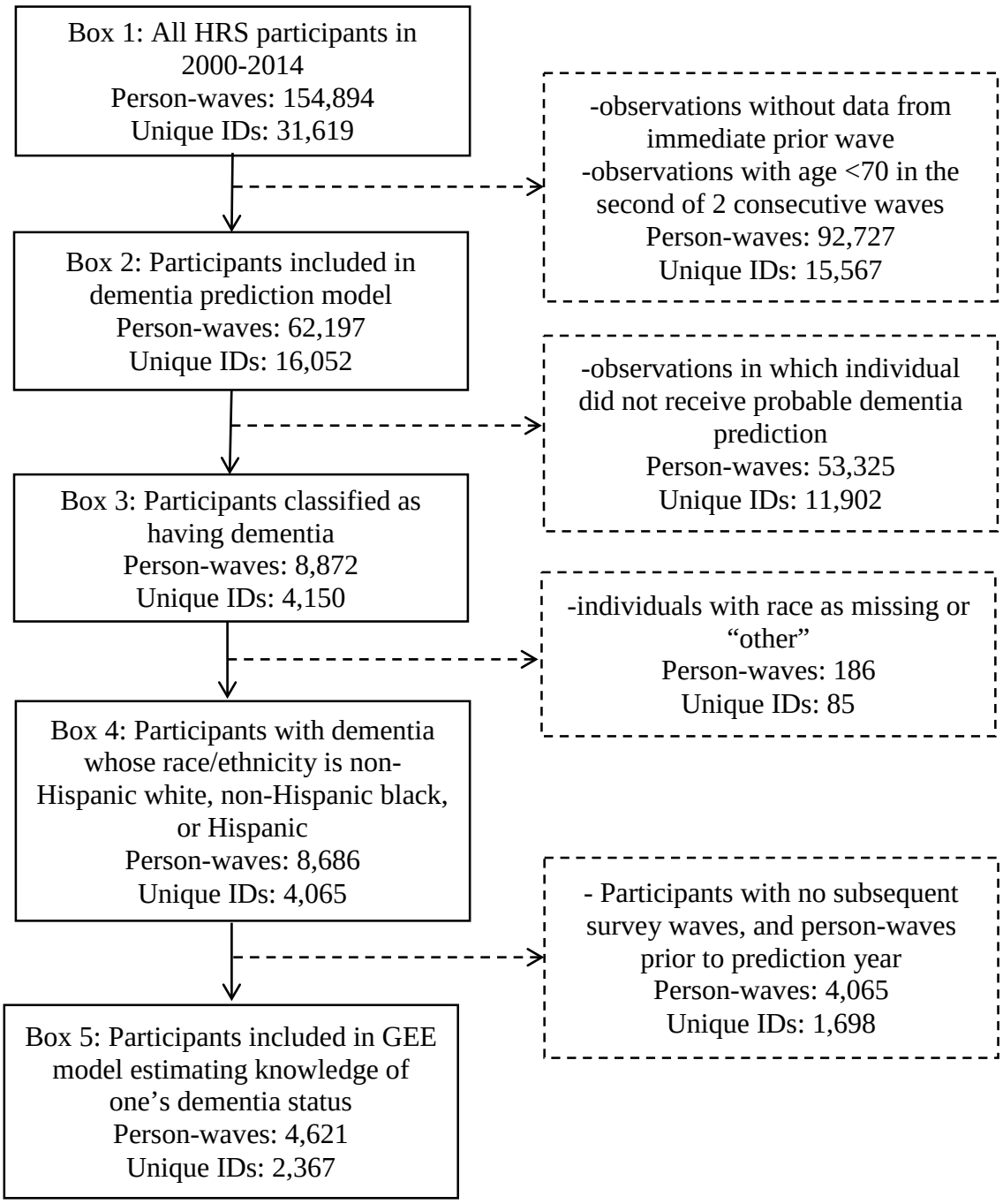
Supplementary Figure S1: Consort diagram



Supplementary Table S1: Baseline characteristics of HRS participants in GEE model of reporting being told about dementia by a physician

	Longitudinal Analyses (n=5,299,921)			
Weighted Observations*	Non-Hispanic white (n=4,111,149)	Non-Hispanic black (n=700,120)	Hispanic (n=488,653)	
Age				†
70-74	3.9%	7.0%	10.0%	
75-79	8.4%	15.9%	11.7%	
80-84	23.2%	32.0%	36.4%	
85+	64.6%	45.1%	41.9%	
Female	68.9%	71.2%	66.2%	
Proxy Respondent	55.1%	64.3%	56.6%	†
Mean TICS Score (SD)[1]	10.47 (0.17)	8.61 (0.32)	10.0 (0.37)	§
Mean IQCODE Score (SD) [2]	3.19 (0.05)	3.10 (0.10)	2.97 (0.13)	
Mean Number of ADL Limitations (SD)[3]	1.98 (0.06)	1.69 (0.11)	2.21 (0.17)	
Mean Number of IADL Limitations (SD)[4]	2.26 (0.05)	2.11 (0.10)	2.07 (014)	
Mean Number of Comorbidities (SD)	3.09 (0.04)	3.04 (0.09)	2.99 (0.12)	
Living in a Nursing Home	31.0%	18.1%	14.6%	†

* Weighted percentages/means using the HRS sample weights
† Weighted chi-squared test p-value < 0.5
§ Weighted ANOVA test p-value < 0.5
1. Only for participants who did not have a proxy respondent (self-reported). Scale from 0-33; Higher scores indicate higher cognitive function
2. Only for participants who had a proxy respondent. Scale from 0-5; Lower scores indicate higher cognitive function
3. ADL: Activities of Daily Living. Numbers are the reported number of activities (6 total) participants have difficulty performing; Lower scores indicate higher functional ability
4. IADL: Instrumental Activities of Daily Living. Numbers are the reported number of activities (5 total) participants have difficulty performing; Lower scores indicate higher functional ability

TECHNICAL APPENDIX: DEMENTIA PREDICTION MODEL

Manuscript title: Racial and Ethnic Differences in Knowledge about One's Dementia Status

The Health and Retirement Study (HRS) is a nationally-representative longitudinal survey panel of older adults in the United States. HRS is sponsored by the National Institute on Aging and conducted by the University of Michigan. The open cohort is either initially interviewed or re-interviewed in even years. Our full sample contains 31,619 respondents who participated in HRS between 2000 (Wave 5) and 2014 (Wave 12) (Technical Appendix Figure 1). We used data from the RAND HRS Longitudinal File 2014 (v.2)[1] and RAND HRS Fat Files from the years 1998-2000. Our sample size is enumerated by year in Technical Appendix Table 1.

Technical Appendix Table 1: HRS samples sizes by race/ethnicity and year

	2000 (Wave 5)	2002 (Wave 6)	2004 (Wave 7)	2006 (Wave 8)	2008 (Wave 9)	2010 (Wave 10)	2012 (Wave 11)	2014 (Wave 12)
Non-Hispanic White	14941	13841	14862	13749	12727	14160	13124	11819
Non-Hispanic Black	2680	2474	2841	2568	2425	4179	3894	3619
Hispanic	1552	1472	1929	1711	1655	2963	2823	2641
Other	402	375	494	440	409	721	702	658
Non-response	4	3	3	1	1	11	11	10
Total	19579	18165	20129	18469	17217	22034	20554	18747

The Aging, Demographics, and Memory Study (ADAMS) was a supplementary study conducted with a subset of HRS respondents between the years 2000 and 2004. Following the method of Hurd et al.[2] we included results from the initial ADAMS cohort, Wave A (N=844). We used the ADAMS tracker and Wave A data files.[3] As part of ADAMS, respondents were given a full in-person clinical workup, along with an extended battery of cognitive tests and

[1] Delia Bugliari NC, Chris Chan, Orla Hayden, Jessica Hayes, Michael Hurd, Regan Main, Joshua Mallett, Colleen McCullough, Erik Meijer, Michael Moldoff, Philip Pantoja, Susann Rohwedder, Patricia St.Clair,. RAND Health and Retirement Study Longitudinal File 2014 (V2). In: RAND Center for the Study of Aging UoM, editor. 2 ed2018.

[2] Hurd MD, Martorell P, Delavande A, Mullen KJ, Langa KM. Monetary costs of dementia in the United States. New England Journal of Medicine. 2013;368(14):1326-34

[3] Aging, Demographics, and Memory Study (ADAMS). In: University of Michigan DU, editor. 7.0 ed2010.

questionnaires. Each respondent was then given a definitive diagnosis of cognitive status. Sample sizes by diagnosis are shown in Technical Appendix Table 2.

Technical Appendix Table 2: ADAMS cognitive status diagnoses

N	Diagnosis	Category
122	Probably AD	Dementia
107	Possible AD	Dementia
22	Probably Vascular Dementia	Dementia
26	Possible Vascular Dementia	Dementia
2	Parkinson's	Dementia
0	Huntington's	Dementia
0	Progressive Supranuclear Palsy	Dementia
1	Normal pressure hydrocephalus	Dementia
23	Dementia of undetermined etiology	Dementia
1	Frontal lobe dementia	Dementia
2	Severe head trauma (with residual)	Dementia
1	Alcoholic dementia	Dementia
0	ALS with dementia	Dementia
0	Hypoperfusion dementia	Dementia
1	Probable Lewy Body dementia	Dementia
0	Post encephalitic dementia	Dementia
0	Pick's disease	Dementia
0	Possible Lewy Body dementia	Dementia
94	Mild-ambiguous	CIND
4	Mild cognitive impairment	CIND
20	Cognitive impairment second to vascular dementia	CIND
34	Stroke	CIND
10	Other neurological conditions	CIND
55	Other medical conditions	CIND
8	Depression	CIND
2	Psychiatric disorder	CIND
8	Mental retardation	CIND
3	Alcohol abuse (past)	CIND
3	Alcohol abuse (current)	CIND
0	CIND, not specified	CIND
307	Normal/non-case	Normal

Dementia prediction model overview

The purpose of the dementia prediction model is to identify the categorical cognitive status of all HRS respondents aged 70 and older as either normal, cognitive-impairment non-dementia (CIND), or dementia. Our methodology and variable selection were based on a dementia prediction model developed by Hurd and colleagues.[2] We first built an ordered probit model with the ADAMS subsample using their clinical diagnosis as our "gold standard" outcome variable. This yielded model equations, coefficients, and cutpoints, which we then applied to the full HRS sample to predict the cognitive status of all HRS respondents.

Ordered probit model with ADAMS

Using the ADAMS subsample (N = 844), we created two separate ordered probit models, one for self-respondents (N = 645) and one for proxy respondents (N = 199). The SAS code can be found in the program titled "ADAMS dataset and ordered probit model". All variables included in the model are from the HRS dataset, because although all cognitive battery questions were asked both in the main HRS survey and the ADAMS subsurvey, variations in question deployment produce subtle differences in answers between ADAMS and HRS. The only variables used from the ADAMS survey itself are the diagnosis outcome variable and respondent's age. Self- and proxy respondents had to be separated because questions regarding cognitive status differ depending on who is the primary respondent. Relevant cognition questions asked to self-respondents are the Telephone Interview of Cognitive Status (TICS) battery, including: dates (composite of ability to state the correct day of the week, year, month, and day number), backwards counting from 20, serial 7s, scissors, cactus, vice president, immediate recall (of 10), delayed recall (of 10). Proxies respond to the Jorm et al. IQCODE battery concerning their perception of the respondent's cognitive ability and its recent changes. Following the method of Hurd et al., we removed respondents who had missing values for all TICS items in their ADAMS data (N = 12).

Both self- and proxy respondents are asked about which activities of daily living (ADLs) and independent activities of daily living (IADLs) they or their respondent has trouble conducting. In our models, we included the following ADLs: eating, transferring, toileting, dressing, bathing, and walking. We also included the following IADLs: meal preparations, grocery shopping, telephoning, medications, managing money. Demographic variables included in both models were age category (70-74; 75-79; 80-84; 85-59; 90+) and education category (less than high school; high school; more than high school).

Both models incorporate variables that represent the change in answer from the prior to the current HRS wave for certain TICS items to control for the individual's rate of cognitive decline. ADAMS includes a variable that enumerates which HRS wave the ADAMS participant was selected from, which helped us to determine which wave constituted the respondent's "prior" wave. The self-respondent model includes variables that indicate the change in score from prior to the current wave for all TICS items. For proxies, we controlled for whether the respondent used a proxy in the prior wave, scores of all TICS items from the prior HRS wave, and the prior IQCODE score.

Variables with missing values were replaced with zero to ensure these respondents would not be dropped from the model, and missing indicators were included in each model to control for non-response. However, we found that respondents either had all TICS items recorded, or none. This caused issues of collinearity when we included missing indicators for all TICS items. Thus we decided that using one missing indicator from TICS, chosen randomly, to represent all missing TICS items was appropriate. For the self-respondent model, we only included the missing indicator for backwards counting from 20 to represent missing data for all other TICS items (dates, serial 7s, scissors, cactus, vice president, immediate recall, delayed recall), and we also included the missing indicator for change in score for backwards counting from 20 from the prior to the current wave to control for missing information in the change variables as well. For the proxy model, we included the dates missing indicator, but did not include a change in score missing indicator due to collinearity.

Predicting dementia status for HRS

Each ordered probit model produces a model equation, corresponding coefficient values, and cutpoints (Technical Appendix Tables 4a-4c for self-respondents, 5a-5c for proxy respondents). To predict the dementia status of all HRS respondents, we first replaced the beta coefficients enumerated in each model equation with their corresponding coefficient values (see SAS program "HRS dataset and dementia predictions"). To assign age category, we used one greater than the respondent's age at the end of the HRS wave in which the other model variables were collected, in order to best match the data from which the prediction model was derived, which used age at ADAMS interview which took place approximately one year after the most recent HRS wave (see Hurd et al.[2]). The outcome in the ADAMS sample data used to train our prediction models was the clinical diagnosis made during the year following the most recent HRS wave in which the respondent participated. Similarly, our modeled predictions should be

interpreted to apply to the year following the most recent HRS wave from which data was used to make that prediction.

We calculated each respondent's numeric outcome score using their respective equations (self-respondent or proxy). The numeric outcome of an ordered probit model is not itself meaningful of any cognitive status or severity of cognitive decline. Next, we calculated each respondent's predicted probability of membership to each dementia category (Dementia, CIND, Normal) per the cutpoints produced by the ordered probit model. We did so using cumulative distribution functions (CDF) with the following equations:

Technical Appendix Table 3: Predicted probability equations

Predicted probability (Dementia) = normal CDF (cutpoint 1 – numeric outcome score)
Predicted probability (CIND) = normal CDF (cutpoint 2 – numeric outcome score) – normal CDF (cutpoint 1 – numeric outcome score)
Predicted probability (Normal) = 1 - normal CDF (cutpoint 2 – numeric outcome score)

To categorize each respondent into a distinct dementia status category for each HRS wave, we compared each predicted probability and chose the largest to represent that respondent's cognitive status for that year. If a respondent was classified as having dementia once based on the prediction model, their imputed status was dementia for all the following HRS waves in which they participated, even if the model predicted their status as CIND or normal (N = 453; N_{waves} = 638). Following this imputation rule, we retained respondents whose predicted status went from dementia to CIND or dementia to CIND to normal, but we removed those whose dementia prediction jumped from dementia to normal between two consecutive waves (N = 20). Like Hurd et al.'s model, our model is used to generate predictions for those ages 70 and older, so participants under 70 were excluded from our sample. Additionally, because the model variables include change in certain scores between the current and immediately prior wave, the equation can only be applied to those with at least two consecutive waves of participation. After applying these exclusion criteria, we were left with 16,052 unique individuals for whom dementia status predictions were generated (see Box 2 of Technical Appendix Figure 1).

Supplementary Excel File ("HRS_DEMENTIA_PREDICTIONS.xlsx") is a balanced long form dataset containing model-predicted probabilities of each cognitive status (normal, CIND, dementia) and predicted and imputed status for each included individual in every HRS wave.

