

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 = "Y0",
                    formula = "-1.5 + 1.77*L2 - 0.3*L3",
                    dist = "binary", link = "logit")
pre_data <- defData(pre_data, varname = "Y1",
                    formula = "-0.7 + 1.77*L2 + 0.7*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(369)
df <- genData(1000, pre_data)
odds <- function(p) {
  return((p/(1 - p)))
}

sd_log_odds <- function(p1, p2, n1, n2){
  return(sqrt(1/(n1*p1) + 1/(n1*(1-p1)) + 1/(n2*p2) + 1/(n2*(1-p2))))
}

log(odds(mean(df$Y1)) / odds(mean(df$Y0)))

## [1] 0.5643276

sd_log_odds(mean(df$Y1), mean(df$Y0), 1000, 1000)

## [1] 0.09790649
```

```
mean(df$Y1-df$Y0)
```

```
## [1] 0.12
```

```
sd(df$Y1-df$Y0)
```

```
## [1] 0.5548945
```

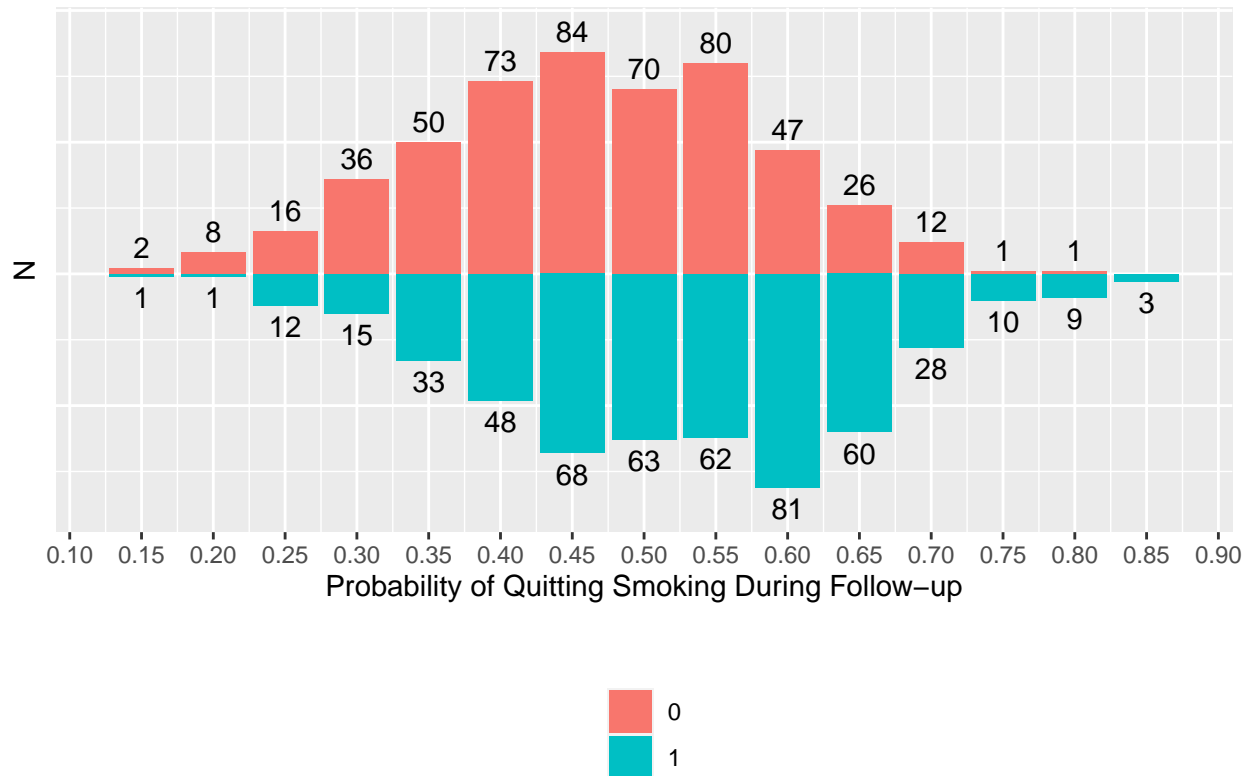
Propensity Score Model

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.
```

