Appendix: R codes for the analyses

```
#Import data file and check data types

library(dplyr)

write <- read.csv("write.csv")

str(write)

#Change the type of all the covariates from integer to factor

write <- mutate_if(write, is.integer, as.factor)

#----------------------------------------------------------------
#Stage 1. Matching through propensity score methods.
#----------------------------------------------------------------

install.packages("MatchIt")

install.packages("optmatch")

library(MatchIt)

library(optmatch)

# Step 1.1: Selecting covariates

#Build propensity score model using all the covariates

Edu <- as.formula(paste("EducationCategory~",paste(names(write)[c(2:25)],collapse="+")))

# Step 1.2: Estimating propensity score and matching.

#Optimal pair matching (one to one)

pair.m <- matchit(Edu, data = write, method = "optimal", distance = "logit", ratio = 1)

#Optimal full matching

#(one to multiple and multiple to one; in this example we set the ratio to be 1 to 8)

full.m <- matchit(Edu, data = write, method = "full", distance = "logit", min.controls = 1/8,

        max.controls = 8)
```

```
#Step 1.3: Evaluating matching results.

summary(pair.m)

plot(pair.m, type="jitter", interactive = FALSE)

plot(pair.m, type="hist", interactive = FALSE)

summary(full.m)

plot(full.m, type="jitter", interactive = FALSE)

plot(full.m, type="hist")

#save the matched data

full <- match.data(full.m)

#----------------------------------------------------------------------------
#Stage 2. Running linear mixed effects regression DIF analyses.
#----------------------------------------------------------------------------

install.packages("lme4")

library(lme4)

install.packages("sjstats")

library(sjstats) #for R-squared approximation

str(full)

full$subclass <- as.factor(full$subclass)

#Compute the ability proxy variable

full $L_zscore <- scale(full $L_score, center = TRUE, scale = TRUE)

full $R_zscore <- scale(full $R_score, center = TRUE, scale = TRUE)

full $S_zscore <- scale(full $S_score, center = TRUE, scale = TRUE)

full $W2_zscore <- scale(full $W2, center = TRUE, scale = TRUE)

full $ztot<- full $L_zscore + full $R_zscore + full $S_zscore + full $W2_zscore
```

```
#Center the ability proxy variable, ztot, within subclass
cmc.ztot <- full %>%
  select(RegID,ztot,subclass) %>%
  group_by(subclass) %>%
  mutate(c.ztot=ztot-mean(ztot))
full <- merge(full, subset(cmc.ztot, select=c("RegID", "c.ztot")),by="RegID")
#Step 2.1: Establishing a null model (Model 1).
M1.null <- lmer(W1 ~ 1 + (1 |subclass), data= full, REML=FALSE)
summary(M1.null)
r2(M1.null)
#Step 2.2: Running a baseline model with ability approximation variable(s) (Model 2).
M2.base <- lmer(W1 ~ 1 + ztot .c + (1 |subclass), data= full, REML=FALSE)
summary(M2.base)
r2(M2.base)
#Step 2.3: Detecting uniform DIF (Model 3).
M3.uniform <- lmer(W1 ~ 1 + ztot.c + EducationCategory + (1 |subclass), data= full,
        REML=FALSE)
summary(M3.uniform)
r2(M3.uniform)
#Step 2.4: Detecting non-uniform DIF (Model 4).
M4.non.uni <- lmer(W1 ~ 1 + ztot.c * EducationCategory + (1 |subclass), data= full,
        REML=FALSE)
summary(M4.non.uni)
```

```
r2(M4.non.uni)

#Model comparison

anova(M2.base, M3.uniform)

anova(M3.uniform, M4.non.uni)

#Step 2.5: Conducting sensitivity analysis.

install.packages("sensitivityfull")

library(sensitivityfull)

library(tidyr)

#Restructure data from long to wide format

full.format<-full[with(full, order(subclass, -(as.numeric(as.character(EducationCategory))))), ]

full.format<-full.format %>% group_by(subclass) %>% mutate(seq.id = row_number())

#Add a column counting number of treatment unit (coded 1) within each subclass

full.format<-full.format %>% group_by(subclass) %>% mutate(count.t =

        sum(EducationCategory == 1))

#Add a column with True/False to identify if a subclass only has 1 treatment unit

full.format$treated1 <- ifelse(full.format$count.t == 1, TRUE, FALSE)

#Remove columns that will not be used in sensitivity analysis

full.format1<-select(full.format, subclass, seq.id, treated1, W1)

full.wide<-spread(full.format1, seq.id, W1)

#Remove extra columns

full.matrix<-full.wide[-c(1:2)]

full.matrix<-as.matrix(full.matrix)

#Sensitivity analysis. Increase the value of gamma from 1 to 2 with an increment of 0.1.
```

```
senfm(full.matrix, full.wide$treated1, gamma = 1, inner = 0, trim = 3, lambda = 1/2,
    tau = 0, alternative="less")
```