ADNI Alzheimer's Disease Neuroimaging Initiative

UWNPSYCHSUM now contains 4 composite measures.
ADNI_EF and ADNI_Mem are described first.
See pages 31-39 for ADNI_Lan and ADNI_VS.

Composite measures of executive function and memory: ADNI_EF and ADNI_Mem.

Laura E. Gibbons, PhD[1], Adam C. Carle, PhD[2], R. Scott Mackin, PhD[3], Danielle Harvey, PhD[4], Shubhabrata Mukherjee, PhD[1], Philip Insel, MS[3], S. McKay Curtis, PhD[1], Alden Gross, PhD, MHS[5], Richard N. Jones, ScD[5], Dan Mungas, PhD[6], Michael Weiner, MD[3], and Paul K. Crane, MD MPH[1], for the Alzheimer's Disease Neuroimaging Initiative*

1. University of Washington, Box 359780, Harborview Medical Center, 325 Ninth Avenue, Seattle, WA 98104, USA
2. University of Cincinnati School of Medicine, Cincinnati Children's Hospital Medical Center, and University of Cincinnati College of Arts and Sciences, 3333 Burnet Avenue, MLC 7014, Cincinnati, OH 45229, USA
3. Center for Imaging of Neurodegenerative Diseases (CIND), San Francisco VA Medical Center, 4150 Clement Street, San Francisco, CA 94121, USA
4. Division of Biostatistics, Department of Public Health Sciences, University of California, One Shields Avenue, Davis, CA 95616, USA.
5. Department of Psychiatry, Institute for Aging Research, Hebrew Senior Life, 1200 Center Street, Boston, MA 02131, USA
6. Department of Neurology, UC Davis Medical Center, 4860 Y Street, Sacramento, CA 95817, USA

Contents	

Summary

We derived composite scores for executive functioning (ADNI-EF) and memory (ADNI-MEM) using data from the ADNI neuropsychological battery using item response theory (IRT) methods. The formation of ADNI-MEM was complicated by the use of different word lists in the Rey Auditory Verbal Learning Test (RAVLT) and the ADAS-Cog, and by Logical Memory I data missing by design. ADNI-EF and ADNI-MEM have been validated in published papers[1,2]. The methods below are adapted from those papers and updated to include issues with ADNI 2 and ADNI GO.

Changes June 6, 2018

The ADNI-MEM scores posted in June 2018, based on data downloaded June 5, will not be consistent with previous files. Additional changes in ADNI 3 were accounted for, and the method for fixing the variances was refined. Our apologies.

Changes May 1, 2018

The ADNI-MEM scores posted May 1, 2018, based on data downloaded April 2, will not be consistent with previous files. Two changes were made. First, there were a few people who received scores when they had no data, because I didn't catch a change in how the strings for the 3 ADAS word recognition variables were encoded in ADNI 3. Second, in previous files the scores would vary slightly each time they were estimated because not all the variances were fixed in the bifactor model. Thanks to questions from careful users, both errors have been corrected.

Method

ADNI-EF

We used baseline data to develop ADNI-EF. Several of the authors (PKC, AC, and DM) reviewed the neuropsychological battery to identify items which could be considered indicators of EF. We refined our item selection using an iterative process in which we constructed a model using confirmatory factor analysis, reviewed findings as a small group, and then constructed a revised model. Our criteria for model fit were the confirmatory fit index (CFI), the Tucker Lewis Index (TLI) and the root mean squared error of approximation (RMSEA), where criteria for excellent fit include CFI>0.95, TLI >0.95, and RMSEA<0.05 [3]. We used Mplus (version 5) [4] with the theta parameterization and the WLSMV estimator. The final model for ADNI-EF included Category Fluency—animals, Category Fluency—vegetables, Trails A and B, Digit span backwards, WAIS-R Digit Symbol Substitution, and 5 Clock Drawing items (circle, symbol, numbers, hands, time).

These EF indicators utilized a variety of response formats that present challenges for constructing summary or composite measures. The formats include counts in a pre-specified time span (Category Fluency, WAIS-R Digit Symbol), times to completion (Trails), number of items completed correctly (Digit Span Backwards), and dichotomous correct/incorrect (clock drawing). We developed ordered categorical transformations of the raw data to facilitate development of composite scores that did not make strong assumptions about the distributions of scores. We recoded the raw scores into ordinal scales with as many as 10 categories, the maximum allowed by Mplus for ordinal variables. Our categorical transformations were based on the empirically observed distributions of the raw data, with a goal of maintaining variability in the tails at the expense of maintaining variability in the middle of the distributions. Specifics on how Category Fluency, WAIS-R Digit Symbol, Digit Span Backwards and Trails A and B were transformed are given in Table 1.

Table 1. Recoding of scores with more than 10 categories for ADNI-EF.

	Categorization									
Original EF Score	0	1	2	3	4	5	6	7	8	9
Category Fluency—Animals	0–5	6–7	8–9	10–12	13–16	17–20	21–23	24	25–27	28–60
Category Fluency—Vegetables	0–3	4–5	6	7–8	9–11	12–14	15–17	18	19–20	21–31

WAIS-R Digit Symbol	0–9	10–15	16–19	20–29	30–38	39–46	47–53	54–56	57–61	62–87
Digit Span Backwards	1–2	3	4	5	6	7	8	9	10	11–12
Trails A	118–150	94–117	73–93	53–72	40–52	32–39	27–31	24–26	21–23	5–20
Trails B	261–300	226–260	196–225	137–195	96–136	73–95	60–72	54–59	49–53	10–48

Bi-factor confirmatory factor analysis models[5] played an important role in ADNI-EF development. In this case, the single factor model was not a good representation of the data. Specifically, as shown in Figure 1, a bi-factor structure that included secondary domain structure for correlations between Category Fluency items and that included a methods factor for the Clock Drawing items produced much improved measures of model fit. We considered additional secondary structure, such as a residual correlation for the Trails tasks or for the two tests involving digits, but these residual correlations had a negligible impact on model fit and did not influence the loadings of other indicators on the primary factor. These considerations resulted in our final confirmatory factor analysis bi-factor model, shown in Figure 1. Model fit was excellent. The CFI was 0.99, the TLI was 0.99, and the RMSEA was 0.049.

We defined the metric for ADNI-EF to have a mean of 0, and standard deviation of 1, based on the 800 participants with complete EF data at baseline. We used item parameters (loadings and thresholds) from the baseline model to compute scores at each follow-up visit. The resulting ADNI-EF scores are available in the ADNI UWNPSYCHSUM file. Final Mplus code to generate the scores is in Appendix 1.

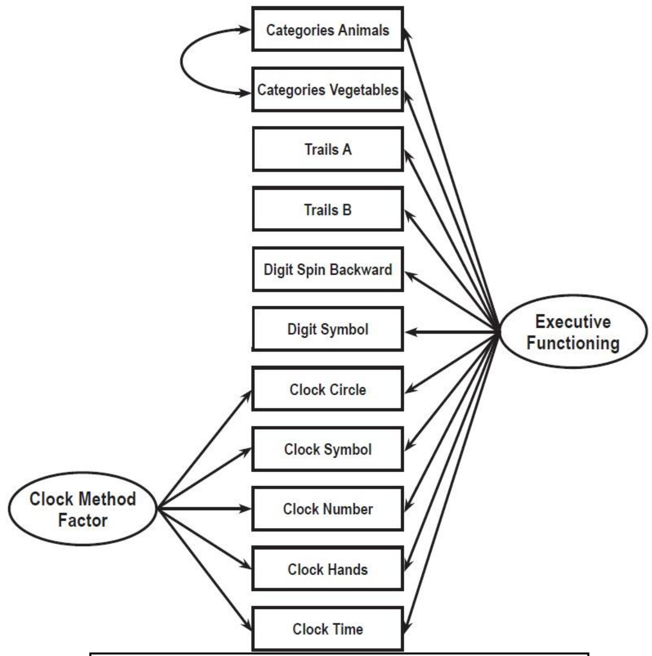


Figure 1. Bifactor model for ADNI-EF

ADNI 2, ADNI GO, and ADNI 3 did not administer Category Fluency - Vegetables, Digit Span Backwards, or WAIS-R Digit Symbol. An advantage of IRT scoring is that we can still compute EF scores in ADNI 2, ADNI GO, and ADNI 3using the other eight items and their item parameters. The measurement error will be larger because of the missing items, but the scores are still valid and ***on the same metric*** as scores based on all the items. This means that scores from the four phases are directly comparable. By the same token, if anyone was missing an item at random, we still were able to calculate a score.

ADNI-MEM

The memory items were also selected based on theory, and analyzed in Mplus, with recoded variables (Table 2).

Table 2 ADNI-Mem items and their recoding.

	Categorization									
Original Memory Score	0	1	2	3	4	5	6	7	8	9
<u>RAVLT</u>										
Trial 1	0/2	3	4	5	6	7	8	9	10	11/15
Trial 2	0/2	3	4	5	6	7	8	9	10	11/15
Trial 3	0/2	3	4	5/6	7/8	9	10	11	12	13/14
Trial 4	0/3	4	5/6	7/8	9	10	11	12	13	14/15
Trial 5	0/3	4	5	6/7	8/9	10/11	12	13	14	15
Interference	0/1	2	3	4	5	6	7	8/15		
Immediate recall	0	1/2	3/4	5/6	7	8	9	10/11	12/13	14/15
30 minute delay	0	1/2	3/4	5/6	7	8	9	10/11	12/13	14/15
Recognition	0	1	2/3	4/5	6/7	8/9	10/11	12/13	14	15
<u>ADAS-Cog</u>										
Trial 1	0/1	2	3	4	5	6	7	8/10		

Trial 2	0/2	3	4	5	6	7	8	9	10	
Trial 3	0/2	3	4	5	6	7	8	9	10	
Recall	0	1	2	3	4	5	6	7	8	9/10
Recognition present	0/3	4	5	6	7	8	9	10	11	12
Recognition absent	0/4	5/6	7	8	9	10	11	12		
<u>Logical Memory*</u>										
Immediate	0/1	2/3	4/5	6/7	8/9	10/12	13/14	15/16	17/18	19/25
Delay	0	1/2	3/4	5/8	9/11	12	13	14/15	16/17	18/25
<u>MMSE*</u>										
Ball recall	2	1								
Flag recall	2	1								
Tree recall	2	1								

*Values from the screening visit were used as baseline scores.

The task of modeling the memory data was complicated by the three forms of the ADAS-Cog word lists and the two forms of the RAVLT word lists. It was not clear if these word lists were equivalent; indeed past experience suggested that the RAVLT lists may be problematic. Furthermore, Logical Memory was only assessed at annual visits. The only indicators consistently present across visits were the three word recall items from the MMSE. Technically these three dichotomous indicators could be used to anchor the scales across time points[6], but we were concerned that this anchoring would be too sparse for valid conclusions to be drawn.

Longitudinal bi-factor models incorporating the different versions of the tests and the missing Logical Memory scores were also theoretically possible, but in practice proved impossible to fit with these items in the ADNI data set. Therefore we investigated whether a single factor model would be appropriate for the longitudinal data. Full details will be available in the published paper, but here is a brief summary. Using the baseline data, a bi-factor model that accounted for methods effects had superior fit compared to a single factor model, and to bi-factor models using content-based subdomains. But based on a comparison of loadings on the overall memory factor and the correlation between the single and bi-factor scores (0.99), we decided it was acceptable to use a single factor model.

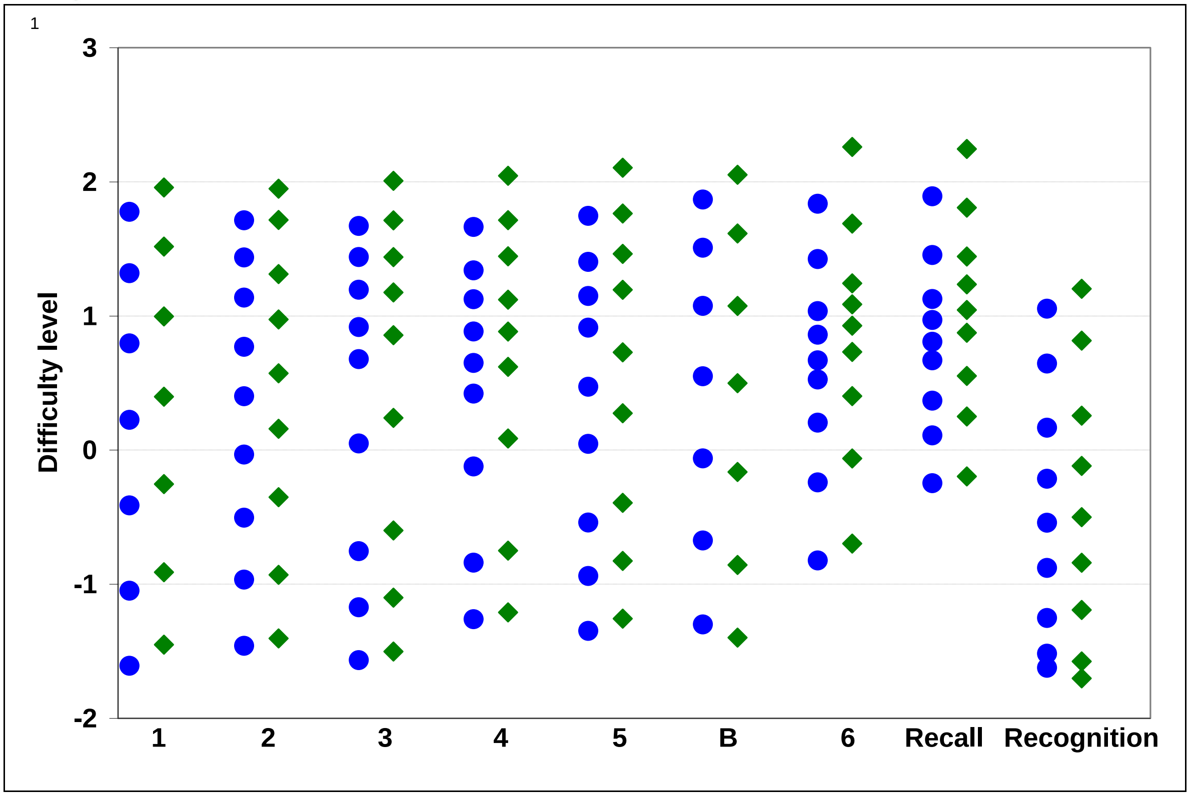
Next we divided the data set into two parts: first, the annual visits with data from all participants (baseline, month 12, and month 24), and second, the other visits (month 6, 18, and 36; the month 36 visit had structurally missing data from people in the AD group). Logical Memory was assessed at each of the visits in the first half of the data set, so those much richer indicators were used as anchors alongside the three dichotomous MMSE indicators. Furthermore, at each of those visits, only the first version of the RAVLT was assessed, so it could also act as an anchor. The only thing that varied at those visits was thus the three different versions of the ADAS-Cog. We fit a longitudinal model using all available data for the annual visits of the first half of the data set. We identified the scale by specifying the variance of the general factor to be 1 at the baseline visit, when its mean was 0. We allowed the mean and the variance of the general factor to vary at other time points, and the general factors were freely correlated with each other. We freely estimated the loadings on the general factor, but constrained those loadings from the same indicators to be the same across time points. For example, for the first MMSE item, we freely estimated the loading on the overall memory factor at each time point, but constrained that loading to be the same at baseline, month 12, and month 24.