The equations, coefficients, and cut points for both models are shown below:

Technical Appendix Table 4a: Ordered probit model of ADAMS self-respondents, equation

Dementia diagnosis (Dementia = 1; CIND = 2; Normal = 3) =
B1*Age category + B2*Education category + B3*Gender + B4*ADLs + B5*IADLs +
B6*ADL change between prior and current wave +
B7*IADL change between prior and current wave + B8*dates +
B9*change in dates between prior and current wave +
B10*Backwards counting from 20 + B11*Scissors + B12*Cactus +
B13*vice president + B14*Immediate recall + B15*Delayed recall +
B16*Missing indicator for backwards counting from 20 +
B17*Missing indicator for change in backwards counting from 20 between prior and current

Technical Appendix Table 4b: Ordered probit model of ADAMS self-respondents, coefficients

N = 645			
	Estimate	Standard Error	p-value
Age category:			
70-74	-1.083	0.191	<.0001
75-79	-0.734	0.170	<.0001
80-84	-0.503	0.149	0.001
85-59	0.001	0.155	0.997
90+	Ref		
Education category:			
Less than HS	-0.178	0.184	0.335
HS	-0.245	0.128	0.055
Greater than HS	Ref		
Gender:			
Male	-0.031	0.111	0.782
Female	Ref		
Functional limitations:			
ADL score	0.065	0.061	0.289
IADL score	-0.223	0.083	0.008
ADL change*	-0.043	0.058	0.460
IADL change*	0.134	0.076	0.075
TICS items:			
Dates	0.261	0.088	0.003
Backwards counting	-0.528	0.214	0.014

Serial 7s	0.128	0.044	0.004
Scissors	-0.599	0.465	0.198
Cactus	0.091	0.167	0.589
Vice President	0.303	0.170	0.076
Immediate recall	0.164	0.058	0.005
Delayed recall	0.192	0.049	<.0001
Change variables*:			
Dates change	-0.014	0.080	0.863
Backwards counting change	0.367	0.201	0.069
Serial 7s change	-0.043	0.039	0.263
Scissors change	0.291	0.389	0.453
Cactus change	0.073	0.172	0.671
Vice President change	-0.074	0.148	0.618
Immediate recall change	-0.067	0.046	0.150
Delayed recall change	-0.083	0.037	0.025
Missing indicators**:			
Backwards counting missing indicator	1.125	0.909	0.216
Backwards counting change missing indicator	-0.350	0.534	0.512
Intercept	0.111	0.528	0.833

*Change in score between prior wave and current wave

**Indicates variable had a missing value and was recoded to 0

Technical Appendix 4c: Ordered probit model of ADAMS self-respondents, cutpoints

Cutpoint	**Estimate**
Cutpoint 1	-0.111
Cutpoint 2	1.207

Technical Appendix Table 5a: Ordered probit model of ADAMS proxy respondents, equation

Dementia diagnosis =
B1*Age category + B2*Education category + B3*Gender + B4*ADLs + B5*IADLs +
B6*ADL change between prior and current wave +
B7*IADL change between prior and current wave + B8*average IQCODE score +
B9*Used a proxy in the prior wave +
B10*Change in IQCODE score between prior and current waves +
B11*Prior wave's score for dates + B12*Prior wave's score for serial 7s +
B13*Prior wave's score for immediate recall +
B14*Prior wave's score for delayed recall +
B15*Missing indicator for prior wave's dates variable

Technical Appendix Table 5b: Ordered probit model of ADAMS proxy respondents, coefficients

N = 199			
	Estimate	Standard Error	p-value
Age category:			
70-74	-1.023	0.446	0.022
75-79	-1.084	0.485	0.025
80-84	-1.063	0.444	0.017
85-59	-0.548	0.457	0.230
90+	Ref		
Education category:			
Less than HS	-0.461	0.558	0.409
HS	-0.455	0.356	0.202
Greater than HS	Ref		
Gender:			
Male	-0.497	0.276	0.072
Female	Ref		
Functional limitations:			
ADL score	-0.042	0.114	0.716
IADL score	-0.483	0.143	0.001
ADL change*	0.046	0.105	0.661
IADL change*	0.402	0.150	0.008
Jorm IQCODE (average)			
Current IQCODE	-0.599	0.183	0.001
Change in IQCODE*	0.174	0.138	0.208
Use of proxy in prior wave	0.992	1.062	0.350

TICS score from prior wave:			
Dates	0.421	0.282	0.136
Backwards counting			
Serial 7s	0.090	0.119	0.449
Scissors			
Cactus			
Vice President	-0.511	0.464	0.270
Immediate recall	0.265	0.170	0.120
Delayed recall	-0.132	0.152	0.387
Missing indicators**:			
Dates missing indicator	1.830	1.377	0.184
Intercept	1.083	1.137	0.341

*Change in score between prior wave and current wave

**Indicates variable had a missing value and was recoded to 0

Technical Appendix Table 5c: Ordered probit model of ADAMS proxy respondents, cutpoints

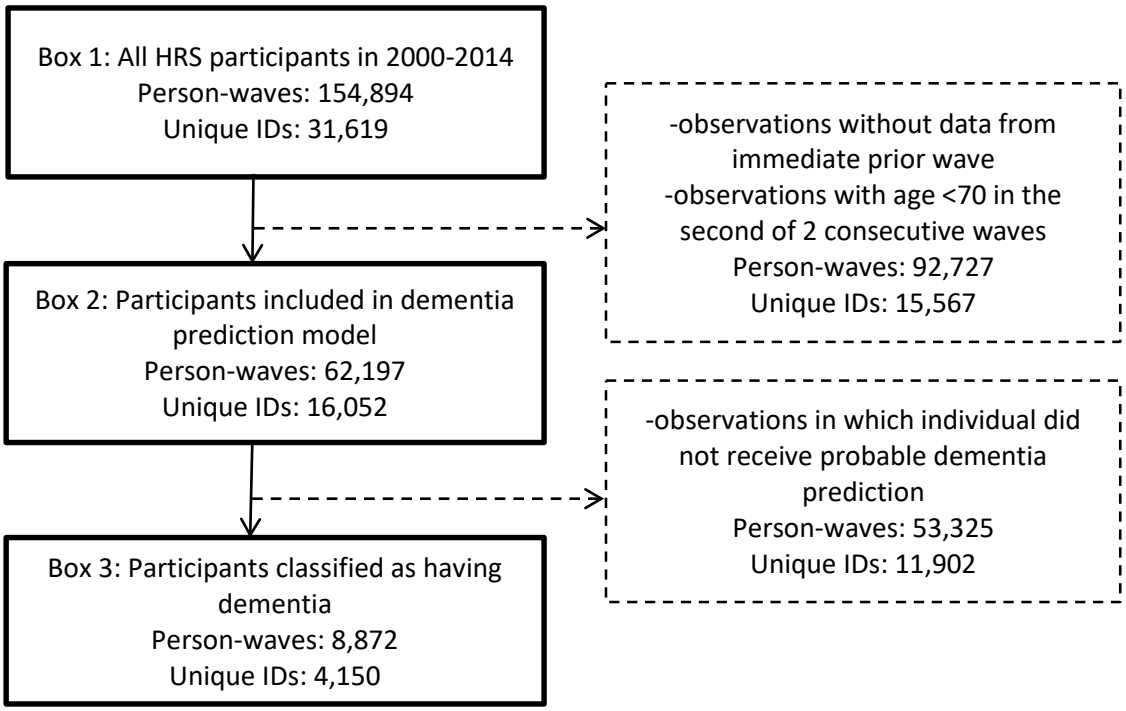
Cutpoint	**Estimate**
Cutpoint 1	-1.083
Cutpoint 2	0.243

Technical Appendix Table 6: Number of HRS respondents by year, model-predicted dementia status, and respondent type*

	2001		2003		2005		2007	
Respondent type	Self	Proxy	Self	Proxy	Self	Proxy	Self	Proxy
Dementia	431	577	507	593	561	518	612	455
CIND	1090	234	1036	220	1088	180	1159	126
Normal	4659	169	4680	184	4823	162	5052	110
	2009		2011		2013		2015	
	Self	Proxy	Self	Proxy	Self	Proxy	Self	Proxy
Dementia	590	435	556	550	668	504	650	479
CIND	1173	136	1304	175	1357	114	1306	96
Normal	5353	102	5379	110	5391	101	5128	72

***Limited to those aged 70 or older in the year before prediction year, with race as either non-Hispanic white, non-Hispanic black, or Hispanic

