

Balance Analysis

Susan Glenn, Jeanne Li, Jingcheng Xu, Stewart Kerr

12/2/2019

In terms of balance, the matching improved balance significantly. All of our after matching standardized differences were below 0.1. Note that we only checked balance on baseline covariates and threw out all of the matched pairs where subjects died in the same wave or lived throughout the followup period. Another possible avenue to address balance issues is enforcing the balance constraints in our risk set matching. This would force balance among important covariates. Alternatively, we could also do risk set matching allowing for matching with multiple controls. Out of the 16 baseline covariates that balance was checked on seven had a higher standardized differences before matching. The code to assess balance is below.

Balance

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.5.3
##
## 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(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.3

balance <- read.csv("../data/initial-balance.csv")
before <- balance[,c(2:17)]
before$FIRST_WS[before$FIRST_WS == -1] <- 0
before$FIRST_WS[before$FIRST_WS > 0] <- 1

propscore.model=glm(FIRST_WS ~ ., family=binomial, x=TRUE, y=TRUE, data=before);

Xmat=propscore.model$x;
treated = propscore.model$y

treatedmat=Xmat[treated==1,]
controlmat.before=Xmat[treated==0,]

#standardized differences before matching
controlmean.before=apply(controlmat.before,2,mean,na.rm=TRUE);
treatmean=apply(treatedmat,2,mean,na.rm=TRUE);
treatvar=apply(treatedmat,2,var,na.rm=TRUE);
controlvar=apply(controlmat.before,2,var,na.rm=TRUE);
stand.diff.before=(treatmean-controlmean.before)/sqrt((treatvar+controlvar)/2)

#Load matches
```

```

data <- read.csv("../data/results-final.csv")
df1 = subset(data, select=c("RABYEAR", "RAEDYRS", "HATOTW", "RSMOKEV", "RSMOKEN", "RDRINKD", "RLTACTF",

# Standardized differences after matching
controlmat.after=df1
controlmean.after=apply(controlmat.after,2,mean);
# Standardized differences after matching
stand.diff.after=(treatmean[-1]-controlmean.after)/sqrt((treatvar[-1]+controlvar[-1])/2);

standBeforeAfter = cbind(stand.diff.before[-1],stand.diff.after[-1])

## Warning in cbind(stand.diff.before[-1], stand.diff.after[-1]): number of
## rows of result is not a multiple of vector length (arg 2)

colnames(standBeforeAfter ) = c("Before Match (Standardized Diff)",
                                "After Match (Standardized Diff)")

knitr::kable(round(abs(standBeforeAfter),3), caption = "Differences in Covariates (Before and After)")

```

Table 1: Differences in Covariates (Before and After)

	Before Match (Standardized Diff)	After Match (Standardized Diff)
RABYEAR	0.078	0.062
RAEDYRS	0.150	0.001
HATOTW	0.080	0.014
RSMOKEV	0.034	0.045
RSMOKEN	0.162	0.023
RDRINKD	0.192	0.057
RLTACTF	0.164	0.020
RVGACTF	0.044	0.062
RBMI	0.176	0.052
RRISK	0.048	0.019
RBEQLRG	0.232	0.002
RAGENDER_1.Male	0.170	0.032
RACE_ETHN_Hispanic	0.230	0.067
RACE_ETHN_NonHispanicBlack	0.362	0.000
RACE_ETHN_NonHispanicOther	0.035	0.062

Love Plot

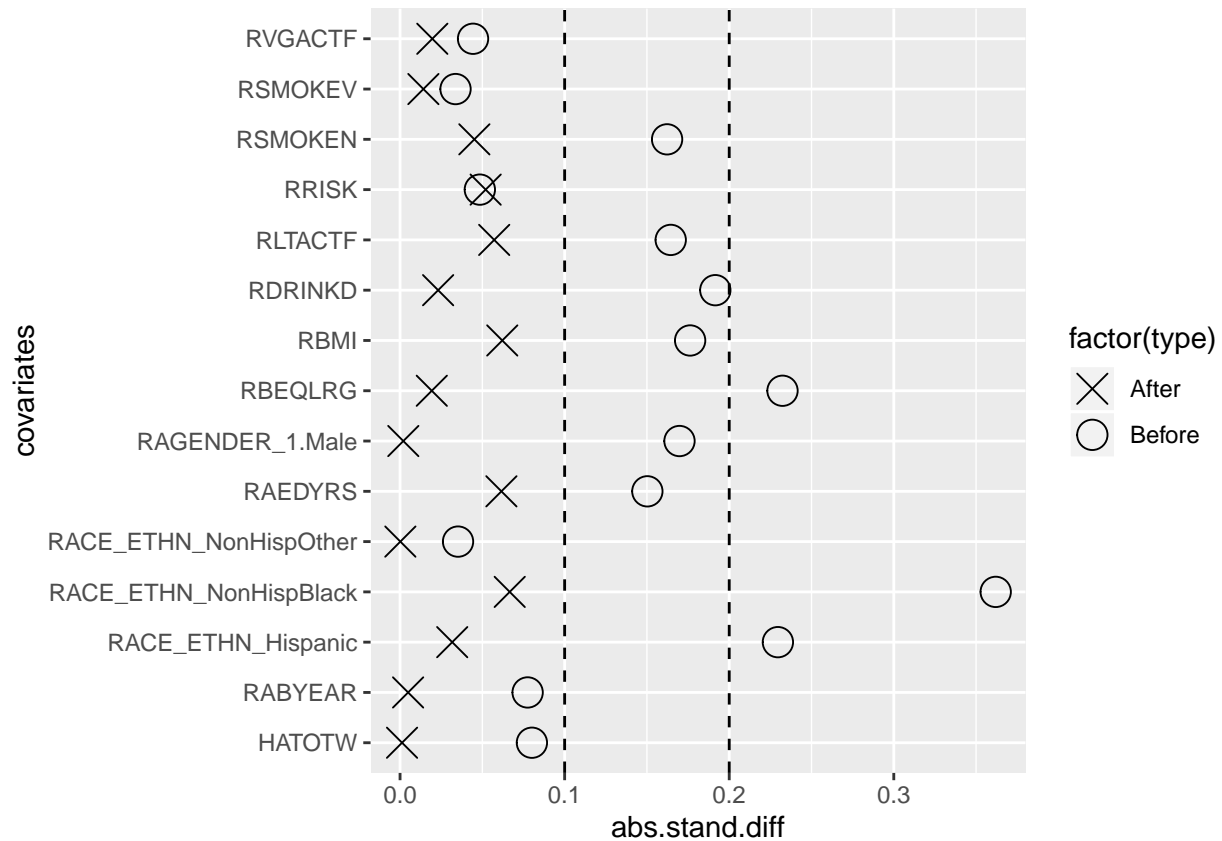
```

