|  |  |  |  |
| --- | --- | --- | --- |
|  | Memory | Common Executive Function | Processing Speed |
| LM Immediate | .60 |  |  |
| LM Delayed | .62 |  |  |
| VR Immediate | .51 |  |  |
| VR Delayed | .51 |  |  |
| CVLT Associates | .49 |  |  |
| CVLT Immediate Free Recall | .65 |  |  |
| CVLT Delayed Free Recall | .66 |  |  |
| Stroop Interference |  | .31 |  |
| Trai Making Task Part 1 |  | -.55 |  |
| Category Switching |  | .22 |  |
| Letter Number Sequencing |  | .49 |  |
| Reading Span Acquisition |  | .38 |  |
| Digit Span Total |  | .44 |  |
| Trail Making Task Part 2 |  |  | -0.37 |
| Trail Making Task Part 3 |  |  | -0.47 |
| Stroop Word |  |  | .61 |
| Stroop Color |  |  | .68 |
| Stroop Reaction Time |  |  | -.41 |
| Choice Reaction Time |  |  | -.46 |
| Note. LMI = Logical Memory; VR = Visual Reproduction; CV = California Verbal Learning Test; DS = Digit Span; TMT = Trail Making Task; LNS = Letter Number Sequencing; RS = Reading Span. Each task is standardized (z-score) before factor calculation. | | | |

R Code:

library(dplyr)

library(psych)

dat<- read.csv("~/netshare/M/NAS VETSA MASTER DATAFILES/Other cognitive measures/Cognitive data practice effects/V1V2V3/Practice\_Corrected\_Cognitive\_Data\_V1V2V3\_2020-06-17.csv",head=T, stringsAsFactors = F)

afqtbx<- read.csv("~/netshare/M/NAS VETSA MASTER DATAFILES/Other cognitive measures/AFQT/DATA/AFQTV1V2V3 11.10.2020.csv")

afqtbx<- afqtbx %>% select(VETSAID, starts\_with("AFQTBXPCTTRAN\_R"))

dat<- dat %>% left\_join(afqtbx, by="VETSAID")

names(dat)

dat<-arrange(dat, VETSAID)

attach(dat)

# J Elman adding in hard-coded corrections per CR's SAS data prep script

dat[dat$VETSAID=="20527A","VRCTOT"] = NA

dat[dat$VETSAID=="20201B","VRCTOT"] = 67

####################################################################

########### Standarize Variables to Wave 1 ################

####################################################################

## Memory

LMITOT\_V1\_stndV1 = scale(LMITOT)

LMITOT\_V2\_stndV1 = (LMITOT\_V2p-mean(LMITOT,na.rm=T))/sd(LMITOT,na.rm=T)

LMITOT\_V3\_stndV1 = (LMITOT\_V3p-mean(LMITOT,na.rm=T))/sd(LMITOT,na.rm=T)

LMDTOT\_V1\_stndV1 = scale(LMDTOT)

LMDTOT\_V2\_stndV1 = (LMDTOT\_V2p-mean(LMDTOT,na.rm=T))/sd(LMDTOT,na.rm=T)

LMDTOT\_V3\_stndV1 = (LMDTOT\_V3p-mean(LMDTOT,na.rm=T))/sd(LMDTOT,na.rm=T)

VRITOT\_V1\_stndV1 = scale(VRITOT)

VRITOT\_V2\_stndV1 = (VRITOT\_V2p-mean(VRITOT,na.rm=T))/sd(VRITOT,na.rm=T)

VRITOT\_V3\_stndV1 = (VRITOT\_V3p-mean(VRITOT,na.rm=T))/sd(VRITOT,na.rm=T)

VRDTOT\_V1\_stndV1 = scale(VRDTOT)

VRDTOT\_V2\_stndV1 = (VRDTOT\_V2p-mean(VRDTOT,na.rm=T))/sd(VRDTOT,na.rm=T)

VRDTOT\_V3\_stndV1 = (VRDTOT\_V3p-mean(VRDTOT,na.rm=T))/sd(VRDTOT,na.rm=T)

VRCTOT\_V1\_stndV1 = scale(VRCTOT)

VRCTOT\_V2\_stndV1 = (VRCTOT\_V2p-mean(VRCTOT,na.rm=T))/sd(VRCTOT,na.rm=T)

VRCTOT\_V3\_stndV1 = (VRCTOT\_V3p-mean(VRCTOT,na.rm=T))/sd(VRCTOT,na.rm=T)

CVA1RAW\_V1\_stndV1 = scale(CVA1RAW)

CVA1RAW\_V2\_stndV1 = (CVA1RAW\_V2p-mean(CVA1RAW,na.rm=T))/sd(CVA1RAW,na.rm=T)

CVA1RAW\_V3\_stndV1 = (CVA1RAW\_V3p-mean(CVA1RAW,na.rm=T))/sd(CVA1RAW,na.rm=T)

CVATOT\_V1\_stndV1 = scale(CVATOT)

CVATOT\_V2\_stndV1 = (CVATOT\_V2p-mean(CVATOT,na.rm=T))/sd(CVATOT,na.rm=T)

CVATOT\_V3\_stndV1 = (CVATOT\_V3p-mean(CVATOT,na.rm=T))/sd(CVATOT,na.rm=T)

CVSDFR\_V1\_stndV1 = scale(CVSDFR)

CVSDFR\_V2\_stndV1 = (CVSDFR\_V2p-mean(CVSDFR,na.rm=T))/sd(CVSDFR,na.rm=T)

CVSDFR\_V3\_stndV1 = (CVSDFR\_V3p-mean(CVSDFR,na.rm=T))/sd(CVSDFR,na.rm=T)

CVLDFR\_V1\_stndV1 = scale(CVLDFR)

CVLDFR\_V2\_stndV1 = (CVLDFR\_V2p-mean(CVLDFR,na.rm=T))/sd(CVLDFR,na.rm=T)

CVLDFR\_V3\_stndV1 = (CVLDFR\_V3p-mean(CVLDFR,na.rm=T))/sd(CVLDFR,na.rm=T)

## EF and Speed

StroopCW\_V1\_stndV1 = scale(STRCWRAW)

StroopCW\_V2\_stndV1=(STRCWRAW\_V2p-mean(STRCWRAW,na.rm=T))/sd(STRCWRAW,na.rm=T)

StroopCW\_V3\_stndV1=(STRCWRAW\_V3p-mean(STRCWRAW,na.rm=T))/sd(STRCWRAW,na.rm=T)

StroopC\_V1\_stndV1 = scale(STRCRAW)

StroopC\_V2\_stndV1=(STRCRAW\_V2p-mean(STRCRAW,na.rm=T))/sd(STRCRAW,na.rm=T)

StroopC\_V3\_stndV1=(STRCRAW\_V3p-mean(STRCRAW,na.rm=T))/sd(STRCRAW,na.rm=T)

StroopW\_V1\_stndV1 = scale(STRWRAW)

StroopW\_V2\_stndV1=(STRWRAW\_V2p-mean(STRWRAW,na.rm=T))/sd(STRWRAW,na.rm=T)

StroopW\_V3\_stndV1=(STRWRAW\_V3p-mean(STRWRAW,na.rm=T))/sd(STRWRAW,na.rm=T)

LNseq\_V1\_stndV1=scale(LNTOT)

LNseq\_V2\_stndV1=(LNTOT\_V2p-mean(LNTOT,na.rm=T))/sd(LNTOT,na.rm=T)

LNseq\_V3\_stndV1=(LNTOT\_V3p-mean(LNTOT,na.rm=T))/sd(LNTOT,na.rm=T)

RSasc\_V1\_stndV1=scale(RSATOT)

RSasc\_V2\_stndV1=(RSATOT\_V2p-mean(RSATOT,na.rm=T))/sd(RSATOT,na.rm=T)

RSasc\_V3\_stndV1=(RSATOT\_V3p-mean(RSATOT,na.rm=T))/sd(RSATOT,na.rm=T)

DStot\_V1\_stndV1=scale(DSTOT)

DStot\_V2\_stndV1=(DSTOT\_V2p-mean(DSTOT,na.rm=T))/sd(DSTOT,na.rm=T)

DStot\_V3\_stndV1=(DSTOT\_V3p-mean(DSTOT,na.rm=T))/sd(DSTOT,na.rm=T)

SStot\_V1\_stndV1 = scale(SSPTOTP)

SStot\_V2\_stndV1 = (SSPTOTP\_V2p-mean(SSPTOTP,na.rm=T))/sd(SSPTOTP,na.rm=T)

SStot\_V3\_stndV1 = (SSPTOTP\_V3p-mean(SSPTOTP,na.rm=T))/sd(SSPTOTP,na.rm=T)

TRL2rt <- exp(TRL2TLOG)

TRL3rt <- exp(TRL3TLOG)

TRL4rt <- exp(TRL4TLOG)

TRL2rt\_V2p <- exp(TRL2TLOG\_V2p)

TRL3rt\_V2p <- exp(TRL3TLOG\_V2p)

TRL4rt\_V2p <- exp(TRL4TLOG\_V2p)

TRL2rt\_V3p <- exp(TRL2TLOG\_V3p)

TRL3rt\_V3p <- exp(TRL3TLOG\_V3p)

TRL4rt\_V3p <- exp(TRL4TLOG\_V3p)

Trail4\_V1\_stndV1=scale(TRL4rt)

Trail4\_V2\_stndV1=(TRL4rt\_V2p-mean(TRL4rt,na.rm=T))/sd(TRL4rt,na.rm=T)

Trail4\_V3\_stndV1=(TRL4rt\_V3p-mean(TRL4rt,na.rm=T))/sd(TRL4rt,na.rm=T)

