BARCODE project

Francesca Biondo

16/09/2022

This script presents the analyses used for the paper titled “Brain-age is associated with progression to dementia in memory clinic patients” (2022), Biondo, F., Jewell, A., Pritchard, M., Aarsland, A., Steves,C.J., Mueller,C., Cole, J.H. https://doi.org/10.1016/j.nicl.2022.103175 (https://doi.org/10.1016/j.nicl.2022.103175)

This research project is referred to as BARCODE - Brain Ageing and Risk of Cognitive Decline with the Maudsley Biomedical Research Centre Clinical Record Interactive Search (CRIS) reference number 19-008. This script uses the data extraction version 11/11/2019 of the BARCODE project.

The electronic health records (EHRs) were accessed via CRIS at the Maudsley BRC and used 3 databases: SLaM (in this script refered to as CRIS), HES (Hospital Episode Statistics) and mortality data via the ONS (Office of National Statistics). The EHRs data extraction was specified by the authors (F.B.& J.H.C) in May-August 2019 and executed by researchers at the Maudsley BRC (A.J. & M.P.).

The first data input to this script is the clinical information from the EHRS of 3666 memory clinic patients and includes the time-of-scan. The second data input is the neuroimaging data (brain-age values for each patient) which is merged with the clinical data just before FILTER4.

In this study we classify the memory clinic patients as of two types: the ones who go on to receive a dementia diagnosis (in this script labelled interchangeably as “DEM”, “Dementia” “futureDD” or “future Dementia Diagnosis”) and the ones who don’t go on to get a dementia diagnosis (labelled interchangeably as “DIAG”, “NonDementia”, “noDD” or “no dementia diagnosis”).

The script has 4 main parts: 1) Setting up: getting the libraries installed + a summary of the operational definition of how the data was extracted 2) Load data and data cleaning: sub-setting of data according to relevant constraints 3) Descriptives: summary statistics + plots 4) Statistical Analyses: Logistic Regression & Cox Proportional Hazards Regression

This script is dynamic such that changing the settings in the “Set settings” chunk will change the type of analysis carried out i.e. the main analyses, or one of the 3 sensitivity analyses.

CONTACT: For any comments and/or report of errors, please email me at f.biondo@ucl.ac.uk (mailto:f.biondo@ucl.ac.uk).

CITE: Please cite this paper if you use any of this work/code: Biondo, F., Jewell, A., Pritchard, M., Aarsland, A., Steves,C.J., Mueller,C., Cole, J.H.(2022) Brain-age is associated with progression to dementia in memory clinic patients. NeuroImage: Clinical, Vol36. https://doi.org/10.1016/j.nicl.2022.103175 (https://doi.org/10.1016/j.nicl.2022.103175)

PART 1: setting-up

Install packages

```
#install Rtools
install.packages('tidyverse')
install.packages('caret')
install.packages('survival')
#uninstall Rtools (C:\Rtools)


install.packages("kableExtra")
install.packages('stats')
install.packages('survival')
install.packages('arm')
install.packages('Amelia')
install.packages('car')

install.packages('cowplot')
install.packages('data.table')
install.packages('dplyr')
install.packages('DescTools')
install.packages('ggplot2')
install.packages('Hmisc')
install.packages('memisc')
install.packages('openxlsx')
install.packages('pastecs')
install.packages('plyr')
install.packages('plotly')
install.packages('pander')
install.packages('reghelper')
install.packages('stringr')
install.packages('survminer')
install.packages('tidyr')
install.packages('readxl')
#install.packages("nlme")
install.packages("lme4")
install.packages("statmod")
install.packages('ggExtra')
```

Call libraries

```
#R version used is 3.6.1
library(ggplot2)
library(plyr)
library(data.table)
library(stats)
library(memisc)
library(openxlsx)
library(stringr)
library(dplyr)
library(tidyr)
library(tidyverse)
library(Amelia)
library(DescTools)
library(survival)
library(survminer)
library(cowplot)
library(pastecs)
library(Hmisc)
library(reghelper)
library(car)
library(caret)
library(plotly)
library(pander)
#library(nlme)
library(lme4)
library(statmod)
library(arm)
library(kableExtra)
library(readxl)
library(ggExtra)
```

Set settings

```
#clear workspace
rm(list = ls())
sessionInfo()
```

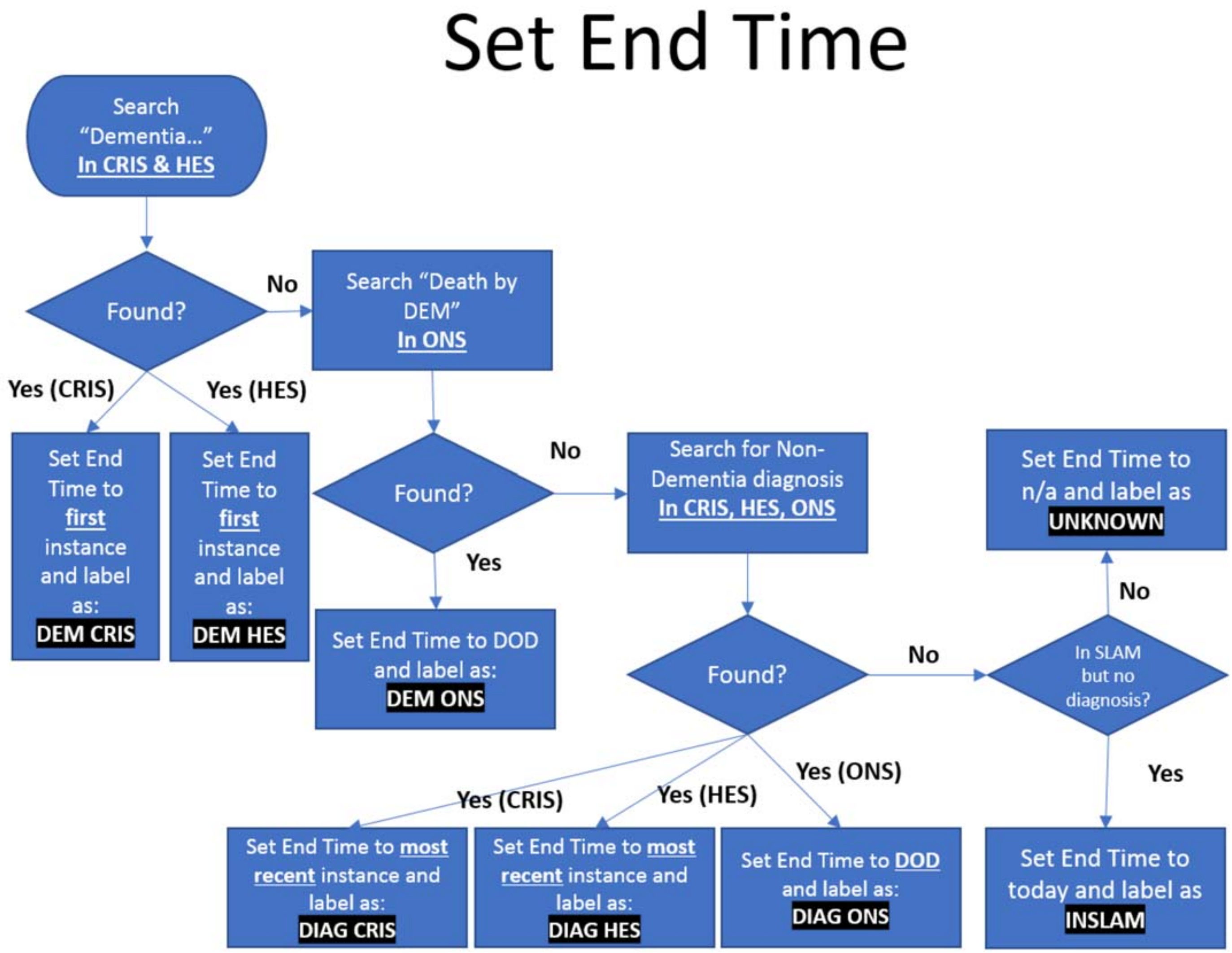
```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252 
## [2] LC_CTYPE=English_United Kingdom.1252   
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C                           
## [5] LC_TIME=English_United Kingdom.1252    
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] ggExtra_0.10.0    readxl_1.4.0      kableExtra_1.3.4  arm_1.12-2       
##  [5] statmod_1.4.36    lme4_1.1-30       Matrix_1.4-0      pander_0.6.5     
##  [9] plotly_4.10.0     caret_6.0-92      car_3.1-0         carData_3.0-5    
## [13] reghelper_1.1.1   Hmisc_4.7-0       Formula_1.2-4     pastecs_1.3.21   
## [17] cowplot_1.1.1     survminer_0.4.9   ggpubr_0.4.0      survival_3.3-1   
## [21] DescTools_0.99.45 Amelia_1.8.0      Rcpp_1.0.9        forcats_0.5.1    
## [25] purrr_0.3.4       readr_2.1.2       tibble_3.1.7      tidyverse_1.3.2  
## [29] tidyr_1.2.0       dplyr_1.0.9       stringr_1.4.0     openxlsx_4.2.5   
## [33] memisc_0.99.30.7  MASS_7.3-55       lattice_0.20-45   data.table_1.14.2
## [37] plyr_1.8.7        ggplot2_3.3.6    
## 
## loaded via a namespace (and not attached):
##   [1] backports_1.4.1      systemfonts_1.0.4    lazyeval_0.2.2      
##   [4] repr_1.1.4           splines_4.1.3        listenv_0.8.0       
##   [7] digest_0.6.29        foreach_1.5.2        htmltools_0.5.2     
##  [10] fansi_1.0.3          magrittr_2.0.3       checkmate_2.1.0     
##  [13] googlesheets4_1.0.0  cluster_2.1.2        tzdb_0.3.0          
##  [16] recipes_1.0.1        globals_0.15.1       modelr_0.1.8        
##  [19] gower_1.0.0          svglite_2.1.0        hardhat_1.2.0       
##  [22] jpeg_0.1-9           colorspace_2.0-3     rvest_1.0.2         
##  [25] haven_2.5.0          xfun_0.31            crayon_1.5.1        
##  [28] jsonlite_1.8.0       Exact_3.1            zoo_1.8-10          
##  [31] iterators_1.0.14     glue_1.6.2           gtable_0.3.0        
##  [34] gargle_1.2.0         ipred_0.9-13         webshot_0.5.3       
##  [37] future.apply_1.9.0   abind_1.4-5          scales_1.2.0        
##  [40] mvtnorm_1.1-3        DBI_1.1.3            rstatix_0.7.0       
##  [43] miniUI_0.1.1.1       viridisLite_0.4.0    xtable_1.8-4        
##  [46] htmlTable_2.4.1      foreign_0.8-82       proxy_0.4-27        
##  [49] km.ci_0.5-6          stats4_4.1.3         lava_1.6.10         
##  [52] prodlim_2019.11.13   htmlwidgets_1.5.4    httr_1.4.3          
##  [55] RColorBrewer_1.1-3   ellipsis_0.3.2       pkgconfig_2.0.3     
##  [58] nnet_7.3-17          sass_0.4.2           dbplyr_2.2.1        
##  [61] deldir_1.0-6         utf8_1.2.2           later_1.3.0         
##  [64] reshape2_1.4.4       tidyselect_1.1.2     rlang_1.0.4         
##  [67] munsell_0.5.0       cellranger_1.1.0    tools_4.1.3        
##  [70] cachem_1.0.6        cli_3.3.0           generics_0.1.3     
##  [73] broom_1.0.0         evaluate_0.15       fastmap_1.1.0      
##  [76] yaml_2.3.5          ModelMetrics_1.2.2.2 knitr_1.39        
##  [79] fs_1.5.2            zip_2.2.0           survMisc_0.5.6     
##  [82] rootSolve_1.8.2.3   future_1.26.1       nlme_3.1-155       
##  [85] mime_0.12           xml2_1.3.3          compiler_4.1.3     
##  [88] rstudioapi_0.13     png_0.1-7           e1071_1.7-11       
##  [91] ggsignif_0.6.3      reprex_2.0.1        bslib_0.4.0        
##  [94] stringi_1.7.6       nloptr_2.0.3        KMsurv_0.1-5       
##  [97] vctrs_0.4.1         pillar_1.8.0        lifecycle_1.0.1    
## [100] jquerylib_0.1.4     lmom_2.9            httpuv_1.6.5       
## [103] R6_2.5.1            latticeExtra_0.6-30 promises_1.2.0.1   
## [106] gridExtra_2.3       parallelly_1.32.0   gld_2.6.5          
## [109] codetools_0.2-18    boot_1.3-28         assertthat_0.2.1   
## [112] withr_2.5.0         expm_0.999-6        parallel_4.1.3     
## [115] hms_1.1.1           grid_4.1.3          rpart_4.1.16       
## [118] timeDate_4021.104   coda_0.19-4         minqa_1.2.4        
## [121] class_7.3-20        rmarkdown_2.14      googledrive_2.0.0  
## [124] pROC_1.18.0         shiny_1.7.2         lubridate_1.8.0    
## [127] base64enc_0.1-3     interp_1.1-3
```

```
#choose which version: 'main_analysis', sensitivity1, sensitivity2, sensitivity3
versionname <- "sensitivity1"

# subversion list based on True/False state of 3 parameters: Center, Scale, Poly; giving the 
following combos:
#1) PolyT, CentT = TFT: data is centered, age and age2 polynomials included
#2) PolyT, CentZ = TTT: data is standardized, age and age2 polynomials included
#3) PolyT, CentF = FFT: data is raw, age and age2 polynomials included
#4) PolyF, CentT = TFF: data is centered,age2 and poly excluded
#5) PolyF, CentZ = TTF: data is standardized, age2 and poly excluded
#6) PolyF, CentF = FFF: data is raw, age2 and poly excluded

# Table outputs are generated for all sub-versions above automatically, however please set su
bversion setting for detailed regression outputs below. As default I use TFT:
fCenter <- T
fScale <- F
fPoly <- T

#other settings
outputpath <- "U:\\aSLaM_2BARCODE_analysis\\Output\\"
inputpath <- "U:\\aSLaM_1BARCODE_data_PID_SLAMonly\\"
picres <- 280 #dpi setting for saving pics, I'm using 280 for high-res and 50 for low-res
options(knitr.table.format = 'markdown')
SET_dp <- 6 #set decimal places for rounding (originally 2dp, but 6dp necessary for norm vol)
```

Data Extraction at the Maudsley BRC (via CRIS)

The EHRs data used was the one extracted on the 11/11/2019 with a 16/06/2020 update for a few variables.

Part of this extraction involved defining the 'end-time' i.e. the way which a patient was defined as Dementia or Non-Dementia and setting a date for this event based on the source of the info (i.e. source is either SLaM (coded here as CRIS), HES or ONS databases). Here is a picture with a logical flowchart which was used to guide the extraction. DEM refers to the the Future Dementia Diagnosis patients and DIAG refers to the No Dementia Diagnosis patients. CRIS(SLAM)/HES/ONS refer to the databases used to establish the patient status: South London and Maudsley hospital data via CRIS, Hospital Episode Statistics and Office of National Statistics.


Set End Time
Search "Dementia..." In CRIS & HES
Found?
No
Search "Death by DEM" In ONS
Yes (CRIS)
Yes (HES)
Set End Time to first instance and label as: DEM CRIS
Set End Time to first instance and label as: DEM HES
Found?
No
Search for Non-Dementia diagnosis In CRIS, HES, ONS
Set End Time to n/a and label as UNKNOWN
Yes
Set End Time to DOD and label as: DEM ONS
No
Found?
No
In SLAM but no diagnosis?
Yes (ONS)
Yes (CRIS)
Yes (HES)
Yes
Set End Time to most recent instance and label as: DIAG CRIS
Set End Time to most recent instance and label as: DIAG HES
Set End Time to DOD and label as: DIAG ONS
Set End Time to today and label as INSLAM

# PART 2: Load and clean data ### Read EHRs data

```
#read data
data <- data.table(read_excel(paste(inputpath,'Data11102019.xlsx',sep="")))

#read data update
data_update <- data.table(read_excel(paste(inputpath,'HONOS65_Updated_ACE_16062020.xlsx',sep
="")))

#merge dataframes
data <- merge(data,data_update,all.x=T,by.x="Brcid",by.y="Brcid")
```

Date functions

```
#Function to convert excel dates to proper dates
xldat2propdat <- function(x) {
  if (class(x)=="POSIXct") {
  date <- as.Date(x,origin="1899-12-30",optional=TRUE)
  } else if (class(x)=="character") {
    x <- as.numeric(x)
  date <- as.Date(x,origin="1899-12-30",optional=TRUE)
  } else if (class(x)=="factor") {
  x <- as.character(x)

  } else {
    print("class input not as expected")
  }

}

#Function to check if class of element is Date
classISdat <- function(x){
  if (class(x)!="Date") {
    print ("Fail! class is not Date")
  } else {
    print ("Success! class is Date")
  }
}
```

Transform Dates

Dates are incorrectly set and need transformation.

```
#select the columns for date conversion
cols=c("Start_Time","OUT_End_time","IN_CRIS_Diag_DEM_date","OUT_CRIS_Diag_DEM_prox_ante_dat
e",
       "OUT_HES_Diag_DEM_prox_ante_date","OUT_ONS_DOD","OUT_CRIS_Diag_other_primary_dist_ante
_date",
       "OUT_HES_Diag_other_primary_dist_ante_date","OUT_HES_dischargedHES_dist_ante_date")

#Before transformation: are these Dates?
data[,lapply(.SD,classISdat),.SDcols=cols]#.SD = subset of Data i.e. the cols
```

```
## [1] "Fail! class is not Date"
## [1] "Fail! class is not Date"
## [1] "Fail! class is not Date"
## [1] "Fail! class is not Date"
## [1] "Fail! class is not Date"
## [1] "Fail! class is not Date"
## [1] "Fail! class is not Date"
## [1] "Fail! class is not Date"
## [1] "Fail! class is not Date"
```

```
##                 Start_Time            OUT_End_time   IN_CRIS_Diag_DEM_date
## 1: Fail! class is not Date Fail! class is not Date Fail! class is not Date
##    OUT_CRIS_Diag_DEM_prox_ante_date OUT_HES_Diag_DEM_prox_ante_date
## 1:          Fail! class is not Date         Fail! class is not Date
##                OUT_ONS_DOD OUT_CRIS_Diag_other_primary_dist_ante_date
## 1: Fail! class is not Date                    Fail! class is not Date
##    OUT_HES_Diag_other_primary_dist_ante_date
## 1:                   Fail! class is not Date
##    OUT_HES_dischargedHES_dist_ante_date
## 1:              Fail! class is not Date
```

```
#Transform and replace those columns with output
data[,(cols):=lapply(.SD,xldat2propdat),.SDcols=cols]# := allows replacement

#After transformation: are these Dates?
data[,lapply(.SD,classISdat),.SDcols=cols]
```

```
## [1] "Success! class is Date"
## [1] "Success! class is Date"
## [1] "Success! class is Date"
## [1] "Success! class is Date"
## [1] "Success! class is Date"
## [1] "Success! class is Date"
## [1] "Success! class is Date"
## [1] "Success! class is Date"
## [1] "Success! class is Date"
```

```
##                Start_Time           OUT_End_time  IN_CRIS_Diag_DEM_date
## 1: Success! class is Date Success! class is Date Success! class is Date
##    OUT_CRIS_Diag_DEM_prox_ante_date OUT_HES_Diag_DEM_prox_ante_date
## 1:           Success! class is Date          Success! class is Date
##               OUT_ONS_DOD OUT_CRIS_Diag_other_primary_dist_ante_date
## 1: Success! class is Date                     Success! class is Date
##    OUT_HES_Diag_other_primary_dist_ante_date
## 1:                    Success! class is Date
##    OUT_HES_dischargedHES_dist_ante_date
## 1:               Success! class is Date
```

