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

# This script is perform Mild cognitive Impairment (MCI) #

# #

# Classification according to Petersen’s criteria and Winblad’s criteria. #

# #

# The data files are from Gothenburg H85-2015 Examination. #

# #

# All input files are based on Research questionnaires and were loaded from the respective files #

# #

# –Named hereinafter referred to as ‘formulars’ - as follows: #

# #

# Psykiatri, Kognition, Anhorig, ADL and Dementia diagnosis file to exclude dementias at baseline. #

# #

# A new education variable based on available longitudinal info 2000-2015 from participants and #

# #

# Their key informants was computed by Simona Sacuiu for the #

# #

# MCI class: H85-2015\_Kognition(2019-04-15)\_bearbetatSFS20190703-EDUC.SAV #

# #

# Script Documentation Start Date: 11-07-2019 #

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

########################################## Load Package into directory: For loop and apply functions

list.packages <- c('dplyr','devtools','sjmisc','digest','gmodels','haven','Hmisc',

'psych','purrr','pastecs','lubridate','installr')

options(warn=0) ## turn Off Warninga

for(lib in list.packages){ # load packages in library

library(lib, require, character.only = TRUE)

devtools::install\_github("strengejacke/sjmisc")

}

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

########################################################################## Set up the working directory

PATH <- "X:/Nazib/Documentations/Simona/MCI\_H85\_2015"

setwd(PATH)

################################################ Use getwd() to access the path to the working directory

####################################################### Load H85-2015 data files into Global Environment

H85.Formulars.path <- file.path(PATH, c("H85-2015\_Psykiatri(2018-12-17).SAV",

"H85-2015\_Kognition(2018-11-30).SAV",

"H85-2015\_ANH(2018-12-07).SAV",

"H85-2015\_ADL(2018-11-30).SAV",

"Demens\_H85-KVUS\_2015.sav",

"H85-2015\_Kognition(2019-04-15)\_bearbetatSFS20190703-EDUC.SAV"))

DataFrames <- lapply(H85.Formulars.path, read\_sav) ####### Load tuples of data frame for each formular

names(DataFrames) <- gsub(".\*/(.\*)\\..\*", "\\1", H85.Formulars.path) ### Unpack names of data frames

str(DataFrames, give.attr = FALSE) ### view the structure of d again

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

############ Data extraction for each formular and variables of interest in Global data frame (DataFrames)

############### Psychiatry Formular Variables:

psyk.data<- as.data.frame(select(DataFrames[[1]],LopNr,KOG1,KOG5,KOG6,KOG9,PSF8,F25DAT))

############### Kognition Formular Variables:

kog.data<- as.data.frame(select(DataFrames[[2]],LopNr,MIN1,MIN2,MIN3,MIN6,MIN7,MIN8,MIN9,MIN10,MIN11,

MIN12,MIN13,MIN14,MIN15,MIN16,MIN25,MIN32,MIN17,MIN18,MIN35,MIN37,MIN66,

MIN44I,MIN45,MIN46,MIN47,MIN48,MIN49,MIN50,DEM1,DEM2,DEM3,DEM4,DEM5,

STA22A,STA23A,STA24A))

############### Anhörig Formular Variables:

anh.data<- as.data.frame(select(DataFrames[[3]],LopNr,AHDATUM,PRA5B,INT4,INT6,INT7,INT10,

INT13,INT19,PRAK1,PRA5A,SPRA1,SPRA2,SPRA3,SPRA4,MIN1ANH,

MIN8ANH,MIN6ANH,MIN10ANH,MIN12ANH,MIN14ANH,MIN13ANH,MIN16ANH,

MIN18ANH,MIN19ANH,MIN20ANH,MIN21ANH,MIN5ANH))

############### ADL Formular Variables:

adl.data<- as.data.frame(select(DataFrames[[4]],LopNr,ADLTE,ADLTF,ADLTG,ADLTH,ADLTJ,ADLTA,

ADLTB,ADLTC,ADLTD,ADLTK,ADLTL,ADLTM,ADLTN,ADLTO))

############### H85-2015 Dementia Diagnosis variable retrieve from Hanna W. (Sample Size in file = 491)

dem.data<- as.data.frame(select(DataFrames[[5]],LopNr,Dem2015))

############### Simona Computed Education Variables Data Frame:

edu.data<- as.data.frame(select(DataFrames[[6]],LopNr,Sex,educ\_sfs))

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

######################### Merge all data into a DataFrame and Exclude all Dementia Cases ( DEM2015 != 1)

MCI.Data.Update<- data.frame(list(psyk.data, kog.data, anh.data, adl.data,dem.data,edu.data) %>%

reduce(left\_join, by = "LopNr") %>%

filter(Dem2015 != 1))

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

################ Replace character Observations with Numeric Inputs and Missing Observations as "Missing"

options(warn = -1)

MCI.Data.Update[]<-lapply(MCI.Data.Update[], as.numeric)

MCI.Data.Update<-MCI.Data.Update%>% replace(is.na(.), "Missing")

############### Subset the data frame to use in the end of the code to extract raw file and write to file

MCI.Data.Update.last<- data.frame(MCI.Data.Update)

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

############## Complaint of Defective Memory(corroborated by knowledgeable informant- optional): SRMemImp

MCI.Data.Update<-MCI.Data.Update%>%

mutate(SRMemImp=ifelse(KOG1=="Missing","Missing",ifelse( KOG1>2 ,"Yes","No")))

SRMemImp.KOG<- select(MCI.Data.Update, LopNr,SRMemImp)

######### Inspect SRMemImp Cross Table Result

CrossTable(MCI.Data.Update$SRMemImp)

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

#################################################################################### SRCogIMP Computation

KOG\_Cols<- select(MCI.Data.Update,KOG1,KOG5,KOG6,KOG9)%>% names()

MCI.Data.Update <- MCI.Data.Update %>%

mutate(SRCogIMP=ifelse(pmin(!!!rlang::syms(KOG\_Cols))=="Missing","Missing",

ifelse(pmin(!!!rlang::syms(KOG\_Cols)) =="Missing" & pmax(!!!rlang::syms(KOG\_Cols)) >2,"No",

ifelse(pmax(!!!rlang::syms(KOG\_Cols)) =="Missing" & pmin(!!!rlang::syms(KOG\_Cols)) <=2,"No",

ifelse(pmax(!!!rlang::syms(KOG\_Cols)) <=2 ,"No",

ifelse(pmax(!!!rlang::syms(KOG\_Cols)) =="Missing"|pmin(!!!rlang::syms(KOG\_Cols)) >2,"Yes",

ifelse(pmax(!!!rlang::syms(KOG\_Cols))>2,"Yes","No")))))))

