

PS Bootstrap Binary Standardization

Hun

2/9/2022

Generating data with true log odds ratio and its standard deviation

```
pre_data <- defData(varname = "L1", formula = "0", variance = 1,
  dist = "normal")
pre_data <- defData(pre_data, varname = "L2", formula = "0", variance = 1,
  dist = "normal")
pre_data <- defData(pre_data, varname = "L3", formula = "0", variance = 1,
  dist = "normal")
pre_data <- defData(pre_data, varname = "A",
  formula = " 0.5*L1 + 0.27*L2 -0.17*L3",
  dist = "binary", link = "logit")
pre_data <- defData(pre_data, varname = "Y",
  formula = "0.5*A + 0.8*L2 + -0.1*L3",
  dist = "binary", link = "logit")

set.seed(7777)
df <- genData(1000, pre_data)

expit <- function(beta) {
  return(exp(beta)/(1 + exp(beta)))
}

ATE <- expit(sum(0.5 + 0.8*df$L2 - 0.1*df$L3)) - expit(sum(0.8*df$L2 - 0.1*df$L3))
# this is not true ATE

# True log odds ratio: 0.5
```

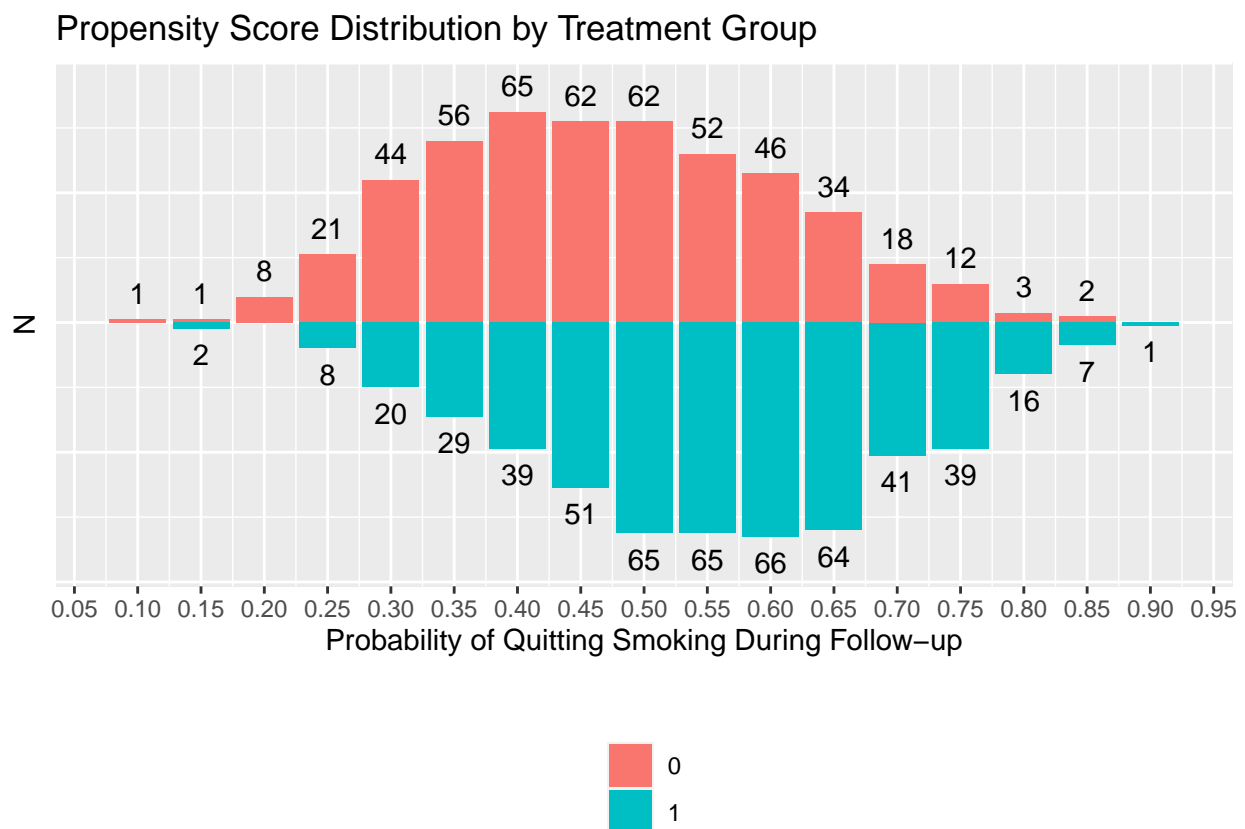
Propensity Score Model

500 pairs Propensity Score distribution

```
df %>%
  mutate(ps.grp = round(ps/0.05) * 0.05) %>%