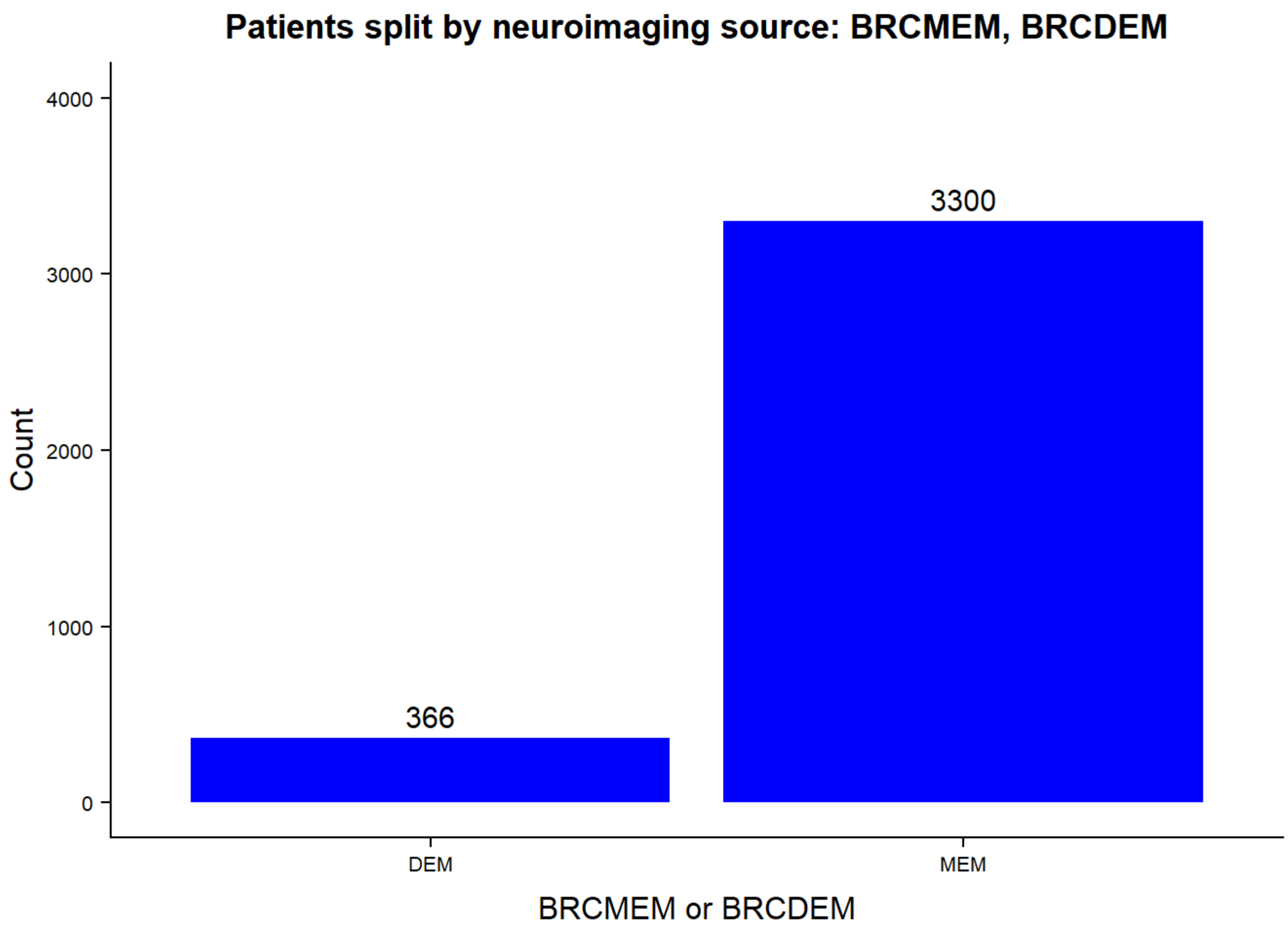
FILTER1: Dementia before scan

Out of those with a Dementia diagnosis, how many had such a diagnosis before their scan? (remove these cases) Similarly, out of those with a Non-Dementia diagnosis, how many had their last diagnosis before the scan? (remove these cases)

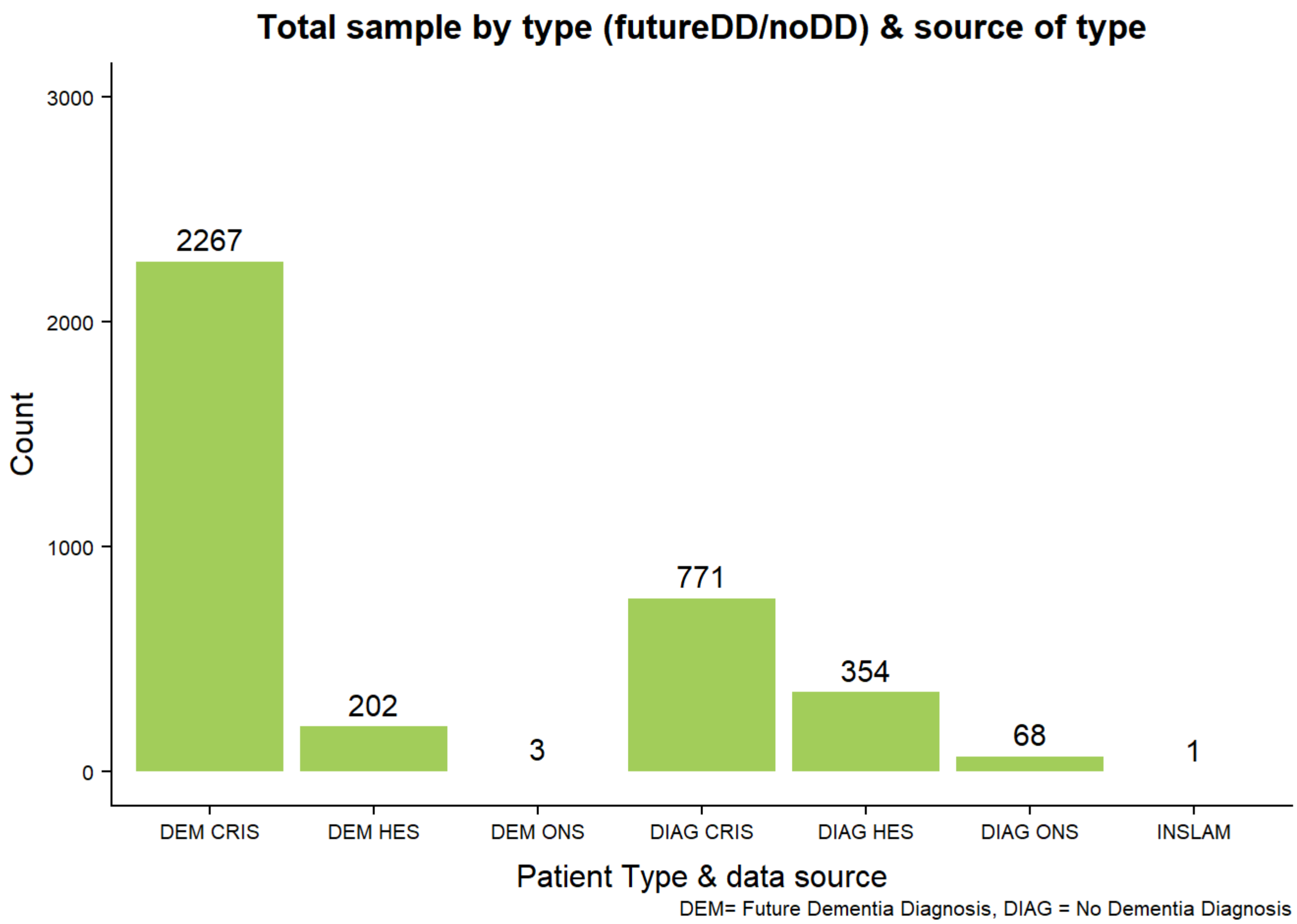
```
#Who was diagnosed before the scan? i.e. whose end time is before the start time?
data <- data%>%
  mutate(FRAN_endb4start=OUT_End_time<Start_Time)
```

First, we'll plot the whole sample by the neuroimaging source: BRCMEM (primarily SLaM memory clinics) and BRCDEM (other psychiatric referrals). Second, we'll plot by patient type (Dementia or Non-Dementia) and the database source from where the patient type was defined from: CRIS, HES or ONS databases.

```
summary_variable1 <- ddply(data,.(Mem_or_Dem), summarise, y=length(Mem_or_Dem)) %>%

  ggplot(aes(x=Mem_or_Dem,y=y,fill=Mem_or_Dem))+ #plot!
  geom_bar(stat="identity",fill="blue")+
  theme_cowplot()+
  geom_text(aes(label=y),vjust=-0.5,cex=4)+
  labs(x="BRCMEM or BRCDEM",y="Count")+
  ylim(0,4000)+
  theme(plot.title=element_text(hjust=0.5,size=13))+
  ggtitle("Patients split by neuroimaging source: BRCMEM, BRCDEM")+
  theme(axis.text.x=element_text(size=8))+
  theme(axis.text.y=element_text(size=8))+
  theme(axis.title.x=element_text(size=12,vjust=-1))+
  theme(axis.title.y=element_text(size=12))

summary_variable1
```



```
summary_variable2 <- ddply(data,.(OUT_End_time_event), summarise, y=length(OUT_End_time_even
t)) %>%

  ggplot(aes(x=OUT_End_time_event,y=y,fill=OUT_End_time_event))+ #plot!
  geom_bar(stat="identity",fill="darkolivegreen3")+
  theme_cowplot()+
  geom_text(aes(label=y),vjust=-0.5,cex=4)+
  labs(x="Patient Type & data source",y="Count",caption = "DEM= Future Dementia Diagnosis, DI
AG = No Dementia Diagnosis")+
  ylim(0,3000)+
  theme(plot.title=element_text(hjust=0.5,size=13),plot.caption=element_text(size=8))+
  ggtitle("Total sample by type (futureDD/noDD) & source of type")+
  theme(axis.text.x=element_text(size=8))+
  theme(axis.text.y=element_text(size=8))+
  theme(axis.title.x=element_text(size=12,vjust=-1))+
  theme(axis.title.y=element_text(size=12))

summary_variable2
```



```
## The total number of patients with a scan =  3666
## The total number of patients with a Dementia diagnosis BEFORE scan =  742
## The total number of patients with a Dementia diagnosis AFTER scan =  1730
## The total number of No Dementia Diagnosis patients with their final diagnosis BEFORE scan
=  72
## The total number of No Dementia Diagnosis patients with their final diagnosis AFTER scan =
1121
## The total number of patients without a diagnosis =  1
```

```
#FILTER: remove all cases those whose End Time is b4 Start Time & undiagnosed
data <- data%>%
  filter(!(FRAN_endb4start=="TRUE"))%>%
  filter(!(startsWith((.$OUT_End_time_event),"INSLAM")))
```

```
## From here onwards, we will only consider the cases with either a Dementia or (final) Non-D
ementia diagnosis AFTER scan = 2851
```

Redefine “End_Time, End_Event and Scan2End_Time” variables

As illustrated in the ‘Set End Time’ flowchart embedded a few code chunks above, the “End_Time Event” variable was defined at extraction as “choose the earliest date for a Dementia diagnosis in the following databases: CRIS, HES and ONS (in this order).” However, this means that the “End_Time Event date” is not necessarily the earliest across the three databases. For example, if a Dementia diagnosis was detected in the CRIS database, this is marked as the earliest date. However, it’s possible that there’s an earlier Dementia diagnosis date in the HES or ONS databases. Hence, revised versions of the “End_Time, End_Event and Scan2End_Time” variables are required that reflect the earliest Dementia diagnosis across the three databases (CRIS, HES, ONS). To do this I will compare the 3 dates (Dementia diagnosis dates across CRIS, HES and ONS) and choose the earliest to define the “End_Time”. But first, I will need to generate a variable that indicates Dates for Death that includes Dementia as a cause.

For the Dementia cases

```
# Extract ONS Dementia Time (death by dementia)
data <- data%>%
  mutate(`FRAN_DeathbyDEM_Date`=ifelse(OUT_ONS_Death_DEM==1,OUT_ONS_DOD,NA))%>% #if death by
dementia then select this date as the Date of Death (DOD)
  mutate(FRAN_DeathbyDEM_Date=as.Date(FRAN_DeathbyDEM_Date,origin="1970-01-01",optional=TRU
E)) #please note different origin to Excel origin
```

```r
# Compare 3 variables containing Dementia diagnosis dates, pick oldest date, and then pick that event

FRAN_DEM_End_Time <- vector("double",length(data$OUT_CRIS_Diag_DEM_prox_ante_date))
FRAN_DEM_End_Time_Event <- vector("character",length(data$OUT_CRIS_Diag_DEM_prox_ante_date))

for (i in seq_along(data$OUT_CRIS_Diag_DEM_prox_ante_date)) {

  FRAN_DEM_End_Time[[i]] <-  suppressWarnings(min(as.numeric(c(data$OUT_CRIS_Diag_DEM_prox_ante_date[[i]],
                                           data$OUT_HES_Diag_DEM_prox_ante_date[[i]],
                                           data$FRAN_DeathbyDEM_Date[[i]])),na.rm=TRUE))

  if (FRAN_DEM_End_Time[[i]]=="Inf"){
    FRAN_DEM_End_Time[[i]] <- NA
    FRAN_DEM_End_Time_Event[[i]] <- NA
  } else {
    FRAN_DEM_End_Time_Event[[i]] <- match(FRAN_DEM_End_Time[[i]],(c(data$OUT_CRIS_Diag_DEM_prox_ante_date[[i]],
                                                                 data$OUT_HES_Diag_DEM_prox_ante_date[[i]],
                                                                 data$FRAN_DeathbyDEM_Date[[i]])))
    #in event of ties I want CRIS to precede HES, and HES to precede ONS; hence the order of variables above.
  }

}

#convert to Date (please note different origin to Excel origin)
FRAN_DEM_End_Time <- as.Date((FRAN_DEM_End_Time),origin="1970-01-01",optional=TRUE)

#convert index (1-3) to string labels
setlabels <- c("DEM CRIS", "DEM HES", "DEM ONS") #set in the order inputted above
setlevels <- c(1,2,3)#set levels = length labels
FRAN_DEM_End_Time_Event <- ordered(FRAN_DEM_End_Time_Event,levels=setlevels,labels=setlabels)
stringsAsFactors <- FALSE



## Save 2 new variables to data table
data <- cbind(data,FRAN_DEM_End_Time,FRAN_DEM_End_Time_Event)

#Add updated "scan to (first) dementia diagnosis time interval" variable (in days)
data <- data%>%
  mutate(`FRAN_scan2DEM_time` =FRAN_DEM_End_Time-Start_Time)
```

We now have 3 key new variables that indicate: i) the earliest time a Dementia diagnosis was detected across all 3 databases,“FRAN_DEM_End_Time”; ii) the database of this earliest time, “FRAN_DEM_End_Time_Event”;and iii) the time interval between the scan (start time) and the earliest Dementia diagnosis, “FRAN_scan2DEM_time”.

In a similar way to the Future Dementia Diagnosis cases, the No Dementia Diagnosis cases need to be re-evaluated. This is a similar but not identical transformation, because in the No Dementia Diagnosis cases we want to establish the End_time based on the latest (closest to today) Non-Dementia diagnostic event (as opposed to the the earliest diagnostic event, as for the Dementia cases). We do this because we want to know the date furthest away from the scan that we are confident the patient did not get a Dementia diagnosis. Before we compare the Non-Dementia Dates across CRIS, HES and ONS databases, we first need to extract the ONS Non-Dementia Date. Please note that “DIAG” = No Dementia Diagnosis.

For the Non-Dementia cases

```
# Extract ONS Non-Dementia Time
data <- data%>%
  mutate(`FRAN_DeathbyOther_Date`=ifelse(OUT_ONS_Death_DEM!=1,OUT_ONS_DOD,NA))%>%
  mutate(FRAN_DeathbyOther_Date=as.Date(FRAN_DeathbyOther_Date,origin="1970-01-01",optional=T
RUE)) #please note different origin to Excel origin
```

```
# Compare 3 variables containing Non-Dementia diagnosis dates, pick most recent date, and the
n pick that event
FRAN_DIAG_End_Time <- vector("double",length(data$FRAN_DeathbyOther_Date))
FRAN_DIAG_End_Time_Event <- vector("character",length(data$FRAN_DeathbyOther_Date))

for (i in seq_along(data$FRAN_DeathbyOther_Date)) {

  FRAN_DIAG_End_Time[[i]] <-  suppressWarnings(max(as.numeric(c(data$FRAN_DeathbyOther_Date
[[i]],
                                           data$OUT_HES_dischargedHES_dist_ante_date[[i]],
                                           data$OUT_CRIS_Diag_other_primary_dist_ante_date
[[i]])),na.rm=TRUE))
  #in event of ties I want DIAG by ONS to precede HES and CRIS

  if (FRAN_DIAG_End_Time[[i]]=="Inf"){
    FRAN_DIAG_End_Time[[i]] <- NA
    FRAN_DIAG_End_Time_Event[[i]] <- NA
  } else {
    FRAN_DIAG_End_Time_Event[[i]] <- match(FRAN_DIAG_End_Time[[i]],(c(data$FRAN_DeathbyOther_
Date[[i]],
                                                                   data$OUT_HES_dischargedHES
_dist_ante_date[[i]],
                                                            data$OUT_CRIS_Diag_other_primary_
dist_ante_date[[i]])))
  #in event of ties I want DIAG by ONS to precede HES and CRIS

  # in fact, I want to make sure that if there's death by Non-Dementia then this overrides HE
S and CRIS
    if (is.na(data$FRAN_DeathbyOther_Date[[i]])==0){
      FRAN_DIAG_End_Time[[i]] <- data$FRAN_DeathbyOther_Date[[i]]
      FRAN_DIAG_End_Time_Event[[i]] <- 1
    }

  }

}

#convert to Date (please note different origin to Excel origin)
FRAN_DIAG_End_Time <- as.Date((FRAN_DIAG_End_Time),origin="1970-01-01",optional=TRUE)

#convert indeces to labels
setlabels2 <- c("DIAG ONS","DIAG HES","DIAG CRIS") #set in the order inputted above. Preceden
ce to ONS, then HES, then CRIS.
setlevels2 <- c(1,2,3)#set levels = length labels
FRAN_DIAG_End_Time_Event <- ordered(FRAN_DIAG_End_Time_Event,levels=setlevels2,labels=setlabe
ls2)
stringsAsFactors <- FALSE

## Save 2 new variables to data table
data <- cbind(data,FRAN_DIAG_End_Time,FRAN_DIAG_End_Time_Event)
```

```
#Add updated "scan to (last) Non-Dementia diagnosis time interval" variable (in days)
data <- data%>%
  mutate(`FRAN_scan2DIAG_time` =FRAN_DIAG_End_Time-Start_Time)
```

We now have 3 additional new variables that indicate: i) the latest time a Non-Dementia diagnosis was detected across all 3 databases, “FRAN_DIAG_End_Time”; ii) the database of this latest time, “FRAN_DIAG_End_Time_Event”;and iii) the time interval between the scan (start time) and the latest Non-Dementia diagnosis, “FRAN_scan2DIAG_time”.

