PS Bootstrap Continuous Outcome

Hun

2/7/2022

```
## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'simstudy' was built under R version 4.1.2

## Warning: package 'MatchIt' was built under R version 4.1.2
```

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 = "2 + 1.5*A + 0.8*L2 + -0.1*L3",
                dist = "nonrandom")

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

# TRUE ATE: 1.5
```

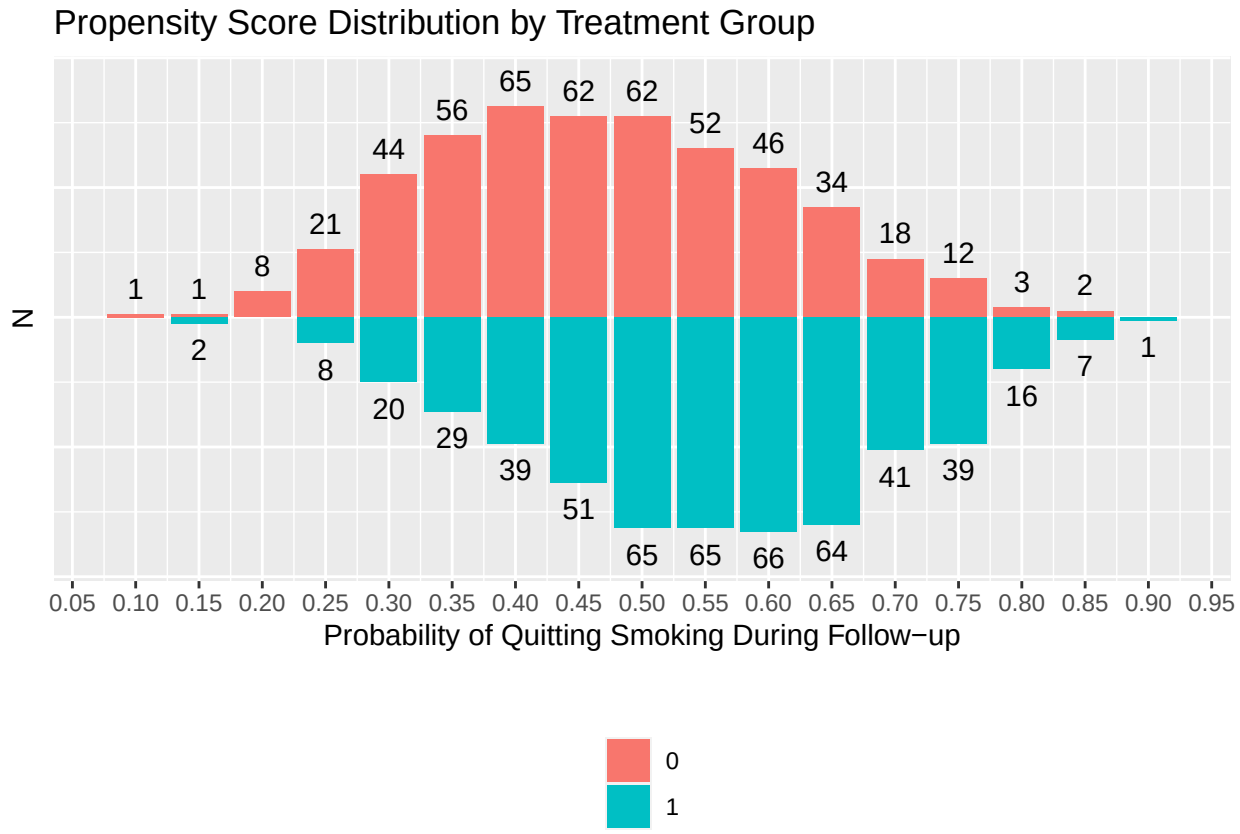
Propensity Score Model

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.
```



