngth=-99.
* if standing under 2 or lying over 2.

if (lorh=2 and agedays2<731) lenhei2=length+.7.
if (lorh=1 and agedays2>=731) lenhei2=length-.7.
if (lorh=2 and agedays2<731)useLngth=1.
if (lorh=1 and agedays2>=731) useLngth=2.
if (lorh=1 and agedays2<731)useLngth=1.
if (lorh=2 and agedays2>=731) useLngth=2.

* if missing the recumbent indicator but have age, we assume they have it right.
if (missing(lorh) and agedays2<731) useLngth=1.
if (missing(lorh) and agedays2>=731) useLngth=2.
if (missing(lorh) and missing(agedays2) and agemos<24) useLngth=1.
if (missing(lorh) and missing(agedays2) and agemos>=24) useLngth=2.

if (missing(agedays2) and lorh=2) useLngth=2.
if (missing(agedays2) and lorh=1) useLngth=1.

* if age missing and indicator missing, use length of child to figure.
if (missing(agedays2) and missing(lorh) and length<87) useLngth=1.
if (missing(agedays2) and missing(lorh) and length>=87) useLngth=2.

* ===== check inputs and create outputs for dbf file ANTHRO check =====.
frequencies variables=sex, lorh.
descriptives variables=weight2, length, lenhei2,agedays2,agedays3,agemos /statistics=mean
stddev min max.

```

```

* if not done already, need age in days to compare to norms.

*next 3 statements keep only what we need (+ bit more, 61 mo), sort, save, and drop unneeded
vars.
select if (agedays2 < (61*30.4375) or missing(agedays2)).
*sort to get ready to match to haz norm chart.
sort cases by sex, agedays2.

xsave outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDSNEW.sav"/keep casenum,
hv001,hv002,hvidx, wgting, region,urbanr,
weight2,sex,agemos,length,lorh, oedema,
lenhei2,uselngth,agedays2,agedays3,dateborn,datevis, ha33.

execute.

* ===== HAZ =====.

* lookup z formula for Height for Age using l, m, s variables.
MATCH FILES FILE="C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav"
/table="C:\Users\Dr. Love\Desktop\ZPR_KIDSNEW\hazlms.sav" /by sex, agedays2.
execute.
compute zhaz=(((lenhei2/m)**l)-1)/(s**l).
xsave outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDSNEW.sav" /drop = l,m,s.
execute.

* ===== BMI Z =====.

* lookup z formula for BMI for Age using l, m, s variables.
MATCH FILES FILE="C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav" /table="C:\Users\Dr.
Love\Desktop\ZPR_KIDS\BMllms.sav" /by sex, agedays2.
execute.
* make sure weight has been given right var name.
compute BMI= weight2*10000/(lenhei2*lenhei2).
compute zBMI=(((BMI/m)**l)-1)/(s**l).
compute sd3pos=m*((1+l*s**3)**(1/l)).
compute sd23pos=sd3pos- m*((1+l*s**2)**(1/l)).
if (zBMI>3) zBMI=3+((BMI-sd3pos)/sd23pos).

compute sd3neg=m*((1+l*s*(-3))**(1/l)).
compute sd23neg= m*((1+l*s*(-2))**(1/l))-sd3neg.
if (zBMI<-3) zBMI=(-3)-((sd3neg-BMI)/sd23neg).

if (oedema='y' or oedema='Y') zbmi=-4.44.
* this makes sure oedemas aren't put in the high or low prevalence categories.
* but are still counted for denominator.
* zbmi = -4.44 with oedema will be adjusted later.

```

xsave outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav" /drop = l,m,s.
execute.

* ===== WAZ =====.

* lookup z formula for Weight to Age using l, m, s variables.

sort cases by sex, agedays2.

MATCH FILES FILE="C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav" /table="C:\Users\Dr. Love\Desktop\ZPR_KIDS\wazlms.sav" /by sex, agedays2.

execute.

compute zwaz=(((weight2/m)**l)-1)/(s*l).

compute sd3pos=m*((1+l*s*3)**(1/l)).

compute sd23pos=sd3pos- m*((1+l*s*2)**(1/l)).

if (zwaz>3) zwaz=3+((weight2-sd3pos)/sd23pos).

compute sd3neg=m*((1+l*s*(-3))**(1/l)).

compute sd23neg= m*((1+l*s*(-2))**(1/l))-sd3neg.

if (zwaz<-3) zwaz=(-3)-((sd3neg-weight2)/sd23neg).

if (oedema='y' or oedema='Y') zwaz=-4.44.

if (missing(agedays2) and (oedema='y' or oedema='Y')) zwaz=-4.44.

* this allows oedemas to be set at < -3 sd for prevalences.

* then set to missing later before calculating means.

* zwaz is set to missing later.

execute.

* ===== WHZ using length =====.

* ===== interpolate length =====.

* determine if length is of 2 decimals significance.

* since the LMS charts are only to one, such as 87 point 2.

* the interp variable =1 if we need to interpolate.

if(uselength=1)length2=lenhei2.

compute interp=0.

if (abs(length2 - (rnd(length2*10))/10) >.001)interp=1.

* if it does then interpolate on lms table.

* first we will find the lower level on LMS chart.

DO IF interp=1.

compute lenlow=trunc(length2*10)/10.

ELSE IF interp=0.

compute lenlow=rnd(length2*10)/10.

* above makes double sure it will match lms chart.

END IF.

* next we will find the upper level on LMS chart.

DO IF interp=1.

```

compute lenhigh=rnd((length2+.05)*10)/10.
ELSE IF interp=0.
  compute lenhigh=rnd(length2*10)/10.
END IF.

```

```

* we will be doing many calculations with length2 .
* so we will remember the original values here.
compute length9=length2.

```

```

* next we will get the LMS numbers for lenlow and lenhigh.
* =====lenlow LMS calculations=====

```

```

compute length2=lenlow.
sort cases by sex, length2.
xsave outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav" /drop = l,m,s.
execute.

```

```

* lookup z formula for Weight for Length using l, m, s variables.
MATCH FILES FILE="C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav" /table="C:\Users\Dr.
Love\Desktop\ZPR_KIDS\wflms.sav" /by sex, length2.
execute.

```

```

* do the lengths--> zscore for lenlow where ll=lenlow.
compute zwhzll=(((weight2/m)**l)-1)/(s*l).
compute sd3pos=m*((1+l*s*3)**(1/l)).
compute sd23pos=sd3pos- m*((1+l*s*2)**(1/l)).
if (zwhzll>3) zwhzll=3+((weight2-sd3pos)/sd23pos).

```

```

compute sd3neg=m*((1+l*s*(-3))**(1/l)).
compute sd23neg= m*((1+l*s*(-2))**(1/l))-sd3neg.
if (zwhzll<-3) zwhzll=-3-((sd3neg-weight2)/sd23neg).

```