We captured point estimates for the loadings and thresholds for the three MMSE items, Logical Memory I and II, and the three versions of the ADAS-Cog from the first half of the data set. We then turned our attention to the second half of the data set that included data from study visits at months 6, 18, and 36. The second version of the RAVLT word list was used at each of these study visits. We used the MMSE items, the ADAS-Cog version 2 (month 6), version 1 (month 18), and version 3 (month 36), and Logical Memory (month 36) as anchors to estimate item parameters for the second version of the RAVLT. The longitudinal modeling strategy was similar to that described for the first half of the data. Because we were fixing item loadings and thresholds for the anchor items, the scale was still anchored to the mean of 0 and variance of 1 at the baseline visit. We freely estimated the means and variances at each of the study visits included in this second half of the data. Script files for the developmental analyses are available from the authors. Final code to estimate the memory scores is in Appendix 2. We extracted the ADNI-MEM factor scores for each participant at each study visit. These analyses were based on data collected through the 36-month visit. The 48-month neuropsych battery had the same memory items in the same versions as the baseline data, so baseline parameters were used to compute 48-month scores. The ADNI-MEM scores are available in the UWNPSYCHSUM file.

Version effects for the RAVLT and the ADAS-Cog.

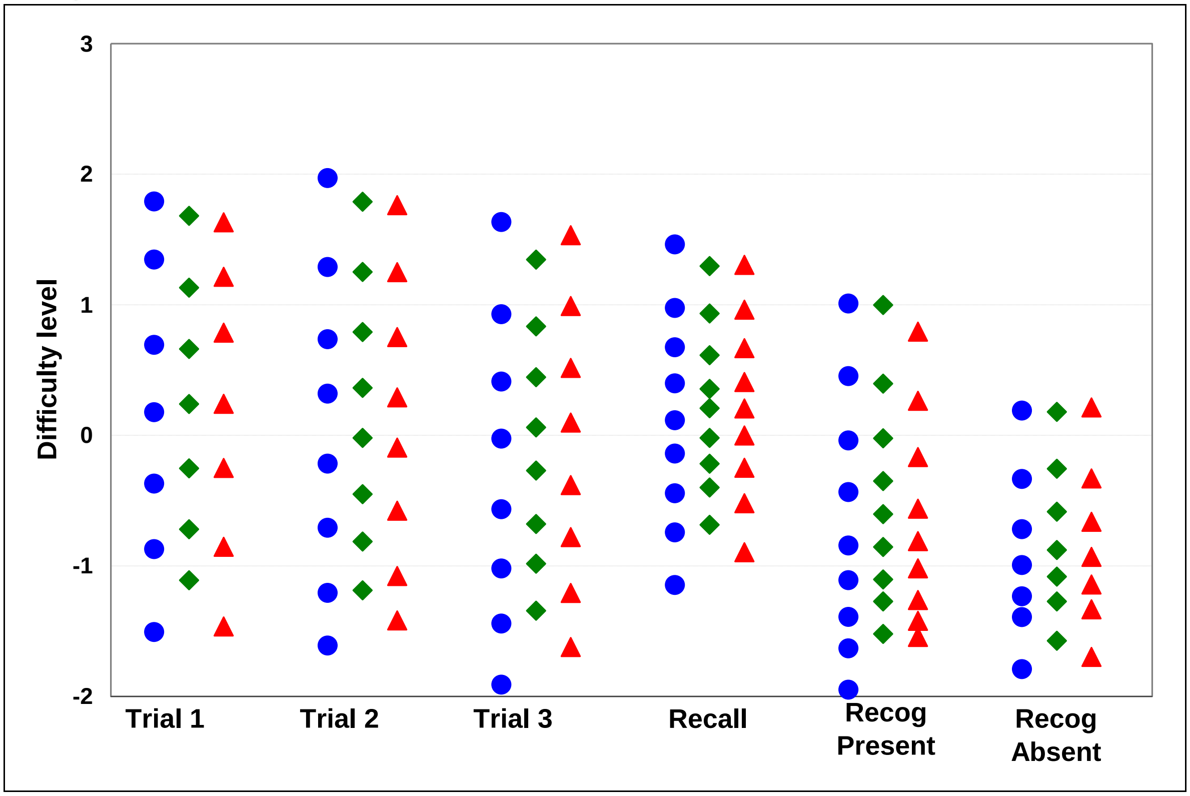
The loadings for each of the indicators from the two versions of the RAVLT were very similar; as a proportion, they ranged from 5% smaller to 3% larger between the two versions. The difficulty levels for the category thresholds, however, displayed important differences between the two versions, as shown in the plot of the item thresholds in Figure 2. For all of the trials with the exception of List B (the distractor list), the Version B list is more difficult (has higher thresholds) than the Version A list.

Fig. 2 Difficulty levels for the elements of the two versions of the RAVLT. The five learning trials are indicated by the numbers 1 through 5; the interference trial by the letter B, the first recall trial by the number 6; delayed recall by "Recall", and the recognition task by "Recognition". Version A difficulty thresholds are denoted with blue circles, while version B difficulty thresholds are denoted with green diamonds. In this plot, the difficulty levels are plotted on the y axis in z-statistic units; higher numbers indicate higher memory ability / higher item difficulty. Considering the two versions of learning trial 1, version A is easier for each threshold. At an overall memory ability level of -0.5, for example, higher proportions of people will be above the first threshold for version A, and lower proportions of people above that same threshold for version B. At every threshold the green diamonds are higher than the blue dots. For the second through 5th learning trials, this difference is dramatic at the top end, as the top threshold on version A is only as difficult as the 2nd to highest threshold on version B.



The ADAS-Cog versions were more similar to each other, at least in terms of category thresholds (see Figure 3). Version 1 had a greater spread of thresholds than Version 2 and to a lesser extent than Version 3, which means that it should be somewhat better able to differentiate among people at the extremes of memory ability with fewer ceiling or floor scores. The loadings for the learning trials and recall of the three versions of the ADAS-Cog list learning task were very similar to each other, with differences ranging from 4 percent lower to 2 percent higher. The recognition present and recognition absent tasks had somewhat dissimilar loadings. In no case were these strong indicators of overall memory (standardized loadings ranged from 0.43 to 0.56, roughly half the magnitude of loadings for the list learning indicators). The largest overall difference in loading between versions was 0.13 for recognition correct between Version A and Version C, which in terms of percentage was a 30% difference in loadings.

Fig. 3 Loadings for the memory items from the three versions of the ADAS-Cog. Recog = Recognition. Version 1 threshold difficulty levels are depicted with blue circles, Version 2 with green diamonds, and Version 3 with red triangles. In this plot, the difficulty levels are plotted on the y axis in z-statistic units; higher numbers indicate higher memory ability / higher item difficulty Version 1 has greater spread than Version 2 and to a lesser extent than version 3, meaning it will have slightly smaller ceiling and floor effects. Unlike the RAVLT, no version appears to be consistently easier or harder than the other versions.



ADNI 3 did not administer the MMSE. As with ADNI-EF, the measurement error will be larger because of the missing items, but the scores are still valid and ***on the same metric*** as scores based on all the items.

Spanish Language

A few participants in ADNI2/GO are being tested in Spanish. We have computed their scores using the item parameters derived in English tests. So far there are too few Spanish test-takers to evaluate whether this is appropriate. If you want to omit their scores, look for even-numbered values of the variable "WORDLIST" in the adas_adnigo2 file.

References

1. Gibbons LE, Carle AC, Mackin RS, et al. A composite score for executive functioning, validated in Alzheimer's Disease Neuroimaging Initiative (ADNI) participants with baseline mild cognitive impairment. *Brain Imaging Behav*. 2012.
2. Crane PK, Carle A, Gibbons LE, et al. Development and assessment of a composite score for memory in the Alzheimer's Disease Neuroimaging Initiative (ADNI). *Brain Imaging Behav*. 2012.

3. Reeve BB, Hays RD, Bjorner JB, et al. Psychometric evaluation and calibration of health-related quality of life item banks: plans for the Patient-Reported Outcomes Measurement Information System (PROMIS). *Med Care.* 2007;45(5 Suppl 1):S22-31.
4. *Mplus: statistical analysis with latent variables* [computer program]. Version 5.1. Los Angeles, CA: Muthén & Muthén; 1998-2007.
5. McDonald RP. *Test theory: a unified treatment.* Mahwah, N.J.: Lawrence Erlbaum; 1999.
6. Reise SP, Widaman KF, Pugh RH. Confirmatory factor analysis and item response theory: Two approaches for exploring measurement invariance. *Psychological Bulletin.* 1993;114(3):552-566.

About the Authors

This document was prepared by Laura Gibbons at the University of Washington. For more information or to ask for updated files, please contact Laura Gibbons by email at gibbonsl@uw.edu or at 1-206-744-1842.

Appendix 1. Mplus code for ADNI-EF

Note that the following ADNI variables were recoded as in Table 1.

EF item	ADNI Name	Recoded Name
Category Fluency—Animals	catanimsc	mecatat
Category Fluency—Vegetables	catvegesc	mecatvt
WAIS-R Digit Symbol	digitscor	medigit
Digit Span Backwards	dspanbac	medsbc
Trails A	traascor	metatne
Trails B	trabscor	metbtne

You'll need to add lines to refer to your data file and the missing value codes.

```
VARIABLE:
     NAMES = rid mecatat mecatvt medigit medsbc metatne metbtne clockcirc
             clocksym clocknum clockhand clocktime ;
     CATEGORICAL = mecatat-clocktime;
     IDVARIABLE = rid;
ANALYSIS:
     PARAMETERIZATION = theta;
  OUTPUT:
     STANDARDIZED;
  SAVEDATA:
     file is ef_scores_bifactor.txt; save=fscores;

  MODEL:
  adni_ef by mecatat @ 0.732 ;
  adni_ef by mecatvt @ 0.755 ;
  adni_ef by medsbc @ 0.599 ;
  adni_ef by metatne @ 1.095 ;
  adni_ef by metbtne @ 1.594 ;
```

```
adni_ef by medigit @ 1.627 ;
adni_ef by clockcirc @ 0.423 ;
adni_ef by clocksym @ 0.610 ;
adni_ef by clocknum @ 0.656 ;
adni_ef by clockhand @ 1.959 ;
adni_ef by clocktime @ 0.985 ;
clock by clockcirc @ 0.486 ;
clock by clocksym @ 0.413 ;
clock by clocknum @ 0.863 ;
clock by clockhand @ 2.040 ;
clock by clocktime @ 0.810 ;
adni_ef with clock @ 0.000 ;
mecatat with mecatvt @ 0.444 ;
adni_ef @ 1.000 ;
clock @ 1.000 ;
[mecatat$1 @ -2.456] ;
[mecatat$2 @ -1.980] ;
[mecatat$3 @ -1.488] ;
[mecatat$4 @ -0.816] ;
[mecatat$5 @ 0.097] ;
[mecatat$6 @ 0.947] ;
[mecatat$7 @ 1.545] ;
[mecatat$8 @ 1.795] ;
[mecatat$9 @ 2.331] ;
[mecatvt$1 @ -2.405] ;
[mecatvt$2 @ -1.782] ;
[mecatvt$3 @ -1.465] ;
[mecatvt$4 @ -0.763] ;
[mecatvt$5 @ 0.158] ;
[mecatvt$6 @ 1.027] ;
[mecatvt$7 @ 1.720] ;
[mecatvt$8 @ 1.975] ;
[mecatvt$9 @ 2.456] ;
[medsbc$1 @ -2.151] ;
[medsbc$2 @ -1.698] ;
[medsbc$3 @ -0.940] ;
[medsbc$4 @ -0.288] ;
[medsbc$5 @ 0.333] ;
[medsbc$6 @ 0.781] ;
[medsbc$7 @ 1.246] ;
[medsbc$8 @ 1.581] ;
[medsbc$9 @ 2.024] ;
[metatne$1 @ -2.663] ;
[metatne$2 @ -2.305] ;
[metatne$3 @ -1.839] ;
[metatne$4 @ -0.994] ;
[metatne$5 @ -0.028] ;
[metatne$6 @ 0.914] ;
[metatne$7 @ 1.733] ;
[metatne$8 @ 2.370] ;
[metatne$9 @ 2.875] ;
[metbtne$1 @ -2.033] ;
[metbtne$2 @ -1.814] ;
[metbtne$3 @ -1.558] ;
[metbtne$4 @ -0.782] ;
[metbtne$5 @ 0.260] ;
[metbtne$6 @ 1.437] ;
[metbtne$7 @ 2.494] ;
[metbtne$8 @ 3.007] ;
[metbtne$9 @ 3.610] ;
[medigit$1 @ -3.921] ;
```

```
[medigit$2 @ -3.141] ;
[medigit$3 @ -2.489] ;
[medigit$4 @ -1.228] ;
[medigit$5 @ 0.090] ;
[medigit$6 @ 1.419] ;
[medigit$7 @ 2.461] ;
[medigit$8 @ 2.910] ;
[medigit$9 @ 3.703] ;
[clockcirc$1 @ -2.581] ;
[clocksym$1 @ -0.733] ;
[clocknum$1 @ -1.602] ;
[clockhand$1 @ -4.513] ;
[clocktime$1 @ -0.663] ;
```

Appendix 2. Stata and Mplus code for ADNI-Mem.