Next, we want to merge the 3 new Dementia and 3 new Non-Dementia variables into one set of 3 global over-arching variables: End_time, End_Event and Scan2End_time. Given that Dementia cases take precedence, we simply populate these 3 global variables using the Dementia variables unless there’s no case of dementia (“NA” Dementia cases (rows)) and hence populate with the corresponding Non-Dementia entry instead.

Merge FRAN_DEM with FRAN_DIAG variables into 3 global variables

```
#Create FRAN_End_Time &  FRAN_End_Time_Event
#note: DEM events are final, so they take precedence over DIAG events. In other words, replace DEM NA events with DIAG events.

data <- data %>%
mutate(`FRAN_End_Time_Event`= ifelse(is.na(FRAN_DEM_End_Time_Event),as.character(FRAN_DIAG_End_Time_Event),as.character(FRAN_DEM_End_Time_Event)))%>%
mutate(`FRAN_End_Time`= ifelse(is.na(FRAN_DEM_End_Time_Event),as.Date(FRAN_DIAG_End_Time),as.Date(FRAN_DEM_End_Time)))

#If after merge, there are any cases labelled as NA, use original End_Time_Event labels + finally calculate "FRAN_End_scan2End"
sum(is.na(data$FRAN_End_Time_Event)) #1 case is NA
```

```
## [1] 1
```

```
data <- data%>%
mutate(`FRAN_End_Time_Event`= ifelse(is.na(FRAN_End_Time_Event),as.character(OUT_End_time_event),as.character(FRAN_End_Time_Event)))%>%
mutate(`FRAN_End_Time`= ifelse(is.na(FRAN_End_Time_Event),OUT_End_time,FRAN_End_Time))%>%
mutate(`FRAN_End_Time`=as.Date(FRAN_End_Time,origin="1970-01-01",optional=TRUE))%>%
mutate(`FRAN_End_scan2End`= ifelse(is.na(FRAN_DEM_End_Time_Event),as.numeric(FRAN_scan2DIAG_time),as.numeric(FRAN_scan2DEM_time)))

sum(is.na(data$FRAN_End_Time_Event)) #0
```

```
## [1] 0
```

FILTER2&3 - Medical history and 3 Months post scan

Filter#2 removes cases where a medical history of dementia (dementia diagnosis before scan) was detected and which failed to be detected at data extraction level. There were only 3 such cases. Filter#3 removes cases where time between scan and diagnosis is 3 months or less

```
#remove 3 residual cases of patients with medical history containing dementia (identified in
a separate search)
#ID1, ID2, ID3 are hidden from public view due to data privacy constraints
indexRemove <- which(data$Brcid== ID1)
data <- data[-indexRemove,]
indexRemove <- which(data$Brcid== ID2)
data <- data[-indexRemove,]
indexRemove <- which(data$Brcid== ID3)
data <- data[-indexRemove,]

#threshold by 3 months (91days: 365/12=30.41 *3= 91.25)
index=data$FRAN_End_scan2End>91
data=data[index,]
```

Merge clinical with neuroimaging data

```
#load brainage data
data_nan <- read.xlsx(paste(inputpath,"merge_brainage_DICOM_keyALERT.xlsx",sep=""), colNames=
T, sheet="step6_essentialcolumns")

#merge with EHRs data
data <- merge(data,data_nan,all.x=T,by.x="Brcid",by.y="Brcid")
```

FILTER4-6 & select variables: Remove missing brainage, scanner info and MMSE scores

Filter#4 removes cases with missing brainage score. Filter#5 removes cases without scanner info or not scanner A. Filter#6 removes cases with missing MMSE scores.

```
## select & rename relevant variables
datasub1 <- cbind.data.frame(Brcid=data$Brcid,
                             nanid=data$nanid,
                             Start_Time=data$Start_Time,
                             age=data$IN_CRIS_age,
                             sex=data$IN_CRIS_sex,
                             MMSE_numerator=as.numeric(as.character(data$IN_CRIS_MMSE_numerat
or)),
                             MMSE_denominator=as.numeric(as.character(data$IN_CRIS_MMSE_denom
inator)),
                             End_Time_Event=data$FRAN_End_Time_Event,
                             End_Time=data$FRAN_End_Time,
                             End_scan2End=data$FRAN_End_scan2End,
                             brain.predictedage=data$brain.predictedage,
                             lower.CI=data$lower.CI,
                             upper.CI=data$upper.CI,
                             scanner=data$scanner,
                             normVol=data$normVol,
                             ACEnum=as.numeric(data$IN_CRIS_ACE_numeratorNEW),
                             ethnicity=as.factor(data$IN_CRIS_ethnicity))

#check for duplicates
duplicate <- duplicated(as.character(datasub1$Brcid))%>%
  datasub1[.,]

#one case is duplicated and this is manually removed in 2 hidden lines of code below (hidden
due to data privacy constraints)
```

```
## subset (i): remove patients (a)without brainage score, (b) w/o scanner info or not scanner
A
datasub1b <- datasub1[which(!is.na(datasub1$brain.predictedage)&datasub1$scanner=="CNSCNS
A"),]
datasub1vs1b <- anti_join(datasub1,datasub1b)
```

```
## We removed cases with i) missing brainage score & or scanner was not the same/no scanner i
nfo = 81 ii) MMSE score was missing = 165
```

```
# subset (ii): remove cases with missing MMSE score
datasub2 <- datasub1b[which(!is.na(datasub1b$MMSE_numerator),),]



#sanity checks
{if (sum(datasub2$scanner=="CNSCNSA")!=nrow(datasub2))stop ('sanity check 1, failed')}
{if (sum(is.na(datasub2$Brcid))!=0)stop ('sanity check 2, failed')}
{if (length(unique(as.character(datasub1vs1b$Brcid)))!=nrow(datasub1vs1b))stop ('sanity check
3, failed')}
{if (length(unique(as.character(datasub2$Brcid)))!=nrow(datasub2))stop ('sanity check 4, fail
ed')}
```

Calculate new variables including brainPAD

```
# group
datasub2$End_Time_Event <- as.character(datasub2$End_Time_Event)
datasub2$group <- startsWith((datasub2$End_Time_Event),"DEM") #dementia (DEM)=1, nondementia
(DIAG)=0
datasub2$group <- factor(as.numeric(datasub2$group),levels=c(1,0),labels=c("futureDD","noD
D"))
#R, annoyingly, recodes this to Dementia(futureDD)=1, Non-Dementia(noDD)=2 (see below)
head(datasub2$group)
```

```
## [1] noDD     noDD     noDD     futureDD noDD     futureDD
## Levels: futureDD noDD
```

```
str(datasub2$group)
```

```
##  Factor w/ 2 levels "futureDD","noDD": 2 2 2 1 2 1 2 2 1 2 ...
```

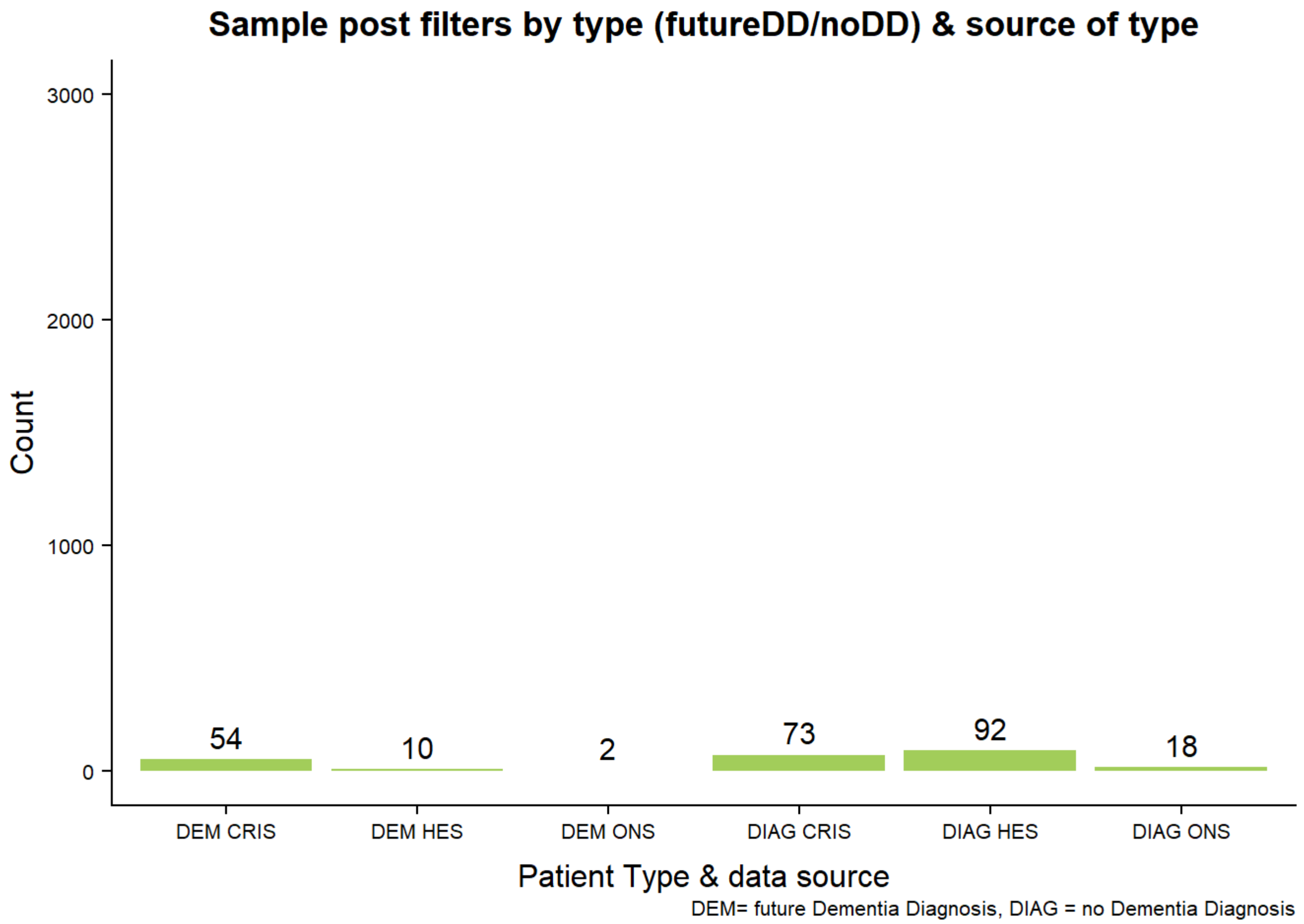
```
# to fix this, manually convert it using fct_rev
datasub2$group=fct_rev(datasub2$group) #use fct_rev so that Dementia=2, Non-Dementia=1



# add variables
datasub2 <- datasub2%>%
  mutate(brainPAD =datasub2$brain.predictedage-datasub2$age,
         MMSE_percentage=MMSE_numerator/MMSE_denominator*100,
         End_scan2Endyears =as.numeric(as.numeric(End_scan2End)/365))

# save
#save(datasub2, file = "datasub2_original.Rda")
```

FILTER7: Subset data for sensitivity analyses (if applicable)

```
if (versionname=="sensitivity1") {
  #threshold time-2-diag to min 3years
  datasub2 <- datasub2[datasub2$End_scan2Endyears>=3, ]
}else if (versionname=="sensitivity2"){
  #sensitivity analysis2: threshold MMSE to min 27
  datasub2 <- datasub2[datasub2$MMSE_numerator>=27, ]
}else if (versionname=="sensitivity3"){
  #sensitivity analysis3: threshold age to min 55
  datasub2 <- datasub2[datasub2$age>=55, ]
}
```

Plot by type post-filters

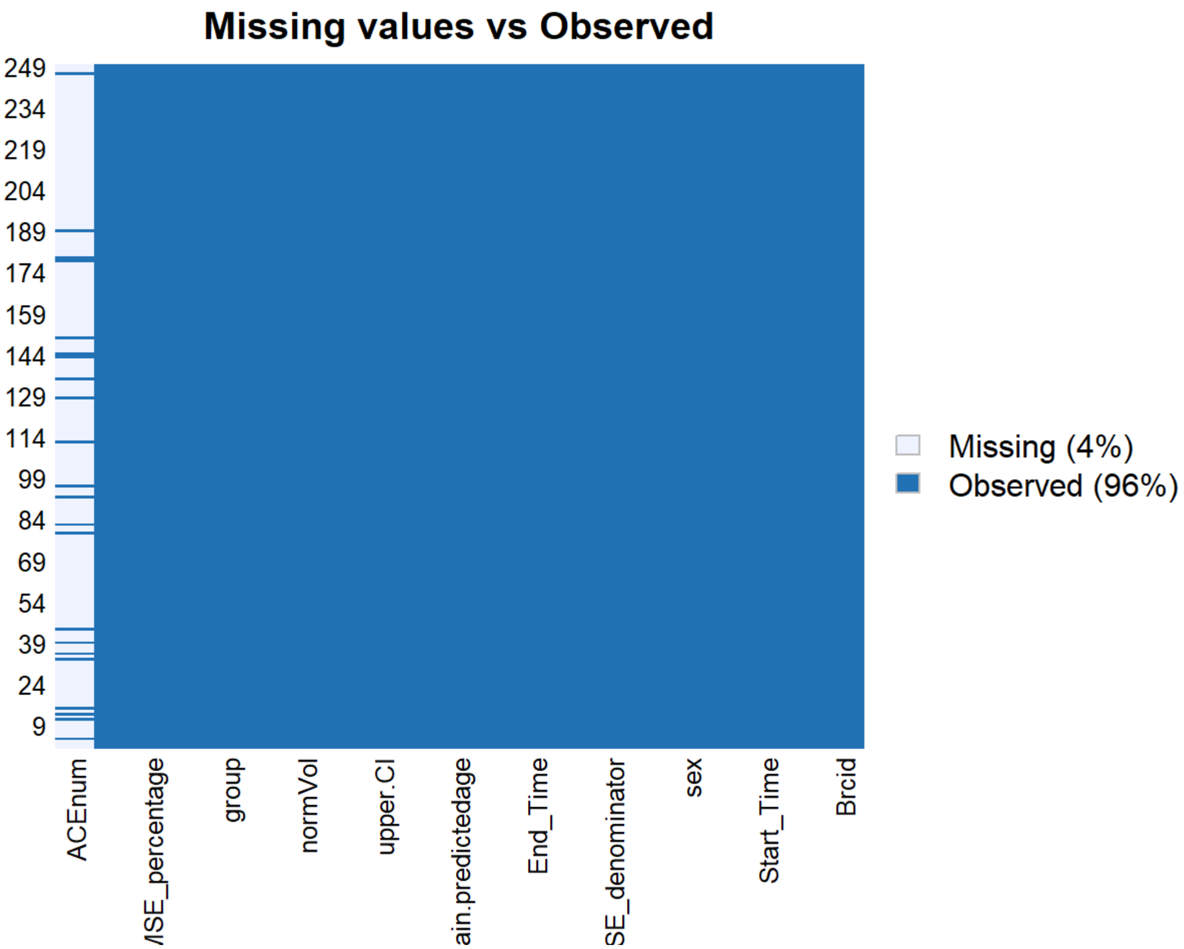
```
summary_variable <- ddply(datasub2,.(End_Time_Event), summarise, y=length(End_Time_Event)) %
>%

  ggplot(aes(x=End_Time_Event,y=y,fill=End_Time_Event))+ #plot!
  geom_bar(stat="identity",fill="darkolivegreen3")+
  theme_cowplot()+
  geom_text(aes(label=y),vjust=-0.5,cex=4)+
  labs(x="Patient Type & data source",y="Count",caption = "DEM= future Dementia Diagnosis, DI
AG = no Dementia Diagnosis")+
  ylim(0,3000)+
  theme(plot.title=element_text(hjust=0.5,size=13),plot.caption=element_text(size=8))+
  ggtitle("Sample post filters by type (futureDD/noDD) & source of type")+
  theme(axis.text.x=element_text(size=8))+
  theme(axis.text.y=element_text(size=8))+
  theme(axis.title.x=element_text(size=12,vjust=-1))+
  theme(axis.title.y=element_text(size=12))

summary_variable
```



PART 3: Descriptives

Descriptive statistics

```
#sapply(datasub2, function(x) sum(is.na(x)))         # check missingness
missmap(datasub2, main="Missing values vs Observed")    # check missingness
```



```
uniq <- sapply(datasub2, function(x) length(unique(x)))#%>%      # check uniqueness

kbl(uniq,caption = "Uniqueness") %>%
  kable_classic(full_width = F, html_font = "Cambria",position = "left")
```

Uniqueness

	x
Brcid	249
nanid	249
Start_Time	211
age	50
sex	2
MMSE_numerator	28
MMSE_denominator	14
End_Time_Event	6
End_Time	220
End_scan2End	222
brain.predictedage	249

	x
lower.CI	249
upper.CI	249
scanner	1
normVol	249
ACEnum	21
ethnicity	13
group	2
brainPAD	249
MMSE_percentage	54
End_scan2Endyears	222

```
#extract descriptive stats
stat_table1 <- format(stat.desc(datasub2), scientific=F) # summary
kbl(stat_table1,caption = "Descriptive Stats") %>%
  kable_classic(full_width = F, html_font = "Cambria",position = "left")%>%
  scroll_box(width = "100%", height = "100%")
```

Descriptive Stats

	Brcid	nanid	Start_Time	age	sex	MMSE_numerator	MMSE_den
nbr.val	NA	249.0000000	249.00000000	249.0000000	NA	249.0000000	249.00000
nbr.null	NA	0.0000000	0.00000000	0.0000000	NA	0.0000000	0.0000000
nbr.na	NA	0.0000000	0.00000000	0.0000000	NA	0.0000000	0.0000000
min	NA	17910.0000000	15015.00000000	27.0000000	NA	2.0000000	16.000000
max	NA	33175.0000000	17030.00000000	95.0000000	NA	30.0000000	30.000000
range	NA	15265.0000000	2015.00000000	68.0000000	NA	28.0000000	14.000000
sum	NA	6231971.0000000	3967240.00000000	16931.0000000	NA	6064.0000000	7106.0000
median	NA	25276.0000000	15951.00000000	70.0000000	NA	27.0000000	30.000000
mean	NA	25027.9959839	15932.69076305	67.9959839	NA	24.3534137	28.538152
SE.mean	NA	254.7448784	33.80506529	0.6835021	NA	0.4024736	0.2256358
CI.mean	NA	501.7393103	66.58163356	1.3462090	NA	0.7927023	0.4444068
var	NA	16158843.3104677	284552.82737401	116.3265967	NA	40.3342726	12.676965
std.dev	NA	4019.8063772	533.43493265	10.7854808	NA	6.3509269	3.5604727
coef.var	NA	0.1606124	0.03348053	0.1586194	NA	0.2607818	0.1247618