Technical Appendix Figure 1: Consort diagram



DEMENTIA PREDICTION MODEL SAS CODE

```
/*************************************************************************************
************\
//*************************************       PART 1: Macros
**************************************\\
/*************************************************************************************
************\

Programmed by: Joanna Emerson
Date: March 19, 2019 (updated May 2, 2019 by Flora Berklein)

Purpose: Data cleaning and analysis for ADAMS and HRS data

Macros included:
      1. Recode non-binary TICS answers into 1 = correct, 0 = incorrect
      2. Create missing indicators for TICS items and replace missing values with 0
      3. Create change in TICS score variables for each TICS item
      4. Create composite functional limitations (ADLs and IADLs) variables
      5. Create composite dates variables
      6. Create composite Jorm IQCODE variables
      7. HRS full wave data cleaning
            a. Prediction model for self-respondents
            b. Prediction model for proxy-respondents
            c. Create probability of dementia and dementia category from prediction
model
            d. Imputed dementia category based on predictions

/*************************************************************************************
************/

* Macro 1: Recode TICS to binary;
%macro TICS_binary(w=, pw=);
      if R&w.BWC20 = 2 then R&w.BWC20 = 1;
      if R&w.SCIS = 2 then R&w.SCIS = 1;
      if R&w.CACT  = 2 then R&w.CACT  = 1;
      if R&w.VP = 2 then R&w.VP = 1;
      if R&pw.BWC20 = 2 then R&pw.BWC20 = 1;
      if R&pw.SCIS = 2 then R&pw.SCIS = 1;
      if R&pw.CACT = 2 then R&pw.CACT = 1;
      if R&pw.VP = 2 then R&pw.VP = 1;
%mend;

*Macro 2: Create TICS items missing indicators and replace missing values with 0;
%macro TICS_missings(w=, pw=);

*create missing indicators variables for all TICS items;
array TICSbinary{28} R&w.BWC20 R&pw.BWC20 R&w.SER7 R&pw.SER7 R&w.SCIS R&pw.SCIS
R&w.IMRC R&pw.IMRC R&w.CACT R&pw.CACT R&w.VP R&pw.VP R&w.DLRC R&pw.DLRC
                                 flag_bwc20&w flag_bwc20&pw flag_s7&w flag_s7&pw
flag_sci&w flag_sci&pw flag_imr&w flag_imr&pw flag_cac&w flag_cac&pw flag_vp&w
flag_vp&pw flag_delr&w flag_delr&pw;
      do i = 1 to 14;
      j = i + 14;
      if TICSbinary{i} = .M | TICSbinary{i}  = .N | TICSbinary{i}  = .Q
|TICSbinary{i}  = .X |TICSbinary{i}  = .S|TICSbinary{i}  = . then TICSbinary{j} = 1;
      else TICSbinary{j} = 0;
end;


*missing indicators in change variables;
if R&w.BWC20  = .M | R&w.BWC20  = .N | R&w.BWC20  = .Q |R&w.BWC20  = .X|R&w.BWC20  = .
|R&w.BWC20  = .S|
```

```
   R&pw.BWC20 = .M | R&pw.BWC20 = .N | R&pw.BWC20 = .Q |R&pw.BWC20 = .X |R&pw.BWC20
= .S|R&pw.BWC20 = . then flag_cbwc20&pw&w. = 1;
else flag_cbwc20&pw&w. = 0;
if R&w.SER7 = .M | R&w.SER7 = .N | R&w.SER7 = .Q |R&w.SER7 = .X |R&w.SER7
= .S|R&w.SER7 = . |
   R&pw.SER7  = .M | R&pw.SER7  = .N | R&pw.SER7  = .Q |R&pw.SER7  = .X |R&pw.SER7
= .S|R&pw.SER7  = . then flag_cs7&pw&w. = 1;
else flag_cs7&pw&w. = 0;
if R&w.IMRC = .M | R&w.IMRC = .N | R&w.IMRC = .Q |R&w.IMRC = .X |R&w.IMRC
= .S|R&w.IMRC = . |
   R&pw.IMRC = .M | R&pw.IMRC = .N | R&pw.IMRC = .Q |R&pw.IMRC = .X |R&pw.IMRC
= .S|R&pw.IMRC = . then flag_cimr&pw&w. = 1;
else flag_cimr&pw&w. = 0;
if R&w.DLRC = .M | R&w.DLRC = .N | R&w.DLRC = .Q |R&w.DLRC = .X |R&w.DLRC
= .S|R&w.DLRC = .|
   R&pw.DLRC = .M | R&pw.DLRC = .N | R&pw.DLRC = .Q |R&pw.DLRC = .X |R&pw.DLRC
= .S|R&pw.DLRC = . then flag_cdelr&pw&w. = 1;
else flag_cdelr&pw&w. = 0;

*replace missings with 0;
array missings{28} R&w.BWC20 R&pw.BWC20 R&w.SER7 R&pw.SER7 R&w.SCIS R&pw.SCIS R&w.IMRC
R&pw.IMRC R&w.CACT R&pw.CACT R&w.VP R&pw.VP R&w.DLRC R&pw.DLRC
                              flag_bwc20&w flag_bwc20&pw flag_s7&w flag_s7&pw
flag_sci&w flag_sci&pw flag_imr&w flag_imr&pw flag_cac&w flag_cac&pw flag_vp&w
flag_vp&pw flag_delr&w flag_delr&pw;
      do x = 1 to 14;
      y = x + 14;
      if missings{x} = . | missings{y} = 1 then missings{x} = 0;
end;

drop i j x y;

%mend;

*Macro 3: Change in TICS item scores from prior to current wave;
%macro TICS_change(w=, pw=);
      cbwc20&pw&w. = R&w.BWC20 - R&pw.BWC20;
      cs7&pw&w. = R&w.SER7 - R&pw.SER7 ;
      csci&pw&w. = R&w.SCIS - R&pw.SCIS;
      ccac&pw&w. = R&w.CACT - R&pw.CACT;
      cvp&pw&w. = R&w.VP - R&pw.VP;
      cimr&pw&w. = R&w.IMRC - R&pw.IMRC;
      cdelr&pw&w. = R&w.DLRC - R&pw.DLRC;
%mend;

*Macro 4: Create composite ADL/IADL functional limitations from prior and current
wave;
%macro
adls(ADLeat=,ADLbed=,ADLtoilt=,ADLdress=,ADLbath=,ADLwalk=,IADLmeal=,IADLshop=,IADLpho
ne=,IADLmeds=,IADLmoney=,pADLeat=,pADLbed=,pADLtoilt=,pADLdress=,pADLbath=,pADLwalk=,p
IADLmeal=,pIADLshop=,pIADLphone=,pIADLmeds=,pIADLmoney=, w=, pw=);

*Create missing indicator for current wave ADLs, populated only if all variables are
missing;
flag_ADL&w=0;
flag_ADLcount = 0;
array missing_c{6} &ADLeat &ADLbed &ADLtoilt &ADLdress &ADLbath &ADLwalk;
      do a = 1 to 6;
      if missing_c{a} = . |missing_c{a} = .X | missing_c{a} = .Z | missing_c{a} = .S
| missing_c{a} = .D | missing_c{a} = .R | missing_c{a} = .N then flag_ADLcount =
flag_ADLcount+1;
end;
drop a;
```

```
if flag_ADLcount = 6 then flag_ADL&w = 1;

*Create missing indicator for prior wave ADLs, populated only if all variables are
missing;

flag_pADLcount = 0;
array missing_p{6} &pADLeat &pADLbed &pADLtoilt &pADLdress &pADLbath &pADLwalk;
      do b = 1 to 6;
      if missing_p{b} = . | missing_p{b} = .X | missing_p{b} = .Z | missing_p{b} = .S
| missing_p{b} = .D | missing_p{b} = .R | missing_p{b} = .N then flag_pADLcount =
flag_pADLcount+1;
end;
drop b;
if flag_pADLcount = 6 then flag_ADL&w = 1;

*Replace missings for both ADLs and IADLs from current and prior wave with 0;
array replace{22} &ADLeat &ADLbed &ADLtoilt &ADLdress &ADLbath &ADLwalk &IADLmeal
&IADLshop &IADLphone &IADLmeds &IADLmoney
                              &pADLeat &pADLbed &pADLtoilt &pADLdress &pADLbath
&pADLwalk &pIADLmeal &pIADLshop &pIADLphone &pIADLmeds &pIADLmoney;
      do c = 1 to 22;
      if replace{c} = . | replace{c} = .X | replace{c} = .Z | replace{c} = .S |
replace{c} = .D | replace{c} = .R | replace{c} = .N | replace{c} = 2 | replace{c} = 9
then replace{c} = 0;
end;
drop c;

*Create composite ADL scores, current wave;
ADLcomposite&w = 0;
array ADL{6} &ADLeat &ADLbed &ADLtoilt &ADLdress &ADLbath &ADLwalk;
      do i = 1 to 6;
      if ADL{i} = 1| ADL{i} = 2 then ADLcomposite&w = ADLcomposite&w  + 1;
end;
drop i;

*Create composite IADL scores, current wave;
IADLcomposite&w = 0;
array iADL{5} &IADLmeal &IADLshop &IADLphone &IADLmeds &IADLmoney;
      do j = 1 to 5;
      if iADL{j} = 1 then IADLcomposite&w = IADLcomposite&w + 1;
end;
drop j;

*Create composite ADL scores, prior wave;
ADLcomposite&pw = 0;
array pADL{6} &pADLeat &pADLbed &pADLtoilt &pADLdress &pADLbath &pADLwalk;
      do i = 1 to 6;
      if pADL{i} = 1 then ADLcomposite&pw = ADLcomposite&pw + 1;
end;
drop i;

*Create composite IADL scores, prior wave;
IADLcomposite&pw = 0;
      array piADL{5} &pIADLmeal &pIADLshop &pIADLphone &pIADLmeds &pIADLmoney;
      do j = 1 to 5;
if piADL{j} = 1 then IADLcomposite&pw  = IADLcomposite&pw  + 1;
end;
drop j;

*create change in ADL scores variable;
ADLchange&pw.&w = ADLcomposite&w - ADLcomposite&pw;
*create change in IADL scores variable;
IADLchange&pw.&w = IADLcomposite&w - IADLcomposite&pw;
```

```
%mend;

*Macro 5: Create composite dates variable from individual TICS items;
%macro dates(day=, month=, year=, dayweek=, pday=, pmonth=, pyear=, pdayweek=, w=,
pw=);

*create flag variable;
array create_d{8} flag_daycount flag_mocount flag_yrcount flag_dwcount flag_pdaycount
flag_pmocount flag_pyrcount flag_pdwcount;
      do j = 1 to 8;
      create_d{j} = 0;
end;
drop j;

*Add to flag count if missing, current wave;
array missing_dc{8} &day &month &year &dayweek flag_daycount flag_mocount flag_yrcount
flag_dwcount;
      do i = 1 to 4;
      k = i + 4;
      if missing_dc{i} = .| missing_dc{i} = .N | missing_dc{i} =.X | missing_dc{i}
= .Z | missing_dc{i} = .S | missing_dc{i} = .D| missing_dc{i} = .R|
         missing_dc{i} = .M then missing_dc{k} = 1;
end;
drop i k;

*Add to flag count if missing, prior wave;
array missing_dp{8} &pday &pmonth &pyear &pdayweek flag_pdaycount flag_pmocount
flag_pyrcount flag_pdwcount;
      do i = 1 to 4;
      k = i + 4;
      if missing_dp{i} = .| missing_dp{i} = .N | missing_dp{i} =.X | missing_dp{i}
= .Z | missing_dp{i} = .S | missing_dp{i} = .D| missing_dp{i} = .R|
         missing_dp{i} = .M then missing_dp{k} = 1;
end;
drop i k;

*Replace missings with 0;
array replace_d{16} flag_daycount flag_mocount flag_yrcount flag_dwcount
flag_pdaycount flag_pmocount flag_pyrcount flag_pdwcount
                              &day &month &year &dayweek &pday &pmonth &pyear
&pdayweek;
      do m = 1 to 8;
      n = m + 8;
      if replace_d{m} = 1 then replace_d{n} = 0;
end;
drop m n;


*flag if ALL dates are empty;
if flag_daycount = 1 & flag_mocount = 1 & flag_yrcount = 1 & flag_dwcount = 1 then
flag_dates&w = 1;
else flag_dates&w = 0;
if flag_pdaycount = 1 & flag_pmocount = 1 & flag_pyrcount = 1 & flag_pdwcount = 1 then
flag_dates&pw = 1;
else flag_dates&pw = 0;

*make composite dates variables;
dates&w = &day + &month + &year + &dayweek;
dates&pw = &pday + &pmonth + &pyear + &pdayweek;
cdates&pw.&w = dates&w -dates&pw;
%mend;
```

```
*Macro 6: Jorm IQCODE scoring: HRS PROVIDED SAS CODE FOR RECODING JORM:
http://hrsonline.isr.umich.edu/sitedocs/userg/dr-006.pdf;
%macro jorm(base_familyfriends=, base_recent=, base_convos=, base_addressphone =,
      base_daymonth=,     base_thingskept=,   base_findthings=,   base_machines=,
      base_newmachines =, base_newthings=,    base_story =,
base_decisions=,    base_money =, base_finances=,     base_math =,  base_reasoning=,
      better_familyfriends=,     better_recent=,     better_convos=,
      better_addressphone =,     better_daymonth=,   better_thingskept=,
      better_findthings=,
better_machines=,   better_newmachines =,      better_newthings=,  better_story =,
      better_decisions=,  better_money =,     better_finances=,   better_math =,
      better_reasoning =, worse_familyfriends=,      worse_recent=,
      worse_convos=,
worse_addressphone =,      worse_daymonth=,    worse_thingskept=,  worse_findthings=,
      worse_machines=,    worse_newmachines =,       worse_newthings=,   worse_story
=,    worse_decisions=,   worse_money =,      worse_finances=,    worse_math =,
      worse_reasoning=,
pbase_familyfriends=,      pbetter_newthings=,        pworse_newthings=,
      pbase_recent=,      pbase_story=,       pbetter_story=,     pworse_story=,
      pbase_convos=,      pbase_decisions=,   pbetter_decisions=,
      pworse_decisions=,  pbase_addressphone=,       pbase_money=,
pbetter_money=,     pworse_money=,      pbase_daymonth=,    pbase_finances=,
      pbetter_finances=,  pworse_finances=,   pbase_thingskept=,  pbase_math=,
      pbetter_math=,      pworse_math=,       pbase_findthings=,  pbase_reasoning=,
      pbetter_reasoning=,
pworse_reasoning=,  pbase_machines=,    pbetter_familyfriends=,
      pworse_familyfriends=,     pbase_newmachines=,        pbetter_recent=,
      pworse_recent=,     pbase_newthings=,   pbetter_convos=,    pworse_convos=,
      pbetter_addressphone=,     pworse_addressphone=,
pbetter_daymonth=,  pworse_daymonth=,   pbetter_thingskept=,       pworse_thingskept=,
      pbetter_findthings=,       pworse_findthings=,        pbetter_machines=,
      pworse_machines=,   pbetter_newmachines=,      pworse_newmachines=, w=, pw=);

*replace missings with 0;
array missing_jorm[96] &base_familyfriends &base_recent &base_convos
&base_addressphone &base_daymonth &base_thingskept &base_findthings &base_machines
&base_newmachines &base_newthings
 &base_story &base_decisions &base_money &base_finances &base_math &base_reasoning
&better_familyfriends &better_recent &better_convos &better_addressphone
&better_daymonth &better_thingskept &better_findthings &better_machines
&better_newmachines
 &better_newthings &better_story &better_decisions &better_money &better_finances
&better_math &better_reasoning &worse_familyfriends &worse_recent &worse_convos
&worse_addressphone &worse_daymonth &worse_thingskept &worse_findthings
&worse_machines &worse_newmachines
 &worse_newthings &worse_story &worse_decisions &worse_money &worse_finances
&worse_math &worse_reasoning &pbase_familyfriends    &pbetter_newthings
      &pworse_newthings   &pbase_recent &pbase_story  &pbetter_story      &pworse_story
      &pbase_convos &pbase_decisions
 &pbetter_decisions &pworse_decisions   &pbase_addressphone &pbase_money
      &pbetter_money      &pworse_money &pbase_daymonth     &pbase_finances
      &pbetter_finances   &pworse_finances    &pbase_thingskept   &pbase_math
      &pbetter_math &pworse_math  &pbase_findthings
 &pbase_reasoning   &pbetter_reasoning  &pworse_reasoning   &pbase_machines
      &pbetter_familyfriends     &pworse_familyfriends      &pbase_newmachines
      &pbetter_recent     &pworse_recent      &pbase_newthings    &pbetter_convos
      &pworse_convos      &pbetter_addressphone
 &pworse_addressphone      &pbetter_daymonth   &pworse_daymonth    &pbetter_thingskept
      &pworse_thingskept  &pbetter_findthings &pworse_findthings  &pbetter_machines
      &pworse_machines    &pbetter_newmachines       &pworse_newmachines;
*replace missings with 0;
do x=1 to 96;
```

```
if missing(missing_jorm[x]) | missing_jorm[x] = 8 | missing_jorm[x] =
9|missing_jorm[x] = " " then missing_jorm[x] = 0;
end;
drop x;

*for base score, current wave;
array base [16] &base_familyfriends &base_recent &base_convos &base_addressphone
&base_daymonth &base_thingskept &base_findthings &base_machines &base_newmachines
&base_newthings
 &base_story &base_decisions &base_money &base_finances &base_math &base_reasoning;
*for scores that got better, current wave;
array better [16] &better_familyfriends &better_recent &better_convos
&better_addressphone &better_daymonth &better_thingskept &better_findthings
&better_machines &better_newmachines
&better_newthings &better_story &better_decisions &better_money &better_finances
&better_math &better_reasoning;
*for scores that got worse, current wave;
array worse [16] &worse_familyfriends &worse_recent &worse_convos &worse_addressphone
&worse_daymonth &worse_thingskept &worse_findthings &worse_machines &worse_newmachines
&worse_newthings &worse_story &worse_decisions &worse_money &worse_finances
&worse_math &worse_reasoning;
*for the total score;
array pc [16] total_familyfriends total_recent total_convos total_addressphone
total_daymonth total_thingskept total_findthings total_machines total_newmachines
total_newthings
total_story total_decisions total_money total_finances total_math total_reasoning;

*run array for base score, current wave;
do i=1 to 16;
 if base[i] eq 1 and better[i] in (1,2) then pc[i]=better[i];
 else if base[i] eq 2 then pc[i]=3;
 else if base[i] eq 3 and worse[i] in (4,5) then pc[i]=worse[i];
 else pc[i]=0;
 end;
drop i;

*create total score, current wave;
iqtot&w=total_familyfriends+total_recent+total_convos+total_addressphone+total_daymont
h+total_thingskept+total_findthings+total_machines+total_newmachines+total_newthings+
total_story+total_decisions+total_money+total_finances+total_math+total_reasoning;

*creates average score variable, current wave;
avgiq&w = (iqtot&w/16);

*for base score, prior wave;
array pbase [16] &pbase_familyfriends   &pbase_recent &pbase_convos &pbase_addressphone
      &pbase_daymonth     &pbase_thingskept   &pbase_findthings
&pbase_machines     &pbase_newmachines  &pbase_newthings    &pbase_story
      &pbase_decisions    &pbase_money  &pbase_finances     &pbase_math
      &pbase_reasoning;
*for scores that got better, prior wave;
array pbetter [16] &pbetter_familyfriends      &pbetter_newthings  &pbetter_recent
      &pbetter_story      &pbetter_convos     &pbetter_decisions
      &pbetter_addressphone
&pbetter_money      &pbetter_daymonth   &pbetter_finances   &pbetter_thingskept
      &pbetter_math &pbetter_findthings &pbetter_reasoning  &pbetter_machines
