Addendum to Technical Paper #5
National Health and Aging Tends Study (NHATS)

SAS Programming Statements for Construction of Dementia Classification in the National Health and Aging Trends Study.

August 1, 2013

Suggested Citation: Spillman, Brenda C and Skehan, Maureen E. 2013. SAS programming statements for construction of dementia classification in the National Health and Aging Trends Study. Addendum to NHATS Technical Paper #5. Baltimore: Johns Hopkins University School of Public Health. Available at www.NHATS.org. The technical paper was prepared with funding from the National Institute on Aging (U01AG032947).
** NOTE: The input file to run this code is the NHATS_Round_1_File**

** FORMATS FOR CONSTRUCTED VARIABLES **;

```sas
proc format;
value r1demclas
  1="1:Probable"
  2="2:Possible"
  3="3:No Impairment"
  -1="-1:NH resident"
  -9="-9:Missing"
;
value ad8dem
  1="1:Probable"
  2="2:Possible"
value clockf
  0-1="0-1:Impaired"
  2-5="2-5:Not impaired"
;
value wordrecf
  0-3="0-3: Impaired 
  4-20="4-20: Not impaired 
;
value dateprf
  0-3="0-3: Impaired 
  4-8="4-8: Not impaired 
;
** DATE STEP CODE FOR CREATING DEMENTIA CLASSIFICATION VARIABLE **;

length r1demclas
  ad8_dem ad8_1-ad8_8 ad8miss_1-ad8miss_8 ad8_score ad8_miss 
  date_mon date_day date_yr date_dow date_sum date_sumr 
  preslast presfirst vplast vpfirst presvp presvpr date_prvp 
  clock_scorer irecall drecall wordrecall0_20 
  clock65 word65 datena65 domain65 
  3;

label r1demclas="NHATS Dementia Classification 65+";

** 1) SET MISSING (RESIDENTIAL CARE FQ ONLY) AND N.A. (NURSING HOME RESIDENTS) **;

if r1dresid=3 then r1demclas=-9 ;
if r1dresid=4 then r1demclas=-1 ;

** 2) CODE PROBABLE IF DEMENTIA DIAGNOSIS REPORTED BY SELF OR PROXY **;

if hc1disescn9=1 and is1resptype in (1,2) then r1demclas=1 ;
** 3a) CODE AD8_SCORE **;

array think    {*} cp1chgthink1-cp1chgthink8; ** QUESTIONNAIRE ITEMS **;
array ad8item {*} ad8_1-ad8_8;
array ad8miss {*} ad8miss_1-ad8miss_8;

ad8_score = 1;
ad8_miss = 1;

do i=1 to dim(ad8item);
** INITIALIZE COUNTS TO NOT APPLICABLE**;
ad8item[i]=-1;
ad8miss[i]=-1;
** ASSIGN VALUES TO AD8 ITEMS IF PROXY AND DEMENTIA CLASS NOT ALREADY ASSIGNED BY REPORTED DIAGNOSIS **;
if is1resptype=2 and r1demclas=. then do;
ad8item[i]=1; ** PROXY REPORTS A CHANGE OR ALZ/DEMENTIA **;
else if think{i}=2 then ad8item[i]=0; ** PROXY REPORTS NO CHANGE **;
ad8_score=sum(of ad8item{*}); ** COUNT AD8 ITEMS **;
ad8_score=sum(of ad8miss{*}); ** COUNT MISSING AD8 ITEMS **;
end;
end;

** 3b) CODE AD8 DEMENTIA CLASS **;

if ad8_score>=2 then ad8_dem=1; ** IF SCORE >=2 THEN MEETS AD8 CRITERION **;
if ad8_score in (0,1) or ad8_miss=8 then ad8_dem=2; ** IF SCORE IS 0 OR 1 OR ALL ITEMS MISSING THEN DOES NOT MEET AD8 CRITERION **;

** 4) UPDATE DEMENTIA CLASSIFICATION VARIABLE WITH AD8 CLASS **;

if r1demclas=. then do;
if ad8_dem=1 then r1demclas=1; ** PROBABLE BASED ON AD8 SCORE **;
if ad8_dem=2 and cg1speaktosp=2 then r1demclas=3; ** NO DIAGNOSIS, DOES NOT MEET AD8 CRITERION, AND PROXY SAYS CANNOTASK SP COGNITIVE ITEMS **;
end;

** 5) CODE DATE ITEMS AND COUNT **;

array cg1date {*} cg1todaydat1-cg1todaydat4;
array date_item {*} date_mon date_day date_yr date_dow;
do i=1 to dim(date_item);