Trail3\_V1\_stndV1=scale(TRL3rt)

Trail3\_V2\_stndV1=(TRL3rt\_V2p-mean(TRL3rt,na.rm=T))/sd(TRL3rt,na.rm=T)

Trail3\_V3\_stndV1=(TRL3rt\_V3p-mean(TRL3rt,na.rm=T))/sd(TRL3rt,na.rm=T)

Trail2\_V1\_stndV1=scale(TRL2rt)

Trail2\_V2\_stndV1=(TRL2rt\_V2p-mean(TRL2rt,na.rm=T))/sd(TRL2rt,na.rm=T)

Trail2\_V3\_stndV1=(TRL2rt\_V3p-mean(TRL2rt,na.rm=T))/sd(TRL2rt,na.rm=T)

AXHITRATE\_V1\_stndV1 = scale(AXHITRATE)

AXHITRATE\_V2\_stndV1 = (AXHITRATE\_V2p-mean(AXHITRATE,na.rm=T))/sd(AXHITRATE,na.rm=T)

BXFARATE\_V2\_stndV1=scale(BXFARATE)

BXFARATE\_V2\_stndV1 = (BXFARATE\_V2p-mean(BXFARATE,na.rm=T))/sd(BXFARATE,na.rm=T)

SRTGMEANLOG\_V1\_stndV1 = scale(SRTGMEANLOG)

SRTGMEANLOG\_V2\_stndV1 = (SRTGMEANLOG\_V2p-mean(SRTGMEANLOG,na.rm=T))/sd(SRTGMEANLOG,na.rm=T)

SRTGMEANLOG\_V3\_stndV1 = (SRTGMEANLOG\_V3p-mean(SRTGMEANLOG,na.rm=T))/sd(SRTGMEANLOG,na.rm=T)

CHRTGMEANLOG\_V1\_stndV1 = scale(CHRTGMEANLOG)

CHRTGMEANLOG\_V2\_stndV1 = (CHRTGMEANLOG\_V2p-mean(CHRTGMEANLOG,na.rm=T))/sd(CHRTGMEANLOG,na.rm=T)

CHRTGMEANLOG\_V3\_stndV1 = (CHRTGMEANLOG\_V3p-mean(CHRTGMEANLOG,na.rm=T))/sd(CHRTGMEANLOG,na.rm=T)

## Fluency

LFFCOR\_V1\_stndV1 = scale(LFFCOR)

LFFCOR\_V2\_stndV1 = (LFFCOR\_V2p-mean(LFFCOR,na.rm=T))/sd(LFFCOR,na.rm=T)

LFFCOR\_V3\_stndV1 = (LFFCOR\_V3p-mean(LFFCOR,na.rm=T))/sd(LFFCOR,na.rm=T)

LFACOR\_V1\_stndV1 = scale(LFACOR)

LFACOR\_V2\_stndV1 = (LFACOR\_V2p-mean(LFACOR,na.rm=T))/sd(LFACOR,na.rm=T)

LFACOR\_V3\_stndV1 = (LFACOR\_V3p-mean(LFACOR,na.rm=T))/sd(LFACOR,na.rm=T)

LFSCOR\_V1\_stndV1 = scale(LFSCOR)

LFSCOR\_V2\_stndV1 = (LFSCOR\_V2p-mean(LFSCOR,na.rm=T))/sd(LFSCOR,na.rm=T)

LFSCOR\_V3\_stndV1 = (LFSCOR\_V3p-mean(LFSCOR,na.rm=T))/sd(LFSCOR,na.rm=T)

LFCOR\_V1\_stndV1 = scale(LFCOR)

LFCOR\_V2\_stndV1 = (LFCOR\_V2p-mean(LFCOR,na.rm=T))/sd(LFCOR,na.rm=T)

LFCOR\_V3\_stndV1 = (LFCOR\_V3p-mean(LFCOR,na.rm=T))/sd(LFCOR,na.rm=T)

CFANCOR\_V1\_stndV1 = scale(CFANCOR)

CFANCOR\_V2\_stndV1 = (CFANCOR\_V2p-mean(CFANCOR,na.rm=T))/sd(CFANCOR,na.rm=T)

CFANCOR\_V3\_stndV1 = (CFANCOR\_V3p-mean(CFANCOR,na.rm=T))/sd(CFANCOR,na.rm=T)

CFBNCOR\_V1\_stndV1 = scale(CFBNCOR)

CFBNCOR\_V2\_stndV1 = (CFBNCOR\_V2p-mean(CFBNCOR,na.rm=T))/sd(CFBNCOR,na.rm=T)

CFBNCOR\_V3\_stndV1 = (CFBNCOR\_V3p-mean(CFBNCOR,na.rm=T))/sd(CFBNCOR,na.rm=T)

CSCOR\_V1\_stndV1 = scale(CSCOR)

CSCOR\_V2\_stndV1 = (CSCOR\_V2p-mean(CSCOR,na.rm=T))/sd(CSCOR,na.rm=T)

CSCOR\_V3\_stndV1 = (CSCOR\_V3p-mean(CSCOR,na.rm=T))/sd(CSCOR,na.rm=T)

CatFluency\_V1\_stndV1=scale(CFCOR)

CatFluency\_V2\_stndV1=(CFCOR\_V2p-mean(CFCOR,na.rm=T))/sd(CFCOR,na.rm=T)

CatFluency\_V3\_stndV1=(CFCOR\_V3p-mean(CFCOR,na.rm=T))/sd(CFCOR,na.rm=T)

CatSwAcc\_V1\_stndV1=scale(CSSACC)

CatSwAcc\_V2\_stndV1=(CSSACC\_V2p-mean(CSSACC,na.rm=T))/sd(CSSACC,na.rm=T)

CatSwAcc\_V3\_stndV1=(CSSACC\_V3p-mean(CSSACC,na.rm=T))/sd(CSSACC,na.rm=T)

#Visuospatial Other

HFTOTCOR\_V1\_stndV1 = scale(HFTOTCOR)

HFTOTCOR\_V2\_stndV1 = (HFTOTCOR\_V2p-mean(HFTOTCOR,na.rm=T))/sd(HFTOTCOR,na.rm=T)

HFTOTCOR\_V3\_stndV1 = (HFTOTCOR\_V3p-mean(HFTOTCOR,na.rm=T))/sd(HFTOTCOR,na.rm=T)

MR1COR\_V1\_stndV1 = scale(MR1COR)

MR1COR\_V2\_stndV1 = (MR1COR\_V2p-mean(MR1COR,na.rm=T))/sd(MR1COR,na.rm=T)

MR1COR\_V3\_stndV1 = (MR1COR\_V3p-mean(MR1COR,na.rm=T))/sd(MR1COR,na.rm=T)

MTXRAW\_V1\_stndV1 = scale(MTXRAW)

MTXRAW\_V2\_stndV1 = (MTXRAW\_V2p-mean(MTXRAW,na.rm=T))/sd(MTXRAW,na.rm=T)

MTXRAW\_V3\_stndV1 = (MTXRAW\_V3p-mean(MTXRAW,na.rm=T))/sd(MTXRAW,na.rm=T)

## VisSpat (including AFQT Box)

MR1COR\_V1\_stndV1 = scale(MR1COR)

MR1COR\_V2\_stndV1 = (MR1COR\_V2p-mean(MR1COR,na.rm=T))/sd(MR1COR,na.rm=T)

MR1COR\_V3\_stndV1 = (MR1COR\_V3p-mean(MR1COR,na.rm=T))/sd(MR1COR,na.rm=T)

AFQTBXPCTTRAN\_R\_V1\_stndV1 = scale(AFQTBXPCTTRAN\_R\_V1)

AFQTBXPCTTRAN\_R\_V2\_stndV1 = (AFQTBXPCTTRAN\_R\_V2p-mean(AFQTBXPCTTRAN\_R\_V1,na.rm=T))/sd(AFQTBXPCTTRAN\_R\_V1,na.rm=T)

AFQTBXPCTTRAN\_R\_V3\_stndV1 = (AFQTBXPCTTRAN\_R\_V3p-mean(AFQTBXPCTTRAN\_R\_V1,na.rm=T))/sd(AFQTBXPCTTRAN\_R\_V1,na.rm=T)

HFTOTCOR\_V1\_stndV1 = scale(HFTOTCOR)

HFTOTCOR\_V2\_stndV1 = (HFTOTCOR\_V2p-mean(HFTOTCOR,na.rm=T))/sd(HFTOTCOR,na.rm=T)

HFTOTCOR\_V3\_stndV1 = (HFTOTCOR\_V3p-mean(HFTOTCOR,na.rm=T))/sd(HFTOTCOR,na.rm=T)

## VisMem

sVRCTOT = sqrt(VRCTOT)

sVRCTOT\_V2p = sqrt(VRCTOT\_V2p)

sVRCTOT\_V3p = sqrt(VRCTOT\_V3p)

sVRCTOT\_V1\_stndV1 = scale(sVRCTOT)

sVRCTOT\_V2\_stndV1 = (sVRCTOT\_V2p-mean(sVRCTOT,na.rm=T))/sd(sVRCTOT,na.rm=T)

sVRCTOT\_V3\_stndV1 = (sVRCTOT\_V3p-mean(sVRCTOT,na.rm=T))/sd(sVRCTOT,na.rm=T)

VRITOT\_V1\_stndV1 = scale(VRITOT)

VRITOT\_V2\_stndV1 = (VRITOT\_V2p-mean(VRITOT,na.rm=T))/sd(VRITOT,na.rm=T)

VRITOT\_V3\_stndV1 = (VRITOT\_V3p-mean(VRITOT,na.rm=T))/sd(VRITOT,na.rm=T)