Plot function

A custom function for plotting to avoid repeating code.

```
# fplotfun inputs are data=dataset, varname=variable name, min=start range, max=end range, gap1=units bins,gap2=units xaxis, maintitle=main title, xlabt=xaxis title, fun.fun=type of plot function, vartype = 'continuous', 'discrete' or 'factorial' (defining way to present variable)

fplotfun <- function(data,varname,min,max,gap1,gap2,maintitle,xlabt, fun.fun,vartype) {

  contxaxis <- F; factxaxis <- F; discxaxis <- F

  #define function2 used in ggplot() depending if plot is continuous, discrete or factorial
  switch(vartype,
         continuous={contxaxis <- T},
         discrete={discxaxis <- T},
         factorial={factxaxis <- T},
         {stop(sprintf("The fplotfun vartype parameter %s does not exist. Define as 'continuous', 'discrete' or 'factorial'.", vartype))}
  )

  {if(contxaxis) fun.fun2 <- as.numeric}
  {if(factxaxis|discxaxis) fun.fun2 <- as.factor}

  #ggplot object
    ggplot(data,aes(x=fun.fun2(get(varname)),color=group,fill=group,group=fun.fun2(group)))+

  #ggplot condiment
    {if(contxaxis)list(fun.fun(breaks=seq(min,max,by=gap1),stat='bin',position="identity", alpha=0.5),scale_x_continuous(breaks=seq(min,max,by=gap2)))}+
    {if(discxaxis)list(fun.fun(position="identity", alpha=0.5),scale_x_discrete(breaks=seq(min,max,by=gap2),expand = expansion(add = 1.5)))} +
    {if(factxaxis)list(fun.fun(position=position_dodge(width=0.8),alpha=0.5),geom_text(stat="count",aes(label=..count..),position=position_dodge(0.9),vjust=-1,hjust=0.5,cex=2.3))} +

    theme_cowplot()+
    scale_y_continuous(expand = c(0,0,0.1,0)) +
    scale_color_manual(values=c("#666666", "#FF3300"))+
    scale_fill_manual(values=c("#666666", "#FF3300"))+
    ggtitle(maintitle)+
    xlab(xlabt)+
    theme(plot.title=element_text(hjust=0.5))+
     theme(plot.subtitle = element_text(hjust = 0.5))+
    theme(axis.text.x=element_text(size=8))+
    theme(axis.text.y=element_text(size=8))+
    theme(axis.title.x=element_text(size=12,vjust=-1))+
    theme(axis.title.y=element_text(size=12))+
    theme(plot.margin = unit(c(0.5,0.5,3,0.5), "lines"))

 }
```

Plots, general

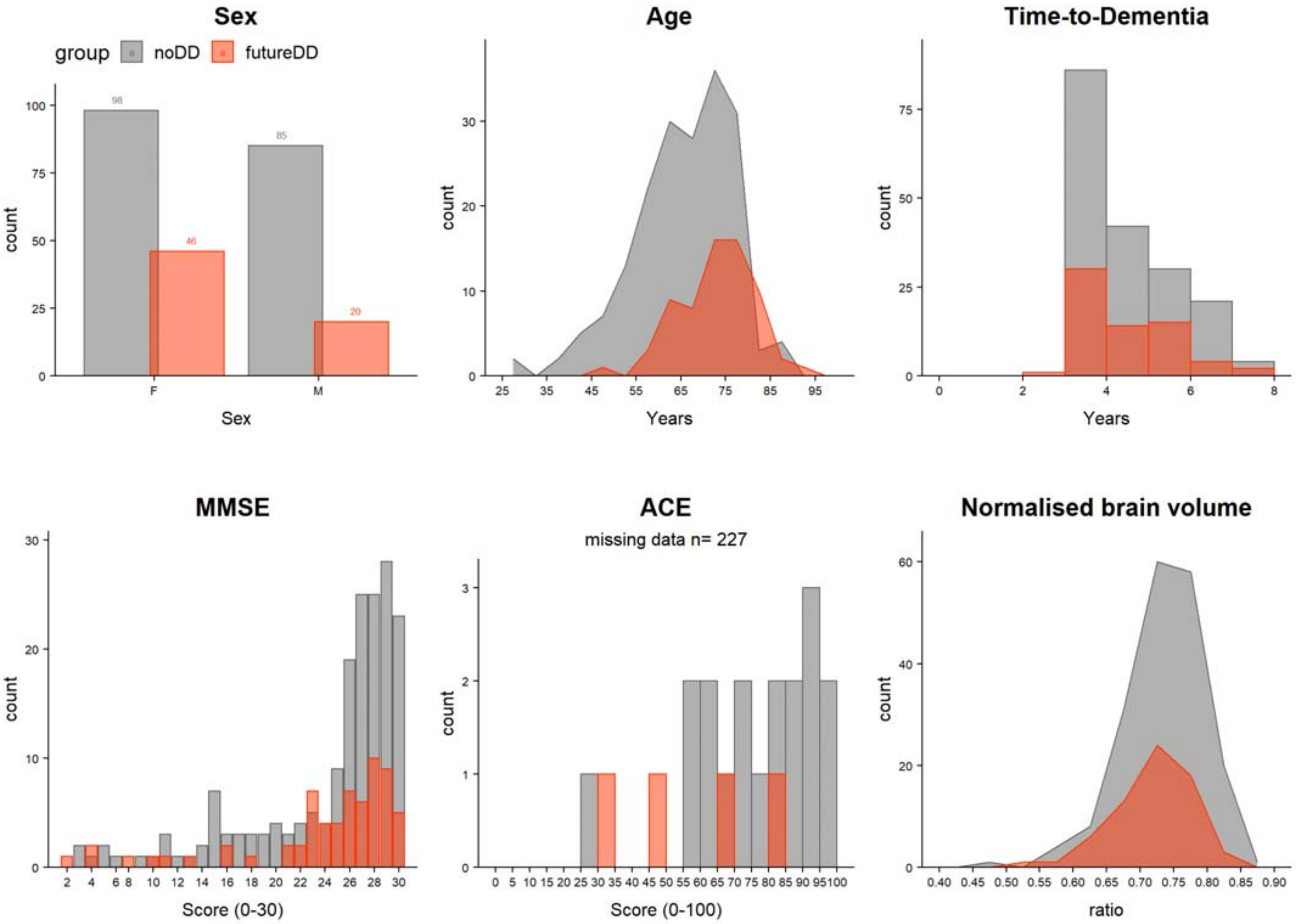
```
#plots
age_plot <- fplotfun(datasub2,"age",25,100,5,10,"Age","Years",geom_area,vartype='continuous')

brainPAD_plot <- fplotfun(datasub2,"brainPAD",-50,50,5,10,"BrainPAD","Years",geom_area,vartyp
e='continuous')

time2diag_plot <- fplotfun(datasub2,"End_scan2Endyears",0,8,1,2,"Time-to-Dementia","Years",ge
om_histogram,vartype='continuous')

MMSEp_plot <- fplotfun(datasub2,"MMSE_numerator",0,30,1,2,"MMSE","Score (0-30)",geom_bar,vart
ype='discrete')

sex_plot <- fplotfun(datasub2,"sex",0,0,0,0,"Sex","Sex",geom_bar,vartype='factorial')

normVol_plot <- fplotfun(datasub2,"normVol",0.4,0.9,0.05,0.05,"Normalised brain volume","rati
o",geom_area,vartype='continuous')

ACEnum_plot <- fplotfun(datasub2,"ACEnum",0,100,5,5,"ACE","Score (0-100)",geom_histogram,vart
ype='continuous')+ labs(subtitle = paste( "missing data n=",sum(is.na(datasub2$ACEnum))))



#ethnicity
#rename levels to remove redundant letter
levels(datasub2$ethnicity)
```

```
##  [1] "African (N)"                      "Any other Asian background (L)"
##  [3] "Any other black background (P)" "Any other ethnic group (S)"
##  [5] "Any other mixed background (G)" "Any other white background (C)"
##  [7] "Bangladeshi (K)"                "British (A)"
##  [9] "Caribbean (M)"                  "Chinese (R)"
## [11] "Indian (H)"                     "Irish (B)"
## [13] "Not Stated (Z)"                 "NULL"
## [15] "Pakistani (J)"                  "White and Black African (E)"
## [17] "White and black Caribbean (D)"
```

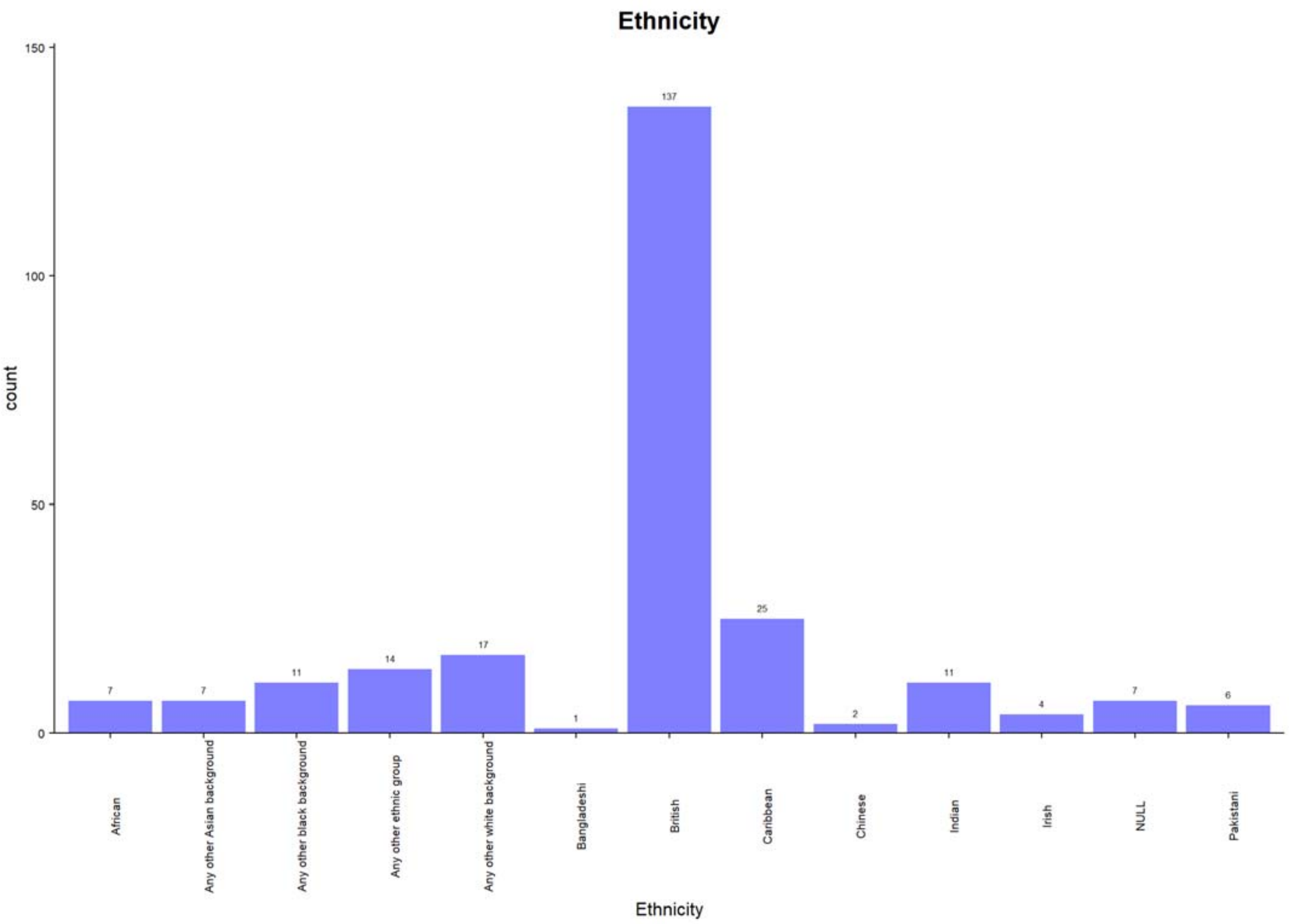
```
levels(datasub2$ethnicity) <- list("African" = "African (N)",
                                   "Any other Asian background"= "Any other Asian background
(L)",
                                   "Any other black background"="Any other black background
(P)",
                                   "Any other ethnic group"="Any other ethnic group (S)",
                                   "Any other mixed background"="Any other mixed background
(G)",
                                   "Any other white background"="Any other white background
(C)",
                                   "Bangladeshi"="Bangladeshi (K)",
                                   "British"="British (A)",
                                   "Caribbean" ="Caribbean (M)",
                                   "Chinese"="Chinese (R)",
                                   "Indian"="Indian (H)",
                                   "Irish"="Irish (B)",
                                   "Not Stated"="Not Stated (Z)",
                                   "NULL"="NULL",
                                   "Pakistani"="Pakistani (J)",
                                   "White and Black African"= "White and Black African (E)",
                                   "White and Black Caribbean" ="White and black Caribbean
(D)")

levels(datasub2$ethnicity)
```

```
##  [1] "African"                    "Any other Asian background"
##  [3] "Any other black background" "Any other ethnic group"
##  [5] "Any other mixed background" "Any other white background"
##  [7] "Bangladeshi"                "British"
##  [9] "Caribbean"                  "Chinese"
## [11] "Indian"                     "Irish"
## [13] "Not Stated"                 "NULL"
## [15] "Pakistani"                  "White and Black African"
## [17] "White and Black Caribbean"
```

```
ethnicity_plot <- ggplot(datasub2,aes(x=as.factor(ethnicity)))+
  geom_bar(position=position_dodge(width=0.8),alpha=0.5,fill="blue")+
  geom_text(stat="count",aes(label=..count..),position=position_dodge(0.9),vjust=-1,hjust=0.
5,cex=2.3) +
   theme_cowplot()+
    scale_y_continuous(expand = c(0,0,0.1,0)) +
    ggtitle("Ethnicity")+
    xlab("Ethnicity")+
    theme(plot.title=element_text(hjust=0.5))+
    theme(plot.subtitle = element_text(hjust = 0.5))+
    theme(axis.text.x=element_text(size=8,angle=90))+
    theme(axis.text.y=element_text(size=8))+
    theme(axis.title.x=element_text(size=12,vjust=-1))+
    theme(axis.title.y=element_text(size=12))+
    theme(plot.margin = unit(c(0.5,0.5,3,0.5), "lines"))


# plot together using cowplot function
all_plots <- cowplot::plot_grid(sex_plot+ theme(legend.position = "top"),age_plot+ theme(lege
nd.position = "none"),
                                time2diag_plot+ theme(legend.position = "none"),MMSEp_plot+ t
heme(legend.position = "none"),
                                ACEnum_plot+ theme(legend.position = "none"),normVol_plot+ th
eme(legend.position = "none"),
                                labels = "", nrow = 2)


all_plots
```



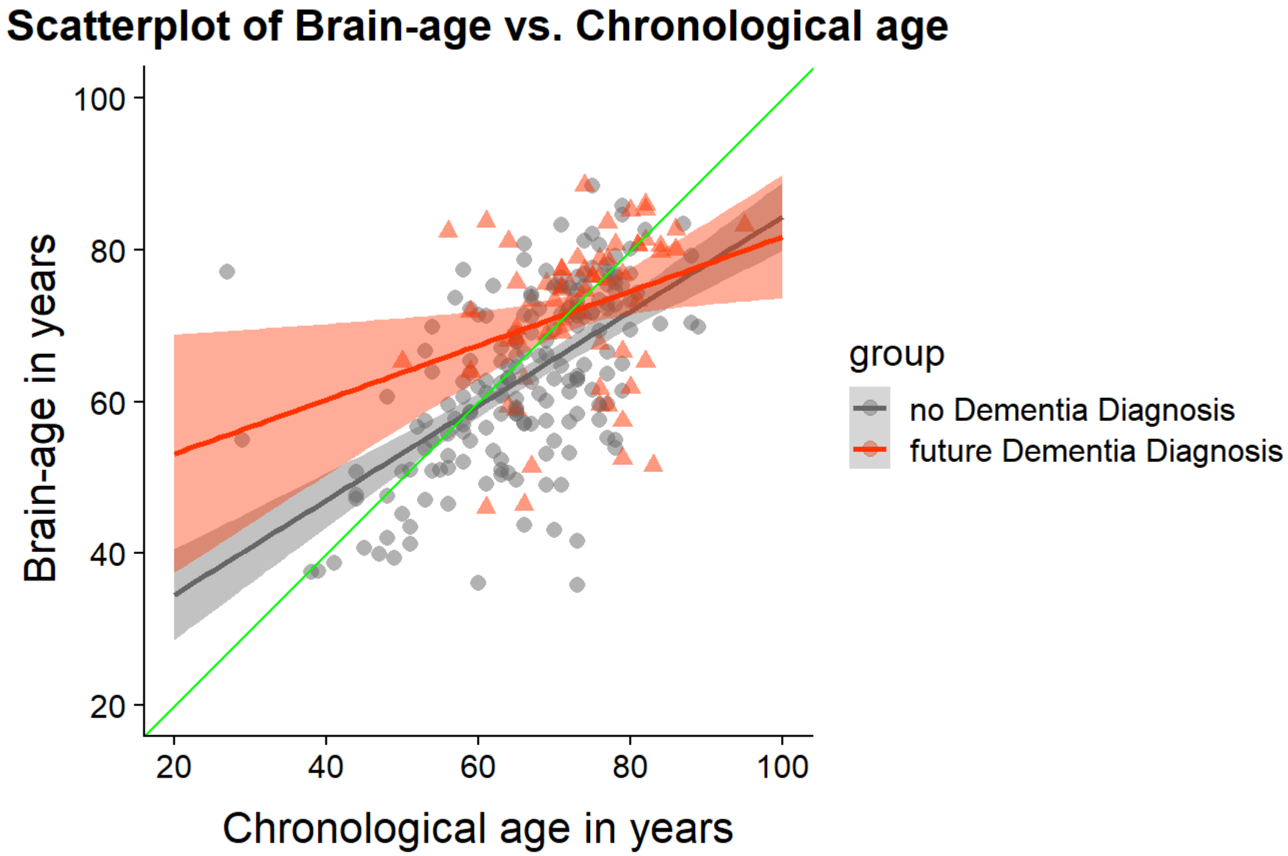
```
ethnicity_plot
```



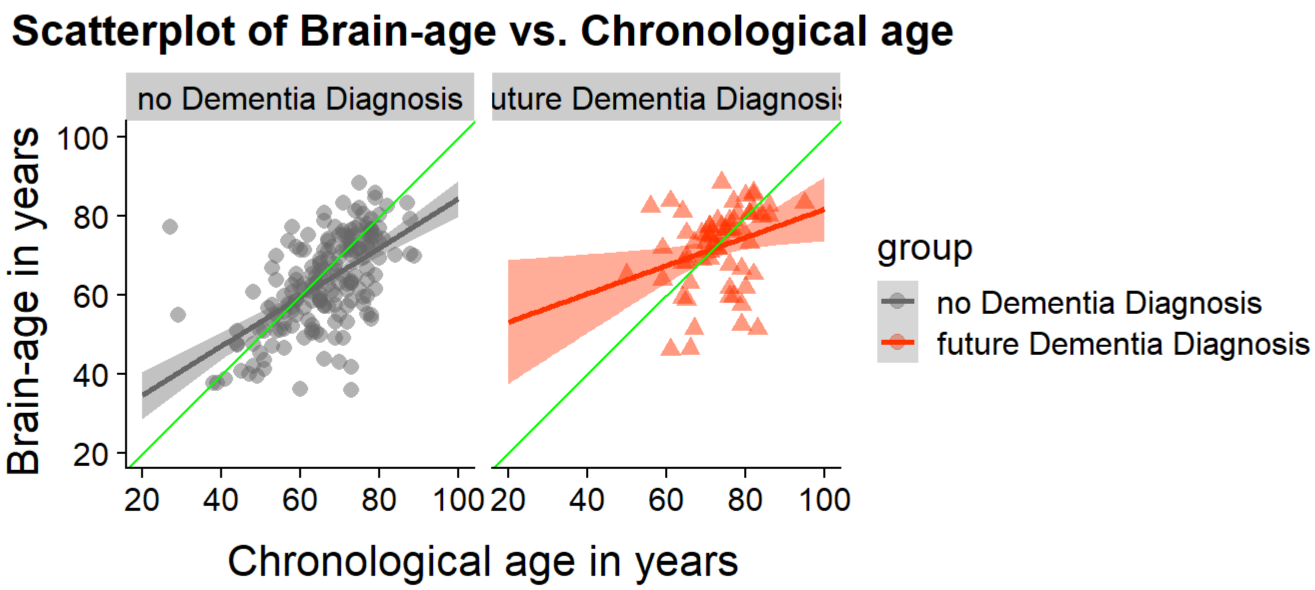
```
#NOTE: for brainPAD plot, see Fig 2

#save pic
filename <- paste(outputpath,"Fig0_allplots_",versionname,".tif",sep="")
ggsave(filename = filename,height = 9, width = 12, print(all_plots, newpage = FALSE), device
= "tiff", dpi = picres, units = "in")

filename <- paste(outputpath,"Fig0_ethnicity",versionname,".tif",sep="")
ggsave(filename = filename,height = 4.5, width = 8, print(ethnicity_plot, newpage = FALSE), d
evice = "tiff", dpi = picres, units = "in")
```

