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.

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.

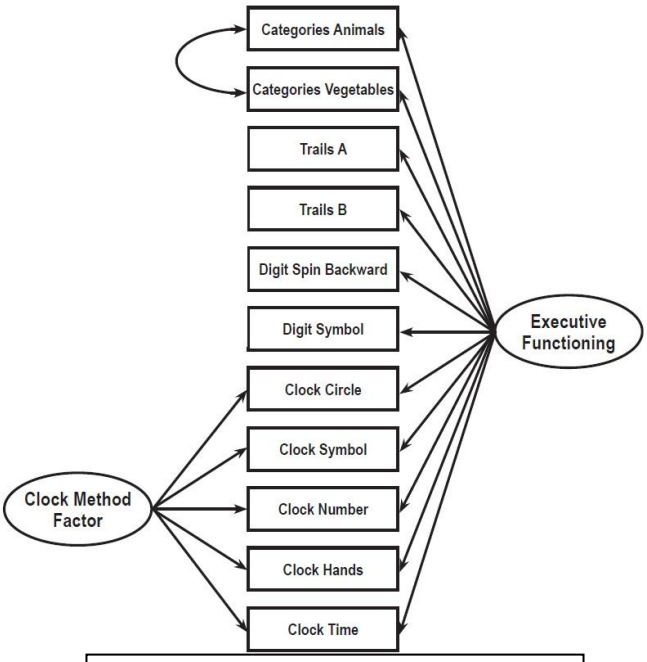


Figure 1. Bifactor model for ADNI-EF

ADNI 2 and ADNI GO 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 and ADNI GO using 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 three 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
RAVLT										
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
ADAS-Cog										
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		
Logical Memory*										
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
MMSE*										
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.

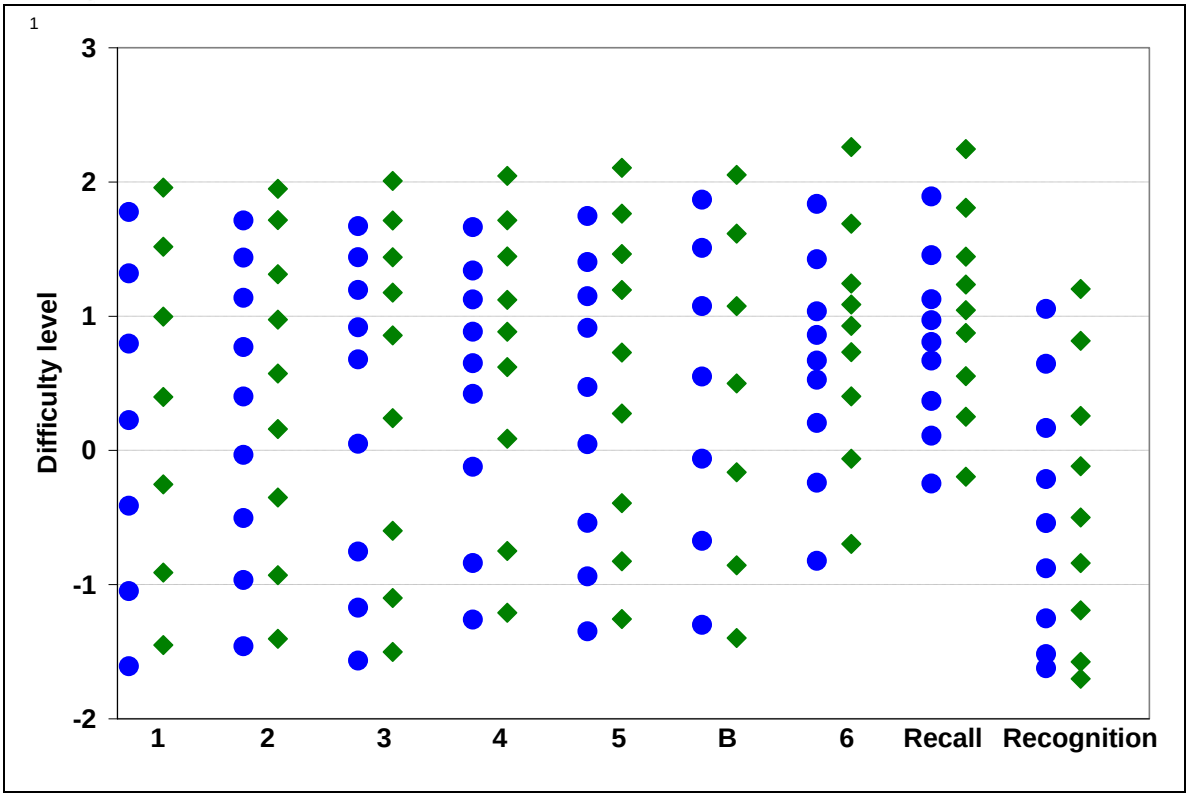
Next we divided the data set into two parts: first, the annual visits with data from all participants (baseline, month 12, and month 24), and second, the other visits (month 6, 18, and 36; the month 36 visit had structurally missing data from people in the AD group). Logical Memory was assessed at each 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.

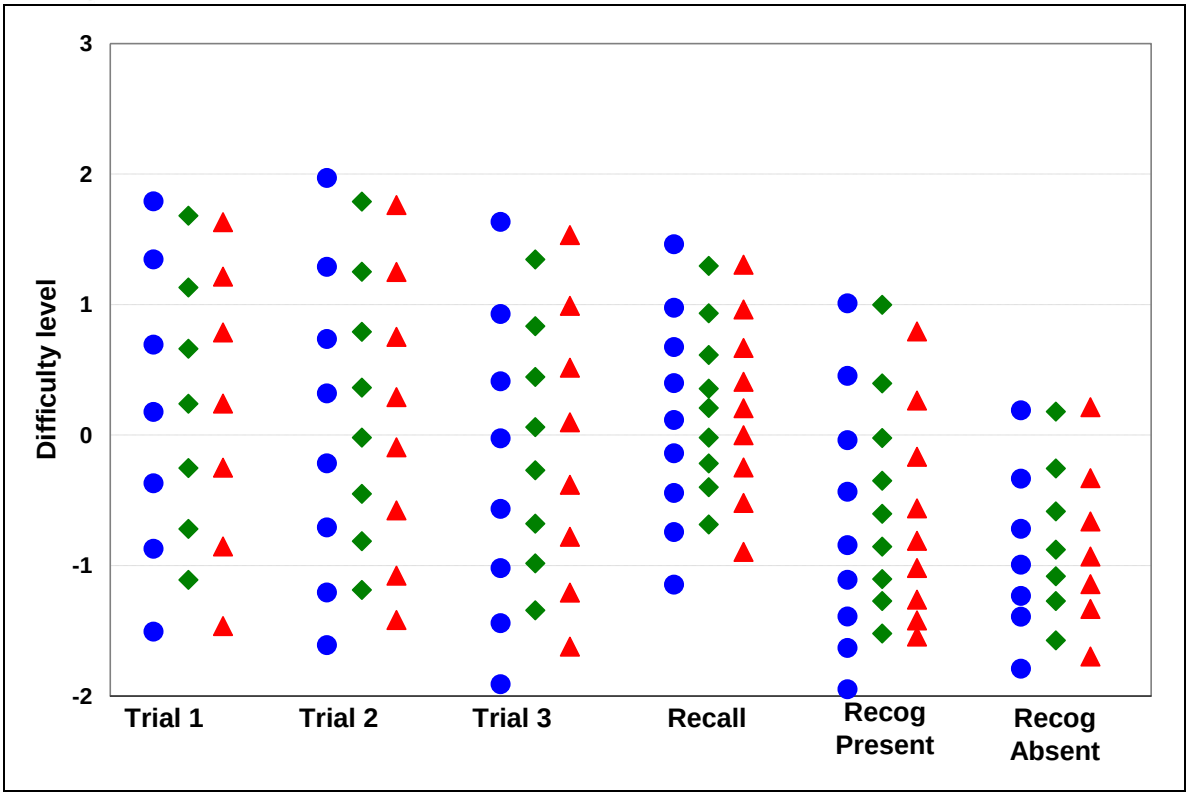
The loadings for each of the indicators from the two versions of the RAVLT were very similar; as a proportion, they ranged from 5% smaller to 3% larger between the two versions. The difficulty levels for the category thresholds, however, displayed important differences between the two versions, as shown 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.



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"
```

Stata and Mplus code (runmplus) to generate ADNI-Mem

This is based on having the recoded items in a data set called ADNI_Mem.dta

```
* a bit unwieldy, but it's safest to give you what was actually done.