We used the Stata program –runmplus-, which is a free user-written download available by typing "ssc install runmplus" in Stata. You must also have Mplus on your computer for this to work.

The memory items were recoded as in Table 2.

Memory item	ADNI Name	Recoded Name
RAVLT		
Trial 1	avtot1	mmra1
Trial 2	avtot2	mmra2
Trial 3	avtot3	mmra3
Trial 4	avtot4	mmra4
Trial 5	avtot5	mmra5
Interference	avtotb	mmrab
Immediate recall	avtot6	mmra6
30 minute delay	avdel30min	mmradrc
Recognition	avdeltot	mmrarc
ADAS-Cog		
Trial 1	cot1sco	mmadlt1
Trial 2	cot2sco	mmadlt2
Trial 3	cot3sco	mmadlt3
Recall	cot4tot	mmadd
Recognition present	*	mmadrg1
Recognition absent	*	mmadrg2
Logical Memory*		
Immediate	limmtotal	mmlmrc
Delay	ldeltotal	mmlmd
MMSE*		
Ball recall	balldl	mmballdl
Flag recall	flagdl	mmflagdl
Tree recall	treedl	mmtreedl

* see code below.

Stata code to create the ADAS recognition items

```
egen adrg1=rowtotal(co8magaz co8wizrd co8van co8leprd co8sea co8train ///
       co8coin co8inst co8board co8anchr co8gem co8fund) if inlist(visnum,0,18,48)
```

```
tempvar a6 a12 b6 b12
egen `a6'=   rowtotal(co8nurse co8wizrd co8leprd co8sea co8ship co8map ///
       co8milk co8vol co8forst co8fund co8edge co8cake) if visnum==6 | visnum==24
replace adrg1=`a6' if visnum==6 | visnum==24
egen `a12'=rowtotal(co8nurse co8wizrd co8sale co8sea co8train co8map ///
       co8axe co8milk co8vol co8gem co8cat co8edge) if visnum==12 | visnum==36
replace adrg1=`a12' if visnum==12 | visnum==36
replace adrg1=. if missing(co8magaz, co8wizrd,co8van, co8leprd, co8sea, co8train, ///
       co8coin, co8inst, co8board, co8anchr, co8gem, co8fund)
la var adrg1 "ADAS-cog correctly identified as seen"

egen adrg2=rowtotal(co8nurse co8sale co8ship co8map co8axe co8carrt ///
       co8milk co8vol co8forst co8cat co8edge co8cake) if inlist(visnum,0,18,48)
egen `b6'=rowtotal(co8magaz co8van co8sale co8train co8coin co8inst ///
       co8axe co8board co8carrt co8anchr co8gem co8cat) if visnum==6 | visnum==24
replace adrg2=`b6'  if visnum==6 | visnum==24
egen `b12'=rowtotal(co8magaz co8van co8leprd co8coin co8ship co8inst ///
       co8board co8carrt co8forst co8anchr co8fund co8cake) if visnum==12 | visnum==36
replace adrg2=`b12' if visnum==12 | visnum==36
replace adrg2=12-adrg2
replace adrg2=. if missing(co8nurse, co8sale, co8ship, co8map, co8axe, co8carrt, ///
       co8milk, co8vol, co8forst, co8cat, co8edge, co8cake)
la var adrg2 "ADAS-cog correctly identified as not seen"
```