SR\_Mem\_Data<- select(MCI.Data.Update, LopNr, SRCogIMP)

######## Inspect SRCogImp Cross Table Result

CrossTable(MCI.Data.Update$SRCogIMP)

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

##################################################### Data Cleaning in Specific Individuals and Variables

MCI.Data.Update$AHDATUM[MCI.Data.Update$LopNr=="11346"] <- 20170103

MCI.Data.Update$MIN14ANH[MCI.Data.Update$LopNr=="6248"] <- 4

MCI.Data.Update$PRA5B[MCI.Data.Update$LopNr=="6436"] <- 3

MCI.Data.Update$PRAK1[MCI.Data.Update$LopNr=="6693"] <- 3

MCI.Data.Update$MIN21ANH[MCI.Data.Update$LopNr=="6796"] <- 3

MCI.Data.Update$MIN21ANH[MCI.Data.Update$LopNr=="7626"] <- 3

MCI.Data.Update$PRA5B[MCI.Data.Update$LopNr=="7948"] <- 3

MCI.Data.Update$PRA5A[MCI.Data.Update$LopNr=="11094"] <- 3

MCI.Data.Update$MIN12ANH[MCI.Data.Update$MIN12ANH==9]<-"Missing"

MCI.Data.Update$MIN13ANH[MCI.Data.Update$MIN13ANH==9]<-"Missing"

MCI.Data.Update$MIN18ANH[MCI.Data.Update$MIN18ANH==9]<-"Missing"

MCI.Data.Update$MIN5ANH[MCI.Data.Update$MIN5ANH==9]<-"Missing"

MCI.Data.Update$MIN14ANH[MCI.Data.Update$MIN14ANH==9]<-"Missing"

MCI.Data.Update$PRA5B[MCI.Data.Update$PRA5B==9]<-"Missing"

MCI.Data.Update$PRA5A[MCI.Data.Update$PRA5A==9]<-"Missing"

MCI.Data.Update$PRAK1[MCI.Data.Update$PRAK1==9]<-"Missing"

MCI.Data.Update$SPRA4[MCI.Data.Update$SPRA4==0]<-3

MCI.Data.Update$INT4[MCI.Data.Update$LopNr=="6653"] <- "Missing"

MCI.Data.Update$INT4[MCI.Data.Update$LopNr=="6846"] <- 3

MCI.Data.Update$INT7[MCI.Data.Update$LopNr=="7874"] <- 3

MCI.Data.Update$INT10[MCI.Data.Update$LopNr=="6674"] <- 4

MCI.Data.Update$INT10[MCI.Data.Update$LopNr=="11086"] <- 4

MCI.Data.Update$INT13[MCI.Data.Update$LopNr=="10897"] <- 3

MCI.Data.Update$INT13[MCI.Data.Update$LopNr=="14882"] <- 3

MCI.Data.Update$INT13[MCI.Data.Update$LopNr=="7660"] <- 3

MCI.Data.Update$INT19[MCI.Data.Update$LopNr=="6226"] <- 3

MCI.Data.Update$INT19[MCI.Data.Update$LopNr=="6430"] <- 3

MCI.Data.Update$INT19[MCI.Data.Update$LopNr=="6889"] <- 3

MCI.Data.Update$MIN14ANH[MCI.Data.Update$LopNr=="6248"] <- 4

MCI.Data.Update$SPRA3[MCI.Data.Update$LopNr=="6229"] <- 3

MCI.Data.Update$PRA5B[MCI.Data.Update$LopNr=="6436"] <- 3

MCI.Data.Update$PRAK1[MCI.Data.Update$LopNr=="6693"] <- 3

MCI.Data.Update$MIN21ANH[MCI.Data.Update$LopNr=="6796"] <- 3

MCI.Data.Update$MIN21ANH[MCI.Data.Update$LopNr=="7626"] <- 3

MCI.Data.Update$PRA5B[MCI.Data.Update$LopNr=="7948"] <- 3

MCI.Data.Update$PRA5A[MCI.Data.Update$LopNr=="11094"] <- 3

MCI.Data.Update$INT4[MCI.Data.Update$INT4 == 9]<- "Missing"

MCI.Data.Update$INT6[MCI.Data.Update$INT6 == 9]<- "Missing"

MCI.Data.Update$INT7[MCI.Data.Update$INT7 == 9]<- "Missing"

MCI.Data.Update$INT10[MCI.Data.Update$INT10 == 9]<- "Missing"

MCI.Data.Update$INT19[MCI.Data.Update$INT19 == 9]<- "Missing"

MCI.Data.Update$SPRA1[MCI.Data.Update$SPRA1 == 9]<- "Missing"

MCI.Data.Update$SPRA2[MCI.Data.Update$SPRA2 == 9]<- "Missing"

MCI.Data.Update$SPRA3[MCI.Data.Update$SPRA3 == 9]<- "Missing"

MCI.Data.Update$SPRA4[MCI.Data.Update$SPRA4 == 9]<- "Missing"

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

######################################## Change dates (F25DAT and AHDATUM) into machine readible format

MCI.Data.Update$F25DAT<- ymd(MCI.Data.Update$F25DAT)

MCI.Data.Update$AHDATUM<- ymd(MCI.Data.Update$AHDATUM)

##### Convert Date to DayDiffPS\_KI between time points

MCI.Data.Update$DayDiffPS\_KI<-as.character(difftime(as.Date(MCI.Data.Update$AHDATUM),

as.Date(MCI.Data.Update$F25DAT),unit="days"))

##### Change Character to Numeric and Store in a data frame

MCI.Data.Update[]<-lapply(MCI.Data.Update[], as.numeric)

MCI.Data.Update<- data.frame(MCI.Data.Update)

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

################################# COMBINE SubKogImp and SR\_IQMemImp BASED ON DayDiffCat USING DEFINITION

# DayDiffPS\_KI grouping function

#0 ='Samma dag '

#1 ='1-183 dagar '

#2 ='184-365 dagar '

#3 ='365-1000 dagar'

#OTHER ='\*\*\*\*\*\*\*\*\*\*\*\*\*\*';