VRDTOT\_V1\_stndV1 = scale(VRDTOT)

VRDTOT\_V2\_stndV1 = (VRDTOT\_V2p-mean(VRDTOT,na.rm=T))/sd(VRDTOT,na.rm=T)

VRDTOT\_V3\_stndV1 = (VRDTOT\_V3p-mean(VRDTOT,na.rm=T))/sd(VRDTOT,na.rm=T)

############################################

### Create interference scores for EF ###

############################################

#

library(lme4)

Stroop\_V1\_lm <- lmer(STRCWRAW~STRCRAW+STRWRAW + (1|CASE), na.action=na.exclude) # Regress EF condition on baseline conditions (controlling for case)

Stroop\_V1 <- resid(Stroop\_V1\_lm)+mean(STRCWRAW, na.rm=T) # save residuals into new variable after adding mean back in

Stroop\_V1z <- scale(Stroop\_V1)

Stroop\_V2\_lm <- lmer(STRCWRAW\_V2p~STRCRAW\_V2p+STRWRAW\_V2p + (1|CASE), na.action=na.exclude)

Stroop\_V2 <- (resid(Stroop\_V2\_lm)-mean(STRCWRAW, na.rm=T)+mean(STRCWRAW\_V2p, na.rm=T))/sd(Stroop\_V1, na.rm=T)

Stroop\_V3\_lm <- lmer(STRCWRAW\_V3p~STRCRAW\_V3p+STRWRAW\_V3p + (1|CASE), na.action=na.exclude)

Stroop\_V3 <- (resid(Stroop\_V3\_lm)-mean(STRCWRAW, na.rm=T)+mean(STRCWRAW\_V3p, na.rm=T))/sd(Stroop\_V1, na.rm=T)

Trail\_V1\_lm <- lmer(TRL4rt~TRL3rt+TRL2rt + (1|CASE), na.action=na.exclude)

Trail\_V1 <- resid(Trail\_V1\_lm)+mean(TRL4rt, na.rm=T)

Trail\_V1z <- scale(Trail\_V1)

Trail\_V2\_lm <- lmer(TRL4rt\_V2p~TRL3rt\_V2p+TRL2rt\_V2p + (1|CASE), na.action=na.exclude)

Trail\_V2 <- (resid(Trail\_V2\_lm)-mean(TRL4rt, na.rm=T)+mean(TRL4rt\_V2p, na.rm=T))/sd(TRL4rt, na.rm=T)

Trail\_V3\_lm <- lmer(TRL4rt\_V3p~TRL3rt\_V3p+TRL2rt\_V3p + (1|CASE), na.action=na.exclude)

Trail\_V3 <- (resid(Trail\_V3\_lm)-mean(TRL4rt, na.rm=T)+mean(TRL4rt\_V3p, na.rm=T))/sd(TRL4rt, na.rm=T)

CatSw\_V1\_lm <- lmer(CSSACC~CFANCOR+CFBNCOR + (1|CASE), na.action=na.exclude)

CatSw\_V1 <- resid(CatSw\_V1\_lm)+mean(TRL4rt, na.rm=T)

CatSw\_V1z <- scale(CatSw\_V1)

CatSw\_V2\_lm <- lmer(CSSACC\_V2p~CFANCOR\_V2p+CFBNCOR\_V2p + (1|CASE), na.action=na.exclude)

CatSw\_V2 <- (resid(CatSw\_V2\_lm)-mean(CSSACC, na.rm=T)+mean(CSSACC\_V2p, na.rm=T))/sd(CSSACC, na.rm=T)

CatSw\_V3\_lm <- lmer(CSSACC\_V3p~CFANCOR\_V3p+CFBNCOR\_V3p + (1|CASE), na.action=na.exclude)

CatSw\_V3 <- (resid(CatSw\_V3\_lm)-mean(CSSACC, na.rm=T)+mean(CSSACC\_V3p, na.rm=T))/sd(CSSACC, na.rm=T)

##### AXCPT Dprime - Not in VETSA 3

#asinTransform <- function(p) { asin(sqrt(p)) }

#AXdprime\_V1\_step1 = AXHITRATE-BXFARATE

#AXdprime\_V1\_step2 = asinTransform(AXdprime\_V1\_step1)

#AXdprime\_V1\_step2[AXdprime\_V1\_step2<0] <- 0

#AXdprime\_V1 = scale(AXdprime\_V1\_step2)

#AXdprime\_V2\_step1 = AXHITRATE\_V2p-BXFARATE\_V2p

#AXdprime\_V2\_step2 = asinTransform(AXdprime\_V2\_step1)

#AXdprime\_V2\_step2[AXdprime\_V2\_step2<0] <- 0

#AXdprime\_V2 = (AXdprime\_V2\_step2-mean(AXdprime\_V1\_step2,na.rm=T))/sd(AXdprime\_V1\_step2, na.rm=T)

################################################

######## IMPUTE MISSING DATA #########

#### prep domains and exclude missing ####

################################################

#Create Datasets for imputation (dataset for each domain and wave, removing sjs missing too many tests, create dataset with and without ID var for MICE)

EF1x <- data.frame(VETSAID, Stroop\_V1z, Trail\_V1z, CatSw\_V1z, LNseq\_V1\_stndV1, RSasc\_V1\_stndV1, DStot\_V1\_stndV1)

EF2x <- data.frame(VETSAID, Stroop\_V2, Trail\_V2, CatSw\_V2, LNseq\_V2\_stndV1, RSasc\_V2\_stndV1, DStot\_V2\_stndV1)

EF3x <- data.frame(VETSAID, Stroop\_V3, Trail\_V3, CatSw\_V3, LNseq\_V3\_stndV1, RSasc\_V3\_stndV1, DStot\_V3\_stndV1)

EF1y <- EF1x[rowSums(is.na(EF1x))<4,] # up to 3 tests can be missing

EF2y <- EF2x[rowSums(is.na(EF2x))<4,]

EF3y <- EF3x[rowSums(is.na(EF3x))<4,]

EF1 <- EF1y[,2:7]

EF2 <- EF2y[,2:7]

EF3 <- EF3y[,2:7]

FLU1x <- data.frame(VETSAID, LFFCOR\_V1\_stndV1, LFACOR\_V1\_stndV1, LFSCOR\_V1\_stndV1, CFANCOR\_V1\_stndV1, CFBNCOR\_V1\_stndV1, CSCOR\_V1\_stndV1)

FLU2x <- data.frame(VETSAID, LFFCOR\_V2\_stndV1, LFACOR\_V2\_stndV1, LFSCOR\_V2\_stndV1, CFANCOR\_V2\_stndV1, CFBNCOR\_V2\_stndV1, CSCOR\_V2\_stndV1)

FLU3x <- data.frame(VETSAID, LFFCOR\_V3\_stndV1, LFACOR\_V3\_stndV1, LFSCOR\_V3\_stndV1, CFANCOR\_V3\_stndV1, CFBNCOR\_V3\_stndV1, CSCOR\_V3\_stndV1)

FLU1y <- FLU1x[rowSums(is.na(FLU1x))<4,] # up to 3 tests can be missing

FLU2y <- FLU2x[rowSums(is.na(FLU2x))<4,]

FLU3y <- FLU3x[rowSums(is.na(FLU3x))<4,]

FLU1 <- FLU1y[,2:7]

FLU2 <- FLU2y[,2:7]

FLU3 <- FLU3y[,2:7]

MEM1x <- data.frame(VETSAID, LMITOT\_V1\_stndV1, LMDTOT\_V1\_stndV1, VRITOT\_V1\_stndV1, VRDTOT\_V1\_stndV1, CVATOT\_V1\_stndV1, CVSDFR\_V1\_stndV1, CVLDFR\_V1\_stndV1)

MEM2x <- data.frame(VETSAID, LMITOT\_V2\_stndV1, LMDTOT\_V2\_stndV1, VRITOT\_V2\_stndV1, VRDTOT\_V2\_stndV1, CVATOT\_V2\_stndV1, CVSDFR\_V2\_stndV1, CVLDFR\_V2\_stndV1)

MEM3x <- data.frame(VETSAID, LMITOT\_V3\_stndV1, LMDTOT\_V3\_stndV1, VRITOT\_V3\_stndV1, VRDTOT\_V3\_stndV1, CVATOT\_V3\_stndV1, CVSDFR\_V3\_stndV1, CVLDFR\_V3\_stndV1)

MEM1y <- MEM1x[rowSums(is.na(MEM1x))<4,] # up to 3 tests can be missing

MEM2y <- MEM2x[rowSums(is.na(MEM2x))<4,]

MEM3y <- MEM3x[rowSums(is.na(MEM3x))<4,]

MEM1 <- MEM1y[,2:8]

MEM2 <- MEM2y[,2:8]

MEM3 <- MEM3y[,2:8]

SPD1x <- data.frame(VETSAID, Trail2\_V1\_stndV1, Trail3\_V1\_stndV1, StroopW\_V1\_stndV1, StroopC\_V1\_stndV1, SRTGMEANLOG\_V1\_stndV1, CHRTGMEANLOG\_V1\_stndV1)

SPD2x <- data.frame(VETSAID, Trail2\_V2\_stndV1, Trail3\_V2\_stndV1, StroopW\_V2\_stndV1, StroopC\_V2\_stndV1, SRTGMEANLOG\_V2\_stndV1, CHRTGMEANLOG\_V2\_stndV1)

SPD3x <- data.frame(VETSAID, Trail2\_V3\_stndV1, Trail3\_V3\_stndV1, StroopW\_V3\_stndV1, StroopC\_V3\_stndV1, SRTGMEANLOG\_V3\_stndV1, CHRTGMEANLOG\_V3\_stndV1)