Propensity Score Distribution by Treatment Group



Logit Outcome Model

```
msm.w <- glm(Y ~ A , data = df_with_weight, family = "binomial", weights = w)
```

```
summary(msm.w)
```

```
##
## Call:
## glm(formula = Y ~ A, family = "binomial", data = df_with_weight,
##      weights = w)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.862  -1.656   1.211   1.534   2.740
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.08814    0.06355  -1.387   0.165
## A           0.53706    0.09062   5.926 3.1e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
## Null deviance: 2756.4 on 999 degrees of freedom
## Residual deviance: 2720.9 on 998 degrees of freedom
## AIC: 2776.6
##
## Number of Fisher Scoring iterations: 4
```

Nearest neighbor propensity score matching

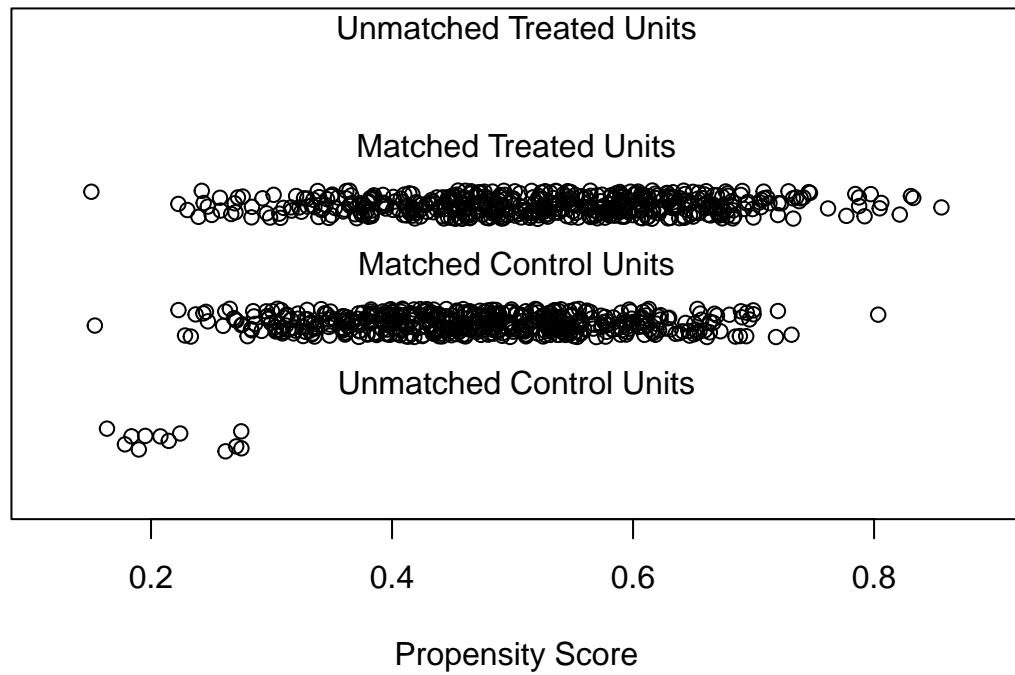
```
matched <- matchit(A ~ L1 + L2 + L3, data = df,
                   distance = "glm", link = "logit",
                   method = "nearest", ratio = 1)
```