use ADNI_Mem, clear
keep mm* rid visnum
* 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
#d ;
local vlist "lmrc    lmd     radrc
             ra6     adrg1   adrg2
             adlt1   adlt2   adlt3
             add     ra1     ra2
             ra3     ra4     ra5
             rab     balldl  flagdl
             treedl  rarc" ;
#d cr
reshape wide `vlist', i(rid) j(visnum)
notes:from ADNI_Mem
save "memory_wide", replace

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

```
* baseline
*****************************************************************
#d ;
local vlist lmrc0 lmd0 radrc0 rarc0 ra60 adrg10 adrg20 adlt10 adlt20
adlt30 add0 ra10 ra20 ra30 ra40 ra50 rab0 balldl0 flagdl0 treedl0;
#d cr
runmplus rid `vlist', categorical(`vlist') id(rid) ///
       model ( ///
MEM0   BY     LMRC0  @      0.837  ;      ///
MEM0   BY     LMD0   @      0.846  ;      ///
MEM0   BY     RADRC0 @      0.876  ;      ///
MEM0   BY     RARC0  @      0.728  ;      ///
MEM0   BY     RA60   @      0.85   ;      ///
MEM0   BY     ADRG10 @      0.43   ;      ///
MEM0   BY     ADRG20 @      0.507  ;      ///
MEM0   BY     ADLT10 @      0.792  ;      ///
MEM0   BY     ADLT20 @      0.858  ;      ///
MEM0   BY     ADLT30 @      0.844  ;      ///
MEM0   BY     ADD0   @      0.898  ;      ///
MEM0   BY     RA10   @      0.661  ;      ///
MEM0   BY     RA20   @      0.807  ;      ///
MEM0   BY     RA30   @      0.852  ;      ///
MEM0   BY     RA40   @      0.884  ;      ///
MEM0   BY     RA50   @      0.882  ;      ///
MEM0   BY     RAB0   @      0.615  ;      ///
MEM0   BY     BALLDL0       @      0.748  ;      ///
MEM0   BY     FLAGDL0       @      0.777  ;      ///
MEM0   BY     TREEDL0       @      0.751  ;      ///
[      LMRC0$1       @      -1.369 ]      ;      ///
[      LMRC0$2       @      -0.841 ]      ;      ///
[      LMRC0$3       @      -0.412 ]      ;      ///
[      LMRC0$4       @      -0.079 ]      ;      ///
[      LMRC0$5       @      0.225  ]      ;      ///
[      LMRC0$6       @      0.638  ]      ;      ///
[      LMRC0$7       @      0.946  ]      ;      ///
[      LMRC0$8       @      1.296  ]      ;      ///
[      LMRC0$9       @      1.702  ]      ;      ///
[      LMD0$1 @      -0.637 ]      ;      ///
[      LMD0$2 @      -0.289 ]      ;      ///
[      LMD0$3 @      0.006  ]      ;      ///
[      LMD0$4 @      0.458  ]      ;      ///
[      LMD0$5 @      0.74   ]      ;      ///
[      LMD0$6 @      0.854  ]      ;      ///
[      LMD0$7 @      0.972  ]      ;      ///
[      LMD0$8 @      1.267  ]      ;      ///
[      LMD0$9 @      1.676  ]      ;      ///
[      RADRC0$1      @      -0.248 ]      ;      ///
[      RADRC0$2      @      0.109  ]      ;      ///
[      RADRC0$3      @      0.368  ]      ;      ///
[      RADRC0$4      @      0.668  ]      ;      ///
[      RADRC0$5      @      0.807  ]      ;      ///
[      RADRC0$6      @      0.97   ]      ;      ///
[      RADRC0$7      @      1.126  ]      ;      ///
[      RADRC0$8      @      1.453  ]      ;      ///
[      RADRC0$9      @      1.891  ]      ;      ///
[      RARC0$1       @      -1.624 ]      ;      ///
[      RARC0$2       @      -1.519 ]      ;      ///
[      RARC0$3       @      -1.253 ]      ;      ///
[      RARC0$4       @      -0.88  ]      ;      ///
[      RARC0$5       @      -0.543 ]      ;      ///
[      RARC0$6       @      -0.215 ]      ;      ///
```


```
[       RARC0$7       @       0.166  ]       ;       ///
[       RARC0$8       @       0.644  ]       ;       ///
[       RARC0$9       @       1.053  ]       ;       ///
[       RA60$1 @      -0.824 ]       ;       ///
[       RA60$2 @      -0.242 ]       ;       ///
[       RA60$3 @      0.203  ]       ;       ///
[       RA60$4 @      0.526  ]       ;       ///
[       RA60$5 @      0.668  ]       ;       ///
[       RA60$6 @      0.859  ]       ;       ///
[       RA60$7 @      1.036  ]       ;       ///
[       RA60$8 @      1.424  ]       ;       ///
[       RA60$9 @      1.835  ]       ;       ///
[       ADRG10$1      @       -1.949 ]       ;       ///
[       ADRG10$2      @       -1.632 ]       ;       ///
[       ADRG10$3      @       -1.392 ]       ;       ///
[       ADRG10$4      @       -1.111 ]       ;       ///
[       ADRG10$5      @       -0.844 ]       ;       ///
[       ADRG10$6      @       -0.436 ]       ;       ///
[       ADRG10$7      @       -0.04  ]       ;       ///
[       ADRG10$8      @       0.453  ]       ;       ///
[       ADRG10$9      @       1.009  ]       ;       ///
[       ADRG20$1      @       -1.791 ]       ;       ///
[       ADRG20$2      @       -1.393 ]       ;       ///
[       ADRG20$3      @       -1.234 ]       ;       ///
[       ADRG20$4      @       -0.995 ]       ;       ///
[       ADRG20$5      @       -0.72  ]       ;       ///
[       ADRG20$6      @       -0.335 ]       ;       ///
[       ADRG20$7      @       0.189  ]       ;       ///
[       ADLT10$1      @       -1.507 ]       ;       ///
[       ADLT10$2      @       -0.872 ]       ;       ///
[       ADLT10$3      @       -0.371 ]       ;       ///
[       ADLT10$4      @       0.177  ]       ;       ///
[       ADLT10$5      @       0.693  ]       ;       ///
[       ADLT10$6      @       1.346  ]       ;       ///
[       ADLT10$7      @       1.791  ]       ;       ///
[       ADLT20$1      @       -1.61  ]       ;       ///
[       ADLT20$2      @       -1.208 ]       ;       ///
[       ADLT20$3      @       -0.709 ]       ;       ///
[       ADLT20$4      @       -0.217 ]       ;       ///
[       ADLT20$5      @       0.319  ]       ;       ///
[       ADLT20$6      @       0.736  ]       ;       ///
[       ADLT20$7      @       1.288  ]       ;       ///
[       ADLT20$8      @       1.97   ]       ;       ///
[       ADLT30$1      @       -1.91  ]       ;       ///
[       ADLT30$2      @       -1.443 ]       ;       ///
[       ADLT30$3      @       -1.02  ]       ;       ///
[       ADLT30$4      @       -0.566 ]       ;       ///
[       ADLT30$5      @       -0.026 ]       ;       ///
[       ADLT30$6      @       0.411  ]       ;       ///
[       ADLT30$7      @       0.927  ]       ;       ///
[       ADLT30$8      @       1.633  ]       ;       ///
[       ADD0$1 @      -1.147 ]       ;       ///
[       ADD0$2 @      -0.744 ]       ;       ///
[       ADD0$3 @      -0.444 ]       ;       ///
[       ADD0$4 @      -0.14  ]       ;       ///
[       ADD0$5 @      0.115  ]       ;       ///
[       ADD0$6 @      0.397  ]       ;       ///
[       ADD0$7 @      0.674  ]       ;       ///
[       ADD0$8 @      0.975  ]       ;       ///
[       ADD0$9 @      1.461  ]       ;       ///
[       RA10$1 @      -1.609 ]       ;       ///
```

```
[      RA10$2 @      -1.049 ]      ;      ///
[      RA10$3 @      -0.413 ]      ;      ///
[      RA10$4 @      0.225  ]      ;      ///
[      RA10$5 @      0.795  ]      ;      ///
[      RA10$6 @      1.319  ]      ;      ///
[      RA10$7 @      1.776  ]      ;      ///
[      RA20$1 @      -1.46  ]      ;      ///
[      RA20$2 @      -0.966 ]      ;      ///
[      RA20$3 @      -0.505 ]      ;      ///
[      RA20$4 @      -0.034 ]      ;      ///
[      RA20$5 @      0.401  ]      ;      ///
[      RA20$6 @      0.769  ]      ;      ///
[      RA20$7 @      1.137  ]      ;      ///
[      RA20$8 @      1.436  ]      ;      ///
[      RA20$9 @      1.713  ]      ;      ///
[      RA30$1 @      -1.567 ]      ;      ///
[      RA30$2 @      -1.172 ]      ;      ///
[      RA30$3 @      -0.755 ]      ;      ///
[      RA30$4 @      0.048  ]      ;      ///
[      RA30$5 @      0.677  ]      ;      ///
[      RA30$6 @      0.917  ]      ;      ///
[      RA30$7 @      1.195  ]      ;      ///
[      RA30$8 @      1.439  ]      ;      ///
[      RA30$9 @      1.67   ]      ;      ///
[      RA40$1 @      -1.261 ]      ;      ///
[      RA40$2 @      -0.84  ]      ;      ///
[      RA40$3 @      -0.122 ]      ;      ///
[      RA40$4 @      0.42   ]      ;      ///
[      RA40$5 @      0.649  ]      ;      ///
[      RA40$6 @      0.884  ]      ;      ///
[      RA40$7 @      1.124  ]      ;      ///
[      RA40$8 @      1.339  ]      ;      ///
[      RA40$9 @      1.663  ]      ;      ///
[      RA50$1 @      -1.348 ]      ;      ///
[      RA50$2 @      -0.94  ]      ;      ///
[      RA50$3 @      -0.541 ]      ;      ///
[      RA50$4 @      0.045  ]      ;      ///
[      RA50$5 @      0.472  ]      ;      ///
[      RA50$6 @      0.912  ]      ;      ///
[      RA50$7 @      1.149  ]      ;      ///
[      RA50$8 @      1.402  ]      ;      ///
[      RA50$9 @      1.746  ]      ;      ///
[      RAB0$1 @      -1.3   ]      ;      ///
[      RAB0$2 @      -0.675 ]      ;      ///
[      RAB0$3 @      -0.063 ]      ;      ///
[      RAB0$4 @      0.549  ]      ;      ///
[      RAB0$5 @      1.075  ]      ;      ///
[      RAB0$6 @      1.508  ]      ;      ///
[      RAB0$7 @      1.868  ]      ;      ///
[      BALLDL0$1     @      -0.509 ]      ;      ///
[      FLAGDL0$1     @      -0.033 ]      ;      ///
[      TREEDL0$1     @      -0.127 ]      ;      ///
) ///
   savedata(save=fscores; file=mempkc111019_00.dat) ///
savelog(trash)

preserve
runmplus_load_savedata, out("trash.out") clear
keep rid mem*
sort rid
tempfile f1
```


```
save `f1'
restore
merge 1:1 rid using `f1'
keep if _merge==3
keep rid mem0
sort rid
capture save "mem0", replace