SPD1y <- SPD1x[rowSums(is.na(SPD1x))<4,] # up to 3 tests can be missing

SPD2y <- SPD2x[rowSums(is.na(SPD2x))<4,]

SPD3y <- SPD3x[rowSums(is.na(SPD3x))<4,]

SPD1 <- SPD1y[,2:7]

SPD2 <- SPD2y[,2:7]

SPD3 <- SPD3y[,2:7]

WM1x <- data.frame(VETSAID, LNseq\_V1\_stndV1, RSasc\_V1\_stndV1, DStot\_V1\_stndV1)

WM2x <- data.frame(VETSAID, LNseq\_V2\_stndV1, RSasc\_V2\_stndV1, DStot\_V2\_stndV1)

WM3x <- data.frame(VETSAID, LNseq\_V3\_stndV1, RSasc\_V3\_stndV1, DStot\_V3\_stndV1)

WM1y <- WM1x[rowSums(is.na(WM1x))<2,] # up to 1 test can be missing

WM2y <- WM2x[rowSums(is.na(WM2x))<2,]

WM3y <- WM3x[rowSums(is.na(WM3x))<2,]

WM1 <- WM1y[,2:4]

WM2 <- WM2y[,2:4]

WM3 <- WM3y[,2:4]

SF1x <- data.frame(VETSAID, CFANCOR\_V1\_stndV1, CFBNCOR\_V1\_stndV1, CSCOR\_V1\_stndV1)

SF2x <- data.frame(VETSAID, CFANCOR\_V2\_stndV1, CFBNCOR\_V2\_stndV1, CSCOR\_V2\_stndV1)

SF3x <- data.frame(VETSAID, CFANCOR\_V3\_stndV1, CFBNCOR\_V3\_stndV1, CSCOR\_V3\_stndV1)

SF1y <- SF1x[rowSums(is.na(SF1x))<2,] # up to 1 test can be missing

SF2y <- SF2x[rowSums(is.na(SF2x))<2,]

SF3y <- SF3x[rowSums(is.na(SF3x))<2,]

SF1 <- SF1y[,2:4]

SF2 <- SF2y[,2:4]

SF3 <- SF3y[,2:4]

SPA1x <- data.frame(VETSAID, MR1COR\_V1\_stndV1, AFQTBXPCTTRAN\_R\_V1\_stndV1, HFTOTCOR\_V1\_stndV1)

SPA2x <- data.frame(VETSAID, MR1COR\_V2\_stndV1, AFQTBXPCTTRAN\_R\_V2\_stndV1, HFTOTCOR\_V2\_stndV1)

SPA3x <- data.frame(VETSAID, MR1COR\_V3\_stndV1, AFQTBXPCTTRAN\_R\_V3\_stndV1, HFTOTCOR\_V3\_stndV1)

SPA1y <- SPA1x[rowSums(is.na(SPA1x))<2,] # up to 1 tests can be missing

SPA2y <- SPA2x[rowSums(is.na(SPA2x))<2,]

SPA3y <- SPA3x[rowSums(is.na(SPA3x))<2,]

SPA1 <- SPA1y[,2:4]

SPA2 <- SPA2y[,2:4]

SPA3 <- SPA3y[,2:4]

VMEM1x <- data.frame(VETSAID, sVRCTOT\_V1\_stndV1, VRITOT\_V1\_stndV1, VRDTOT\_V1\_stndV1)

VMEM2x <- data.frame(VETSAID, sVRCTOT\_V2\_stndV1, VRITOT\_V2\_stndV1, VRDTOT\_V2\_stndV1)

VMEM3x <- data.frame(VETSAID, sVRCTOT\_V3\_stndV1, VRITOT\_V3\_stndV1, VRDTOT\_V3\_stndV1)

VMEM1y <- VMEM1x[rowSums(is.na(VMEM1x))<2,] # up to 1 tests can be missing

VMEM2y <- VMEM2x[rowSums(is.na(VMEM2x))<2,]

VMEM3y <- VMEM3x[rowSums(is.na(VMEM3x))<2,]

VMEM1 <- VMEM1y[,2:4]

VMEM2 <- VMEM2y[,2:4]

VMEM3 <- VMEM3y[,2:4]

###################################

#### Actually run imputation ######

###################################

library(mice)

FLU1\_imp <- mice(FLU1,m=5) # imputing the variables, 5 times

FLU2\_imp <- mice(FLU2,m=5)

FLU3\_imp <- mice(FLU3,m=5)

EF1\_imp <- mice(EF1,m=5)

EF2\_imp <- mice(EF2,m=5)

EF3\_imp <- mice(EF3,m=5)

MEM1\_imp <- mice(MEM1,m=5)

MEM2\_imp <- mice(MEM2,m=5)

MEM3\_imp <- mice(MEM3,m=5)

SPD1\_imp <- mice(SPD1,m=5)

SPD2\_imp <- mice(SPD2,m=5)

SPD3\_imp <- mice(SPD3,m=5)

WM1\_imp <- mice(WM1,m=5)

WM2\_imp <- mice(WM2,m=5)

WM3\_imp <- mice(WM3,m=5)

SF1\_imp <- mice(SF1,m=5)

SF2\_imp <- mice(SF2,m=5)

SF3\_imp <- mice(SF3,m=5)

SPA1\_imp <- mice(SPA1,m=5) # imputing the variables, 5 times

SPA2\_imp <- mice(SPA2,m=5)

SPA3\_imp <- mice(SPA3,m=5)

VMEM1\_imp <- mice(VMEM1,m=5)

VMEM2\_imp <- mice(VMEM2,m=5)

VMEM3\_imp <- mice(VMEM3,m=5)

#data2 <- complete(imp,"long") #creating complete data set from imputed variables, stacked by imputation

FLU1\_data2 <- complete(FLU1\_imp,"long") #creating complete data set from imputed variables, stacked by imputation

FLU2\_data2 <- complete(FLU2\_imp,"long")

FLU3\_data2 <- complete(FLU3\_imp,"long")

EF1\_data2 <- complete(EF1\_imp,"long")

EF2\_data2 <- complete(EF2\_imp,"long")

EF3\_data2 <- complete(EF3\_imp,"long")

MEM1\_data2 <- complete(MEM1\_imp,"long")

MEM2\_data2 <- complete(MEM2\_imp,"long")

MEM3\_data2 <- complete(MEM3\_imp,"long")

SPD1\_data2 <- complete(SPD1\_imp,"long")

SPD2\_data2 <- complete(SPD2\_imp,"long")

SPD3\_data2 <- complete(SPD3\_imp,"long")

WM1\_data2 <- complete(WM1\_imp,"long")

WM2\_data2 <- complete(WM2\_imp,"long")

WM3\_data2 <- complete(WM3\_imp,"long")

SF1\_data2 <- complete(SF1\_imp,"long")

SF2\_data2 <- complete(SF2\_imp,"long")

SF3\_data2 <- complete(SF3\_imp,"long")

SPA1\_data2 <- complete(SPA1\_imp,"long") #creating complete data set from imputed variables, stacked by imputation

SPA2\_data2 <- complete(SPA2\_imp,"long")

SPA3\_data2 <- complete(SPA3\_imp,"long")

VMEM1\_data2 <- complete(VMEM1\_imp,"long")

VMEM2\_data2 <- complete(VMEM2\_imp,"long")

VMEM3\_data2 <- complete(VMEM3\_imp,"long")

dataimp <- FLU1 # temporary dataset with imputed data

for(i in 1:dim(FLU1y)[1]){dataimp[i,] <- colMeans(FLU1\_data2[FLU1\_data2$.id==i,3:8])} # average values across 5 imputations

FLU1\_Imputed <- cbind(FLU1y[,1],dataimp) # combine with original list of IDs

names(FLU1\_Imputed)[1] <- "VETSAID" # rename first column

# Checking Summaries post imputation, there should be no missing data

#summary(FLU1\_Imputed)

#names(FLU1\_Imputed)

#write.csv(V3\_Imp, "V3\_Imp.csv")

dataimp <- FLU2

for(i in 1:dim(FLU2y)[1]){dataimp[i,] <- colMeans(FLU2\_data2[FLU2\_data2$.id==i,3:8])} # averaging values across 5 imputations

FLU2\_Imputed <- cbind(FLU2y[,1],dataimp)

names(FLU2\_Imputed)[1] <- "VETSAID"

dataimp <- FLU3

for(i in 1:dim(FLU3y)[1]){dataimp[i,] <- colMeans(FLU3\_data2[FLU3\_data2$.id==i,3:8])} # averaging values across 5 imputations

FLU3\_Imputed <- cbind(FLU3y[,1],dataimp)

names(FLU3\_Imputed)[1] <- "VETSAID"

dataimp <- EF1

for(i in 1:dim(EF1y)[1]){dataimp[i,] <- colMeans(EF1\_data2[EF1\_data2$.id==i,3:8])} # averaging values across 5 imputations

EF1\_Imputed <- cbind(EF1y[,1],dataimp)

names(EF1\_Imputed)[1] <- "VETSAID"

dataimp <- EF2

for(i in 1:dim(EF2y)[1]){dataimp[i,] <- colMeans(EF2\_data2[EF2\_data2$.id==i,3:8])} # averaging values across 5 imputations

EF2\_Imputed <- cbind(EF2y[,1],dataimp)

names(EF2\_Imputed)[1] <- "VETSAID"

dataimp <- EF3

for(i in 1:dim(EF3y)[1]){dataimp[i,] <- colMeans(EF3\_data2[EF3\_data2$.id==i,3:8])} # averaging values across 5 imputations

