PS Bootstrap Binary Standardization

Hun

2/9/2022

Generating data with true log odds ratio and its standard deviation

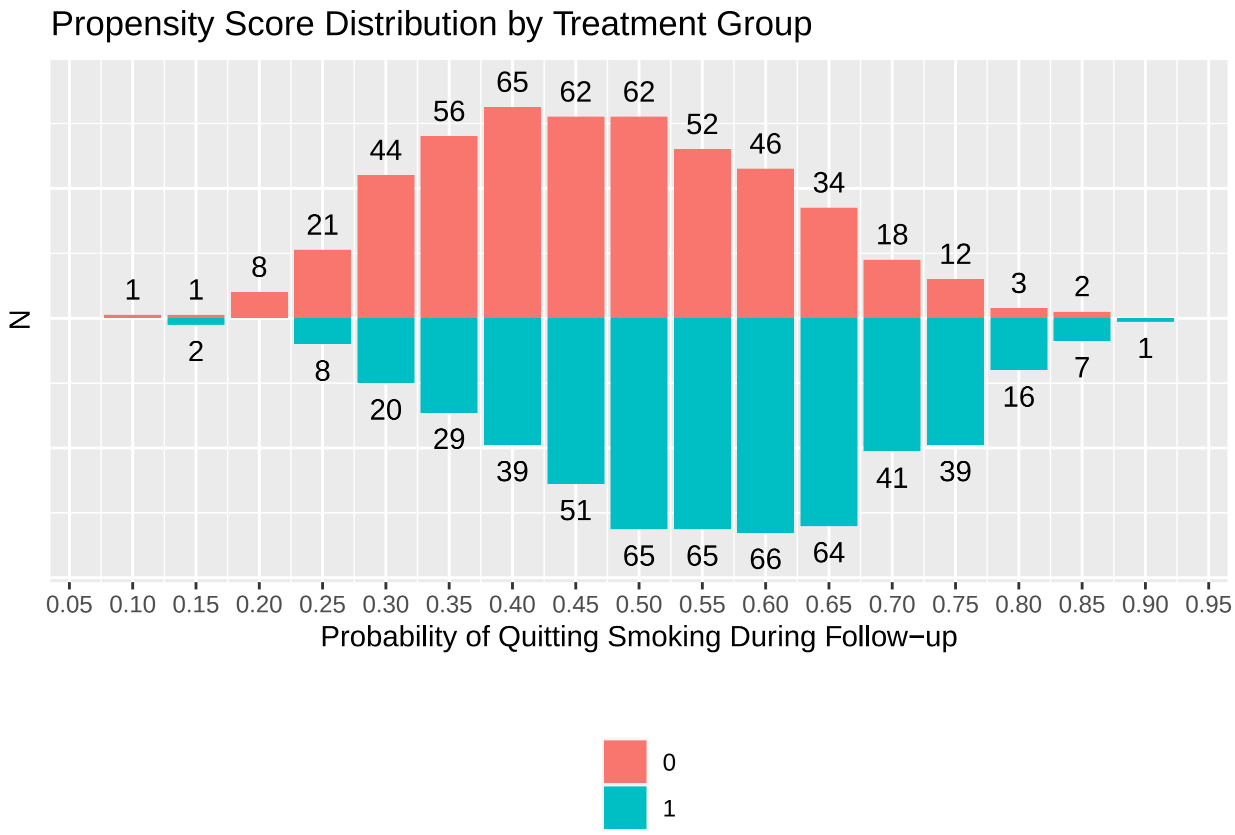
```
pre_data <- defData(varname = "L1", formula = "0", variance = 1,
                dist = "normal")
pre_data <- defData(pre_data, varname = "L2", formula = "0", variance = 1,
                dist = "normal")
pre_data <- defData(pre_data, varname = "L3", formula = "0", variance = 1,
                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.
```