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

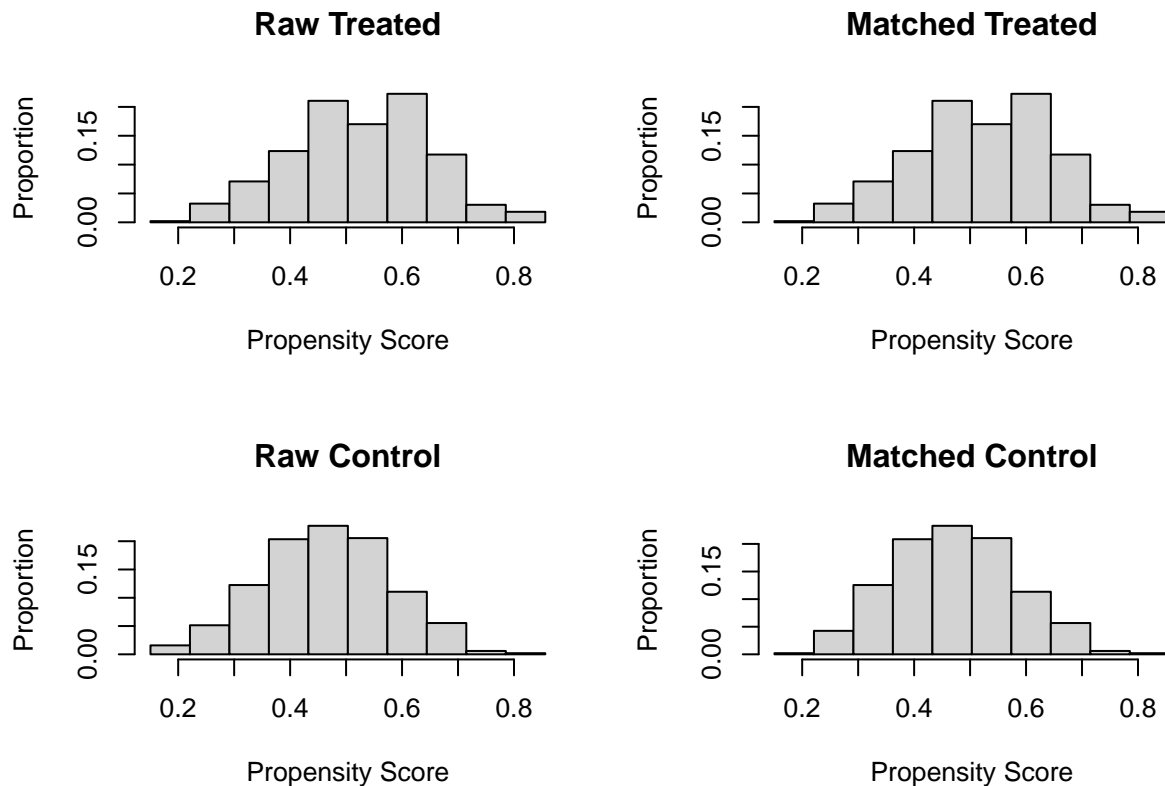
```
## $nn
##           Control Treated
## All (ESS)      506      494
## All           506      494
## Matched (ESS)  494      494
## Matched       494      494
## Unmatched      12        0
## Discarded       0        0
```