&pbetter_newmachines;
*for scores that got worse, prior wave;
array pworse [16] &pworse_familyfriends &pworse_newthings   &pworse_recent
      &pworse_story &pworse_convos      &pworse_decisions   &pworse_addressphone
&pworse_money &pworse_daymonth    &pworse_finances    &pworse_thingskept  &pworse_math
      &pworse_findthings  &pworse_reasoning   &pworse_machines
      &pworse_newmachines;
```

```
*for the total score;
array ppc [16] ptotal_familyfriends     ptotal_story  ptotal_recent ptotal_decisions
       ptotal_convos ptotal_money  ptotal_addressphone ptotal_finances
       ptotal_daymonth
ptotal_math   ptotal_thingskept    ptotal_reasoning    ptotal_findthings
       ptotal_machines     ptotal_newmachines  ptotal_newthings;
*run array for base score, prior wave;
do j=1 to 16;
 if pbase[j] eq 1 and pbetter[j] in (1,2) then ppc[j]=pbetter[j];
 else if pbase[j] eq 2 then ppc[j]=3;
 else if pbase[j] eq 3 and pworse[j] in (4,5) then ppc[j]=pworse[j];
 else ppc[j]=0;
 end;
drop j;

*create total score, prior wave;
iqtot&pw=ptotal_familyfriends+   ptotal_story+ptotal_recent+      ptotal_decisions+
       ptotal_convos+      ptotal_money+ptotal_addressphone+      ptotal_finances+
       ptotal_daymonth+    ptotal_math+  ptotal_thingskept+  ptotal_reasoning+
       ptotal_findthings+  ptotal_machines+    ptotal_newmachines+ ptotal_newthings;

*creates average score variable, prior wave;
avgiq&pw = (iqtot&pw/16);

*create change variable;
ciqtot&pw.&w = avgiq&w-avgiq&pw;

%mend;

*Format: dementia status;
proc format;
value  dementia_status 1 = "1. Dementia"
                       2 = "2. CIND"
                       3 = "3. Normal";
run;


*Macro 7a: Prediction model for self respondents;
%macro hurd_self(age=, edu=, sex=, ADL=, IADL=, cADL=, cIADL=, dates=, bwc20=, s7=,
sci=, cac=, vp=, imr=, delr=,
                 pdates=, pbwc20=, ps7=, psci=, pcac=, pvp=, pimr=,
pdelr=,flagcADL =, cdates = ,
                 cbwc20=, cs7=, csci=, ccac=, cvp=, cimr =, cdelr =,
flag_bwc20=, flag_cbwc20=);

*Step 6. Categorize education and age;
educat = 0;
if &edu <= 1 then educat = 1;
if &edu <4 & &edu > 1 then educat = 2;
if &edu < 8 & &edu >= 4 then educat = 3;

*Step 7. Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Categorical coefficients;
*Age;
if &age >=89 then b_age = -1.082719;
else if &age>=84 & &age<=88 then b_age = -0.733927;
else if &age>=79 & &age<=83 then b_age = -0.503213;
else if &age>=74 & &age<=78 then b_age = 0.000602;
else b_age = 0;
*Education;
if educat = 2 then b_edu = -0.244768;
else if educat = 3 then b_edu = -0.177725;
```

```
else b_edu = 0;

*Binary coefficients;
*Sex;
if &sex = 2 then b_sex = -0.030765;
else b_sex = 0;

*Continuous coefficients;
*ADLS and IADLS;
b_ADL = 0.065182;
b_IADL = -0.223037;
b_cADL = -0.042915;
b_cIADL = 0.13442;
*TICS items;
b_dates = 0.261419;
b_bwc20 = -0.527701;
b_s7 = 0.127647;
b_sci =-0.598867;
b_cac = 0.090591;
b_vp = 0.302807;
b_imr = 0.163782;
b_delr =0.192291;
b_cdates =-0.013743;
b_cbwc20 =0.366754;
b_cs7 = -0.043094;
b_csci = 0.291344;
b_ccac = 0.073087;
b_cvp = -0.073849;
b_cimr = -0.066951;
b_cdelr = -0.083095;
b_mbwc20 =1.124801;
b_mcbwc20 = -0.350093;

*equation;
if &age >= 70 then dementiascore_self = b_age + b_edu + b_sex + b_ADL*&ADL +
b_IADL*&IADL + b_cADL*&cADL + b_cIADL*&cIADL + b_dates*&dates + b_bwc20*&bwc20 +
b_s7*&s7 + b_sci*&sci +
b_cac*&cac + b_vp*&vp + b_imr*&imr + b_delr*&delr + b_cdates*&cdates +
b_cbwc20*&cbwc20 + b_cs7*&cs7 + b_csci*&csci + b_ccac*&ccac + b_cvp*&cvp +
b_cimr*&cimr +
b_cdelr*&cdelr + b_mbwc20*&flag_bwc20 + b_mcbwc20*&flag_cbwc20;

%mend;

*Marco 7b: Prediction model for proxy-respondents;
 %macro hurd_proxy(age=, edu=, sex=, ADL=, IADL=, chADL=, chIADL=, pdates=, pproxy =,
ps7=, pvp =, pimr=, pdelr=, jorm=, cjorm=, flag_pdates=);

*Step 6. Categorize education and age
*Create categorical education level variable;
educat = 0;
if &edu <= 1 then educat = 1;
if &edu <4 & &edu > 1 then educat = 2;
if &edu < 8 & &edu >= 4 then educat = 3;

*Step 7. Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Categorical coefficients;
*Age;
if &age >=89 then b_age = -1.023215;
else if &age>=84 & &age<=88 then b_age = -1.083911;
else if &age>=79 & &age<=83 then b_age = -1.06336;
else if &age>=74 & &age<=78 then b_age = -0.548397;
```

```
else b_age = 0;
*Education;
if educat = 2 then b_edu = -0.454604;
else if educat = 3 then b_edu = -0.460817;
else b_edu = 0;

*Binary coefficients;
*Sex;
if &sex = 2 then b_sex = -0.496928;
else b_sex = 0;

*Coefficients for continuous variables;
b_ADL = -0.04152;
b_IADL = -0.483236;
b_cADL = 0.045943;
b_cIADL = 0.401849;
b_jorm = -0.598973;
b_pproxy = 0.992279;
b_cjorm =0.174425;
b_pdates = 0.420868;
b_ps7 = 0.090047;
b_pvp = -0.511297;
b_pimr = 0.265229;
b_pdelr = -0.131548;
b_mpdates = 1.830266;

*Equation;
if &age >= 70 then dementiascore_proxy = b_age + b_edu + b_sex + b_ADL*&ADL +
b_IADL*&IADL + b_cADL*&chADL + b_cIADL*&chIADL + b_jorm*&jorm + b_pproxy*&pproxy +
b_cjorm*&cjorm
+b_pdates*&pdates+ b_ps7*&ps7 + b_pvp*&pvp + b_pimr*&pimr + b_pdelr*&pdelr +
b_mpdates*&flag_pdates;


 %mend;

*Macro 7c: create probability of dementia and dementia category from prediction model;
%macro predictedprobability(w);

*(Ordered probit model cutpoints self: c1 = -0.1110266, c2 = 1.207072);
*(Ordered probit model cutpoints proxy: c1 = -1.083051, c2 = 0.2425114);

*DEMENTIA PROBABILITY;
*SELF RESPONDENTS;
D_predictedprob_self = 0;
if R&w.PROXY = 0 then D_predictedprob_self = cdf("NORMAL", -0.1110266-
dementiascore_self);
*PROXY RESPONDENTS;
D_predictedprob_proxy = 0;
if R&w.PROXY = 1 then D_predictedprob_proxy = cdf("NORMAL", -1.083051-
dementiascore_proxy);
*combine;
if R&w.PROXY = 0 then D_predictedprob&w = D_predictedprob_self;
else if R&w.PROXY = 1 then D_predictedprob&w = D_predictedprob_proxy;

*CIND PROBABILITY;
*SELF RESPONDENTS;
C_predictedprob_self = 0;
if R&w.PROXY = 0 then C_predictedprob_self = cdf("NORMAL", 1.207072-
dementiascore_self) - cdf("NORMAL", -0.1110266-dementiascore_self);
*PROXY RESPONDENTS;
C_predictedprob_proxy = 0;
```

```
if R&w.PROXY = 1 then C_predictedprob_proxy = cdf("NORMAL", 0.2425114-
dementiascore_proxy) - cdf("NORMAL", -1.083051-dementiascore_proxy);
*combine;
if R&w.PROXY = 0 then C_predictedprob&w = C_predictedprob_self;
else if R&w.PROXY = 1 then C_predictedprob&w = C_predictedprob_proxy;

*NORMAL PROBABILITY;
*SELF RESPONDENTS;
N_predictedprob_self = 0;
if R&w.PROXY = 0 then N_predictedprob_self = 1 - cdf("NORMAL", 1.207072-
dementiascore_self);
*PROXY RESPONDENTS;
N_predictedprob_proxy = 0;
if R&w.PROXY = 1 then N_predictedprob_proxy = 1 - cdf("NORMAL", 0.2425114-
dementiascore_proxy);
*combine;
if R&w.PROXY = 0 then N_predictedprob&w = N_predictedprob_self;
else if R&w.PROXY = 1 then N_predictedprob&w = N_predictedprob_proxy;

*Categorize into Dementia, CIND, Normal - highest probability value = category;
Dementia_cat&w = .;
if R&w.AGEY_E >= 70 & D_predictedprob&w >= C_predictedprob&w & D_predictedprob&w >=
N_predictedprob&w then Dementia_cat&w = 1;
if R&w.AGEY_E & C_predictedprob&w > D_predictedprob&w & C_predictedprob&w >=
N_predictedprob&w then Dementia_cat&w = 2;
if R&w.AGEY_E & N_predictedprob&w > C_predictedprob&w & N_predictedprob&w >
D_predictedprob&w then Dementia_cat&w = 3;

format Dementia_cat&w dementia_status.;

%mend;

*Macro 7d: create imputed dementia category such that if participant has received
dementia prediction in prior wave, imputed category is dementia in every subsequent
wave he/she participates in. If participant has not previously received a dementia
prediction, imputed category is equal to predicted category;
%macro onceDimputeD(w);
Imputed_dementiacat&w = .;
if INW&w = 1 AND Dementia_cat&w = 1
      then dementia_seen = 1;
if INW&w = 1 AND Dementia_cat&w = 1
      then Imputed_dementiacat&w = 1;
else if INW&w = 1 and dementia_seen = 1
      then Imputed_dementiacat&w = 1;
else if INW&w = 1 and dementia_seen = .
      then Imputed_dementiacat&w = Dementia_cat&w;
%mend;
```

```
/*************************************************************************************
************\
//*********************************       PART 2: Data import
**********************************\\
/*************************************************************************************
************\

Programmed by: Joanna Emerson
Date: March 20, 2019 (updated April 22, 2019 by Flora Berklein)

Purpose: Import SAS datasets for ADAMS and HRS waves

Datasets included:
- ADAMS Wave A (HRS restricted): Final diagnosis
- ADAMS Tracker file (HRS restricted): Proxy use, age, assessment status
- RAND HRS Longitudinal File: Demographics, TICS scores, functional limitations
- HRS FAT File: 1998-2014: Jorm IQCODE questions

/*************************************************************************************
************/

/*************************************************************************************
************\
                          Note: ADAMS data access is restricted by the HRS and
University of Michigan.
                                                              Import code is not
included here
/*************************************************************************************
************/


**********FULL HRS RAND**********
*import RAND full dataset with formats;
DATA RANDfull LIBNAME LIBRARY '\userdata\...\sasfmts.sas7bdat';
*new RAND data;
SET '\userdata\...\randhrs1992_2014v2.sas7bdat';
run;
proc sort data=RANDfull;
by HHIDPN;
run;
**********WAVE 4**********
*import RAND fat files - 98 file;
data HRSfat98;
*SET "file path to h98f2c.sas7bdat";
keep HHIDPN F1389-F1459;
run;
proc sort data=HRSfat98;
by HHIDPN;
run;
*Merge: only those who participated in wave 4;
data HRS.wave4;
set RANDfull (where=(R4IWSTAT = 1));
merge HRSfat98;
by HHIDPN;
run;

**********WAVE 5**********;
data RANDfat00;
*SET "file path to h00f1c.sas7bdat";
keep HHIDPN G1543-G1613;
run;
proc sort data=RANDfat00;
by HHIDPN;
```

```
run;
*Merge - only those who participated in wave 5;
data HRS.wave5;
set RANDfull (where=(R5IWSTAT = 1));
merge RANDfat00;
by HHIDPN;
run;

**********WAVE 6**********;
data HRSfat02;
*SET "file path to h02f2c.sas7bdat';
keep HHIDPN HD506-HD553;
run;
proc sort data=HRSfat02;
by HHIDPN;
run;
*Merge - only those who participated in wave 6;
data HRS.wave6;
set RANDfull (where=(R6IWSTAT = 1));
merge HRSfat02;
by HHIDPN;
run;

**********WAVE 7**********;
data HRSfat04;
*SET "file path to h04f1b.sas7bdat';
keep HHIDPN JD506-JD553;
run;
proc sort data=HRSfat04;
by HHIDPN;
run;
*Merge - only those who participated in wave 7;
data HRS.wave7;
set RANDfull (where=(R7IWSTAT = 1));
merge HRSfat04;
by HHIDPN;
run;

**********WAVE 8**********;
data HRSfat06;
*SET "file path to h06f3a.sas7bdat';
keep HHIDPN KD506-KD553;
run;
proc sort data=HRSfat06;
by HHIDPN;
run;
*Merge - only those who participated in wave 8;
data HRS.wave8;
set RANDfull (where=(R8IWSTAT = 1));
merge HRSfat06;
by HHIDPN;
run;

**********WAVE 9**********;
data HRSfat08;
*SET "file path to h08f3a.sas7bdat';
keep HHIDPN LD506-LD553;
run;
proc sort data=HRSfat08;
by HHIDPN;
run;
*Merge - only those who participated in wave 9;
data HRS.wave9;
```

```
set RANDfull (where=(R9IWSTAT = 1));
merge HRSfat08;
by HHIDPN;
run;

**********WAVE 10**********;
data HRSfat10;
*SET "file path to hd10f5d.sas7bdat';
keep HHIDPN MD506-MD553;
run;
proc sort data=HRSfat10;
by HHIDPN;
run;
*Merge - only those who participated in wave 10;
data HRS.wave10;
set RANDfull (where=(R10IWSTAT = 1));
merge HRSfat10;
by HHIDPN;
run;

**********WAVE 11**********;
data HRSfat12;
*SET "file path to h12f1a.sas7bdat';
keep HHIDPN ND506-ND553;
run;
proc sort data=HRSfat12;
by HHIDPN;
run;
*Merge - only those who participated in wave 11;
data HRS.wave11;
set RANDfull (where=(R11IWSTAT = 1));
merge HRSfat12;
by HHIDPN;
run;

**********WAVE 12**********;
data HRSfat14;
*SET "file path to h14f2a.sas7bdat';
keep HHIDPN OD506-OD553;
run;
proc sort data=HRSfat14;
by HHIDPN;
run;
*Merge - only those who participated in wave 12;
data HRS.wave12;
set RANDfull (where=(R12IWSTAT = 1));
merge HRSfat14;
by HHIDPN;
run;
```

```
/*******************************************************************************
************\
//*********************************       PART 3: ADAMS dataset
*********************************\\
///************************     Data cleaning and ordered probit model
************************\\\
/*******************************************************************************
************\

Programmed by: Joanna Emerson
Date: March 19, 2019 (updated April 22, 2019 by Flora Berklein)

Purpose: Create an orded probit model of ADAMS respondent dementia diagnosis,
              the coefficients for which will be used to predict dementia status of
all HRS respondents

Based on Hurd ordered probit model from 2010 NEJM article:
       http://hrsonline.isr.umich.edu/index.php?p=shoavail&iyear=CB

Datasets uses:
- ADAMS Wave A (HRS restricted): Final diagnosis
- ADAMS Tracker file (HRS restricted): Proxy use, age, assessment status
- HRS Wave 4 (1998) (RAND HRS Longitudinal File): Demographics, TICS scores,
functional limitations
- HRS Wave 5 (2000) (RAND HRS Longitudinal File): Demographics, TICS scores,
functional limitations
- HRS Wave 6 (2002) (RAND HRS Longitudinal File): Demographics, TICS scores,
functional limitations

Steps:
Data Step (separate for those selected into ADAMS from 2000 versus 2002 HRS wave):
       1. Merge datasets
       2. Categorize final diagnosis variable into categorical dementia status
(Dementia, CIND, Normal)
       3. Recode non-binary TICS answers into 1 = correct, 0 = incorrect
       4. Create missing indicators for TICS items and replace missing values with 0
       5. Create change in TICS score variables for each TICS item
       6. Categorize education and age
       7. Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables
       8. Combine datasets of those selected into ADAMS from 2000 HRS and 2002 HRS
       9. Create model coefficient variables for TICS items that vary depending on
ADAMS initiation date
       10. Remove respondents with missing values for all TICS items
Ordered probit models (separate for self-respondents versus proxy-respondents):
       11. Run ordered probit model with proc qlim:
             Outcome: Categorical dementia status from ADAMS (Dementia, CIND, Normal)
             Predictors: demographics, functional limitations, TICS items from HRS
       12. Output predicted probabilities of each categorical outcome (Dementia, CIND,
Normal)
       13. Data step: Categorize respondents into Dementia, CIND, Normal per predicted
probabilities


/*******************************************************************************
************/


/*********************************      DATASETS
*******************************/

*Dataset 1: for those selected from HRS 2000 into ADAMS;
data work.ADAMS00;
```

```
*STEP 1. Merge datasets;
merge HRS.ADAMS1TRK_R (in=a) work.ADAMSa (in=b) HRS.wave5 (in=c) HRS.wave4 (in=d);
by HHIDPN;
*Include only if assessed in Wave A and selected in 2000;
if AASSESS = 1 & WAVESEL = 1;

*STEP 2. Categorize final diagnosis variable into categorical dementia status (1
=Dementia, 2=CIND, 3=Normal);
if ADFDX1 > 0 & ADFDX1 < 20 | ADFDX1 = 32 then dementia_diagnosis = 1;
else if ADFDX1 >= 20 & ADFDX1 < 31 | ADFDX1 = 33 then dementia_diagnosis = 2;
else if ADFDX1 = 31 then dementia_diagnosis = 3;
format dementia_diagnosis dementia_status.;

*STEP 3. Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=5, pw=4);

*STEP 4. Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=5, pw=4);

*STEP 5.  Create change in TICS score variables for each TICS item;
%TICS_change(w=5, pw=4);

*STEP 6. Categorize education and age;
*Categorical education level variable: LT HS= 3 (ref), HS = 2, MT HS = 1;
if RAEDEGRM <= 1 then educat = 3;
if RAEDEGRM <4 & RAEDEGRM > 1 then educat = 2;
if RAEDEGRM < 8 & RAEDEGRM >= 4 then educat = 1;

*Categorical age variable: 90+ = 1, 85-89 = 2, 80-84 = 3, 75-79=4, lt 75 = 5 (ref);
if AAGE >=90 then age_cat = 1;
else if AAGE>=85 & AAGE<=89 then age_cat = 2;
else if AAGE>=80 & AAGE<=84 then age_cat = 3;
else if AAGE>=75 & AAGE<=79 then age_cat = 4;