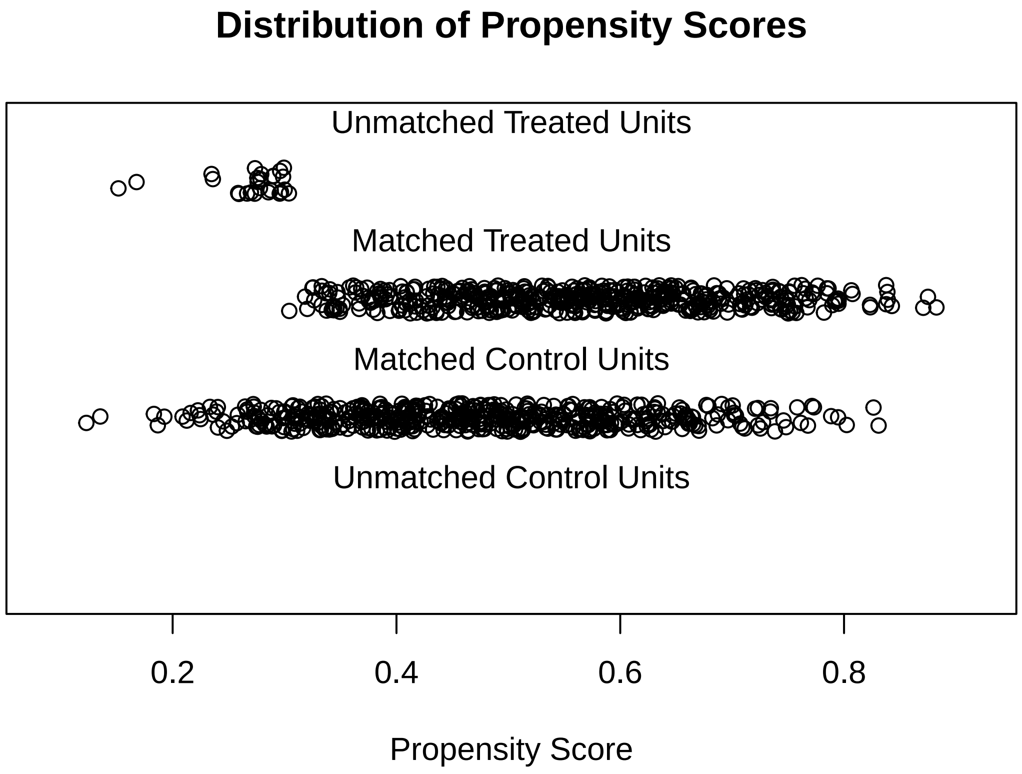
Nearest neighbor propensity score matching

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

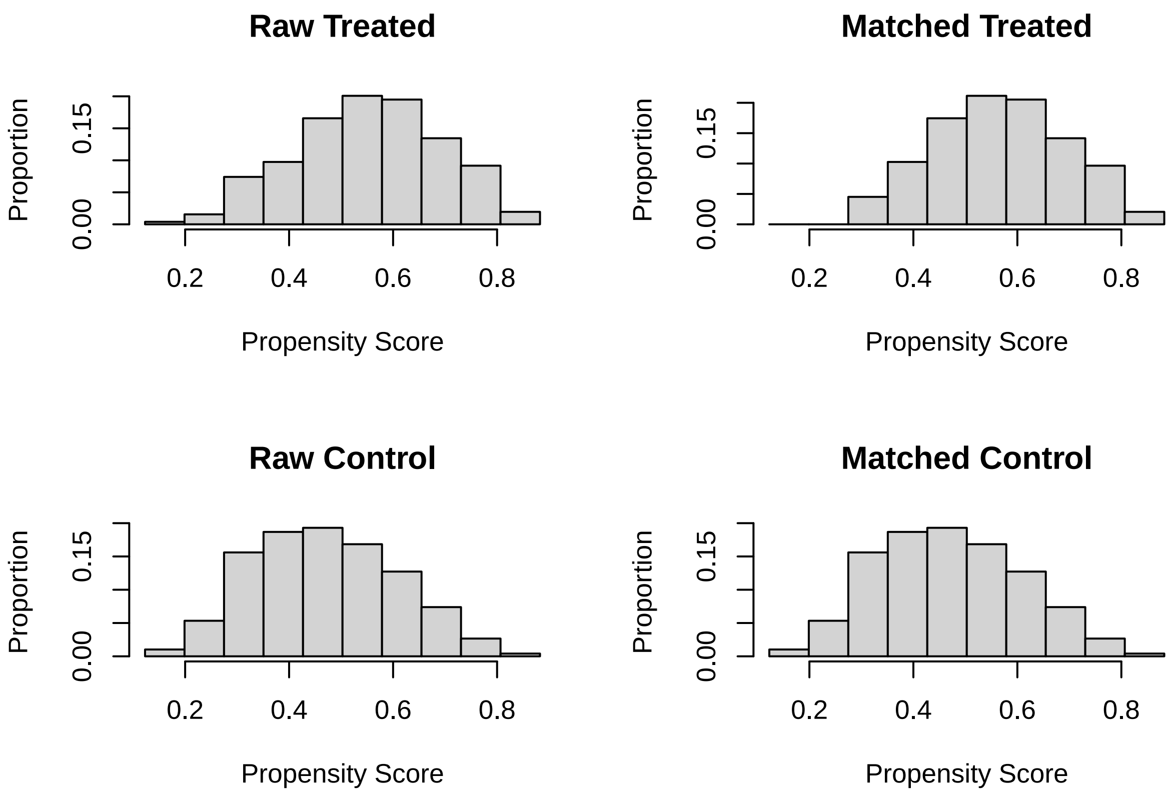
```
summary(matched)[2]
```

```
## $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)
```