EF3\_Imputed <- cbind(EF3y[,1],dataimp)

names(EF3\_Imputed)[1] <- "VETSAID"

dataimp <- MEM1

for(i in 1:dim(MEM1y)[1]){dataimp[i,] <- colMeans(MEM1\_data2[MEM1\_data2$.id==i,3:9])} # averaging values across 5 imputations

MEM1\_Imputed <- cbind(MEM1y[,1],dataimp)

names(MEM1\_Imputed)[1] <- "VETSAID"

dataimp <- MEM2

for(i in 1:dim(MEM2y)[1]){dataimp[i,] <- colMeans(MEM2\_data2[MEM2\_data2$.id==i,3:9])} # averaging values across 5 imputations

MEM2\_Imputed <- cbind(MEM2y[,1],dataimp)

names(MEM2\_Imputed)[1] <- "VETSAID"

dataimp <- MEM3

for(i in 1:dim(MEM3y)[1]){dataimp[i,] <- colMeans(MEM3\_data2[MEM3\_data2$.id==i,3:9])} # averaging values across 5 imputations

MEM3\_Imputed <- cbind(MEM3y[,1],dataimp)

names(MEM3\_Imputed)[1] <- "VETSAID"

dataimp <- SPD1

for(i in 1:dim(SPD1y)[1]){dataimp[i,] <- colMeans(SPD1\_data2[SPD1\_data2$.id==i,3:8])} # averaging values across 5 imputations

SPD1\_Imputed <- cbind(SPD1y[,1],dataimp)

names(SPD1\_Imputed)[1] <- "VETSAID"

dataimp <- SPD2

for(i in 1:dim(SPD2y)[1]){dataimp[i,] <- colMeans(SPD2\_data2[SPD2\_data2$.id==i,3:8])} # averaging values across 5 imputations

SPD2\_Imputed <- cbind(SPD2y[,1],dataimp)

names(SPD2\_Imputed)[1] <- "VETSAID"

dataimp <- SPD3

for(i in 1:dim(SPD3y)[1]){dataimp[i,] <- colMeans(SPD3\_data2[SPD3\_data2$.id==i,3:8])} # averaging values across 5 imputations

SPD3\_Imputed <- cbind(SPD3y[,1],dataimp)

names(SPD3\_Imputed)[1] <- "VETSAID"

dataimp <- WM1

for(i in 1:dim(WM1y)[1]){dataimp[i,] <- colMeans(WM1\_data2[WM1\_data2$.id==i,3:5])} # averaging values across 5 imputations

WM1\_Imputed <- cbind(WM1y[,1],dataimp)

names(WM1\_Imputed) <- c("VETSAID","LN1\_wm","RS1\_wm","DS1\_wm") ## rename variables so they don't conflict with EF domain

dataimp <- WM2

for(i in 1:dim(WM2y)[1]){dataimp[i,] <- colMeans(WM2\_data2[WM2\_data2$.id==i,3:5])} # averaging values across 5 imputations

WM2\_Imputed <- cbind(WM2y[,1],dataimp)

names(WM2\_Imputed) <- c("VETSAID","LN2\_wm","RS2\_wm","DS2\_wm") ## rename variables so they don't conflict with EF domain

dataimp <- WM3

for(i in 1:dim(WM3y)[1]){dataimp[i,] <- colMeans(WM3\_data2[WM3\_data2$.id==i,3:5])} # averaging values across 5 imputations

WM3\_Imputed <- cbind(WM3y[,1],dataimp)

names(WM3\_Imputed) <- c("VETSAID","LN3\_wm","RS3\_wm","DS3\_wm") ## rename variables so they don't conflict with EF domain

dataimp <- SF1

for(i in 1:dim(SF1y)[1]){dataimp[i,] <- colMeans(SF1\_data2[SF1\_data2$.id==i,3:5])} # averaging values across 5 imputations

SF1\_Imputed <- cbind(SF1y[,1],dataimp)

names(SF1)

names(SF1\_Imputed) <- c("VETSAID","AN1\_sf","BN1\_sf","CS1\_sf") ## rename variables so they don't conflict with fluency domain

dataimp <- SF2

for(i in 1:dim(SF2y)[1]){dataimp[i,] <- colMeans(SF2\_data2[SF2\_data2$.id==i,3:5])} # averaging values across 5 imputations

SF2\_Imputed <- cbind(SF2y[,1],dataimp)

names(SF2\_Imputed) <- c("VETSAID","AN2\_sf","BN2\_sf","CS2\_sf") ## rename variables so they don't conflict with fluency domain

dataimp <- SF3

for(i in 1:dim(SF3y)[1]){dataimp[i,] <- colMeans(SF3\_data2[SF3\_data2$.id==i,3:5])} # averaging values across 5 imputations

SF3\_Imputed <- cbind(SF3y[,1],dataimp)

names(SF3\_Imputed) <- c("VETSAID","AN3\_sf","BN3\_sf","CS3\_sf")

dataimp <- SPA1

for(i in 1:dim(SPA1y)[1]){dataimp[i,] <- colMeans(SPA1\_data2[SPA1\_data2$.id==i,3:5])} # averaging values across 5 imputations

SPA1\_Imputed <- cbind(SPA1y[,1],dataimp)

names(SPA1\_Imputed)[1] <- "VETSAID"

dataimp <- SPA2

for(i in 1:dim(SPA2y)[1]){dataimp[i,] <- colMeans(SPA2\_data2[SPA2\_data2$.id==i,3:5])} # averaging values across 5 imputations

SPA2\_Imputed <- cbind(SPA2y[,1],dataimp)

names(SPA2\_Imputed)[1] <- "VETSAID"

dataimp <- SPA3

for(i in 1:dim(SPA3y)[1]){dataimp[i,] <- colMeans(SPA3\_data2[SPA3\_data2$.id==i,3:5])} # averaging values across 5 imputations

SPA3\_Imputed <- cbind(SPA3y[,1],dataimp)

names(SPA3\_Imputed)[1] <- "VETSAID"

dataimp <- VMEM1

for(i in 1:dim(VMEM1y)[1]){dataimp[i,] <- colMeans(VMEM1\_data2[VMEM1\_data2$.id==i,3:5])} # averaging values across 5 imputations

VMEM1\_Imputed <- cbind(VMEM1y[,1],dataimp)

names(VMEM1\_Imputed)[1] <- "VETSAID"

dataimp <- VMEM2

for(i in 1:dim(VMEM2y)[1]){dataimp[i,] <- colMeans(VMEM2\_data2[VMEM2\_data2$.id==i,3:5])} # averaging values across 5 imputations

VMEM2\_Imputed <- cbind(VMEM2y[,1],dataimp)

names(VMEM2\_Imputed)[1] <- "VETSAID"

dataimp <- VMEM3

for(i in 1:dim(VMEM3y)[1]){dataimp[i,] <- colMeans(VMEM3\_data2[VMEM3\_data2$.id==i,3:5])} # averaging values across 5 imputations

VMEM3\_Imputed <- cbind(VMEM3y[,1],dataimp)

names(VMEM3\_Imputed)[1] <- "VETSAID"

###########################################

##### Create Factor Scores ######

###########################################

##############################

##### NO IMPUTATION ##########

##############################

# Sum across standardized factor loadings from latent variable models, then re-scale all factor scores based on M/SD of wave 1 factor scores

commonEF\_V1x <- .31\*Stroop\_V1z+ (-1)\*.55\*Trail\_V1z+ .22\*CatSw\_V1z+ .49\*LNseq\_V1\_stndV1 + .38\*RSasc\_V1\_stndV1 + .44\*DStot\_V1\_stndV1

commonEF\_V2x <- .31\*Stroop\_V2 + (-1)\*.55\*Trail\_V2 + .22\*CatSw\_V2 + .49\*LNseq\_V2\_stndV1 + .38\*RSasc\_V2\_stndV1 + .44\*DStot\_V2\_stndV1

commonEF\_V3x <- .31\*Stroop\_V3 + (-1)\*.55\*Trail\_V3 + .22\*CatSw\_V3 + .49\*LNseq\_V3\_stndV1 + .38\*RSasc\_V3\_stndV1 + .44\*DStot\_V3\_stndV1

commonEF\_V1 <- scale(commonEF\_V1x)

commonEF\_V2 <- (commonEF\_V2x-mean(commonEF\_V1x,na.rm=T))/sd(commonEF\_V1x,na.rm=T)

commonEF\_V3 <- (commonEF\_V3x-mean(commonEF\_V1x,na.rm=T))/sd(commonEF\_V1x,na.rm=T)

WM\_V1x <- .705\*LNseq\_V1\_stndV1 + .606\*RSasc\_V1\_stndV1 + .828\*DStot\_V1\_stndV1

WM\_V2x <- .705\*LNseq\_V2\_stndV1 + .606\*RSasc\_V2\_stndV1 + .828\*DStot\_V2\_stndV1