  group_by(A, ps.grp) %>%
  summarize(n = n()) %>%
  ungroup() %>%
  mutate(n2 = ifelse(A == 0, yes = n, no = -1*n)) %>%
  ggplot(aes(x = ps.grp, y = n2, fill = as.factor(A))) +
```

```
geom_bar(stat = 'identity', position = 'identity') +
geom_text(aes(label = n, x = ps.grp, y = n2 + ifelse(A == 0, 8, -8))) +
xlab('Probability of Quitting Smoking During Follow-up') +
ylab('N') +
ggtitle('Propensity Score Distribution by Treatment Group') +
scale_fill_discrete('') +
scale_x_continuous(breaks = seq(0, 1, 0.05)) +
theme(legend.position = 'bottom', legend.direction = 'vertical',
      axis.ticks.y = element_blank(),
      axis.text.y = element_blank())
```

'summarise()' has grouped output by 'A'. You can override using the '.groups' argument.



Nearest neighbor propensity score matching

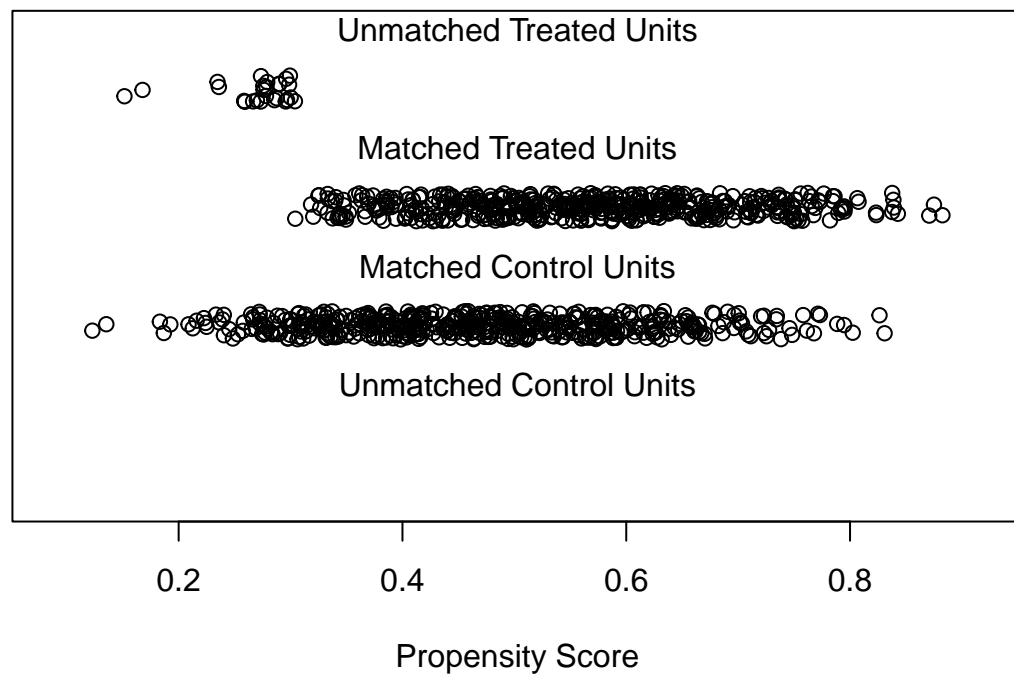
```
matched <- matchit(A ~ L1 + L2 + L3, data = df,
                   distance = "glm", link = "logit",
                   method = "nearest", ratio = 1)
```

```
summary(matched)[2]
```