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

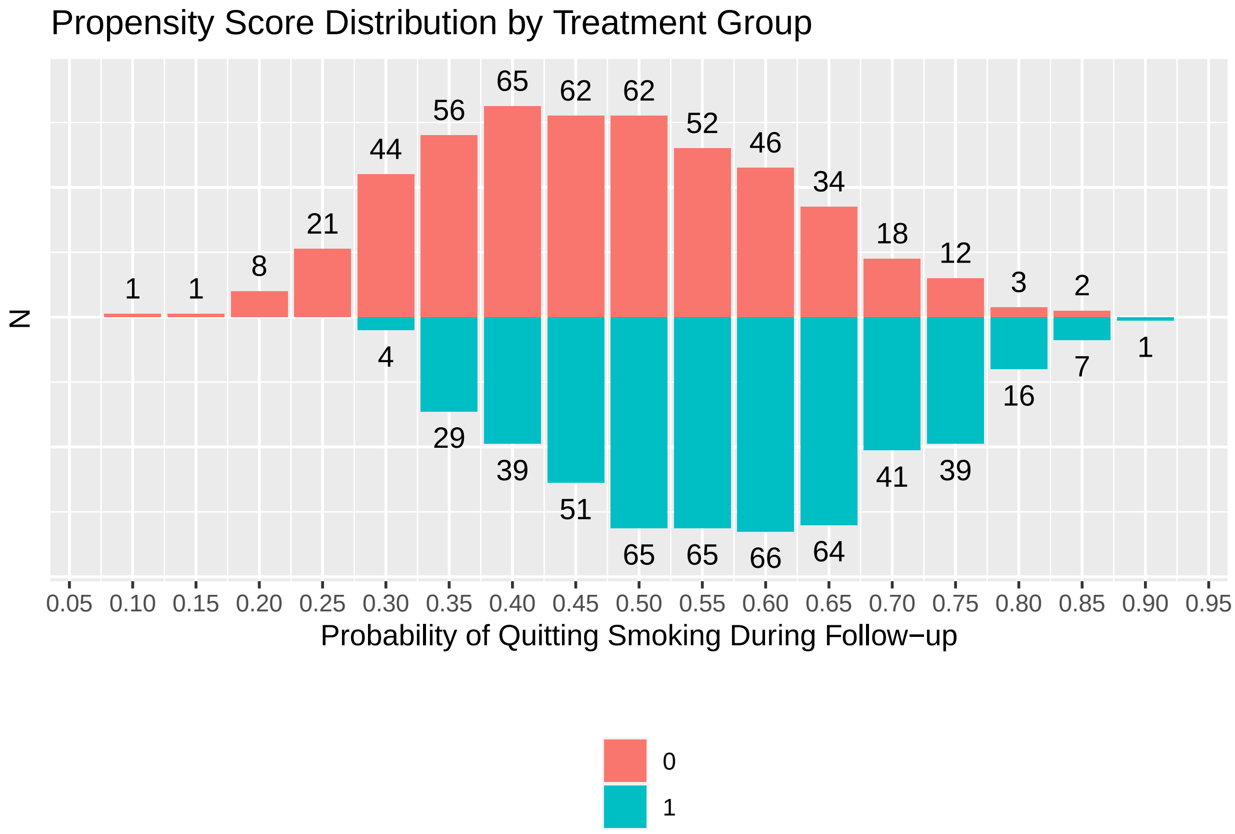


```
matched_df <-
  match.data(matched)
```

