

Addendum to Technical Paper #5

National Health and Aging Trends Study (NHATS)

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

August 1, 2013

Revised September 21, 2022

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).

```

** FORMATS FOR CONSTRUCTED VARIABLES **;
proc format;
value demclas
  1="1:Probable dementia"
  2="2:Possible dementia"
  3="3:No dementia"
  -1="-1:Deceased or nursing home resident in R1 and R2"
  -9="-9:Missing"
;
value ad8dem
  1="1:Meets dementia criteria"
  2="2:Does not meet dementia criteria"
;
value domain_1
  0="0:Does not meet criteria"
  1="1:Meets criteria"
;
value domain65_1
  0="0 Not impaired"
  1="Impaired in 1 domain"
  2="Impaired in 2 domains"
  3="Impaired in 3 domains"
;
run;

** RUN Proc Format.sas BEFORE CREATING DEMCLAS FOR ROUND 1**;

libname nhatsr1 "[location]";

Data nhatsr1.Round1;
set nhatsr1.NHATS_Round_1_SP_File;
** DATE STEP CODE FOR CREATING DEMENTIA CLASSIFICATION VARIABLE
**;
length rldemclas
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 rldemclas="NHATS Dementia Classification 65+";
** 1) SET MISSING (RESIDENTIAL CARE FQ ONLY) AND N.A. (NURSING
HOME RESIDENTS) **;
if rldresid=3 then rldemclas=-9 ;
if rldresid=4 then rldemclas=-1 ;

```

```

** 2) CODE PROBABLE IF DEMENTIA DIAGNOSIS REPORTED BY SELF OR
PROXY **;
if hcldisescn9=1 and islresptype in (1,2) then rldemclas=1 ;

** 3a) CODE AD8_SCORE **;
array think {*} cplchgthink1-cplchgthink8; ** 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 islresptype=2 and rldemclas=. then do;
ad8item{i}=.;
if think{i} in (1,3) then 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 **;
if ad8item{i} in (0,1) then ad8miss{i}=0;
else if ad8item{i}=. then ad8miss{i}=1;
ad8_miss=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 rldemclas=. then do;
if ad8_dem=1 then rldemclas=1; ** PROBABLE BASED ON AD8 SCORE
**;
if ad8_dem=2 and cglspeaktosp=2 then rldemclas=3; ** NO
DIAGNOSIS, DOES NOT MEET AD8 CRITERION, AND PROXY SAYS CANNOT
ASK SP COGNITIVE ITEMS **;
end;
** 5) CODE DATE ITEMS AND COUNT **;
array cgldate {*} cgltodaydat1-cgltodaydat4;
array date_item {*} date_mon date_day date_yr date_dow;

```

```

do i=1 to dim(date_item);
if cgldate{i} > 0 then date_item{i}=cgldate{i}; ** CODE ONLY
YES/NO RESPONSES: MISSING/N.A. CODES -1,-9 LEFT MISSING **;
if cgldate{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 cglspeaktosp=2 then date_sum=-2; ** PROXY SAYS CAN'T SPEAK TO
SP **;
else if cglspeaktosp=1 and max(of cgldate{*})=-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 cglpres {*} cglpresidna1 cglpresidna3 cglvpname1
cglvpname3;
array pres_item {*} preslast presfirst vplast vpfirst;
do i=1 to dim(pres_item);
if cglpres{i} > 0 then pres_item{i}=cglpres{i}; ** CODE ONLY
YES/NO RESPONSES: MISSING/N.A. CODES -1,-9 LEFT MISSING **;
if cglpres{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 cglspeaktosp=2 then presvp=-2; ** PROXY SAYS CAN'T SPEAK TO
SP **;
else if cglspeaktosp=1 and max(of cglpres{*})=-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 **;

date_prvp=sum(date_sumr,presvpr);
** 8) EXECUTIVE FUNCTION DOMAIN: CLOCK DRAWING SCORE **;
clock_scorer=cgldclkdraw;

```

```

if cglldclkdraw in (-2,-9) then clock_scorer=.;
if cglldclkdraw in (-3,-4,-7) then clock_scorer=0;
** IMPUTE MEAN SCORE TO PERSONS MISSING A CLOCK **;
if cglldclkdraw=-9 and cglspeaktosp=1 then clock_scorer=2; ** IF
PROXY SAID CAN ASK SP **;
if cglldclkdraw=-9 and cglspeaktosp=-1 then clock_scorer=3; ** IF
SELF RESPONDENT **;
** 9) MEMORY DOMAIN: IMMEDIATE AND DELAYED WORD RECALL **;
array cglrecall {*} cglldwrddimrc cglldwrddlyrc;
array word_recall {*} irecall drecall;
do i=1 to dim(word_recall);
word_recall{i}=cglrecall{i};
if cglrecall{i} in (-2,-1) then word_recall{i}=.;
if cglrecall{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=clock65+word65+datena65;
end;
** 11) UPDATE COGNITIVE CLASSIFICATION **;
if rldemclas=. and cglspeaktosp in (-1,1) then do;

if 2 <= domain65 <=3 then rldemclas=1; ** PROBABLE DEMENTIA **;
if domain65 =1 then rldemclas=2; ** POSSIBLE DEMENTIA **;
if domain65 =0 then rldemclas=3; ** NO DEMENTIA **;
end;

keep spid rldemclas domain65 ad8_dem clock65 word65 datena65;

format
rldemclas demclas.
ad8_dem ad8dem.
clock65 word65 datena65 domain_1.
domain65 domain65_1.;

run;

```