```

execute.

```

```

* =====lenhigh LMS calculations=====

```

```

compute length2=lenhigh.
sort cases by sex, length2.
xsave outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav" /drop = l,m,s.
execute.

```

```

* lookup z formula for Weight for Length using l, m, s variables.
MATCH FILES FILE="C:\Users\Dr. Love\Desktop\ZPR_KIDS\tempx.sav" /table="C:\Users\Dr.
Love\Desktop\ZPR_KIDS\wflms.sav" /by sex, length2.
execute.

```

```

* do the lengths--> zscore for lenhigh where lh=lenhigh.
compute zwhzlh=(((weight2/m)**l)-1)/(s*l).
compute sd3pos=m*((1+l*s*3)**(1/l)).
compute sd23pos=sd3pos- m*((1+l*s*2)**(1/l)).
if (zwhzlh>3) zwhzlh=3+((weight2-sd3pos)/sd23pos).

```

```

compute sd3neg=m*((1+l*s*(-3))**(1/l)).
compute sd23neg= m*((1+l*s*(-2))**(1/l))-sd3neg.
if (zwhzlh<-3) zwhzlh=-3-((sd3neg-weight2)/sd23neg).

```

```

* ===== calculations for height used =====.
* use criteria for when height used, height2 is height properly measured.
if (uselngth=2) height2=lenhei2.
* we will be doing many calculations with height2 .
* so we will remember the original value here.
compute height9=height2.

* determine if height is of 2 decimals significance.
* since the LMS charts are only to one, such as 87 point 2.
* the interph variable =1 if we need to interpolate.
compute interph=0.
.
if (abs(height2 - (rnd(height2*10))/10) >.001)interph=1.
* if it does then interpolate on lms table.
* first we will find the lower level on LMS chart.
DO IF interph=1.
  compute hgtlow=trunc(height2*10)/10.
ELSE IF interph=0.
  compute hgtlow=rnd(height2*10)/10.
END IF.
* next we will find the upper level on LMS chart.
DO IF interph=1.
  compute hgthigh=rnd((height2+.05)*10)/10.
ELSE IF interph=0.
  compute hgthigh=rnd(height2*10)/10.
END IF.
* =====WHZ using hgtlow =====.
compute height2=hgtlow.
sort cases by sex, height2.
xsave outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDS\tempxx.sav" /drop = l,m,s.
execute.

```

```

* lookup z formula for Weight for Height using l, m, s variables.

```

```

MATCH FILES FILE="C:\Users\Dr. Love\Desktop\ZPR_KIDS\tempxx.sav" /table="C:\Users\Dr.
Love\Desktop\ZPR_KIDS\wfhms.sav" /by sex, height2.
execute.

```

```

* do the heights where hl=hgtlow.
if (height2>0) zwhzhl=(((weight2/m)**l)-1)/(s*l).
if (height2>0) sd3pos=m*((1+l*s*3)**(1/l)).
if (height2>0) sd23pos=sd3pos- m*((1+l*s*2)**(1/l)).
if (height2>0 and zwhzhl>3) zwhzhl=3+((weight2-sd3pos)/sd23pos).

if (height2>0) sd3neg=m*((1+l*s*(-3))**(1/l)).
if (height2>0) sd23neg= m*((1+l*s*(-2))**(1/l))-sd3neg.
if (height2>0 and zwhzhl<-3) zwhzhl=(-3)-((sd3neg-weight2)/sd23neg).

```

```
* =====WHZ using hgthigh =====.
compute height2=hgthigh.
sort cases by sex, height2.
xsave outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDS\tempx.sav" /drop = l,m,s.
execute.
```

```
* lookup z formula for Weight for Height using l, m, s variables.
```

```
MATCH FILES FILE="C:\Users\Dr. Love\Desktop\ZPR_KIDS\tempx.sav" /table="C:\Users\Dr.
Love\Desktop\ZPR_KIDS\wfhims.sav" /by sex, height2.
execute.
```

```
* do the heights where hh=hgthigh.
if (height2>0) zwhzhh=(((weight2/m)**l)-1)/(s*l).
if (height2>0) sd3pos=m*((1+l*s*3)**(1/l)).
if (height2>0) sd23pos=sd3pos- m*((1+l*s*2)**(1/l)).
if (height2>0 and zwhzhh>3) zwhzhh=3+((weight2-sd3pos)/sd23pos).
```

```
if (height2>0) sd3neg=m*((1+l*s*(-3))**(1/l)).
if (height2>0) sd23neg= m*((1+l*s*(-2))**(1/l))-sd3neg.
if (height2>0 and zwhzhh<-3) zwhzhh=(-3)-((sd3neg-weight2)/sd23neg).
```

```
* =====now do interpolation and choose length or height =====.
```

```
* length9 is somewhere between lenlow and lenhigh.
```

```
* find the ratios with #s like 52,20 52,26 and 52,30.
```

```
compute aboveh=length9-lenlow.
```

```
compute ratiol=aboveh/.1.
```

```
* note that the greater the length, the less the z.
```

```
compute zwhz= zwhzll-((zwhzll-zwhzlh)*ratiol).
```

```
* now for height.
```

```
* height is defined only if useLngth=2 and.
```

```
* will replace the length calculations if defined.
```

```
compute aboveh=height9-hgtlow.
```

```
compute ratioh=aboveh/.1.
```

```
* note that the greater the height, the less the z.
```

```
if (useLngth=2) zwhz= zwhzhl-((zwhzhl-zwhzhh)*ratioh).
```

```
if (oedema='y' or oedema='Y') zwhz=-4.44.
```

```
xsave outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDS\tempxx.sav" /drop = l,m,s.
```