<u>Stata and Mplus code (runmplus) to generate ADNI-Mem</u>

This is based on having the recoded items in a data set called ADNI_Mem.dta, and that you have a numeric variable for viscode (visit number) called “visnum”.

```
* use Paul's codes to get memory scores
* unwieldy, but safest to use what he did.

use ADNI_Mem, clear
keep mmlmrc-mmtreedl mmrarc rid visnum phase
* how he did it
rename mmlmrc lmrc
rename mmlmd lmd
rename mmradrc radrc
rename mmrarc rarc
rename mmra6 ra6
rename mmadrg1 adrg1
rename mmadrg2 adrg2
rename mmadlt1 adlt1
rename mmadlt2 adlt2
rename mmadlt3 adlt3
rename mmadd add
rename mmra1 ra1
rename mmra2 ra2
rename mmra3 ra3
rename mmra4 ra4
rename mmra5 ra5
rename mmrab rab
rename mmballdl balldl
rename mmflagdl flagdl
rename mmtreedl treedl
gen idvis=rid*1000+visnum

gen version=.
```

```
*1=English Word List 1; 3=English Word List 2; 5=English Word List 3; 7=English Word
List 4;
*2=Spanish Word List 1; 4=Spanish Word List 2; 6=Spanish Word List 3; 8=Spanish Word
List 4
* only 1-4 so far. see "ADNI versions.docx" for why this isn't used.

*************** update if needed ***********************************
replace version=1 if inlist(visnum,0,48,60,72,84,96,108,120,132,144,156,168,180) ///
       | (inlist(visnum,12,24,36) & phase~="ADNI1")             // 1 A LM
replace version=2 if inlist(visnum,6)                                  // 2 B
replace version=3 if inlist(visnum,12) & phase=="ADNI1"  // 3 A LM
replace version=4 if inlist(visnum,18)                                 // 1 B
replace version=5 if inlist(visnum,24)  & phase=="ADNI1" // 2 A LM
replace version=6 if inlist(visnum,36)  & phase=="ADNI1" // 3 B LM

save temp, replace
set more off

* version1

*** problem with 4526 month 24 *********************************
use temp, clear
keep if version==1
runmplus idvis lmrc-rarc, categorical(lmrc-rarc) id(idvis) ///
       model ( ///
mem @ 0.997; ///
mem    by     lmrc   @      0.837  ;      ///
mem    by     lmd    @      0.846  ;      ///
mem    by     radrc  @      0.876  ;      ///
mem    by     rarc   @      0.728  ;      ///
mem    by     ra6    @      0.85   ;      ///
mem    by     adrg1  @      0.43   ;      ///
mem    by     adrg2  @      0.507  ;      ///
mem    by     adlt1  @      0.792  ;      ///
mem    by     adlt2  @      0.858  ;      ///
mem    by     adlt3  @      0.844  ;      ///
mem    by     add    @      0.898  ;      ///
mem    by     ra1    @      0.661  ;      ///
mem    by     ra2    @      0.807  ;      ///
mem    by     ra3    @      0.852  ;      ///
mem    by     ra4    @      0.884  ;      ///
mem    by     ra5    @      0.882  ;      ///
mem    by     rab    @      0.615  ;      ///
mem    by     balldl @      0.748  ;      ///
mem    by     flagdl @      0.777  ;      ///
mem    by     treedl @      0.751  ;      ///
[      lmrc$1 @      -1.369 ]      ;      ///
[      lmrc$2 @      -0.841 ]      ;      ///
[      lmrc$3 @      -0.412 ]      ;      ///
[      lmrc$4 @      -0.079 ]      ;      ///
[      lmrc$5 @      0.225  ]      ;      ///
[      lmrc$6 @      0.638  ]      ;      ///
[      lmrc$7 @      0.946  ]      ;      ///
[      lmrc$8 @      1.296  ]      ;      ///
[      lmrc$9 @      1.702  ]      ;      ///
[      lmd$1  @      -0.637 ]      ;      ///
[      lmd$2  @      -0.289 ]      ;      ///
[      lmd$3  @      0.006  ]      ;      ///
[      lmd$4  @      0.458  ]      ;      ///
[      lmd$5  @      0.74   ]      ;      ///
[      lmd$6  @      0.854  ]      ;      ///
```

```
[	lmd$7	@	0.972	]	;	///
[	lmd$8	@	1.267	]	;	///
[	lmd$9	@	1.676	]	;	///
[	radrc$1	@	-0.248	]	;	///
[	radrc$2	@	0.109	]	;	///
[	radrc$3	@	0.368	]	;	///
[	radrc$4	@	0.668	]	;	///
[	radrc$5	@	0.807	]	;	///
[	radrc$6	@	0.97	]	;	///
[	radrc$7	@	1.126	]	;	///
[	radrc$8	@	1.453	]	;	///
[	radrc$9	@	1.891	]	;	///
[	rarc$1	@	-1.624	]	;	///
[	rarc$2	@	-1.519	]	;	///
[	rarc$3	@	-1.253	]	;	///
[	rarc$4	@	-0.88	]	;	///
[	rarc$5	@	-0.543	]	;	///
[	rarc$6	@	-0.215	]	;	///
[	rarc$7	@	0.166	]	;	///
[	rarc$8	@	0.644	]	;	///
[	rarc$9	@	1.053	]	;	///
[	ra6$1	@	-0.824	]	;	///
[	ra6$2	@	-0.242	]	;	///
[	ra6$3	@	0.203	]	;	///
[	ra6$4	@	0.526	]	;	///
[	ra6$5	@	0.668	]	;	///
[	ra6$6	@	0.859	]	;	///
[	ra6$7	@	1.036	]	;	///
[	ra6$8	@	1.424	]	;	///
[	ra6$9	@	1.835	]	;	///
[	adrg1$1	@	-1.949	]	;	///
[	adrg1$2	@	-1.632	]	;	///
[	adrg1$3	@	-1.392	]	;	///
[	adrg1$4	@	-1.111	]	;	///
[	adrg1$5	@	-0.844	]	;	///
[	adrg1$6	@	-0.436	]	;	///
[	adrg1$7	@	-0.04	]	;	///
[	adrg1$8	@	0.453	]	;	///
[	adrg1$9	@	1.009	]	;	///
[	adrg2$1	@	-1.791	]	;	///
[	adrg2$2	@	-1.393	]	;	///
[	adrg2$3	@	-1.234	]	;	///
[	adrg2$4	@	-0.995	]	;	///
[	adrg2$5	@	-0.72	]	;	///
[	adrg2$6	@	-0.335	]	;	///
[	adrg2$7	@	0.189	]	;	///
[	adlt1$1	@	-1.507	]	;	///
[	adlt1$2	@	-0.872	]	;	///
[	adlt1$3	@	-0.371	]	;	///
[	adlt1$4	@	0.177	]	;	///
[	adlt1$5	@	0.693	]	;	///
[	adlt1$6	@	1.346	]	;	///
[	adlt1$7	@	1.791	]	;	///
[	adlt2$1	@	-1.61	]	;	///
[	adlt2$2	@	-1.208	]	;	///
[	adlt2$3	@	-0.709	]	;	///
[	adlt2$4	@	-0.217	]	;	///
[	adlt2$5	@	0.319	]	;	///
[	adlt2$6	@	0.736	]	;	///
[	adlt2$7	@	1.288	]	;	///
[	adlt2$8	@	1.97	]	;	///
```

```
[	adlt3$1	@	-1.91	]	;	///
[	adlt3$2	@	-1.443	]	;	///
[	adlt3$3	@	-1.02	]	;	///
[	adlt3$4	@	-0.566	]	;	///
[	adlt3$5	@	-0.026	]	;	///
[	adlt3$6	@	0.411	]	;	///
[	adlt3$7	@	0.927	]	;	///
[	adlt3$8	@	1.633	]	;	///
[	add$1	@	-1.147	]	;	///
[	add$2	@	-0.744	]	;	///
[	add$3	@	-0.444	]	;	///
[	add$4	@	-0.14	]	;	///
[	add$5	@	0.115	]	;	///
[	add$6	@	0.397	]	;	///
[	add$7	@	0.674	]	;	///
[	add$8	@	0.975	]	;	///
[	add$9	@	1.461	]	;	///
[	ra1$1	@	-1.609	]	;	///
[	ra1$2	@	-1.049	]	;	///
[	ra1$3	@	-0.413	]	;	///
[	ra1$4	@	0.225	]	;	///
[	ra1$5	@	0.795	]	;	///
[	ra1$6	@	1.319	]	;	///
[	ra1$7	@	1.776	]	;	///
[	ra2$1	@	-1.46	]	;	///
[	ra2$2	@	-0.966	]	;	///
[	ra2$3	@	-0.505	]	;	///
[	ra2$4	@	-0.034	]	;	///
[	ra2$5	@	0.401	]	;	///
[	ra2$6	@	0.769	]	;	///
[	ra2$7	@	1.137	]	;	///
[	ra2$8	@	1.436	]	;	///
[	ra2$9	@	1.713	]	;	///
[	ra3$1	@	-1.567	]	;	///
[	ra3$2	@	-1.172	]	;	///
[	ra3$3	@	-0.755	]	;	///
[	ra3$4	@	0.048	]	;	///
[	ra3$5	@	0.677	]	;	///
[	ra3$6	@	0.917	]	;	///
[	ra3$7	@	1.195	]	;	///
[	ra3$8	@	1.439	]	;	///
[	ra3$9	@	1.67	]	;	///
[	ra4$1	@	-1.261	]	;	///
[	ra4$2	@	-0.84	]	;	///
[	ra4$3	@	-0.122	]	;	///
[	ra4$4	@	0.42	]	;	///
[	ra4$5	@	0.649	]	;	///
[	ra4$6	@	0.884	]	;	///
[	ra4$7	@	1.124	]	;	///
[	ra4$8	@	1.339	]	;	///
[	ra4$9	@	1.663	]	;	///
[	ra5$1	@	-1.348	]	;	///
[	ra5$2	@	-0.94	]	;	///
[	ra5$3	@	-0.541	]	;	///
[	ra5$4	@	0.045	]	;	///
[	ra5$5	@	0.472	]	;	///
[	ra5$6	@	0.912	]	;	///
[	ra5$7	@	1.149	]	;	///
[	ra5$8	@	1.402	]	;	///
[	ra5$9	@	1.746	]	;	///
[	rab$1	@	-1.3	]	;	///
```

```
[       rab$2  @       -0.675 ]       ;       ///
[       rab$3  @       -0.063 ]       ;       ///
[       rab$4  @       0.549  ]       ;       ///
[       rab$5  @       1.075  ]       ;       ///
[       rab$6  @       1.508  ]       ;       ///
[       rab$7  @       1.868  ]       ;       ///
[       balldl$1       @       -0.509 ]       ;       ///
[       flagdl$1       @       -0.033 ]       ;       ///
[       treedl$1       @       -0.127 ]       ;       ///
) ///
   savedata(save=fscores; file=trash.dat) savelog(trash)

preserve
infix idvis 212-288 mem 203-210 using trash.dat, clear
tempfile f1
save `f1'
restore
merge 1:1 idvis using `f1', keep(3)
keep idvis mem
capture save "mem1", replace


* version 2
use temp, clear
keep if version==2
runmplus idvis radrc-rarc, categorical(radrc-rarc) id(idvis) ///
       model ( ///
MEM @1.073; ///
MEM    by     radrc  @       0.885  ;       ///
MEM    by     rarc   @       0.716  ;       ///
MEM    by     ra6    @       0.862  ;       ///
MEM    by     adrg1  @       0.478  ;       ///
MEM    by     adrg2  @       0.533  ;       ///
MEM    by     adlt1  @       0.79   ;       ///
MEM    by     adlt2  @       0.828  ;       ///
MEM    by     adlt3  @       0.835  ;       ///
MEM    by     add    @       0.862  ;       ///
MEM    by     ra1    @       0.649  ;       ///
MEM    by     ra2    @       0.826  ;       ///
MEM    by     ra3    @       0.876  ;       ///
MEM    by     ra4    @       0.884  ;       ///
MEM    by     ra5    @       0.877  ;       ///
MEM    by     rab    @       0.582  ;       ///
MEM    by     balldl @       0.748  ;       ///
MEM    by     flagdl @       0.777  ;       ///
MEM    by     treedl @       0.751  ;       ///
[       radrc$1        @       -0.196 ]       ;       ///
[       radrc$2        @       0.25   ]       ;       ///
[       radrc$3        @       0.553  ]       ;       ///
[       radrc$4        @       0.875  ]       ;       ///
[       radrc$5        @       1.044  ]       ;       ///
[       radrc$6        @       1.235  ]       ;       ///
[       radrc$7        @       1.443  ]       ;       ///
[       radrc$8        @       1.807  ]       ;       ///
[       radrc$9        @       2.245  ]       ;       ///
[       rarc$1 @       -1.702 ]       ;       ///
[       rarc$2 @       -1.576 ]       ;       ///
[       rarc$3 @       -1.192 ]       ;       ///
[       rarc$4 @       -0.841 ]       ;       ///
[       rarc$5 @       -0.5   ]       ;       ///
[       rarc$6 @       -0.118 ]       ;       ///
```

```
[       rarc$7 @       0.257  ]       ;       ///
[       rarc$8 @       0.816  ]       ;       ///
[       rarc$9 @       1.203  ]       ;       ///
[       ra6$1  @       -0.697 ]       ;       ///
[       ra6$2  @       -0.062 ]       ;       ///
[       ra6$3  @       0.402  ]       ;       ///
[       ra6$4  @       0.732  ]       ;       ///
[       ra6$5  @       0.927  ]       ;       ///
[       ra6$6  @       1.087  ]       ;       ///
[       ra6$7  @       1.243  ]       ;       ///
[       ra6$8  @       1.688  ]       ;       ///
[       ra6$9  @       2.26   ]       ;       ///
[       adrg1$1        @       -1.521 ]       ;       ///
[       adrg1$2        @       -1.272 ]       ;       ///
[       adrg1$3        @       -1.104 ]       ;       ///
[       adrg1$4        @       -0.856 ]       ;       ///
[       adrg1$5        @       -0.603 ]       ;       ///
[       adrg1$6        @       -0.35  ]       ;       ///
[       adrg1$7        @       -0.023 ]       ;       ///
[       adrg1$8        @       0.395  ]       ;       ///
[       adrg1$9        @       0.998  ]       ;       ///
[       adrg2$1        @       -1.574 ]       ;       ///
[       adrg2$2        @       -1.272 ]       ;       ///
[       adrg2$3        @       -1.083 ]       ;       ///
[       adrg2$4        @       -0.879 ]       ;       ///
[       adrg2$5        @       -0.585 ]       ;       ///
[       adrg2$6        @       -0.256 ]       ;       ///
[       adrg2$7        @       0.179  ]       ;       ///
[       adlt1$1        @       -1.111 ]       ;       ///
[       adlt1$2        @       -0.719 ]       ;       ///
[       adlt1$3        @       -0.253 ]       ;       ///
[       adlt1$4        @       0.24   ]       ;       ///
[       adlt1$5        @       0.662  ]       ;       ///
[       adlt1$6        @       1.131  ]       ;       ///
[       adlt1$7        @       1.681  ]       ;       ///
[       adlt2$1        @       -1.187 ]       ;       ///
[       adlt2$2        @       -0.814 ]       ;       ///
[       adlt2$3        @       -0.45  ]       ;       ///
[       adlt2$4        @       -0.019 ]       ;       ///
[       adlt2$5        @       0.364  ]       ;       ///
[       adlt2$6        @       0.791  ]       ;       ///
[       adlt2$7        @       1.252  ]       ;       ///
[       adlt2$8        @       1.789  ]       ;       ///
[       adlt3$1        @       -1.343 ]       ;       ///
[       adlt3$2        @       -0.984 ]       ;       ///
[       adlt3$3        @       -0.68  ]       ;       ///
[       adlt3$4        @       -0.27  ]       ;       ///
[       adlt3$5        @       0.06   ]       ;       ///
[       adlt3$6        @       0.445  ]       ;       ///
[       adlt3$7        @       0.835  ]       ;       ///
[       adlt3$8        @       1.345  ]       ;       ///
[       add$1  @       -0.685 ]       ;       ///
[       add$2  @       -0.399 ]       ;       ///
[       add$3  @       -0.217 ]       ;       ///
[       add$4  @       -0.019 ]       ;       ///
[       add$5  @       0.207  ]       ;       ///
[       add$6  @       0.355  ]       ;       ///
[       add$7  @       0.613  ]       ;       ///
[       add$8  @       0.934  ]       ;       ///
[       add$9  @       1.297  ]       ;       ///
[       ra1$1  @       -1.451 ]       ;       ///
```

```
[	ra1$2	@	-0.911 ]	;	///
[	ra1$3	@	-0.253 ]	;	///
[	ra1$4	@	0.398  ]	;	///
[	ra1$5	@	0.997  ]	;	///
[	ra1$6	@	1.517  ]	;	///
[	ra1$7	@	1.958  ]	;	///
[	ra2$1	@	-1.405 ]	;	///
[	ra2$2	@	-0.931 ]	;	///
[	ra2$3	@	-0.351 ]	;	///
[	ra2$4	@	0.159  ]	;	///
[	ra2$5	@	0.572  ]	;	///
[	ra2$6	@	0.974  ]	;	///
[	ra2$7	@	1.313  ]	;	///
[	ra2$8	@	1.716  ]	;	///
[	ra2$9	@	1.948  ]	;	///
[	ra3$1	@	-1.502 ]	;	///
[	ra3$2	@	-1.101 ]	;	///
[	ra3$3	@	-0.599 ]	;	///
[	ra3$4	@	0.24   ]	;	///
[	ra3$5	@	0.857  ]	;	///
[	ra3$6	@	1.176  ]	;	///
[	ra3$7	@	1.438  ]	;	///
[	ra3$8	@	1.712  ]	;	///
[	ra3$9	@	2.007  ]	;	///
[	ra4$1	@	-1.211 ]	;	///
[	ra4$2	@	-0.75  ]	;	///
[	ra4$3	@	0.086  ]	;	///
[	ra4$4	@	0.621  ]	;	///
[	ra4$5	@	0.884  ]	;	///
[	ra4$6	@	1.121  ]	;	///
[	ra4$7	@	1.444  ]	;	///
[	ra4$8	@	1.714  ]	;	///
[	ra4$9	@	2.045  ]	;	///
[	ra5$1	@	-1.257 ]	;	///
[	ra5$2	@	-0.826 ]	;	///
[	ra5$3	@	-0.393 ]	;	///
[	ra5$4	@	0.274  ]	;	///
[	ra5$5	@	0.728  ]	;	///
[	ra5$6	@	1.195  ]	;	///
[	ra5$7	@	1.463  ]	;	///
[	ra5$8	@	1.764  ]	;	///
[	ra5$9	@	2.105  ]	;	///
[	rab$1	@	-1.399 ]	;	///
[	rab$2	@	-0.857 ]	;	///
[	rab$3	@	-0.164 ]	;	///
[	rab$4	@	0.498  ]	;	///
[	rab$5	@	1.075  ]	;	///
[	rab$6	@	1.614  ]	;	///
[	rab$7	@	2.052  ]	;	///
[	balldl$1	@	-0.463 ]	;	///
[	flagdl$1	@	-0.024 ]	;	///
[	treedl$1	@	-0.139 ]	;	///
) ///
   savedata(save=fscores; file=trash.dat) savelog(trash)

preserve
infix idvis 191-198 mem 185-190 using trash.dat, clear
tempfile f1
save `f1'
restore
merge 1:1 idvis using `f1', keep(3)
```

```
keep idvis mem
capture save "mem2", replace

* verion 3
use temp, clear
keep if version==3
runmplus idvis lmrc-rarc, categorical(lmrc-rarc) id(idvis) ///
       model ( ///
mem @1.107; ///
mem    by     lmrc   @      0.837  ;      ///
mem    by     lmd    @      0.846  ;      ///
mem    by     radrc  @      0.876  ;      ///
mem    by     rarc   @      0.728  ;      ///
mem    by     ra6    @      0.85   ;      ///
mem    by     adrg1  @      0.559  ;      ///
mem    by     adrg2  @      0.473  ;      ///
mem    by     adlt1  @      0.765  ;      ///
mem    by     adlt2  @      0.839  ;      ///
mem    by     adlt3  @      0.847  ;      ///
mem    by     add    @      0.877  ;      ///
mem    by     ra1    @      0.661  ;      ///
mem    by     ra2    @      0.807  ;      ///
mem    by     ra3    @      0.852  ;      ///
mem    by     ra4    @      0.884  ;      ///
mem    by     ra5    @      0.882  ;      ///
mem    by     rab    @      0.615  ;      ///
mem    by     balldl @      0.748  ;      ///
mem    by     flagdl @      0.777  ;      ///
mem    by     treedl @      0.751  ;      ///
[      lmrc$1 @      -1.369 ]      ;      ///
[      lmrc$2 @      -0.841 ]      ;      ///
[      lmrc$3 @      -0.412 ]      ;      ///
[      lmrc$4 @      -0.079 ]      ;      ///
[      lmrc$5 @      0.225  ]      ;      ///
[      lmrc$6 @      0.638  ]      ;      ///
[      lmrc$7 @      0.946  ]      ;      ///
[      lmrc$8 @      1.296  ]      ;      ///