else if AAGE<75 then age_cat = 5;

*Replace proxy = 0 if proxy is missing;
if R4PROXY = . then R4PROXY = 0;
if R5PROXY = . then R5PROXY = 0;
if PROXY = . then PROXY = 0;

*STEP 7. Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R5EAT,ADLbed=R5BED,ADLtoilt=R5TOILT,ADLdress=R5DRESS,ADLbath=R5BATH,ADLwa
lk=R5WALKR,IADLmeal=R5MEALS,

IADLshop=R5SHOPA,IADLphone=R5PHONEA,IADLmeds=R5MEDSA,IADLmoney=R5MONEYA,pADLeat=R4EAT,
pADLbed=R4BED,pADLtoilt=R4TOILT,pADLdress=R4DRESS,pADLbath=R4BATH,

pADLwalk=R4WALKR,pIADLmeal=R4MEALS,pIADLshop=R4SHOPA,pIADLphone=R4PHONEA,pIADLmeds=R4M
EDSA,pIADLmoney=R4MONEYA,
        w=5, pw=4);

*Dates;
%dates(day=R5DY, month=R5MO, year=R5YR, dayweek=R5DW, pday=R4DY, pmonth=R4MO,
pyear=R4YR, pdayweek=R4DW, w=5, pw=4);

*Jorm IQCODE;
%jorm(base_familyfriends=G1543,      base_recent=G1548,      base_convos=G1553,
base_addressphone =G1558,  base_daymonth=G1563,       base_thingskept=G1568,
base_findthings=G1573,         base_machines=G1578,   base_newmachines =G1583,
       base_newthings=G1588,      base_story =G1593,    base_decisions=G1598,
```

```
base_money =G1602,      base_finances=G1605,   base_math =G1608,
       base_reasoning=G1611,
      better_familyfriends=G1544, better_recent=G1549, better_convos=G1554,
       better_addressphone =G1559,      better_daymonth=G1564,
       better_thingskept=G1569, better_findthings=G1574,     better_machines=G1579,
       better_newmachines =G1584, better_newthings=G1589,    better_story =G1594,
better_decisions=G1599, better_money =G1603,   better_finances=G1606,     better_math
=G1609,       better_reasoning =G1612,
      worse_familyfriends=G1545,  worse_recent=G1550,  worse_convos=G1555,
       worse_addressphone =G1560, worse_daymonth=G1565,      worse_thingskept=G1570,
worse_findthings=G1575,    worse_machines=G1580,      worse_newmachines =G1585,
       worse_newthings=G1590,     worse_story =G1595,  worse_decisions=G1600,
worse_money =G1604, worse_finances=G1607,      worse_math =G1610,
       worse_reasoning=G1613,
       pbase_familyfriends=F1389,   pbase_recent=F1394,      pbase_convos=F1399,
pbase_addressphone =F1404, pbase_daymonth=F1409,      pbase_thingskept=F1414,
pbase_findthings=F1419,       pbase_machines=F1424,   pbase_newmachines =F1429,
       pbase_newthings=F1434,     pbase_story =F1439,     pbase_decisions=F1444,
       pbase_money =F1448,     pbase_finances=F1451,  pbase_math =F1454,
       pbase_reasoning=F1457,
      pbetter_familyfriends=F1390, pbetter_recent=F1395,    pbetter_convos=F1400,
       pbetter_addressphone =F1405,      pbetter_daymonth=F1410,
       pbetter_thingskept=F1415,  pbetter_findthings=F1420,  pbetter_machines=F1425,
       pbetter_newmachines =F1430,       pbetter_newthings=F1435,   pbetter_story
=F1440,       pbetter_decisions=F1445,   pbetter_money =F1449,
       pbetter_finances=F1452,    pbetter_math =F1455,       pbetter_reasoning =F1458,
      pworse_familyfriends=F1391,  pworse_recent=F1396,     pworse_convos=F1401,
pworse_addressphone =F1406,       pworse_daymonth=F1411,    pworse_thingskept=F1416,
pworse_findthings=F1421,       pworse_machines=F1426, pworse_newmachines =F1431,
       pworse_newthings=F1436,    pworse_story =F1441,
pworse_decisions=F1446,    pworse_money =F1450,          pworse_finances=F1453,
       pworse_math =F1456, pworse_reasoning=F1459,
       w=5, pw=4);

keep HHIDPN RAHHIDPN RAEDEGRM RAGENDER raedyrs RACE ETHNIC age_cat educat
dementia_diagnosis educat female AAGE AASSESS WAVESEL ADFDX1 PROXY
        R4PROXY R4EAT R4BED R4TOILT R4DRESS R4BATH R4WALKR R4MEALS R4SHOPA R4PHONEA
R4MEDSA R4MONEYA R4DY R4MO R4YR R4DW R4BWC20 R4SER7 R4SCIS R4CACT R4VP R4IMRC R4DLRC
R4AGEY_E INW4
        R5PROXY R5EAT R5BED R5TOILT R5DRESS R5BATH R5WALKR R5MEALS R5SHOPA R5PHONEA
R5MEDSA R5MONEYA R5DY R5MO R5YR R5DW R5BWC20 R5SER7 R5SCIS R5CACT R5VP R5IMRC R5DLRC
R5AGEY_E INW5
        ADLcomposite5 ADLcomposite4 IADLcomposite5 IADLcomposite4 ADLchange45
IADLchange45 flag_ADL5 ADLchange45 IADLchange45
        dates5 dates4 cdates45 flag_dates5 flag_dates4 G1543-G1613 F1389-F1459
        iqtot5 avgiq5 iqtot4 avgiq4 ciqtot45
        flag_bwc205 flag_bwc204 flag_s75 flag_s74 flag_sci5 flag_sci4 flag_imr5
flag_imr4 flag_cac5 flag_cac4 flag_vp5 flag_vp4 flag_delr5 flag_delr4
        flag_cbwc2045 flag_cs745 flag_cimr45 flag_cdelr45 cbwc2045 cs745 csci45 ccac45
cvp45 cimr45 cdelr45;

rename flag_ADL5=flag_ADL flag_dates5=flag_dates flag_dates4 = flag_pdates
flag_bwc205=flag_bwc20 flag_bwc204=flag_pbwc20
          flag_s75=flag_s7 flag_s74=flag_ps7 flag_sci5=flag_sci flag_sci4=flag_psci
flag_imr5=flag_imr flag_imr4=flag_pimr flag_cac5=flag_cac flag_cac4=flag_pcac
          flag_vp5=flag_vp flag_vp4=flag_pvp flag_delr5=flag_delr
flag_delr4=flag_pdelr flag_cbwc2045=flag_cbwc20 flag_cs745=flag_cs7
flag_cimr45=flag_cimr flag_cdelr45=flag_cdelr
          ADLcomposite5=ADLcomposite ADLcomposite4=pADLcomposite
IADLcomposite5=IADLcomposite IADLcomposite4=pIADLcomposite ADLchange45=ADLchange
          IADLchange45=IADLchange flag_ADL5=flag_ADL flag_ADL4=flag_pADL
ADLchange45=ADLchange IADLchange45=IADLchange45
```

```
         dates5=dates dates4=pdates cdates45=cdates flag_dates5=flag_dates
flag_dates4=flag_pdates
         iqtot5=iqtot avgiq5=avgiq iqtot4=piqtot avgiq4=pavgiq ciqtot45=ciqtot
         cbwc2045=cbwc20 cs745=cs7 csci45=csci ccac45=ccac cvp45=cvp cimr45=cimr
cdelr45=cdelr;

run;

*Dataset 2: for those selected from HRS 2002 into ADAMS;
data work.ADAMS02;
*STEP 1. Merge datasets;
merge HRS.ADAMS1TRK_R (in=a) work.ADAMSa (in=b) HRS.wave6 (in=c) HRS.wave5 (in=d);
by HHIDPN;
*Include only if assessed in Wave A and selected in 2002;
if AASSESS = 1 & WAVESEL = 2;

*STEP 2. Categorize final diagnosis variable into categorical dementia status (1
=Dementia, 2=CIND, 3=Normal);
if ADFDX1 > 0 & ADFDX1 < 20 | ADFDX1 = 32 then dementia_diagnosis = 1;
else if ADFDX1 >= 20 & ADFDX1 < 31 | ADFDX1 = 33 then dementia_diagnosis = 2;
else if ADFDX1 = 31 then dementia_diagnosis = 3;
format dementia_diagnosis dementia_status.;

*STEP 3. Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=6, pw=5);

*STEP 4. Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=6, pw=5);

*STEP 5.  Create change in TICS score variables for each TICS item;
%TICS_change(w=6, pw=5);

*STEP 6. Categorize education and age;
*Categorical education level variable: LT HS= 3 (ref), HS = 2, MT HS = 1;
if RAEDEGRM <= 1 then educat = 3;
if RAEDEGRM <4 & RAEDEGRM > 1 then educat = 2;
if RAEDEGRM < 8 & RAEDEGRM >= 4 then educat = 1;

*Categorical age variable: 90+ = 1, 85-89 = 2, 80-84 = 3, 75-79=4, lt 75 = 5 (ref);
if AAGE >=90 then age_cat = 1;
else if AAGE>=85 & AAGE<=89 then age_cat = 2;
else if AAGE>=80 & AAGE<=84 then age_cat = 3;
else if AAGE>=75 & AAGE<=79 then age_cat = 4;
else if AAGE<75 then age_cat = 5;

*Replace proxy = 0 if proxy is missing;
if R5PROXY = . then R5PROXY = 0;
if R6PROXY = . then R6PROXY = 0;
if PROXY = . then PROXY = 0;

*STEP 7. Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R6EAT,ADLbed=R6BED,ADLtoilt=R6TOILT,ADLdress=R6DRESS,ADLbath=R6BATH,ADLwa
lk=R6WALKR,IADLmeal=R6MEALS,

IADLshop=R6SHOPA,IADLphone=R6PHONEA,IADLmeds=R6MEDSA,IADLmoney=R6MONEYA,pADLeat=R5EAT,
pADLbed=R5BED,pADLtoilt=R5TOILT,pADLdress=R5DRESS,pADLbath=R5BATH,

pADLwalk=R5WALKR,pIADLmeal=R5MEALS,pIADLshop=R5SHOPA,pIADLphone=R5PHONEA,pIADLmeds=R5M
EDSA,pIADLmoney=R5MONEYA,
        w=6, pw=5);
```

```
*Dates;
%dates(day=R6DY, month=R6MO, year=R6YR, dayweek=R6DW, pday=R5DY, pmonth=R5MO,
pyear=R5YR, pdayweek=R5DW, w=6, pw=5);

*Jorm IQCODE;
%jorm(base_familyfriends=HD506,      base_recent=HD509,      base_convos=HD512,
base_addressphone =HD515,  base_daymonth=HD518,       base_thingskept=HD521,
base_findthings=HD524,         base_machines=HD527,   base_newmachines =HD530,
      base_newthings=HD533,      base_story =HD536,   base_decisions=HD539,
base_money =HD542,      base_finances=HD545,   base_math =HD548,
      base_reasoning=HD551,
      better_familyfriends=HD507, better_recent=HD510, better_convos=HD513,
      better_addressphone =HD516,       better_daymonth=HD519,
      better_thingskept=HD522, better_findthings=HD525,     better_machines=HD528,
      better_newmachines =HD531, better_newthings=HD534,    better_story =HD537,
better_decisions=HD540, better_money =HD543,   better_finances=HD546,     better_math
=HD549,       better_reasoning =HD552,
      worse_familyfriends=HD508, worse_recent=HD511,  worse_convos=HD514,
      worse_addressphone =HD517, worse_daymonth=HD520,      worse_thingskept=HD523,
worse_findthings=HD526,    worse_machines=HD529,      worse_newmachines =HD532,
      worse_newthings=HD535,     worse_story =HD538,  worse_decisions=HD541,
worse_money =HD544, worse_finances=HD547,      worse_math =HD550,
      worse_reasoning=HD553,
pbase_familyfriends=G1543,     pbase_recent=G1548,     pbase_convos=G1553,
pbase_addressphone =G1558, pbase_daymonth=G1563,      pbase_thingskept=G1568,
pbase_findthings=G1573,        pbase_machines=G1578,  pbase_newmachines =G1583,
      pbase_newthings=G1588,     pbase_story =G1593,  pbase_decisions=G1598,
pbase_money =G1602,     pbase_finances=G1605,  pbase_math =G1608,
      pbase_reasoning=G1611,
      pbetter_familyfriends=G1544,     pbetter_recent=G1549, pbetter_convos=G1554,
      pbetter_addressphone =G1559,     pbetter_daymonth=G1564,
      pbetter_thingskept=G1569, pbetter_findthings=G1574,   pbetter_machines=G1579,
      pbetter_newmachines =G1584,      pbetter_newthings=G1589,   pbetter_story
=G1594, pbetter_decisions=G1599, pbetter_money =G1603,      pbetter_finances=G1606,
      pbetter_math =G1609,      pbetter_reasoning =G1612,
      pworse_familyfriends=G1545, pworse_recent=G1550,        pworse_convos=G1555,
      pworse_addressphone =G1560,      pworse_daymonth=G1565,
      pworse_thingskept=G1570,    pworse_findthings=G1575,  pworse_machines=G1580,
      pworse_newmachines =G1585, pworse_newthings=G1590,    pworse_story =G1595,
pworse_decisions=G1600,    pworse_money =G1604,      pworse_finances=G1607,
      pworse_math =G1610, pworse_reasoning=G1613, w=6, pw=5);

keep HHIDPN RAHHIDPN RAEDEGRM RAGENDER raedyrs RACE ETHNIC age_cat educat
dementia_diagnosis educat female AAGE AASSESS WAVESEL ADFDX1 PROXY
       R5PROXY R5EAT R5BED R5TOILT R5DRESS R5BATH R5WALKR R5MEALS R5SHOPA R5PHONEA
R5MEDSA R5MONEYA R5DY R5MO R5YR R5DW R5BWC20 R5SER7 R5SCIS R5CACT R5VP R5IMRC R5DLRC
R5AGEY_E INW5
       R6PROXY R6EAT R6BED R6TOILT R6DRESS R6BATH R6WALKR R6MEALS R6SHOPA R6PHONEA
R6MEDSA R6MONEYA R6DY R6MO R6YR R6DW R6BWC20 R6SER7 R6SCIS R6CACT R6VP R6IMRC R6DLRC
R6AGEY_E INW6
       ADLcomposite6 ADLcomposite5 IADLcomposite6 IADLcomposite5 ADLchange56
IADLchange56 flag_ADL6 flag_ADL5 ADLchange56 IADLchange56
       dates6 dates5 cdates56 flag_dates6 flag_dates5
       iqtot6 avgiq6 iqtot5 avgiq5 ciqtot56
       flag_bwc206 flag_bwc205 flag_s76 flag_s75 flag_sci6 flag_sci5 flag_imr6
flag_imr5 flag_cac6 flag_cac5 flag_vp6 flag_vp5 flag_delr6 flag_delr5
       flag_cbwc2056 flag_cs756 flag_cimr56 flag_cdelr56 cbwc2056 cs756 csci56 ccac56
cvp56 cimr56 cdelr56;

*Rename variables that must merge between waves;
rename flag_ADL6=flag_ADL flag_dates6=flag_dates flag_dates5 = flag_pdates
flag_bwc206=flag_bwc20 flag_bwc205=flag_pbwc20
```

```
         flag_s76=flag_s7 flag_s75=flag_ps7 flag_sci6=flag_sci flag_sci5=flag_psci
flag_imr6=flag_imr flag_imr5=flag_pimr flag_cac6=flag_cac flag_cac5=flag_pcac
         flag_vp6=flag_vp flag_vp5=flag_pvp flag_delr6=flag_delr
flag_delr5=flag_pdelr flag_cbwc2056=flag_cbwc20 flag_cs756=flag_cs7
flag_cimr56=flag_cimr flag_cdelr56=flag_cdelr
         ADLcomposite6=ADLcomposite ADLcomposite5=pADLcomposite
IADLcomposite6=IADLcomposite IADLcomposite5=pIADLcomposite ADLchange56=ADLchange
         IADLchange56=IADLchange flag_ADL6=flag_ADL flag_ADL5=flag_pADL
ADLchange56=ADLchange IADLchange56=IADLchange56
         dates6=dates dates5=pdates cdates56=cdates flag_dates6=flag_dates
flag_dates5=flag_pdates
         iqtot6=iqtot avgiq6=avgiq iqtot5=piqtot avgiq5=pavgiq ciqtot56=ciqtot
         cbwc2056=cbwc20 cs756=cs7 csci56=csci ccac56=ccac cvp56=cvp cimr56=cimr
cdelr56=cdelr;

run;

*STEP 8. Combine datasets of those selected into ADAMS from 2000 HRS and 2002 HRS;
data probitmodel;
merge ADAMS00 ADAMS02;
by HHIDPN;

*STEP 9. Create model coefficient variables for TICS items that vary depending on
ADAMS initiation date;
*Coefficients for current wave;
array coef{21} R5BWC20 R5SER7 R5SCIS R5CACT R5VP R5IMRC R5DLRC R6BWC20 R6SER7 R6SCIS
R6CACT R6VP R6IMRC R6DLRC bwc20 s7 sci cac vp imr delr;
      do i=1 to 7;
      j = i+7;
      k = i+14;
      if WAVESEL = 1 then coef{k}=coef{i};
      else if WAVESEL = 2 then coef{k}=coef{j};
end;
drop i j k;

*Coefficients for 2 waves prior to ADAMS assessment - for proxies only;
array coef_p{15} R4PROXY R4SER7 R4VP R4IMRC R4DLRC R5PROXY R5SER7 R5VP R5IMRC R5DLRC
pPROXY ps7 pvp pimr pdelr;
      do i=1 to 5;
      j = i+5;
      k = i+10;
      if WAVESEL = 1 then coef_p{k}=coef_p{i};
      else if WAVESEL = 2 then coef_p{k}=coef_p{j};
end;
drop i j k;

*STEP 10. Remove respondents with missing values for all TICS items that do not use a
proxy;
missings= flag_dates +flag_pdates+ flag_bwc20 +flag_pbwc20 +flag_s7+ flag_ps7
+flag_sci+ flag_psci +flag_imr +flag_pimr +flag_cac +flag_pcac+ flag_vp+ flag_pvp+
flag_delr +flag_pdelr +flag_cbwc20 +flag_cs7 +flag_cimr +flag_cdelr;
if missings = 20 & PROXY = 0 then DELETE;

run;



/***********************************      MODELS
*******************************/

*Step 11. Run ordered probit model with proc qlim:
            Outcome: Categorical dementia status from ADAMS (Dementia, CIND, Normal)
            Predictors: demographics, functional limitations, TICS items from HRS
```

```
*Ordered probit model, self respondents;
proc qlim data = work.probitmodel;
where PROXY = 0;
class age_cat educat;
model dementia_diagnosis = age_cat educat RAGENDER
                                          ADLcomposite IADLcomposite ADLchange
IADLchange
                                          dates cdates bwc20 s7 sci cac vp imr delr
                                          cbwc20 cs7 csci ccac cvp cimr cdelr
                                          flag_bwc20 flag_cbwc20
                                          / discrete;

*Step 12. Output predicted probabilities of each categorical outcome (Dementia, CIND,
Normal);
output out=probpredictions_self(keep = HHIDPN Prob1_dementia_diagnosis
Prob2_dementia_diagnosis Prob3_dementia_diagnosis) proball;
run;

*Step 11. Run ordered probit model with proc qlim:
            Outcome: Categorical dementia status from ADAMS (Dementia, CIND, Normal)
            Predictors: demographics, functional limitations, TICS items from HRS
*Ordered probit model: proxy respondents;
proc qlim data = work.probitmodel;
where PROXY = 1;
class age_cat educat;
model dementia_diagnosis = age_cat educat RAGENDER
                                          ADLcomposite IADLcomposite ADLchange
IADLchange
                                          avgiq pPROXY ciqtot pdates ps7 pvp pimr

pdelr flag_pdates
                                          / discrete;
*Step 12. Output predicted probabilities of each categorical outcome (Dementia, CIND,
Normal);
output out=probpredictions_proxy(keep = HHIDPN Prob1_dementia_diagnosis
Prob2_dementia_diagnosis Prob3_dementia_diagnosis) proball;
run;

*Step 13. Data step: Categorize respondents into Dementia, CIND, Normal per predicted
probabilities;
data ADAMS_dementia_predictions;
merge work.probpredictions_self (in=a) work.probpredictions_proxy (in=b);
by HHIDPN;
if a|b;

*Categorize dementia status: highest probability = category (1=Dementia, 2=CIND,
3=Normal);
if Prob1_dementia_diagnosis > Prob2_dementia_diagnosis & Prob1_dementia_diagnosis >
Prob3_dementia_diagnosis then dementia_category = 1;
if Prob2_dementia_diagnosis > Prob1_dementia_diagnosis & Prob2_dementia_diagnosis >
Prob3_dementia_diagnosis then dementia_category = 2;
if Prob3_dementia_diagnosis > Prob1_dementia_diagnosis & Prob3_dementia_diagnosis >
Prob2_dementia_diagnosis then dementia_category = 3;
format dementia_category dementia_status.;

*Rename variables;