```
execute.
```

```
* ===== convert Z scores to 2 decimals =====.
```

```
compute zwhz=(rnd(zwhz*100))/100.
```

```
compute zhaz=(rnd(zhaz*100))/100.
```

```
compute zwaz=(rnd(zwaz*100))/100.
```

```
compute zbmi=(rnd(zbmi*100))/100.
```

```
* ===== Flags for missing values =====.
```

```
compute whzflag=0.
```

```
compute hazflag=0.
```

compute wazflag=0.
compute bmiflag=0.
compute whznoage=0.

if (zwhz < -5 or zwhz >5) whzflag =1.
if (zhaz < -6 or zhaz >6) hazflag =1.
if (zwaz < -6 or zwaz >5) wazflag =1.
if (zbmi < -5 or zbmi >5) bmiflag=1.

if ((zwhz < -5 or zwhz >5) and (zhaz < -6 or zhaz >6)) flag4 =1.
if ((zwhz < -5 or zwhz >5) and (zwaz < -6 or zwaz >5)) flag5 =1.
if ((zhaz < -6 or zhaz >6) and (zwaz < -6 or zwaz >5)) flag6 =1.
if ((zwhz < -5 or zwhz >5) and (zhaz < -6 or zhaz >6) and (zwaz < -6 or zwaz >5)) flag7 =1.
if ((bmiflag=1) and (whzflag=0) and (hazflag=0) and (wazflag=0)) flag8=1.

* no BMI flags set as 1-7.

if (zwhz < -5 or zwhz >5) flagnew =1.
if (zhaz < -6 or zhaz >6) flagnew =2.
if (zwaz < -6 or zwaz >5) flagnew =3.
if ((zwhz < -5 or zwhz >5) and (zhaz < -6 or zhaz >6)) flagnew =4.
if ((zwhz < -5 or zwhz >5) and (zwaz < -6 or zwaz >5)) flagnew =5.
if ((zhaz < -6 or zhaz >6) and (zwaz < -6 or zwaz >5)) flagnew =6.
if ((zwhz < -5 or zwhz >5) and (zhaz < -6 or zhaz >6) and (zwaz < -6 or zwaz >5)) flagnew =7.
if ((bmiflag=1) and (whzflag=0) and (hazflag=0) and (wazflag=0)) flagnew=8.
* =====declare flagged values missing=====.

if (zwhz < -5 or zwhz >5) zwhz =999.99.
if (zhaz < -6 or zhaz >6) zhaz =999.99.
if (zwaz < -6 or zwaz >5) zwaz =999.99.
if (zbmi < -5 or zbmi >5) zbmi=999.99.
MISSING VALUES zwhz,zhaz,zwaz,zbmi (999.99).

* we copy the zscores to oedema copies for special oedema adjustments later.

compute oedzbmi=zbmi.
compute oedzwhz=zwhz.
compute oedzwaz=zwaz.

Missing values oedzwhz,oedzwaz,oedzbmi (999.99).

* we want to see how many valid whz scores are there with no age.

if(missing(agedays2) and (zwhz >=-5 and zwhz <=5)) whznoage=1.

* count flags.

* =====
WEIGHT BY wgting.

* we convert these to standard DHS variables.

compute hcw5=zhaz*100.

compute hcw8=zwaz*100.

compute hcw11=zwhz*100.

compute hbmi=zBMI*100.

* putting children back in categories, if we are initially given months only, use months here.

* if given daysold or calculating from day month of birth, then use agedays2 calculation below.

* compute hcw1=agemos.

compute hcw1=trunc((agedays3/30.4375)).

if (missing(hcw1) and missing(agemos)) whznoage=1.

frequencies variables=whzflag,hazflag,wazflag,flag4,flag5,flag6,flag7,fl

ag8,bmiflag,flagnew,whznoage.

* include the next select if only if you need to have make sure all kids have ages <60 mo.

* select if (hcw1<60).

* include next 2 lines if all z score values must be present for each counted child.

* count hwmissed=hcw1,hcw5,hcw8,hcw11(missing).

* select if (hcwmissed eq 0).

IF (hcw11 > 100) whplus1 = 1.

IF (hcw11 <= 100) whplus1= 2 .

IF (hcw11 > 200) whplus2 = 1.

IF (hcw11 <= 200) whplus2= 2 .

IF (hcw11 > 300) whplus3 = 1.

IF (hcw11 <= 300) whplus3= 2 .

IF (hcw11 < -200) whmin2 = 1 .

IF (hcw11 >= -200) whmin2 = 2 .

IF (hcw11 < -300) whmin3 = 1 .

IF (hcw11 >=-300) whmin3 = 2 .

IF (hbmi < -200) bmimin2 = 1 .

IF (hbmi >= -200) bmimin2 = 2 .

IF (hbmi < -300) bmimin3 = 1 .

IF (hbmi >=-300) bmimin3 = 2 .

IF (hbmi > 100) bmiplus1 = 1 .

IF (hbmi <= 100) bmiplus1 = 2 .

IF (hbmi > 200) bmiplus2 = 1 .

IF (hbmi <= 200) bmiplus2 = 2 .

IF (hbmi > 300) bmiplus3 = 1 .

IF (hbmi <=300) bmiplus3 = 2 .

IF (hcw5 < -200) hamin2 = 1 .

IF (hcw5 >= -200) hamin2 = 2 .

IF (hcw5 < -300) hamin3 = 1 .

IF (hcw5 >=-300) hamin3 = 2 .

IF (hcw8 < -200) wamin2 = 1 .

IF (hcw8 >= -200) wamin2 = 2 .

IF (hcw8 < -300) wamin3 = 1 .

IF (hcw8 >=-300) wamin3 = 2 .

*Now that prevalences are done, we declare oedema z's of -4.44 as missing.
*get rid of zbmi=-4.44 with oedema so it is not calculated in mean zbmi.
*so we can use these z variables for means, & the oedz's for prevalences & N.

```
If (zbmi=-4.44 and (oedema='y' or oedema='Y'))zbmi=888.  
If (zwhz=-4.44 and (oedema='y' or oedema='Y'))zwhz=888.  
If (zwaz=-4.44 and (oedema='y' or oedema='Y'))zwaz=888.  
MISSING VALUES zbmi (999.99, 888).  
MISSING VALUES zwhz,zwaz (999.99, 888).
```

```
IF (whznoage=1) ageclass=0.  
IF (hcv1 >= 0 and hcv1 <6) ageclass = 1 .  
IF (hcv1 >= 6 and hcv1 <12) ageclass = 2 .  
IF (hcv1 >= 12 and hcv1 <24) ageclass = 3 .  
IF (hcv1 >= 24 and hcv1 <36) ageclass = 4 .  
IF (hcv1 >=36 and hcv1 <48) ageclass = 5 .  
IF (hcv1 >=48 and hcv1 <60) ageclass = 6 .
```

```
compute wthtsd=zwhz.  
compute htagesd=zhaz.  
compute wtagesd=zwaz.  
compute bmisd=zbmi.
```

VARIABLE LABELS

```
whmin2 "wt/ht -2SD"  
/whmin3 "wt/ht -3SD"  
/whplus1 "wt/ht +1SD"  
/whplus2 "wt/ht +2SD"  
/whplus3 "wt/ht +3SD"  
/hamin2 "ht/age -2SD"  
/hamin3 "ht/age -3SD"  
/wamin2 "wt/age -2SD"  
/wamin3 "wt/age -3SD"  
/bmimin2 "bmi/age -2SD"  
/bmimin3 "bmi/age -3SD"  
/bmiplus1 "bmi/age +1SD"  
/bmiplus2 "bmi/age +2SD"  
/bmiplus3 "bmi/age +3SD"  
/bmisd "bmi/age SD"  
/ wthtsd "wt/ht SD"  
/ htagesd "ht/age SD"  
/ wtagesd "wt/age SD".
```

```
VALUE LABELS sex 1 "male" 2 "female"  
/whmin3 1 "less 3SD" 2 "OK"  
/whmin2 1 "less 2SD" 2 "OK"  
/hamin2 1 "less 2SD" 2 "OK"
```