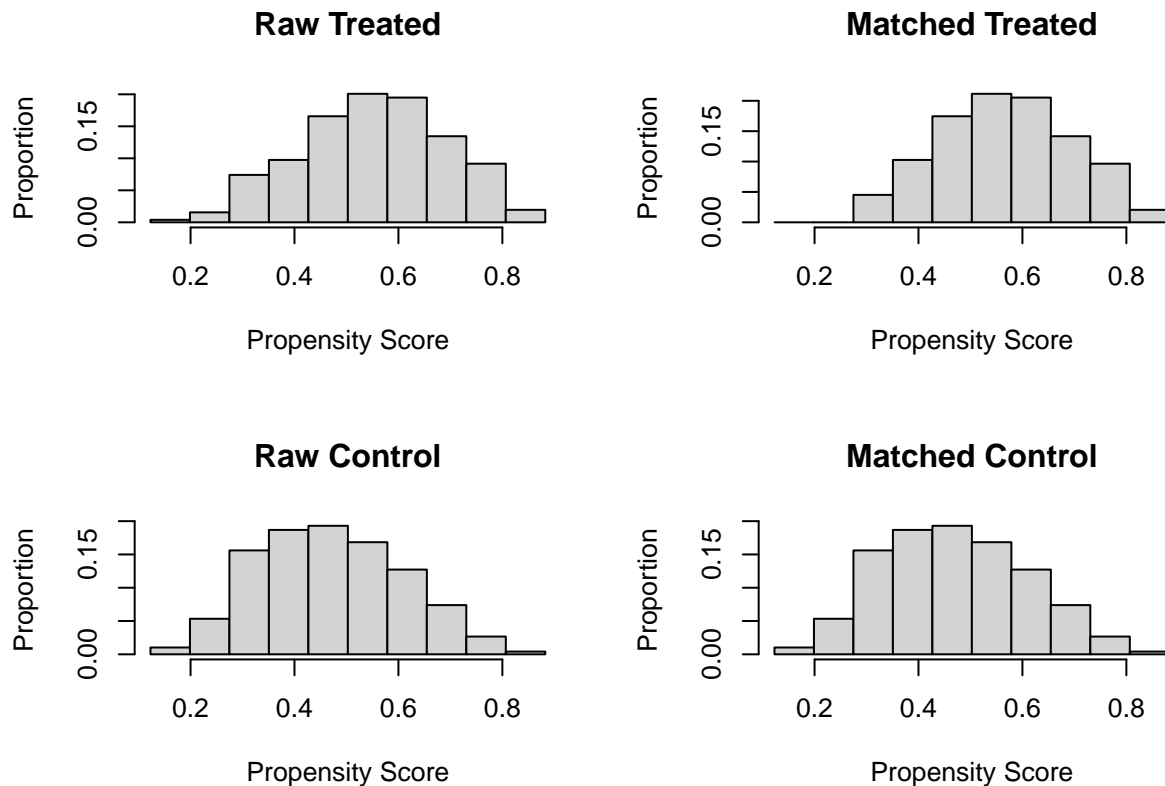
```
## $nn
##           Control Treated
## All (ESS)      487      513
## All            487      513
## Matched (ESS)  487      487
## Matched        487      487
## Unmatched       0       26
## Discarded      0        0
```

```
plot(matched, type = "jitter", interactive = FALSE)
```

Distribution of Propensity Scores



```
plot(matched, type = "histogram")
```



```
matched_df <-  
  match.data(matched)
```