Figure 2: brain-age vs chronological age scatterplot

```
brainage_plot1 <- ggplot(datasub2,aes(x=as.numeric(age),y=as.numeric(brain.predictedage), color=group, fill=group,shape=group))+

  geom_point(position="identity", alpha=0.5, size=2.5 )+
  coord_fixed(ratio=1)+theme_cowplot()+
  geom_smooth(method='lm',se=TRUE,fullrange=TRUE)+
  geom_abline(slope =1,intercept=0,color='green')+
  scale_x_continuous(breaks=seq(20,100,by=10))+
  scale_y_continuous(breaks=seq(20,100,by=10))+
  theme(plot.title=element_text(hjust=0.5))+
  scale_color_manual(values=c("#666666", "#FF3300"),labels = c("no Dementia Diagnosis", "future Dementia Diagnosis"))+
  scale_fill_manual(values=c("#666666", "#FF3300"))+
  xlim(20,100)+ylim(20,100)+
  ggtitle("Scatterplot of Brain-age vs. Chronological age")+
  xlab("Chronological age in years")+ ylab("Brain-age in years ")+
  theme(axis.text.x=element_text(size=12))+
  theme(axis.text.y=element_text(size=12))+
  theme(axis.title.x=element_text(size=16,vjust=-1))+
  theme(axis.title.y=element_text(size=16))+
  theme(plot.margin = unit(c(0.5,0.5,3,0.5), "lines"))+
  guides(fill = FALSE,color=guide_legend("group"),shape=F)

brainage_plot1
```



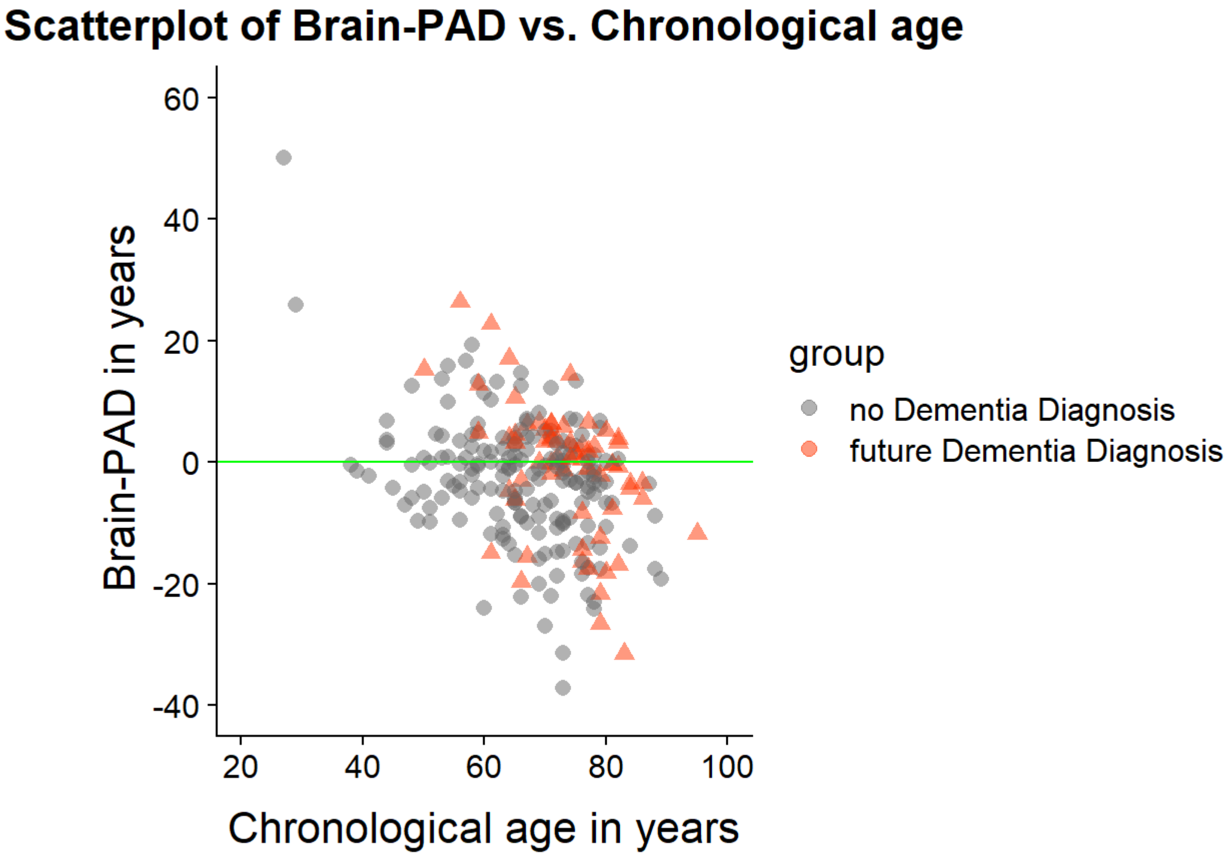
```
###modify legend labels in facet_grid below
legend_names <- list(
  'noDD'="no Dementia Diagnosis",
  'futureDD'="future Dementia Diagnosis"
)
DD_labeller <- function(variable,value){
  return(legend_names[value])
}


brainage_plot2 <- brainage_plot1 + facet_grid(. ~ group,labeller=DD_labeller)
brainage_plot2
```



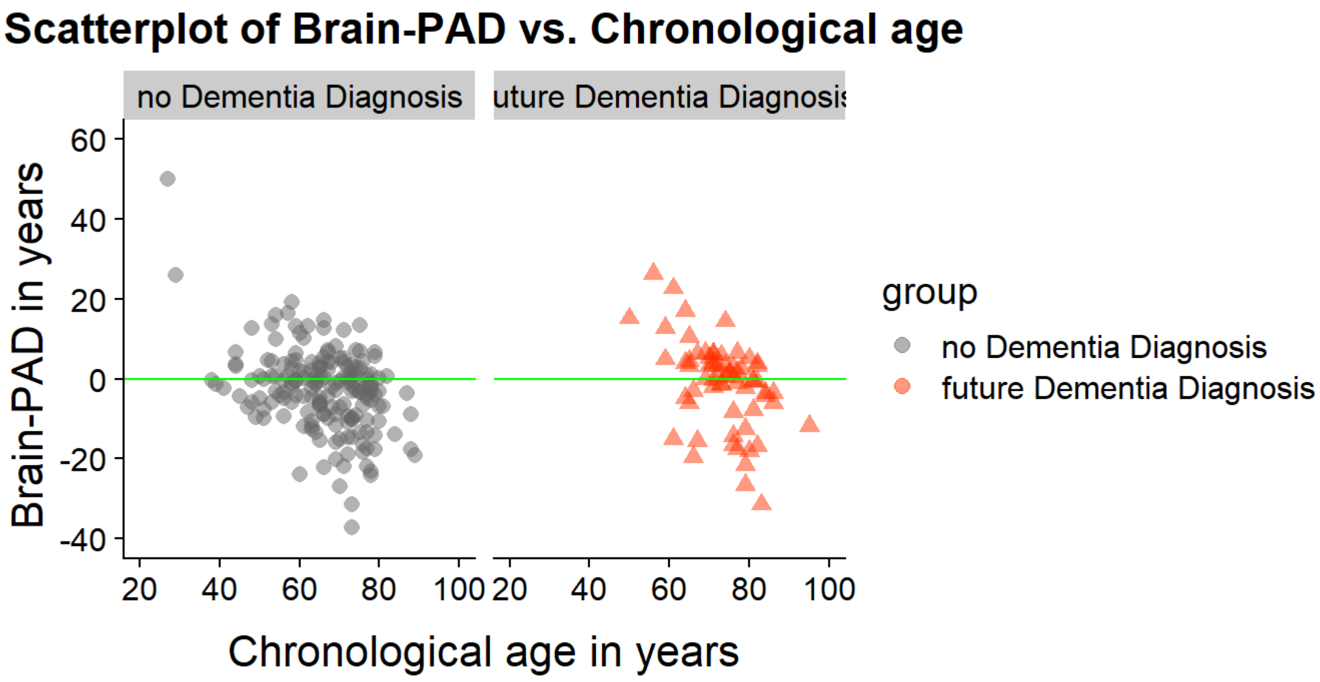
```
# save
filename <- paste(outputpath,"Fig2_facet_scatter_brainage_",versionname,".tif",sep="")
ggsave(brainage_plot2,width=300, height=300, dpi=picres, device = "tiff",filename = filename,
units = 'mm')
```

Figure 3: brain-PAD vs chronological age scatterplot

```
brainPAD_plot2 <- ggplot(datasub2,aes(x=as.numeric(age),y=as.numeric(brainPAD), color=group,
fill=group,shape=group))+

  geom_point(position="identity", alpha=0.5, size=2.5 )+
  coord_fixed(ratio=1)+theme_cowplot()+
  #geom_smooth(method='lm',se=TRUE,fullrange=TRUE)+
  geom_abline(slope =0,intercept=0,color='green')+
  scale_x_continuous(breaks=seq(20,100,by=10))+
  scale_y_continuous(breaks=seq(-40,60,by=10))+
  theme(plot.title=element_text(hjust=0.5))+
  scale_color_manual(values=c("#666666", "#FF3300"),labels = c("no Dementia Diagnosis", "futu
re Dementia Diagnosis"))+
  scale_fill_manual(values=c("#666666", "#FF3300"))+
  xlim(20,100)+ylim(-40,60)+
  ggtitle("Scatterplot of Brain-PAD vs. Chronological age")+
  xlab("Chronological age in years")+ ylab("Brain-PAD in years ")+
  theme(axis.text.x=element_text(size=12))+
  theme(axis.text.y=element_text(size=12))+
  theme(axis.title.x=element_text(size=16,vjust=-1))+
  theme(axis.title.y=element_text(size=16))+
  theme(plot.margin = unit(c(0.5,0.5,3,0.5), "lines"))+
  guides(fill = FALSE,color=guide_legend("group"),shape=F)

brainPAD_plot2
```

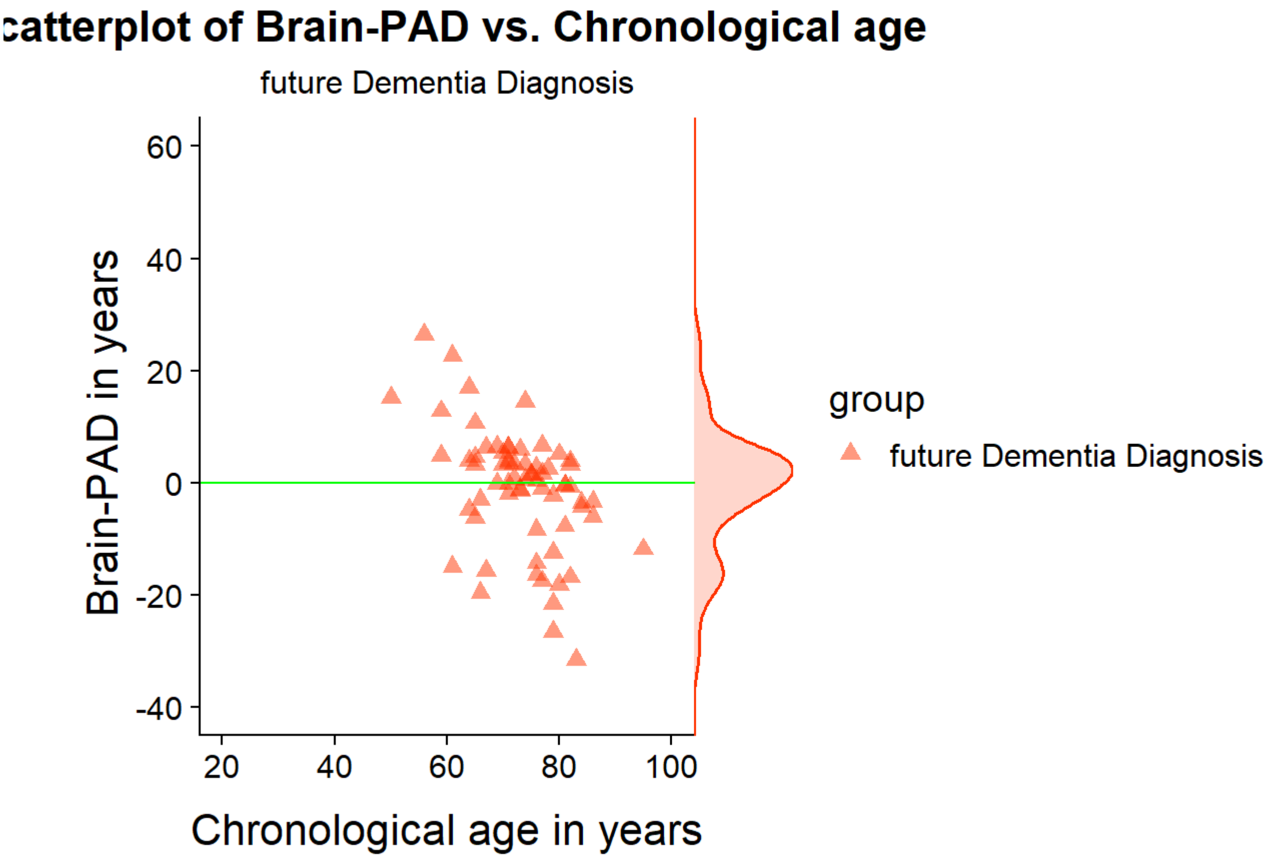


```
brainPAD_plot3 <- brainPAD_plot2 + facet_grid(. ~ group,labeller=DD_labeller)
brainPAD_plot3
```

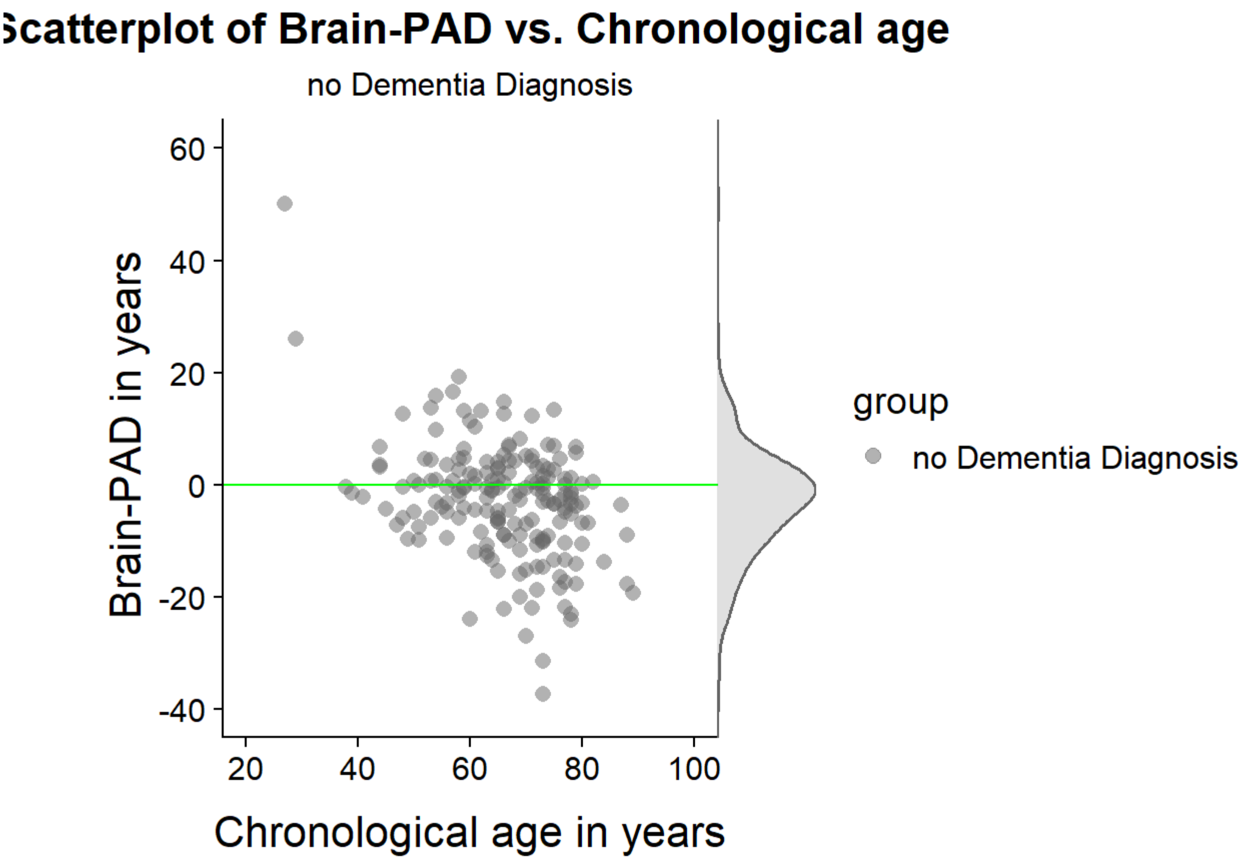


```
# save
filename <- paste(outputpath,"Fig3_facet_scatter_brainPAD_",versionname,".tif",sep="")
ggsave(brainPAD_plot3,width=300, height=300, dpi=picres, device = "tiff",filename = filename,
units = 'mm')


####################
# Marginal densities along y axis
# Need to set coord_flip = TRUE, if you plan to use coord_flip()
ydens <- axis_canvas(brainPAD_plot2, axis = "y", coord_flip = TRUE)+
  geom_density(data= datasub2, aes(x = brainPAD, fill = group, colour=group),
               alpha = 0.2, size = 0.7)+
  coord_flip()+
  scale_fill_manual(values=c("#666666", "#FF3300"))+
  scale_color_manual(values=c("#666666", "#FF3300"))



p2 <- insert_yaxis_grob(brainPAD_plot2, ydens, grid::unit(.2, "null"), position = "right")
#ggdraw(p2)


################### separate plots + density plots
#subset data
  datasub2D <- datasub2[datasub2$group =='futureDD',]
  datasub2noD <- datasub2[datasub2$group =='noDD',]

#Dementia ggroup
brainPAD_plot2a <- ggplot(datasub2D,aes(x=as.numeric(age),y=as.numeric(brainPAD), color=grou
p, fill=group,shape=group))+

  geom_point(position="identity", alpha=0.5, size=2.5, shape=17 )+
  coord_fixed(ratio=1)+theme_cowplot()+
  #geom_smooth(method='lm',se=TRUE,fullrange=TRUE)+
  geom_abline(slope =0,intercept=0,color='green')+
  scale_x_continuous(breaks=seq(20,100,by=10))+
  scale_y_continuous(breaks=seq(-40,60,by=10))+
  theme(plot.title=element_text(hjust=0.5))+
  scale_color_manual(values=c("#FF3300"),labels = c("future Dementia Diagnosis"))+
  scale_fill_manual(values=c("#FF3300"))+
  xlim(20,100)+ylim(-40,60)+
  ggtitle("Scatterplot of Brain-PAD vs. Chronological age")+
  labs(subtitle = "future Dementia Diagnosis")+
  theme(plot.subtitle = element_text(hjust = 0.5))+
  xlab("Chronological age in years")+ ylab("Brain-PAD in years ")+
  theme(axis.text.x=element_text(size=12))+
  theme(axis.text.y=element_text(size=12))+
  theme(axis.title.x=element_text(size=16,vjust=-1))+
  theme(axis.title.y=element_text(size=16))+
  theme(plot.margin = unit(c(0.5,0.5,3,0.5), "lines"))+
  guides(fill = FALSE,color=guide_legend("group"),shape=F)
```

```
#brainPAD_plot2a

####################
# Marginal densities along y axis
# Need to set coord_flip = TRUE, if you plan to use coord_flip()
ydens2 <- axis_canvas(brainPAD_plot2a, axis = "y", coord_flip = TRUE)+
  geom_density(data= datasub2D, aes(x = brainPAD, fill = group, colour=group),
               alpha = 0.2, size = 0.7)+
  coord_flip()+
  scale_fill_manual(values=c("#FF3300"))+
  scale_color_manual(values=c("#FF3300"))

brainPAD_plot2ai<- insert_yaxis_grob(brainPAD_plot2a, ydens, grid::unit(.2, "null"), position
= "right")
#ggdraw(brainPAD_plot2ai)

brainPAD_plot2aii <- insert_yaxis_grob(brainPAD_plot2a, ydens2, grid::unit(.2, "null"), posit
ion = "right")
ggdraw(brainPAD_plot2aii)
```



```
#no Dementia group
brainPAD_plot2b <- ggplot(datasub2noD,aes(x=as.numeric(age),y=as.numeric(brainPAD), color=group, fill=group,shape=group))+

  geom_point(position="identity", alpha=0.5, size=2.5 )+
  coord_fixed(ratio=1)+theme_cowplot()+
  #geom_smooth(method='lm',se=TRUE,fullrange=TRUE)+
  geom_abline(slope =0,intercept=0,color='green')+
  scale_x_continuous(breaks=seq(20,100,by=10))+
  scale_y_continuous(breaks=seq(-40,60,by=10))+
  theme(plot.title=element_text(hjust=0.5))+
  scale_color_manual(values=c("#666666"),labels = c("no Dementia Diagnosis"))+
  scale_fill_manual(values=c("#666666"))+
  xlim(20,100)+ylim(-40,60)+
  ggtitle("Scatterplot of Brain-PAD vs. Chronological age")+
  labs(subtitle = "no Dementia Diagnosis")+
  theme(plot.subtitle = element_text(hjust = 0.5))+
  xlab("Chronological age in years")+ ylab("Brain-PAD in years ")+
  theme(axis.text.x=element_text(size=12))+
  theme(axis.text.y=element_text(size=12))+
  theme(axis.title.x=element_text(size=16,vjust=-1))+
  theme(axis.title.y=element_text(size=16))+
  theme(plot.margin = unit(c(0.5,0.5,3,0.5), "lines"))+
  guides(fill = FALSE,color=guide_legend("group"),shape=F)

#brainPAD_plot2b

####################
# Marginal densities along y axis
# Need to set coord_flip = TRUE, if you plan to use coord_flip()
ydens3 <- axis_canvas(brainPAD_plot2b, axis = "y", coord_flip = TRUE)+
  geom_density(data= datasub2noD, aes(x = brainPAD, fill = group, colour=group),
               alpha = 0.2, size = 0.7)+
  coord_flip()+
  scale_fill_manual(values=c("#666666"))+
  scale_color_manual(values=c("#666666"))

brainPAD_plot2bii<- insert_yaxis_grob(brainPAD_plot2b, ydens3, grid::unit(.2, "null"), position = "right")
ggdraw(brainPAD_plot2bii)
```



```
wh=200 #width and height
# save
filename <- paste(outputpath,"Fig3aBOUND_scatter_brainPAD_withdensity_futureDD_",versionnam
e,".tif",sep="")
ggsave(brainPAD_plot2ai,width=wh, height=wh, dpi=picres, device = "tiff",filename = filename,
units = 'mm')

filename <- paste(outputpath,"Fig3bBOUND_scatter_brainPAD_withdensity_noDD_",versionname,".ti
f",sep="")
ggsave(brainPAD_plot2b,width=wh, height=wh, dpi=picres, device = "tiff",filename = filename,u
nits = 'mm')


filename <- paste(outputpath,"Fig3aSEP_scatter_brainPAD_withdensity_futureDD_",versionname,".
tif",sep="")
ggsave(brainPAD_plot2aii,width=wh, height=wh, dpi=picres, device = "tiff",filename = filenam
e,units = 'mm')

filename <- paste(outputpath,"Fig3bSEP_scatter_brainPAD_withdensity_noDD_",versionname,".ti
f",sep="")
ggsave(brainPAD_plot2bii,width=wh, height=wh, dpi=picres, device = "tiff",filename = filenam
e,units = 'mm')
```