library(ggplot2)
abs.stand.diff.before=abs(stand.diff.before[-1])
abs.stand.diff.after=abs(stand.diff.after)
covariates=names(stand.diff.before[-1])

plot.dataframe = data.frame(abs.stand.diff=c(abs.stand.diff.before,abs.stand.diff.after), covariates=rep(covariates,2))

ggplot(plot.dataframe,aes(x=abs.stand.diff,y=covariates))+geom_point(size=5,aes(shape=factor(type)))+scale_shape_manual(values=c(1,2))

```



Plots

```
# library(plotly)
# library(viridis)
# library(reshape2)
# library(ggpubr)
# ggg <- subset(data, outcome >= 0)
# ggg$outcome <- as.factor(ggg$outcome)
# ggg$treated <- as.factor(ggg$treated)
#
# l <- qplot(x=RIEARN, data=ggg, facets = treated ~., fill=..count.., geom="histogram")
# l <- l + scale_fill_gradient(low="blue", high="red")
# l <- l + theme_minimal()
#
# k <- qplot(x=HATOTW, data=ggg, facets = treated ~., fill=..count.., geom="histogram")
# k <- k + scale_fill_gradient(low="blue", high="red")
# k <- k + theme_minimal()
#
# h <- qplot(x=RBMI, data=ggg, facets = treated ~., fill=..count.., geom="histogram")
# h <- h + scale_fill_gradient(low="blue", high="red")
# h <- h + theme_minimal()
#
# j <- qplot(x=RADYEAR, data=ggg, facets = treated ~., fill=..count.., geom="histogram")
# j <- j + scale_fill_gradient(low="blue", high="red")
# j <- j + theme_minimal()
```

```

#
# figure5 <- ggarrange(k, l, h, j,
#                       labels = c("HATOW", "RIEARN", "RBMI", "RADYEAR"),
#                       ncol = 2, nrow = 2)
#
# values <- c(0, 1)
# #p <- ggplot(ggg, aes(RSMOKEN, values))
# #p + geom_bar(stat = "identity", aes(fill = treated), position = "dodge")
#
# p <- ggplot(data, aes(x=treated, y=RAEDYRS, group=treated)) +
#   geom_boxplot(fill=NA, alpha=0.5) +
#   geom_jitter(aes(colour=outcome), width=0.35, alpha=0.5)
# p <- p + scale_color_viridis(option = "C")
#
# data$outcome <- as.factor(data$outcome)
# data$treated <- as.factor(data$treated)
# # k <- ggplot(data = data, aes(y = RADYEAR, x = treated, fill = treated, color=treated), show.legend = FALSE) +
# #   geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8, show.legend = FALSE) +
# #   geom_point(aes(y = RADYEAR, color = outcome), position = position_jitter(width = .15), size = .5, alpha = .5) +
# #   geom_boxplot(width = .1, outlier.shape = NA, alpha = 0.5, show.legend = FALSE) +
# #   expand_limits(x = 5.25) +
# #   scale_colour_manual(values=c("red", "darkviolet", "seagreen")) +
# #   scale_fill_manual(values=c("gray92", "gray91")) +
# #   coord_flip() +
# #   theme_bw() +
# #   raincloud_theme
#
# ggg$outcome <- as.factor(ggg$outcome)
# ggg$treated <- as.factor(ggg$treated)
# k <- ggplot(data = ggg, aes(y = RADYEAR, x = treated, fill = treated, color=treated), show.legend = FALSE) +
#   geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8, show.legend = FALSE) +
#   geom_point(aes(y = RADYEAR, color = outcome), position = position_jitter(width = .15), size = .5, alpha = .5) +
#   geom_boxplot(width = .1, outlier.shape = NA, alpha = 0.5, show.legend = FALSE) +
#   expand_limits(x = 5.25) +
#   scale_colour_manual(values=c("darkviolet", "red")) +
#   scale_fill_manual(values=c("gray92", "gray91")) +
#   coord_flip() +
#   theme_bw() +
#   raincloud_theme

```