rename dementia_category=Dementia_cat Prob1_dementia_diagnosis=Prob_dementia
Prob2_dementia_diagnosis=Prob_cind Prob3_dementia_diagnosis=Prob_normal;

run;
```

```
/*************************************************************************************
************\
//*********************************       PART 4: HRS datasets
*********************************\\
///***********************     Data cleaning and dementia predictions
***********************\\\
/*************************************************************************************
************\

Programmed by Jojo Emerson, March 2019. Updated by Flora Berklein, April 22, 2019.
Datasets uses:
- RAND HRS Longitudinal File 2014 (V2): Demographics, TICS scores, functional
limitations
- HRS FAT Files: 1998-2014: Jorm IQCODE questions

Purpose: Predict dementia status of all HRS respondents using ordered probit model

Steps:
Data Step (separate for each wave):
      1. Merge datasets (prior wave and current wave)
      2. Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables
      3. Recode non-binary TICS answers into 1 = correct, 0 = incorrect
      4. Create missing indicators for TICS items and replace missing values with 0
      5. Create change in TICS score variables for each TICS item
      6. Categorize education and age
      7. Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents)
      8. Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal
Data Step (combined data):
      9. Drop those who go from Dementia to Normal between waves
    10. If respondent has one dementia prediction, replace remaining categorical
predictions to Dementia

/*************************************************************************************
************/

/*******************************      CROSS SECTIONAL DATASETS
*****************************/
**WAVE 5**;
*Sort datasets by unique ID;
proc sort data = HRS.wave5;
by HHIDPN;
run;
proc sort data = HRS.wave4;
by HHIDPN;
run;

*Step 1. Merge datasets (prior wave and current wave);
data predictions5;
merge HRS.wave4 (in=a) HRS.wave5 (in=b);
by HHIDPN;
if a & b;

*Step 2 Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R5EAT,ADLbed=R5BED,ADLtoilt=R5TOILT,ADLdress=R5DRESS,ADLbath=R5BATH,ADLwa
lk=R5WALKR,IADLmeal=R5MEALS,
IADLshop=R5SHOPA,IADLphone=R5PHONEA,IADLmeds=R5MEDSA,IADLmoney=R5MONEYA,pADLeat=R4EAT,
pADLbed=R4BED,pADLtoilt=R4TOILT,pADLdress=R4DRESS,pADLbath=R4BATH,
```

```
pADLwalk=R4WALKR,pIADLmeal=R4MEALS,pIADLshop=R4SHOPA,pIADLphone=R4PHONEA,pIADLmeds=R4M
EDSA,pIADLmoney=R4MONEYA, w=5, pw=4);

*Dates;
%dates(day=R5DY, month=R5MO, year=R5YR, dayweek=R5DW, pday=R4DY, pmonth=R4MO,
pyear=R4YR, pdayweek=R4DW, w=5, pw=4);

%jorm(base_familyfriends=G1543,      base_recent=G1548,      base_convos=G1553,
base_addressphone =G1558,  base_daymonth=G1563,       base_thingskept=G1568,
base_findthings=G1573,         base_machines=G1578,   base_newmachines =G1583,
      base_newthings=G1588,      base_story =G1593,   base_decisions=G1598,
base_money =G1602,      base_finances=G1605,   base_math =G1608,
      base_reasoning=G1611,
      better_familyfriends=G1544, better_recent=G1549, better_convos=G1554,
      better_addressphone =G1559,       better_daymonth=G1564,
      better_thingskept=G1569, better_findthings=G1574,     better_machines=G1579,
      better_newmachines =G1584, better_newthings=G1589,    better_story =G1594,
better_decisions=G1599, better_money =G1603,   better_finances=G1606,    better_math
=G1609,      better_reasoning =G1612,
      worse_familyfriends=G1545,  worse_recent=G1550,  worse_convos=G1555,
      worse_addressphone =G1560, worse_daymonth=G1565,      worse_thingskept=G1570,
worse_findthings=G1575,    worse_machines=G1580,     worse_newmachines =G1585,
      worse_newthings=G1590,     worse_story =G1595,  worse_decisions=G1600,
worse_money =G1604, worse_finances=G1607,      worse_math =G1610,
      worse_reasoning=G1613,
      pbase_familyfriends=F1389,   pbase_recent=F1394,     pbase_convos=F1399,
pbase_addressphone =F1404, pbase_daymonth=F1409,      pbase_thingskept=F1414,
pbase_findthings=F1419,       pbase_machines=F1424,   pbase_newmachines =F1429,
      pbase_newthings=F1434,     pbase_story =F1439,     pbase_decisions=F1444,
      pbase_money =F1448,     pbase_finances=F1451,  pbase_math =F1454,
      pbase_reasoning=F1457,
      pbetter_familyfriends=F1390, pbetter_recent=F1395,    pbetter_convos=F1400,
      pbetter_addressphone =F1405,      pbetter_daymonth=F1410,
      pbetter_thingskept=F1415,  pbetter_findthings=F1420,  pbetter_machines=F1425,
      pbetter_newmachines =F1430,       pbetter_newthings=F1435,   pbetter_story
=F1440,      pbetter_decisions=F1445,   pbetter_money =F1449,
      pbetter_finances=F1452,   pbetter_math =F1455,       pbetter_reasoning =F1458,
      pworse_familyfriends=F1391,  pworse_recent=F1396,     pworse_convos=F1401,
pworse_addressphone =F1406,     pworse_daymonth=F1411,     pworse_thingskept=F1416,
pworse_findthings=F1421,     pworse_machines=F1426, pworse_newmachines =F1431,
      pworse_newthings=F1436,    pworse_story =F1441,
pworse_decisions=F1446,    pworse_money =F1450,          pworse_finances=F1453,
      pworse_math =F1456, pworse_reasoning=F1459,
        w=5, pw=4);

*If respondent doesnt have a proxy measure for the prior wave, recode them as 0 (non-
proxy);
if R4PROXY = "" then R4PROXY = 0;

*Step 3 Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=5, pw=4);

*Step 4 Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=5, pw=4);

*Step 5 Create change in TICS score variables for each TICS item;
%TICS_change(w=5, pw=4);

*Step 7 Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Self-respondents;
%hurd_self(age=R5AGEY_E, edu=RAEDEGRM, sex=RAGENDER, ADL=ADLcomposite5,
IADL=IADLcomposite5, cADL=ADLchange45, cIADL=IADLchange45,dates=dates5,
```

```
                bwc20=R5BWC20, s7=R5SER7, sci=R5SCIS, cac=R5CACT, vp=R5VP, imr=R5IMRC,
delr=R5DLRC, pdates=dates4,
                pbwc20=R4BWC20, ps7=R4SER7, psci=R4SCIS, pcac=R4CACT, pvp=R4VP,
pimr=R4IMRC, pdelr=R4DLRC, flagcADL = flag_ADL4, cdates = cdates45,
                cbwc20=cbwc2045, cs7=cs745, csci=csci45, ccac=ccac45, cvp=cvp45, cimr
=cimr45, cdelr =cdelr45, flag_bwc20=flag_bwc205, flag_cbwc20=flag_cbwc2045);

*Proxy-respondents;
%hurd_proxy(age=R5AGEY_E, edu=RAEDEGRM, sex=RAGENDER,
                    ADL=ADLcomposite5, IADL=IADLcomposite5, chADL=ADLchange45,
chIADL=IADLchange45, pdates=dates4,
                    pproxy =R4PROXY, ps7=R4SER7, pvp =R4VP, pimr=R4IMRC, pdelr=R4DLRC,
jorm=avgiq5, cjorm=ciqtot45, flag_pdates=flag_dates4);

*Step 8 Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal;
%predictedprobability(5);

*keep pertinent variables;
keep HHIDPN D_predictedprob5 C_predictedprob5 N_predictedprob5 Dementia_cat5 INW4 INW5
INW6 INW7 INW8 INW9 INW10 INW11 INW12;
run;


**WAVE 6**;
*Sort datasets by unique ID;
proc sort data = HRS.wave6;
by HHIDPN;
run;
proc sort data = HRS.wave5;
by HHIDPN;
run;

*Step 1. Merge datasets (prior wave and current wave);
data predictions6;
merge HRS.wave5 (in=a) HRS.wave6 (in=b);
by HHIDPN;
if a & b;

*Step 2 Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R6EAT,ADLbed=R6BED,ADLtoilt=R6TOILT,ADLdress=R6DRESS,ADLbath=R6BATH,ADLwa
lk=R6WALKR,IADLmeal=R6MEALS,
IADLshop=R6SHOPA,IADLphone=R6PHONEA,IADLmeds=R6MEDSA,IADLmoney=R6MONEYA,pADLeat=R5EAT,
pADLbed=R5BED,pADLtoilt=R5TOILT,pADLdress=R5DRESS,pADLbath=R5BATH,
pADLwalk=R5WALKR,pIADLmeal=R5MEALS,pIADLshop=R5SHOPA,pIADLphone=R5PHONEA,pIADLmeds=R5M
EDSA,pIADLmoney=R5MONEYA, w=6, pw=5);

*Dates;
%dates(day=R6DY, month=R6MO, year=R6YR, dayweek=R6DW, pday=R5DY, pmonth=R5MO,
pyear=R5YR, pdayweek=R5DW, w=6, pw=5);

%jorm(base_familyfriends=HD506,      base_recent=HD509,      base_convos=HD512,
base_addressphone =HD515,  base_daymonth=HD518,       base_thingskept=HD521,
base_findthings=HD524,         base_machines=HD527,   base_newmachines =HD530,
       base_newthings=HD533,      base_story =HD536,   base_decisions=HD539,
base_money =HD542,      base_finances=HD545,   base_math =HD548,
       base_reasoning=HD551,
      better_familyfriends=HD507, better_recent=HD510, better_convos=HD513,
       better_addressphone =HD516,       better_daymonth=HD519,
       better_thingskept=HD522, better_findthings=HD525,     better_machines=HD528,
       better_newmachines =HD531, better_newthings=HD534,    better_story =HD537,
```

```
better_decisions=HD540, better_money =HD543,   better_finances=HD546,     better_math
=HD549,      better_reasoning =HD552,
      worse_familyfriends=HD508, worse_recent=HD511,  worse_convos=HD514,
       worse_addressphone =HD517, worse_daymonth=HD520,      worse_thingskept=HD523,
worse_findthings=HD526,    worse_machines=HD529,      worse_newmachines =HD532,
       worse_newthings=HD535,     worse_story =HD538,  worse_decisions=HD541,
worse_money =HD544, worse_finances=HD547,      worse_math =HD550,
       worse_reasoning=HD553,
pbase_familyfriends=G1543,     pbase_recent=G1548,     pbase_convos=G1553,
pbase_addressphone =G1558, pbase_daymonth=G1563,      pbase_thingskept=G1568,
pbase_findthings=G1573,        pbase_machines=G1578,  pbase_newmachines =G1583,
       pbase_newthings=G1588,     pbase_story =G1593,  pbase_decisions=G1598,
pbase_money =G1602,     pbase_finances=G1605,  pbase_math =G1608,
       pbase_reasoning=G1611,
      pbetter_familyfriends=G1544,      pbetter_recent=G1549, pbetter_convos=G1554,
       pbetter_addressphone =G1559,      pbetter_daymonth=G1564,
       pbetter_thingskept=G1569, pbetter_findthings=G1574,  pbetter_machines=G1579,
       pbetter_newmachines =G1584,       pbetter_newthings=G1589,   pbetter_story
=G1594, pbetter_decisions=G1599, pbetter_money =G1603,      pbetter_finances=G1606,
       pbetter_math =G1609,       pbetter_reasoning =G1612,
      pworse_familyfriends=G1545, pworse_recent=G1550,        pworse_convos=G1555,
       pworse_addressphone =G1560,       pworse_daymonth=G1565,
       pworse_thingskept=G1570,    pworse_findthings=G1575,  pworse_machines=G1580,
       pworse_newmachines =G1585, pworse_newthings=G1590,    pworse_story =G1595,
pworse_decisions=G1600,     pworse_money =G1604,      pworse_finances=G1607,
       pworse_math =G1610, pworse_reasoning=G1613,
w=6, pw=5);

*If respondent doesnt have a proxy measure for the prior wave, recode them as 0 (non-
proxy);
if R5PROXY = "" then R5PROXY = 0;

*Step 3 Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=6, pw=5);

*Step 4 Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=6, pw=5);

*Step 5 Create change in TICS score variables for each TICS item;
%TICS_change(w=6, pw=5);

*Step 7 Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Self-respondents;
%hurd_self(age=R6AGEY_E, edu=RAEDEGRM, sex=RAGENDER, ADL=ADLcomposite6,
IADL=IADLcomposite6, cADL=ADLchange56, cIADL=IADLchange56,dates=dates6,
                bwc20=R6BWC20, s7=R6SER7, sci=R6SCIS, cac=R6CACT, vp=R6VP, imr=R6IMRC,
delr=R6DLRC, pdates=dates5,
                pbwc20=R5BWC20, ps7=R5SER7, psci=R5SCIS, pcac=R5CACT, pvp=R5VP,
pimr=R5IMRC, pdelr=R5DLRC, flagcADL = flag_ADL5, cdates = cdates56,
                cbwc20=cbwc2056, cs7=cs756, csci=csci56, ccac=ccac56, cvp=cvp56, cimr
=cimr56, cdelr =cdelr56, flag_bwc20=flag_bwc206, flag_cbwc20=flag_cbwc2056);

*Proxy-respondents;
%hurd_proxy(age=R6AGEY_E, edu=RAEDEGRM, sex=RAGENDER,
                    ADL=ADLcomposite6, IADL=IADLcomposite6, chADL=ADLchange56,
chIADL=IADLchange56, pdates=dates5,
                    pproxy =R5PROXY, ps7=R5SER7, pvp =R5VP, pimr=R5IMRC, pdelr=R5DLRC,
jorm=avgiq6, cjorm=ciqtot56, flag_pdates=flag_dates5);

*Step 8 Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal;
%predictedprobability(6);
```

```
*keep pertinent variables;
keep HHIDPN D_predictedprob6 C_predictedprob6 N_predictedprob6 Dementia_cat6 INW4 INW5
INW6 INW7 INW8 INW9 INW10 INW11 INW12;
run;


**WAVE 7**;
*Sort datasets by unique ID;
proc sort data = HRS.wave7;
by HHIDPN;
run;
proc sort data = HRS.wave6;
by HHIDPN;
run;

*Step 1. Merge datasets (prior wave and current wave);
data predictions7;
merge HRS.wave6 (in=a) HRS.wave7 (in=b);
by HHIDPN;
if a & b;

*Step 2 Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R7EAT,ADLbed=R7BED,ADLtoilt=R7TOILT,ADLdress=R7DRESS,ADLbath=R7BATH,ADLwa
lk=R7WALKR,IADLmeal=R7MEALS,
IADLshop=R7SHOPA,IADLphone=R7PHONEA,IADLmeds=R7MEDSA,IADLmoney=R7MONEYA,pADLeat=R6EAT,
pADLbed=R6BED,pADLtoilt=R6TOILT,pADLdress=R6DRESS,pADLbath=R6BATH,
pADLwalk=R6WALKR,pIADLmeal=R6MEALS,pIADLshop=R6SHOPA,pIADLphone=R6PHONEA,pIADLmeds=R6M
EDSA,pIADLmoney=R6MONEYA, w=7, pw=6);

*Dates;
%dates(day=R7DY, month=R7MO, year=R7YR, dayweek=R7DW, pday=R6DY, pmonth=R6MO,
pyear=R6YR, pdayweek=R6DW, w=7, pw=6);

%jorm(base_familyfriends=JD506,     base_recent=JD509,      base_convos=JD512,
base_addressphone =JD515,  base_daymonth=JD518,       base_thingskept=JD521,
base_findthings=JD524,          base_machines=JD527,   base_newmachines =JD530,
      base_newthings=JD533,     base_story =JD536,    base_decisions=JD539,
base_money =JD542,      base_finances=JD545,   base_math =JD548,
      base_reasoning=JD551,
      better_familyfriends=JD507, better_recent=JD510, better_convos=JD513,
       better_addressphone =JD516,       better_daymonth=JD519,
       better_thingskept=JD522, better_findthings=JD525,     better_machines=JD528,
       better_newmachines =JD531, better_newthings=JD534,    better_story =JD537,
better_decisions=JD540, better_money =JD543,   better_finances=JD546,     better_math
=JD549,       better_reasoning =JD552,
      worse_familyfriends=JD508,  worse_recent=JD511,  worse_convos=JD514,
       worse_addressphone =JD517, worse_daymonth=JD520,      worse_thingskept=JD523,
worse_findthings=JD526,    worse_machines=JD529,      worse_newmachines =JD532,
       worse_newthings=JD535,     worse_story =JD538,  worse_decisions=JD541,
worse_money =JD544, worse_finances=JD547,      worse_math =JD550,
       worse_reasoning=JD553,
pbase_familyfriends=HD506,     pbase_recent=HD509,     pbase_convos=HD512,
pbase_addressphone =HD515, pbase_daymonth=HD518,      pbase_thingskept=HD521,
pbase_findthings=HD524,         pbase_machines=HD527,  pbase_newmachines =HD530,
       pbase_newthings=HD533,     pbase_story =HD536,  pbase_decisions=HD539,
pbase_money =HD542,     pbase_finances=HD545,  pbase_math =HD548,
       pbase_reasoning=HD551,
      pbetter_familyfriends=HD507,      pbetter_recent=HD510, pbetter_convos=HD513,
       pbetter_addressphone =HD516,      pbetter_daymonth=HD519,
       pbetter_thingskept=HD522, pbetter_findthings=HD525,   pbetter_machines=HD528,
```

```
      pbetter_newmachines =HD531,      pbetter_newthings=HD534,   pbetter_story
=HD537, pbetter_decisions=HD540, pbetter_money =HD543,      pbetter_finances=HD546,
      pbetter_math =HD549,       pbetter_reasoning =HD552,
     pworse_familyfriends=HD508, pworse_recent=HD511,        pworse_convos=HD514,
      pworse_addressphone =HD517,      pworse_daymonth=HD520,
      pworse_thingskept=HD523,    pworse_findthings=HD526,  pworse_machines=HD529,
      pworse_newmachines =HD532, pworse_newthings=HD535,    pworse_story =HD538,
pworse_decisions=HD541,     pworse_money =HD544,      pworse_finances=HD547,
      pworse_math =HD550, pworse_reasoning=HD553,
w=7, pw=6);

*If respondent doesnt have a proxy measure for the prior wave, recode them as 0 (non-
proxy);
if R6PROXY = "" then R6PROXY = 0;

*Step 3 Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=7, pw=6);

*Step 4 Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=7, pw=6);

*Step 5 Create change in TICS score variables for each TICS item;
%TICS_change(w=7, pw=6);

*Step 7 Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Self-respondents;
%hurd_self(age=R7AGEY_E, edu=RAEDEGRM, sex=RAGENDER, ADL=ADLcomposite7,
IADL=IADLcomposite7, cADL=ADLchange67, cIADL=IADLchange67,dates=dates7,
                bwc20=R7BWC20, s7=R7SER7, sci=R7SCIS, cac=R7CACT, vp=R7VP, imr=R7IMRC,
delr=R7DLRC, pdates=dates6,
                pbwc20=R6BWC20, ps7=R6SER7, psci=R6SCIS, pcac=R6CACT, pvp=R6VP,
pimr=R6IMRC, pdelr=R6DLRC, flagcADL = flag_ADL6, cdates = cdates67,
                cbwc20=cbwc2067, cs7=cs767, csci=csci67, ccac=ccac67, cvp=cvp67, cimr