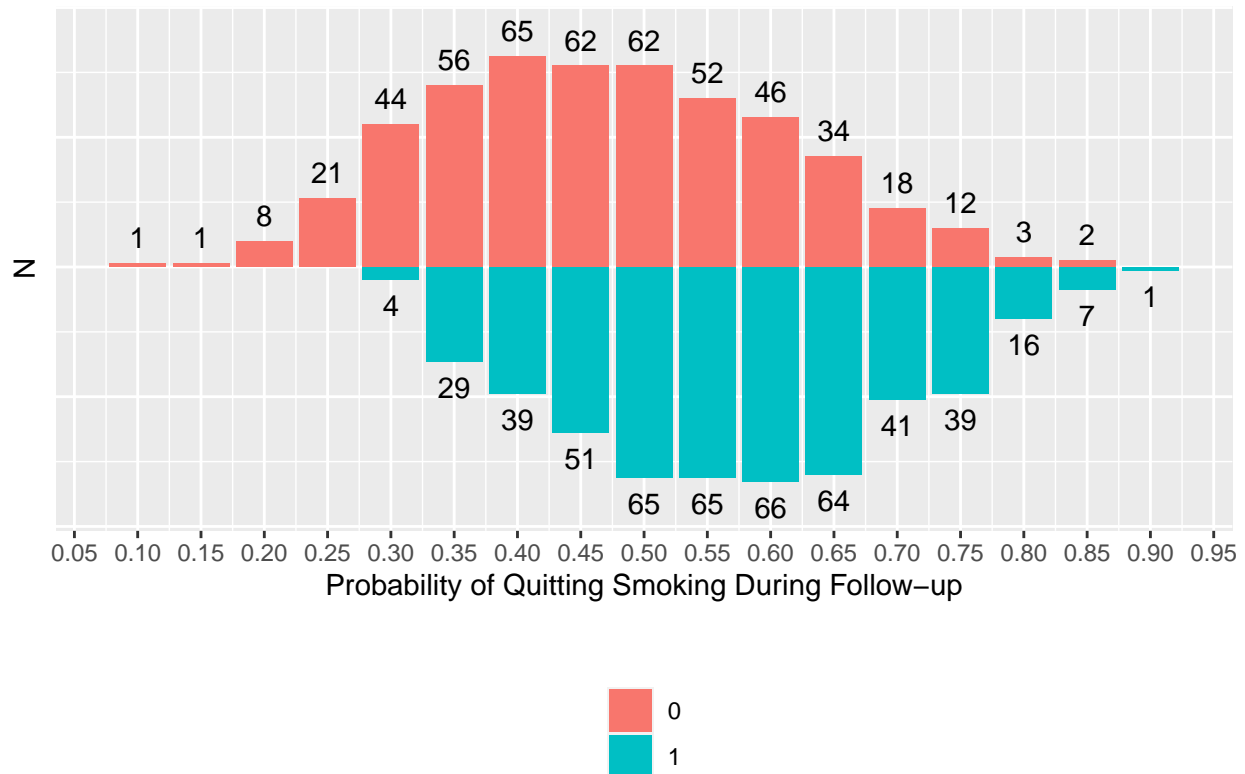
495 pairs propensity score distribution

```
matched_df %>%  
  mutate(ps.grp = round(ps/0.05) * 0.05) %>%  
  group_by(A, ps.grp) %>%  
  summarize(n = n()) %>%  
  ungroup() %>%  
  mutate(n2 = ifelse(A == 0, yes = n, no = -1*n)) %>%  
  ggplot(aes(x = ps.grp, y = n2, fill = as.factor(A))) +  
  geom_bar(stat = 'identity', position = 'identity') +  
  geom_text(aes(label = n, x = ps.grp, y = n2 + ifelse(A == 0, 8, -8))) +  
  xlab('Probability of Quitting Smoking During Follow-up') +  
  ylab('N') +  
  ggtitle('Propensity Score Distribution by Treatment Group') +  
  scale_fill_discrete('') +  
  scale_x_continuous(breaks = seq(0, 1, 0.05)) +  
  theme(legend.position = 'bottom', legend.direction = 'vertical',  
        axis.ticks.y = element_blank(),  
        axis.text.y = element_blank())
```

'summarise()' has grouped output by 'A'. You can override using the '.groups'

```
## argument.
```

Propensity Score Distribution by Treatment Group



simple bootstrap

```
nboot <- 100
# set up a matrix to store results
boots <- data.frame(i = 1:nboot,
                    se_OR = NA,
                    mean_log_OR = NA,
                    mean1 = NA,
                    mean0 = NA,
                    difference = NA
                    )
# loop to perform the bootstrapping

for(i in 1:nboot) {
  # sample with replacement
  sampl <- matched_df %>% filter(subclass %in% sample(levels(subclass),487, replace = TRUE))

  bootmod <- glm(Y ~ A + ps, data = sampl,
                 weights = weights, family = binomial)

  # create new data sets
  sampl.treated <- sampl %>%
```

```

mutate(A = 1)

sampl.untreated <- sampl %>%
  mutate(A = 0)

# predict values
sampl.treated$pred.y <-
  predict(bootmod, sampl.treated, type = "response")

sampl.untreated$pred.y <-
  predict(bootmod, sampl.untreated, type = "response")

# output results

boots[i, "mean_log_OR"] <- summary(bootmod)$coeff[2,1]

boots[i, "se_OR"] <- summary(bootmod)$coeff[2,2]

boots[i, "mean1"] <- mean(sampl.treated$pred.y)
boots[i, "mean0"] <- mean(sampl.untreated$pred.y)
boots[i, "difference"] <- boots[i, "mean1"] - boots[i, "mean0"]

se_ATE <- sd(boots$difference)

ATE <- mean(boots$difference)

# once loop is done, print the results
if (i == nboot){
  cat("ATE:")
  cat(ATE)
  cat("\n")
  cat("\n")
  cat("se_ATE:")
  cat(se_ATE)
  cat("\n")
  cat("\n")
  cat("95% CI for ATE:")
  cat(ATE - 1.96*se_ATE,
      ", ",
      ATE + 1.96*se_ATE)
  cat("\n")
  cat("\n")
  cat("log OR:")
  cat(mean(boots$mean_log_OR))
  cat("\n")
  cat("\n")
  cat("se_OR:")
  cat(mean(boots$se_OR))
  cat("\n")
  cat("\n")
  cat("95% CI for log odds ratio:")
  cat(mean(boots$mean_log_OR) - 1.96*mean(boots$se_OR),
      ", ",