*********************************************************************
* 48 month the same (but take out [adrg148$9])
*********************************************************************
use "memory_wide", clear

#d ;
local vlist lmrc48 lmd48 radrc48 rarc48 ra648 adrg148 adrg248 adlt148 adlt248
adlt348 add48 ra148 ra248 ra348 ra448 ra548 rab48 balldl48 flagdl48 treedl48;
#d cr
runmplus rid `vlist', categorical(`vlist') id(rid) ///
       model ( ///
mem48  BY     LMRC48 @      0.837  ;      ///
mem48  BY     LMD48  @      0.846  ;      ///
mem48  BY     RADRC48       @      0.876  ;      ///
mem48  BY     RARC48 @      0.728  ;      ///
mem48  BY     RA648  @      0.85   ;      ///
mem48  BY     ADRG148       @      0.43   ;      ///
mem48  BY     ADRG248       @      0.507  ;      ///
mem48  BY     ADLT148       @      0.792  ;      ///
mem48  BY     ADLT248       @      0.858  ;      ///
mem48  BY     ADLT348       @      0.844  ;      ///
mem48  BY     ADD48  @      0.898  ;      ///
mem48  BY     RA148  @      0.661  ;      ///
mem48  BY     RA248  @      0.807  ;      ///
mem48  BY     RA348  @      0.852  ;      ///
mem48  BY     RA448  @      0.884  ;      ///
mem48  BY     RA548  @      0.882  ;      ///
mem48  BY     RAB48  @      0.615  ;      ///
mem48  BY     BALLDL48      @      0.748  ;      ///
mem48  BY     FLAGDL48      @      0.777  ;      ///
mem48  BY     TREEDL48      @      0.751  ;      ///
[      LMRC48$1      @      -1.369 ]      ;      ///
[      LMRC48$2      @      -0.841 ]      ;      ///
[      LMRC48$3      @      -0.412 ]      ;      ///
[      LMRC48$4      @      -0.079 ]      ;      ///
[      LMRC48$5      @      0.225  ]      ;      ///
[      LMRC48$6      @      0.638  ]      ;      ///
[      LMRC48$7      @      0.946  ]      ;      ///
[      LMRC48$8      @      1.296  ]      ;      ///
[      LMRC48$9      @      1.702  ]      ;      ///
[      LMD48$1       @      -0.637 ]      ;      ///
[      LMD48$2       @      -0.289 ]      ;      ///
[      LMD48$3       @      0.006  ]      ;      ///
[      LMD48$4       @      0.458  ]      ;      ///
[      LMD48$5       @      0.74   ]      ;      ///
[      LMD48$6       @      0.854  ]      ;      ///
[      LMD48$7       @      0.972  ]      ;      ///
[      LMD48$8       @      1.267  ]      ;      ///
[      LMD48$9       @      1.676  ]      ;      ///
[      RADRC48$1     @      -0.248 ]      ;      ///
[      RADRC48$2     @      0.109  ]      ;      ///
[      RADRC48$3     @      0.368  ]      ;      ///
[      RADRC48$4     @      0.668  ]      ;      ///
```

```
[       RADRC48$5     @       0.807  ]       ;       ///
[       RADRC48$6     @       0.97   ]       ;       ///
[       RADRC48$7     @       1.126  ]       ;       ///
[       RADRC48$8     @       1.453  ]       ;       ///
[       RADRC48$9     @       1.891  ]       ;       ///
[       RARC48$1      @       -1.624 ]       ;       ///
[       RARC48$2      @       -1.519 ]       ;       ///
[       RARC48$3      @       -1.253 ]       ;       ///
[       RARC48$4      @       -0.88  ]       ;       ///
[       RARC48$5      @       -0.543 ]       ;       ///
[       RARC48$6      @       -0.215 ]       ;       ///
[       RARC48$7      @       0.166  ]       ;       ///
[       RARC48$8      @       0.644  ]       ;       ///
[       RARC48$9      @       1.053  ]       ;       ///
[       RA648$1       @       -0.824 ]       ;       ///
[       RA648$2       @       -0.242 ]       ;       ///
[       RA648$3       @       0.203  ]       ;       ///
[       RA648$4       @       0.526  ]       ;       ///
[       RA648$5       @       0.668  ]       ;       ///
[       RA648$6       @       0.859  ]       ;       ///
[       RA648$7       @       1.036  ]       ;       ///
[       RA648$8       @       1.424  ]       ;       ///
[       RA648$9       @       1.835  ]       ;       ///
[       ADRG148$1     @       -1.949 ]       ;       ///
[       ADRG148$2     @       -1.632 ]       ;       ///
[       ADRG148$3     @       -1.392 ]       ;       ///
[       ADRG148$4     @       -1.111 ]       ;       ///
[       ADRG148$5     @       -0.844 ]       ;       ///
[       ADRG148$6     @       -0.436 ]       ;       ///
[       ADRG148$7     @       -0.04  ]       ;       ///
[       ADRG148$8     @       0.453  ]       ;       ///
[       ADRG248$1     @       -1.791 ]       ;       ///
[       ADRG248$2     @       -1.393 ]       ;       ///
[       ADRG248$3     @       -1.234 ]       ;       ///
[       ADRG248$4     @       -0.995 ]       ;       ///
[       ADRG248$5     @       -0.72  ]       ;       ///
[       ADRG248$6     @       -0.335 ]       ;       ///
[       ADRG248$7     @       0.189  ]       ;       ///
[       ADLT148$1     @       -1.507 ]       ;       ///
[       ADLT148$2     @       -0.872 ]       ;       ///
[       ADLT148$3     @       -0.371 ]       ;       ///
[       ADLT148$4     @       0.177  ]       ;       ///
[       ADLT148$5     @       0.693  ]       ;       ///
[       ADLT148$6     @       1.346  ]       ;       ///
[       ADLT148$7     @       1.791  ]       ;       ///
[       ADLT248$1     @       -1.61  ]       ;       ///
[       ADLT248$2     @       -1.208 ]       ;       ///
[       ADLT248$3     @       -0.709 ]       ;       ///
[       ADLT248$4     @       -0.217 ]       ;       ///
[       ADLT248$5     @       0.319  ]       ;       ///
[       ADLT248$6     @       0.736  ]       ;       ///
[       ADLT248$7     @       1.288  ]       ;       ///
[       ADLT248$8     @       1.97   ]       ;       ///
[       ADLT348$1     @       -1.91  ]       ;       ///
[       ADLT348$2     @       -1.443 ]       ;       ///
[       ADLT348$3     @       -1.02  ]       ;       ///
[       ADLT348$4     @       -0.566 ]       ;       ///
[       ADLT348$5     @       -0.026 ]       ;       ///
[       ADLT348$6     @       0.411  ]       ;       ///
[       ADLT348$7     @       0.927  ]       ;       ///
[       ADLT348$8     @       1.633  ]       ;       ///
```

```
[       ADD48$1       @       -1.147 ]       ;       ///
[       ADD48$2       @       -0.744 ]       ;       ///
[       ADD48$3       @       -0.444 ]       ;       ///
[       ADD48$4       @       -0.14  ]       ;       ///
[       ADD48$5       @       0.115  ]       ;       ///
[       ADD48$6       @       0.397  ]       ;       ///
[       ADD48$7       @       0.674  ]       ;       ///
[       ADD48$8       @       0.975  ]       ;       ///
[       ADD48$9       @       1.461  ]       ;       ///
[       RA148$1       @       -1.609 ]       ;       ///
[       RA148$2       @       -1.049 ]       ;       ///
[       RA148$3       @       -0.413 ]       ;       ///
[       RA148$4       @       0.225  ]       ;       ///
[       RA148$5       @       0.795  ]       ;       ///
[       RA148$6       @       1.319  ]       ;       ///
[       RA148$7       @       1.776  ]       ;       ///
[       RA248$1       @       -1.46  ]       ;       ///
[       RA248$2       @       -0.966 ]       ;       ///
[       RA248$3       @       -0.505 ]       ;       ///
[       RA248$4       @       -0.034 ]       ;       ///
[       RA248$5       @       0.401  ]       ;       ///
[       RA248$6       @       0.769  ]       ;       ///
[       RA248$7       @       1.137  ]       ;       ///
[       RA248$8       @       1.436  ]       ;       ///
[       RA248$9       @       1.713  ]       ;       ///
[       RA348$1       @       -1.567 ]       ;       ///
[       RA348$2       @       -1.172 ]       ;       ///
[       RA348$3       @       -0.755 ]       ;       ///
[       RA348$4       @       0.048  ]       ;       ///
[       RA348$5       @       0.677  ]       ;       ///
[       RA348$6       @       0.917  ]       ;       ///
[       RA348$7       @       1.195  ]       ;       ///
[       RA348$8       @       1.439  ]       ;       ///
[       RA348$9       @       1.67   ]       ;       ///
[       RA448$1       @       -1.261 ]       ;       ///
[       RA448$2       @       -0.84  ]       ;       ///
[       RA448$3       @       -0.122 ]       ;       ///
[       RA448$4       @       0.42   ]       ;       ///
[       RA448$5       @       0.649  ]       ;       ///
[       RA448$6       @       0.884  ]       ;       ///
[       RA448$7       @       1.124  ]       ;       ///
[       RA448$8       @       1.339  ]       ;       ///
[       RA448$9       @       1.663  ]       ;       ///
[       RA548$1       @       -1.348 ]       ;       ///
[       RA548$2       @       -0.94  ]       ;       ///
[       RA548$3       @       -0.541 ]       ;       ///
[       RA548$4       @       0.045  ]       ;       ///
[       RA548$5       @       0.472  ]       ;       ///
[       RA548$6       @       0.912  ]       ;       ///
[       RA548$7       @       1.149  ]       ;       ///
[       RA548$8       @       1.402  ]       ;       ///
[       RA548$9       @       1.746  ]       ;       ///
[       RAB48$1       @       -1.3   ]       ;       ///
[       RAB48$2       @       -0.675 ]       ;       ///
[       RAB48$3       @       -0.063 ]       ;       ///
[       RAB48$4       @       0.549  ]       ;       ///
[       RAB48$5       @       1.075  ]       ;       ///
[       RAB48$6       @       1.508  ]       ;       ///
[       RAB48$7       @       1.868  ]       ;       ///
[       BALLDL48$1    @       -0.509 ]       ;       ///