####################################################### For loop to compute DayDiffCat (cut-off variable)

for(i in 1:nrow(MCI.Data.Update)){ # DayDiffPS\_KI grouping function

if(!is.na(MCI.Data.Update$DayDiffPS\_KI)[i] <=0){

MCI.Data.Update$DayDiffCat[i]='Missing'}

else if(MCI.Data.Update$DayDiffPS\_KI[i]>= 1 && MCI.Data.Update$DayDiffPS\_KI[i]<184){

MCI.Data.Update$DayDiffCat[i]=1}

else if(MCI.Data.Update$DayDiffPS\_KI[i]>= 184 && MCI.Data.Update$DayDiffPS\_KI[i]<366){

MCI.Data.Update$DayDiffCat[i]=2}

else if(MCI.Data.Update$DayDiffPS\_KI[i]>= 365 && MCI.Data.Update$DayDiffPS\_KI[i]<1001){

MCI.Data.Update$DayDiffCat[i]=3}

else {MCI.Data.Update$DayDiffCat[i]=0}

}

##### Inspect Computed Day Difference Groups

CrossTable(MCI.Data.Update$DayDiffCat)

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

###################################################################### 16-items Memory IQCODE Computation

IQCODE.1.Variables<-c("MIN1ANH","MIN10ANH","MIN12ANH","MIN13ANH","MIN14ANH","MIN16ANH","MIN18ANH",

"MIN19ANH","MIN20ANH","MIN21ANH","MIN5ANH","MIN6ANH","MIN8ANH","PRAK1","PRA5A",

"PRA5B")

MCI.Data.Update<-MCI.Data.Update%>%

mutate(IQmemAvg=rep(((14\*3)+(2\*4))/length(IQCODE.1.Variables),nrow(MCI.Data.Update)),

IQmemIndiv=rowMeans(select(.,IQCODE.1.Variables),na.rm=TRUE),

IQmemBin=ifelse(IQmemIndiv>IQmemAvg, "Yes","No"))

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

######################## Weighing the effect of number of missing to compute IQCODEMEM-16 Item: 19-06-19

MCI.Data.Update$N\_Miss\_1 <- rowSums( is.na(MCI.Data.Update[,IQCODE.1.Variables]))

MCI.Data.Update <- row\_count(MCI.Data.Update, IQCODE.1.Variables, count = 4, append = TRUE)

MCI.Data.Update <- row\_count(MCI.Data.Update, IQCODE.1.Variables, count = 5, append = TRUE)

MCI.Data.Update<- mutate(MCI.Data.Update, IQmemBin\_2 = ifelse(rowcount > 2 | rowcount1>0,"Yes","No"))

MCI.Data.Update<- mutate(MCI.Data.Update, Miss\_1= 16 - N\_Miss\_1)

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

####### Computing IQCODESBIN accounting for number of Missing in each person data and final IQCODEBINMEM

for(i in 1:nrow(MCI.Data.Update)){

TotIQcode\_1 = length(IQCODE.1.Variables)

if(MCI.Data.Update$N\_Miss\_1[i] > 0){ ###25th percental

MCI.Data.Update$IQCodeBin\_Miss\_1[i]=(((TotIQcode\_1 - MCI.Data.Update$N\_Miss\_1[i] - 2)\*3)+(2\*4))/

(TotIQcode\_1 - MCI.Data.Update$N\_Miss\_1[i])}

else {MCI.Data.Update$IQCodeBin\_Miss\_1[i]=(((TotIQcode\_1 -MCI.Data.Update$N\_Miss\_1[i] - 2)\*3)+(2\*4))/

(TotIQcode\_1 - MCI.Data.Update$N\_Miss\_1[i])}

}

##### IQCODEMEM-16 ltems Mem Final

options(warn = -1)

MCI.Data.Update$IQmemIndiv[MCI.Data.Update$IQmemIndiv == "NaN"]<- 'NA'

MCI.Data.Update$IQCodeBin\_Miss\_1[MCI.Data.Update$IQCodeBin\_Miss\_1 == "Inf"]<- 'NA'

MCI.Data.Update<- mutate(MCI.Data.Update,

IQCODEBINMEM\_FINAL = ifelse(IQmemIndiv=="NA" & IQCodeBin\_Miss\_1 =='NA','Missing',

ifelse(IQmemIndiv > IQCodeBin\_Miss\_1, "Yes","No")))

##### Inspect IQCODEBINMEM FINAL

CrossTable(MCI.Data.Update$IQCODEBINMEM\_FINAL)

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

############################################################################ 26-items IQCODE Computation

# NOTE:variables MIN7 from ANHÖRIG FORMULÄR 2000  MIN18ANH and MIN15 from ANHÖRIG FORMULÄR 2000 MIN16ANH

IQCODE.2.Variables<-c("MIN1ANH","MIN5ANH","MIN6ANH","MIN8ANH","MIN10ANH","MIN12ANH","MIN13ANH","MIN14ANH",

"MIN16ANH","MIN18ANH","MIN19ANH","MIN20ANH","MIN21ANH","PRAK1","PRA5A","PRA5B",

"INT4","INT6","INT7","INT10","INT13","INT19","SPRA1","SPRA2" ,"SPRA3","SPRA4")

#MCI.Data.Update[]<-lapply(MCI.Data.Update[], as.numeric)

#MCI.Data.Update<- data.frame(MCI.Data.Update)

MCI.Data.Update<-MCI.Data.Update%>%

mutate(IQcodeAvg=rep(((24\*3)+(2\*4))/length(IQCODE.2.Variables),nrow(MCI.Data.Update)),

IQcodeIndiv=rowMeans(select(.,IQCODE.2.Variables),na.rm=TRUE),

IQcodeBin=ifelse(IQcodeIndiv>IQcodeAvg, "Yes","No"))

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

###################################### Weighing the effect of number of missing to compute IQCODE-26 Item:

##### Compute Number of Missing for Individual and counts

MCI.Data.Update<- select(MCI.Data.Update, -N\_Miss\_1,-rowcount,-rowcount1)

MCI.Data.Update$N\_Miss\_2 <- rowSums( is.na( MCI.Data.Update[,IQCODE.2.Variables]))

MCI.Data.Update <- row\_count(MCI.Data.Update, IQCODE.2.Variables, count = 4, append = TRUE)

MCI.Data.Update <- row\_count(MCI.Data.Update, IQCODE.2.Variables, count = 5, append = TRUE)

MCI.Data.Update<- mutate(MCI.Data.Update, IQcodeBin\_2 = ifelse(rowcount > 2 | rowcount1>0,"Yes","No"))

MCI.Data.Update<- mutate(MCI.Data.Update, Miss= 26 - N\_Miss\_2)

