Supplementary material

1. R-code for running propensity score analysis

We have used different R-packages

```r
install.packages("foreign")       # Installing the "foreign" package for reading data files in to r
install.packages("MatchIt")       # Installing the "MatchIt" package for Matching
install.packages("sandwich")      # Installing "sandwich" package for estimating Confidence Intervals for estimated treatment effects using Z score
install.packages("nonrandom")     # Installing "nonrandom" package to extract propensity score matched dataset for plots

# To load libraries in to R working space
library (MatchIt)                        # Loading "MatchIt" library
library(sandwich)                       # Loading "sandwich" library
library(nonrandom)                      # Loading "nonrandom" library

sample.dat <- read.delim("filepath", header = TRUE, sep="\t")          # syntax to read the data in to r

# Matching the data using MatchIt package: caliper width of 0.25 on the logit of SD propensity score without replacement
set.seed <- 1234           # we used to replicate the study

dataset = sample.dat, t = treatment status (Oral Nutritional Supplementation: Yes/No), y = outcome measure (Hospital Length of Stay), and X1 – X11 = covariates

# Nearest neighbor matching with caliper and without replacement
set.seed <- 1234           # we used to replicate the study

caliper.width <- 0.25
m.out <- matchit(t ~ x1+x2+x3+x4+x5+x6+x7+x7*x7+x8+x9+x9*x9+x9+x10+x11+x12+x9*x10, data = sample.dat, method = "nearest", caliper = caliper.width, unif.seed = 1945, int.seed = 1906)  # unif.seed = 1945, int.seed = 1906 were used to get the same set of mated pairs as it might differ with different simulation runs.
summary(m.out)              # Summary of the matched dataset

# Nearest neighbor matching with caliper and replacement

m.out.rep <- matchit(t ~ x1+x2+x3+x4+x5+x6+x7+x7*x7+x8+x9*x9+x9+x10+x11+x12+x9*x10, data = sample.dat, method = "nearest", caliper = caliper.width, replace = TRUE)
summary(m.out.rep)
```
# Balance Diagnostics
# Standardized difference before and after matching
summary(m.out, standardize = T)

# Plots
plot(summary(m.out, standardize = T), interactive = F, main = "Standardized Difference Plots before and After Matching")  # Absolute standardized difference plots
plot(m.out, type = "jitter", interactive = F)  # Jitter plots
plot(m.out, type = "hist")  # Propensity score histograms before and after matching for both groups

# Getting the matched dataset from MatchIt
m.data <- match.data(m.out)

# Estimating propensity score in the original unmatched data using logistic regression model
sample.dat$P.score <- glm(t ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x9*x10+x11+x12+x9*x10, family=binomial, data = sample.dat)$fitted
# the fitted values from the model are the propensity score the subjects in the data

# Function to make quintiles of the ps for stratification on the PS
quintile <- function(x){
  q <- quantile(x, c(.2,.4,.6,.8))
quintile <- 1 + 1*(x>q[1]) + 1*(x>q[2]) + 1*(x>q[3]) + 1*(x>q[4])
quintile
}

# Function to make deciles of the ps for stratification on the PS
decile <- function(x){
  q <- quantile(x,c(.1,.2,.3,.4,.5,.6,.7,.8,.9))
decile <- 1 + 1*(x>q[1]) + 1*(x>q[2]) + 1*(x>q[3]) + 1*(x>q[4]) + 1*(x>q[5]) + 1*(x>q[6]) + 1*(x>q[7]) + 1*(x>q[8]) + 1*(x>q[9])
decile
}

# Constructing quintiles and deciles of ps to be used for stratification on the PS
sample.dat$P.score.quin <- quintile(sample.dat$P.score)
sample.dat$P.score.deci <- decile(sample.dat$P.score)

# Estimating treatment effect using different methods
# Estimating treatment effect using conventional linear regression model, crude effect
Out.crude <- lm(y ~ t, data=sample.dat)  # linear regression model
summary(Out.crude)  # summary of the result from the model
# Estimating treatment effect using conventional linear regression model in the original data adjusting for covariates

```r
Out.reg <- lm(y ~ t + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x9 + x9 + x10 + x11 + x12 + x9 + x10, data=sample.dat) # linear regression model
summary(Out.reg) # summary of the result from the model
```