Table 1 function

```r
ftablefun <- function(data,bygroup) {
options(scipen = 999)

# extract the precise summary stats I want
table_summary <- data %>%
  # select variables you want in demographics table
  select(group, age, sex, MMSE_numerator, End_scan2Endyears,ACEnum,normVol) %>%

  # calculate demographics by group if applicable
  {if(bygroup) group_by(.,group) else .}%>%

  # get summary statistics
  summarise(N = n(),
            NFEMALE = sum(sex == "F"),
            NMALE = sum(sex == "M"),
            PFEMALE= (NFEMALE/(NFEMALE+NMALE)*100),
            AGE_PCOUNT=(((sum(!is.na(age)))/N)*100),
            AGE_MEAN = mean(age,na.rm=TRUE),
            AGE_SD=sd(age,na.rm=TRUE),
            AGE_MIN=min(age,na.rm=TRUE),
            AGE_MAX=max(age,na.rm=TRUE),
            AGE_MD=median(age,na.rm=TRUE),
            TIME_PCOUNT=(((sum(!is.na(End_scan2Endyears)))/N)*100),
            TIME_MEAN = mean(End_scan2Endyears,na.rm=TRUE),
            TIME_SD=sd(End_scan2Endyears,na.rm=TRUE),
            TIME_MIN=min(End_scan2Endyears,na.rm=TRUE),
            TIME_MAX=max(End_scan2Endyears,na.rm=TRUE),
            TIME_MD=median(End_scan2Endyears,na.rm=TRUE),
            MMSE_PCOUNT=(((sum(!is.na(MMSE_numerator)))/N)*100),
            MMSE_MEAN = mean(MMSE_numerator,na.rm=TRUE),
            MMSE_SD=sd(MMSE_numerator,na.rm=TRUE),
            MMSE_MIN=min(MMSE_numerator,na.rm=TRUE),
            MMSE_MAX=max(MMSE_numerator,na.rm=TRUE),
            MMSE_MD=median(MMSE_numerator,na.rm=TRUE),
            ACE_PCOUNT=(((sum(!is.na(ACEnum)))/N)*100),
            ACE_MEAN = mean(ACEnum,na.rm=TRUE),
            ACE_SD=sd(ACEnum,na.rm=TRUE),
            ACE_MIN=min(ACEnum,na.rm=TRUE),
            ACE_MAX=max(ACEnum,na.rm=TRUE),
            ACE_MD=median(ACEnum,na.rm=TRUE),
            NVOL_PCOUNT=(((sum(!is.na(normVol)))/N)*100),
            NVOL_MEAN = mean(normVol,na.rm=TRUE),
            NVOL_SD=sd(normVol,na.rm=TRUE),
            NVOL_MIN=min(normVol,na.rm=TRUE),
            NVOL_MAX=max(normVol,na.rm=TRUE),
            NVOL_MD=median(normVol,na.rm=TRUE))

# add group column if missing
{if(bygroup==F) table_summary$group <- 'All'}
```

```
# Table1 part A - round to 2d.p
tableA <- table_summary %>%

  # Round-up measures to 2 dp
  mutate_at(c("PFEMALE","AGE_PCOUNT","AGE_MEAN","AGE_SD","AGE_MIN","AGE_MAX","AGE_MD",
            "TIME_PCOUNT","TIME_MEAN","TIME_SD","TIME_MIN","TIME_MAX","TIME_MD",
            "MMSE_PCOUNT","MMSE_MEAN","MMSE_SD","MMSE_MIN","MMSE_MAX","MMSE_MD",
            "ACE_PCOUNT","ACE_MEAN","ACE_SD","ACE_MIN","ACE_MAX","ACE_MD","NVOL_PCOUNT"), ~fo
rmat(round(., 2), nsmall = 2))%>%

  # Create new variables and paste the mean and sd together as a string to have Mean (SD) in
the same table cell
  mutate(`Female %` = paste(PFEMALE),
        `Age Complete cases %` = paste(AGE_PCOUNT),
        `Age Mean (SD)` = paste(AGE_MEAN, paste(" (", AGE_SD, ") ", sep = "")),
        `Age Median` = paste(AGE_MD),
        `Age Min-Max` = paste(AGE_MIN,"-",AGE_MAX, sep = ""),

        `MMSE Complete cases %` = paste(MMSE_PCOUNT),
        `MMSE Mean (SD)` = paste(MMSE_MEAN, paste(" (", MMSE_SD, ") ", sep = "")),
        `MMSE Median` = paste(MMSE_MD),
        `MMSE Min-Max` = paste(MMSE_MIN,"-",MMSE_MAX, sep = ""),

        `Time-to-Event Complete cases %` = paste(TIME_PCOUNT),
        `Time-to-Event Mean (SD)` = paste(TIME_MEAN, paste(" (", TIME_SD, ") ", sep = "")),
        `Time-to-Event Median` = paste(TIME_MD),
        `Time-to-Event Min-Max` = paste(TIME_MIN,"-",TIME_MAX, sep = ""),

        `ACE Complete cases %` = paste(ACE_PCOUNT),
        `ACE Mean (SD)` = paste(ACE_MEAN, paste(" (", ACE_SD, ") ", sep = "")),
        `ACE Median` = paste(ACE_MD),
        `ACE Min-Max` = paste(ACE_MIN,"-",ACE_MAX, sep = ""),

         `Normalised Brain Volume Complete cases %` = paste(NVOL_PCOUNT),)%>%

  # select our presentable variables for the table
  select(`Group`=group,
         N, `Female %`,
         `Age Complete cases %` ,`Age Mean (SD)`, `Age Median`, `Age Min-Max`,
         `MMSE Complete cases %`,`MMSE Mean (SD)`,`MMSE Median`,`MMSE Min-Max`,
         `Time-to-Event Complete cases %`,`Time-to-Event Mean (SD)`,`Time-to-Event Median`,`T
ime-to-Event Min-Max`,
         `ACE Complete cases %`,`ACE Mean (SD)` ,`ACE Median`,`ACE Min-Max`,`Normalised Brain
Volume Complete cases %`)


# Table2 part B - round to 4d.p
tableB <- table_summary %>%

  # Round-up measures to 4dp
  mutate_at(c("NVOL_MEAN","NVOL_SD","NVOL_MIN","NVOL_MAX","NVOL_MD"), ~format(round(., 4), ns
mall = 4))%>%

  # Create new variables and paste the mean and sd together as a string to have Mean (SD) in
the same table cell
  mutate(`Normalised Brain Volume Mean (SD)` = paste(NVOL_MEAN, paste(" (", NVOL_SD, ") ", se
p = "")),
        `Normalised Brain Volume Median` = paste(NVOL_MD),
        `Normalised Brain Volume Min-Max` = paste(NVOL_MIN,"-",NVOL_MAX, sep = ""),)%>%



  # select our presentable variables for the table
  select(`Normalised Brain Volume Mean (SD)` ,`Normalised Brain Volume Median`,`Normalised Br
ain Volume Min-Max`)

# Bind tables A and B
tableAB <- cbind(tableA,tableB)

# These are some options for the pander library to output the table
panderOptions('table.split.table', 100)

# this command outputs the table with the caption
#pander(tableAB)
options(scipen = 0)
return (tableAB)
}
```

Table 1: data characteristics

```r
table1 <- ftablefun(datasub2,bygroup=F)
table2 <- ftablefun(datasub2,bygroup=T)

# Bind tables 1 &2
table3 <- rbind(table1,table2)


#transpose & adjust col names
table3t <- t(table3)%>%
as.data.table(., keep.rownames = T)
cnames <- as.character(table3t[1, ])
colnames(table3t) <- cnames
table3t <- table3t[-1, ]

#run t-tests for relevant variables
ttest_AGE <- t.test(datasub2$age~datasub2$group)
ttest_MMSE <- t.test(datasub2$MMSE_numerator~datasub2$group)
ttest_TIME <- t.test(datasub2$End_scan2Endyears~datasub2$group)
ttest_ACE <- t.test(datasub2$ACEnum~datasub2$group)
ttest_NVOL <- t.test(datasub2$normVol~datasub2$group)

#create num version of sex variable
datasub2 <- datasub2%>%
  mutate('sexNum'=as.factor(sex))%>%
  mutate('sexNum'=as.numeric(sexNum))

#wtest_SEX <- wilcox.test(as.numeric(sex)~group,data=datasub2) #report median
wtest_SEX <- wilcox.test(sexNum~group,data=datasub2) #report median

#add column
table4 <- table3t %>%
  add_column('noDD vs. futureDD (p-value)' = 'NA', .after = "futureDD")


#write function to convert p-values: round to 4dp and if lower than 0.0001 to '<0.0001'
col_pval <- function(df) {
  output <- vector("double", length(df))
  for (i in seq_along(df)) {
    if(as.numeric(df[[i]]) < 0.0001){
      output[i] <- ("<0.0001")
    } else {
      output[i] <- format(round(df[[i]], 4), nsmall = 4)
    }
  }
  output
}


#fill column with t/w-test p-values
table4[table4$Group=="Age Mean (SD)",'noDD vs. futureDD (p-value)'] <- col_pval(ttest_AGE$p.value)
```

```
table4[table4$Group=="MMSE Mean (SD)",'noDD vs. futureDD (p-value)'] <- col_pval(ttest_MMSE$p.value)
table4[table4$Group=="ACE Mean (SD)",'noDD vs. futureDD (p-value)'] <- col_pval(ttest_ACE$p.value)
table4[table4$Group=="Normalised Brain Volume Mean (SD)",'noDD vs. futureDD (p-value)'] <- col_pval(ttest_NVOL$p.value)
table4[table4$Group=="Female %",'noDD vs. futureDD (p-value)'] <- col_pval(wtest_SEX$p.value)

# These are some options for the pander library to output the table
panderOptions('table.split.table', 150)
# this command outputs the table with the caption
pander(table4 , caption = "(1) Sample characteristics")
```

(1) Sample characteristics

Group	All	noDD	futureDD	noDD vs. futureDD (p-value)
N	249	183	66	NA
Female %	57.83	53.55	69.70	0.0231
Age Complete cases %	100.00	100.00	100.00	NA
Age Mean (SD)	68.00 (10.79)	66.10 (11.01)	73.24 (8.14)	<0.0001
Age Median	70.00	67.00	74.00	NA
Age Min-Max	27.00-95.00	27.00-89.00	50.00-95.00	NA
MMSE Complete cases %	100.00	100.00	100.00	NA
MMSE Mean (SD)	24.35 (6.35)	24.50 (6.25)	23.94 (6.65)	0.5502
MMSE Median	27.00	27.00	26.00	NA
MMSE Min-Max	2.00-30.00	3.00-30.00	2.00-30.00	NA
Time-to-Event Complete cases %	100.00	100.00	100.00	NA
Time-to-Event Mean (SD)	4.40 (1.17)	4.42 (1.18)	4.35 (1.15)	NA
Time-to-Event Median	4.11	4.08	4.13	NA
Time-to-Event Min-Max	3.00-7.81	3.01-7.81	3.00-7.49	NA
ACE Complete cases %	8.84	9.84	6.06	NA
ACE Mean (SD)	73.27 (19.40)	76.61 (17.93)	58.25 (21.08)	0.1807
ACE Median	77.00	80.50	58.50	NA
ACE Min-Max	28.00-99.00	28.00-99.00	34.00-82.00	NA
Normalised Brain Volume Complete cases %	100.00	100.00	100.00	NA

Group	All	noDD	futureDD	noDD vs. futureDD (p-value)
Normalised Brain Volume Mean (SD)	0.7307 (0.0577)	0.7349 (0.0575)	0.7188 (0.0572)	0.0521
Normalised Brain Volume Median	0.7369	0.7392	0.7201	NA
Normalised Brain Volume Min-Max	0.4738-0.8629	0.4738-0.8629	0.5423-0.8204	NA

```
#save table to file for export
filename <- paste(outputpath,"Table1_",versionname,".xlsx",sep="")
write.xlsx(table4,filename, rowNames=T, colNames=T)
```

PART 4: Statistical Analysis

Logistic Regression

```
flogregfun <- function(data,Center,Scale,Poly) {

temp_data <- data%>%
  select(group, age, sex, MMSE_numerator, brainPAD, normVol)%>%
  mutate(`MMSE`=MMSE_numerator,
           MMSE_numerator=NULL)%>%

  {if(Poly) mutate(.,`age`=poly(age,2)[,1],`age2`=poly(age,2)[,2]) else .}%>%
  {if(Poly) relocate(.,age2, .after = age) else .}%>%
  mutate_if(is.numeric,~scale(.,center=Center,scale=Scale),na.rm = TRUE)


temp_model_log <- temp_data %>%
  glm(group ~.,family=binomial(link='logit'),data=.)
return(temp_model_log)
}

# run #Center, Scale and Poly settings set at start of code
model_log <- flogregfun (datasub2,Center=fCenter,Scale=fScale,Poly=fPoly)
summary(model_log)
```

```
##
## Call:
## glm(formula = group ~ ., family = binomial(link = "logit"), data = .)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5402  -0.7987  -0.5086   0.9285   2.3282
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.960969   0.217417  -4.420 9.87e-06 ***
## age         19.240740   4.607379   4.176 2.97e-05 ***
## age2        -1.855929   3.934332  -0.472  0.63712
## sexM        -0.871473   0.335953  -2.594  0.00949 **
## brainPAD     0.060438   0.018563   3.256  0.00113 **
## normVol     -0.247459   3.080523  -0.080  0.93597
## MMSE        -0.003421   0.024484  -0.140  0.88889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 287.99  on 248  degrees of freedom
## Residual deviance: 244.59  on 242  degrees of freedom
## AIC: 258.59
##
## Number of Fisher Scoring iterations: 6
```

```
# anova(model_log, test="Chisq")
# PseudoR2(model_log,c("McFadden","Nagel"))
```

Test assumptions of Logit

```
# ##### test if the distribution of residuals of Logit is normal
# #instead use binned distribution via binnedplot function (residuals should be within bound)
# #https://www.rdocumentation.org/packages/arm/versions/1.11-2/topics/binnedplot
# #https://easystats.github.io/performance/reference/binned_residuals.html
 x <- predict(model_log)
 y <- resid(model_log)
 binnedplot(x,y)
```

