Appendix

R script to replicate (supplementary) analyses

Data pre-reserved DOI: [omitted for anonymous peer-review]

```
##### Load data
load("dat.Rdata")

#### keep full dataset
datfull <- dat

##### Check missingness pattern
dependent <- "SE_use"
explanatory <- c("SES", "Performance", "Study_profile", "Intellectual_ability",
"Motivation","Exam_track", "Gender")
missing_mar <- dat %>%
  missing_compare(dependent, explanatory)

#### complete cases
dat <- na.omit(dat)

##### Inspect data
str(dat)

##### define formula for PSM
ps.formula0 <- formula("cohort ~ SES + Performance + Study_profile + Intellectual_ability +
Motivation + Exam_track + Gender")

#####  No matching; constructing a pre-match matchit object.
m.out0 <- matchit(ps.formula0, data = dat,
                  method = NULL, distance = "glm")
summary(m.out0)

##### Nearest neighbour
m.outNN <- matchit(ps.formula0, data = dat,
                   method = "nearest", distance = "glm")

summary(m.outNN, un = FALSE)

#####  Full matching
set.seed(100)
library(optmatch)
m.outfull <- matchit(ps.formula0, data = dat,
                     method = "full", distance = "glm")

table1_all <- select(as.data.frame(round(summary(m.outfull)$sum.all,2)), 1:3)
```

```
table1_match <- select(as.data.frame(round(summary(m.outfull)$sum.matched,2)), 1:3)

library(sjPlot)
sjPlot::tab_df(table1_match)

#####  Assess balance
summary(m.outfull, un = FALSE)
plot(summary(m.outfull))
bal.tab(m.outfull, stats = c("mean.diffs", "variance.ratios", "ks.statistics"))
bal.tab(m.outNN, stats = c("mean.diffs", "variance.ratios", "ks.statistics"))

#####  Plot distributional balance
p1 <- bal.plot(m.outfull, var.name="distance", type="density",  position = "bottom",
colors=c("grey26","grey92"), which="both", mirror = logical, sample.names = c("Unmatched",
"Matched"))

#######  Define empty models

#####  class
emptymodel0 <- glmer(DV_SE_Use ~ 1 + (1|class), family=binomial, data= dat,
                nAGQ = 0, weights = m.outfull$weights,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))
summary(emptymodel0)

#####  school
emptymodel1 <- glmer(DV_SE_Use ~ 1 + (1|school), family=binomial, data= dat,
                nAGQ = 0, weights = m.outfull$weights,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))
summary(emptymodel1)

#####  class and school
emptymodel2 <- glmer(DV_SE_Use ~ 1 + (1|school/class), family=binomial, data= dat,
                nAGQ = 0, weights = m.outfull$weights,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))
summary(emptymodel2)
##### Compare empty models
anova(emptymodel0, emptymodel1, emptymodel2)

####### Model 1: cohort-shadow education relationship
fit1 <- glmer(DV_SE_Use ~ cohort + (1|school/class), weights = m.outfull$weights,
            control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)),
            data = dat, family = binomial())
summary(fit1)

####### Model 2: same but with student-level covariates
fit2 <- glmer(DV_SE_Use ~ cohort + SES + Performance + Study_profile +  Intellectual_ability +
Motivation + Exam_track + Gender + (1|school/class), weights = m.outfull$weights,
            control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)),
```

```
            data = dat, family = binomial())
summary(fit2)

####### Model 3: same but with student-level and higher level covariates
# school proportions of students with high SES
High_SES_mean_school <- aggregate(dat$SES=="high", by = list(dat$school), FUN = mean)
colnames(High_SES_mean_school) <- c("school","High_SES_mean_school")

# performance by class
Class_mean_performance <- aggregate(dat$Performance, by = list(dat$class), FUN=mean)
colnames(Class_mean_performance)[1] <- "class"
colnames(Class_mean_performance)[2] <- "Class_mean_performance"
dat <- merge(x = dat, y = Class_mean_performance, by= "class",  all.x=T)

fit3 <- glmer(DV_SE_Use ~ cohort + SES + Performance + Study_profile +  Intellectual_ability +
Motivation +   Exam_track + Gender + Class_mean_performance + High_SES_mean_school +
(1|school/class),  weights = m.outfull$weights,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)), data = dat, family =
binomial())
summary(fit3)

####### examine the number of missing cases
propMissing <- apply(data.frame(is.na(datfull)),2,mean)
round(propMissing,2)

####### create own pedicton matrix
####### it is important to impute seperately for the treatment and control group
long.imputation <- c()
for (group in 0:1) {
  #default value of .1 for minimum corrlation
  predictor.selection <- quickpred(subset(datfull,cohort==group), method='pearson',
exclude=c("Ll_Id"))
  imputation <- mice(subset(datfull,cohort==group), m=20, method="pmm",
visitSequence="monotone", predictorMatrix = predictor.selection)
  long.imputation = rbind(long.imputation,complete(imputation, action="long"))}

#######create a list of all imputed datasets
dat_all=list()
for(i in 1:20){dat_all[[i]] = subset(long.imputation, subset=.imp==i)}
#######formula for PSM
ps.formula0 <- formula("cohort ~ SES + Performance + Study_profile + Intellectual_ability +
Motivation + Exam_track + Gender")

#######Matchthem requires a mids object
datimp <- datlist2mids(dat_all)

######## Weighting the Imputed Datasets
```

```
weighted.datasets1 <- weightthem(ps.formula0, datimp, approach = 'across', method = 'ps', distance =