[      lmrc$9 @      1.702  ]      ;      ///
[      lmd$1  @      -0.637 ]      ;      ///
[      lmd$2  @      -0.289 ]      ;      ///
[      lmd$3  @      0.006  ]      ;      ///
[      lmd$4  @      0.458  ]      ;      ///
[      lmd$5  @      0.74   ]      ;      ///
[      lmd$6  @      0.854  ]      ;      ///
[      lmd$7  @      0.972  ]      ;      ///
[      lmd$8  @      1.267  ]      ;      ///
[      lmd$9  @      1.676  ]      ;      ///
[      radrc$1       @      -0.248 ]      ;      ///
[      radrc$2       @      0.109  ]      ;      ///
[      radrc$3       @      0.368  ]      ;      ///
[      radrc$4       @      0.668  ]      ;      ///
[      radrc$5       @      0.807  ]      ;      ///
[      radrc$6       @      0.97   ]      ;      ///
[      radrc$7       @      1.126  ]      ;      ///
[      radrc$8       @      1.453  ]      ;      ///
[      radrc$9       @      1.891  ]      ;      ///
[      rarc$1 @      -1.624 ]      ;      ///
[      rarc$2 @      -1.519 ]      ;      ///
[      rarc$3 @      -1.253 ]      ;      ///
[      rarc$4 @      -0.88  ]      ;      ///
[      rarc$5 @      -0.543 ]      ;      ///
```

```
[	rarc$6 @	-0.215 ]	;	///
[	rarc$7 @	0.166  ]	;	///
[	rarc$8 @	0.644  ]	;	///
[	rarc$9 @	1.053  ]	;	///
[	ra6$1  @	-0.824 ]	;	///
[	ra6$2  @	-0.242 ]	;	///
[	ra6$3  @	0.203  ]	;	///
[	ra6$4  @	0.526  ]	;	///
[	ra6$5  @	0.668  ]	;	///
[	ra6$6  @	0.859  ]	;	///
[	ra6$7  @	1.036  ]	;	///
[	ra6$8  @	1.424  ]	;	///
[	ra6$9  @	1.835  ]	;	///
[	adrg1$1		@	-1.545 ]	;	///
[	adrg1$2		@	-1.421 ]	;	///
[	adrg1$3		@	-1.262 ]	;	///
[	adrg1$4		@	-1.019 ]	;	///
[	adrg1$5		@	-0.81  ]	;	///
[	adrg1$6		@	-0.562 ]	;	///
[	adrg1$7		@	-0.167 ]	;	///
[	adrg1$8		@	0.265  ]	;	///
[	adrg1$9		@	0.795  ]	;	///
[	adrg2$1		@	-1.697 ]	;	///
[	adrg2$2		@	-1.333 ]	;	///
[	adrg2$3		@	-1.142 ]	;	///
[	adrg2$4		@	-0.931 ]	;	///
[	adrg2$5		@	-0.662 ]	;	///
[	adrg2$6		@	-0.331 ]	;	///
[	adrg2$7		@	0.214  ]	;	///
[	adlt1$1		@	-1.464 ]	;	///
[	adlt1$2		@	-0.854 ]	;	///
[	adlt1$3		@	-0.25  ]	;	///
[	adlt1$4		@	0.242  ]	;	///
[	adlt1$5		@	0.786  ]	;	///
[	adlt1$6		@	1.214  ]	;	///
[	adlt1$7		@	1.631  ]	;	///
[	adlt2$1		@	-1.417 ]	;	///
[	adlt2$2		@	-1.078 ]	;	///
[	adlt2$3		@	-0.577 ]	;	///
[	adlt2$4		@	-0.094 ]	;	///
[	adlt2$5		@	0.291  ]	;	///
[	adlt2$6		@	0.753  ]	;	///
[	adlt2$7		@	1.25   ]	;	///
[	adlt2$8		@	1.762  ]	;	///
[	adlt3$1		@	-1.621 ]	;	///
[	adlt3$2		@	-1.208 ]	;	///
[	adlt3$3		@	-0.779 ]	;	///
[	adlt3$4		@	-0.381 ]	;	///
[	adlt3$5		@	0.098  ]	;	///
[	adlt3$6		@	0.516  ]	;	///
[	adlt3$7		@	0.991  ]	;	///
[	adlt3$8		@	1.533  ]	;	///
[	add$1  @	-0.895 ]	;	///
[	add$2  @	-0.521 ]	;	///
[	add$3  @	-0.248 ]	;	///
[	add$4  @	-0.002 ]	;	///
[	add$5  @	0.206  ]	;	///
[	add$6  @	0.408  ]	;	///
[	add$7  @	0.668  ]	;	///
[	add$8  @	0.962  ]	;	///
[	add$9  @	1.305  ]	;	///
```

```
[      ra1$1  @      -1.609 ]      ;      ///
[      ra1$2  @      -1.049 ]      ;      ///
[      ra1$3  @      -0.413 ]      ;      ///
[      ra1$4  @      0.225  ]      ;      ///
[      ra1$5  @      0.795  ]      ;      ///
[      ra1$6  @      1.319  ]      ;      ///
[      ra1$7  @      1.776  ]      ;      ///
[      ra2$1  @      -1.46  ]      ;      ///
[      ra2$2  @      -0.966 ]      ;      ///
[      ra2$3  @      -0.505 ]      ;      ///
[      ra2$4  @      -0.034 ]      ;      ///
[      ra2$5  @      0.401  ]      ;      ///
[      ra2$6  @      0.769  ]      ;      ///
[      ra2$7  @      1.137  ]      ;      ///
[      ra2$8  @      1.436  ]      ;      ///
[      ra2$9  @      1.713  ]      ;      ///
[      ra3$1  @      -1.567 ]      ;      ///
[      ra3$2  @      -1.172 ]      ;      ///
[      ra3$3  @      -0.755 ]      ;      ///
[      ra3$4  @      0.048  ]      ;      ///
[      ra3$5  @      0.677  ]      ;      ///
[      ra3$6  @      0.917  ]      ;      ///
[      ra3$7  @      1.195  ]      ;      ///
[      ra3$8  @      1.439  ]      ;      ///
[      ra3$9  @      1.67   ]      ;      ///
[      ra4$1  @      -1.261 ]      ;      ///
[      ra4$2  @      -0.84  ]      ;      ///
[      ra4$3  @      -0.122 ]      ;      ///
[      ra4$4  @      0.42   ]      ;      ///
[      ra4$5  @      0.649  ]      ;      ///
[      ra4$6  @      0.884  ]      ;      ///
[      ra4$7  @      1.124  ]      ;      ///
[      ra4$8  @      1.339  ]      ;      ///
[      ra4$9  @      1.663  ]      ;      ///
[      ra5$1  @      -1.348 ]      ;      ///
[      ra5$2  @      -0.94  ]      ;      ///
[      ra5$3  @      -0.541 ]      ;      ///
[      ra5$4  @      0.045  ]      ;      ///
[      ra5$5  @      0.472  ]      ;      ///
[      ra5$6  @      0.912  ]      ;      ///
[      ra5$7  @      1.149  ]      ;      ///
[      ra5$8  @      1.402  ]      ;      ///
[      ra5$9  @      1.746  ]      ;      ///
[      rab$1  @      -1.3   ]      ;      ///
[      rab$2  @      -0.675 ]      ;      ///
[      rab$3  @      -0.063 ]      ;      ///
[      rab$4  @      0.549  ]      ;      ///
[      rab$5  @      1.075  ]      ;      ///
[      rab$6  @      1.508  ]      ;      ///
[      rab$7  @      1.868  ]      ;      ///
[      balldl$1      @      -0.509 ]      ;      ///
[      flagdl$1      @      -0.033 ]      ;      ///
[      treedl$1      @      -0.127 ]      ;      ///
) ///
   savedata(save=fscores; file=trash.dat) savelog(trash)

preserve
infix idvis 212-288 mem 203-210 using trash.dat, clear
tempfile f1
save `f1'
restore
```

```
merge 1:1 idvis using `f1', keep(3)
keep idvis mem
capture save "mem3", replace

* version4
use temp, clear
keep if version==4
runmplus idvis radrc-rarc, categorical(radrc-rarc) id(idvis) ///
       model ( ///
mem @0.962; ///
mem    by     radrc  @      0.885  ;      ///
mem    by     rarc   @      0.716  ;      ///
mem    by     ra6    @      0.862  ;      ///
mem    by     adrg1  @      0.43   ;      ///
mem    by     adrg2  @      0.507  ;      ///
mem    by     adlt1  @      0.792  ;      ///
mem    by     adlt2  @      0.858  ;      ///
mem    by     adlt3  @      0.844  ;      ///
mem    by     add    @      0.898  ;      ///
mem    by     ra1    @      0.649  ;      ///
mem    by     ra2    @      0.826  ;      ///
mem    by     ra3    @      0.876  ;      ///
mem    by     ra4    @      0.884  ;      ///
mem    by     ra5    @      0.877  ;      ///
mem    by     rab    @      0.582  ;      ///
mem    by     balldl @      0.748  ;      ///
mem    by     flagdl @      0.777  ;      ///
mem    by     treedl @      0.751  ;      ///
[      radrc$1       @      -0.196 ]      ;      ///
[      radrc$2       @      0.25   ]      ;      ///
[      radrc$3       @      0.553  ]      ;      ///
[      radrc$4       @      0.875  ]      ;      ///
[      radrc$5       @      1.044  ]      ;      ///
[      radrc$6       @      1.235  ]      ;      ///
[      radrc$7       @      1.443  ]      ;      ///
[      radrc$8       @      1.807  ]      ;      ///
[      radrc$9       @      2.245  ]      ;      ///
[      rarc$1 @      -1.702 ]      ;      ///
[      rarc$2 @      -1.576 ]      ;      ///
[      rarc$3 @      -1.192 ]      ;      ///
[      rarc$4 @      -0.841 ]      ;      ///
[      rarc$5 @      -0.5   ]      ;      ///
[      rarc$6 @      -0.118 ]      ;      ///
[      rarc$7 @      0.257  ]      ;      ///
[      rarc$8 @      0.816  ]      ;      ///
[      rarc$9 @      1.203  ]      ;      ///
[      ra6$1  @      -0.697 ]      ;      ///
[      ra6$2  @      -0.062 ]      ;      ///
[      ra6$3  @      0.402  ]      ;      ///
[      ra6$4  @      0.732  ]      ;      ///
[      ra6$5  @      0.927  ]      ;      ///
[      ra6$6  @      1.087  ]      ;      ///
[      ra6$7  @      1.243  ]      ;      ///
[      ra6$8  @      1.688  ]      ;      ///
[      ra6$9  @      2.26   ]      ;      ///
[      adrg1$1       @      -1.949 ]      ;      ///
[      adrg1$2       @      -1.632 ]      ;      ///
[      adrg1$3       @      -1.392 ]      ;      ///
[      adrg1$4       @      -1.111 ]      ;      ///
[      adrg1$5       @      -0.844 ]      ;      ///
[      adrg1$6       @      -0.436 ]      ;      ///
```

```
[      adrg1$7      @      -0.04  ]      ;      ///
[      adrg1$8      @      0.453  ]      ;      ///
[      adrg1$9      @      1.009  ]      ;      ///
[      adrg2$1      @      -1.791 ]      ;      ///
[      adrg2$2      @      -1.393 ]      ;      ///
[      adrg2$3      @      -1.234 ]      ;      ///
[      adrg2$4      @      -0.995 ]      ;      ///
[      adrg2$5      @      -0.72  ]      ;      ///
[      adrg2$6      @      -0.335 ]      ;      ///
[      adrg2$7      @      0.189  ]      ;      ///
[      adlt1$1      @      -1.507 ]      ;      ///
[      adlt1$2      @      -0.872 ]      ;      ///
[      adlt1$3      @      -0.371 ]      ;      ///
[      adlt1$4      @      0.177  ]      ;      ///
[      adlt1$5      @      0.693  ]      ;      ///
[      adlt1$6      @      1.346  ]      ;      ///
[      adlt1$7      @      1.791  ]      ;      ///
[      adlt2$1      @      -1.61  ]      ;      ///
[      adlt2$2      @      -1.208 ]      ;      ///
[      adlt2$3      @      -0.709 ]      ;      ///
[      adlt2$4      @      -0.217 ]      ;      ///
[      adlt2$5      @      0.319  ]      ;      ///
[      adlt2$6      @      0.736  ]      ;      ///
[      adlt2$7      @      1.288  ]      ;      ///
[      adlt2$8      @      1.97   ]      ;      ///
[      adlt3$1      @      -1.91  ]      ;      ///
[      adlt3$2      @      -1.443 ]      ;      ///
[      adlt3$3      @      -1.02  ]      ;      ///
[      adlt3$4      @      -0.566 ]      ;      ///
[      adlt3$5      @      -0.026 ]      ;      ///
[      adlt3$6      @      0.411  ]      ;      ///
[      adlt3$7      @      0.927  ]      ;      ///
[      adlt3$8      @      1.633  ]      ;      ///
[      add$1  @      -1.147 ]      ;      ///
[      add$2  @      -0.744 ]      ;      ///
[      add$3  @      -0.444 ]      ;      ///
[      add$4  @      -0.14  ]      ;      ///
[      add$5  @      0.115  ]      ;      ///
[      add$6  @      0.397  ]      ;      ///
[      add$7  @      0.674  ]      ;      ///
[      add$8  @      0.975  ]      ;      ///
[      add$9  @      1.461  ]      ;      ///
[      ra1$1  @      -1.451 ]      ;      ///
[      ra1$2  @      -0.911 ]      ;      ///
[      ra1$3  @      -0.253 ]      ;      ///
[      ra1$4  @      0.398  ]      ;      ///
[      ra1$5  @      0.997  ]      ;      ///
[      ra1$6  @      1.517  ]      ;      ///
[      ra1$7  @      1.958  ]      ;      ///
[      ra2$1  @      -1.405 ]      ;      ///
[      ra2$2  @      -0.931 ]      ;      ///
[      ra2$3  @      -0.351 ]      ;      ///
[      ra2$4  @      0.159  ]      ;      ///
[      ra2$5  @      0.572  ]      ;      ///
[      ra2$6  @      0.974  ]      ;      ///
[      ra2$7  @      1.313  ]      ;      ///
[      ra2$8  @      1.716  ]      ;      ///
[      ra2$9  @      1.948  ]      ;      ///
[      ra3$1  @      -1.502 ]      ;      ///
[      ra3$2  @      -1.101 ]      ;      ///
[      ra3$3  @      -0.599 ]      ;      ///
```

```
[       ra3$4  @      0.24   ]      ;      ///
[       ra3$5  @      0.857  ]      ;      ///
[       ra3$6  @      1.176  ]      ;      ///
[       ra3$7  @      1.438  ]      ;      ///
[       ra3$8  @      1.712  ]      ;      ///
[       ra3$9  @      2.007  ]      ;      ///
[       ra4$1  @      -1.211 ]      ;      ///
[       ra4$2  @      -0.75  ]      ;      ///
[       ra4$3  @      0.086  ]      ;      ///
[       ra4$4  @      0.621  ]      ;      ///
[       ra4$5  @      0.884  ]      ;      ///
[       ra4$6  @      1.121  ]      ;      ///
[       ra4$7  @      1.444  ]      ;      ///
[       ra4$8  @      1.714  ]      ;      ///
[       ra4$9  @      2.045  ]      ;      ///
[       ra5$1  @      -1.257 ]      ;      ///
[       ra5$2  @      -0.826 ]      ;      ///
[       ra5$3  @      -0.393 ]      ;      ///
[       ra5$4  @      0.274  ]      ;      ///
[       ra5$5  @      0.728  ]      ;      ///
[       ra5$6  @      1.195  ]      ;      ///
[       ra5$7  @      1.463  ]      ;      ///
[       ra5$8  @      1.764  ]      ;      ///
[       ra5$9  @      2.105  ]      ;      ///
[       rab$1  @      -1.399 ]      ;      ///
[       rab$2  @      -0.857 ]      ;      ///
[       rab$3  @      -0.164 ]      ;      ///
[       rab$4  @      0.498  ]      ;      ///
[       rab$5  @      1.075  ]      ;      ///
[       rab$6  @      1.614  ]      ;      ///
[       rab$7  @      2.052  ]      ;      ///
[       balldl$1      @      -0.632 ]      ;      ///
[       flagdl$1      @      -0.153 ]      ;      ///
[       treedl$1      @      -0.269 ]      ;      ///
) ///
   savedata(save=fscores; file=trash.dat) savelog(trash)

preserve
infix idvis 191-198 mem 185-190 using trash.dat, clear
tempfile f1
save `f1'
restore
merge 1:1 idvis using `f1', keep(3)
keep idvis mem
capture save "mem4", replace

* version 5
use temp, clear
keep if version==5
runmplus idvis lmrc-rarc, categorical(lmrc-rarc) id(idvis) ///
       model ( ///
mem @1.179; ///
mem    by     lmrc   @      0.837  ;      ///
mem    by     lmd    @      0.846  ;      ///
mem    by     radrc  @      0.876  ;      ///
mem    by     rarc   @      0.728  ;      ///
mem    by     ra6    @      0.85   ;      ///
mem    by     adrg1  @      0.478  ;      ///
mem    by     adrg2  @      0.533  ;      ///
mem    by     adlt1  @      0.79   ;      ///
mem    by     adlt2  @      0.828  ;      ///
```

```
mem     by      adlt3  @       0.835  ;       ///
mem     by      add    @       0.862  ;       ///
mem     by      ra1    @       0.661  ;       ///
mem     by      ra2    @       0.807  ;       ///
mem     by      ra3    @       0.852  ;       ///
mem     by      ra4    @       0.884  ;       ///
mem     by      ra5    @       0.882  ;       ///
mem     by      rab    @       0.615  ;       ///
mem     by      balldl @       0.748  ;       ///
mem     by      flagdl @       0.777  ;       ///
mem     by      treedl @       0.751  ;       ///
[       lmrc$1 @       -1.369 ]       ;       ///
[       lmrc$2 @       -0.841 ]       ;       ///
[       lmrc$3 @       -0.412 ]       ;       ///
[       lmrc$4 @       -0.079 ]       ;       ///
[       lmrc$5 @       0.225  ]       ;       ///
[       lmrc$6 @       0.638  ]       ;       ///
[       lmrc$7 @       0.946  ]       ;       ///
[       lmrc$8 @       1.296  ]       ;       ///
[       lmrc$9 @       1.702  ]       ;       ///
[       lmd$1  @       -0.637 ]       ;       ///
[       lmd$2  @       -0.289 ]       ;       ///
[       lmd$3  @       0.006  ]       ;       ///
[       lmd$4  @       0.458  ]       ;       ///
[       lmd$5  @       0.74   ]       ;       ///
[       lmd$6  @       0.854  ]       ;       ///
[       lmd$7  @       0.972  ]       ;       ///
[       lmd$8  @       1.267  ]       ;       ///
[       lmd$9  @       1.676  ]       ;       ///
[       radrc$1        @       -0.248 ]       ;       ///
[       radrc$2        @       0.109  ]       ;       ///
[       radrc$3        @       0.368  ]       ;       ///
[       radrc$4        @       0.668  ]       ;       ///
[       radrc$5        @       0.807  ]       ;       ///
[       radrc$6        @       0.97   ]       ;       ///
[       radrc$7        @       1.126  ]       ;       ///
[       radrc$8        @       1.453  ]       ;       ///