[       FLAGDL48$1    @       -0.033 ]       ;       ///
```


```
[      TREEDL48$1    @      -0.127 ]      ;      ///
) ///
   savedata(save=fscores; file=mempkc111019_00.dat) ///
savelog(trash)

preserve
runmplus_load_savedata, out("trash.out") clear
keep rid mem*
sort rid
tempfile f1
save `f1'
restore
merge 1:1 rid using `f1'
keep if _merge==3
keep rid mem48
sort rid
capture save "mem48", replace

*********************************************************************
* 6-month
*********************************************************************
use "memory_wide", clear

#d ;
local vlist radrc6 rarc6 ra66 adrg16 adrg26 adlt16 adlt26 adlt36 add6
ra16 ra26 ra36 ra46 ra56 rab6 balldl6 flagdl6 treedl6;
#d cr

runmplus rid `vlist', categorical(`vlist') id(rid) ///
      model ( ///
MEM6   BY     RADRC6 @      0.885  ;      ///
MEM6   BY     RARC6  @      0.716  ;      ///
MEM6   BY     RA66   @      0.862  ;      ///
MEM6   BY     ADRG16 @      0.478  ;      ///
MEM6   BY     ADRG26 @      0.533  ;      ///
MEM6   BY     ADLT16 @      0.79   ;      ///
MEM6   BY     ADLT26 @      0.828  ;      ///
MEM6   BY     ADLT36 @      0.835  ;      ///
MEM6   BY     ADD6   @      0.862  ;      ///
MEM6   BY     RA16   @      0.649  ;      ///
MEM6   BY     RA26   @      0.826  ;      ///
MEM6   BY     RA36   @      0.876  ;      ///
MEM6   BY     RA46   @      0.884  ;      ///
MEM6   BY     RA56   @      0.877  ;      ///
MEM6   BY     RAB6   @      0.582  ;      ///
MEM6   BY     BALLDL6      @      0.748  ;      ///
MEM6   BY     FLAGDL6      @      0.777  ;      ///
MEM6   BY     TREEDL6      @      0.751  ;      ///
[      RADRC6$1      @      -0.196 ]      ;      ///
[      RADRC6$2      @      0.25   ]      ;      ///
[      RADRC6$3      @      0.553  ]      ;      ///
[      RADRC6$4      @      0.875  ]      ;      ///
[      RADRC6$5      @      1.044  ]      ;      ///
[      RADRC6$6      @      1.235  ]      ;      ///
[      RADRC6$7      @      1.443  ]      ;      ///
[      RADRC6$8      @      1.807  ]      ;      ///
[      RADRC6$9      @      2.245  ]      ;      ///
[      RARC6$1       @      -1.702 ]      ;      ///
[      RARC6$2       @      -1.576 ]      ;      ///
[      RARC6$3       @      -1.192 ]      ;      ///
[      RARC6$4       @      -0.841 ]      ;      ///
```


```
[       RARC6$5       @       -0.5    ]       ;       ///
[       RARC6$6       @       -0.118  ]       ;       ///
[       RARC6$7       @       0.257   ]       ;       ///
[       RARC6$8       @       0.816   ]       ;       ///
[       RARC6$9       @       1.203   ]       ;       ///
[       RA66$1 @      -0.697  ]       ;       ///
[       RA66$2 @      -0.062  ]       ;       ///
[       RA66$3 @      0.402   ]       ;       ///
[       RA66$4 @      0.732   ]       ;       ///
[       RA66$5 @      0.927   ]       ;       ///
[       RA66$6 @      1.087   ]       ;       ///
[       RA66$7 @      1.243   ]       ;       ///
[       RA66$8 @      1.688   ]       ;       ///
[       RA66$9 @      2.26    ]       ;       ///
[       ADRG16$1      @       -1.521  ]       ;       ///
[       ADRG16$2      @       -1.272  ]       ;       ///
[       ADRG16$3      @       -1.104  ]       ;       ///
[       ADRG16$4      @       -0.856  ]       ;       ///
[       ADRG16$5      @       -0.603  ]       ;       ///
[       ADRG16$6      @       -0.35   ]       ;       ///
[       ADRG16$7      @       -0.023  ]       ;       ///
[       ADRG16$8      @       0.395   ]       ;       ///
[       ADRG16$9      @       0.998   ]       ;       ///
[       ADRG26$1      @       -1.574  ]       ;       ///
[       ADRG26$2      @       -1.272  ]       ;       ///
[       ADRG26$3      @       -1.083  ]       ;       ///
[       ADRG26$4      @       -0.879  ]       ;       ///
[       ADRG26$5      @       -0.585  ]       ;       ///
[       ADRG26$6      @       -0.256  ]       ;       ///
[       ADRG26$7      @       0.179   ]       ;       ///
[       ADLT16$1      @       -1.111  ]       ;       ///
[       ADLT16$2      @       -0.719  ]       ;       ///
[       ADLT16$3      @       -0.253  ]       ;       ///
[       ADLT16$4      @       0.24    ]       ;       ///
[       ADLT16$5      @       0.662   ]       ;       ///
[       ADLT16$6      @       1.131   ]       ;       ///
[       ADLT16$7      @       1.681   ]       ;       ///
[       ADLT26$1      @       -1.187  ]       ;       ///
[       ADLT26$2      @       -0.814  ]       ;       ///
[       ADLT26$3      @       -0.45   ]       ;       ///
[       ADLT26$4      @       -0.019  ]       ;       ///
[       ADLT26$5      @       0.364   ]       ;       ///
[       ADLT26$6      @       0.791   ]       ;       ///
[       ADLT26$7      @       1.252   ]       ;       ///
[       ADLT26$8      @       1.789   ]       ;       ///
[       ADLT36$1      @       -1.343  ]       ;       ///
[       ADLT36$2      @       -0.984  ]       ;       ///
[       ADLT36$3      @       -0.68   ]       ;       ///
[       ADLT36$4      @       -0.27   ]       ;       ///
[       ADLT36$5      @       0.06    ]       ;       ///
[       ADLT36$6      @       0.445   ]       ;       ///
[       ADLT36$7      @       0.835   ]       ;       ///
[       ADLT36$8      @       1.345   ]       ;       ///
[       ADD6$1 @      -0.685  ]       ;       ///
[       ADD6$2 @      -0.399  ]       ;       ///
[       ADD6$3 @      -0.217  ]       ;       ///
[       ADD6$4 @      -0.019  ]       ;       ///
[       ADD6$5 @      0.207   ]       ;       ///
[       ADD6$6 @      0.355   ]       ;       ///
[       ADD6$7 @      0.613   ]       ;       ///
[       ADD6$8 @      0.934   ]       ;       ///
```

```
[      ADD6$9 @      1.297  ]      ;      ///
[      RA16$1 @      -1.451 ]      ;      ///
[      RA16$2 @      -0.911 ]      ;      ///
[      RA16$3 @      -0.253 ]      ;      ///
[      RA16$4 @      0.398  ]      ;      ///
[      RA16$5 @      0.997  ]      ;      ///
[      RA16$6 @      1.517  ]      ;      ///
[      RA16$7 @      1.958  ]      ;      ///
[      RA26$1 @      -1.405 ]      ;      ///
[      RA26$2 @      -0.931 ]      ;      ///
[      RA26$3 @      -0.351 ]      ;      ///
[      RA26$4 @      0.159  ]      ;      ///
[      RA26$5 @      0.572  ]      ;      ///
[      RA26$6 @      0.974  ]      ;      ///
[      RA26$7 @      1.313  ]      ;      ///
[      RA26$8 @      1.716  ]      ;      ///
[      RA26$9 @      1.948  ]      ;      ///
[      RA36$1 @      -1.502 ]      ;      ///
[      RA36$2 @      -1.101 ]      ;      ///
[      RA36$3 @      -0.599 ]      ;      ///
[      RA36$4 @      0.24   ]      ;      ///
[      RA36$5 @      0.857  ]      ;      ///
[      RA36$6 @      1.176  ]      ;      ///
[      RA36$7 @      1.438  ]      ;      ///
[      RA36$8 @      1.712  ]      ;      ///
[      RA36$9 @      2.007  ]      ;      ///
[      RA46$1 @      -1.211 ]      ;      ///
[      RA46$2 @      -0.75  ]      ;      ///
[      RA46$3 @      0.086  ]      ;      ///
[      RA46$4 @      0.621  ]      ;      ///
[      RA46$5 @      0.884  ]      ;      ///
[      RA46$6 @      1.121  ]      ;      ///
[      RA46$7 @      1.444  ]      ;      ///
[      RA46$8 @      1.714  ]      ;      ///
[      RA46$9 @      2.045  ]      ;      ///
[      RA56$1 @      -1.257 ]      ;      ///
[      RA56$2 @      -0.826 ]      ;      ///
[      RA56$3 @      -0.393 ]      ;      ///
[      RA56$4 @      0.274  ]      ;      ///
[      RA56$5 @      0.728  ]      ;      ///
[      RA56$6 @      1.195  ]      ;      ///
[      RA56$7 @      1.463  ]      ;      ///
[      RA56$8 @      1.764  ]      ;      ///
[      RA56$9 @      2.105  ]      ;      ///
[      RAB6$1 @      -1.399 ]      ;      ///
[      RAB6$2 @      -0.857 ]      ;      ///
[      RAB6$3 @      -0.164 ]      ;      ///
[      RAB6$4 @      0.498  ]      ;      ///
[      RAB6$5 @      1.075  ]      ;      ///
[      RAB6$6 @      1.614  ]      ;      ///
[      RAB6$7 @      2.052  ]      ;      ///
[      BALLDL6$1     @      -0.463 ]      ;      ///
[      FLAGDL6$1     @      -0.024 ]      ;      ///
[      TREEDL6$1     @      -0.139 ]      ;      ///
) ///
   savedata(save=fscores; file=mempkc110608.dat) ///
savelog(trash)

preserve
runmplus_load_savedata, out("trash.out") clear
keep rid mem*
```

```
sort rid
tempfile f1
save `f1'
restore
merge 1:1 rid using `f1'
keep if _merge==3
keep rid mem6
sort rid
capture save "mem6", replace

*****************************************************************
* 12-month
*****************************************************************
use "memory_wide", clear

#d ;
local vlist lmrc12 lmd12 radrc12 rarc12 ra612 adrg112 adrg212 adlt112 adlt212 adlt312
add12 ra112 ra212 ra312 ra412 ra512 rab12 balldl12 flagdl12 treedl12;
#d cr

runmplus rid `vlist', categorical(`vlist') id(rid) ///
       model ( ///
MEM12  BY     LMRC12 @      0.837  ;      ///
MEM12  BY     LMD12  @      0.846  ;      ///
MEM12  BY     RADRC12       @      0.876  ;      ///
MEM12  BY     RARC12 @      0.728  ;      ///
MEM12  BY     RA612  @      0.85   ;      ///
MEM12  BY     ADRG112       @      0.559  ;      ///
MEM12  BY     ADRG212       @      0.473  ;      ///
MEM12  BY     ADLT112       @      0.765  ;      ///
MEM12  BY     ADLT212       @      0.839  ;      ///
MEM12  BY     ADLT312       @      0.847  ;      ///