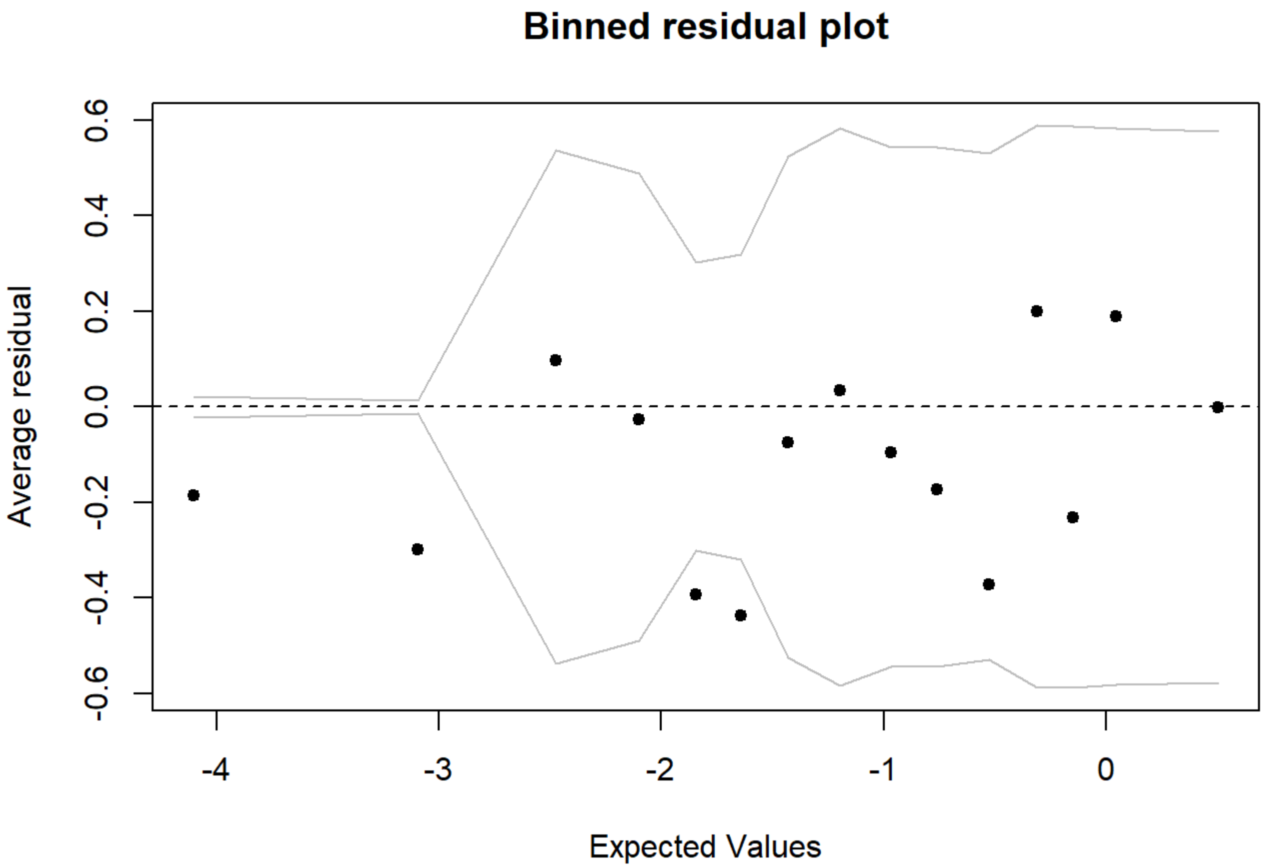


Table 2 function

```
flogtablefun <- function(data,type,Center,Scale,Poly){

  #recode type
  if (type=='log') {
    log <- T
  } else if ( type=='cox') {
    log <- F
  } else {
    stop(sprintf("Define second parameter, type, as either 'log' or 'cox'"))
  }


  #run logistic/cox regression function accordingly
  {if(log)  model <- flogregfun (data,Center,Scale,Poly) else model <- fcoxregfun (data,Center,Scale,Poly)[[1]]}

  #version labels
  if(Poly) {
    polystatus <- ' with quadratic age (polynomial)'
    polyshort <- 'PolyT'
  } else {
    polystatus <- ' without quadratic age'
    polyshort <- 'PolyF'
  }

  if (Center==T&Scale==T) {
    centeringstatus <- 'standardized'
    centeringshort <- 'CentZ'
  } else if ( Center==T&Scale==F) {
    centeringstatus <- 'centred'
    centeringshort <- 'CentT'
  } else if ( Center==F&Scale==F) {
    centeringstatus <- 'not-centred'
    centeringshort <- 'CentF'
  } else {
    stop(sprintf("Variable transformation in flogregfun embedded in this function can be either uncentred, centred or standardized (z-scores)"))
  }

  options(scipen = 999) #turn scientific notation off


  #select variables
  #first extract main stats from model
  modelcoef <- exp(coefficients(model))
  modelcoefCI <- exp(confint(model,level=0.95))
  {if(log) modelpvalues <- coef(summary(model))[,4] else modelpvalues <- coef(summary(model))[,5] }

  # then define variables
  #n/N (n=dementia case, N=total cases)
  {if(log) MOD_N <- nrow(model$data) else MOD_N <- model$n}
  {if(log) MOD_n <- sum(str_starts(model$data[['group']], pattern="futureDD", negate = FALSE)) else MOD_n <- model$nevent}

   #OR (odds ratio)
  MOD_ageCF <- modelcoef["age"]
  {if(Poly) MOD_age2CF <- modelcoef["age2"] }
  MOD_sexCF <- modelcoef["sexM"]
  MOD_brainPADCF <- modelcoef["brainPAD"]
  MOD_MMSECF <- modelcoef["MMSE"]
  MOD_normVolCF <- modelcoef["normVol"]

  {if(Poly) MOD_OR <- cbind.data.frame(MOD_ageCF,MOD_age2CF,MOD_sexCF,MOD_brainPADCF,MOD_MMSECF,MOD_normVolCF) else MOD_OR <- cbind.data.frame(MOD_ageCF,MOD_sexCF,MOD_brainPADCF,MOD_MMSECF,MOD_normVolCF) }

  #CI_OR lower
  MOD_ageCIlow <- modelcoefCI["age",1]
  {if(Poly) MOD_age2CIlow <- modelcoefCI["age2",1] }
  MOD_sexCIlow <- modelcoefCI["sexM",1]
  MOD_brainPADCIlow <- modelcoefCI["brainPAD",1]
  MOD_MMSECIlow <- modelcoefCI["MMSE",1]
  MOD_normVolCIlow <- modelcoefCI["normVol",1]

  {if(Poly)MOD_CIlow <- cbind.data.frame(MOD_ageCIlow,MOD_age2CIlow,MOD_sexCIlow,MOD_brainPADCIlow,MOD_MMSECIlow,MOD_normVolCIlow) else MOD_CIlow <- cbind.data.frame(MOD_ageCIlow,MOD_sexCIlow,MOD_brainPADCIlow,MOD_MMSECIlow,MOD_normVolCIlow)}


  #CI_OR upper
  MOD_ageCIup <- modelcoefCI["age",2]
  {if(Poly) MOD_age2CIup <- modelcoefCI["age2",2]}
  MOD_sexCIup <- modelcoefCI["sexM",2]
  MOD_brainPADCIup <- modelcoefCI["brainPAD",2]
  MOD_MMSECIup <- modelcoefCI["MMSE",2]
  MOD_normVolCIup <- modelcoefCI["normVol",2]

  {if(Poly) MOD_CIup <- cbind.data.frame(MOD_ageCIup,MOD_age2CIup,MOD_sexCIup,MOD_brainPADCIup,MOD_MMSECIup,MOD_normVolCIup) else MOD_CIup <- cbind.data.frame(MOD_ageCIup,MOD_sexCIup,MOD_brainPADCIup,MOD_MMSECIup,MOD_normVolCIup)}

  #pvalue
  MOD_agepval <- modelpvalues["age"]
  {if(Poly) MOD_age2pval <- modelpvalues["age2"]}
  MOD_sexpval <- modelpvalues["sexM"]
  MOD_brainPADpval <- modelpvalues["brainPAD"]
  MOD_MMSEpval <- modelpvalues["MMSE"]
  MOD_normVolpval <- modelpvalues["normVol"]

  {if(Poly) MOD_pval <- cbind.data.frame(MOD_agepval,MOD_age2pval,MOD_sexpval,MOD_brainPADpval,MOD_MMSEpval,MOD_normVolpval) else MOD_pval <- cbind.data.frame(MOD_agepval,MOD_sexpval,MOD
_brainPADpval,MOD_MMSEpval,MOD_normVolpval) }

  #round up
  #to ?dp  (DETERMINED AT START: SET_dp)
  MOD_ORr <- format(round(MOD_OR, SET_dp), nsmall = SET_dp)
  MOD_CIlowr <- format(round(MOD_CIlow, SET_dp), nsmall = SET_dp)
  MOD_CIupr <- format(round(MOD_CIup, SET_dp), nsmall = SET_dp)
  #to 4dp
  MOD_pvalr <- col_pval(MOD_pval) #calls function col_pval defined earlier in this script


  #presentation: "brainPAD n/N OR (CI) p-value"
  MOD_Nn  <-  paste(MOD_n,MOD_N, sep="/")
  MOD_ORCI <- paste(MOD_ORr, paste(" (", MOD_CIlowr, "-", MOD_CIupr, ") ", sep = ""))

  table2MOD <- rbind.data.frame(MOD_Nn,MOD_ORCI,MOD_pvalr)
  {if(Poly) names(table2MOD) <- c("age","age2","sex","brainPAD","MMSE","normVol") else names
(table2MOD) <- c("age","sex","brainPAD","MMSE","normVol")}
  {if(log) row.names(table2MOD) <- c("n/N","OR(CI)","p-value") else row.names(table2MOD) <- c
("n/N","HR(CI)","p-value") }


  #transpose & adjust col names
  table2MODt <- t(table2MOD)%>%
    as.data.table(., keep.rownames = T)


  # These are some options for the pander library to output the table
  panderOptions('table.split.table', 150)

  # define caption
  #fullversionname <- paste("(2) Logistic Regression Results, version = ",versionname, polyst
atus, " Vars are ", centeringstatus,sep="")
  {if(log) fullversionname <- paste("(2a) Logistic Regression Results, version = ",versionnam
e, polystatus, " Vars are ", centeringstatus,sep="") else fullversionname <- paste("(2b) Cox
PH Regression Results, version = ",versionname, polystatus, " Vars are ", centeringstatus,sep
="")}

  #define filename for saving to xls
  {if(log) shortname <- paste("Table2_LOG_",versionname,"_",polyshort,"_",centeringshort,".xl
sx",sep="") else shortname <- paste("Table2_COX_",versionname,"_",polyshort,"_",centeringshor
t,".xlsx",sep="") }

  # this command outputs the table with the caption
  pander(table2MODt , caption = fullversionname)

  options(scipen = 0) #turn scientific notation back on

  lt=list(table2MODt,fullversionname,shortname)
  return(lt)
}
```

Table 2a Log regression results

```
# subversion list
#1) PolyT, CentT = TFT: data is centred, age and age2 polynomials include
#2) PolyT, CentZ = TTT: data is standardized, age and age2 polynomials included
#3) PolyT, CentF = FFT: data is raw, age and age2 polynomials included
#4) PolyF, CentT = TFF: data is centred,age2 and poly excluded
#5) PolyF, CentZ = TTF: data is standardized, age2 and poly excluded
#6) PolyF, CentF = FFF: data is raw, age2 and poly excluded

subversion <- c("TFT","TTT", "FFT","TFF","TTF","FFF")

#function to convert TFT to logical
ftruefun <- function(x){
  {if(x=='T') T else F}
}

#loop through subversion
for (i in seq_along(subversion)){
  v <- subversion[i]

  table2LOG <- flogtablefun(datasub2,type='log',Center=ftruefun(substring(v,1,1)),Scale=ftrue
fun(substring(v,2,2)),Poly=ftruefun(substring(v,3,3)))

  table2LOG2 <- table2LOG[[1]] #extract table

  # print to console
  tt <- kbl(table2LOG2,caption=paste("Table",i," ",table2LOG[[2]])) %>%
  kable_classic(full_width = T, html_font = "Cambria",position = "left")%>%
  print(tt)
  cat("\n\n")

  nr <- nrow(table2LOG2)
  table2LOG2[[nr+1,1]] <- paste("Caption: ",table2LOG[[2]]) #add caption

  #save table to file for export
  write.xlsx(table2LOG2, paste(outputpath,table2LOG[[3]],sep=""), rowNames=T, colNames=T) #sa
ve to xls

}
```

Table 1 (2a) Logistic Regression Results, version = sensitivity1 with quadratic age (polynomial) Vars are centred

rn	n/N	OR(CI)	p-value
age	66/249	227063486.084891 (91257.098850-9320118318770.177734)	<0.0001
age2	66/249	0.156308 (0.000021-91.005837)	0.6371
sex	66/249	0.418335 (0.212484-0.797087)	0.0095
brainPAD	66/249	1.062302 (1.025635-1.103392)	0.0011
MMSE	66/249	0.996585 (0.950144-1.046726)	0.8889
normVol	66/249	0.780782 (0.002112-406.238978)	0.9360

Table 2 (2a) Logistic Regression Results, version = sensitivity1 with quadratic age (polynomial) Vars are standardized

rn	n/N	OR(CI)	p-value
age	66/249	3.393250 (2.065272-6.661301)	<0.0001
age2	66/249	0.888828 (0.503860-1.331684)	0.6371
sex	66/249	0.418335 (0.212484-0.797087)	0.0095
brainPAD	66/249	1.882383 (1.303321-2.800304)	0.0011
MMSE	66/249	0.978510 (0.722673-1.336468)	0.8889
normVol	66/249	0.985812 (0.700667-1.414653)	0.9360

Table 3 (2a) Logistic Regression Results, version = sensitivity1 with quadratic age (polynomial) Vars are not-centred

rn	n/N	OR(CI)	p-value
age	66/249	227063486.084892 (91257.098850-9320118318770.277344)	<0.0001
age2	66/249	0.156308 (0.000021-91.005837)	0.6371
sex	66/249	0.418335 (0.212484-0.797087)	0.0095
brainPAD	66/249	1.062302 (1.025635-1.103392)	0.0011
MMSE	66/249	0.996585 (0.950144-1.046726)	0.8889
normVol	66/249	0.780782 (0.002112-406.238978)	0.9360

Table 4 (2a) Logistic Regression Results, version = sensitivity1 without quadratic age Vars are centred

rn	n/N	OR(CI)	p-value
age	66/249	1.112583 (1.068569-1.163642)	<0.0001
sex	66/249	0.417290 (0.211635-0.796090)	0.0094
brainPAD	66/249	1.062133 (1.025712-1.102803)	0.0011
MMSE	66/249	0.997371 (0.951036-1.047425)	0.9142
normVol	66/249	0.841160 (0.002321-434.670676)	0.9551

Table 5 (2a) Logistic Regression Results, version = sensitivity1 without quadratic age Vars are standardized

rn	n/N	OR(CI)	p-value
age	66/249	3.160206 (2.044786-5.127378)	<0.0001
sex	66/249	0.417290 (0.211635-0.796090)	0.0094
brainPAD	66/249	1.879258 (1.304337-2.784689)	0.0011
MMSE	66/249	0.983423 (0.726992-1.342145)	0.9142
normVol	66/249	0.990061 (0.704485-1.420190)	0.9551

Table 6 (2a) Logistic Regression Results, version = sensitivity1 without quadratic age Vars are not-centred

rn	n/N	OR(CI)	p-value
age	66/249	1.112583 (1.068569-1.163642)	<0.0001
sex	66/249	0.417290 (0.211635-0.796090)	0.0094
brainPAD	66/249	1.062133 (1.025712-1.102803)	0.0011
MMSE	66/249	0.997371 (0.951036-1.047425)	0.9142
normVol	66/249	0.841160 (0.002321-434.670676)	0.9551

Cox Proportional Hazards Regression

```
## Cox Proportional Hazards regression function
fcoxregfun <- function(data,Center,Scale,Poly) {

  temp_data <- data%>%
    select(group, age, sex, MMSE_numerator, brainPAD, normVol,End_scan2Endyears)%>%
    mutate( `time`=End_scan2Endyears,
            End_scan2Endyears=NULL,
            `MMSE`=MMSE_numerator,
            MMSE_numerator=NULL)%>%

    {if(Poly) mutate(.,`age`=poly(age,2)[,1],`age2`=poly(age,2)[,2]) else .}%>%
    {if(Poly) relocate(.,age2, .after = age) else .}%>%
    mutate_if(is.numeric,~scale(.,center=Center,scale=Scale),na.rm = TRUE)

  #survival object
  temp_S <- Surv(time = temp_data$time, event = as.numeric(temp_data$group)) #DEM=2, nonDEM=1

  #cox proportional hazards regression using survival object + covariates

  temp_model_cox <- temp_data %>%
    {if(Poly) coxph(temp_S  ~ brainPAD + age + age2+ sex + MMSE + normVol, data = .) else cox
ph(temp_S  ~ brainPAD + age + sex + MMSE + normVol, data = .)}

  lt=list(temp_model_cox,temp_S)
  return(lt)
}

## Run Cox Proportional Hazards regression
# run #Center, Scale and Poly settings set at start of code
model_cox <- fcoxregfun (datasub2,Center=fCenter,Scale=fScale,Poly=fPoly)%>%
  .[[1]] #extract model
 summary(model_cox)
```

```
## Call:
## coxph(formula = temp_S ~ brainPAD + age + age2 + sex + MMSE +
##     normVol, data = .)
##
##   n= 249, number of events= 66
##
##                coef  exp(coef)   se(coef)      z Pr(>|z|)
## brainPAD  5.441e-02  1.056e+00  1.583e-02  3.437 0.000589 ***
## age       1.536e+01  4.707e+06  3.904e+00  3.936 8.29e-05 ***
## age2     -3.737e+00  2.382e-02  2.885e+00 -1.296 0.195104
## sexM     -5.890e-01  5.549e-01  2.758e-01 -2.135 0.032723 *
## MMSE      2.260e-03  1.002e+00  1.826e-02  0.124 0.901505
## normVol  -1.706e-01  8.431e-01  2.559e+00 -0.067 0.946842
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## brainPAD 1.056e+00  9.470e-01 1.024e+00 1.089e+00
## age      4.707e+06  2.125e-07 2.238e+03 9.900e+09
## age2     2.382e-02  4.199e+01 8.347e-05 6.796e+00
## sexM     5.549e-01  1.802e+00 3.231e-01 9.527e-01
## MMSE     1.002e+00  9.977e-01 9.670e-01 1.039e+00
## normVol  8.431e-01  1.186e+00 5.590e-03 1.272e+02
##
## Concordance= 0.696  (se = 0.033 )
## Likelihood ratio test= 35.03  on 6 df,   p=4e-06
## Wald test            = 25.78  on 6 df,   p=2e-04
## Score (logrank) test = 28.81  on 6 df,   p=7e-05
```

