diagSexModels\_noimpute.R

butellyn

2020-09-22

### This script runs GAMMs to see if there are main effects and interactions  
### for sex and longitudinal diagnostic labels for the social cognition tests.  
### (Sanity check for Kosha)  
###  
### Ellyn Butler  
### September 22, 2020  
  
# September 8, 2020: Bart says look into documentation on 'by' in mgcv to  
# understand why factor need to be ordered  
#https://stat.ethz.ch/R-manual/R-patched/library/mgcv/html/gam.models.html  
  
set.seed(20)  
  
library('dplyr')

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library('reshape2')  
library('ggpubr')

## Loading required package: ggplot2

library('psych')

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library('lme4')

## Loading required package: Matrix

## Registered S3 methods overwritten by 'lme4':  
## method from  
## cooks.distance.influence.merMod car   
## influence.merMod car   
## dfbeta.influence.merMod car   
## dfbetas.influence.merMod car

library('gamm4')

## Loading required package: mgcv

## Loading required package: nlme

##   
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':  
##   
## lmList

## The following object is masked from 'package:dplyr':  
##   
## collapse

## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.

## This is gamm4 0.2-6

library('sjPlot') # Version 2.8.4

## Learn more about sjPlot with 'browseVignettes("sjPlot")'.

############################### Social Cog Tests ###############################  
  
clin\_df <- read.csv('~/Documents/pncLongitudinalPsychosis/data/clinical/pnc\_longitudinal\_diagnosis\_n752\_202007.csv')  
names(clin\_df)[names(clin\_df) == 't1'] <- 'first\_diagnosis'  
names(clin\_df)[names(clin\_df) == 'tfinal2'] <- 'last\_diagnosis'  
clin\_df$first\_diagnosis <- as.character(clin\_df$first\_diagnosis)  
clin\_df$last\_diagnosis <- as.character(clin\_df$last\_diagnosis)  
clin\_df$first\_diagnosis <- recode(clin\_df$first\_diagnosis, 'TD'='TD', 'other'='OP', 'PS'='PS')  
clin\_df$last\_diagnosis <- recode(clin\_df$last\_diagnosis, 'TD'='TD', 'other'='OP', 'PS'='PS')  
clin\_df$t1\_tfinal <- recode(clin\_df$t1\_tfinal, 'TD\_TD'='TD\_TD', 'TD\_other'='TD\_OP',  
 'TD\_PS'='TD\_PS', 'other\_TD'='OP\_TD', 'other\_other'='OP\_OP', 'other\_PS'='OP\_PS',  
 'PS\_TD'='PS\_TD', 'PS\_other'='PS\_OP', 'PS\_PS'='PS\_PS')  
  
demo\_df <- read.csv('~/Documents/pncLongitudinalPsychosis/data/demographics/baseline/n1601\_demographics\_go1\_20161212.csv')  
  
cnb\_df <- read.csv('~/Documents/pncLongitudinalPsychosis/data/cognitive/CNB\_Longitudinal\_Core\_11February2020.csv')  
cnb\_df <- cnb\_df[, c('bblid', 'Age', 'timepoint', 'Test', 'ACC\_raw', 'RT\_raw')]  
cnb\_df <- cnb\_df[cnb\_df$bblid %in% clin\_df$bblid,]  
  
# Reverse code the RT data (want faster to be higher)  
cnb\_df$RT\_raw <- -cnb\_df$RT\_raw  
  
cnb\_df1 <- dcast(cnb\_df, bblid + Age + timepoint ~ Test, value.var='ACC\_raw')  
names(cnb\_df1) <- c('bblid', 'Age', 'Timepoint',  
 paste0(names(cnb\_df1)[4:length(names(cnb\_df1))], '\_ACC'))  
cnb\_df2 <- dcast(cnb\_df, bblid + Age + timepoint ~ Test, value.var='RT\_raw')  
names(cnb\_df2) <- c('bblid', 'Age', 'Timepoint',  
 paste0(names(cnb\_df2)[4:length(names(cnb\_df2))], '\_RT'))  
cnb\_df <- merge(cnb\_df1, cnb\_df2)  
  
cnb\_df <- merge(cnb\_df, demo\_df)  
cnb\_df$sex <- recode(cnb\_df$sex, `2`='Female', `1`='Male')  
  