MEM12  BY     ADD12  @      0.877  ;      ///
MEM12  BY     RA112  @      0.661  ;      ///
MEM12  BY     RA212  @      0.807  ;      ///
MEM12  BY     RA312  @      0.852  ;      ///
MEM12  BY     RA412  @      0.884  ;      ///
MEM12  BY     RA512  @      0.882  ;      ///
MEM12  BY     RAB12  @      0.615  ;      ///
MEM12  BY     BALLDL12      @      0.748  ;      ///
MEM12  BY     FLAGDL12      @      0.777  ;      ///
MEM12  BY     TREEDL12      @      0.751  ;      ///
[      LMRC12$1      @      -1.369 ]      ;      ///
[      LMRC12$2      @      -0.841 ]      ;      ///
[      LMRC12$3      @      -0.412 ]      ;      ///
[      LMRC12$4      @      -0.079 ]      ;      ///
[      LMRC12$5      @      0.225  ]      ;      ///
[      LMRC12$6      @      0.638  ]      ;      ///
[      LMRC12$7      @      0.946  ]      ;      ///
[      LMRC12$8      @      1.296  ]      ;      ///
[      LMRC12$9      @      1.702  ]      ;      ///
[      LMD12$1       @      -0.637 ]      ;      ///
[      LMD12$2       @      -0.289 ]      ;      ///
[      LMD12$3       @      0.006  ]      ;      ///
[      LMD12$4       @      0.458  ]      ;      ///
[      LMD12$5       @      0.74   ]      ;      ///
[      LMD12$6       @      0.854  ]      ;      ///
[      LMD12$7       @      0.972  ]      ;      ///
[      LMD12$8       @      1.267  ]      ;      ///
[      LMD12$9       @      1.676  ]      ;      ///
[      RADRC12$1     @      -0.248 ]      ;      ///
```

```
[       RADRC12$2     @       0.109  ]       ;       ///
[       RADRC12$3     @       0.368  ]       ;       ///
[       RADRC12$4     @       0.668  ]       ;       ///
[       RADRC12$5     @       0.807  ]       ;       ///
[       RADRC12$6     @       0.97   ]       ;       ///
[       RADRC12$7     @       1.126  ]       ;       ///
[       RADRC12$8     @       1.453  ]       ;       ///
[       RADRC12$9     @       1.891  ]       ;       ///
[       RARC12$1      @       -1.624 ]       ;       ///
[       RARC12$2      @       -1.519 ]       ;       ///
[       RARC12$3      @       -1.253 ]       ;       ///
[       RARC12$4      @       -0.88  ]       ;       ///
[       RARC12$5      @       -0.543 ]       ;       ///
[       RARC12$6      @       -0.215 ]       ;       ///
[       RARC12$7      @       0.166  ]       ;       ///
[       RARC12$8      @       0.644  ]       ;       ///
[       RARC12$9      @       1.053  ]       ;       ///
[       RA612$1       @       -0.824 ]       ;       ///
[       RA612$2       @       -0.242 ]       ;       ///
[       RA612$3       @       0.203  ]       ;       ///
[       RA612$4       @       0.526  ]       ;       ///
[       RA612$5       @       0.668  ]       ;       ///
[       RA612$6       @       0.859  ]       ;       ///
[       RA612$7       @       1.036  ]       ;       ///
[       RA612$8       @       1.424  ]       ;       ///
[       RA612$9       @       1.835  ]       ;       ///
[       ADRG112$1     @       -1.545 ]       ;       ///
[       ADRG112$2     @       -1.421 ]       ;       ///
[       ADRG112$3     @       -1.262 ]       ;       ///
[       ADRG112$4     @       -1.019 ]       ;       ///
[       ADRG112$5     @       -0.81  ]       ;       ///
[       ADRG112$6     @       -0.562 ]       ;       ///
[       ADRG112$7     @       -0.167 ]       ;       ///
[       ADRG112$8     @       0.265  ]       ;       ///
[       ADRG112$9     @       0.795  ]       ;       ///
[       ADRG212$1     @       -1.697 ]       ;       ///
[       ADRG212$2     @       -1.333 ]       ;       ///
[       ADRG212$3     @       -1.142 ]       ;       ///
[       ADRG212$4     @       -0.931 ]       ;       ///
[       ADRG212$5     @       -0.662 ]       ;       ///
[       ADRG212$6     @       -0.331 ]       ;       ///
[       ADRG212$7     @       0.214  ]       ;       ///
[       ADLT112$1     @       -1.464 ]       ;       ///
[       ADLT112$2     @       -0.854 ]       ;       ///
[       ADLT112$3     @       -0.25  ]       ;       ///
[       ADLT112$4     @       0.242  ]       ;       ///
[       ADLT112$5     @       0.786  ]       ;       ///
[       ADLT112$6     @       1.214  ]       ;       ///
[       ADLT112$7     @       1.631  ]       ;       ///
[       ADLT212$1     @       -1.417 ]       ;       ///
[       ADLT212$2     @       -1.078 ]       ;       ///
[       ADLT212$3     @       -0.577 ]       ;       ///
[       ADLT212$4     @       -0.094 ]       ;       ///
[       ADLT212$5     @       0.291  ]       ;       ///
[       ADLT212$6     @       0.753  ]       ;       ///
[       ADLT212$7     @       1.25   ]       ;       ///
[       ADLT212$8     @       1.762  ]       ;       ///
[       ADLT312$1     @       -1.621 ]       ;       ///
[       ADLT312$2     @       -1.208 ]       ;       ///
[       ADLT312$3     @       -0.779 ]       ;       ///
[       ADLT312$4     @       -0.381 ]       ;       ///
```

```
[       ADLT312$5     @       0.098  ]       ;       ///
[       ADLT312$6     @       0.516  ]       ;       ///
[       ADLT312$7     @       0.991  ]       ;       ///
[       ADLT312$8     @       1.533  ]       ;       ///
[       ADD12$1       @       -0.895 ]       ;       ///
[       ADD12$2       @       -0.521 ]       ;       ///
[       ADD12$3       @       -0.248 ]       ;       ///
[       ADD12$4       @       -0.002 ]       ;       ///
[       ADD12$5       @       0.206  ]       ;       ///
[       ADD12$6       @       0.408  ]       ;       ///
[       ADD12$7       @       0.668  ]       ;       ///
[       ADD12$8       @       0.962  ]       ;       ///
[       ADD12$9       @       1.305  ]       ;       ///
[       RA112$1       @       -1.609 ]       ;       ///
[       RA112$2       @       -1.049 ]       ;       ///
[       RA112$3       @       -0.413 ]       ;       ///
[       RA112$4       @       0.225  ]       ;       ///
[       RA112$5       @       0.795  ]       ;       ///
[       RA112$6       @       1.319  ]       ;       ///
[       RA112$7       @       1.776  ]       ;       ///
[       RA212$1       @       -1.46  ]       ;       ///
[       RA212$2       @       -0.966 ]       ;       ///
[       RA212$3       @       -0.505 ]       ;       ///
[       RA212$4       @       -0.034 ]       ;       ///
[       RA212$5       @       0.401  ]       ;       ///
[       RA212$6       @       0.769  ]       ;       ///
[       RA212$7       @       1.137  ]       ;       ///
[       RA212$8       @       1.436  ]       ;       ///
[       RA212$9       @       1.713  ]       ;       ///
[       RA312$1       @       -1.567 ]       ;       ///
[       RA312$2       @       -1.172 ]       ;       ///
[       RA312$3       @       -0.755 ]       ;       ///
[       RA312$4       @       0.048  ]       ;       ///
[       RA312$5       @       0.677  ]       ;       ///
[       RA312$6       @       0.917  ]       ;       ///
[       RA312$7       @       1.195  ]       ;       ///
[       RA312$8       @       1.439  ]       ;       ///
[       RA312$9       @       1.67   ]       ;       ///
[       RA412$1       @       -1.261 ]       ;       ///
[       RA412$2       @       -0.84  ]       ;       ///
[       RA412$3       @       -0.122 ]       ;       ///
[       RA412$4       @       0.42   ]       ;       ///
[       RA412$5       @       0.649  ]       ;       ///
[       RA412$6       @       0.884  ]       ;       ///
[       RA412$7       @       1.124  ]       ;       ///
[       RA412$8       @       1.339  ]       ;       ///
[       RA412$9       @       1.663  ]       ;       ///
[       RA512$1       @       -1.348 ]       ;       ///
[       RA512$2       @       -0.94  ]       ;       ///
[       RA512$3       @       -0.541 ]       ;       ///
[       RA512$4       @       0.045  ]       ;       ///
[       RA512$5       @       0.472  ]       ;       ///
[       RA512$6       @       0.912  ]       ;       ///
[       RA512$7       @       1.149  ]       ;       ///
[       RA512$8       @       1.402  ]       ;       ///
[       RA512$9       @       1.746  ]       ;       ///
[       RAB12$1       @       -1.3   ]       ;       ///
[       RAB12$2       @       -0.675 ]       ;       ///
[       RAB12$3       @       -0.063 ]       ;       ///
[       RAB12$4       @       0.549  ]       ;       ///
[       RAB12$5       @       1.075  ]       ;       ///
```


```
[      RAB12$6       @      1.508  ]      ;      ///
[      RAB12$7       @      1.868  ]      ;      ///
[      BALLDL12$1    @      -0.509 ]      ;      ///
[      FLAGDL12$1    @      -0.033 ]      ;      ///
[      TREEDL12$1    @      -0.127 ]      ;      ///
) ///
   savedata(save=fscores; file=mempkc111019_12.dat) ///
savelog(trash)

preserve
runmplus_load_savedata, out("trash.out") clear
keep rid mem*
sort rid
tempfile f1
save `f1'
restore
merge 1:1 rid using `f1'
keep if _merge==3
keep rid mem12
sort rid
capture save "mem12", replace

*********************************************************************
* 18 month
*********************************************************************
use "memory_wide", clear

#d ;
local vlist radrc18 rarc18 ra618 adrg118 adrg218 adlt118 adlt218 adlt318 add18
ra118 ra218 ra318 ra418 ra518 rab18 balldl18 flagdl18 treedl18;
#d cr

runmplus rid `vlist', categorical(`vlist') id(rid) ///
      model ( ///
MEM18  BY     RADRC18       @      0.885  ;      ///
MEM18  BY     RARC18 @      0.716  ;      ///