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

######## Function for final computation of IQCODE Binary taken into account the number of missing inputs

for(i in 1:nrow(MCI.Data.Update)){

TotIQcode <- length(IQCODE.2.Variables)

if(MCI.Data.Update$N\_Miss\_2[i] > 0){ ###25th percental

MCI.Data.Update$IQCodeBin\_Miss\_2[i]=(((TotIQcode - MCI.Data.Update$N\_Miss\_2[i] - 2)\*3)+(2\*4))/

(TotIQcode - MCI.Data.Update$N\_Miss\_2[i])}

else {MCI.Data.Update$IQCodeBin\_Miss\_2[i]=(((TotIQcode - MCI.Data.Update$N\_Miss\_2[i] - 2)\*3)+(2\*4))/

(TotIQcode - MCI.Data.Update$N\_Miss\_2[i])}

}

options(warn = -1)

MCI.Data.Update$IQcodeIndiv[MCI.Data.Update$IQcodeIndiv == "NaN"]<- 'NA'

MCI.Data.Update$IQCodeBin\_Miss\_2[MCI.Data.Update$IQCodeBin\_Miss\_2== "Inf"]<- 'NA'

MCI.Data.Update<- mutate(MCI.Data.Update, IQCODEBIN\_FINAL =ifelse(IQcodeIndiv=="NA"

& IQCodeBin\_Miss\_2 =='NA','Missing',ifelse(IQcodeIndiv > IQCodeBin\_Miss\_2, "Yes","No")))

##### Inspect IQCODEBIN FINAL

CrossTable(MCI.Data.Update$IQCODEBIN\_FINAL)

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

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

#colnames(MCI.Data.Update)

MCI.Data.Update<- select(MCI.Data.Update, -SRCogIMP,-SRMemImp)

MCI.Data.Update.j<- data.frame(list(MCI.Data.Update,SRMemImp.KOG,SR\_Mem\_Data) %>%

reduce(left\_join, by = "LopNr"))

MCI.Data.Update<-MCI.Data.Update.j%>%

mutate(SubKogImp =ifelse(DayDiffCat =="Missing",'Missing',

ifelse(DayDiffCat <3 & (SRCogIMP =="Yes" | IQCODEBIN\_FINAL =="Yes"),"Yes",

ifelse(DayDiffCat <3 & (SRCogIMP =="No" & IQCODEBIN\_FINAL =="Yes"),"Yes",

ifelse(DayDiffCat ==3 & SRCogIMP =="Yes"& IQCODEBIN\_FINAL =="No","Yes","No")))),

SR\_IQMemImp =ifelse(DayDiffCat == 'Missing','Missing',

ifelse(DayDiffCat <3 & (SRMemImp =="Yes" | IQCODEBINMEM\_FINAL =="Yes"),"Yes",

ifelse(DayDiffCat <3 & (SRMemImp =="No" & IQCODEBINMEM\_FINAL =="Yes"),"Yes",

ifelse(DayDiffCat >= 3 & SRMemImp =="Yes","Yes","No")))))

##### Updated information

MCI.Data.Update$SubKogImp <- ifelse(MCI.Data.Update$SubKogImp =='Missing',

MCI.Data.Update$SRCogIMP, MCI.Data.Update$SubKogImp)

MCI.Data.Update$SR\_IQMemImp <- ifelse(MCI.Data.Update$SR\_IQMemImp =='Missing',

MCI.Data.Update$SRMemImp, MCI.Data.Update$SR\_IQMemImp)

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

############################# NormalGenKog(Normal General Cognitive Function MMSE.Value of MIN18 >23)

### Adjust with both Handicap and zero: According to standardized Mini-Mental State Examination (MMSE)

############################################# DR. Molloy Weighted Approach for those who are Handicap.

MCI.Data.Update$MIN7[MCI.Data.Update$LopNr ==10382] <- 3

MCI.Data.Update$MIN87[MCI.Data.Update$LopNr ==10382] <- 19

options(warn = -1)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero<- 'NA'

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==11366] <- (28\*30)/(30-1)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==11083] <- (28 \*30)/(30-1)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==10098] <- (22\*30)/(30-3)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr == 7106] <- (28\*30)/(30-1)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==7067] <- (29\*30)/(30-1)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==6969] <- (27 \*30)/(30-2)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==6935] <-(28 \*30)/(30-1)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==6739] <- (22 \*30)/(30-1)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==7497] <-26

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==6527] <- (25 \*30)/(30-4)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==10931] <- (24\*30)/(30-5)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==7829] <- (25 \*30)/(30-5)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==6757] <- (15\*30)/(30-8)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==10038] <- (26 \*30)/(30-3)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==6899] <- (22\*30)/(30-2)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==7433] <- (23\*30)/(30-2)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$LopNr ==10382] <- (19 \*30)/(30-6)

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero<-

round(as.numeric(MCI.Data.Update$MIN\_adjust\_Handicap\_Zero), digits = 0)

MCI.Data.Update$MIN18<- as.numeric(as.character(MCI.Data.Update$MIN18))

MCI.Data.Update$MIN18[MCI.Data.Update$MIN18=="<NA>"]<-'M'

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[MCI.Data.Update$MIN\_adjust\_Handicap\_Zero=="<NA>"]<-'M'

MCI.Data.Update$MIN\_adjust\_Handicap\_Zero[is.na(MCI.Data.Update$MIN\_adjust\_Handicap\_Zero)]<-

MCI.Data.Update$MIN18[is.na(MCI.Data.Update$MIN\_adjust\_Handicap\_Zero)]

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

######################################################################## Final NormalGenKog Computation

MCI.Data.Update<- MCI.Data.Update %>%

mutate(NormalGenKog=ifelse(MIN\_adjust\_Handicap\_Zero > 23,"Yes","No"))

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

###################################################################### Abnormal Memory Function for Age

MCI.Data.Update$N\_Miss.abmp <-

rowSums(is.na(MCI.Data.Update[,c("MIN1","MIN2","MIN3","MIN25","MIN32","MIN35","MIN37")]))

MCI.Data.Update<- MCI.Data.Update %>% rowwise() %>%

mutate(SumMemPSyk=sum(MIN1,MIN2,MIN3,MIN25,MIN32,MIN35,MIN37, na.rm=TRUE),

AbnormalMemPSyk=ifelse(SumMemPSyk>0,"Yes","No"))

options(warn = -1)

MCI.Data.Update$SumMemPSyk[MCI.Data.Update$N\_Miss.abmp==7] <- "NA"