WM\_V3x <- .705\*LNseq\_V3\_stndV1 + .606\*RSasc\_V3\_stndV1 + .828\*DStot\_V3\_stndV1

work\_mem\_V1 <- scale(WM\_V1x)

work\_mem\_V2 <- (WM\_V2x-mean(WM\_V1x,na.rm=T))/sd(WM\_V1x,na.rm=T)

work\_mem\_V3 <- (WM\_V3x-mean(WM\_V1x,na.rm=T))/sd(WM\_V1x,na.rm=T)

fluency\_V1x <- .77\*LFFCOR\_V1\_stndV1 + .78\*LFACOR\_V1\_stndV1 + .80\*LFSCOR\_V1\_stndV1 + .43\*CFANCOR\_V1\_stndV1 + .54\*CFBNCOR\_V1\_stndV1 + .36\*CSCOR\_V1\_stndV1

fluency\_V2x <- .77\*LFFCOR\_V2\_stndV1 + .78\*LFACOR\_V2\_stndV1 + .80\*LFSCOR\_V2\_stndV1 + .43\*CFANCOR\_V2\_stndV1 + .54\*CFBNCOR\_V2\_stndV1 + .36\*CSCOR\_V2\_stndV1

fluency\_V3x <- .77\*LFFCOR\_V3\_stndV1 + .78\*LFACOR\_V3\_stndV1 + .80\*LFSCOR\_V3\_stndV1 + .43\*CFANCOR\_V3\_stndV1 + .54\*CFBNCOR\_V3\_stndV1 + .36\*CSCOR\_V3\_stndV1

fluency\_V1 <- scale(fluency\_V1x)

fluency\_V2 <- (fluency\_V2x-mean(fluency\_V1x,na.rm=T))/sd(fluency\_V1x,na.rm=T)

fluency\_V3 <- (fluency\_V3x-mean(fluency\_V1x,na.rm=T))/sd(fluency\_V1x,na.rm=T)

semantic\_flu\_V1x <- .685\*CFANCOR\_V1\_stndV1 + .623\*CFBNCOR\_V1\_stndV1 + .516\*CSCOR\_V1\_stndV1

semantic\_flu\_V2x <- .685\*CFANCOR\_V2\_stndV1 + .623\*CFBNCOR\_V2\_stndV1 + .516\*CSCOR\_V2\_stndV1

semantic\_flu\_V3x <- .685\*CFANCOR\_V3\_stndV1 + .623\*CFBNCOR\_V3\_stndV1 + .516\*CSCOR\_V3\_stndV1

semantic\_flu\_V1 <- scale(semantic\_flu\_V1x)

semantic\_flu\_V2 <- (semantic\_flu\_V2x-mean(semantic\_flu\_V1x,na.rm=T))/sd(semantic\_flu\_V1x,na.rm=T)

semantic\_flu\_V3 <- (semantic\_flu\_V3x-mean(semantic\_flu\_V1x,na.rm=T))/sd(semantic\_flu\_V1x,na.rm=T)

# Original Version without CVLT learning trials

#memory\_V1x <- .87\*.57\*LMITOT\_V1\_stndV1 + .97\*.57\*LMDTOT\_V1\_stndV1 + .64\*.47\*VRITOT\_V1\_stndV1 + .88\*.47\*VRDTOT\_V1\_stndV1 + .83\*.68\*CVSDFR\_V1\_stndV1 + .96\*.68\*CVLDFR\_V1\_stndV1

#memory\_V2x <- .87\*.57\*LMITOT\_V2\_stndV1 + .97\*.57\*LMDTOT\_V2\_stndV1 + .64\*.47\*VRITOT\_V2\_stndV1 + .88\*.47\*VRDTOT\_V2\_stndV1 + .83\*.68\*CVSDFR\_V2\_stndV1 + .96\*.68\*CVLDFR\_V2\_stndV1

#memory\_V3x <- .87\*.57\*LMITOT\_V3\_stndV1 + .97\*.57\*LMDTOT\_V3\_stndV1 + .64\*.47\*VRITOT\_V3\_stndV1 + .88\*.47\*VRDTOT\_V3\_stndV1 + .83\*.68\*CVSDFR\_V3\_stndV1 + .96\*.68\*CVLDFR\_V3\_stndV1

#memory\_V1 <- scale(memory\_V1x)

#memory\_V2 <- (memory\_V2x-mean(memory\_V1x,na.rm=T))/sd(memory\_V1x,na.rm=T)

#memory\_V3 <- (memory\_V3x-mean(memory\_V1x,na.rm=T))/sd(memory\_V1x,na.rm=T)

memory\_V1x <- .91\*.66\*LMITOT\_V1\_stndV1 + .94\*.66\*LMDTOT\_V1\_stndV1 + .75\*.60\*VRITOT\_V1\_stndV1 + .85\*.60\*VRDTOT\_V1\_stndV1 + .18\*CVATOT\_V1\_stndV1 + .68\*.72\*CVATOT\_V1\_stndV1 + .90\*.72\*CVSDFR\_V1\_stndV1 + .91\*.72\*CVLDFR\_V1\_stndV1

memory\_V2x <- .91\*.66\*LMITOT\_V2\_stndV1 + .94\*.66\*LMDTOT\_V2\_stndV1 + .75\*.60\*VRITOT\_V2\_stndV1 + .85\*.60\*VRDTOT\_V2\_stndV1 + .18\*CVATOT\_V2\_stndV1 + .68\*.72\*CVATOT\_V2\_stndV1 + .90\*.72\*CVSDFR\_V2\_stndV1 + .91\*.72\*CVLDFR\_V2\_stndV1

memory\_V3x <- .91\*.66\*LMITOT\_V3\_stndV1 + .94\*.66\*LMDTOT\_V3\_stndV1 + .75\*.60\*VRITOT\_V3\_stndV1 + .85\*.60\*VRDTOT\_V3\_stndV1 + .18\*CVATOT\_V3\_stndV1 + .68\*.72\*CVATOT\_V3\_stndV1 + .90\*.72\*CVSDFR\_V3\_stndV1 + .91\*.72\*CVLDFR\_V3\_stndV1

memory\_V1 <- scale(memory\_V1x)

memory\_V2 <- (memory\_V2x-mean(memory\_V1x,na.rm=T))/sd(memory\_V1x,na.rm=T)

memory\_V3 <- (memory\_V3x-mean(memory\_V1x,na.rm=T))/sd(memory\_V1x,na.rm=T)

speed\_V1x <- (-1)\*.66\*.56\*Trail2\_V1\_stndV1 + (-1)\*.84\*.56\*Trail3\_V1\_stndV1 + .74\*.82\*StroopW\_V1\_stndV1 + .83\*.82\*StroopC\_V1\_stndV1 + (-1)\*.76\*.54\*SRTGMEANLOG\_V1\_stndV1 + (-1)\*.86\*.54\*CHRTGMEANLOG\_V1\_stndV1

speed\_V2x <- (-1)\*.66\*.56\*Trail2\_V2\_stndV1 + (-1)\*.84\*.56\*Trail3\_V2\_stndV1 + .74\*.82\*StroopW\_V2\_stndV1 + .83\*.82\*StroopC\_V2\_stndV1 + (-1)\*.76\*.54\*SRTGMEANLOG\_V2\_stndV1 + (-1)\*.86\*.54\*CHRTGMEANLOG\_V2\_stndV1

speed\_V3x <- (-1)\*.66\*.56\*Trail2\_V3\_stndV1 + (-1)\*.84\*.56\*Trail3\_V3\_stndV1 + .74\*.82\*StroopW\_V3\_stndV1 + .83\*.82\*StroopC\_V3\_stndV1 + (-1)\*.76\*.54\*SRTGMEANLOG\_V3\_stndV1 + (-1)\*.86\*.54\*CHRTGMEANLOG\_V3\_stndV1

speed\_V1 <- scale(speed\_V1x)

speed\_V2 <- (speed\_V2x-mean(speed\_V1x,na.rm=T))/sd(speed\_V1x,na.rm=T)

speed\_V3 <- (speed\_V3x-mean(speed\_V1x,na.rm=T))/sd(speed\_V1x,na.rm=T)

#Standardized Scoring Coefficients from Proc Factor n=1 priors=smc method=ML (CR)

SPA\_V1x <- 0.16715\*MR1COR\_V1\_stndV1 + 0.27901\*AFQTBXPCTTRAN\_R\_V1\_stndV1 + 0.59952\*HFTOTCOR\_V1\_stndV1

SPA\_V2x <- 0.16715\*MR1COR\_V2\_stndV1 + 0.27901\*AFQTBXPCTTRAN\_R\_V2\_stndV1 + 0.59952\*HFTOTCOR\_V2\_stndV1

SPA\_V3x <- 0.16715\*MR1COR\_V3\_stndV1 + 0.27901\*AFQTBXPCTTRAN\_R\_V3\_stndV1 + 0.59952\*HFTOTCOR\_V3\_stndV1

VisSpat\_V1 <- scale(SPA\_V1x)

VisSpat\_V2 <- (SPA\_V2x-mean(SPA\_V1x,na.rm=T))/sd(SPA\_V1x,na.rm=T)

VisSpat\_V3 <- (SPA\_V3x-mean(SPA\_V1x,na.rm=T))/sd(SPA\_V1x,na.rm=T)

VMEM\_V1x <- 0.05350\*sVRCTOT\_V1\_stndV1 + 0.80128\*VRITOT\_V1\_stndV1 + 0.16498\*VRDTOT\_V1\_stndV1

VMEM\_V2x <- 0.05350\*sVRCTOT\_V2\_stndV1 + 0.80128\*VRITOT\_V2\_stndV1 + 0.16498\*VRDTOT\_V2\_stndV1

VMEM\_V3x <- 0.05350\*sVRCTOT\_V3\_stndV1 + 0.80128\*VRITOT\_V3\_stndV1 + 0.16498\*VRDTOT\_V3\_stndV1

VisMem\_V1 <- scale(VMEM\_V1x)

VisMem\_V2 <- (VMEM\_V2x-mean(VMEM\_V1x,na.rm=T))/sd(VMEM\_V1x,na.rm=T)

VisMem\_V3 <- (VMEM\_V3x-mean(VMEM\_V1x,na.rm=T))/sd(VMEM\_V1x,na.rm=T)

###################################################

##### Factor Scores from Imputed Data ##########

###################################################

library(dplyr)

#merge imputed datasets

Part1FLU <- merge(merge(FLU1\_Imputed, FLU2\_Imputed, by = "VETSAID", all=T),

FLU3\_Imputed, by = "VETSAID", all=T)