```

/hamin3 1 "less 3SD" 2 "OK"
/wamin2 1 "less 2SD" 2 "OK"
/wamin3 1 "less 3SD" 2 "OK"
/whplus1 1 "gtr 1SD" 2 "OK"
/whplus2 1 "gtr 2SD" 2 "OK"
/whplus3 1 "gtr 3SD" 2 "OK"
/bmimin2 1 "less 2SD" 2 "OK"
/bmimin3 1 "less 3SD" 2 "OK"
/bmiplus1 1 "gtr 1SD" 2 "OK"
/bmiplus2 1 "gtr 2SD" 2 "OK"
/bmiplus3 1 "gtr 3SD" 2 "OK"
/ageclass 0 "no age" 1 "0-5 mo." 2 "6-11 mo." 3 "12-23 mo."
4 "24-35 mo." 5 "36-47 mo." 6 "48-59 mo." 7 "60-71 mo."
8 "72-83 mo." 9 "84-95 mo." 10 "96-107 mo."
11 "108-119 mo." 12 "120-131 mo.".

```

CROSSTABS

```

/TABLES=ageclass by wamin3,wamin2,hamin3,hamin2,whmin3,whmin2,whplus1,whplus2,wh
plus3,bmimin3,bmimin2,bmiplus1,bmiplus2,bmiplus3
/ageclass by wamin3,wamin2,hamin3,hamin2,whmin3,whmin2,whplus1,whplus2,wh
plus3,bmimin3,bmimin2,bmiplus1,bmiplus2,bmiplus3 by sex
/region,urbanr by wamin3,wamin2,hamin3,hamin2,whmin3,whmin2,whplus1,whplus2,wh
plus3,bmimin3,bmimin2,bmiplus1,bmiplus2,bmiplus3
/FORMAT= AVALUE NOINDEX BOX LABELS TABLES
/CELLS= COUNT ROW .

```

MEANS

```

Tables= wtagesd, htagesd, wthtsd, bmisd by sex by ageclass
/tables= wtagesd, htagesd, wthtsd, bmisd by ageclass
/tables= wtagesd, htagesd, wthtsd, bmisd by region,urbanr.
set header off.

```

- *We now will do prevalences using cutoff scores, which may.
- *give slightly different %'s than crosstabs above due to .
- *the way SPSS rounds to the nearest whole N on weighted crosstabs.

```

select if (sex=1 or sex=2).
SET Printback=Off Length=None Width=132.
* select if (ageclass>0).
* ===== weight/age tables =====.

```

SORT CASES BY ageclass (A) .

```

Temporary.
Select if (ageclass >0).
Report
/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)

```

```

MARGINS(1,61)
/TITLE=
LEFT 'ZPR_KIDS using WHO Child Growth Standards'
/VARIABLES
oedzwaz 'wt/ageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'wt/ageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'wt/ageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
wtagesd 'wt/ageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'wt/ageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK (TOTAL)
/SUMMARY VALIDN(oedzwaz) ADD( PLT(-3)(oedzwaz)) ($dummy01(PCT)(1)) ADD(
PLT(-2)(oedzwaz)) ($dummy02(PCT)(1)) MEAN(wtagesd) ADD(STDDEV(wtagesd))
($dummy03(F)(2)) 'Grand Total' (1)
/BREAK ageclass (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(11)
/SUMMARY VALIDN(oedzwaz) SKIP(0) ADD( PLT(-3)(oedzwaz))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzwaz)) ($dummy02(PCT)(1))
MEAN(wtagesd) ADD(STDDEV(wtagesd)) ($dummy03(F)(2)).
SORT CASES BY sex (A) ageclass (A) .

```

* =====.

Temporary.

Select if (ageclass > 0).

Report

```

/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,71)

```

```

/TITLE=
LEFT 'ZPR_KID using WHO Child Growth Standards'

```

```

/VARIABLES
oedzwaz 'wt/ageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'wt/ageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'wt/ageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
wtagesd 'wt/age SD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'wt/ageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK sex (LABELS) (LEFT) (OFFSET(0)) (SKIP(1))(8)
/SUMMARY VALIDN(oedzwaz) SKIP(1) ADD( PLT(-3)(oedzwaz))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzwaz)) ($dummy02(PCT)(1))
MEAN(wtagesd) ADD(STDDEV(wtagesd)) ($dummy03(F)(2)) 'Subtotal sex' (1)
/BREAK ageclass (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(11)
/SUMMARY VALIDN(oedzwaz) SKIP(0) ADD( PLT(-3)(oedzwaz))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzwaz)) ($dummy02(PCT)(1))
MEAN(wtagesd) ADD(STDDEV(wtagesd)) ($dummy03(F)(2)).
* ===== height/age tables =====.

```

SORT CASES BY ageclass (A) .

Temporary.

Select if (ageclass > 0).

Report

```

/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(1)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,61)
/TITLE= LEFT 'ZPR_KID using WHO Child Growth Standards'
/VARIABLES
htagesd 'ht/ageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'ht/ageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'ht/ageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'ht/ageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'ht/ageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK (TOTAL)
/SUMMARY VALIDN(htagesd) ADD( PLT(-3)(htagesd)) ($dummy01(PCT)(1)) ADD(
PLT(-2)(htagesd)) ($dummy02(PCT)(1)) ADD(MEAN(htagesd)) ($dummy03(F)(2))
ADD(STDDEV(htagesd)) ($dummy04(F)(2)) 'Grand Total' (1)
/BREAK ageclass (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(11)
/SUMMARY VALIDN(htagesd) SKIP(0) ADD( PLT(-3)(htagesd)) ($dummy01(PCT)
(1)) ADD( PLT(-2)(htagesd)) ($dummy02(PCT)(1)) ADD(MEAN(htagesd)) ($dummy03
(F)(2)) ADD(STDDEV(htagesd)) ($dummy04(F)(2)).
SORT CASES BY sex (A) ageclass (A) .

```

Temporary.

Select if (ageclass >0).

* =====.

Report

```

/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(1)
UNDERSCORE(ON) ONEBREAKCOL(ON) INDENT(2)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,63) /TITLE= LEFT 'ZPR_KID using WHO Child Growth Standards' /VARIABLES
htagesd 'ht/ageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'ht/ageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'ht/ageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'ht/ageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'ht/ageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK sex (LABELS) (LEFT) (OFFSET(0)) (SKIP(1))(13)
/SUMMARY VALIDN(htagesd) SKIP(1) ADD( PLT(-3)(htagesd)) ($dummy01(PCT)(1
)) ADD( PLT(-2)(htagesd)) ($dummy02(PCT)(1)) ADD(MEAN(htagesd)) ($dummy03(F
)(2)) ADD(STDDEV(htagesd)) ($dummy04(F)(2)) 'Subtotal sex' (1)
/BREAK ageclass (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))
/SUMMARY VALIDN(htagesd) SKIP(0) ADD( PLT(-3)(htagesd)) ($dummy01(PCT)
(1)) ADD( PLT(-2)(htagesd)) ($dummy02(PCT)(1)) ADD(MEAN(htagesd)) ($dummy03
(F)(2)) ADD(STDDEV(htagesd)) ($dummy04(F)(2)).
* ===== weight/height tables =====.

```

SORT CASES BY ageclass (A) .

Temporary.

Select if (ageclass >= 0).

Report