MCI.Data.Update$AbnormalMemPSyk[MCI.Data.Update$N\_Miss.abmp==7] <- "NA"

MCI.Data.Update$AbnormalMemPSyk[MCI.Data.Update$AbnormalMemPSyk == 'NA']<- 'M'

##### Inspect AbnormalMemPSyk

CrossTable(MCI.Data.Update$AbnormalMemPSyk)

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

################################################## Word Fluency variable (WFC\_BIN) classify according

### To Educational levels low, medium and high according to variable educ\_SFS (Formal education (years)

### kohort 1930) computed by Simona Sacuiu July2019 using cut-off according to Tombaugh Tombaugh et al.

### Normative data stratified by age and educ for verbal fluency testing.ArchClinNeuropsych1999

options(warn = -1)

MCI.Data.Update$MIN66[MCI.Data.Update$MIN66 == 99] <- 'NA'

MCI.Data.Update$MIN66[MCI.Data.Update$MIN66 == 77] <- 'NA'

MCI.Data.Update<-MCI.Data.Update %>% rowwise()%>% mutate(educ\_level = ifelse(educ\_sfs < 9, 'low',

ifelse(educ\_sfs >=9 & educ\_sfs < 12,'medium','high')))

MCI.Data.Update$educ\_level[is.na(MCI.Data.Update$educ\_level)] <-'Missing'

MCI.Data.Update$MIN66<- as.numeric(MCI.Data.Update$MIN66)

hist(MCI.Data.Update$MIN66, col='blue', main = "Word Fluency Distribution", xlab = 'MIN66')

MCI.Data.Update$educ\_level<- as.factor(MCI.Data.Update$educ\_level)

describeBy(MCI.Data.Update$MIN66, group=MCI.Data.Update$educ\_level)

############################################################################# Tombaugh et. al Archieve

#MCI.Data.Update<-MCI.Data.Update %>% rowwise()%>%

# mutate(WFC\_Bin = ifelse(educ\_level == 'low' & (MIN66 < (13.1 - (3.8 \* 1.5))), 1,

#ifelse(educ\_level == 'medium' & (MIN66 < (13.9 - (3.4\* 1.5))),1,

#ifelse(educ\_level == 'high' & (MIN66 < (16.3 - (4.3\* 1.5))),1,

#ifelse(educ\_level =='Missing' & (MIN66 < (13.1 - (3.8\* 1.5))),1,0)))))

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

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

#### Word Fluency variable(WFC\_BIN) classify using cut-offs at sample MEAN(SD) by Educational levels low,

#### medium and high according to variable educ\_SFS (Formal education (years) kohort 1930)

#### Computed by Simona Sacuiu July 2019.

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

MCI.Data.Update<-MCI.Data.Update %>% rowwise()%>%

mutate(WFC\_Bin = ifelse(educ\_level == 'low' & (MIN66 < (16.63 - (5.41 \* 1.5))), 1,

ifelse(educ\_level == 'medium' & (MIN66 < (19.11 - (5.97\* 1.5))),1,

ifelse(educ\_level == 'high' & (MIN66 < (21.56 - (5.66\* 1.5))),1,

ifelse(educ\_level =='Missing' & (MIN66 < (16.40 - (5.70\* 1.5))),1,0)))))

MCI.Data.Update$WFC\_Bin<- as.factor(MCI.Data.Update$WFC\_Bin)

MCI.Data.Update$WFC\_Bin[MCI.Data.Update$LopNr == 14847] <- 'NA'

MCI.Data.Update$WFC\_Bin[is.na(MCI.Data.Update$WFC\_Bin)] <- 'Missing'

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

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

wfc<-select(MCI.Data.Update, LopNr, WFC\_Bin)

cc<- select(MCI.Data.Update.j, LopNr, MIN44I,MIN45,MIN46,MIN47,MIN48,MIN49,MIN50, STA22A,STA23A,STA24A)

all.wfc<- merge(cc, wfc, by.x = 'LopNr', by.y = 'LopNr' ,all = T)

all.wfc<- as.data.frame(all.wfc)

all.wfc$WFC\_Bin<- as.numeric(as.character(all.wfc$WFC\_Bin))

options(warn = -1)

all.wfc$MIN44I[all.wfc$MIN44I > 6]<- "NA"

all.wfc$MIN46[all.wfc$MIN46 > 6]<- "NA"

all.wfc$MIN45[all.wfc$MIN45 > 6]<- "NA"

all.wfc$MIN47[all.wfc$MIN47 > 4]<- "NA"

all.wfc$MIN48[all.wfc$MIN48 > 4]<- "NA"

all.wfc$MIN49[all.wfc$MIN49 > 4]<- "NA"

all.wfc$MIN50[all.wfc$MIN50 > 4]<- "NA"

all.wfc$MIN44I<- as.numeric(as.character(all.wfc$MIN44I))

all.wfc$MIN46<- as.numeric(as.character(all.wfc$MIN46))

all.wfc$MIN45<- as.numeric(as.character(all.wfc$MIN45))

all.wfc$MIN47<- as.numeric(as.character(all.wfc$MIN47))

all.wfc$MIN48<- as.numeric(as.character(all.wfc$MIN48))

all.wfc$MIN49<- as.numeric(as.character(all.wfc$MIN49))

all.wfc$MIN50<- as.numeric(as.character(all.wfc$MIN50))

all.wfc$STA22A<- as.numeric(as.character(all.wfc$STA22A))

all.wfc$STA23A<- as.numeric(as.character(all.wfc$STA23A))

all.wfc$STA24A<- as.numeric(as.character(all.wfc$STA24A))

all.wfc<- all.wfc %>%

rowwise() %>%

mutate(ANM.sum = sum(MIN44I,MIN45,MIN46,MIN47,MIN48,MIN49,MIN50,STA22A,STA23A,STA24A,WFC\_Bin, na.rm=T))

##### Count Missing

all.wfc$N\_Miss\_abnmp <- rowSums(is.na( all.wfc[,c('MIN44I','MIN45','MIN46','MIN47','MIN48','MIN49',

'MIN50','STA22A','STA23A','STA24A','WFC\_Bin')]))

all.wfc<-all.wfc %>%

mutate(AbnormalNonMemPSyk=ifelse(N\_Miss\_abnmp > 10 & ANM.sum >= 0 ,"NA",

ifelse(N\_Miss\_abnmp < 10 & ANM.sum > 0 ,"Yes",

ifelse(N\_Miss\_abnmp == 10 & ANM.sum >0 ,"Yes","No"))))