Part2EF <- merge(merge(merge(Part1FLU, EF1\_Imputed, by = "VETSAID", all=T),

EF2\_Imputed, by = "VETSAID", all=T),

EF3\_Imputed, by = "VETSAID", all=T)

Part3MEM <- merge(merge(merge(Part2EF, MEM1\_Imputed, by = "VETSAID", all=T),

MEM2\_Imputed, by = "VETSAID", all=T),

MEM3\_Imputed, by = "VETSAID", all=T)

Part4SPD <- merge(merge(merge(Part3MEM, SPD1\_Imputed, by = "VETSAID", all=T),

SPD2\_Imputed, by = "VETSAID", all=T),

SPD3\_Imputed, by = "VETSAID", all=T)

Part5WM <- merge(merge(merge(Part4SPD, WM1\_Imputed, by = "VETSAID", all=T),

WM2\_Imputed, by = "VETSAID", all=T),

WM3\_Imputed, by = "VETSAID", all=T)

Part6SF <- merge(merge(merge(Part5WM, SF1\_Imputed, by = "VETSAID", all=T),

SF2\_Imputed, by = "VETSAID", all=T),

SF3\_Imputed, by = "VETSAID", all=T)

Part7SPA <- merge(merge(merge(Part6SF, SPA1\_Imputed, by = "VETSAID", all=T),

SPA2\_Imputed, by = "VETSAID", all=T),

SPA3\_Imputed, by = "VETSAID", all=T)

Part8VMEM <- merge(merge(merge(Part7SPA, VMEM1\_Imputed, by = "VETSAID", all=T),

VMEM2\_Imputed, by = "VETSAID", all=T),

VMEM3\_Imputed, by = "VETSAID", all=T)

IDs <- data.frame(VETSAID, VETSAGRP)

all\_Imputed <- merge(Part8VMEM, IDs, by = "VETSAID", all=T)

# Remove duplicated variables and rename

all\_Imputed <- all\_Imputed %>% select(-ends\_with(".y"))

names(all\_Imputed) = gsub(".x","",names(all\_Imputed))

commonEF\_V1x\_imputed <- .31\*all\_Imputed$Stroop\_V1z+ (-1)\*.55\*all\_Imputed$Trail\_V1z+ .22\*all\_Imputed$CatSw\_V1z+ .49\*all\_Imputed$LNseq\_V1\_stndV1 + .38\*all\_Imputed$RSasc\_V1\_stndV1 + .44\*all\_Imputed$DStot\_V1\_stndV1

commonEF\_V2x\_imputed <- .31\*all\_Imputed$Stroop\_V2 + (-1)\*.55\*all\_Imputed$Trail\_V2 + .22\*all\_Imputed$CatSw\_V2 + .49\*all\_Imputed$LNseq\_V2\_stndV1 + .38\*all\_Imputed$RSasc\_V2\_stndV1 + .44\*all\_Imputed$DStot\_V2\_stndV1

commonEF\_V3x\_imputed <- .31\*all\_Imputed$Stroop\_V3 + (-1)\*.55\*all\_Imputed$Trail\_V3 + .22\*all\_Imputed$CatSw\_V3 + .49\*all\_Imputed$LNseq\_V3\_stndV1 + .38\*all\_Imputed$RSasc\_V3\_stndV1 + .44\*all\_Imputed$DStot\_V3\_stndV1

commonEF\_V1\_imputed <- scale(commonEF\_V1x\_imputed)

commonEF\_V2\_imputed <- (commonEF\_V2x\_imputed-mean(commonEF\_V1x\_imputed,na.rm=T))/sd(commonEF\_V1x\_imputed,na.rm=T)

commonEF\_V3\_imputed <- (commonEF\_V3x\_imputed-mean(commonEF\_V1x\_imputed,na.rm=T))/sd(commonEF\_V1x\_imputed,na.rm=T)

WM\_V1x\_imputed <- .705\*all\_Imputed$LN1\_wm + .606\*all\_Imputed$RS1\_wm + .828\*all\_Imputed$DS1\_wm

WM\_V2x\_imputed <- .705\*all\_Imputed$LN2\_wm + .606\*all\_Imputed$RS2\_wm + .828\*all\_Imputed$DS2\_wm

WM\_V3x\_imputed <- .705\*all\_Imputed$LN3\_wm + .606\*all\_Imputed$RS3\_wm + .828\*all\_Imputed$DS3\_wm

work\_mem\_V1\_imputed <- scale(WM\_V1x\_imputed)

work\_mem\_V2\_imputed <- (WM\_V2x\_imputed-mean(WM\_V1x\_imputed,na.rm=T))/sd(WM\_V1x\_imputed,na.rm=T)

work\_mem\_V3\_imputed <- (WM\_V3x\_imputed-mean(WM\_V1x\_imputed,na.rm=T))/sd(WM\_V1x\_imputed,na.rm=T)

fluency\_V1x\_imputed <- .77\*all\_Imputed$LFFCOR\_V1\_stndV1 + .78\*all\_Imputed$LFACOR\_V1\_stndV1 + .80\*all\_Imputed$LFSCOR\_V1\_stndV1 + .43\*all\_Imputed$CFANCOR\_V1\_stndV1 + .54\*all\_Imputed$CFBNCOR\_V1\_stndV1 + .36\*all\_Imputed$CSCOR\_V1\_stndV1

fluency\_V2x\_imputed <- .77\*all\_Imputed$LFFCOR\_V2\_stndV1 + .78\*all\_Imputed$LFACOR\_V2\_stndV1 + .80\*all\_Imputed$LFSCOR\_V2\_stndV1 + .43\*all\_Imputed$CFANCOR\_V2\_stndV1 + .54\*all\_Imputed$CFBNCOR\_V2\_stndV1 + .36\*all\_Imputed$CSCOR\_V2\_stndV1

fluency\_V3x\_imputed <- .77\*all\_Imputed$LFFCOR\_V3\_stndV1 + .78\*all\_Imputed$LFACOR\_V3\_stndV1 + .80\*all\_Imputed$LFSCOR\_V3\_stndV1 + .43\*all\_Imputed$CFANCOR\_V3\_stndV1 + .54\*all\_Imputed$CFBNCOR\_V3\_stndV1 + .36\*all\_Imputed$CSCOR\_V3\_stndV1

fluency\_V1\_imputed <- scale(fluency\_V1x\_imputed)

fluency\_V2\_imputed <- (fluency\_V2x\_imputed-mean(fluency\_V1x\_imputed,na.rm=T))/sd(fluency\_V1x\_imputed,na.rm=T)

fluency\_V3\_imputed <- (fluency\_V3x\_imputed-mean(fluency\_V1x\_imputed,na.rm=T))/sd(fluency\_V1x\_imputed,na.rm=T)

semantic\_flu\_V1x\_imputed <- .685\*all\_Imputed$AN1\_sf + .623\*all\_Imputed$BN1\_sf + .516\*all\_Imputed$CS1\_sf

semantic\_flu\_V2x\_imputed <- .685\*all\_Imputed$AN2\_sf + .623\*all\_Imputed$BN2\_sf + .516\*all\_Imputed$CS2\_sf

semantic\_flu\_V3x\_imputed <- .685\*all\_Imputed$AN3\_sf + .623\*all\_Imputed$BN3\_sf + .516\*all\_Imputed$CS3\_sf

semantic\_flu\_V1\_imputed <- scale(semantic\_flu\_V1x\_imputed)

semantic\_flu\_V2\_imputed <- (semantic\_flu\_V2x\_imputed-mean(semantic\_flu\_V1x\_imputed,na.rm=T))/sd(semantic\_flu\_V1x\_imputed,na.rm=T)

semantic\_flu\_V3\_imputed <- (semantic\_flu\_V3x\_imputed-mean(semantic\_flu\_V1x\_imputed,na.rm=T))/sd(semantic\_flu\_V1x\_imputed,na.rm=T)

memory\_V1x\_imputed <- .91\*.66\*all\_Imputed$LMITOT\_V1\_stndV1 + .94\*.66\*all\_Imputed$LMDTOT\_V1\_stndV1 + .75\*.60\*all\_Imputed$VRITOT\_V1\_stndV1 + .85\*.60\*all\_Imputed$VRDTOT\_V1\_stndV1 + .18\*all\_Imputed$CVATOT\_V1\_stndV1 + .68\*.72\*all\_Imputed$CVATOT\_V1\_stndV1 + .90\*.72\*all\_Imputed$CVSDFR\_V1\_stndV1 + .91\*.72\*all\_Imputed$CVLDFR\_V1\_stndV1

memory\_V2x\_imputed <- .91\*.66\*all\_Imputed$LMITOT\_V2\_stndV1 + .94\*.66\*all\_Imputed$LMDTOT\_V2\_stndV1 + .75\*.60\*all\_Imputed$VRITOT\_V2\_stndV1 + .85\*.60\*all\_Imputed$VRDTOT\_V2\_stndV1 + .18\*all\_Imputed$CVATOT\_V2\_stndV1 + .68\*.72\*all\_Imputed$CVATOT\_V2\_stndV1 + .90\*.72\*all\_Imputed$CVSDFR\_V2\_stndV1 + .91\*.72\*all\_Imputed$CVLDFR\_V2\_stndV1

memory\_V3x\_imputed <- .91\*.66\*all\_Imputed$LMITOT\_V3\_stndV1 + .94\*.66\*all\_Imputed$LMDTOT\_V3\_stndV1 + .75\*.60\*all\_Imputed$VRITOT\_V3\_stndV1 + .85\*.60\*all\_Imputed$VRDTOT\_V3\_stndV1 + .18\*all\_Imputed$CVATOT\_V3\_stndV1 + .68\*.72\*all\_Imputed$CVATOT\_V3\_stndV1 + .90\*.72\*all\_Imputed$CVSDFR\_V3\_stndV1 + .91\*.72\*all\_Imputed$CVLDFR\_V3\_stndV1