[       radrc$9        @       1.891  ]       ;       ///
[       rarc$1 @       -1.624 ]       ;       ///
[       rarc$2 @       -1.519 ]       ;       ///
[       rarc$3 @       -1.253 ]       ;       ///
[       rarc$4 @       -0.88  ]       ;       ///
[       rarc$5 @       -0.543 ]       ;       ///
[       rarc$6 @       -0.215 ]       ;       ///
[       rarc$7 @       0.166  ]       ;       ///
[       rarc$8 @       0.644  ]       ;       ///
[       rarc$9 @       1.053  ]       ;       ///
[       ra6$1  @       -0.824 ]       ;       ///
[       ra6$2  @       -0.242 ]       ;       ///
[       ra6$3  @       0.203  ]       ;       ///
[       ra6$4  @       0.526  ]       ;       ///
[       ra6$5  @       0.668  ]       ;       ///
[       ra6$6  @       0.859  ]       ;       ///
[       ra6$7  @       1.036  ]       ;       ///
[       ra6$8  @       1.424  ]       ;       ///
[       ra6$9  @       1.835  ]       ;       ///
[       adrg1$1        @       -1.521 ]       ;       ///
[       adrg1$2        @       -1.272 ]       ;       ///
[       adrg1$3        @       -1.104 ]       ;       ///
[       adrg1$4        @       -0.856 ]       ;       ///
[       adrg1$5        @       -0.603 ]       ;       ///
```

```
[       adrg1$6      @      -0.35  ]      ;      ///
[       adrg1$7      @      -0.023 ]      ;      ///
[       adrg1$8      @      0.395  ]      ;      ///
[       adrg1$9      @      0.998  ]      ;      ///
[       adrg2$1      @      -1.574 ]      ;      ///
[       adrg2$2      @      -1.272 ]      ;      ///
[       adrg2$3      @      -1.083 ]      ;      ///
[       adrg2$4      @      -0.879 ]      ;      ///
[       adrg2$5      @      -0.585 ]      ;      ///
[       adrg2$6      @      -0.256 ]      ;      ///
[       adrg2$7      @      0.179  ]      ;      ///
[       adlt1$1      @      -1.111 ]      ;      ///
[       adlt1$2      @      -0.719 ]      ;      ///
[       adlt1$3      @      -0.253 ]      ;      ///
[       adlt1$4      @      0.24   ]      ;      ///
[       adlt1$5      @      0.662  ]      ;      ///
[       adlt1$6      @      1.131  ]      ;      ///
[       adlt1$7      @      1.681  ]      ;      ///
[       adlt2$1      @      -1.187 ]      ;      ///
[       adlt2$2      @      -0.814 ]      ;      ///
[       adlt2$3      @      -0.45  ]      ;      ///
[       adlt2$4      @      -0.019 ]      ;      ///
[       adlt2$5      @      0.364  ]      ;      ///
[       adlt2$6      @      0.791  ]      ;      ///
[       adlt2$7      @      1.252  ]      ;      ///
[       adlt2$8      @      1.789  ]      ;      ///
[       adlt3$1      @      -1.343 ]      ;      ///
[       adlt3$2      @      -0.984 ]      ;      ///
[       adlt3$3      @      -0.68  ]      ;      ///
[       adlt3$4      @      -0.27  ]      ;      ///
[       adlt3$5      @      0.06   ]      ;      ///
[       adlt3$6      @      0.445  ]      ;      ///
[       adlt3$7      @      0.835  ]      ;      ///
[       adlt3$8      @      1.345  ]      ;      ///
[       add$1  @      -0.685 ]      ;      ///
[       add$2  @      -0.399 ]      ;      ///
[       add$3  @      -0.217 ]      ;      ///
[       add$4  @      -0.019 ]      ;      ///
[       add$5  @      0.207  ]      ;      ///
[       add$6  @      0.355  ]      ;      ///
[       add$7  @      0.613  ]      ;      ///
[       add$8  @      0.934  ]      ;      ///
[       add$9  @      1.297  ]      ;      ///
[       ra1$1  @      -1.609 ]      ;      ///
[       ra1$2  @      -1.049 ]      ;      ///
[       ra1$3  @      -0.413 ]      ;      ///
[       ra1$4  @      0.225  ]      ;      ///
[       ra1$5  @      0.795  ]      ;      ///
[       ra1$6  @      1.319  ]      ;      ///
[       ra1$7  @      1.776  ]      ;      ///
[       ra2$1  @      -1.46  ]      ;      ///
[       ra2$2  @      -0.966 ]      ;      ///
[       ra2$3  @      -0.505 ]      ;      ///
[       ra2$4  @      -0.034 ]      ;      ///
[       ra2$5  @      0.401  ]      ;      ///
[       ra2$6  @      0.769  ]      ;      ///
[       ra2$7  @      1.137  ]      ;      ///
[       ra2$8  @      1.436  ]      ;      ///
[       ra2$9  @      1.713  ]      ;      ///
[       ra3$1  @      -1.567 ]      ;      ///
[       ra3$2  @      -1.172 ]      ;      ///
```

```
[      ra3$3  @      -0.755 ]      ;      ///
[      ra3$4  @      0.048  ]      ;      ///
[      ra3$5  @      0.677  ]      ;      ///
[      ra3$6  @      0.917  ]      ;      ///
[      ra3$7  @      1.195  ]      ;      ///
[      ra3$8  @      1.439  ]      ;      ///
[      ra3$9  @      1.67   ]      ;      ///
[      ra4$1  @      -1.261 ]      ;      ///
[      ra4$2  @      -0.84  ]      ;      ///
[      ra4$3  @      -0.122 ]      ;      ///
[      ra4$4  @      0.42   ]      ;      ///
[      ra4$5  @      0.649  ]      ;      ///
[      ra4$6  @      0.884  ]      ;      ///
[      ra4$7  @      1.124  ]      ;      ///
[      ra4$8  @      1.339  ]      ;      ///
[      ra4$9  @      1.663  ]      ;      ///
[      ra5$1  @      -1.348 ]      ;      ///
[      ra5$2  @      -0.94  ]      ;      ///
[      ra5$3  @      -0.541 ]      ;      ///
[      ra5$4  @      0.045  ]      ;      ///
[      ra5$5  @      0.472  ]      ;      ///
[      ra5$6  @      0.912  ]      ;      ///
[      ra5$7  @      1.149  ]      ;      ///
[      ra5$8  @      1.402  ]      ;      ///
[      ra5$9  @      1.746  ]      ;      ///
[      rab$1  @      -1.3   ]      ;      ///
[      rab$2  @      -0.675 ]      ;      ///
[      rab$3  @      -0.063 ]      ;      ///
[      rab$4  @      0.549  ]      ;      ///
[      rab$5  @      1.075  ]      ;      ///
[      rab$6  @      1.508  ]      ;      ///
[      rab$7  @      1.868  ]      ;      ///
[      balldl$1      @      -0.509 ]      ;      ///
[      flagdl$1      @      -0.033 ]      ;      ///
[      treedl$1      @      -0.127 ]      ;      ///
) ///
   savedata(save=fscores; file=trash.dat) savelog(trash)

preserve
infix idvis 212-288 mem 203-210 using trash.dat, clear
tempfile f1
save `f1'
restore
merge 1:1 idvis using `f1', keep(3)
keep idvis mem
capture save "mem5", replace

* version6
use temp, clear
keep if version==6
runmplus idvis lmrc-rarc, categorical(lmrc-rarc) id(idvis) ///
       model ( ///
mem @1.137; ///
mem    by     lmrc   @      0.837  ;      ///
mem    by     lmd    @      0.846  ;      ///
mem    by     radrc  @      0.885  ;      ///
mem    by     rarc   @      0.716  ;      ///
mem    by     ra6    @      0.862  ;      ///
mem    by     adrg1  @      0.559  ;      ///
mem    by     adrg2  @      0.473  ;      ///
mem    by     adlt1  @      0.765  ;      ///
```

```
mem     by      adlt2  @       0.839  ;       ///
mem     by      adlt3  @       0.847  ;       ///
mem     by      add    @       0.877  ;       ///
mem     by      ra1    @       0.649  ;       ///
mem     by      ra2    @       0.826  ;       ///
mem     by      ra3    @       0.876  ;       ///
mem     by      ra4    @       0.884  ;       ///
mem     by      ra5    @       0.877  ;       ///
mem     by      rab    @       0.582  ;       ///
mem     by      balldl @       0.748  ;       ///
mem     by      flagdl @       0.777  ;       ///
mem     by      treedl @       0.751  ;       ///
[       lmrc$1 @       -1.369 ]       ;       ///
[       lmrc$2 @       -0.841 ]       ;       ///
[       lmrc$3 @       -0.412 ]       ;       ///
[       lmrc$4 @       -0.079 ]       ;       ///
[       lmrc$5 @       0.225  ]       ;       ///
[       lmrc$6 @       0.638  ]       ;       ///
[       lmrc$7 @       0.946  ]       ;       ///
[       lmrc$8 @       1.296  ]       ;       ///
[       lmrc$9 @       1.702  ]       ;       ///
[       lmd$1  @       -0.637 ]       ;       ///
[       lmd$2  @       -0.289 ]       ;       ///
[       lmd$3  @       0.006  ]       ;       ///
[       lmd$4  @       0.458  ]       ;       ///
[       lmd$5  @       0.74   ]       ;       ///
[       lmd$6  @       0.854  ]       ;       ///
[       lmd$7  @       0.972  ]       ;       ///
[       lmd$8  @       1.267  ]       ;       ///
[       lmd$9  @       1.676  ]       ;       ///
[       radrc$1        @       -0.196 ]       ;       ///
[       radrc$2        @       0.25   ]       ;       ///
[       radrc$3        @       0.553  ]       ;       ///
[       radrc$4        @       0.875  ]       ;       ///
[       radrc$5        @       1.044  ]       ;       ///
[       radrc$6        @       1.235  ]       ;       ///
[       radrc$7        @       1.443  ]       ;       ///
[       radrc$8        @       1.807  ]       ;       ///
[       radrc$9        @       2.245  ]       ;       ///
[       rarc$1 @       -1.702 ]       ;       ///
[       rarc$2 @       -1.576 ]       ;       ///
[       rarc$3 @       -1.192 ]       ;       ///
[       rarc$4 @       -0.841 ]       ;       ///
[       rarc$5 @       -0.5   ]       ;       ///
[       rarc$6 @       -0.118 ]       ;       ///
[       rarc$7 @       0.257  ]       ;       ///
[       rarc$8 @       0.816  ]       ;       ///
[       rarc$9 @       1.203  ]       ;       ///
[       ra6$1  @       -0.697 ]       ;       ///
[       ra6$2  @       -0.062 ]       ;       ///
[       ra6$3  @       0.402  ]       ;       ///
[       ra6$4  @       0.732  ]       ;       ///
[       ra6$5  @       0.927  ]       ;       ///
[       ra6$6  @       1.087  ]       ;       ///
[       ra6$7  @       1.243  ]       ;       ///
[       ra6$8  @       1.688  ]       ;       ///
[       ra6$9  @       2.26   ]       ;       ///
[       adrg1$1        @       -1.573 ]       ;       ///
[       adrg1$2        @       -1.349 ]       ;       ///
[       adrg1$3        @       -1.168 ]       ;       ///
[       adrg1$4        @       -1.023 ]       ;       ///
```

```
[       adrg1$5       @       -0.791 ]       ;       ///
[       adrg1$6       @       -0.459 ]       ;       ///
[       adrg1$7       @       -0.089 ]       ;       ///
[       adrg1$8       @       0.347  ]       ;       ///
[       adrg1$9       @       0.96   ]       ;       ///
[       adrg2$1       @       -2.114 ]       ;       ///
[       adrg2$2       @       -1.548 ]       ;       ///
[       adrg2$3       @       -1.195 ]       ;       ///
[       adrg2$4       @       -0.958 ]       ;       ///
[       adrg2$5       @       -0.736 ]       ;       ///
[       adrg2$6       @       -0.328 ]       ;       ///
[       adrg2$7       @       0.251  ]       ;       ///
[       adlt1$1       @       -1.201 ]       ;       ///
[       adlt1$2       @       -0.8   ]       ;       ///
[       adlt1$3       @       -0.214 ]       ;       ///
[       adlt1$4       @       0.285  ]       ;       ///
[       adlt1$5       @       0.767  ]       ;       ///
[       adlt1$6       @       1.321  ]       ;       ///
[       adlt1$7       @       1.808  ]       ;       ///
[       adlt2$1       @       -1.228 ]       ;       ///
[       adlt2$2       @       -0.859 ]       ;       ///
[       adlt2$3       @       -0.429 ]       ;       ///
[       adlt2$4       @       -0.09  ]       ;       ///
[       adlt2$5       @       0.32   ]       ;       ///
[       adlt2$6       @       0.72   ]       ;       ///
[       adlt2$7       @       1.221  ]       ;       ///
[       adlt2$8       @       1.893  ]       ;       ///
[       adlt3$1       @       -1.299 ]       ;       ///
[       adlt3$2       @       -1.011 ]       ;       ///
[       adlt3$3       @       -0.659 ]       ;       ///
[       adlt3$4       @       -0.277 ]       ;       ///
[       adlt3$5       @       0.112  ]       ;       ///
[       adlt3$6       @       0.513  ]       ;       ///
[       adlt3$7       @       0.986  ]       ;       ///
[       adlt3$8       @       1.592  ]       ;       ///
[       add$1  @       -0.704 ]       ;       ///
[       add$2  @       -0.379 ]       ;       ///
[       add$3  @       -0.157 ]       ;       ///
[       add$4  @       -0.014 ]       ;       ///
[       add$5  @       0.145  ]       ;       ///
[       add$6  @       0.376  ]       ;       ///
[       add$7  @       0.612  ]       ;       ///
[       add$8  @       1.049  ]       ;       ///
[       add$9  @       1.499  ]       ;       ///
[       ra1$1  @       -1.451 ]       ;       ///
[       ra1$2  @       -0.911 ]       ;       ///
[       ra1$3  @       -0.253 ]       ;       ///
[       ra1$4  @       0.398  ]       ;       ///
[       ra1$5  @       0.997  ]       ;       ///
[       ra1$6  @       1.517  ]       ;       ///
[       ra1$7  @       1.958  ]       ;       ///
[       ra2$1  @       -1.405 ]       ;       ///
[       ra2$2  @       -0.931 ]       ;       ///
[       ra2$3  @       -0.351 ]       ;       ///
[       ra2$4  @       0.159  ]       ;       ///
[       ra2$5  @       0.572  ]       ;       ///
[       ra2$6  @       0.974  ]       ;       ///
[       ra2$7  @       1.313  ]       ;       ///
[       ra2$8  @       1.716  ]       ;       ///
[       ra2$9  @       1.948  ]       ;       ///
[       ra3$1  @       -1.502 ]       ;       ///
```

```
[       ra3$2  @       -1.101 ]       ;       ///
[       ra3$3  @       -0.599 ]       ;       ///
[       ra3$4  @       0.24   ]       ;       ///
[       ra3$5  @       0.857  ]       ;       ///
[       ra3$6  @       1.176  ]       ;       ///
[       ra3$7  @       1.438  ]       ;       ///
[       ra3$8  @       1.712  ]       ;       ///
[       ra3$9  @       2.007  ]       ;       ///
[       ra4$1  @       -1.211 ]       ;       ///
[       ra4$2  @       -0.75  ]       ;       ///
[       ra4$3  @       0.086  ]       ;       ///
[       ra4$4  @       0.621  ]       ;       ///
[       ra4$5  @       0.884  ]       ;       ///
[       ra4$6  @       1.121  ]       ;       ///
[       ra4$7  @       1.444  ]       ;       ///
[       ra4$8  @       1.714  ]       ;       ///
[       ra4$9  @       2.045  ]       ;       ///
[       ra5$1  @       -1.257 ]       ;       ///
[       ra5$2  @       -0.826 ]       ;       ///
[       ra5$3  @       -0.393 ]       ;       ///
[       ra5$4  @       0.274  ]       ;       ///
[       ra5$5  @       0.728  ]       ;       ///
[       ra5$6  @       1.195  ]       ;       ///
[       ra5$7  @       1.463  ]       ;       ///
[       ra5$8  @       1.764  ]       ;       ///
[       ra5$9  @       2.105  ]       ;       ///
[       rab$1  @       -1.399 ]       ;       ///
[       rab$2  @       -0.857 ]       ;       ///
[       rab$3  @       -0.164 ]       ;       ///
[       rab$4  @       0.498  ]       ;       ///
[       rab$5  @       1.075  ]       ;       ///
[       rab$6  @       1.614  ]       ;       ///
[       rab$7  @       2.052  ]       ;       ///
[       balldl$1       @       -0.408 ]       ;       ///
[       flagdl$1       @       0.028  ]       ;       ///
[       treedl$1       @       -0.143 ]       ;       ///
) ///
   savedata(save=fscores; file=trash.dat) savelog(trash)

preserve
infix idvis 212-288 mem 203-210 using trash.dat, clear
tempfile f1
save `f1'
restore
merge 1:1 idvis using `f1', keep(3)
keep idvis mem
capture save "mem6", replace

! erase __*.dct
! erase trash.*
! erase temp.dta

forvalues i=1/5 {
       merge 1:1 idvis using mem`i', nogen update replace
       }

rename mem adni_mem
replace adni_mem=. if rid==4526 & visnum==24 // factor score not possible
la var adni_mem "Memory summary score"
```