  if cg1date(i) > 0 then date_item(i)=cg1date(i); ** CODE ONLY YES/NO RESPONSES: MISSING/N.A. CODES -1,-9 LEFT MISSING **;  
  if cg1date(i) in (-7,2) then date_item(i)=0; ** 2:NO/DK OR -7:REFUSED RECODED TO 0:NO/DK/RF **;  
  date_sum=sum(of date_item{*}); ** COUNT CORRECT DATE ITEMS **;  
end;

if date_sum=. then do;
  if cg1speaktosp=2 then date_sum=-2; ** PROXY SAYS CAN'T SPEAK TO SP **;  
  else if cg1speaktosp=1 and max(of cg1date{*})=-1 then date_sum=-3; ** PROXY SAYS CAN SPEAK TO SP BUT SP UNABLE TO ANSWER **;  
end;
date_sumr=date_sum;

if date_sum=-2 then date_sumr=.; ** MISSING IF PROXY SAYS CAN'T SPEAK TO SP **;  
else if date_sum=-3 then date_sumr=0; ** 0 IF SP UNABLE TO ANSWER **;  

** 6) PRESIDENT AND VICE PRESIDENT NAME ITEMS AND COUNT **;

array cg1pres {*} cg1presidna1 cg1presidna3 cg1vpname1 cg1vpname3;
array pres_item {*} preslast presfirst vplast vpfirst;
do i=1 to dim(pres_item);
  if cg1pres(i) > 0 then pres_item(i)=cg1pres(i); ** CODE ONLY YES/NO RESPONSES: MISSING/N.A. CODES -1,-9 LEFT MISSING **;  
  if cg1pres(i) in (-7,2) then pres_item(i)=0; ** 2:NO/DK OR -7:REFUSED RECODED TO 0:NO/DK/RF **;  
presvp=sum(of pres_item{*}); ** COUNT CORRECT PRESIDENT/VEEP NAME ITEMS **;  
end;

if presvp=. then do;
  if cg1speaktosp=2 then presvp=-2; ** PROXY SAYS CAN'T SPEAK TO SP **;  
  else if cg1speaktosp=1 and max(of cg1pres{*})=-1 then presvp=-3; ** PROXY SAYS CAN SPEAK TO SP BUT SP UNABLE TO ANSWER **;  
end;
presvpr=presvp;

if presvp=-2 then presvpr=.; ** MISSING IF PROXY SAYS CAN'T SPEAK TO SP **;  
else if presvp=-3 then presvpr=0; ** 0 IF SP UNABLE TO ANSWER **;  

** 7) ORIENTATION DOMAIN: SUM OF DATE RECALL AND PRESIDENT NAMING **;
**8) EXECUTIVE FUNCTION DOMAIN: CLOCK DRAWING SCORE**

clock_scorer=cg1dclkdraw;

if cg1dclkdraw in (-2,-9) then clock_scorer=.;
if cg1dclkdraw in (-3,-4,-7) then clock_scorer=0;

**IMPUTE MEAN SCORE TO PERSONS MISSING A CLOCK**;
if cg1dclkdraw=-9 and cg1speaktosp=1 then clock_scorer=2; **IF PROXY SAID CAN ASK SP**;
if cg1dclkdraw=-9 and cg1speaktosp=-1 then clock_scorer=3; **IF SELF RESPONDENT**;

**9) MEMORY DOMAIN: IMMEDIATE AND DELAYED WORD RECALL**

array cg1recall {*} cg1dwrddlyrc cg1dwrdimmrc;
array word_recall {*} irecall drecall;

do i=1 to dim(word_recall);
    word_recall{i}=cg1recall{i};
    if cg1recall{i} in (-2,-1) then word_recall{i}=.;
    if cg1recall{i} in (-7,-3) then word_recall{i}=0;
wordrecall0_20=sum(of word_recall{*});
end;

**10) CREATE COGNITIVE DOMAINS FOR ALL ELIGIBLE**

**I.E. PROXY BUT PROXY SAYS CAN ASK SP, NOT FQ ONLY, NOT NH**;

if 1 < clock_scorer <=5 then clock65=0;
if 0 <= clock_scorer <=1 then clock65=1;

if 3 < wordrecall0_20 <=20 then word65=0;
if 0 <= wordrecall0_20 <=3 then word65=1;

if 3 < date_prvp <= 8 then datena65=0;
if 0 <= date_prvp <= 3 then datena65=1;

**10) CREATE COGNITIVE DOMAIN SCORE**;

array domains {*} clock65 word65 datena65;

do i=1 to dim(domains);
    domain65=sum(of domains{*});
end;

**11) UPDATE COGNITIVE CLASSIFICATION**;

if r1demclas=. and cg1speaktosp in (-1,1) then do;
if 2 <= domain65 <= 3 then r1demclas=1;  ** PROBABLE DEMENTIA  **;
if domain65 = 1 then r1demclas=2;  ** POSSIBLE DEMENTIA  **;
if domain65 = 0 then r1demclas=3;  ** NO DEMENTIA  **;
end;