```
/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(1)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,91)
/TITLE=
LEFT 'ZPR_KID using WHO Child Growth Standards'
/VARIABLES
oedzwhz 'wt/ht SD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'wt/ht SD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'wt/ht SD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'wt/ht SD' '> +1' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'wt/ht SD' '> +2' (RIGHT) (OFFSET(0)) (8)
$dummy05 (DUMMY) 'wt/ht SD' '> +3' (RIGHT) (OFFSET(0))(8)
wthtsd 'wt/ht SD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy06 (DUMMY) 'wt/ht SD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK (TOTAL)
/SUMMARY VALIDN(oedzwhz) ADD( PLT(-3)(oedzwhz)) ($dummy01(PCT)(1)) ADD(
PLT(-2)(oedzwhz)) ($dummy02(PCT)(1)) ADD( PGT(+1)(oedzwhz))
($dummy03(PCT)(1)) ADD( PGT(+2)(oedzwhz)) ($dummy04(PCT)(1)) ADD( PGT(
+3)(oedzwhz)) ($dummy05(PCT)(1)) MEAN(wthtsd) ADD(STDDEV(wthtsd))
($dummy06(F)(2)) 'Grand Total' (1)
/BREAK ageclass (LABELS) (LEFT) (OFFSET(0))(SKIP(0))(11)
/SUMMARY VALIDN(oedzwhz) SKIP(0) ADD( PLT(-3)(oedzwhz))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzwhz)) ($dummy02(PCT)(1)) ADD( PGT(
+1)(oedzwhz)) ($dummy03(PCT)(1)) ADD( PGT(+2)(oedzwhz)) ($dummy04(PCT)(1))
ADD( PGT(+3)(oedzwhz)) ($dummy05(PCT)(1)) MEAN(wthtsd)
ADD(STDDEV(wthtsd)) ($dummy06(F)(2)).
```

* .
SORT CASES BY sex (A) ageclass (A) .

Temporary.

Select if (ageclass >= 0).

Report

```
/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,101)
/TITLE=
LEFT 'ZPR_KID using WHO Child Growth Standards'
/VARIABLES
oedzwhz 'wt/ht SD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'wt/ht SD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'wt/ht SD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'wt/ht SD' '> +1' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'wt/ht SD' '> +2' (RIGHT) (OFFSET(0)) (8)
$dummy05 (DUMMY) 'wt/ht SD' '> +3' (RIGHT) (OFFSET(0)) (8)
```

```

wthtsd 'wt/ht SD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy06 (DUMMY) 'wt/ht SD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK sex (LABELS) (LEFT) (OFFSET(0)) (SKIP(1))(8)
/SUMMARY VALIDN(oedzwhz) SKIP(1) ADD( PLT(-3)(oedzwhz))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzwhz)) ($dummy02(PCT)(1)) ADD( PGT(
+1)(oedzwhz)) ($dummy03(PCT)(1)) ADD( PGT(+2)(oedzwhz)) ($dummy04(PCT)(1))
ADD( PGT(+3)(oedzwhz)) ($dummy05(PCT)(1)) MEAN(wthtsd)
ADD(STDDEV(wthtsd)) ($dummy06(F)(2)) 'Subtotal sex' (1)
/BREAK ageclass (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(11)
/SUMMARY VALIDN(oedzwhz) SKIP(0) ADD( PLT(-3)(oedzwhz))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzwhz)) ($dummy02(PCT)(1)) ADD( PGT(
+1)(oedzwhz)) ($dummy03(PCT)(1)) ADD( PGT(+2)(oedzwhz)) ($dummy04(PCT)(1))
ADD( PGT(+3)(oedzwhz)) ($dummy05(PCT)(1)) MEAN(wthtsd)
ADD(STDDEV(wthtsd)) ($dummy06(F)(2)).
* ===== BMI/age tables =====.
SORT CASES BY ageclass (A) .
Temporary.
Select if (ageclass > 0).
Report
/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(1)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,91)
/TITLE=
LEFT 'ZPR_KID using WHO Child Growth Standards'
/VARIABLES
oedzbmi 'BMLageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'BMLageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'BMLageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'BMLageSD' '> +1' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'BMLageSD' '> +2' (RIGHT) (OFFSET(0)) (8)
$dummy05 (DUMMY) 'BMLageSD' '> +3' (RIGHT) (OFFSET(0)) (8)
bmisd 'bmi/age SD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy06 (DUMMY) 'BMLageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK (TOTAL)
/SUMMARY VALIDN(oedzbmi) ADD( PLT(-3)(oedzbmi)) ($dummy01(PCT)(1)) ADD(
PLT(-2)(oedzbmi)) ($dummy02(PCT)(1)) ADD( PGT(+1)(oedzbmi))
($dummy03(PCT)(1)) ADD( PGT(+2)(oedzbmi)) ($dummy04(PCT)(1)) ADD( PGT(
+3)(oedzbmi)) ($dummy05(PCT)(1)) MEAN(bmisd) ADD(STDDEV(bmisd))
($dummy06(F)(2)) 'Grand Total' (1)
/BREAK ageclass (LABELS) (LEFT) (OFFSET(0))(SKIP(0))(11)
/SUMMARY VALIDN(oedzbmi) SKIP(0) ADD( PLT(-3)(oedzbmi))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzbmi)) ($dummy02(PCT)(1)) ADD( PGT(
+1)(oedzbmi)) ($dummy03(PCT)(1)) ADD( PGT(+2)(oedzbmi)) ($dummy04(PCT)(1))
ADD( PGT(+3)(oedzbmi)) ($dummy05(PCT)(1)) MEAN(bmisd) ADD(STDDEV(bmisd))
($dummy06(F)(2)).
* .

```

SORT CASES BY sex (A) ageclass (A) .

Temporary.

Select if (ageclass > 0).

Report