ADNI-Lan and ADNI-VS:

Composite measures of language and visuospatial functioning: ADNI-Lan and ADNI-VS

Seo-Eun Choi, PhD[1], Shubhabrata Mukherjee, PhD[1], Laura E. Gibbons, PhD[1], R. Elizabeth Sanders, BA[1], Richard N. Jones, ScD[2], Douglas Tommett, MS[2], Jesse Mez, MD MS[3], Emily Trittschuh, PhD[4], Andrew Saykin, PsychD[5], Melissa Lamar, PhD[6], Laura Rabin, PhD[7], Nancy Foldi, PhD[8], Sietske Sikkes, PhD[9], and Paul K. Crane, MD MPH[1], for the Alzheimer's Disease Neuroimaging Initiative[*]

1. University of Washington, Box 359780, Harborview Medical Center, 325 Ninth Avenue, Seattle, WA 98104
2. Brown University, Psychiatry and Human Behavior, Providence, RI 02912
3. Department of Neurology, School of Medicine, Boston University, Boston, MA 02118
4. Department of Psychiatry and Behavioral Sciences, School of Medicine, University of Washington, Seattle, WA 98195
5. Department of Radiology, Indiana University, Indianapolis, IN 47405
6. Division of Behavioral Science, Rush Medical College, Chicago, IL 60612
7. Department of Psychology, Brooklyn College, Brooklyn, NY 11210
8. Department of Psychology, Queens College, Flushing, NY 11367
9. Department of Epidemiology & Biostatistics, Amsterdam UMC, Vrije Universiteit Amsterdam, Amsterdam, Netherlands

Summary

We derived composite scores for language (ADNI-Lan) and visuospatial functioning (ADNI-VS) using data from the ADNI neuropsychological battery with item response theory (IRT) methods. ADNI-Lan and ADNI-VS have been validated in a paper that is being submitted for publication. The methods below are adapted from that paper.

Method

We used baseline and follow-up data to develop ADNI-Lan and ADNI-VS. Several of the authors (PKC, AC, and ET) reviewed the neuropsychological battery to identify items which could be considered indicators of language or visuospatial functioning (VS). We refined our item

---

[*] Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

selection using an iterative process in which we constructed a model using confirmatory factor analysis. We used R (version 3.6.1) and Mplus (version 8.3).

For WLSMV models, our criteria for model fit were the confirmatory fit index (CFI), the Tucker Lewis Index (TLI) and the root mean squared error of approximation (RMSEA), where criteria for excellent fit include CFI > 0.95, TLI > 0.95, and RMSEA < 0.05 [5].

ADNI-Lan

The ADNI neuropsychological battery contained three language-related tests: 1. Category Fluency – Animals; 2 Category Fluency – Vegetables; 3. Boston Naming (Total).

The Mini Mental State Exam (MMSE) included eight language tasks: 1. Naming an object – Watch; 2. Naming an object – Pencil; 3. Repeating a sentence; 4. Reading a sentence; 5. Writing a sentence; 6.-8. Following a Series of Instructions. Each task was dichotomous.

The ADAS-Cog included three different language tasks: 1. Following commands; 2. Object Naming; 3. Ideational Practice.

In addition, there were six language items on the Montreal Cognitive Assessment (MoCA): Letter F Fluency, three animal naming items, and two sentence repetition tasks [8].

These items are listed in Table 1. Due to very few people missing those items, we dropped Naming Objects (Pencil and Watch) for modeling.

Table 1. Language items. The final model did not contain the MMSE-Naming Object items (Watch, Pencil).

Tests	Language items
Neuropsychological Battery	Category Fluency – Animals
	Category Fluency – Vegetables
	Boston Naming (Total)
ADAS-Cognitive Behavior	Following Commands
	Object Naming
	Ideational Practice
MMSE	Naming an Object – Watch
	Naming an Object – Pencil
	Repeating a Sentence
	Reading a Sentence
	Writing a Sentence
	Following a Series of Instructions
MoCA	Letter F Fluency

	Animal Naming – Lion, Camel, Rhino
	Sentence Repetition (two tasks)

These language indicators utilized a variety of response formats that presented challenges for constructing composite scores. The formats include counts in a pre-specified time span (Category Fluency, Boston Naming (Total), Letter F Fluency), counts of tasks performed correctly (Following Commands, Ideational Practice), polytomous ordinal responses (Object Naming, categories of 0-5 depending on how many tasks were performed correctly, out of 17 tasks) and dichotomous correct/incorrect (all other language items). We developed ordered categorical transformations of the raw data to facilitate development of composite scores that did not make strong assumptions about the distributions of scores. We recoded the raw scores into ordinal scales with as many as 10 categories, the maximum allowed by Mplus for ordinal variables. Our categorical transformations were based on the empirically observed distributions of the raw data, with a goal of maintaining variability in the tails at the expense of maintaining variability in the middle of the distributions. Specifics on how Category Fluency, Boston Naming (Total) and Letter F Fluency were transformed are given in Table 2.

Table 2 also lists three ADAS-Cog items, which were recoded due to extremely low frequencies at the baseline. Their original scores of 2-5 were merged while others were left as they were.

Table 2. Recoding of language scores. ADAS-Cog items with [*] were reverse coded so that the higher score indicates the better performance.

Original Language Score	Categorization									
	0	1	2	3	4	5	6	7	8	9
Category Fluency-Animal	0-4	5-8	9-11	12-14	15-17	18-20	21-23	24-27	28-31	32-41
Category Fluency-Vegetable	0-2	3-5	6-8	9-10	11-12	13-14	15-16	17-19	20-22	23-31
Boston Naming (Total)	0-8	9-11	12-14	15-17	18-20	21-22	23-24	25-26	27-28	29-30
Letter F Fluency	0-2	3-5	6-8	9-11	12-14	15-17	18-20	21-23	24-26	27-34
Following Commands*	2-5	1	0							
Object Naming*	2-5	1	0							
Ideational Practice*	2-5	1	0							

Further information about ADNI can be found in [2][6][7]. The study was conducted after Institutional Review Board approval at each site. Written informed consent was obtained from all study participants, or their authorized representatives.

Diagnosis of amnestic MCI required patient-reported memory complaints, objective memory deficits, intact functional activities, a global Clinical Dementia Rating (CDR) score of 0.5 [4], and a MMSE score of 24 or higher [1]. Participants with AD met the National Institute of

Neurological and Communicative Disease and Stroke – Alzheimer's Disease and Related Disorders Association criteria for people AD [3].

Category Fluency – Animals and Vegetables, Boston Naming (Total), and Letter F Fluency were numeric, not categorical. Our initial modeling of language focused on baseline data to determine whether a single factor model which treated those items as numeric performed equivalent to or better than those IRT models with them categorized (Table 2). Composite scores were computed using a unidimensional model for language.