'glm')

######## Assessing Balance on the Weighted Datasets
bal.tab(weighted.datasets1, abs = TRUE)

######## Analyzing the Weighted Datasets
weighted.models1 <- with(data = weighted.datasets1,
                         expr = glmer(DV_SE_Use ~ cohort + (1|school/class), family=binomial,
                                      control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))

######## Pooling the Causal Effect Estimates (Obtained from the Weighted Datasets)
testEstimates(weighted.models1$analyses)
est1 <- testEstimates(weighted.models1$analyses)

######## Caclulating odds ratio
pool.OR1 <- exp(cbind(est1$estimates[,1], (est1$estimates[,1]-1.96*(est1$estimates[,2])),
                (est1$estimates[,1]+1.96*(est1$estimates[,2]))))
colnames(pool.OR1) <- (c("OR", "95% LO", "95% UP"))
round(pool.OR1,2)
####### create own pedicton matrix
####### it is important to impute seperately for the treatment and control group
long.imputation <- c()
for (group in 0:1) {
  #default value of .1 for minimum corrlation
  predictor.selection <- quickpred(subset(dat,cohort==group), method='pearson', exclude=c("Ll_Id"))
  imputation <- mice(subset(dat,cohort==group), m=20, method="pmm", visitSequence="monotone",
predictorMatrix = predictor.selection)
  long.imputation = rbind(long.imputation,complete(imputation, action="long"))}

#######create a list of all imputed datasets
dat_all=list()
for(i in 1:20){dat_all[[i]] = subset(long.imputation, subset=.imp==i)}

#######formula for PSM
ps.formula0 <- formula("cohort ~ SES + Performance + Study_profile + Intellectual_ability +
Motivation + Exam_track + Gender")

#######Matchthem requires a mids object
datimp <- datlist2mids(dat_all)

######## Weighting the Imputed Datasets
weighted.datasets1 <- weightthem(ps.formula0, datimp, approach = 'across', method = 'ps', distance =
'glm')

######## Assessing Balance on the Weighted Datasets
bal.tab(weighted.datasets1, un = TRUE, disp = c("means","sds"), stats = c("mean.diffs"))
```

```
######## Analyzing the Weighted Datasets
weighted.models1 <- with(data = weighted.datasets1,
                  expr = glmer(DV_SE_Use ~ cohort + (1|school/class), family=binomial,
                          control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))

######## Pooling the Causal Effect Estimates (Obtained from the Weighted Datasets)
testEstimates(weighted.models1$analyses)
est1 <- testEstimates(weighted.models1$analyses)

######## Caclulating odds ratio
pool.OR1 <- exp(cbind(est1$estimates[,1], (est1$estimates[,1]-1.96*(est1$estimates[,2])),
                (est1$estimates[,1]+1.96*(est1$estimates[,2]))))
colnames(pool.OR1) <- (c("OR", "95% LO", "95% UP"))
round(pool.OR1,2)

#####  class
emptymodel0 <- with(data = weighted.datasets1,
                expr = glmer(DV_SE_Use ~ 1 + (1|class), family=binomial, data= dat,
                 nAGQ = 0, weights = m.outfull$weights,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))

#####  school
emptymodel1 <- with(data = weighted.datasets1,
                expr = glmer(DV_SE_Use ~ 1 + (1|school), family=binomial, data= dat,
                          nAGQ = 0, weights = m.outfull$weights,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))

#####  class and school
emptymodel2 <- with(data = weighted.datasets1,
                expr = glmer(DV_SE_Use ~ 1 + (1|school/class), family=binomial, data= dat,
                          nAGQ = 0, weights = m.outfull$weights,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))))

for (i in 1:20) {
print(anova(emptymodel0$analyses[[i]], emptymodel1$analyses[[i]], emptymodel2$analyses[[i]]))
}

####### Model 1: cohort-shadow education relationship
fit1 <- with(data = weighted.datasets1,
          expr = glmer(DV_SE_Use ~ cohort + (1|school/class), weights = m.outfull$weights,
          control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)),
          family = binomial()))

######## Pooling the Causal Effect Estimates (Obtained from the Weighted Datasets)
testEstimates(fit1$analyses)
est2 <- testEstimates(fit1$analyses)

####### Model 2: same but with student-level covariates
```

```
fit2 <- with(data = weighted.datasets1,
           expr = glmer(DV_SE_Use ~ cohort + SES + Performance + Study_profile +
Intellectual_ability + Motivation + Exam_track + Gender + (1|school/class), weights =
m.outfull$weights,
            control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)),
             family = binomial()))

######## Pooling the Causal Effect Estimates (Obtained from the Weighted Datasets)
testEstimates(fit2$analyses)
est3 <- testEstimates(fit2$analyses)

####### Model 3: same but with student-level and higher level covariates
fit3 <- with(data = weighted.datasets1,
           expr = glmer(DV_SE_Use ~ cohort + SES + Performance + Study_profile +
Intellectual_ability +  Motivation +   Exam_track + Gender + Class_mean_performance +
High_SES_mean_school + (1|school/class),
                      weights = m.outfull$weights,
control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)),
                      family = binomial()))

######## Pooling the Causal Effect Estimates (Obtained from the Weighted Datasets)
testEstimates(fit3$analyses)
est4 <- testEstimates(fit3$analyses)
```