MEM18  BY     RA618  @      0.862  ;      ///
MEM18  BY     ADRG118       @      0.43   ;      ///
MEM18  BY     ADRG218       @      0.507  ;      ///
MEM18  BY     ADLT118       @      0.792  ;      ///
MEM18  BY     ADLT218       @      0.858  ;      ///
MEM18  BY     ADLT318       @      0.844  ;      ///
MEM18  BY     ADD18  @      0.898  ;      ///
MEM18  BY     RA118  @      0.649  ;      ///
MEM18  BY     RA218  @      0.826  ;      ///
MEM18  BY     RA318  @      0.876  ;      ///
MEM18  BY     RA418  @      0.884  ;      ///
MEM18  BY     RA518  @      0.877  ;      ///
MEM18  BY     RAB18  @      0.582  ;      ///
MEM18  BY     BALLDL18      @      0.748  ;      ///
MEM18  BY     FLAGDL18      @      0.777  ;      ///
MEM18  BY     TREEDL18      @      0.751  ;      ///
[      RADRC18$1     @      -0.196 ]      ;      ///
[      RADRC18$2     @      0.25   ]      ;      ///
[      RADRC18$3     @      0.553  ]      ;      ///
[      RADRC18$4     @      0.875  ]      ;      ///
[      RADRC18$5     @      1.044  ]      ;      ///
[      RADRC18$6     @      1.235  ]      ;      ///
[      RADRC18$7     @      1.443  ]      ;      ///
[      RADRC18$8     @      1.807  ]      ;      ///
[      RADRC18$9     @      2.245  ]      ;      ///
```


```
[       RARC18$1      @       -1.702 ]       ;       ///
[       RARC18$2      @       -1.576 ]       ;       ///
[       RARC18$3      @       -1.192 ]       ;       ///
[       RARC18$4      @       -0.841 ]       ;       ///
[       RARC18$5      @       -0.5   ]       ;       ///
[       RARC18$6      @       -0.118 ]       ;       ///
[       RARC18$7      @       0.257  ]       ;       ///
[       RARC18$8      @       0.816  ]       ;       ///
[       RARC18$9      @       1.203  ]       ;       ///
[       RA618$1       @       -0.697 ]       ;       ///
[       RA618$2       @       -0.062 ]       ;       ///
[       RA618$3       @       0.402  ]       ;       ///
[       RA618$4       @       0.732  ]       ;       ///
[       RA618$5       @       0.927  ]       ;       ///
[       RA618$6       @       1.087  ]       ;       ///
[       RA618$7       @       1.243  ]       ;       ///
[       RA618$8       @       1.688  ]       ;       ///
[       RA618$9       @       2.26   ]       ;       ///
[       ADRG118$1     @       -1.949 ]       ;       ///
[       ADRG118$2     @       -1.632 ]       ;       ///
[       ADRG118$3     @       -1.392 ]       ;       ///
[       ADRG118$4     @       -1.111 ]       ;       ///
[       ADRG118$5     @       -0.844 ]       ;       ///
[       ADRG118$6     @       -0.436 ]       ;       ///
[       ADRG118$7     @       -0.04  ]       ;       ///
[       ADRG118$8     @       0.453  ]       ;       ///
[       ADRG118$9     @       1.009  ]       ;       ///
[       ADRG218$1     @       -1.791 ]       ;       ///
[       ADRG218$2     @       -1.393 ]       ;       ///
[       ADRG218$3     @       -1.234 ]       ;       ///
[       ADRG218$4     @       -0.995 ]       ;       ///
[       ADRG218$5     @       -0.72  ]       ;       ///
[       ADRG218$6     @       -0.335 ]       ;       ///
[       ADRG218$7     @       0.189  ]       ;       ///
[       ADLT118$1     @       -1.507 ]       ;       ///
[       ADLT118$2     @       -0.872 ]       ;       ///
[       ADLT118$3     @       -0.371 ]       ;       ///
[       ADLT118$4     @       0.177  ]       ;       ///
[       ADLT118$5     @       0.693  ]       ;       ///
[       ADLT118$6     @       1.346  ]       ;       ///
[       ADLT118$7     @       1.791  ]       ;       ///
[       ADLT218$1     @       -1.61  ]       ;       ///
[       ADLT218$2     @       -1.208 ]       ;       ///
[       ADLT218$3     @       -0.709 ]       ;       ///
[       ADLT218$4     @       -0.217 ]       ;       ///
[       ADLT218$5     @       0.319  ]       ;       ///
[       ADLT218$6     @       0.736  ]       ;       ///
[       ADLT218$7     @       1.288  ]       ;       ///
[       ADLT218$8     @       1.97   ]       ;       ///
[       ADLT318$1     @       -1.91  ]       ;       ///
[       ADLT318$2     @       -1.443 ]       ;       ///
[       ADLT318$3     @       -1.02  ]       ;       ///
[       ADLT318$4     @       -0.566 ]       ;       ///
[       ADLT318$5     @       -0.026 ]       ;       ///
[       ADLT318$6     @       0.411  ]       ;       ///
[       ADLT318$7     @       0.927  ]       ;       ///
[       ADLT318$8     @       1.633  ]       ;       ///
[       ADD18$1       @       -1.147 ]       ;       ///
[       ADD18$2       @       -0.744 ]       ;       ///
[       ADD18$3       @       -0.444 ]       ;       ///
[       ADD18$4       @       -0.14  ]       ;       ///
```

```
[      ADD18$5       @      0.115  ]      ;      ///
[      ADD18$6       @      0.397  ]      ;      ///
[      ADD18$7       @      0.674  ]      ;      ///
[      ADD18$8       @      0.975  ]      ;      ///
[      ADD18$9       @      1.461  ]      ;      ///
[      RA118$1       @      -1.451 ]      ;      ///
[      RA118$2       @      -0.911 ]      ;      ///
[      RA118$3       @      -0.253 ]      ;      ///
[      RA118$4       @      0.398  ]      ;      ///
[      RA118$5       @      0.997  ]      ;      ///
[      RA118$6       @      1.517  ]      ;      ///
[      RA118$7       @      1.958  ]      ;      ///
[      RA218$1       @      -1.405 ]      ;      ///
[      RA218$2       @      -0.931 ]      ;      ///
[      RA218$3       @      -0.351 ]      ;      ///
[      RA218$4       @      0.159  ]      ;      ///
[      RA218$5       @      0.572  ]      ;      ///
[      RA218$6       @      0.974  ]      ;      ///
[      RA218$7       @      1.313  ]      ;      ///
[      RA218$8       @      1.716  ]      ;      ///
[      RA218$9       @      1.948  ]      ;      ///
[      RA318$1       @      -1.502 ]      ;      ///
[      RA318$2       @      -1.101 ]      ;      ///
[      RA318$3       @      -0.599 ]      ;      ///
[      RA318$4       @      0.24   ]      ;      ///
[      RA318$5       @      0.857  ]      ;      ///
[      RA318$6       @      1.176  ]      ;      ///
[      RA318$7       @      1.438  ]      ;      ///
[      RA318$8       @      1.712  ]      ;      ///
[      RA318$9       @      2.007  ]      ;      ///
[      RA418$1       @      -1.211 ]      ;      ///
[      RA418$2       @      -0.75  ]      ;      ///
[      RA418$3       @      0.086  ]      ;      ///
[      RA418$4       @      0.621  ]      ;      ///
[      RA418$5       @      0.884  ]      ;      ///
[      RA418$6       @      1.121  ]      ;      ///
[      RA418$7       @      1.444  ]      ;      ///
[      RA418$8       @      1.714  ]      ;      ///
[      RA418$9       @      2.045  ]      ;      ///
[      RA518$1       @      -1.257 ]      ;      ///
[      RA518$2       @      -0.826 ]      ;      ///
[      RA518$3       @      -0.393 ]      ;      ///
[      RA518$4       @      0.274  ]      ;      ///
[      RA518$5       @      0.728  ]      ;      ///
[      RA518$6       @      1.195  ]      ;      ///
[      RA518$7       @      1.463  ]      ;      ///
[      RA518$8       @      1.764  ]      ;      ///
[      RA518$9       @      2.105  ]      ;      ///
[      RAB18$1       @      -1.399 ]      ;      ///
[      RAB18$2       @      -0.857 ]      ;      ///
[      RAB18$3       @      -0.164 ]      ;      ///
[      RAB18$4       @      0.498  ]      ;      ///
[      RAB18$5       @      1.075  ]      ;      ///
[      RAB18$6       @      1.614  ]      ;      ///
[      RAB18$7       @      2.052  ]      ;      ///
[      BALLDL18$1    @      -0.632 ]      ;      ///
[      FLAGDL18$1    @      -0.153 ]      ;      ///
[      TREEDL18$1    @      -0.269 ]      ;      ///
) ///
   savedata(save=fscores; file=mempkc111019_18.dat) ///
savelog(trash)
```

```
preserve
runmplus_load_savedata, out("trash.out") clear
keep rid mem*
sort rid
tempfile f1
save `f1'
restore
merge 1:1 rid using `f1'
keep if _merge==3
keep rid mem18
sort rid
capture save "mem18", replace

*****************************************************************
* 24 month
*****************************************************************
use "memory_wide", clear
#d ;
local vlist lmrc24 lmd24 radrc24 rarc24 ra624 adrg124 adrg224 adlt124 adlt224 adlt324
add24 ra124 ra224 ra324 ra424 ra524 rab24 balldl24 flagdl24 treedl24 ;
#d cr

runmplus rid `vlist', categorical(`vlist') id(rid) ///
      model ( ///
MEM24  BY     LMRC24 @      0.837  ;      ///
MEM24  BY     LMD24  @      0.846  ;      ///
MEM24  BY     RADRC24       @      0.876  ;      ///
MEM24  BY     RARC24 @      0.728  ;      ///
MEM24  BY     RA624  @      0.85   ;      ///
MEM24  BY     ADRG124       @      0.478  ;      ///
MEM24  BY     ADRG224       @      0.533  ;      ///
MEM24  BY     ADLT124       @      0.79   ;      ///
MEM24  BY     ADLT224       @      0.828  ;      ///
MEM24  BY     ADLT324       @      0.835  ;      ///
MEM24  BY     ADD24  @      0.862  ;      ///
MEM24  BY     RA124  @      0.661  ;      ///
MEM24  BY     RA224  @      0.807  ;      ///
MEM24  BY     RA324  @      0.852  ;      ///
MEM24  BY     RA424  @      0.884  ;      ///
MEM24  BY     RA524  @      0.882  ;      ///
MEM24  BY     RAB24  @      0.615  ;      ///
MEM24  BY     BALLDL24      @      0.748  ;      ///
MEM24  BY     FLAGDL24      @      0.777  ;      ///