```
/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING'.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,101)
```

/TITLE=

LEFT 'ZPR_KID using WHO Child Growth Standards'

/VARIABLES

oedzbmi 'BMLageSD' 'N' (RIGHT) (OFFSET(0)) (8)

\$dummy01 (DUMMY) 'BMLageSD' '< -3' (RIGHT) (OFFSET(0)) (8)

\$dummy02 (DUMMY) 'BMLageSD' '< -2' (RIGHT) (OFFSET(0)) (8)

\$dummy03 (DUMMY) 'BMLageSD' '> +1' (RIGHT) (OFFSET(0)) (8)

\$dummy04 (DUMMY) 'BMLageSD' '> +2' (RIGHT) (OFFSET(0)) (8)

\$dummy05 (DUMMY) 'BMLageSD' '> +3' (RIGHT) (OFFSET(0)) (8)

bmisd 'bmiageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)

\$dummy06 (DUMMY) 'bmiageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)

/BREAK sex (LABELS) (LEFT) (OFFSET(0)) (SKIP(1))(8)

/SUMMARY VALIDN(oedzbmi) SKIP(1) ADD(PLT(-3)(oedzbmi)

(\$dummy01(PCT)(1)) ADD(PLT(-2)(oedzbmi)) (\$dummy02(PCT)(1)) ADD(PGT(+1)(oedzbmi)) (\$dummy03(PCT)(1)) ADD(PGT(+2)(oedzbmi)) (\$dummy04(PCT)(1))

ADD(PGT(+3)(oedzbmi)) (\$dummy05(PCT)(1)) MEAN(bmisd) ADD(STDDEV(bmisd))

(\$dummy06(F)(2)) 'Subtotal sex' (1)

/BREAK ageclass (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(11)

/SUMMARY VALIDN(oedzbmi) SKIP(0) ADD(PLT(-3)(oedzbmi)

(\$dummy01(PCT)(1)) ADD(PLT(-2)(oedzbmi)) (\$dummy02(PCT)(1)) ADD(PGT(+1)(oedzbmi)) (\$dummy03(PCT)(1)) ADD(PGT(+2)(oedzbmi)) (\$dummy04(PCT)(1))

ADD(PGT(+3)(oedzbmi)) (\$dummy05(PCT)(1)) MEAN(bmisd) ADD(STDDEV(bmisd))

(\$dummy06(F)(2)).

* =====weight to age region tables =====.

SORT CASES BY region .

Temporary.

Select if (ageclass > 0).

Report

```
/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING'.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,77)
```

/TITLE=

LEFT 'ZPR_KID using WHO Child Growth Standards'

/VARIABLES

oedzwaz 'wt/ageSD' 'N' (RIGHT) (OFFSET(0)) (8)

\$dummy01 (DUMMY) 'wt/ageSD' '< -3' (RIGHT) (OFFSET(0)) (8)

\$dummy02 (DUMMY) 'wt/ageSD' '< -2' (RIGHT) (OFFSET(0)) (8)


```

$dummy03 (DUMMY) 'wt/ageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'wt/ageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK region (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(26)
/SUMMARY VALIDN(oedzwaz) SKIP(0) ADD( PLT(-3)(oedzwaz)) ($dummy01(PCT)
(1)) ADD( PLT(-2)(oedzwaz)) ($dummy02(PCT)(1)) ADD(MEAN(wtagesd)) ($dummy03
(F)(2)) ADD(STDDEV(wtagesd)) ($dummy04(F)(2)).

```

SORT CASES BY urbanr .

Temporary.

Select if (ageclass > 0).

Report

```

/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING'.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,78)

```

/TITLE=

LEFT 'ZPR_KID using WHO Child Growth Standards'

/VARIABLES

```

oedzwaz 'wt/ageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'wt/ageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'wt/ageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'wt/ageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'wt/ageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK urbanr (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(9)
/SUMMARY VALIDN(oedzwaz) SKIP(0) ADD( PLT(-3)(oedzwaz)) ($dummy01(PCT)
(1)) ADD( PLT(-2)(oedzwaz)) ($dummy02(PCT)(1)) ADD(MEAN(wtagesd)) ($dummy03
(F)(2)) ADD(STDDEV(wtagesd)) ($dummy04(F)(2)).

```

* =====height to age region tables =====.

SORT CASES BY region .

Temporary.

Select if (ageclass > 0).

Report

```

/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING'.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,108)

```

/TITLE=

LEFT 'ZPR_KID using WHO Child Growth Standards'

/VARIABLES

```

htagesd 'ht/ageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'ht/ageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'ht/ageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'ht/ageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'ht/ageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK region (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(28)

```

```
/SUMMARY VALIDN(htagesd) SKIP(0) ADD( PLT(-3)(htagesd)) ($dummy01(PCT)
(1)) ADD( PLT(-2)(htagesd)) ($dummy02(PCT)(1)) ADD(MEAN(htagesd)) ($dummy03
(F)(2)) ADD(STDDEV(htagesd)) ($dummy04(F)(2)).
```

SORT CASES BY urbanr .

Temporary.

Select if (ageclass > 0).

Report

```
/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING'. ' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,78)
```

```
/TITLE=
```

```
LEFT 'ZPR_KID using WHO Child Growth Standards'
```

```
/VARIABLES
```

```
htagesd 'ht/ageSD' 'N' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy01 (DUMMY) 'ht/ageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy02 (DUMMY) 'ht/ageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy03 (DUMMY) 'ht/ageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy04 (DUMMY) 'ht/ageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
```

```
/BREAK urbanr (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(9)
```

```
/SUMMARY VALIDN(htagesd) SKIP(0) ADD( PLT(-3)(htagesd)) ($dummy01(PCT)
(1)) ADD( PLT(-2)(htagesd)) ($dummy02(PCT)(1)) ADD(MEAN(htagesd)) ($dummy03
(F)(2)) ADD(STDDEV(htagesd)) ($dummy04(F)(2)).
```

* ===== weight to height region tables =====.

SORT CASES BY region .

Temporary.

Select if (ageclass >= 0).

Report

```
/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING'. ' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,108)
```

```
/TITLE=
```

```
LEFT 'ZPR_KID using WHO Child Growth Standards'
```

```
/VARIABLES
```

```
oedzwhz 'wt/ht SD' 'N' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy01 (DUMMY) 'wt/ht SD' '< -3' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy02 (DUMMY) 'wt/ht SD' '< -2' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy03 (DUMMY) 'wt/ht SD' '> +1' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy04 (DUMMY) 'wt/ht SD' '> +2' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy05 (DUMMY) 'wt/ht SD' '> +3' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy06 (DUMMY) 'wt/ht SD' 'Mean' (RIGHT) (OFFSET(0)) (8)
```

```
$dummy07 (DUMMY) 'wt/ht SD' 'StdDev' (RIGHT) (OFFSET(0))(8)
```

```

/BREAK region (LABELS) (LEFT) (OFFSET(0))(SKIP(0))(28)
/SUMMARY VALIDN(oedzwhz) SKIP(0) ADD( PLT(-3)(oedzwhz))
($dummy01(PCT)(1)) ADD( PLT(-2)( oedzwhz)) ($dummy02(PCT)(1)) ADD( PGT(
+1)(oedzwhz)) ($dummy03(PCT)(1)) ADD( PGT(+2)(oedzwhz)) ($dummy04(PCT)(1))
ADD( PGT(+3)(oedzwhz)) ($dummy05(PCT)(1)) ADD(MEAN(whtsd))
($dummy06(F)(2)) ADD(STDDEV(whtsd)) ($dummy07(F)(2)).

```

SORT CASES BY urbanr .

Temporary.

Select if (ageclass >= 0).

Report