# Estimating 95% confidence intervals using robust standard errors for the parameter estimates to control for mild violation of the distribution assumption: variance = mean.
# We have used R package "sandwich"

```r
cov.Out.reg <- vcovHC(Out.reg, type="HC0")
std.err <- sqrt(diag(cov.Out.reg))
r.est <- cbind(Estimate=coef(Out.reg), "Robust SE" = std.err, "Pr(|z|)" = 2 * pnorm(abs(coef(Out.reg)/std.err), lower.tail=FALSE),
LL = coef(Out.reg) - 1.96 * std.err,
UL = coef(Out.reg) + 1.96 * std.err)
r.est.all <- c(r.est[2,1], r.est[2,4], r.est[2,5]) # the estimate and the 95% confidence intervals
```

# Estimating treatment effect in the matched data

```r
Out.match <- lm(y ~ t, data=m.data) # Note the data is matched one and not the original sample.dat
summary(Out.match)
```

# Extracting the estimate and 95% CI from the model

```r
cov.Out.match <- vcovHC(Out.match, type="HC0")
std.err <- sqrt(diag(cov.Out.match))
r.est.match <- cbind(Estimate=coef(Out.match), "Robust SE" = std.err, "Pr(|z|)" = 2 * pnorm(abs(coef(Out.match)/std.err), lower.tail=FALSE),
LL = coef(Out.match) - 1.96 * std.err,
UL = coef(Out.match) + 1.96 * std.err)
r.est.match.all <- c(r.est.match[2,1], r.est.match[2,4], r.est.match[2,5]) # the estimate and the 95% confidence intervals
```

# Estimating treatment effect using stratification on the PS, model-based adjustment using the quintiles and deciles of the PS as a categorical variable

# Fitting the model using PS Quintiles

```r
Out.p.quin <- lm(y ~ t + factor(sample.dat$P.score.quin), data=sample.dat)
summary(Out.p.quin)
```

# Extracting the estimate and 95% CI from the model

```r
cov.Out.p.quin <- vcovHC(Out.p.quin, type="HC0")
std.err <- sqrt(diag(cov.Out.p.quin))
r.est.quin <- cbind(Estimate=coef(Out.p.quin), "Robust SE" = std.err, "Pr(|z|)" = 2 * pnorm(abs(coef(Out.p.quin)/std.err), lower.tail=FALSE),
LL = coef(Out.p.quin) - 1.96 * std.err,
UL = coef(Out.p.quin) + 1.96 * std.err)
r.est.quin.all <- c(r.est.quin[2,1], r.est.quin[2,4], r.est.quin[2,5]) # the estimate and the 95% confidence intervals
# Fitting the model using PS Deciles

```r
Out.p.deci <- lm(y ~ t + factor(sample.dat$P.score.deci) , data=sample.dat)
summary(Out.p.deci)
```

# Extracting the estimate and 95% CI from the model

```r
cov.Out.p.deci <- vcovHC(Out.p.deci, type="HC0")
std.err <- sqrt(diag(cov.Out.p.deci))
r.est.deci <- cbind(Estimate= coef(Out.p.deci), "Robust SE" = std.err,"Pr(>|z|)" = 2 * pnorm(abs(coef(Out.p.deci)/std.err), lower.tail=FALSE), LL = coef(Out.p.deci) - 1.96 * std.err, UL = coef(Out.p.deci) + 1.96 * std.err)
r.est.deci.all <- c(r.est.deci[2,1], r.est.deci[2,4], r.est.deci[2,5])  # the estimate and the 95% confidence intervals
```

# Estimating treatment effect using covariate adjustment using the PS

```r
Out.p.cov <- lm(y ~ t + P.score, data=sample.dat)
summary(Out.p.cov)
```

# Extracting the estimate and 95% CI from the model