=cimr67, cdelr =cdelr67, flag_bwc20=flag_bwc207, flag_cbwc20=flag_cbwc2067);

*Proxy-respondents;
%hurd_proxy(age=R7AGEY_E, edu=RAEDEGRM, sex=RAGENDER,
                    ADL=ADLcomposite7, IADL=IADLcomposite7, chADL=ADLchange67,
chIADL=IADLchange67, pdates=dates6,
                    pproxy =R6PROXY, ps7=R6SER7, pvp =R6VP, pimr=R6IMRC, pdelr=R6DLRC,
jorm=avgiq7, cjorm=ciqtot67, flag_pdates=flag_dates6);

*Step 8 Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal;
%predictedprobability(7);

*keep pertinent variables;
keep HHIDPN D_predictedprob7 C_predictedprob7 N_predictedprob7 Dementia_cat7 INW4 INW5
INW6 INW7 INW8 INW9 INW10 INW11 INW12;
run;


**WAVE 8**;
*Sort datasets by unique ID;
proc sort data = HRS.wave8;
by HHIDPN;
run;
proc sort data = HRS.wave7;
by HHIDPN;
run;

*Step 1. Merge datasets (prior wave and current wave);
```

```
data predictions8;
merge HRS.wave7 (in=a) HRS.wave8 (in=b);
by HHIDPN;
if a & b;

*Step 2 Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R8EAT,ADLbed=R8BED,ADLtoilt=R8TOILT,ADLdress=R8DRESS,ADLbath=R8BATH,ADLwa
lk=R8WALKR,IADLmeal=R8MEALS,
IADLshop=R8SHOPA,IADLphone=R8PHONEA,IADLmeds=R8MEDSA,IADLmoney=R8MONEYA,pADLeat=R7EAT,
pADLbed=R7BED,pADLtoilt=R7TOILT,pADLdress=R7DRESS,pADLbath=R7BATH,
pADLwalk=R7WALKR,pIADLmeal=R7MEALS,pIADLshop=R7SHOPA,pIADLphone=R7PHONEA,pIADLmeds=R7M
EDSA,pIADLmoney=R7MONEYA, w=8, pw=7);

*Dates;
%dates(day=R8DY, month=R8MO, year=R8YR, dayweek=R8DW, pday=R7DY, pmonth=R7MO,
pyear=R7YR, pdayweek=R7DW, w=8, pw=7);

%jorm(base_familyfriends=KD506,      base_recent=KD509,      base_convos=KD512,
base_addressphone =KD515,  base_daymonth=KD518,       base_thingskept=KD521,
base_findthings=KD524,         base_machines=KD527,   base_newmachines =KD530,
      base_newthings=KD533,      base_story =KD536,   base_decisions=KD539,
base_money =KD542,      base_finances=KD545,   base_math =KD548,
      base_reasoning=KD551,
      better_familyfriends=KD507, better_recent=KD510, better_convos=KD513,
      better_addressphone =KD516,      better_daymonth=KD519,
      better_thingskept=KD522, better_findthings=KD525,    better_machines=KD528,
      better_newmachines =KD531, better_newthings=KD534,    better_story =KD537,
better_decisions=KD540, better_money =KD543,   better_finances=KD546,     better_math
=KD549,      better_reasoning =KD552,
      worse_familyfriends=KD508, worse_recent=KD511,  worse_convos=KD514,
      worse_addressphone =KD517, worse_daymonth=KD520,      worse_thingskept=KD523,
worse_findthings=KD526,    worse_machines=KD529,      worse_newmachines =KD532,
      worse_newthings=KD535,     worse_story =KD538,  worse_decisions=KD541,
worse_money =KD544, worse_finances=KD547,      worse_math =KD550,
      worse_reasoning=KD553,
pbase_familyfriends=JD506,      pbase_recent=JD509,     pbase_convos=JD512,
pbase_addressphone =JD515, pbase_daymonth=JD518,      pbase_thingskept=JD521,
pbase_findthings=JD524,         pbase_machines=JD527,  pbase_newmachines =JD530,
      pbase_newthings=JD533,     pbase_story =JD536,  pbase_decisions=JD539,
pbase_money =JD542,     pbase_finances=JD545,  pbase_math =JD548,
      pbase_reasoning=JD551,
      pbetter_familyfriends=JD507,      pbetter_recent=JD510, pbetter_convos=JD513,
      pbetter_addressphone =JD516,      pbetter_daymonth=JD519,
      pbetter_thingskept=JD522, pbetter_findthings=JD525,   pbetter_machines=JD528,
      pbetter_newmachines =JD531,       pbetter_newthings=JD534,   pbetter_story
=JD537, pbetter_decisions=JD540, pbetter_money =JD543,      pbetter_finances=JD546,
      pbetter_math =JD549,       pbetter_reasoning =JD552,
      pworse_familyfriends=JD508, pworse_recent=JD511,        pworse_convos=JD514,
      pworse_addressphone =JD517,       pworse_daymonth=JD520,
      pworse_thingskept=JD523,    pworse_findthings=JD526,  pworse_machines=JD529,
      pworse_newmachines =JD532, pworse_newthings=JD535,    pworse_story =JD538,
pworse_decisions=JD541,     pworse_money =JD544,      pworse_finances=JD547,
      pworse_math =JD550, pworse_reasoning=JD553,
w=8, pw=7);

*If respondent doesnt have a proxy measure for the prior wave, recode them as 0 (non-
proxy);
if R7PROXY = "" then R7PROXY = 0;

*Step 3 Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=8, pw=7);
```

```
*Step 4 Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=8, pw=7);

*Step 5 Create change in TICS score variables for each TICS item;
%TICS_change(w=8, pw=7);

*Step 7 Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Self-respondents;
%hurd_self(age=R8AGEY_E, edu=RAEDEGRM, sex=RAGENDER, ADL=ADLcomposite8,
IADL=IADLcomposite8, cADL=ADLchange78, cIADL=IADLchange78,dates=dates8,
                bwc20=R8BWC20, s7=R8SER7, sci=R8SCIS, cac=R8CACT, vp=R8VP, imr=R8IMRC,
delr=R8DLRC, pdates=dates7,
                pbwc20=R7BWC20, ps7=R7SER7, psci=R7SCIS, pcac=R7CACT, pvp=R7VP,
pimr=R7IMRC, pdelr=R7DLRC, flagcADL = flag_ADL7, cdates = cdates78,
                cbwc20=cbwc2078, cs7=cs778, csci=csci78, ccac=ccac78, cvp=cvp78, cimr
=cimr78, cdelr =cdelr78, flag_bwc20=flag_bwc208, flag_cbwc20=flag_cbwc2078);

*Proxy-respondents;
%hurd_proxy(age=R8AGEY_E, edu=RAEDEGRM, sex=RAGENDER,
                    ADL=ADLcomposite8, IADL=IADLcomposite8, chADL=ADLchange78,
chIADL=IADLchange78, pdates=dates7,
                    pproxy =R7PROXY, ps7=R7SER7, pvp =R7VP, pimr=R7IMRC, pdelr=R7DLRC,
jorm=avgiq8, cjorm=ciqtot78, flag_pdates=flag_dates7);

*Step 8 Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal;
%predictedprobability(8);

*keep pertinent variables;
keep HHIDPN D_predictedprob8 C_predictedprob8 N_predictedprob8 Dementia_cat8 INW4 INW5
INW6 INW7 INW8 INW9 INW10 INW11 INW12;
run;

**WAVE 9**;
*Sort datasets by unique ID;
proc sort data = HRS.wave9;
by HHIDPN;
run;
proc sort data = HRS.wave8;
by HHIDPN;
run;

*Step 1. Merge datasets (prior wave and current wave);
data predictions9;
merge HRS.wave8 (in=a) HRS.wave9 (in=b);
by HHIDPN;
if a & b;

*Step 2 Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R9EAT,ADLbed=R9BED,ADLtoilt=R9TOILT,ADLdress=R9DRESS,ADLbath=R9BATH,ADLwa
lk=R9WALKR,IADLmeal=R9MEALS,
IADLshop=R9SHOPA,IADLphone=R9PHONEA,IADLmeds=R9MEDSA,IADLmoney=R9MONEYA,pADLeat=R8EAT,
pADLbed=R8BED,pADLtoilt=R8TOILT,pADLdress=R8DRESS,pADLbath=R8BATH,
pADLwalk=R8WALKR,pIADLmeal=R8MEALS,pIADLshop=R8SHOPA,pIADLphone=R8PHONEA,pIADLmeds=R8M
EDSA,pIADLmoney=R8MONEYA, w=9, pw=8);

*Dates;
%dates(day=R9DY, month=R9MO, year=R9YR, dayweek=R9DW, pday=R8DY, pmonth=R8MO,
pyear=R8YR, pdayweek=R8DW, w=9, pw=8);
```

```
%jorm(base_familyfriends=LD506,      base_recent=LD509,      base_convos=LD512,
base_addressphone =LD515,  base_daymonth=LD518,       base_thingskept=LD521,
base_findthings=LD524,         base_machines=LD527,   base_newmachines =LD530,
      base_newthings=LD533,      base_story =LD536,   base_decisions=LD539,
base_money =LD542,      base_finances=LD545,   base_math =LD548,
      base_reasoning=LD551,
      better_familyfriends=LD507, better_recent=LD510, better_convos=LD513,
       better_addressphone =LD516,       better_daymonth=LD519,
       better_thingskept=LD522, better_findthings=LD525,     better_machines=LD528,
       better_newmachines =LD531, better_newthings=LD534,    better_story =LD537,
better_decisions=LD540, better_money =LD543,   better_finances=LD546,     better_math
=LD549,      better_reasoning =LD552,
      worse_familyfriends=LD508,  worse_recent=LD511,  worse_convos=LD514,
       worse_addressphone =LD517, worse_daymonth=LD520,      worse_thingskept=LD523,
worse_findthings=LD526,    worse_machines=LD529,      worse_newmachines =LD532,
       worse_newthings=LD535,     worse_story =LD538,  worse_decisions=LD541,
worse_money =LD544, worse_finances=LD547,      worse_math =LD550,
       worse_reasoning=LD553,
pbase_familyfriends=KD506,     pbase_recent=KD509,     pbase_convos=KD512,
pbase_addressphone =KD515, pbase_daymonth=KD518,      pbase_thingskept=KD521,
pbase_findthings=KD524,        pbase_machines=KD527,  pbase_newmachines =KD530,
       pbase_newthings=KD533,     pbase_story =KD536,  pbase_decisions=KD539,
pbase_money =KD542,     pbase_finances=KD545,  pbase_math =KD548,
       pbase_reasoning=KD551,
      pbetter_familyfriends=KD507,      pbetter_recent=KD510, pbetter_convos=KD513,
       pbetter_addressphone =KD516,     pbetter_daymonth=KD519,
       pbetter_thingskept=KD522, pbetter_findthings=KD525,   pbetter_machines=KD528,
       pbetter_newmachines =KD531,      pbetter_newthings=KD534,   pbetter_story
=KD537, pbetter_decisions=KD540, pbetter_money =KD543,      pbetter_finances=KD546,
       pbetter_math =KD549,       pbetter_reasoning =KD552,
      pworse_familyfriends=KD508, pworse_recent=KD511,        pworse_convos=KD514,
       pworse_addressphone =KD517,      pworse_daymonth=KD520,
       pworse_thingskept=KD523,    pworse_findthings=KD526,  pworse_machines=KD529,
       pworse_newmachines =KD532, pworse_newthings=KD535,    pworse_story =KD538,
pworse_decisions=KD541,     pworse_money =KD544,      pworse_finances=KD547,
       pworse_math =KD550, pworse_reasoning=KD553,
         w=9, pw=8);

*If respondent doesnt have a proxy measure for the prior wave, recode them as 0 (non-
proxy);
if R8PROXY = "" then R8PROXY = 0;

*Step 3 Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=9, pw=8);

*Step 4 Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=9, pw=8);

*Step 5 Create change in TICS score variables for each TICS item;
%TICS_change(w=9, pw=8);

*Step 7 Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Self-respondents;
%hurd_self(age=R9AGEY_E, edu=RAEDEGRM, sex=RAGENDER, ADL=ADLcomposite9,
IADL=IADLcomposite9, cADL=ADLchange89, cIADL=IADLchange89,dates=dates9,
                bwc20=R9BWC20, s7=R9SER7, sci=R9SCIS, cac=R9CACT, vp=R9VP, imr=R9IMRC,
delr=R9DLRC, pdates=dates8,
                pbwc20=R8BWC20, ps7=R8SER7, psci=R8SCIS, pcac=R8CACT, pvp=R8VP,
pimr=R8IMRC, pdelr=R8DLRC, flagcADL = flag_ADL8, cdates = cdates89,
                cbwc20=cbwc2089, cs7=cs789, csci=csci89, ccac=ccac89, cvp=cvp89, cimr
=cimr89, cdelr =cdelr89, flag_bwc20=flag_bwc209, flag_cbwc20=flag_cbwc2089);
```

```
*Proxy-respondents;
%hurd_proxy(age=R9AGEY_E, edu=RAEDEGRM, sex=RAGENDER,
                    ADL=ADLcomposite9, IADL=IADLcomposite9, chADL=ADLchange89,
chIADL=IADLchange89, pdates=dates8,
                    pproxy =R8PROXY, ps7=R8SER7, pvp =R8VP, pimr=R8IMRC, pdelr=R8DLRC,
jorm=avgiq9, cjorm=ciqtot89, flag_pdates=flag_dates8);

*Step 8 Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal;
%predictedprobability(9);

*keep pertinent variables;
keep HHIDPN D_predictedprob9 C_predictedprob9 N_predictedprob9 Dementia_cat9 INW4 INW5
INW6 INW7 INW8 INW9 INW10 INW11 INW12;
run;


**WAVE 10**;
*Sort datasets by unique ID;
proc sort data = HRS.wave10;
by HHIDPN;
run;
proc sort data = HRS.wave9;
by HHIDPN;
run;

*Step 1. Merge datasets (prior wave and current wave);
data predictions10;
merge HRS.wave9 (in=a) HRS.wave10 (in=b);
by HHIDPN;
if a & b;

*Step 2 Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R10EAT,ADLbed=R10BED,ADLtoilt=R10TOILT,ADLdress=R10DRESS,ADLbath=R10BATH,
ADLwalk=R10WALKR,IADLmeal=R10MEALS,
IADLshop=R10SHOPA,IADLphone=R10PHONEA,IADLmeds=R10MEDSA,IADLmoney=R10MONEYA,pADLeat=R9
EAT,pADLbed=R9BED,pADLtoilt=R9TOILT,pADLdress=R9DRESS,pADLbath=R9BATH,
pADLwalk=R9WALKR,pIADLmeal=R9MEALS,pIADLshop=R9SHOPA,pIADLphone=R9PHONEA,pIADLmeds=R9M
EDSA,pIADLmoney=R9MONEYA, w=10, pw=9);

*Dates;
%dates(day=R10DY, month=R10MO, year=R10YR, dayweek=R10DW, pday=R9DY, pmonth=R9MO,
pyear=R9YR, pdayweek=R9DW, w=10, pw=9);

*Jorm IQCODE;
%jorm(base_familyfriends=MD506,      base_recent=MD509,      base_convos=MD512,
base_addressphone =MD515,  base_daymonth=MD518,       base_thingskept=MD521,
base_findthings=MD524,         base_machines=MD527,   base_newmachines =MD530,
       base_newthings=MD533,      base_story =MD536,   base_decisions=MD539,
base_money =MD542,      base_finances=MD545,   base_math =MD548,
       base_reasoning=MD551,
      better_familyfriends=MD507,better_recent=MD510, better_convos=MD513,
       better_addressphone =MD516,       better_daymonth=MD519,
       better_thingskept=MD522, better_findthings=MD525,     better_machines=MD528,
       better_newmachines =MD531, better_newthings=MD534,    better_story =MD537,
better_decisions=MD540, better_money =MD543,   better_finances=MD546,     better_math
=MD549,       better_reasoning =MD552,
      worse_familyfriends=MD508,  worse_recent=MD511,  worse_convos=MD514,
       worse_addressphone =MD517, worse_daymonth=MD520,      worse_thingskept=MD523,
worse_findthings=MD526,    worse_machines=MD529,      worse_newmachines =MD532,
```

```
      worse_newthings=MD535,     worse_story =MD538,  worse_decisions=MD541,
worse_money =MD544, worse_finances=MD547,      worse_math =MD550,
      worse_reasoning=MD553,
pbase_familyfriends=LD506,     pbase_recent=LD509,     pbase_convos=LD512,
pbase_addressphone =LD515, pbase_daymonth=LD518,      pbase_thingskept=LD521,
pbase_findthings=LD524,        pbase_machines=LD527,  pbase_newmachines =LD530,
      pbase_newthings=LD533,     pbase_story =LD536,  pbase_decisions=LD539,
pbase_money =LD542,     pbase_finances=LD545,  pbase_math =LD548,
      pbase_reasoning=LD551,
      pbetter_familyfriends=LD507,      pbetter_recent=LD510, pbetter_convos=LD513,
      pbetter_addressphone =LD516,      pbetter_daymonth=LD519,
      pbetter_thingskept=LD522, pbetter_findthings=LD525,   pbetter_machines=LD528,
      pbetter_newmachines =LD531,       pbetter_newthings=LD534,   pbetter_story
=LD537, pbetter_decisions=LD540, pbetter_money =LD543,      pbetter_finances=LD546,
      pbetter_math =LD549,       pbetter_reasoning =LD552,
      pworse_familyfriends=LD508, pworse_recent=LD511,        pworse_convos=LD514,
      pworse_addressphone =LD517,       pworse_daymonth=LD520,
      pworse_thingskept=LD523,    pworse_findthings=LD526,  pworse_machines=LD529,
      pworse_newmachines =LD532, pworse_newthings=LD535,    pworse_story =LD538,
pworse_decisions=LD541,     pworse_money =LD544,      pworse_finances=LD547,
      pworse_math =LD550, pworse_reasoning=LD553,
w=10, pw=9);

*If respondent doesnt have a proxy measure for the prior wave, recode them as 0 (non-
proxy);
if R9PROXY = "" then R9PROXY = 0;

*Step 3 Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=10, pw=9);

*Step 4 Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=10, pw=9);

*Step 5 Create change in TICS score variables for each TICS item;
%TICS_change(w=10, pw=9);

*Step 7 Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Self-respondents;
%hurd_self(age=R10AGEY_E, edu=RAEDEGRM, sex=RAGENDER, ADL=ADLcomposite10,
IADL=IADLcomposite10, cADL=ADLchange910, cIADL=IADLchange910,dates=dates10,
                bwc20=R10BWC20, s7=R10SER7, sci=R10SCIS, cac=R10CACT, vp=R10VP,
imr=R10IMRC, delr=R10DLRC, pdates=dates9,
                pbwc20=R9BWC20, ps7=R9SER7, psci=R9SCIS, pcac=R9CACT, pvp=R9VP,
pimr=R9IMRC, pdelr=R9DLRC, flagcADL = flag_ADL9, cdates = cdates910,
                cbwc20=cbwc20910, cs7=cs7910, csci=csci910, ccac=ccac910, cvp=cvp910,
cimr =cimr910, cdelr =cdelr910, flag_bwc20=flag_bwc2010, flag_cbwc20=flag_cbwc20910);

*Proxy-respondents;
%hurd_proxy(age=R10AGEY_E, edu=RAEDEGRM, sex=RAGENDER,
                    ADL=ADLcomposite10, IADL=IADLcomposite10, chADL=ADLchange910,
chIADL=IADLchange910, pdates=dates9,
                    pproxy =R9PROXY, ps7=R9SER7, pvp =R9VP, pimr=R9IMRC, pdelr=R9DLRC,
jorm=avgiq10, cjorm=ciqtot910, flag_pdates=flag_dates9);

*Step 8 Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal;
%predictedprobability(10);

*keep pertinent variables;
keep HHIDPN D_predictedprob10 C_predictedprob10 N_predictedprob10 Dementia_cat10 INW4
INW5 INW6 INW7 INW8 INW9 INW10 INW11 INW12;
run;
```

```
**WAVE 11**;
*Sort datasets by unique ID;
proc sort data = HRS.wave11;
by HHIDPN;
run;
proc sort data = HRS.wave10;
by HHIDPN;
run;

*Step 1. Merge datasets (prior wave and current wave);
data predictions11;
merge HRS.wave10 (in=a) HRS.wave11 (in=b);
by HHIDPN;
if a & b;

*Step 2 Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R11EAT,ADLbed=R11BED,ADLtoilt=R11TOILT,ADLdress=R11DRESS,ADLbath=R11BATH,
ADLwalk=R11WALKR,IADLmeal=R11MEALS,