```

/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,89)

```

/TITLE=

LEFT 'ZPR_KID using WHO Child Growth Standards'

/VARIABLES

```

oedzwhz 'wt/ht SD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'wt/ht SD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'wt/ht SD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'wt/ht SD' '> +1' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'wt/ht SD' '> +2' (RIGHT) (OFFSET(0)) (8)
$dummy05 (DUMMY) 'wt/ht SD' '> +3' (RIGHT) (OFFSET(0)) (8)
$dummy06 (DUMMY) 'wt/ht SD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy07 (DUMMY) 'wt/ht SD' 'StdDev' (RIGHT) (OFFSET(0))(8)

```

```

/BREAK urbanr (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(9)

```

```

/SUMMARY VALIDN(oedzwhz) SKIP(0) ADD( PLT(-3)(oedzwhz))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzwhz)) ($dummy02(PCT)(1)) ADD( PGT(
+1)(oedzwhz)) ($dummy03(PCT)(1)) ADD( PGT(+2)(oedzwhz)) ($dummy04(PCT)(1))
ADD( PGT(+3)(oedzwhz)) ($dummy05(PCT)(1)) ADD(MEAN(whtsd))
($dummy06(F)(2)) ADD(STDDEV(whtsd)) ($dummy07(F)(2)).

```

* ===== BMI to age region tables =====.

SORT CASES BY region .

Temporary.

Select if (ageclass > 0).

Report

```

/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,108)

```

/TITLE=

LEFT 'ZPR_KID using WHO Child Growth Standards'

```

/VARIABLES
oedzbmi 'bmiageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'bmiageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'bmiageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'bmiageSD' '> +1' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'bmiageSD' '> +2' (RIGHT) (OFFSET(0)) (8)
$dummy05 (DUMMY) 'bmiageSD' '> +3' (RIGHT) (OFFSET(0)) (8)
bmisd 'bmiageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy06 (DUMMY) 'bmiageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK region (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(28)
/SUMMARY VALIDN(oedzbmi) SKIP(0) ADD( PLT(-3)(oedzbmi))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzbmi)) ($dummy02(PCT)(1)) ADD( PGT(
+1)(oedzbmi)) ($dummy03(PCT)(1)) ADD( PGT(+2)(oedzbmi)) ($dummy04(PCT)(1))
ADD( PGT(+3)(oedzbmi)) ($dummy05(PCT)(1)) MEAN(bmisd) ADD(STDDEV(bmisd))
($dummy06(F)(2)).

```

SORT CASES BY urbanr .

Temporary.

Select if (ageclass > 0).

Report

```

/FORMAT= CHWRAP(ON) BRKSPACE(-1) SUMSPACE(0) AUTOMATIC
PREVIEW(OFF) CHALIGN(BOTTOM) CHDSPACE(0)
UNDERSCORE(ON) ONEBREAKCOL(OFF)
PAGE(1) MISSING'.' LENGTH(1, 99999)ALIGN(LEFT) TSPACE(0) FTSPACE(0)
MARGINS(1,99)
/TITLE=

```

LEFT 'ZPR_KID using WHO Child Growth Standards'

/VARIABLES

```

oedzbmi 'BMIageSD' 'N' (RIGHT) (OFFSET(0)) (8)
$dummy01 (DUMMY) 'BMIageSD' '< -3' (RIGHT) (OFFSET(0)) (8)
$dummy02 (DUMMY) 'BMIageSD' '< -2' (RIGHT) (OFFSET(0)) (8)
$dummy03 (DUMMY) 'BMIageSD' '> +1' (RIGHT) (OFFSET(0)) (8)
$dummy04 (DUMMY) 'BMIageSD' '> +2' (RIGHT) (OFFSET(0)) (8)
$dummy05 (DUMMY) 'BMIageSD' '> +3' (RIGHT) (OFFSET(0)) (8)
$dummy06 (DUMMY) 'BMIageSD' 'Mean' (RIGHT) (OFFSET(0)) (8)
$dummy07 (DUMMY) 'BMIageSD' 'StdDev' (RIGHT) (OFFSET(0)) (8)
/BREAK urbanr (LABELS) (LEFT) (OFFSET(0)) (SKIP(0))(19)
/SUMMARY VALIDN(oedzbmi) SKIP(0) ADD( PLT(-3)(oedzbmi))
($dummy01(PCT)(1)) ADD( PLT(-2)(oedzbmi)) ($dummy02(PCT)(1)) ADD( PGT(
+1)(oedzbmi)) ($dummy03(PCT)(1)) ADD( PGT(+2)(oedzbmi)) ($dummy04(PCT)(1))
ADD( PGT(+3)(oedzbmi)) ($dummy05(PCT)(1)) ADD(MEAN(bmisd)) ($dummy06(F)(2))
ADD(STDDEV(bmisd)) ($dummy07(F)(2)).

```

* omit next two lines if you need to see intermediate results to debug.

XSAVE outfile="C:\Users\Dr. Love\Desktop\ZPR_KIDS_z.sav" /keep=casenum to
zbmi,zwaz,zwhz,whzflag to whznoage.

ERASE FILE='C:\Users\Dr. Love\Desktop\ZPR_KIDS.sav'.

ERASE FILE='C:\Users\Dr. Love\Desktop\ZPR_KIDS\sav'.

set header on.

set printback=on length=None Width=132.

execute.

File Attachments

1) [igrowup_DHShh.sps](#), downloaded 478 times

Subject: Re: Using igrowup SPSS syntax

Posted by [Mlue](#) on Mon, 25 Mar 2019 10:38:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello BLM,

Try the following codes (SPSS) to match some of the results in the report. Please note that I have never used igrowup and I'm not sure what it does.

I hope this helps. I have also attached text files with extra syntaxes.

Example for WASTING

```
*/ [ OPEN THE DATA - PR RECODE FILE (MWPR7HFL) ] **.
```

```
** CHILD NUTRITIONAL STATUS.
```

```
** Table 11.1 Nutritional status of children.
```

```
** MALAWI DHS 2015-16 **.
```

```
** **
```

```
GET
```

```
FILE='C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\MW_2015-16_DHS_03042019_835_52565_SPSS\MWPR7HSV\MWPR7HFL.SAV'.
```

```
DATASET NAME DataSet1 WINDOW=FRONT.
```

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