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

######################################################################## ADL according ADL-index 0-1

ADL\_Variables<- select(MCI.Data.Update,ADLTE,ADLTF,ADLTG,ADLTH,ADLTJ) %>% names()

MCI.Data.Update<- MCI.Data.Update %>% rowwise() %>%

mutate(ADLpreserved=ifelse(pmax(!!!rlang::syms(ADL\_Variables))==0,"Yes",

ifelse(pmax(!!!rlang::syms(ADL\_Variables))> 0 & MIN18 > 26,"Yes",

ifelse(pmax(!!!rlang::syms(ADL\_Variables))> 0 & MIN18 =="NA","No",

ifelse(pmax(!!!rlang::syms(ADL\_Variables))> 0 & MIN18 <= 26,"No")))))

##### Check for row sum Missing

MCI.Data.Update$N\_Miss\_adl <-

rowSums(is.na( MCI.Data.Update[,c('ADLTE','ADLTF','ADLTG','ADLTH','ADLTJ')]))

options(warn = -1)

MCI.Data.Update$ADLpreserved[MCI.Data.Update$N\_Miss\_adl==5] <- "NA"

MCI.Data.Update$ADLpreserved[is.na(MCI.Data.Update$ADLpreserved)]<-"No"

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

##################################################################################### iADL Computation

iadl\_Variables<-select(MCI.Data.Update,ADLTA,ADLTB,ADLTC,ADLTD)%>% names()

MCI.Data.Update<- MCI.Data.Update %>% rowwise() %>%

mutate(iADLpreserved=ifelse(pmax(!!!rlang::syms(iadl\_Variables))< 1 ,"Yes" ,

ifelse(pmax(!!!rlang::syms(iadl\_Variables))== 0 & MIN18 == "NA" ,"Yes",

ifelse(sum(ADLTA,ADLTB,ADLTC,ADLTD, na.rm=TRUE) <= 4 & MIN18 > 26,"Yes",

ifelse(sum(ADLTA,ADLTB,ADLTC,ADLTD, na.rm=TRUE) >= 1 & MIN18 > 26,"Yes","No")))))

##### Compute number of Missing for iadl variables

MCI.Data.Update$N\_Miss\_iadl <- rowSums( is.na( MCI.Data.Update[,c('ADLTA','ADLTB','ADLTC','ADLTD')]))

MCI.Data.Update$iADLpreserved[MCI.Data.Update$N\_Miss\_iadl>=4] <- "Missing"

MCI.Data.Update$iADLpreserved[is.na(MCI.Data.Update$iADLpreserved)] <- "No"

MCI.Data.Update.iadl<- select(MCI.Data.Update,N\_Miss\_iadl,ADLTA,ADLTB,ADLTC,ADLTD,MIN18 ,iADLpreserved)

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

################################### Final Data Processing and cleaning for Petersen MCI Diagnosis

PetersenMCI\_data<- data.frame(MCI.Data.Update)

options(warn = -1)

PetersenMCI\_data$Dementia<- rep("N", 365)

PetersenMCI\_data[PetersenMCI\_data=="Yes"]<- "Y"

PetersenMCI\_data[PetersenMCI\_data=="No"]<- "N"

PetersenMCI\_data[is.na(PetersenMCI\_data)]<- 'M'

PetersenMCI\_data<- data.frame(PetersenMCI\_data)

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

################################################## Computing Petersen Base Diagnosis before sub-classes

for(i in 1:nrow(PetersenMCI\_data)){

if(PetersenMCI\_data$Dementia[i] =="N"& PetersenMCI\_data$SR\_IQMemImp[i]=="Y"

& PetersenMCI\_data$NormalGenKog[i]=="Y"

& PetersenMCI\_data$AbnormalMemPSyk[i]=="Y"

& PetersenMCI\_data$ADLpreserved[i]=="Y"){ ###25th percental

PetersenMCI\_data$PetersensMCI\_Base[i]="Y"}

else if(PetersenMCI\_data$Dement[i] =="N"& (PetersenMCI\_data$SR\_IQMemImp[i]=="M"

| PetersenMCI\_data$NormalGenKog[i]=="M"

| PetersenMCI\_data$AbnormalMemPSyk[i]=="M"

| PetersenMCI\_data$ADLpreserved[i]=="M")){

PetersenMCI\_data$PetersensMCI\_Base[i]="M"}

else { PetersenMCI\_data$PetersensMCI\_Base[i]="N"}

}

##### Inspect Petersen base files

CrossTable(PetersenMCI\_data$PetersensMCI\_Base)

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

################################################################# MCI Classification Groups By Petersen

PetersenMCI\_data<- PetersenMCI\_data %>%

mutate(PetersensMCI = ifelse(PetersensMCI\_Base=="Y","PetersenMCI",

ifelse(PetersenMCI\_data$Dementia =="N"

& PetersenMCI\_data$SR\_IQMemImp=="N"

& PetersenMCI\_data$NormalGenKog=="Y"

& PetersenMCI\_data$AbnormalMemPSyk=="N"

& PetersenMCI\_data$ADLpreserved=="Y","Intact",

ifelse(PetersenMCI\_data$Dement =="N"

& PetersenMCI\_data$SR\_IQMemImp=="Y"

& PetersenMCI\_data$NormalGenKog=="Y"

& PetersenMCI\_data$AbnormalMemPSyk=="N"

& PetersenMCI\_data$ADLpreserved=="Y","Subjective Memory Impairment",

ifelse(PetersenMCI\_data$Dement =="N" & PetersenMCI\_data$SR\_IQMemImp=="N"

& PetersenMCI\_data$NormalGenKog=="Y"

& PetersenMCI\_data$AbnormalMemPSyk=="Y"

& PetersenMCI\_data$ADLpreserved=="Y","Objective Memory Impairment","Oklar")))))

PetersenMCI\_data$DEM1<- as.numeric(as.character(PetersenMCI\_data$DEM1))

PetersenMCI\_data$DEM2<- as.numeric(as.character(PetersenMCI\_data$DEM2))

PetersenMCI\_data$DEM3<- as.numeric(as.character(PetersenMCI\_data$DEM3))

PetersenMCI\_data$DEM4<- as.numeric(as.character(PetersenMCI\_data$DEM4))

PetersenMCI\_data$DEM5<- as.numeric(as.character(PetersenMCI\_data$DEM5))

PetersenMCI\_data$ADLTE<- as.numeric(as.character(PetersenMCI\_data$ADLTE))