```r
cov.Out.p.cov <- vcovHC(Out.p.cov, type="HC0")
std.err <- sqrt(diag(cov.Out.p.cov))
r.est.cov <- cbind(Estimate= coef(Out.p.cov), "Robust SE" = std.err,"Pr(>|z|)" = 2 * pnorm(abs(coef(Out.p.cov)/std.err), lower.tail=FALSE), LL = coef(Out.p.cov) - 1.96 * std.err, UL = coef(Out.p.cov) + 1.96 * std.err)
r.est.cov.all <- c(r.est.cov[2,1], r.est.cov[2,4], r.est.cov[2,5])  # the estimate and the 95% confidence intervals
```

# Estimating treatment effect using inverse probability of treatment weighting using the PS

# Estimating inverse probability of treatment weights using PS

# Unstabilized weights

```r
sample.dat$iptw[sample.dat$t==1] <- 1/sample.dat$P.score[sample.dat$t==1]
sample.dat$iptw[sample.dat$t==0] <- 1/(1-sample.dat$P.score[sample.dat$t==0])
summary(sample.dat$iptw)
```

# Fitting the model using the weights

```r
Out.p.iptw <- lm(y ~ t, data=sample.dat, weights=iptw)
summary(Out.p.iptw)
```
# Extracting the estimate and 95% CI from the model

cov.Out.p.iptw <- vcovHC(Out.p.iptw, type="HC0")
std.err <- sqrt(diag(cov.Out.p.iptw))
r.est.iptw <- cbind(Estimate=coef(Out.p.iptw), "Robust SE" = std.err, "Pr(>|z|)|" = 2 * pnorm(abs(coef(Out.p.iptw)/std.err), lower.tail=FALSE),
LL = coef(Out.p.iptw) - 1.96 * std.err,
UL = coef(Out.p.iptw) + 1.96 * std.err)
r.est.iptw.all <- c(r.est.iptw[2,1], r.est.iptw[2,4], r.est.iptw[2,5])  # the estimate and the 95% confidence intervals

# Estimating inverse probability of treatment weights using PS with stabilization using the prevalence of treatment

sample.dat$S.iptw[sample.dat$t==1] <- (length(sample.dat$y[sample.dat$t==1])/length(sample.dat$y))/sample.dat$P.score[sample.dat$t==1]
sample.dat$S.iptw[sample.dat$t==0] <- (length(sample.dat$y[sample.dat$t==0])/length(sample.dat$y))/(1-sample.dat$P.score)[sample.dat$t==0]
summary(sample.dat$S.iptw)

# Fitting the model using the weights

Out.p.siptw <- glm(y ~ t, data=sample.dat, weights=S.iptw)
summary(Out.p.siptw)

# Extracting the estimate and 95% CI from the model

cov.Out.p.siptw <- vcovHC(Out.p.siptw, type="HC0")
std.err <- sqrt(diag(cov.Out.p.siptw))
r.est.siptw <- cbind(Estimate=coef(Out.p.siptw), "Robust SE" = std.err, "Pr(>|z|)|" = 2 * pnorm(abs(coef(Out.p.siptw)/std.err), lower.tail=FALSE),
LL = coef(Out.p.siptw) - 1.96 * std.err,
UL = coef(Out.p.siptw) + 1.96 * std.err)
r.est.siptw.all <- c(r.est.siptw[2,1], r.est.siptw[2,4], r.est.siptw[2,5])  # the estimate and the 95% confidence intervals
# Additional r codes used for constructing the figures

# Syntax used to generate propensity score density plots two treatment groups using kernel density function (Figure 2)

```r
Data.ps <- pscore(data = sample.dat, formula = t~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x9*x9+x9*x10, name.pscore = "ps")
# From non random package to use plot.pscore function below
names(Data.ps)
lablist.y<-as.vector(c("0", "20", "40", "60", "80", "100"))

plot.pscore(x = Data.ps,
#main = "PS distribution",
with.legend = FALSE,
par.1 = list(lty=1, lwd=2,cex = 2),
par.0 = list(lty=3, lwd=2,cex = 2),
xlab = "Propensity Score",
ylab = "Density",labels= FALSE,
ylim = c(0,100),
xlim = c(0,1), cex.lab=1.5, cex.axis=1.5,bty="n",
par.dens=list(kernel="gaussian"))