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

Distribution of Propensity Scores



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



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

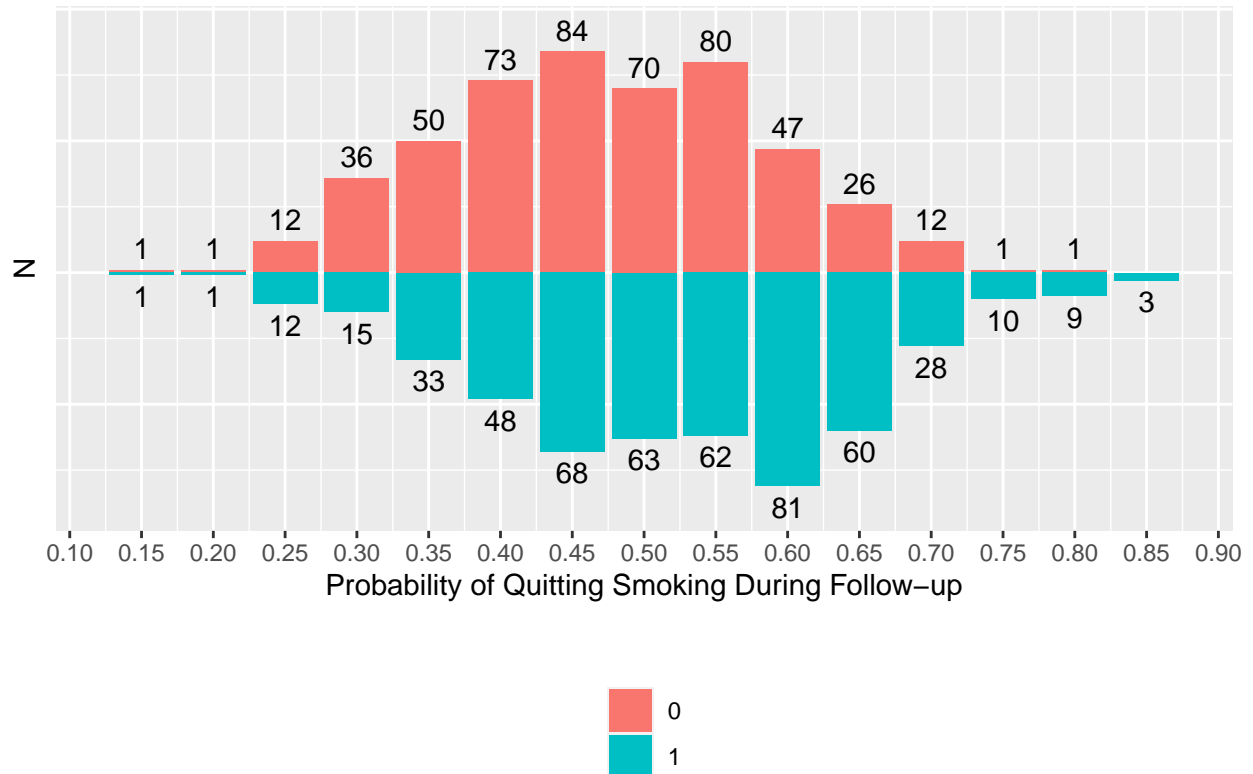
495 pairs propensity score distribution

```
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),494, 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))
  }
}

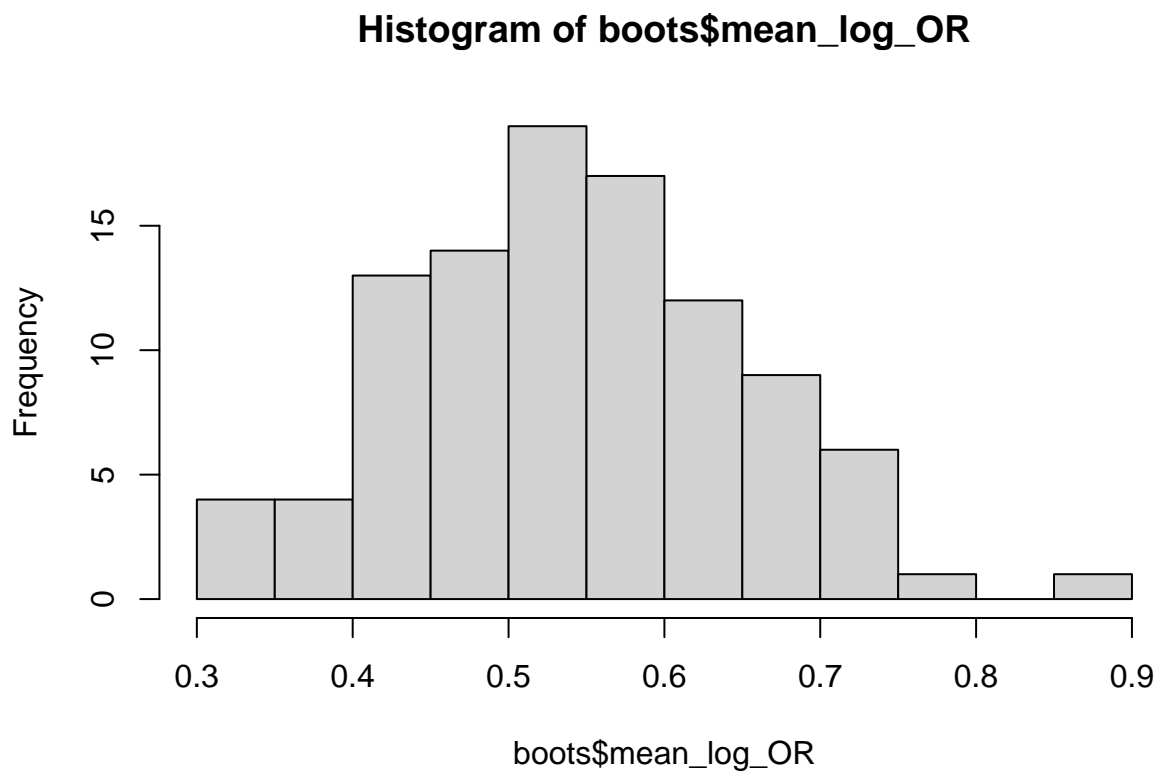
```

```

## ATE:0.1319943
##
## se_ATE:0.02595282
##
## 95% CI for ATE:0.08112679 , 0.1828619
##
## log OR:0.5450371
##
## se_OR:0.1675714
##
## 95% CI for log odds ratio:0.2165972 , 0.8734769

```

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



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