# Remove TAP\_RT, and rename TAP\_ACC to TAP\_RT  
cnb\_df <- cnb\_df[,!(names(cnb\_df) %in% c('TAP\_RT', 'MPRAXIS\_ACC'))]  
names(cnb\_df)[names(cnb\_df) == 'TAP\_ACC'] <- 'TAP\_RT'  
  
tests\_acc <- c('ADT', 'CPF', 'CPT', 'CPW', 'ER40', 'MEDF', 'NBACK', 'PCET',  
 'PLOT', 'PMAT', 'PVRT', 'VOLT')  
tests\_rt <- c('ADT', 'CPF', 'CPT', 'CPW', 'ER40', 'MEDF', 'NBACK', 'PCET',  
 'PLOT', 'PMAT', 'PVRT', 'VOLT', 'TAP', 'MPRAXIS')  
  
cnb\_df[, c(paste0(tests\_acc, '\_ACC'), paste0(tests\_rt, '\_RT'))] <- sapply(cnb\_df[,  
 c(paste0(tests\_acc, '\_ACC'), paste0(tests\_rt, '\_RT'))], scale)  
  
getDiagnoses <- function(i) {  
 bblid <- cnb\_df[i, 'bblid']  
 c(clin\_df[clin\_df$bblid == bblid, 'first\_diagnosis'], clin\_df[clin\_df$bblid == bblid, 'last\_diagnosis'], as.character(clin\_df[clin\_df$bblid == bblid, 't1\_tfinal']))  
}  
  
cnb\_df[,c('first\_diagnosis', 'last\_diagnosis', 't1\_tfinal')] <- t(sapply(1:nrow(cnb\_df), getDiagnoses))  
  
cnb\_df$first\_diagnosis <- ordered(cnb\_df$first\_diagnosis, c('TD', 'OP', 'PS'))  
cnb\_df$last\_diagnosis <- ordered(cnb\_df$last\_diagnosis, c('TD', 'OP', 'PS'))  
cnb\_df$t1\_tfinal <- relevel(factor(cnb\_df$t1\_tfinal), ref='TD\_TD')  
  
  
  
  
#################################### GAMMs ####################################  
  
cnb\_df$sex <- as.factor(as.character(cnb\_df$sex))  
cnb\_df$sex <- relevel(cnb\_df$sex, 'Male')  
cnb\_df$t1\_tfinal <- relevel(cnb\_df$t1\_tfinal, 'TD\_TD')  
  
cnb\_df$oSex <- ordered(cnb\_df$sex, c('Male', 'Female'))  
  
cnb\_df$oT1\_Tfinal <- ordered(cnb\_df$t1\_tfinal, c('TD\_TD', 'OP\_OP', 'OP\_PS',  
 'OP\_TD', 'PS\_OP', 'PS\_PS', 'PS\_TD', 'TD\_OP', 'TD\_PS'))  
  
tests <- c('ADT\_ACC', 'ER40\_ACC', 'MEDF\_ACC')  
  
test <- 'ADT\_ACC'  
#for (test in tests) {  
 test\_df <- cnb\_df[!is.na(cnb\_df[, test]), ]  
 row.names(test\_df) <- 1:nrow(test\_df)  
  
 #### Sex Models  
 mod1\_sex <- gamm4(as.formula(paste0(test, " ~ sex + s(Age, k=10, bs='cr')")),  
 data=test\_df, random=~(1|bblid), REML=TRUE)  
 mod2\_sex <- gamm4(as.formula(paste0(test, " ~ sex + s(Age, k=10, bs='cr') +  
 s(Age, by=oSex, k=10, bs='cr')")), data=cnb\_df, random=~(1|bblid), REML=TRUE)  
  
 print(tab\_model(mod1\_sex$gam, mod2\_sex$gam))  
  
 #### Diagnosis Models  
 mod1\_diag <- gamm4(as.formula(paste0(test, " ~ t1\_tfinal + s(Age, k=10, bs='cr')")),  
 data=cnb\_df, random=~(1|bblid), REML=TRUE)  
 mod2\_diag <- gamm4(as.formula(paste0(test, " ~ t1\_tfinal + s(Age, k=10, bs='cr') +  
 s(Age, by=oT1\_Tfinal, k=10, bs='cr')")), data=cnb\_df, random=~(1|bblid), REML=TRUE)  
  
 print(tab\_model(mod1\_diag$gam, mod2\_diag$gam))  
#}