memory\_V1\_imputed <- scale(memory\_V1x\_imputed)

memory\_V2\_imputed <- (memory\_V2x\_imputed-mean(memory\_V1x\_imputed,na.rm=T))/sd(memory\_V1x\_imputed,na.rm=T)

memory\_V3\_imputed <- (memory\_V3x\_imputed-mean(memory\_V1x\_imputed,na.rm=T))/sd(memory\_V1x\_imputed,na.rm=T)

speed\_V1x\_imputed <- (-1)\*.66\*.56\*all\_Imputed$Trail2\_V1\_stndV1 + (-1)\*.84\*.56\*all\_Imputed$Trail3\_V1\_stndV1 + .74\*.82\*all\_Imputed$StroopW\_V1\_stndV1 + .83\*.82\*all\_Imputed$StroopC\_V1\_stndV1 + (-1)\*.76\*.54\*all\_Imputed$SRTGMEANLOG\_V1\_stndV1 + (-1)\*.86\*.54\*all\_Imputed$CHRTGMEANLOG\_V1\_stndV1

speed\_V2x\_imputed <- (-1)\*.66\*.56\*all\_Imputed$Trail2\_V2\_stndV1 + (-1)\*.84\*.56\*all\_Imputed$Trail3\_V2\_stndV1 + .74\*.82\*all\_Imputed$StroopW\_V2\_stndV1 + .83\*.82\*all\_Imputed$StroopC\_V2\_stndV1 + (-1)\*.76\*.54\*all\_Imputed$SRTGMEANLOG\_V2\_stndV1 + (-1)\*.86\*.54\*all\_Imputed$CHRTGMEANLOG\_V2\_stndV1

speed\_V3x\_imputed <- (-1)\*.66\*.56\*all\_Imputed$Trail2\_V3\_stndV1 + (-1)\*.84\*.56\*all\_Imputed$Trail3\_V3\_stndV1 + .74\*.82\*all\_Imputed$StroopW\_V3\_stndV1 + .83\*.82\*all\_Imputed$StroopC\_V3\_stndV1 + (-1)\*.76\*.54\*all\_Imputed$SRTGMEANLOG\_V3\_stndV1 + (-1)\*.86\*.54\*all\_Imputed$CHRTGMEANLOG\_V3\_stndV1

speed\_V1\_imputed <- scale(speed\_V1x\_imputed)

speed\_V2\_imputed <- (speed\_V2x\_imputed-mean(speed\_V1x\_imputed,na.rm=T))/sd(speed\_V1x\_imputed,na.rm=T)

speed\_V3\_imputed <- (speed\_V3x\_imputed-mean(speed\_V1x\_imputed,na.rm=T))/sd(speed\_V1x\_imputed,na.rm=T)

SPA\_V1x\_imputed <- 0.16715\*all\_Imputed$MR1COR\_V1\_stndV1 + 0.27901\*all\_Imputed$AFQTBXPCTTRAN\_R\_V1\_stndV1 + 0.59952\*all\_Imputed$HFTOTCOR\_V1\_stndV1

SPA\_V2x\_imputed <- 0.16715\*all\_Imputed$MR1COR\_V2\_stndV1 + 0.27901\*all\_Imputed$AFQTBXPCTTRAN\_R\_V2\_stndV1 + 0.59952\*all\_Imputed$HFTOTCOR\_V2\_stndV1

SPA\_V3x\_imputed <- 0.16715\*all\_Imputed$MR1COR\_V3\_stndV1 + 0.27901\*all\_Imputed$AFQTBXPCTTRAN\_R\_V3\_stndV1 + 0.59952\*all\_Imputed$HFTOTCOR\_V3\_stndV1

VisSpat\_V1\_imputed <- scale(SPA\_V1x\_imputed)

VisSpat\_V2\_imputed <- (SPA\_V2x\_imputed-mean(SPA\_V1x\_imputed,na.rm=T))/sd(SPA\_V1x\_imputed,na.rm=T)

VisSpat\_V3\_imputed <- (SPA\_V3x\_imputed-mean(SPA\_V1x\_imputed,na.rm=T))/sd(SPA\_V1x\_imputed,na.rm=T)

VMEM\_V1x\_imputed <- 0.05350\*all\_Imputed$sVRCTOT\_V1\_stndV1 + 0.80128\*all\_Imputed$VRITOT\_V1\_stndV1 + 0.16498\*all\_Imputed$VRDTOT\_V1\_stndV1

VMEM\_V2x\_imputed <- 0.05350\*all\_Imputed$sVRCTOT\_V2\_stndV1 + 0.80128\*all\_Imputed$VRITOT\_V2\_stndV1 + 0.16498\*all\_Imputed$VRDTOT\_V2\_stndV1

VMEM\_V3x\_imputed <- 0.05350\*all\_Imputed$sVRCTOT\_V3\_stndV1 + 0.80128\*all\_Imputed$VRITOT\_V3\_stndV1 + 0.16498\*all\_Imputed$VRDTOT\_V3\_stndV1

VisMem\_V1\_imputed <- scale(VMEM\_V1x\_imputed)

VisMem\_V2\_imputed <- (VMEM\_V2x\_imputed-mean(VMEM\_V1x\_imputed,na.rm=T))/sd(VMEM\_V1x\_imputed,na.rm=T)

VisMem\_V3\_imputed <- (VMEM\_V3x\_imputed-mean(VMEM\_V1x\_imputed,na.rm=T))/sd(VMEM\_V1x\_imputed,na.rm=T)

############################

### Export Scores to csv ###

############################

## non imputed

allScores <- data.frame(VETSAID, VETSAGRP,

memory\_V1, memory\_V2, memory\_V3,

speed\_V1, speed\_V2, speed\_V3,

commonEF\_V1, commonEF\_V2, commonEF\_V3,

work\_mem\_V1, work\_mem\_V2, work\_mem\_V3,

fluency\_V1, fluency\_V2, fluency\_V3,

semantic\_flu\_V1, semantic\_flu\_V2, semantic\_flu\_V3,

VisSpat\_V1, VisSpat\_V2, VisSpat\_V3,

VisMem\_V1,VisMem\_V2,VisMem\_V3)

describe(allScores)

write.csv(allScores, "~/netshare/M/NAS VETSA MASTER DATAFILES/Other cognitive measures/Cognitive Factor Scores V1V2V3/V1V2V3\_CogData\_FactorScores\_7\_22\_2021.csv", row.names = FALSE)

#head(allScores)

## imputed

allScores\_imp <- data.frame(VETSAID, VETSAGRP,

memory\_V1\_imputed, memory\_V2\_imputed, memory\_V3\_imputed,

speed\_V1\_imputed, speed\_V2\_imputed, speed\_V3\_imputed,

commonEF\_V1\_imputed, commonEF\_V2\_imputed, commonEF\_V3\_imputed,

work\_mem\_V1\_imputed, work\_mem\_V2\_imputed, work\_mem\_V3\_imputed,

fluency\_V1\_imputed, fluency\_V2\_imputed, fluency\_V3\_imputed,

semantic\_flu\_V1\_imputed, semantic\_flu\_V2\_imputed, semantic\_flu\_V3\_imputed,

VisSpat\_V1\_imputed, VisSpat\_V2\_imputed, VisSpat\_V3\_imputed,

VisMem\_V1\_imputed,VisMem\_V2\_imputed,VisMem\_V3\_imputed)

impnames <- c("VETSAID", "VETSAGRP",

"memory\_V1", "memory\_V2", "memory\_V3",

"speed\_V1", "speed\_V2", "speed\_V3",

"commonEF\_V1", "commonEF\_V2", "commonEF\_V3",

"work\_mem\_V1", "work\_mem\_V2", "work\_mem\_V3",

"fluency\_V1", "fluency\_V2", "fluency\_V3",

"semantic\_flu\_V1", "semantic\_flu\_V2", "semantic\_flu\_V3",

"vis\_spat\_V1", "vis\_spat\_V2", "vis\_spat\_V3",

"vis\_mem\_V1","vis\_mem\_V2","vis\_mem\_V3")

names(allScores\_imp) <- impnames

describe(allScores\_imp)

describe(allScores)

### Exclude final WM and SS IDs that were missing too many subdomain tasks in full imputation (for EF/fluency)

# allScores\_imp$WorkMem\_V1[allScores\_imp$VETSAID %in% WM1\_exclude\_IDs]<- NA

# allScores\_imp$WorkMem\_V2[allScores\_imp$VETSAID %in% WM2\_exclude\_IDs]<- NA

# allScores\_imp$WorkMem\_V3[allScores\_imp$VETSAID %in% WM3\_exclude\_IDs]<- NA

# allScores\_imp$Semantic\_Flu\_V1[allScores\_imp$VETSAID %in% SS1\_exclude\_IDs]<- NA

# allScores\_imp$Semantic\_Flu\_V2[allScores\_imp$VETSAID %in% SS2\_exclude\_IDs]<- NA

# allScores\_imp$Semantic\_Flu\_V3[allScores\_imp$VETSAID %in% SS3\_exclude\_IDs]<- NA

#describe(allScores\_imp)

write.csv(allScores\_imp, "~/netshare/M/NAS VETSA MASTER DATAFILES/Other cognitive measures/Cognitive Factor Scores V1V2V3/V1V2V3\_CogData\_FactorScores\_Imputed\_7\_22\_2021.csv", row.names = FALSE)