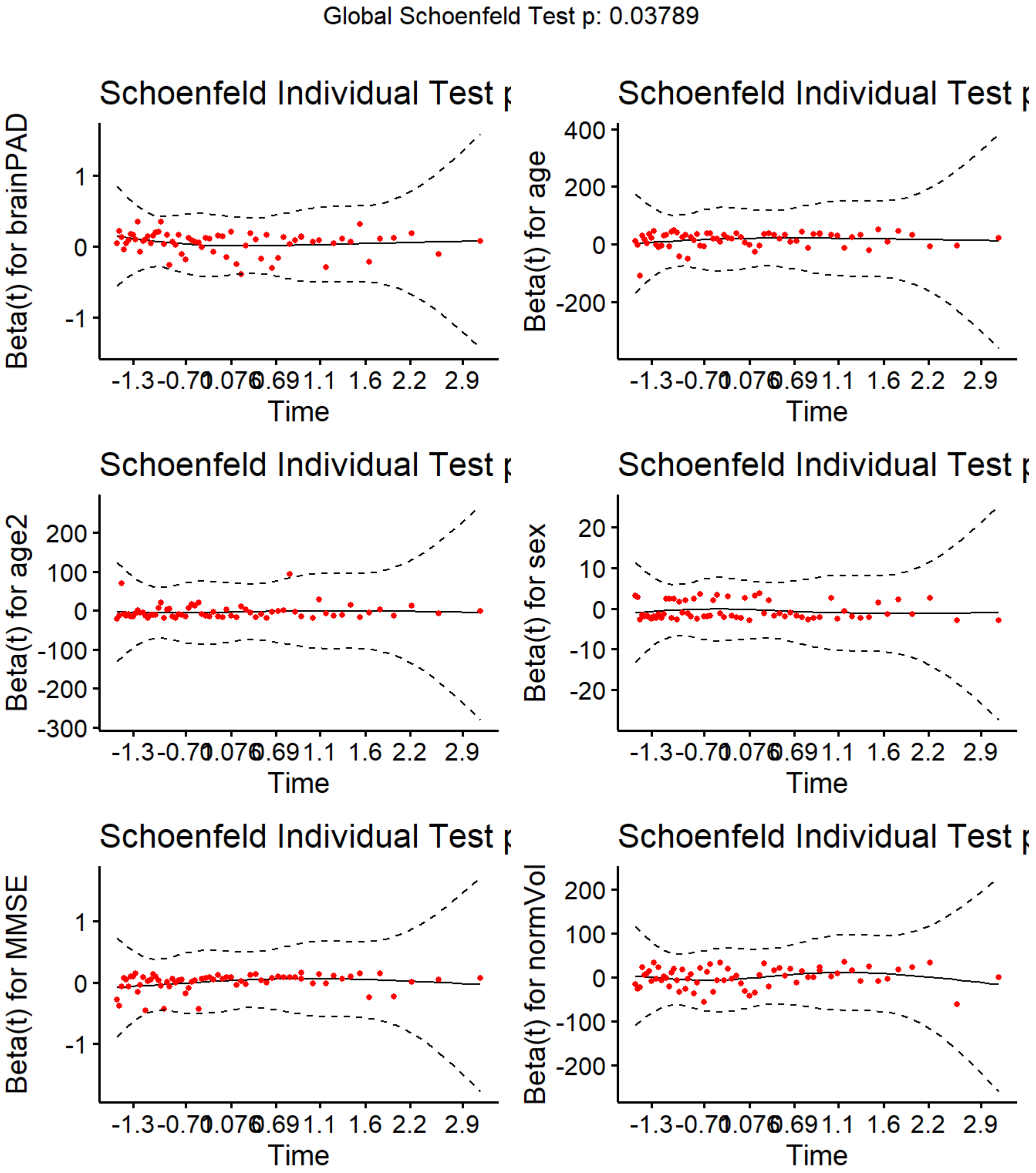
```
## The hazard ratio for brain-PAD on time-to-disease-progression was HR (95% CI) =  1.06 ( 1.
02 - 1.09 ) . That means for every additional +1 year of brain-PAD there is a 1.06  increase
in the likelihood of disease diagnosis. Extrapolated over 5 years of brain-PAD, there is a 1.
31 increase in the likelihood of Dementia diagnosis.
```

Test assumptions of Cox Regression

```
# Check the assumptions of proportional hazards are met
cox.zph(model_cox)
```

```
##           chisq df     p
## brainPAD  5.306  1 0.021
## age       4.250  1 0.039
## age2      2.778  1 0.096
## sex       0.647  1 0.421
## MMSE      3.041  1 0.081
## normVol   0.229  1 0.633
## GLOBAL   13.344  6 0.038
```

```
#GLOBAL p>0.05 hence assumptions met

ggcoxzph(cox.zph(model_cox)) #correlations of DVs with time to ensure that assumptions of ph
are met
```



```
plot(cox.zph(model_cox)[1]) #brainPAD only
```

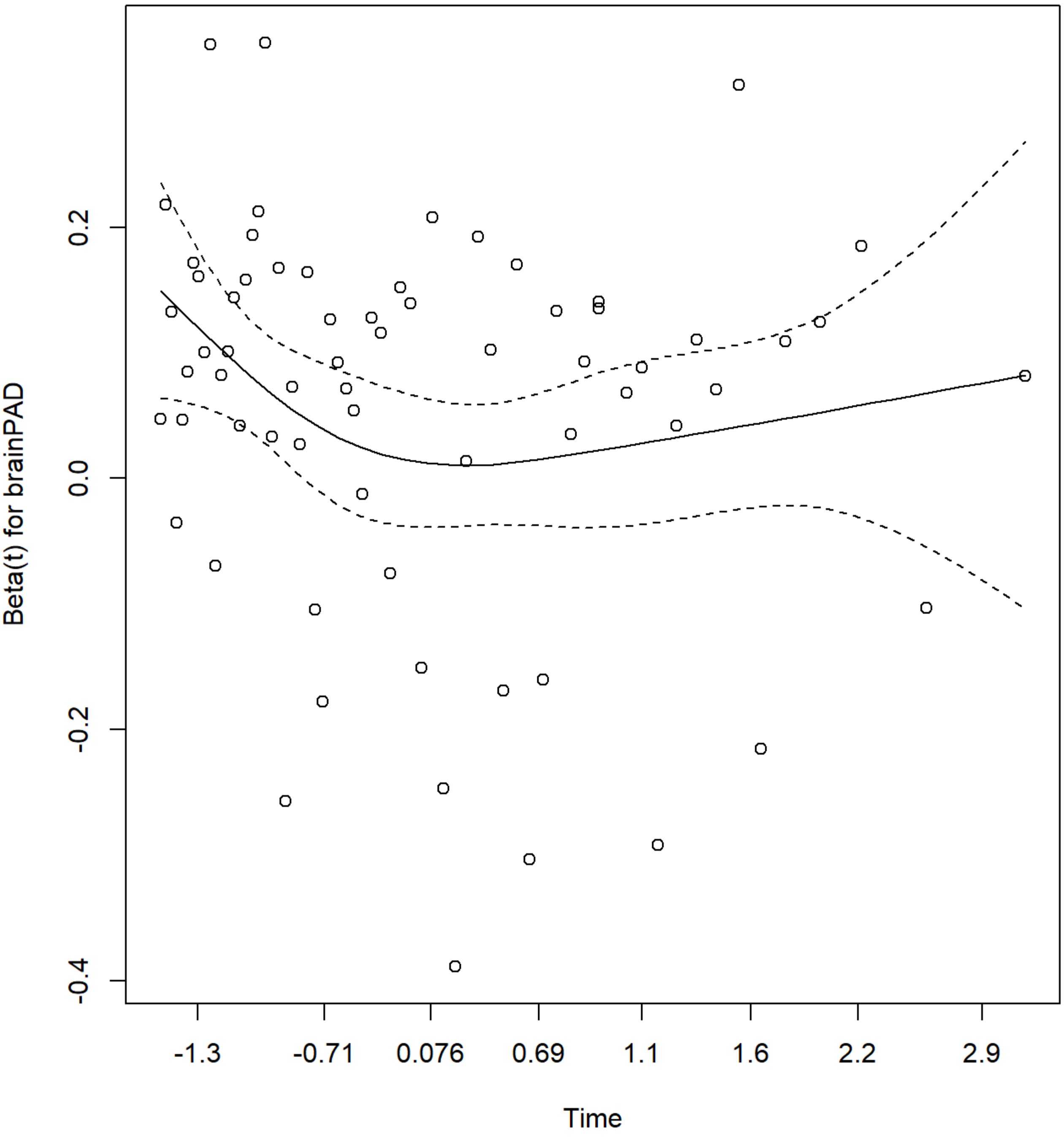


Table 2b Cox regression results

```
# subversion list
#1) PolyT, CentT = TFT: data is centred, age and age2 polynomials include
#2) PolyT, CentZ = TTT: data is standardized, age and age2 polynomials included
#3) PolyT, CentF = FFT: data is raw, age and age2 polynomials included
#4) PolyF, CentT = TFF: data is centred,age2 and poly excluded
#5) PolyF, CentZ = TTF: data is standardized, age2 and poly excluded
#6) PolyF, CentF = FFF: data is raw, age2 and poly excluded

subversion <- c("TFT","TTT", "FFT","TFF","TTF","FFF")


#loop through subversion
for (i in seq_along(subversion)){
  v <- subversion[i]

  table2COX <- flogtablefun(datasub2,type='cox',Center=ftruefun(substring(v,1,1)),Scale=ftruefun(substring(v,2,2)),Poly=ftruefun(substring(v,3,3)))

  table2COX2 <- table2COX[[1]] #extract table

  # print to console
  tt <- kbl(table2COX2,caption=paste("Table",i," ",table2COX[[2]])) %>%
  kable_classic(full_width = T, html_font = "Cambria",position = "left")%>%
  print(tt)
  cat("\n\n")

  nr <- nrow(table2COX2)
  table2COX2[[nr+1,1]] <- paste("Caption: ",table2COX[[2]]) #add caption

  #save table to file for export
  write.xlsx(table2COX2, paste(outputpath,table2COX[[3]],sep=""), row.names=T, col.names=T) # save to xls

}
```

Table 1 (2b) Cox PH Regression Results, version = sensitivity1 with quadratic age (polynomial) Vars are centred

rn	n/N	HR(CI)	p-value
age	66/249	4706979.462720 (2238.048771-9899541040.957680)	<0.0001
age2	66/249	0.023817 (0.000083-6.796039)	0.1951
sex	66/249	0.554857 (0.323136-0.952744)	0.0327
brainPAD	66/249	1.055914 (1.023654-1.089192)	0.0006
MMSE	66/249	1.002263 (0.967019-1.038791)	0.9015
normVol	66/249	0.843127 (0.005590-127.170156)	0.9468

Table 2 (2b) Cox PH Regression Results, version = sensitivity1 with quadratic age (polynomial) Vars are standardized

rn	n/N	HR(CI)	p-value
age	66/249	2.652892 (1.631988-4.312431)	<0.0001

rn	n/N	HR(CI)	p-value
age2	66/249	0.788737 (0.550828-1.129401)	0.1951
sex	66/249	0.554857 (0.323136-0.952744)	0.0327
brainPAD	66/249	1.767239 (1.277206-2.445286)	0.0006
MMSE	66/249	1.014459 (0.808165-1.273412)	0.9015
normVol	66/249	0.990195 (0.741171-1.322886)	0.9468

Table 3 (2b) Cox PH Regression Results, version = sensitivity1 with quadratic age (polynomial) Vars are not-centred

rn	n/N	HR(CI)	p-value
age	66/249	4706979.462720 (2238.048771-9899541040.957748)	<0.0001
age2	66/249	0.023817 (0.000083-6.796039)	0.1951
sex	66/249	0.554857 (0.323136-0.952744)	0.0327
brainPAD	66/249	1.055914 (1.023654-1.089192)	0.0006
MMSE	66/249	1.002263 (0.967019-1.038791)	0.9015
normVol	66/249	0.843127 (0.005590-127.170156)	0.9468

Table 4 (2b) Cox PH Regression Results, version = sensitivity1 without quadratic age Vars are centred

rn	n/N	HR(CI)	p-value
age	66/249	1.077591 (1.044483-1.111748)	<0.0001
sex	66/249	0.556918 (0.326173-0.950899)	0.0320
brainPAD	66/249	1.052061 (1.021597-1.083433)	0.0007
MMSE	66/249	1.003209 (0.967917-1.039788)	0.8608
normVol	66/249	1.360047 (0.009716-190.370519)	0.9029

Table 5 (2b) Cox PH Regression Results, version = sensitivity1 without quadratic age Vars are standardized

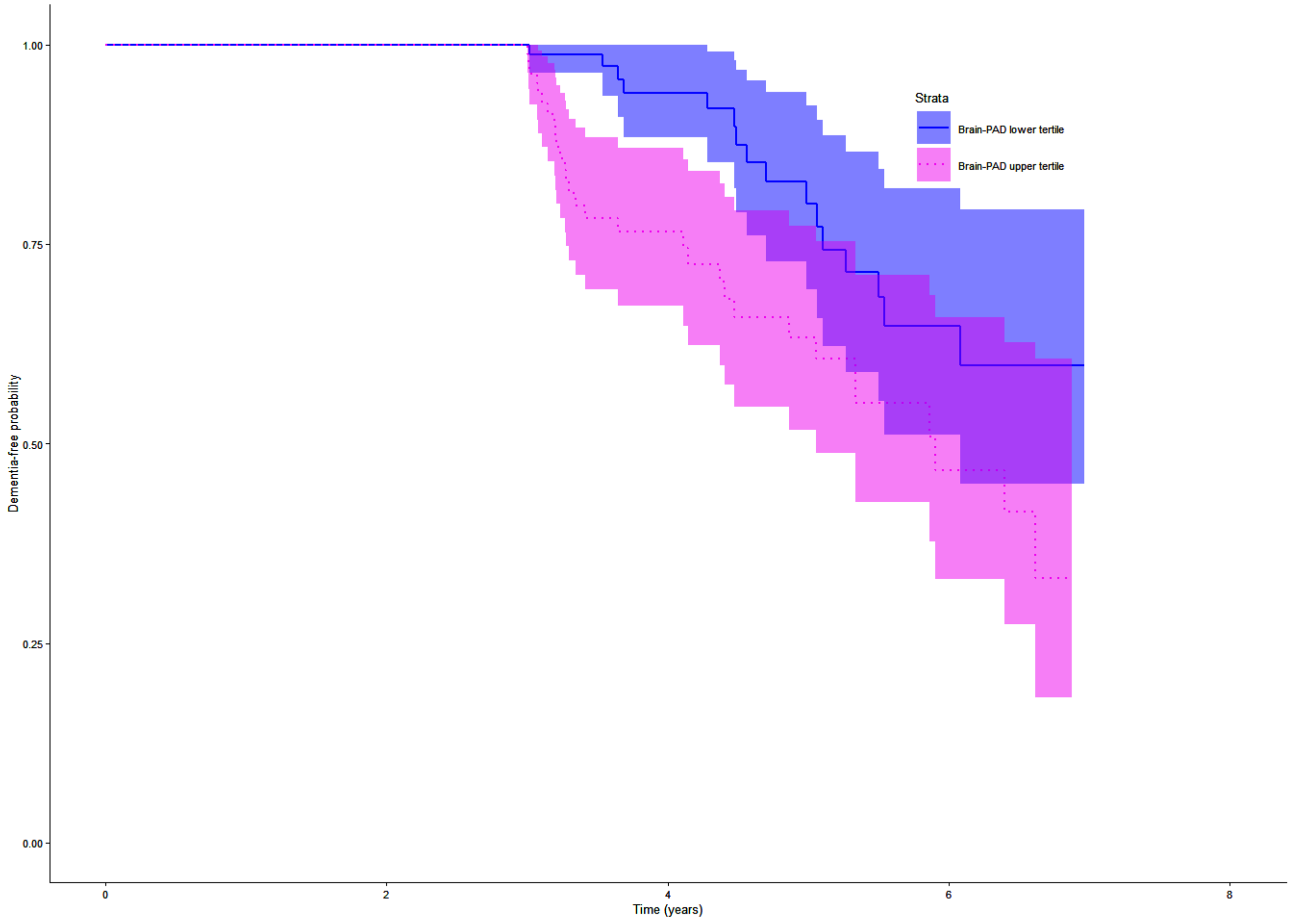
rn	n/N	HR(CI)	p-value
age	66/249	2.238876 (1.599038-3.134740)	<0.0001
sex	66/249	0.556918 (0.326173-0.950899)	0.0320
brainPAD	66/249	1.700894 (1.250604-2.313314)	0.0007
MMSE	66/249	1.020555 (0.812940-1.281192)	0.8608
normVol	66/249	1.017917 (0.765216-1.354068)	0.9029

Table 6 (2b) Cox PH Regression Results, version = sensitivity1 without quadratic age Vars are not-centred

rn	n/N	HR(CI)	p-value
age	66/249	1.077591 (1.044483-1.111748)	<0.0001
sex	66/249	0.556918 (0.326173-0.950899)	0.0320
brainPAD	66/249	1.052061 (1.021597-1.083433)	0.0007
MMSE	66/249	1.003209 (0.967917-1.039788)	0.8608
normVol	66/249	1.360047 (0.009716-190.370519)	0.9029

Figure 4: Kaplan-Meier plot

```
## Prep data for Kaplan-Meir plot
dataKM <- datasub2 %>%
  dplyr::mutate(split_brainPAD = ntile(brainPAD, 3))%>% #create tertile groups
  .[.$split_brainPAD!=2,] #remove middle tertile

#extract response function from coxreg and get surviva fit object
KM_S <- fcoxregfun (dataKM,Center=F,Scale=F,Poly=T)[[2]] #for illustration no need to center
sv.object <- survfit(KM_S  ~ split_brainPAD, data = dataKM)

survplot <- ggsurvplot(sv.object,
                       ggtheme = theme_cowplot(font_size = 5, line_size = 0.2, rel_large =
1),
                       risk.table = F, risk.table.fontsize = 1.2,cumcensor = F, conf.int = T,
                       palette = c("blue1", "magenta2"),linetype = c(1,3),size = 0.5,
                       censor = F,xlab = "Time (years)",ylab = "Dementia-free probability",
                       legend = c(0.7,0.85),legend.labs = c("Brain-PAD lower tertile", "Brain
-PAD upper tertile"))

survplot$plot <- survplot$plot + theme(legend.key.size = unit(1, "line")) + xlim(0,7)
survplot

filename <- paste(outputpath,"Fig4_KM_brain-PAD_",versionname,".tif",sep="")
ggsave(filename = filename, height = 50, width = 80, print(survplot, newpage = FALSE), device
= "tiff", dpi = picres, units = "mm")
```



Assessing multicollinearity

```
#correlation matrix
temp_data <- datasub2%>%
    select(age, MMSE_numerator, brainPAD, normVol,End_scan2Endyears)
corrmatrix <- rcorr(as.matrix(temp_data))
corrmatrix
```

```
##                     age MMSE_numerator brainPAD normVol End_scan2Endyears
## age                1.00          -0.01    -0.38   -0.38              0.02
## MMSE_numerator    -0.01           1.00    -0.07    0.11              0.02
## brainPAD          -0.38          -0.07     1.00    0.00             -0.06
## normVol           -0.38           0.11     0.00    1.00             -0.02
## End_scan2Endyears  0.02           0.02    -0.06   -0.02              1.00
## 
## n= 249 
## 
## 
## P
##                   age    MMSE_numerator brainPAD normVol End_scan2Endyears
## age                      0.8995         0.0000   0.0000  0.7276           
## MMSE_numerator    0.8995                0.2831   0.0706  0.7938           
## brainPAD          0.0000 0.2831                  0.9674  0.3613           
## normVol           0.0000 0.0706         0.9674           0.7161           
## End_scan2Endyears 0.7276 0.7938         0.3613   0.7161
```

```
df.corrmatrix.r=data.frame(corrmatrix$r)
df.corrmatrix.p=data.frame(corrmatrix$P)
filename1 <- paste(outputpath,"correlationmatrix_r_",versionname,".xlsx",sep="")
filename2 <- paste(outputpath,"correlationmatrix_p_",versionname,".xlsx",sep="")
write.xlsx(df.corrmatrix.r,filename1, rowNames=T, colNames=T)
write.xlsx(df.corrmatrix.p,filename2, rowNames=T, colNames=T)



#VIF  - variance inflation factor
car::vif(model_log) #extract VIFs based on logistic regression model
```

```
##      age     age2      sex brainPAD  normVol     MMSE 
## 2.048780 1.711069 1.043289 1.246013 1.149888 1.027895
```