lablist.y  <- as.vector(c("0", "20", "40", "60", "80", "100"))
lablist.x  <- as.vector(c("0", "0.2", "0.4", "0.6", "0.8", "1"))
axis(1, at=seq(1, 10, by=1), labels = FALSE)
text(x = seq(0, 1, by=0.2), par("usr")[3], labels = lablist.x, srt = 0, pos = 1, xpd = TRUE)
axis(2, at=seq(0, 100, by=20), labels = FALSE)
text(y = seq(0,100, by=20), par("usr")[1], labels = lablist.y, srt = 0, pos = 2, xpd = TRUE)

legend(0.2, 60, c("Treated", "Untreated"), lty=c(1,3), lwd =c(3,3), col= c("black","black"), cex = c(1.5,1.5))
```

# Code used to generate histograms of propensity score before and after matching in the two treatment groups (Figure 3)

```r
Before.match.treat <- sample.dat$ps[sample.dat$t==1]  # a vector of PS for treated groups before matching
Before.match.untreat <- sample.dat$ps[sample.dat$t==0]  # a vector of PS for untreated groups before matching
After.match.treat <- m.data$distance[m.data$t == 1]  # a vector of PS for treated groups after matching
After.match.untreat <- m.data$distance[m.data$t == 0]  # a vector of PS for treated groups after matching

layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE))  # we created a matrix of 2 rows and two columns for our four plots

hist(Before.match.treat, main = "(A) Before Matching Treated", xlab = "Propensity Score", ylab="Density",cex.lab=1.2)
hist(After.match.treat, main = "(B) After Matching Treated", xlab = "Propensity Score", ylab="Density",cex.lab=1.2)
hist(Before.match.untreat, main = "(C) Before Matching Untreated", xlab = "Propensity Score", ylab="Density",cex.lab=1.2)
hist(After.match.untreat, main = "(D) After Matching Untreated", xlab = "Propensity Score", ylab="Density",cex.lab=1.2)
```
# Code used to generate Standardized difference plots (Figure 4)

```r
par(mar=c(5,6,4,2)+0.2)
plot(x, y, xlim = c(0, 120), ylim = c(1,13), type = "n", xlab = "Absolute Standardized Difference,%", ylab = "", labels = FALSE, cex.lab=1.2, cex.axis=1, bty="n")
mtext("Covariates", side=2, line=5, cex=1.2)
points(SD.plot$SD.Before, SD.plot$Cov.Id, cex = 1.5, pch = 18, col = "black")
points(SD.plot$SD.After, SD.plot$Cov.Id, cex = 1.5, pch = 13, col = "black")
abline(v=10,lty=2)

lablist.x <- as.vector(c("0", "20", "40", "60", "80", "100","120"))
axis(1, at=seq(0, 120, by=20), labels = FALSE)
text(x = seq(0, 120, by=20), par("usr")[3], labels = lablist.x, srt = 0, pos = 1, xpd = TRUE)
axis(2, at=seq(0, 13, by=1), labels = FALSE)
text(y = seq(1,13, by=1), par("usr")[1], labels = lablist.y, srt = 0, pos = 2, xpd = TRUE)
legend(60, 4, c("Before Matching", "After Matching"), pch=c(18,13), bty="o", col=c("black","black"))  # legend
```
2. Sensitivity Analysis for Matching with Replacement and Smaller Caliper

When matching was conducted with replacement, among 44,000 patient population (ONS use = 675 and non-ONS use = 43,325); 673 of the treated patients were matched with 649 untreated patients with only 2 extra treated patient found an untreated match compared to matching without replacement. This is mainly due to the large size of the untreated population. However, balance of covariates was better improved with the maximum absolute standardized difference of 8.4 compared to 13.29 in matching without replacement. ONS use was associated with 2.42-days 95 % CI [-2.53, -2.30] decrease in LOS similar to 2.42-days 95 % CI [-2.53, -2.30] in matching without replacement. Reducing the caliper with to 0.10 standard deviation on the logit of the PS did not alter the result. ONS use resulted in 2.42-days decrease 95% CI [-2.53, -2.30] in LOS. Among 44,000 patient population (ONS use = 675 and non-ONS use = 43,325), 668 of treated patients were matched with 668 with 7 treated patients left unmatched due to smaller caliper width. The maximum absolute standardized difference was 5.4 as expected.