PetersenMCI\_data$ADLTF<- as.numeric(as.character(PetersenMCI\_data$ADLTF))

PetersenMCI\_data$ADLTG<- as.numeric(as.character(PetersenMCI\_data$ADLTG))

PetersenMCI\_data$ADLTH<- as.numeric(as.character(PetersenMCI\_data$ADLTH))

PetersenMCI\_data$ADLTJ<- as.numeric(as.character(PetersenMCI\_data$ADLTJ))

PetersenMCI\_data$SR\_IQMemImp[PetersenMCI\_data$SR\_IQMemImp=='Missing']<- 'M'

PetersenMCI\_data$SR\_IQMemImp<- as.factor(as.character(PetersenMCI\_data$SR\_IQMemImp))

PetersenMCI\_data$NormalGenKog<- as.factor(as.character(PetersenMCI\_data$NormalGenKog))

PetersenMCI\_data$AbnormalMemPSyk<- as.factor(as.character(PetersenMCI\_data$AbnormalMemPSyk))

PetersenMCI\_data$ADLpreserved<- as.factor(as.character(PetersenMCI\_data$ADLpreserved))

PetersenMCI\_data$dem.sum <- rowSums(PetersenMCI\_data[,c('DEM1','DEM2','DEM3','DEM4','DEM5')])

options(warn = -1)

PetersenMCI\_data$dem.sum[is.na(dem.sum)]<- 'M'

PetersenMCI\_data$adl.sum <- rowSums(PetersenMCI\_data[,c('ADLTF','ADLTF','ADLTG','ADLTH','ADLTJ')])

PetersenMCI\_data$adl.sum<- as.numeric(PetersenMCI\_data$adl.sum)

options(warn = -1)

PetersenMCI\_data$adl.sum[is.na(PetersenMCI\_data$adl.sum)]<- 'M'

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

############################################################################# Sub-classification group Oklar (N = 25)

for(i in 1:nrow(PetersenMCI\_data)){

if(PetersenMCI\_data$PetersensMCI[i]=="Oklar"

& PetersenMCI\_data$SR\_IQMemImp[i]=="N"

& PetersenMCI\_data$NormalGenKog[i]=="Y"

& PetersenMCI\_data$AbnormalMemPSyk[i]=="N"

& PetersenMCI\_data$ADLpreserved[i]=="NA"

& as.numeric(!is.na(PetersenMCI\_data$dem.sum)) <= 2){

PetersenMCI\_data$PetersensMCI.1[i]='Intact'}

else if(PetersenMCI\_data$PetersensMCI[i]=="Oklar"

& PetersenMCI\_data$SR\_IQMemImp[i]=="N"

& PetersenMCI\_data$NormalGenKog[i]=="Y"

& PetersenMCI\_data$AbnormalMemPSyk[i]=="N"

& PetersenMCI\_data$ADLpreserved[i]=="N"

& PetersenMCI\_data$ADLTE[i]==1

& PetersenMCI\_data$adl.sum[i]==0){

PetersenMCI\_data$PetersensMCI.1[i]='Intact'}

else if(PetersenMCI\_data$PetersensMCI[i]=="Oklar"

& (PetersenMCI\_data$SR\_IQMemImp[i]=="N"|PetersenMCI\_data$SR\_IQMemImp[i]=="M")

& (PetersenMCI\_data$NormalGenKog[i]=="N"|PetersenMCI\_data$NormalGenKog[i]=="Y")

& PetersenMCI\_data$AbnormalMemPSyk[i]=="Y"

& PetersenMCI\_data$ADLpreserved[i]=="Y"){

PetersenMCI\_data$PetersensMCI.1[i]='OMI'}

else if(PetersenMCI\_data$PetersensMCI[i]=="Oklar"

& PetersenMCI\_data$SR\_IQMemImp[i]=="N"

& (PetersenMCI\_data$NormalGenKog[i]=="Y"|PetersenMCI\_data$NormalGenKog[i]=="N")

& PetersenMCI\_data$AbnormalMemPSyk[i]=="Y"

& PetersenMCI\_data$ADLpreserved[i]=="N"

& PetersenMCI\_data$ADLTE[i]==1

& PetersenMCI\_data$adl.sum[i]==0){

PetersenMCI\_data$PetersensMCI.1[i]='OMI'}

else {PetersenMCI\_data$PetersensMCI.1[i]=PetersenMCI\_data$PetersensMCI[i]}

}

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

######################################################## Final MCi Classification according to Petersens

for(i in 1:nrow(PetersenMCI\_data)){

if(PetersenMCI\_data$PetersensMCI.1[i]=="OMI"){

PetersenMCI\_data$PetersensMCI\_FINAL[i]='Objective Memory Impairment'}

else{

PetersenMCI\_data$PetersensMCI\_FINAL[i]=PetersenMCI\_data$PetersensMCI.1[i]}

}

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

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

PetersenMCI\_data$iADLpreserved<- as.character(PetersenMCI\_data$iADLpreserved)

PetersenMCI\_data$ADLpreserved<- as.character(PetersenMCI\_data$ADLpreserved)

options(warn = -1)

PetersenMCI\_data$ADLpreserved[PetersenMCI\_data$ADLpreserved =='NA']<- 'M'

PetersenMCI\_data$iADLpreserved[PetersenMCI\_data$iADLpreserved =='Missing']<- 'M'

all.wfc.1<- data.frame(select(all.wfc, LopNr, AbnormalNonMemPSyk))

options(warn = -1)

all.wfc.1[all.wfc.1 == 'NA']<- 'M'

all.wfc.1[all.wfc.1== 'Yes']<- 'Y'

all.wfc.1[all.wfc.1== 'No']<- 'N'

PetersenMCI\_data<- data.frame(list(PetersenMCI\_data,all.wfc.1)%>% reduce(left\_join, by = 'LopNr'))

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

############################################################################ Everyday Function Preserved

winbladMCI\_data<-PetersenMCI\_data %>%

mutate(EverydayFuncPreserved = ifelse(ADLpreserved == "N" & iADLpreserved == "N","N",

ifelse(ADLpreserved == "M"& iADLpreserved == "M" ,"M","Y")),

ObjectiveCogImp = ifelse(AbnormalMemPSyk == "M" | AbnormalNonMemPSyk =="M","M",

ifelse(AbnormalMemPSyk == "Y" ,"Y","N")))

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

################ WinbladMCI diagnonsis base data|(AbnormalNonMemPSyk == "Y" & AbnormalNonMemPSyk !="M")

for(i in 1:nrow(winbladMCI\_data)){

if(winbladMCI\_data$Dementia[i] =="N"& winbladMCI\_data$SubKogImp[i]=="Y"