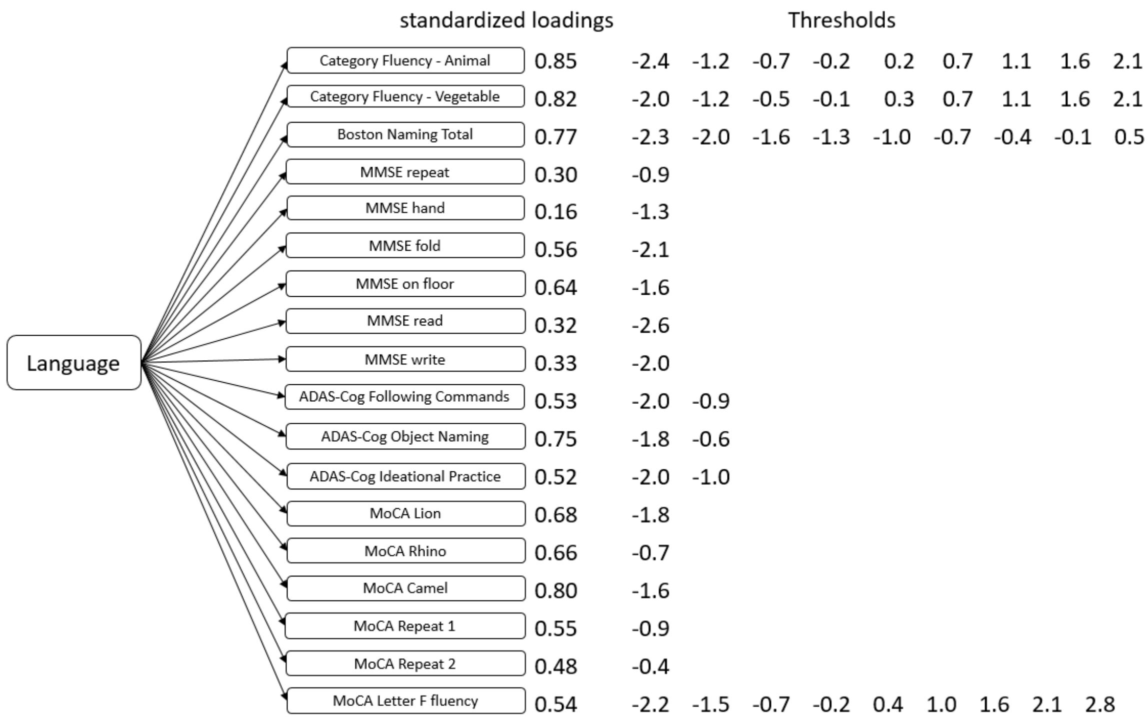
Table 3. Phase-specific language items.

Phase	Language items
Common (all phases)	Neuropsychological Battery
	Category Fluency – Animals
	MMSE
	Naming an Object – Watch
	Naming an Object – Pencil
	Repeating a sentence
	Reading a sentence
	Writing a sentence
	Following a Series of Instructions (three items)
	ADAS-Cog
	Following Commands, Object Naming, Ideational Practice
ADNI1 only	Category Fluency – Vegetables
ADNI1,2/GO only	Boston Naming (Total)
ADNI2/GO, 3 only	Animal Naming – Lion, Camel, Rhino
	Sentence Repetition (two tasks)
	Letter F Fluency

Not all items were administered over all phases of ADNI1, 2/GO and 3 (Table 3). Due to several items being missing by design in a large number of participants, an MLR estimator was needed for language. An advantage of IRT scoring is that we can still compute ADNI-Lan scores in each phase using whatever items are available in that phase, with their item parameters. The scores will be on the same metric, which means that scores from the four phases are directly comparable. By the same token, if anyone was missing an item that was supposed to be administered, we still were able to calculate a composite score.

We evaluated dimensionality. We looked at ADNI1 separately from ADNI2/GO, so as to be able to use the WLSMV estimator to assess model fit. A single factor model fit well in the ADNI1 data, with TLI 0.96, CFI 0.97, and RMSEA 0.039. The single factor model fit fairly well in the ADNI2/GO data, with TLI 0.94, CFI 0.93, and RMSEA 0.028. We considered alternate bi-factor models but opted to stay with a single factor model, in part because fewer than 1% of scores had differences greater than 0.3. The final single factor model is presented in Figure 1.

Figure1. Single factor model on language, with standardized loadings and thresholds.



We employed an item-bank approach, by fitting IRT models over these four steps:

Step 1. Fit a model using ADNI1 items, at baseline, among those who participated ADNI1. Obtain parameters for the loadings and thresholds. The mean and variance of the latent variable were fixed 0 and 1, respectively.

Step 2. Fit a model using ADNI2/GO/3 items at baseline, among those who participated ADNI2/GO or ADNI3. Table 5 tells us that six MoCA items were new in ADNI2/GO/3. Freely estimate parameters for these new items, fixing the parameters of the other items to the values obtained in Step 1.

Step 3. Fit a model using everyone in ADNI1, 2/GO and 3 at baseline and follow-up, using the item parameters from Steps 1 and 2, freely estimating the mean and variance of the latent variable.

Step 4. Obtain the composite score (ADNI-Lan) from a model fixing all parameters (loadings, thresholds) as well as the mean and variance of the latent variable from Step 3.

ADNI-VS

There are seven items related to VS (Table 5). Five of these were based on copying a clock; 1. Copy circle; 2. Copy symmetry; 3. Copy number; 4. Copy hand; 5. Copy time. They were dichotomous.

The ADAS-Cog included one VS task, Constructional Praxis. This test assessed the participant's ability to copy four geometric forms, ranging from a very simple one (circle) to a fairly difficult one (cube). The original record counted how many geometric figures couldn't be completed. As a higher score implied better performance in other items, Constructional Praxis was reverse-coded for modeling.

The MMSE included one VS task, Copy Design. The interviewer presented the participant with the construction stimulus page and said "copy this design". It was dichotomous.

Table 5. VS items.

Tests	Items
Neuropsychological Battery	Clock copy – Circle
	Clock copy – Symmetry
	Clock copy – Numbers
	Clock copy – Hands
	Clock copy – Time
ADAS- Cognitive Behavior	Constructional praxis
MMSE	Copy design

To avoid extremely low frequencies at baseline, original scores of 3-5 for Constructional Praxis were merged while others were left as they were (Table 6).

Table 6. Table for recoding of VS scores. It was reverse coded so that the higher score indicates the better performance.

Original VS Score	Categorization			
	1	2	3	4
Constructional Praxis	3-5	2	1	0

Fit statistics for the single factor model were acceptable, with TLI 0.97, CFI 0.96, and RMSEA 0.026.

All seven items were administered over all phases. However, to be consistent with ADNI-Lan calculation, we followed the same steps as we did with ADNI-Lan.

Step 1. Fit a model using all VS items, at baseline, at all phases. Estimate the loading and threshold parameters. The mean and variance of the latent variable were fixed 0 and 1, respectively.

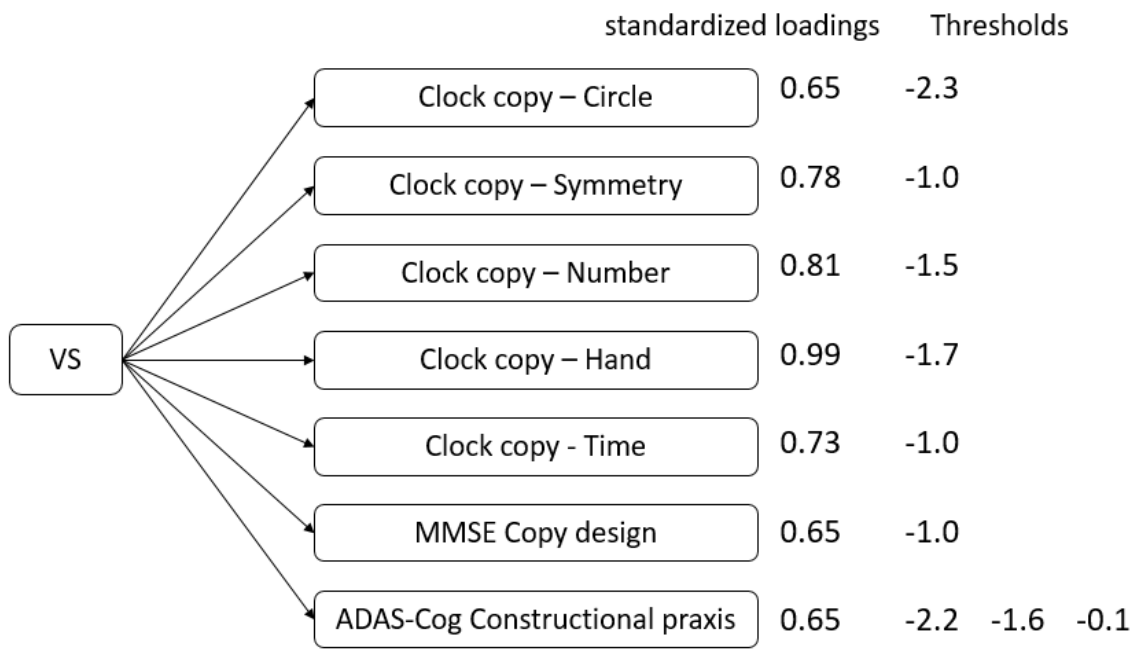
Step 2. Fit a model over all phases, at baseline and follow-up, using the item parameters obtained in Step 1, and freely estimating the mean and variance of the latent variable.

Step 3. Obtain the composite score (ADNI-VS) by fixing all parameters, including the mean and variance of the latent variable.

The WLSMV estimator was used for ADNI-VS. Fit statistics for the single factor model were acceptable, with TLI 0.97, CFI 0.96, and RMSEA 0.026. These results led us to choose a single factor model for the visuospatial domain as well.

The final unidimensional (single factor) model is presented in Figure 2.

Figure2. Single factor model on VS, with standardized loadings and thresholds



Spanish Language

A few participants have being tested in Spanish. We have computed their scores using the item parameters derived in English tests. So far there are too few Spanish test-takers to evaluate whether this is appropriate. If you want to omit their scores, look for even-numbered values of the variable "WORDLIST" in the adas_adnigo2 file.

References

[1] Folstein, M. F., Folstein, S. E., McHugh, P. R. (1975) Mini-mental state. A practical method for grading the cognitive state of patients for the clinician. *Journal of Psychiatric Research, 12(3)*, 189-198.

[2] Jack, C. R., Bernstein, M. A., Borowski, B. J., Gunter, J. L., Fox, N. C.,Thompson, P. M., Schuff, N., Krueger, G., Killiany, R. J., Decarli, C. S., Dale, A. M., Carmichael, O. W., Tosun, D., Weiner, M. W. (2010) Update on the Magnetic Resonance Imaging core of the Alzheimer's Disease Neuroimaging Initiative. *Alzheimer's & Dementia: The Journal of the Alzheimer's Association*, *6(3)*, 212-220.

[3] McKhann, G., Drachman, D., Folstein, M., Katzman, R., Price, D., Stadlan, E. M. (1984) Clinical diagnosis of Alzheimer's disease: Report of the NINCDS-ADRDA work group under the auspices of department of health and human services task force on Alzheimer's disease, *Neurology, 34(7)*, 939-944.

[4] Morris, J. C. (1993) The Clinical Dementia Rating (CDR): current version and scoring rules, *Neurology*, *43(11)*, 2412-4.

[5] Reeve, B. B., Hayes, R. D. Bjorner, J. B., Karon, F. C., Crane, P. K., Teresi, J. A., Thissen, D., Revicki, D. A., Weiss, D. J., Hambleton, R. K., Liu, H., Gershon, R., Reise, S. P., Lai, J., Cella, D. (2007) Psychometric evaluation and calibration of health-related qualify of life item banks: plans for the Patient-Reported Outcomes Measurement Information System (PROMIS). *Medical Care, 45(5)*, S22-31.

[6] ADNI-MRI http://adni.loni.usc.edu/methods/mri-tool/

[7] ADNI Procedure Manuals http://adni.loni.usc.edu/methods/documents/

[8] Montreal Cognitive Assessment (MoCA) https://www.mocatest.org/

[data] [9] ADNI data http://adni.loni.usc.edu/data-samples/access-data/

About the Authors

This document was prepared by Seo-Eun Choi at the University of Washington. For more information please contact Seo-Eun Choi by email at sc214782@uw.edu.

Appendix 1. R code for ADNI-Lan

Note that the following ADNI variables were recoded as in Table A.1.

Table A.1. Recoded language items

Language item	ADNI Name	Recoded Name
Category Fluency – Animals	catanimsc	rcatanim
Category Fluency – Vegetables	catvegesc	rcatveges
Boston Naming (Total)	bnttotal	rbnttota
ADAS-Cog: Following Commands	q2score	rq2score
ADAS-Cog: Object Naming	q5scre	rq5score
ADAS-Cog: Ideational Practice	q6score	rq6score
Letter F Fluency	ffluency	rffluenc

We used the R library, MplusAutomation (version 0.7-3), which is available by typing "install.packages("MplusAutomation")" in R. You must also have Mplus on your computer for this to work.

This is based on having the recoded items in a data set called ADNI-Lan-recoded.RData.

Code is available here:

https://github.com/gibbonsl/ADNI-Composite-scores---Lan-and-VS

Appendix 2. R code for ADNI-VS

Note that the following ADNI variable was recoded as in Table A.2.

Table A.2. Recoded VS item

VS item	ADNI Name	Recoded Name
ADAS-Cog: Constructional Praxis	q3score	rq3score

We used the R library, MplusAutomation (version 0.7-3), which is available by typing "install.packages("MplusAutomation")" in R. You must also have Mplus on your computer for this to work.

This is based on having the recoded items in a data set called ADNI-VS-recoded.RData.

Code is available here:

https://github.com/gibbonsl/ADNI-Composite-scores---Lan-and-VS