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.
```



simple bootstrap

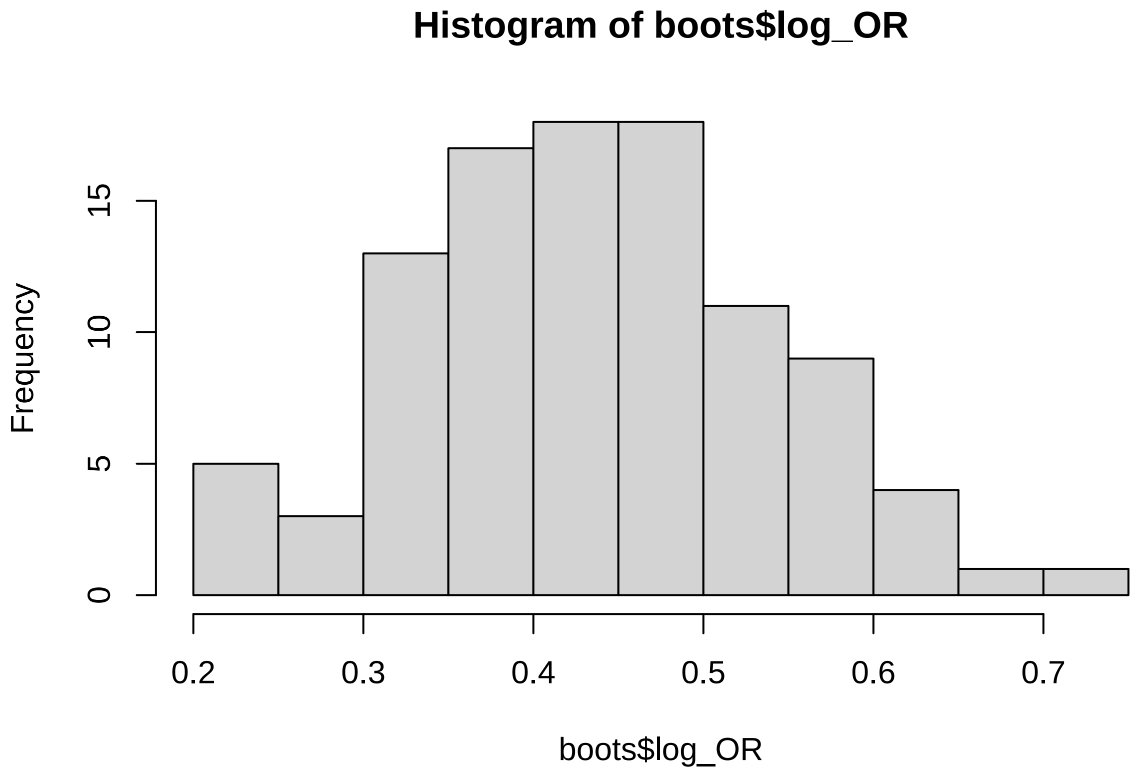
```
nboot <- 100
# set up a matrix to store results
boots <- data.frame(i = 1:nboot,
                    se_ATE = NA,
                    se_OR = NA,
                    log_OR = NA,
                    mean1 = NA,
                    mean0 = NA,
                    difference = NA
                 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

for (i in 1:nboot) {
  # sample with replacement
  sampl <- matched_df %>% filter(subclass %in% sample(levels(subclass),500, replace =  TRUE))

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

  # create new data sets
```

```r
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, "log_OR"] <- summary(bootmod)$coeff[2,1]

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

boots[i, "se_ATE"] <-
  sqrt((summary(bootmod)$coeff[2,2]*mean(sampl.treated$pred.y) *
      (1 - mean(sampl.treated$pred.y)))^2 +
  (summary(bootmod)$coeff[2,2]*mean(sampl.untreated$pred.y) *
      (1 - mean(sampl.untreated$pred.y)))^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"]

mean_log_OR <- mean(boots$log_OR)

Empirical_se_ATE <- sd(boots$difference)

mean_se_ATE <- mean(boots$se_ATE)

Empirical_se_log_OR <- sd(boots$log_OR)

mean_se_log_OR <- mean(boots$se_OR)

ATE <- mean(boots$difference)