IADLshop=R11SHOPA,IADLphone=R11PHONEA,IADLmeds=R11MEDSA,IADLmoney=R11MONEYA,pADLeat=R1
0EAT,pADLbed=R10BED,pADLtoilt=R10TOILT,pADLdress=R10DRESS,pADLbath=R10BATH,
pADLwalk=R10WALKR,pIADLmeal=R10MEALS,pIADLshop=R10SHOPA,pIADLphone=R10PHONEA,pIADLmeds
=R10MEDSA,pIADLmoney=R10MONEYA, w=11, pw=10);

*Dates;
%dates(day=R11DY, month=R11MO, year=R11YR, dayweek=R11DW, pday=R10DY, pmonth=R10MO,
pyear=R10YR, pdayweek=R10DW, w=11, pw=10);

*Jorm IQCODE;
%jorm(base_familyfriends=ND506,      base_recent=ND509,      base_convos=ND512,
base_addressphone =ND515,  base_daymonth=ND518,       base_thingskept=ND521,
base_findthings=ND524,        base_machines=ND527,   base_newmachines =ND530,
       base_newthings=ND533,      base_story =ND536,   base_decisions=ND539,
base_money =ND542,      base_finances=ND545,   base_math =ND548,
       base_reasoning=ND551,
      better_familyfriends=ND507, better_recent=ND510, better_convos=ND513,
       better_addressphone =ND516,      better_daymonth=ND519,
       better_thingskept=ND522, better_findthings=ND525,     better_machines=ND528,
       better_newmachines =ND531, better_newthings=ND534,    better_story =ND537,
better_decisions=ND540, better_money =ND543,   better_finances=ND546,     better_math
=ND549,      better_reasoning =ND552,
      worse_familyfriends=ND508,  worse_recent=ND511,  worse_convos=ND514,
       worse_addressphone =ND517, worse_daymonth=ND520,      worse_thingskept=ND523,
worse_findthings=ND526,    worse_machines=ND529,      worse_newmachines =ND532,
       worse_newthings=ND535,     worse_story =ND538,  worse_decisions=ND541,
worse_money =ND544, worse_finances=ND547,      worse_math =ND550,
       worse_reasoning=ND553,
pbase_familyfriends=MD506,     pbase_recent=MD509,     pbase_convos=MD512,
pbase_addressphone =MD515, pbase_daymonth=MD518,      pbase_thingskept=MD521,
pbase_findthings=MD524,        pbase_machines=MD527,  pbase_newmachines =MD530,
       pbase_newthings=MD533,     pbase_story =MD536,  pbase_decisions=MD539,
pbase_money =MD542,     pbase_finances=MD545,  pbase_math =MD548,
       pbase_reasoning=MD551,
      pbetter_familyfriends=MD507,      pbetter_recent=MD510, pbetter_convos=MD513,
       pbetter_addressphone =MD516,     pbetter_daymonth=MD519,
       pbetter_thingskept=MD522, pbetter_findthings=MD525,   pbetter_machines=MD528,
       pbetter_newmachines =MD531,      pbetter_newthings=MD534,   pbetter_story
=MD537, pbetter_decisions=MD540, pbetter_money =MD543,      pbetter_finances=MD546,
       pbetter_math =MD549,       pbetter_reasoning =MD552,
      pworse_familyfriends=MD508, pworse_recent=MD511,        pworse_convos=MD514,
       pworse_addressphone =MD517,      pworse_daymonth=MD520,
```

```
      pworse_thingskept=MD523,    pworse_findthings=MD526,  pworse_machines=MD529,
      pworse_newmachines =MD532, pworse_newthings=MD535,    pworse_story =MD538,
pworse_decisions=MD541,     pworse_money =MD544,      pworse_finances=MD547,
      pworse_math =MD550, pworse_reasoning=MD553,
w=11, pw=10);

*If respondent doesnt have a proxy measure for the prior wave, recode them as 0 (non-
proxy);
if R10PROXY = "" then R10PROXY = 0;

*Step 3 Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=11, pw=10);

*Step 4 Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=11, pw=10);

*Step 5 Create change in TICS score variables for each TICS item;
%TICS_change(w=11, pw=10);

*Step 7 Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Self-respondents;
%hurd_self(age=R11AGEY_E, edu=RAEDEGRM, sex=RAGENDER, ADL=ADLcomposite11,
IADL=IADLcomposite11, cADL=ADLchange1011, cIADL=IADLchange1011,dates=dates11,
                bwc20=R11BWC20, s7=R11SER7, sci=R11SCIS, cac=R11CACT, vp=R11VP,
imr=R11IMRC, delr=R11DLRC, pdates=dates10,
                pbwc20=R10BWC20, ps7=R10SER7, psci=R10SCIS, pcac=R10CACT, pvp=R10VP,
pimr=R10IMRC, pdelr=R10DLRC, flagcADL = flag_ADL10, cdates = cdates1011,
                cbwc20=cbwc201011, cs7=cs71011, csci=csci1011, ccac=ccac1011,
cvp=cvp1011, cimr =cimr1011, cdelr =cdelr1011, flag_bwc20=flag_bwc2011,
flag_cbwc20=flag_cbwc201011);

*Proxy-respondents;
%hurd_proxy(age=R11AGEY_E, edu=RAEDEGRM, sex=RAGENDER,
                    ADL=ADLcomposite11, IADL=IADLcomposite11, chADL=ADLchange1011,
chIADL=IADLchange1011, pdates=dates10,
                    pproxy =R10PROXY, ps7=R10SER7, pvp =R10VP, pimr=R10IMRC,
pdelr=R10DLRC, jorm=avgiq11, cjorm=ciqtot1011, flag_pdates=flag_dates10);

*Step 8 Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal;
%predictedprobability(11);

*keep pertinent variables;
keep HHIDPN D_predictedprob11 C_predictedprob11 N_predictedprob11 Dementia_cat11 INW4
INW5 INW6 INW7 INW8 INW9 INW10 INW11 INW12;
run;


**WAVE 12**;
*Sort datasets by unique ID;
proc sort data = HRS.wave12;
by HHIDPN;
run;
proc sort data = HRS.wave11;
by HHIDPN;
run;

*Step 1. Merge datasets (prior wave and current wave);
data predictions12;
merge HRS.wave11 (in=a) HRS.wave12 (in=b);
by HHIDPN;
if a & b;
```

```
*Step 2 Create composite functional limitations (ADLs and IADLs), dates, and Jorm
IQCODE variables;
*Functional limitations;
%adls(ADLeat=R12EAT,ADLbed=R12BED,ADLtoilt=R12TOILT,ADLdress=R12DRESS,ADLbath=R12BATH,
ADLwalk=R12WALKR,IADLmeal=R12MEALS,
IADLshop=R12SHOPA,IADLphone=R12PHONEA,IADLmeds=R12MEDSA,IADLmoney=R12MONEYA,pADLeat=R1
1EAT,pADLbed=R11BED,pADLtoilt=R11TOILT,pADLdress=R11DRESS,pADLbath=R11BATH,
pADLwalk=R11WALKR,pIADLmeal=R11MEALS,pIADLshop=R11SHOPA,pIADLphone=R11PHONEA,pIADLmeds
=R11MEDSA,pIADLmoney=R11MONEYA, w=12, pw=11);

*Dates;
%dates(day=R12DY, month=R12MO, year=R12YR, dayweek=R12DW, pday=R11DY, pmonth=R11MO,
pyear=R11YR, pdayweek=R11DW, w=12, pw=11);

*Jorm IQCODE;
%jorm(base_familyfriends=OD506,      base_recent=OD509,     base_convos=OD512,
base_addressphone =OD515,  base_daymonth=OD518,       base_thingskept=OD521,
base_findthings=OD524,         base_machines=OD527,   base_newmachines =OD530,
      base_newthings=OD533,      base_story =OD536,   base_decisions=OD539,
base_money =OD542,      base_finances=OD545,   base_math =OD548,
      base_reasoning=OD551,
      better_familyfriends=OD507, better_recent=OD510, better_convos=OD513,
       better_addressphone =OD516,      better_daymonth=OD519,
       better_thingskept=OD522, better_findthings=OD525,    better_machines=OD528,
       better_newmachines =OD531, better_newthings=OD534,    better_story =OD537,
better_decisions=OD540, better_money =OD543,   better_finances=OD546,     better_math
=OD549,      better_reasoning =OD552,
      worse_familyfriends=OD508,  worse_recent=OD511,  worse_convos=OD514,
       worse_addressphone =OD517, worse_daymonth=OD520,      worse_thingskept=OD523,
worse_findthings=OD526,    worse_machines=OD529,      worse_newmachines =OD532,
       worse_newthings=OD535,     worse_story =OD538,  worse_decisions=OD541,
worse_money =OD544, worse_finances=OD547,      worse_math =OD550,
       worse_reasoning=OD553,
pbase_familyfriends=ND506,    pbase_recent=ND509,     pbase_convos=ND512,
pbase_addressphone =ND515, pbase_daymonth=ND518,      pbase_thingskept=ND521,
pbase_findthings=ND524,       pbase_machines=ND527,  pbase_newmachines =ND530,
      pbase_newthings=ND533,     pbase_story =ND536,  pbase_decisions=ND539,
pbase_money =ND542,     pbase_finances=ND545,  pbase_math =ND548,
      pbase_reasoning=ND551,
      pbetter_familyfriends=ND507,      pbetter_recent=ND510, pbetter_convos=ND513,
       pbetter_addressphone =ND516,     pbetter_daymonth=ND519,
       pbetter_thingskept=ND522, pbetter_findthings=ND525,   pbetter_machines=ND528,
       pbetter_newmachines =ND531,      pbetter_newthings=ND534,   pbetter_story
=ND537, pbetter_decisions=ND540, pbetter_money =ND543,      pbetter_finances=ND546,
      pbetter_math =ND549,       pbetter_reasoning =ND552,
      pworse_familyfriends=ND508, pworse_recent=ND511,        pworse_convos=ND514,
       pworse_addressphone =ND517,      pworse_daymonth=ND520,
       pworse_thingskept=ND523,    pworse_findthings=ND526,  pworse_machines=ND529,
       pworse_newmachines =ND532, pworse_newthings=ND535,    pworse_story =ND538,
pworse_decisions=ND541,     pworse_money =ND544,      pworse_finances=ND547,
      pworse_math =ND550, pworse_reasoning=ND553,
      w=12, pw=11);

*If respondent doesnt have a proxy measure for the prior wave, recode them as 0 (non-
proxy);
if R11PROXY = "" then R11PROXY = 0;

*Step 3: Recode non-binary TICS answers into 1 = correct, 0 = incorrect;
%TICS_binary(w=12, pw=11);

*Step 4: Create missing indicators for TICS items and replace missing values with 0;
%TICS_missings(w=12, pw=11);
```

```
*Step 5: Create change in TICS score variables for each TICS item;
%TICS_change(w=12, pw=11);

*Step 7 Define coefficients for Hurd equation and run equation for each respondent
(separate for self-respondents and proxy-respondents);
*Self-respondents;
%hurd_self(age=R12AGEY_E, edu=RAEDEGRM, sex=RAGENDER, ADL=ADLcomposite12,
IADL=IADLcomposite12, cADL=ADLchange1112, cIADL=IADLchange1112,dates=dates12,
                bwc20=R12BWC20, s7=R12SER7, sci=R12SCIS, cac=R12CACT, vp=R12VP,
imr=R12IMRC, delr=R12DLRC, pdates=dates11,
                pbwc20=R11BWC20, ps7=R11SER7, psci=R11SCIS, pcac=R11CACT, pvp=R11VP,
pimr=R11IMRC, pdelr=R11DLRC, flagcADL = flag_ADL11, cdates = cdates1112,
                cbwc20=cbwc201112, cs7=cs71112, csci=csci1112, ccac=ccac1112,
cvp=cvp1112, cimr =cimr1112, cdelr =cdelr1112, flag_bwc20=flag_bwc2012,
flag_cbwc20=flag_cbwc201112);

*Proxy-respondents;
%hurd_proxy(age=R12AGEY_E, edu=RAEDEGRM, sex=RAGENDER,
                    ADL=ADLcomposite12, IADL=IADLcomposite12, chADL=ADLchange1112,
chIADL=IADLchange1112, pdates=dates11,
                    pproxy =R11PROXY, ps7=R11SER7, pvp =R11VP, pimr=R11IMRC,
pdelr=R11DLRC, jorm=avgiq12, cjorm=ciqtot1112, flag_pdates=flag_dates11);

*Step 8 Calculate predicted probabilities and categorize respondents into Dementia,
CIND, Normal;
%predictedprobability(12);

*keep pertinent variables;
keep HHIDPN D_predictedprob12 C_predictedprob12 N_predictedprob12 Dementia_cat12 INW4
INW5 INW6 INW7 INW8 INW9 INW10 INW11 INW12;
run;


/**********************************     COMBINED DATASET
*****************************/
data HRS_dementia_predictions;
*Merge all waves;
merge predictions5 (in=a) predictions6 (in=b) predictions7 (in=c) predictions8 (in=d)
predictions9 (in=e) predictions10 (in=f) predictions11 (in=g) predictions12 (in=h);
by HHIDPN;

*Step 9. Drop those who go from Dementia to Normal between waves;
array DtoN{8} Dementia_cat5 Dementia_cat6 Dementia_cat7 Dementia_cat8 Dementia_cat9
Dementia_cat10 Dementia_cat11 Dementia_cat12;
      do i = 1 to 7;
      j = i + 1;
      if DtoN{j} = 3 & DtoN{i} = 1 then delete;
end;
drop i j;

*Step 10. If respondent has one dementia prediction, create Imputed_dementiacat
variable to flag any subsequent waves in which the respondent participated but
received a non-Dementia prediction (CIND or Normal);
%onceDimputeD(w=5);
%onceDimputeD(w=6);
%onceDimputeD(w=7);
%onceDimputeD(w=8);
%onceDimputeD(w=9);
%onceDimputeD(w=10);
%onceDimputeD(w=11);
%onceDimputeD(w=12);
drop dementia_seen INW4 INW5 INW6 INW7 INW8 INW9 INW10 INW11 INW12;
```

```
if Imputed_dementiacat5 = . & Imputed_dementiacat6 = . & Imputed_dementiacat7 = . &
Imputed_dementiacat8 = . &
Imputed_dementiacat9 = . & Imputed_dementiacat10 = . & Imputed_dementiacat11 = . &
Imputed_dementiacat12 = .
       then delete;

run;

*Step 11. Transpose data into longform;
data wide1;
set HRS_dementia_predictions;
keep HHIDPN Dementia_cat5-Dementia_cat12;
run;
*transpose data from wide to long on HHIDPN;
proc transpose data=wide1 out=long1;
  by HHIDPN;
run;
data long1;
*rename score column weight;
  set long1 (rename=(col1=predicted_category));
*create year var according to suffix of varname;
  if _name_ = "Dementia_cat5" then year = 2000;
  if _name_ = "Dementia_cat6" then year = 2002;
  if _name_ = "Dementia_cat7" then year = 2004;
  if _name_ = "Dementia_cat8" then year = 2006;
  if _name_ = "Dementia_cat9" then year = 2008;
  if _name_ = "Dementia_cat10" then year = 2010;
  if _name_ = "Dementia_cat11" then year = 2012;
  if _name_ = "Dementia_cat12" then year = 2014;
 *create unique ID per respondent per year;
  uniqueID =  catx("_",HHIDPN, year);
  drop _name_ _label_;
run;

*Predicted probability: normal;
data wide2;
set HRS_dementia_predictions;
keep HHIDPN N_predictedprob5-N_predictedprob12;
run;
*transpose data from wide to long on HHIDPN;
proc transpose data=wide2 out=long2;
  by HHIDPN;
run;
data long2;
*rename score column weight;
  set long2 (rename=(col1=prob_normal));
*create year var according to suffix of varname;
  if _name_ = "N_predictedprob5" then year = 2000;
  if _name_ = "N_predictedprob6" then year = 2002;
  if _name_ = "N_predictedprob7" then year = 2004;
  if _name_ = "N_predictedprob8" then year = 2006;
  if _name_ = "N_predictedprob9" then year = 2008;
  if _name_ = "N_predictedprob10" then year = 2010;
  if _name_ = "N_predictedprob11" then year = 2012;
  if _name_ = "N_predictedprob12" then year = 2014;
 *create unique ID per respondent per year;
  uniqueID =  catx("_",HHIDPN, year);
  drop _name_ _label_;
run;

*Predicted probability: CIND;
data wide3;
```

```
set HRS_dementia_predictions;
keep HHIDPN C_predictedprob5-C_predictedprob12;
run;
*transpose data from wide to long on HHIDPN;
proc transpose data=wide3 out=long3;
  by HHIDPN;
run;
data long3;
*rename score column weight;
  set long3 (rename=(col1=prob_cind));
*create year var according to suffix of varname;
  if _name_ = "C_predictedprob5" then year = 2000;
  if _name_ = "C_predictedprob6" then year = 2002;
  if _name_ = "C_predictedprob7" then year = 2004;
  if _name_ = "C_predictedprob8" then year = 2006;
  if _name_ = "C_predictedprob9" then year = 2008;
  if _name_ = "C_predictedprob10" then year = 2010;
  if _name_ = "C_predictedprob11" then year = 2012;
  if _name_ = "C_predictedprob12" then year = 2014;
 *create unique ID per respondent per year;
  uniqueID =  catx("_",HHIDPN, year);
  drop _name_ _label_;
run;

*Predicted probability: dementia;
data wide4;
set HRS_dementia_predictions;
keep HHIDPN D_predictedprob5-D_predictedprob12;
run;
*transpose data from wide to long on HHIDPN;
proc transpose data=wide4 out=long4;
  by HHIDPN;
run;
data long4;
*rename score column weight;
  set long4 (rename=(col1=prob_dementia));
*create year var according to suffix of varname;
  if _name_ = "D_predictedprob5" then year = 2000;
  if _name_ = "D_predictedprob6" then year = 2002;
  if _name_ = "D_predictedprob7" then year = 2004;
  if _name_ = "D_predictedprob8" then year = 2006;
  if _name_ = "D_predictedprob9" then year = 2008;
  if _name_ = "D_predictedprob10" then year = 2010;
  if _name_ = "D_predictedprob11" then year = 2012;
  if _name_ = "D_predictedprob12" then year = 2014;
 *create unique ID per respondent per year;
  uniqueID =  catx("_",HHIDPN, year);
  drop _name_ _label_;
run;

*Imputed dementia category;
data wide5;
set HRS_dementia_predictions;
keep HHIDPN Imputed_dementiacat5-Imputed_dementiacat12;
run;
*transpose data from wide to long on HHIDPN;
proc transpose data=wide5 out=long5;
  by HHIDPN;
run;
data long5;
*rename imputed category column Imputed_dementiacat;
  set long5 (rename=(col1=Imputed_dementiacat));
*create year var according to suffix of varname;
```

```
  if _name_ = "Imputed_dementiacat5" then year = 2000;
  if _name_ = "Imputed_dementiacat6" then year = 2002;
  if _name_ = "Imputed_dementiacat7" then year = 2004;
  if _name_ = "Imputed_dementiacat8" then year = 2006;
  if _name_ = "Imputed_dementiacat9" then year = 2008;
  if _name_ = "Imputed_dementiacat10" then year = 2010;
  if _name_ = "Imputed_dementiacat11" then year = 2012;
  if _name_ = "Imputed_dementiacat12" then year = 2014;
 *create unique ID per respondent per year;
  uniqueID =  catx("_",HHIDPN, year);
  drop _name_;
run;

*sort;
proc sort data = long1;
by uniqueID;
run;
proc sort data = long2;
by uniqueID;
run;
proc sort data = long3;
by uniqueID;
run;
proc sort data = long4;
by uniqueID;
run;
proc sort data = long5;
by uniqueID;
run;

*merge all long form data to make one dataset;
data HRS_dementia_predictions;
merge long1 long2 long3 long4 long5;
by uniqueID;
drop uniqueID;

run;
```