& winbladMCI\_data$ObjectiveCogImp[i]=="Y"

& winbladMCI\_data$EverydayFuncPreserved[i]=="Y"

){ ###25th percental

winbladMCI\_data$WinbladMCI\_Base[i]="Y"}

else if(winbladMCI\_data$Dementia[i] =="N"& (winbladMCI\_data$SubKogImp[i]=="M"

| winbladMCI\_data$ObjectiveCogImp[i]=="M"

| winbladMCI\_data$EverydayFuncPreserved[i]=="M"

)){ ###25th percental

winbladMCI\_data$WinbladMCI\_Base[i]="M"}

else { winbladMCI\_data$WinbladMCI\_Base[i]="N"}

}

############################################################################## WinbladMCI Diagnosis final

winbladMCI\_data<- winbladMCI\_data %>%

mutate(WinbladsMCI = ifelse(WinbladMCI\_Base=="Y","WinbladsMCI",

ifelse(winbladMCI\_data$Dementia =="N"

& winbladMCI\_data$SubKogImp=="N"

& winbladMCI\_data$ObjectiveCogImp=="N"

& winbladMCI\_data$EverydayFuncPreserved=="Y","Intact",

ifelse(winbladMCI\_data$Dementia =="N"

& winbladMCI\_data$SubKogImp=="Y"

& winbladMCI\_data$ObjectiveCogImp=="N"

& winbladMCI\_data$EverydayFuncPreserved=="Y","Subjective Cognitive Impairment",

ifelse(winbladMCI\_data$Dementia =="N"

& winbladMCI\_data$SubKogImp=="N"

& winbladMCI\_data$ObjectiveCogImp=="Y"

& winbladMCI\_data$EverydayFuncPreserved=="Y",

"Objective Cognitive Impairment","Oklar")))))

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

winbladMCI\_data$DEM1<- as.numeric(as.character(winbladMCI\_data$DEM1))

winbladMCI\_data$DEM2<- as.numeric(as.character(winbladMCI\_data$DEM2))

winbladMCI\_data$DEM3<- as.numeric(as.character(winbladMCI\_data$DEM3))

winbladMCI\_data$DEM4<- as.numeric(as.character(winbladMCI\_data$DEM4))

winbladMCI\_data$DEM5<- as.numeric(as.character(winbladMCI\_data$DEM5))

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

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

winbladMCI\_data$WinbladsMCI<- as.factor(as.character(winbladMCI\_data$WinbladsMCI))

winbladMCI\_data$SRCogIMP<- as.factor(as.character(winbladMCI\_data$SRCogIMP))

winbladMCI\_data$IQCODEBIN\_FINAL<- as.factor(as.character(winbladMCI\_data$IQCODEBIN\_FINAL))

winbladMCI\_data$SubKogImp<- as.factor(as.character(winbladMCI\_data$SubKogImp))

winbladMCI\_data$ObjectiveCogImp<- as.factor(as.character(winbladMCI\_data$ObjectiveCogImp))

winbladMCI\_data$EverydayFuncPreserved<- as.factor(as.character(winbladMCI\_data$EverydayFuncPreserved))

winbladMCI\_data$dem.sum <- rowSums(winbladMCI\_data[,c('DEM1','DEM2','DEM3','DEM4','DEM5')])

options(warn = -1)

winbladMCI\_data$IQCODEBIN\_FINAL[winbladMCI\_data$IQCODEBIN\_FINAL == "Missing"]<-'M'

winbladMCI\_data$SRCogIMP[winbladMCI\_data$SRCogIMP=='Missing']<- 'M'

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

################################################################ Sub-classification group Oklar (N = 25)

winbladMCI\_data$WinbladsMCI<- as.character(winbladMCI\_data$WinbladsMCI)

for(i in 1:nrow(winbladMCI\_data)){

if(winbladMCI\_data$WinbladsMCI[i]=="Oklar"

& winbladMCI\_data$SRCogIMP[i]=="N"

& winbladMCI\_data$IQCODEBIN\_FINAL[i]=="N"

& winbladMCI\_data$SubKogImp[i]=="N"

& winbladMCI\_data$ObjectiveCogImp[i]=="N"

& winbladMCI\_data$EverydayFuncPreserved[i]=="M"

& winbladMCI\_data$dem.sum[i]== 0){

winbladMCI\_data$WinbladsMCI.1[i]='Intact'}

else {winbladMCI\_data$WinbladsMCI.1[i]=winbladMCI\_data$WinbladsMCI[i]}

}

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

########################### Cross Tables to check Summary Results

CrossTable(PetersenMCI\_data$PetersensMCI\_FINAL)

CrossTable(winbladMCI\_data$DayDiffCat)

CrossTable(winbladMCI\_data$SR\_IQMemImp)

CrossTable(winbladMCI\_data$SRMemImp)

CrossTable(winbladMCI\_data$IQCODEBINMEM\_FINAL)

CrossTable(PetersenMCI\_data$NormalGenKog)

CrossTable(PetersenMCI\_data$NormalGenKog)

CrossTable(PetersenMCI\_data$AbnormalMemPSyk)

CrossTable(PetersenMCI\_data$ADLpreserved)

CrossTable(winbladMCI\_data$WinbladsMCI.1)

CrossTable(winbladMCI\_data$SubKogImp)

CrossTable(winbladMCI\_data$SRCogIMP)

CrossTable(winbladMCI\_data$IQCODEBIN\_FINAL)

CrossTable(winbladMCI\_data$ObjectiveCogImp)

CrossTable(winbladMCI\_data$AbnormalMemPSyk)

CrossTable(winbladMCI\_data$WFC\_Bin)

CrossTable(winbladMCI\_data$AbnormalNonMemPSyk)

CrossTable(winbladMCI\_data$EverydayFuncPreserved)

CrossTable(winbladMCI\_data$iADLpreserved)

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

################################################################################ Write Data to CSV file

colnames(winbladMCI\_data)

winbladMCI\_data.1<- select(winbladMCI\_data,1,88:134)

tot.dat<- data.frame(list(MCI.Data.Update.last,winbladMCI\_data.1)

%>% reduce(left\_join, by='LopNr'))

options(warn = -1)

tot.dat[tot.dat== "Missing"]<- 'M'

#write.csv(tot.dat, 'H85-2015-MCI\_090819.csv')

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

print("MCI H85 2015 Diagnosis Algorithm Executed")