# once loop is done, print the results
if (i == nboot) {
  cat("ATE:")
  cat(ATE)
  cat("\n")
  cat("\n")
  cat("Empirical_se_ATE:")
  cat(Empirical_se_ATE)
  cat("\n")
  cat("\n")
  cat("mean_se_ATE:")
```

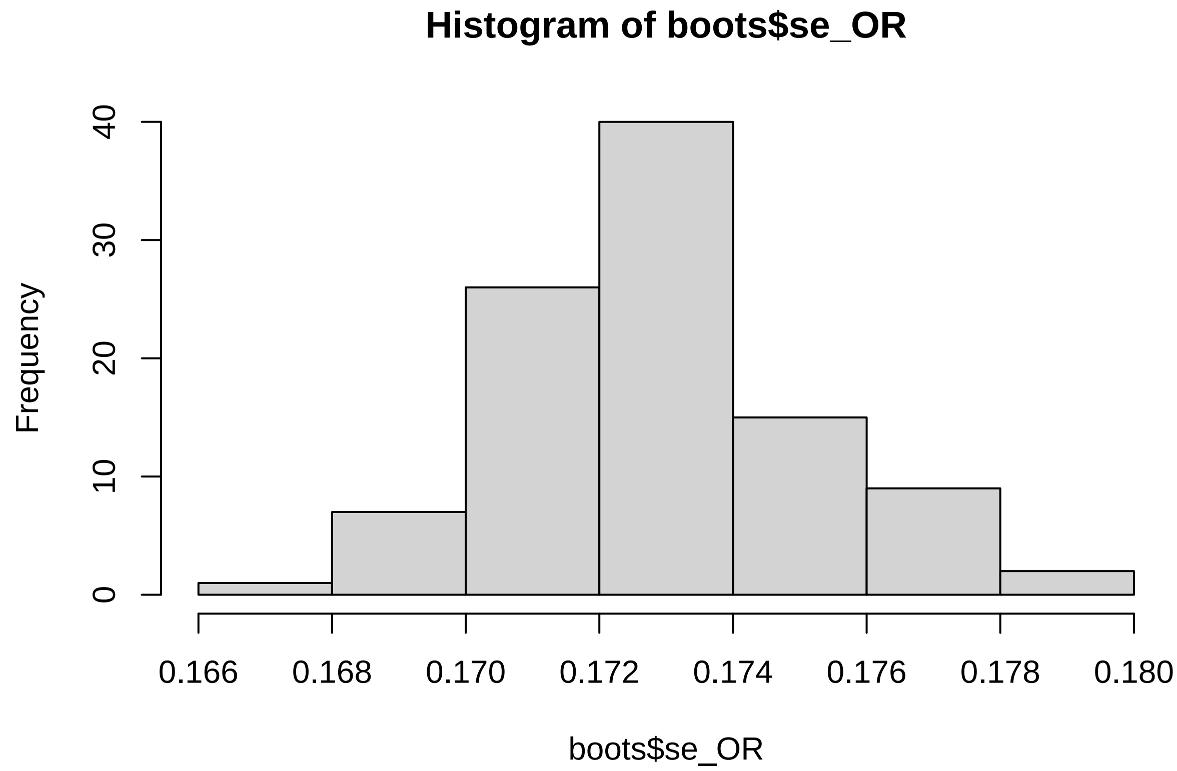
```
    cat(mean_se_ATE)
    cat("\n")
    cat("\n")
    cat("95% CI for ATE:")
    cat(ATE - 1.96*Empirical_se_ATE,
        ",",
        ATE + 1.96*Empirical_se_ATE)
    cat("\n")
    cat("\n")
    cat("mean_log_OR:")
    cat(mean_log_OR)
    cat("\n")
    cat("\n")
    cat("Empirical_se_log_OR:")
    cat(Empirical_se_log_OR)
    cat("\n")
    cat("\n")
    cat("mean_se_log_OR:")
    cat(mean_se_log_OR)
    cat("\n")
    cat("\n")
    cat("95% CI for log odds ratio:")
    cat(mean_log_OR - 1.96*mean_se_log_OR,
        ",",
        mean_log_OR + 1.96*mean_se_log_OR)
  }
}
```

```
## ATE:0.1068614
##
## Empirical_se_ATE:0.025629
##
## mean_se_ATE:0.0601082
##
## 95% CI for ATE:0.05662853 , 0.1570942
##
## mean_log_OR:0.4353995
##
## Empirical_se_log_OR:0.1046255
##
## mean_se_log_OR:0.172854
##
## 95% CI for log odds ratio:0.09660568 , 0.7741934
```

```
hist(boots$log_OR)
```



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



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


summary(a)
```

```
##
## Call:
## glm(formula = Y ~ A + ps, family = binomial, data = sampl, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5387  -1.1270   0.8968   1.0713   1.4361
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8868     0.3264  -2.717  0.00660 **
## A             0.5997     0.1728   3.470  0.00052 ***
## ps            1.3180     0.6373   2.068  0.03863 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 868.96  on 627  degrees of freedom
## Residual deviance: 844.52  on 625  degrees of freedom
## AIC: 850.52
##
## Number of Fisher Scoring iterations: 4
```