MEM24  BY     TREEDL24      @      0.751  ;      ///
[      LMRC24$1      @      -1.369 ]      ;      ///
[      LMRC24$2      @      -0.841 ]      ;      ///
[      LMRC24$3      @      -0.412 ]      ;      ///
[      LMRC24$4      @      -0.079 ]      ;      ///
[      LMRC24$5      @      0.225  ]      ;      ///
[      LMRC24$6      @      0.638  ]      ;      ///
[      LMRC24$7      @      0.946  ]      ;      ///
[      LMRC24$8      @      1.296  ]      ;      ///
[      LMRC24$9      @      1.702  ]      ;      ///
[      LMD24$1       @      -0.637 ]      ;      ///
[      LMD24$2       @      -0.289 ]      ;      ///
[      LMD24$3       @      0.006  ]      ;      ///
[      LMD24$4       @      0.458  ]      ;      ///
[      LMD24$5       @      0.74   ]      ;      ///
[      LMD24$6       @      0.854  ]      ;      ///
[      LMD24$7       @      0.972  ]      ;      ///
```


```
[       LMD24$8       @       1.267  ]       ;       ///
[       LMD24$9       @       1.676  ]       ;       ///
[       RADRC24$1     @       -0.248 ]       ;       ///
[       RADRC24$2     @       0.109  ]       ;       ///
[       RADRC24$3     @       0.368  ]       ;       ///
[       RADRC24$4     @       0.668  ]       ;       ///
[       RADRC24$5     @       0.807  ]       ;       ///
[       RADRC24$6     @       0.97   ]       ;       ///
[       RADRC24$7     @       1.126  ]       ;       ///
[       RADRC24$8     @       1.453  ]       ;       ///
[       RADRC24$9     @       1.891  ]       ;       ///
[       RARC24$1      @       -1.624 ]       ;       ///
[       RARC24$2      @       -1.519 ]       ;       ///
[       RARC24$3      @       -1.253 ]       ;       ///
[       RARC24$4      @       -0.88  ]       ;       ///
[       RARC24$5      @       -0.543 ]       ;       ///
[       RARC24$6      @       -0.215 ]       ;       ///
[       RARC24$7      @       0.166  ]       ;       ///
[       RARC24$8      @       0.644  ]       ;       ///
[       RARC24$9      @       1.053  ]       ;       ///
[       RA624$1       @       -0.824 ]       ;       ///
[       RA624$2       @       -0.242 ]       ;       ///
[       RA624$3       @       0.203  ]       ;       ///
[       RA624$4       @       0.526  ]       ;       ///
[       RA624$5       @       0.668  ]       ;       ///
[       RA624$6       @       0.859  ]       ;       ///
[       RA624$7       @       1.036  ]       ;       ///
[       RA624$8       @       1.424  ]       ;       ///
[       RA624$9       @       1.835  ]       ;       ///
[       ADRG124$1     @       -1.521 ]       ;       ///
[       ADRG124$2     @       -1.272 ]       ;       ///
[       ADRG124$3     @       -1.104 ]       ;       ///
[       ADRG124$4     @       -0.856 ]       ;       ///
[       ADRG124$5     @       -0.603 ]       ;       ///
[       ADRG124$6     @       -0.35  ]       ;       ///
[       ADRG124$7     @       -0.023 ]       ;       ///
[       ADRG124$8     @       0.395  ]       ;       ///
[       ADRG124$9     @       0.998  ]       ;       ///
[       ADRG224$1     @       -1.574 ]       ;       ///
[       ADRG224$2     @       -1.272 ]       ;       ///
[       ADRG224$3     @       -1.083 ]       ;       ///
[       ADRG224$4     @       -0.879 ]       ;       ///
[       ADRG224$5     @       -0.585 ]       ;       ///
[       ADRG224$6     @       -0.256 ]       ;       ///
[       ADRG224$7     @       0.179  ]       ;       ///
[       ADLT124$1     @       -1.111 ]       ;       ///
[       ADLT124$2     @       -0.719 ]       ;       ///
[       ADLT124$3     @       -0.253 ]       ;       ///
[       ADLT124$4     @       0.24   ]       ;       ///
[       ADLT124$5     @       0.662  ]       ;       ///
[       ADLT124$6     @       1.131  ]       ;       ///
[       ADLT124$7     @       1.681  ]       ;       ///
[       ADLT224$1     @       -1.187 ]       ;       ///
[       ADLT224$2     @       -0.814 ]       ;       ///
[       ADLT224$3     @       -0.45  ]       ;       ///
[       ADLT224$4     @       -0.019 ]       ;       ///
[       ADLT224$5     @       0.364  ]       ;       ///
[       ADLT224$6     @       0.791  ]       ;       ///
[       ADLT224$7     @       1.252  ]       ;       ///
[       ADLT224$8     @       1.789  ]       ;       ///
[       ADLT324$1     @       -1.343 ]       ;       ///
```

```
[       ADLT324$2     @       -0.984 ]       ;       ///
[       ADLT324$3     @       -0.68  ]       ;       ///
[       ADLT324$4     @       -0.27  ]       ;       ///
[       ADLT324$5     @       0.06   ]       ;       ///
[       ADLT324$6     @       0.445  ]       ;       ///
[       ADLT324$7     @       0.835  ]       ;       ///
[       ADLT324$8     @       1.345  ]       ;       ///
[       ADD24$1       @       -0.685 ]       ;       ///
[       ADD24$2       @       -0.399 ]       ;       ///
[       ADD24$3       @       -0.217 ]       ;       ///
[       ADD24$4       @       -0.019 ]       ;       ///
[       ADD24$5       @       0.207  ]       ;       ///
[       ADD24$6       @       0.355  ]       ;       ///
[       ADD24$7       @       0.613  ]       ;       ///
[       ADD24$8       @       0.934  ]       ;       ///
[       ADD24$9       @       1.297  ]       ;       ///
[       RA124$1       @       -1.609 ]       ;       ///
[       RA124$2       @       -1.049 ]       ;       ///
[       RA124$3       @       -0.413 ]       ;       ///
[       RA124$4       @       0.225  ]       ;       ///
[       RA124$5       @       0.795  ]       ;       ///
[       RA124$6       @       1.319  ]       ;       ///
[       RA124$7       @       1.776  ]       ;       ///
[       RA224$1       @       -1.46  ]       ;       ///
[       RA224$2       @       -0.966 ]       ;       ///
[       RA224$3       @       -0.505 ]       ;       ///
[       RA224$4       @       -0.034 ]       ;       ///
[       RA224$5       @       0.401  ]       ;       ///
[       RA224$6       @       0.769  ]       ;       ///
[       RA224$7       @       1.137  ]       ;       ///
[       RA224$8       @       1.436  ]       ;       ///
[       RA224$9       @       1.713  ]       ;       ///
[       RA324$1       @       -1.567 ]       ;       ///
[       RA324$2       @       -1.172 ]       ;       ///
[       RA324$3       @       -0.755 ]       ;       ///
[       RA324$4       @       0.048  ]       ;       ///
[       RA324$5       @       0.677  ]       ;       ///
[       RA324$6       @       0.917  ]       ;       ///
[       RA324$7       @       1.195  ]       ;       ///
[       RA324$8       @       1.439  ]       ;       ///
[       RA324$9       @       1.67   ]       ;       ///
[       RA424$1       @       -1.261 ]       ;       ///
[       RA424$2       @       -0.84  ]       ;       ///
[       RA424$3       @       -0.122 ]       ;       ///
[       RA424$4       @       0.42   ]       ;       ///
[       RA424$5       @       0.649  ]       ;       ///
[       RA424$6       @       0.884  ]       ;       ///
[       RA424$7       @       1.124  ]       ;       ///
[       RA424$8       @       1.339  ]       ;       ///
[       RA424$9       @       1.663  ]       ;       ///
[       RA524$1       @       -1.348 ]       ;       ///
[       RA524$2       @       -0.94  ]       ;       ///
[       RA524$3       @       -0.541 ]       ;       ///
[       RA524$4       @       0.045  ]       ;       ///
[       RA524$5       @       0.472  ]       ;       ///
[       RA524$6       @       0.912  ]       ;       ///
[       RA524$7       @       1.149  ]       ;       ///
[       RA524$8       @       1.402  ]       ;       ///
[       RA524$9       @       1.746  ]       ;       ///
[       RAB24$1       @       -1.3   ]       ;       ///
[       RAB24$2       @       -0.675 ]       ;       ///
```


```
[      RAB24$3       @      -0.063 ]      ;      ///
[      RAB24$4       @      0.549  ]      ;      ///
[      RAB24$5       @      1.075  ]      ;      ///
[      RAB24$6       @      1.508  ]      ;      ///
[      RAB24$7       @      1.868  ]      ;      ///
[      BALLDL24$1    @      -0.509 ]      ;      ///
[      FLAGDL24$1    @      -0.033 ]      ;      ///
[      TREEDL24$1    @      -0.127 ]      ;      ///
) ///
   savedata(save=fscores; file=mempkc111019_24.dat) ///
savelog(trash)

preserve
runmplus_load_savedata, out("trash.out") clear
keep rid mem*
sort rid
tempfile f1
save `f1'
restore
merge 1:1 rid using `f1'
keep if _merge==3
keep rid mem24
sort rid
capture save "mem24", replace

*********************************************************************
* 36 month
*********************************************************************
use "memory_wide", clear
#d ;
local vlist lmrc36 lmd36 radrc36 rarc36 ra636 adrg136 adrg236 adlt136 adlt236 adlt336
add36 ra136 ra236 ra336 ra436 ra536 rab36 balldl36 flagdl36 treedl36;
#d cr

runmplus rid `vlist', categorical(`vlist') id(rid) ///
      model ( ///
MEM36  BY     LMRC36 @      0.837  ;      ///
MEM36  BY     LMD36  @      0.846  ;      ///
MEM36  BY     RADRC36       @      0.885  ;      ///
MEM36  BY     RARC36 @      0.716  ;      ///
MEM36  BY     RA636  @      0.862  ;      ///
MEM36  BY     ADRG136       @      0.559  ;      ///
MEM36  BY     ADRG236       @      0.473  ;      ///
MEM36  BY     ADLT136       @      0.765  ;      ///
MEM36  BY     ADLT236       @      0.839  ;      ///