```
** WEIGHT VARIABLE.
```

```
COMPUTE weight = hv005/1000000.
```

```
WEIGHT BY weight.
```

```
** COMPLEX SURVEY VARIABLES.
```

```
COMPUTE psu = hv021.
```

```
COMPUTE strata = hv023.
```

```
*****
```

RENAME VARIABLES (hc27 = sex) (hv270 = wealth) (hv025 = residence) (hv024 = region).

** CHILD AGE IN MONTHS.

RECODE hc1 (0 THRU 5 = 1) (6 THRU 8 = 2) (9 THRU 11 = 3)
(12 THRU 17 = 4) (18 THRU 23 = 5) (24 THRU 35 = 6) (36 THRU 47 = 7)
(48 THRU 59 = 8) INTO child_age.

VARIABLE LABELS child_age 'Child age (months)'.
EXECUTE.

VALUE LABELS child_age 1 "0-5 months" 2 "6-8 months" 3 "9-11 months" 4 "12-17 months" 5
"18-23 months" 6 "24-35 months" 7 "36-47 months" 8 "48-59 months".

SELECT IF child_age LE 8.

** WASTING = Weight-for-height.

COMPUTE wasting=\$SYSMIS.

IF hv103=1 wasting=0.

IF missing(hc72) wasting=\$SYSMIS.

IF (hc72>=9996) wasting= \$SYSMIS.

IF (hc72 LT -200 AND hv103=1) wasting = 1.

EXECUTE.

VARIABLE LABELS wasting 'Wasting children'.
VALUE LABELS wasting 0 "Not wasting" 1 "Wasting" .

SELECT IF wasting LE 1.

** CHECK.

FREQUENCIES VARIABLES= wasting
/ORDER=ANALYSIS.

CROSSTABS

/TABLES=child_age sex wealth residence region BY wasting
/FORMAT=AVALUE TABLES
/CELLS=ROW
/COUNT ROUND CELL.

CROSSTABS

/TABLES=child_age sex wealth residence region BY wasting
/FORMAT=AVALUE TABLES
/CELLS=COUNT
/COUNT ROUND CELL.

/* COMPLEX SURVEYS */.

* Analysis Preparation Wizard.

CSPLAN ANALYSIS

/PLAN
FILE='C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\MAMAWIDHS2015_16_PRCPLAN.csaplan'
/PLANVARS ANALYSISWEIGHT=weight
/SRSESTIMATOR TYPE=WOR
/PRINT PLAN
/DESIGN STRATA=strata CLUSTER=psu
/ESTIMATOR TYPE=WR.

* Complex Samples Frequencies.

CSTABULATE

/PLAN
FILE='C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\MAMAWIDHS2015_16_PRCPLAN.csaplan'
/TABLES VARIABLES=wasting
/CELLS POPSIZE TABLEPCT
/STATISTICS DEFF
/MISSING SCOPE=TABLE CLASSMISSING=EXCLUDE.

* Complex Samples Crosstabs.

CSTABULATE

/PLAN
FILE='C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\MAMAWIDHS2015_16_PRCPLAN.csaplan'
/TABLES VARIABLES=child_age sex wealth residence region BY wasting

```
/CELLS ROWPCT
/STATISTICS CV
/MISSING SCOPE=TABLE CLASSMISSING=EXCLUDE.
```

I also don't know why you want to merge the PR and KR files. Nonetheless, the following code will assist with the merging.

Example for MERGING PR and KR files

```
/** BEFORE YOU START.
* PLEASE MAKE SURE THAT THE PR & KR DATA FILES ARE SAVED ON THE SAME
FOLDER
**/
```

```
**** OPEN THE KR FILE ****.
GET
FILE='C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR
DATA\MWKR7HFL.SAV'.
DATASET NAME KR_Dataset WINDOW=FRONT.
```

```
COMPUTE line=B16.
SORT CASES BY V001(A) V002(A) line(A).
```

```
SAVE OUTFILE
="C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR
DATA\MWKR7HFLtemp.SAV".
```

```
*****
```

```
**** OPEN THE PR FILE ****.
GET
FILE='C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR
DATA\MWPR7HFL.SAV'.
DATASET NAME PR_Dataset WINDOW=FRONT.
```

```
COMPUTE V001=HV001.
COMPUTE V002=HV002.
COMPUTE line=HVIDX.
SORT CASES BY V001(A) V002(A) line(A).
```

```
SAVE OUTFILE =
"C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR
DATA\MWPR7HFLtemp.SAV".
DATASET CLOSE KR_Dataset.
```

```
*****
```


**** THE MERGE (ONE TO MANY) ****.

*MATCH FILES FILE =

```
"C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR  
DATA\MWKR7HFLtemp.SAV" /IN=inKRfile  
  /TABLE = "C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR  
DATA\MWPR7HFLtemp.SAV" /IN=inPRfile  
  /BY V001 V002 line.
```

MATCH FILES

```
FILE="C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR  
DATA\MWKR7HFLtemp.SAV" /IN=inKRfile  
  /FILE= "C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR  
DATA\MWPR7HFLtemp.SAV" /IN=inPRfile  
  /BY V001 V002 line.  
SELECT IF inPRfile EQ 1.  
DATASET CLOSE PR_Dataset.  
DATASET NAME MERGED_Dataset WINDOW=FRONT.
```

SAVE OUTFILE =

```
"C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\KR_PR  
DATA\MERGED_KR_PR_MALAWI.SAV".
```

File Attachments

- 1) [MERGING DATA PR & KR DATA ON SPSS - MALAWI DHS 2015-16 _ SYNTAX.txt](#), downloaded 482 times
 - 2) [STUNTING - NUTRITIONAL STATUS OF CHILDREN - MALAWI DHS 2015-16 - SPSS.txt](#), downloaded 482 times
 - 3) [UNDERWEIGHT - NUTRITIONAL STATUS OF CHILDREN - MALAWI DHS 2015-16 - SPSS.txt](#), downloaded 457 times
 - 4) [WASTING - NUTRITIONAL STATUS OF CHILDREN - MALAWI DHS 2015-16 - SPSS.txt](#), downloaded 466 times
-

Subject: Re: Using igrowup SPSS syntax

Posted by [BLM](#) on Thu, 04 Apr 2019 19:17:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Mlue,

Thank you so much for your help. When I used the syntax you sent for Wasting, I still get errors. Please see the attached document with the syntax errors.

Thank you again!

File Attachments

- 1) [Capture Syntax errors.PNG](#), downloaded 416 times
-

Subject: Re: Using igrowup SPSS syntax
Posted by [Mlue](#) on Thu, 04 Apr 2019 21:25:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

You need to change the path/directory to where your data is located...

This path responds to the place where I had saved the data:

'C:\Users\User1\Documents\MW_2015-16_DHS_05242018_253_52565\MW_2015-16_DHS_03042019_835_52565_SPSS\MWPR7HS\MWPR7HFL.SAV'.

If your data is located in the C-drive, under a folder called "MALAWI_DHS", then your path/directory could be:

'C:\BLM\MALAWI_DHS\MWPR7HFL.SAV'.

Read more on this:

1. <https://www.spss-tutorials.com/spss-opening-data-with-syntax/>
 2. <https://www.spss-tutorials.com/spss-cd-command/>
-

Subject: Re: Using igrowup SPSS syntax
Posted by [BLM](#) on Fri, 05 Apr 2019 00:30:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Thank you so much! I figured it out ;)

One last question, how would I go about getting food security index using the same data? Thank you again!