```

```

    mean(boots$mean_log_OR) + 1.96*mean(boots$se_OR))
  }
}

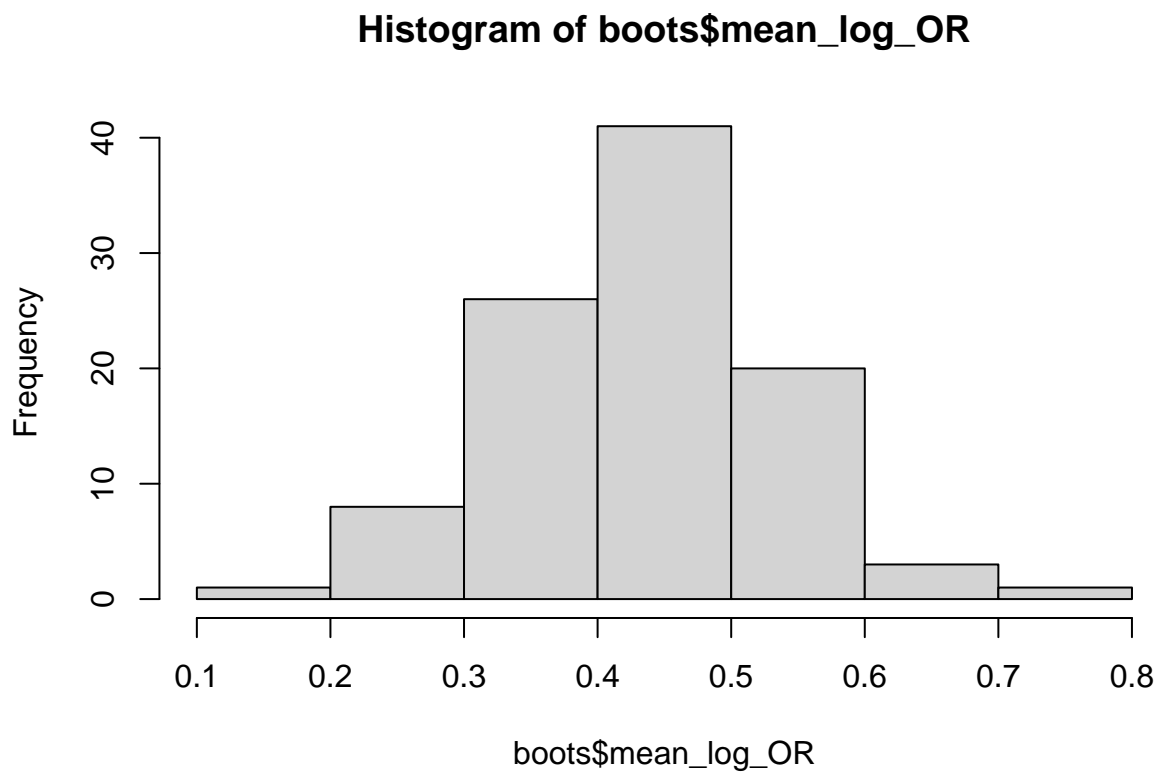
```

```

## ATE:0.1064552
##
## se_ATE:0.02492412
##
## 95% CI for ATE:0.05760391 , 0.1553065
##
## log OR:0.434012
##
## se_OR:0.1739941
##
## 95% CI for log odds ratio:0.09298361 , 0.7750404

```

```
hist(boots$mean_log_OR)
```



```
hist(boots$se_OR)
```