MEM36  BY     ADLT336       @      0.847  ;      ///
MEM36  BY     ADD36  @      0.877  ;      ///
MEM36  BY     RA136  @      0.649  ;      ///
MEM36  BY     RA236  @      0.826  ;      ///
MEM36  BY     RA336  @      0.876  ;      ///
MEM36  BY     RA436  @      0.884  ;      ///
MEM36  BY     RA536  @      0.877  ;      ///
MEM36  BY     RAB36  @      0.582  ;      ///
MEM36  BY     BALLDL36      @      0.748  ;      ///
MEM36  BY     FLAGDL36      @      0.777  ;      ///
MEM36  BY     TREEDL36      @      0.751  ;      ///
[      LMRC36$1      @      -1.369 ]      ;      ///
[      LMRC36$2      @      -0.841 ]      ;      ///
[      LMRC36$3      @      -0.412 ]      ;      ///
[      LMRC36$4      @      -0.079 ]      ;      ///
[      LMRC36$5      @      0.225  ]      ;      ///
```

```
[       LMRC36$6      @       0.638  ]       ;       ///
[       LMRC36$7      @       0.946  ]       ;       ///
[       LMRC36$8      @       1.296  ]       ;       ///
[       LMRC36$9      @       1.702  ]       ;       ///
[       LMD36$1       @       -0.637 ]       ;       ///
[       LMD36$2       @       -0.289 ]       ;       ///
[       LMD36$3       @       0.006  ]       ;       ///
[       LMD36$4       @       0.458  ]       ;       ///
[       LMD36$5       @       0.74   ]       ;       ///
[       LMD36$6       @       0.854  ]       ;       ///
[       LMD36$7       @       0.972  ]       ;       ///
[       LMD36$8       @       1.267  ]       ;       ///
[       LMD36$9       @       1.676  ]       ;       ///
[       RADRC36$1     @       -0.196 ]       ;       ///
[       RADRC36$2     @       0.25   ]       ;       ///
[       RADRC36$3     @       0.553  ]       ;       ///
[       RADRC36$4     @       0.875  ]       ;       ///
[       RADRC36$5     @       1.044  ]       ;       ///
[       RADRC36$6     @       1.235  ]       ;       ///
[       RADRC36$7     @       1.443  ]       ;       ///
[       RADRC36$8     @       1.807  ]       ;       ///
[       RADRC36$9     @       2.245  ]       ;       ///
[       RARC36$1      @       -1.702 ]       ;       ///
[       RARC36$2      @       -1.576 ]       ;       ///
[       RARC36$3      @       -1.192 ]       ;       ///
[       RARC36$4      @       -0.841 ]       ;       ///
[       RARC36$5      @       -0.5   ]       ;       ///
[       RARC36$6      @       -0.118 ]       ;       ///
[       RARC36$7      @       0.257  ]       ;       ///
[       RARC36$8      @       0.816  ]       ;       ///
[       RARC36$9      @       1.203  ]       ;       ///
[       RA636$1       @       -0.697 ]       ;       ///
[       RA636$2       @       -0.062 ]       ;       ///
[       RA636$3       @       0.402  ]       ;       ///
[       RA636$4       @       0.732  ]       ;       ///
[       RA636$5       @       0.927  ]       ;       ///
[       RA636$6       @       1.087  ]       ;       ///
[       RA636$7       @       1.243  ]       ;       ///
[       RA636$8       @       1.688  ]       ;       ///
[       RA636$9       @       2.26   ]       ;       ///
[       ADRG136$1     @       -1.573 ]       ;       ///
[       ADRG136$2     @       -1.349 ]       ;       ///
[       ADRG136$3     @       -1.168 ]       ;       ///
[       ADRG136$4     @       -1.023 ]       ;       ///
[       ADRG136$5     @       -0.791 ]       ;       ///
[       ADRG136$6     @       -0.459 ]       ;       ///
[       ADRG136$7     @       -0.089 ]       ;       ///
[       ADRG136$8     @       0.347  ]       ;       ///
[       ADRG136$9     @       0.96   ]       ;       ///
[       ADRG236$1     @       -2.114 ]       ;       ///
[       ADRG236$2     @       -1.548 ]       ;       ///
[       ADRG236$3     @       -1.195 ]       ;       ///
[       ADRG236$4     @       -0.958 ]       ;       ///
[       ADRG236$5     @       -0.736 ]       ;       ///
[       ADRG236$6     @       -0.328 ]       ;       ///
[       ADRG236$7     @       0.251  ]       ;       ///
[       ADLT136$1     @       -1.201 ]       ;       ///
[       ADLT136$2     @       -0.8   ]       ;       ///
[       ADLT136$3     @       -0.214 ]       ;       ///
[       ADLT136$4     @       0.285  ]       ;       ///
[       ADLT136$5     @       0.767  ]       ;       ///
```

```
[       ADLT136$6     @       1.321  ]       ;       ///
[       ADLT136$7     @       1.808  ]       ;       ///
[       ADLT236$1     @       -1.228 ]       ;       ///
[       ADLT236$2     @       -0.859 ]       ;       ///
[       ADLT236$3     @       -0.429 ]       ;       ///
[       ADLT236$4     @       -0.09  ]       ;       ///
[       ADLT236$5     @       0.32   ]       ;       ///
[       ADLT236$6     @       0.72   ]       ;       ///
[       ADLT236$7     @       1.221  ]       ;       ///
[       ADLT236$8     @       1.893  ]       ;       ///
[       ADLT336$1     @       -1.299 ]       ;       ///
[       ADLT336$2     @       -1.011 ]       ;       ///
[       ADLT336$3     @       -0.659 ]       ;       ///
[       ADLT336$4     @       -0.277 ]       ;       ///
[       ADLT336$5     @       0.112  ]       ;       ///
[       ADLT336$6     @       0.513  ]       ;       ///
[       ADLT336$7     @       0.986  ]       ;       ///
[       ADLT336$8     @       1.592  ]       ;       ///
[       ADD36$1       @       -0.704 ]       ;       ///
[       ADD36$2       @       -0.379 ]       ;       ///
[       ADD36$3       @       -0.157 ]       ;       ///
[       ADD36$4       @       -0.014 ]       ;       ///
[       ADD36$5       @       0.145  ]       ;       ///
[       ADD36$6       @       0.376  ]       ;       ///
[       ADD36$7       @       0.612  ]       ;       ///
[       ADD36$8       @       1.049  ]       ;       ///
[       ADD36$9       @       1.499  ]       ;       ///
[       RA136$1       @       -1.451 ]       ;       ///
[       RA136$2       @       -0.911 ]       ;       ///
[       RA136$3       @       -0.253 ]       ;       ///
[       RA136$4       @       0.398  ]       ;       ///
[       RA136$5       @       0.997  ]       ;       ///
[       RA136$6       @       1.517  ]       ;       ///
[       RA136$7       @       1.958  ]       ;       ///
[       RA236$1       @       -1.405 ]       ;       ///
[       RA236$2       @       -0.931 ]       ;       ///
[       RA236$3       @       -0.351 ]       ;       ///
[       RA236$4       @       0.159  ]       ;       ///
[       RA236$5       @       0.572  ]       ;       ///
[       RA236$6       @       0.974  ]       ;       ///
[       RA236$7       @       1.313  ]       ;       ///
[       RA236$8       @       1.716  ]       ;       ///
[       RA236$9       @       1.948  ]       ;       ///
[       RA336$1       @       -1.502 ]       ;       ///
[       RA336$2       @       -1.101 ]       ;       ///
[       RA336$3       @       -0.599 ]       ;       ///
[       RA336$4       @       0.24   ]       ;       ///
[       RA336$5       @       0.857  ]       ;       ///
[       RA336$6       @       1.176  ]       ;       ///
[       RA336$7       @       1.438  ]       ;       ///
[       RA336$8       @       1.712  ]       ;       ///
[       RA336$9       @       2.007  ]       ;       ///
[       RA436$1       @       -1.211 ]       ;       ///
[       RA436$2       @       -0.75  ]       ;       ///
[       RA436$3       @       0.086  ]       ;       ///
[       RA436$4       @       0.621  ]       ;       ///
[       RA436$5       @       0.884  ]       ;       ///
[       RA436$6       @       1.121  ]       ;       ///
[       RA436$7       @       1.444  ]       ;       ///
[       RA436$8       @       1.714  ]       ;       ///
[       RA436$9       @       2.045  ]       ;       ///
```


```
[      RA536$1      @      -1.257 ]      ;      ///
[      RA536$2      @      -0.826 ]      ;      ///
[      RA536$3      @      -0.393 ]      ;      ///
[      RA536$4      @      0.274  ]      ;      ///
[      RA536$5      @      0.728  ]      ;      ///
[      RA536$6      @      1.195  ]      ;      ///
[      RA536$7      @      1.463  ]      ;      ///
[      RA536$8      @      1.764  ]      ;      ///
[      RA536$9      @      2.105  ]      ;      ///
[      RAB36$1      @      -1.399 ]      ;      ///
[      RAB36$2      @      -0.857 ]      ;      ///
[      RAB36$3      @      -0.164 ]      ;      ///
[      RAB36$4      @      0.498  ]      ;      ///
[      RAB36$5      @      1.075  ]      ;      ///
[      RAB36$6      @      1.614  ]      ;      ///
[      RAB36$7      @      2.052  ]      ;      ///
[      BALLDL36$1   @      -0.408 ]      ;      ///
[      FLAGDL36$1   @      0.028  ]      ;      ///
[      TREEDL36$1   @      -0.143 ]      ;      ///
) ///
   savedata(save=fscores; file=mempkc111019_36.dat) ///
savelog(trash)

preserve
runmplus_load_savedata, out("trash.out") clear
keep rid mem*
sort rid
tempfile f1
save `f1'
restore
merge 1:1 rid using `f1'
keep if _merge==3
keep rid mem36
sort rid
capture save "mem36", replace

! erase ef_scores_bifactor.txt
! erase mempkc*.dat
! erase __*.dct
! erase trash.*
! erase memory_wide.dta

foreach i in 0 6 12 18 24 36 48 {
       tempfile m`i'
       use mem`i', clear
       reshape long mem, i(rid) j(visnum)
       save `m`i''
       merge 1:1 rid visnum using ADNI_Mem, nogen update
       save ADNI_Mem, replace
       }

rename mem adni_mem
la var adni_mem "Memory summary score"

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