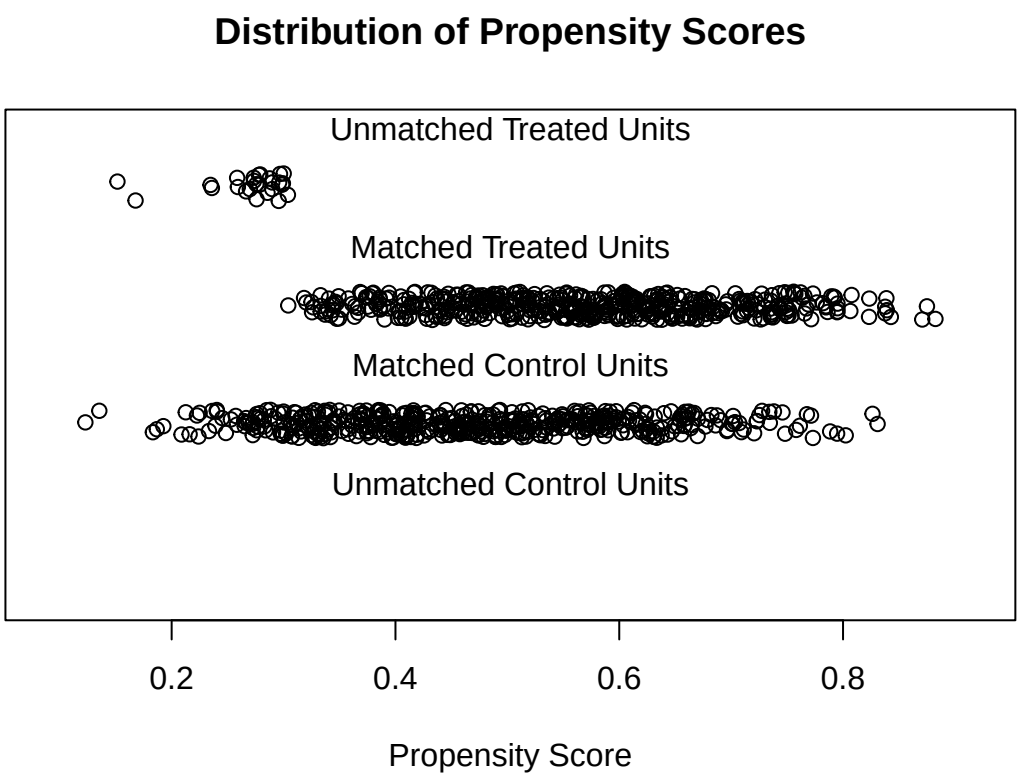
Nearest neighbor propensity score matching

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

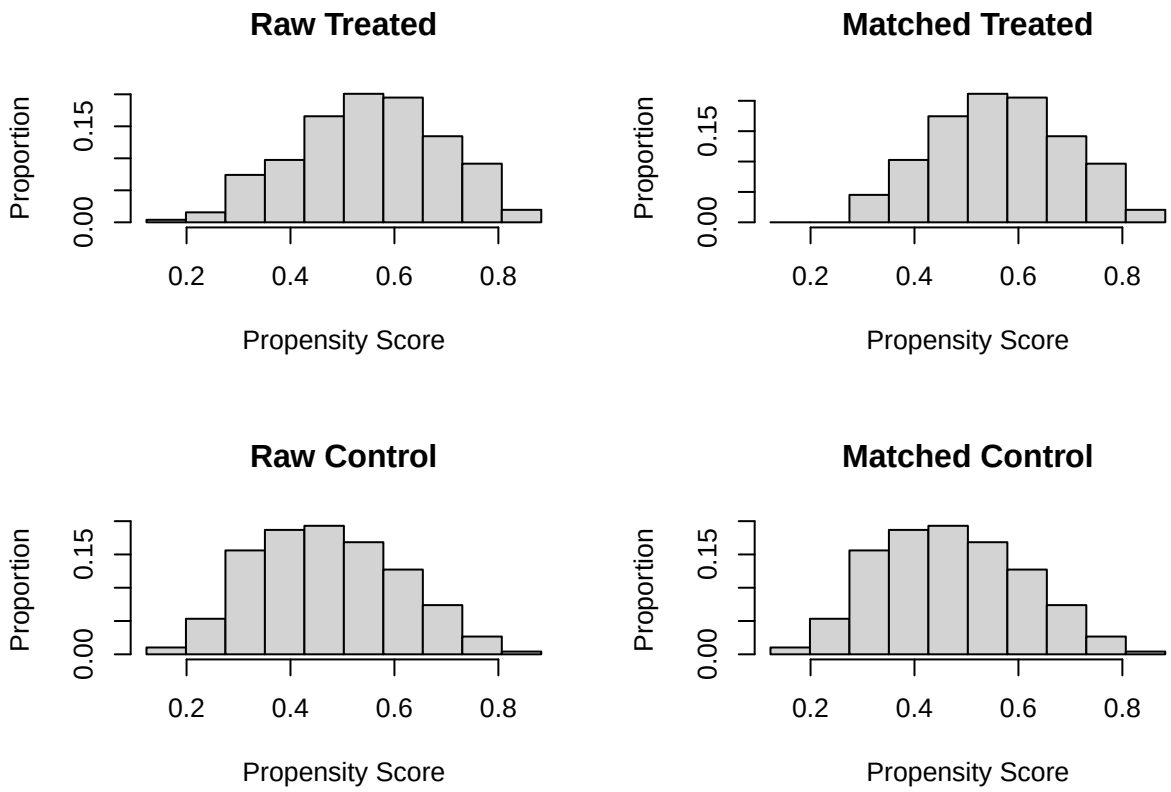
```
summary(matched)[2]
```

```
## $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")
```



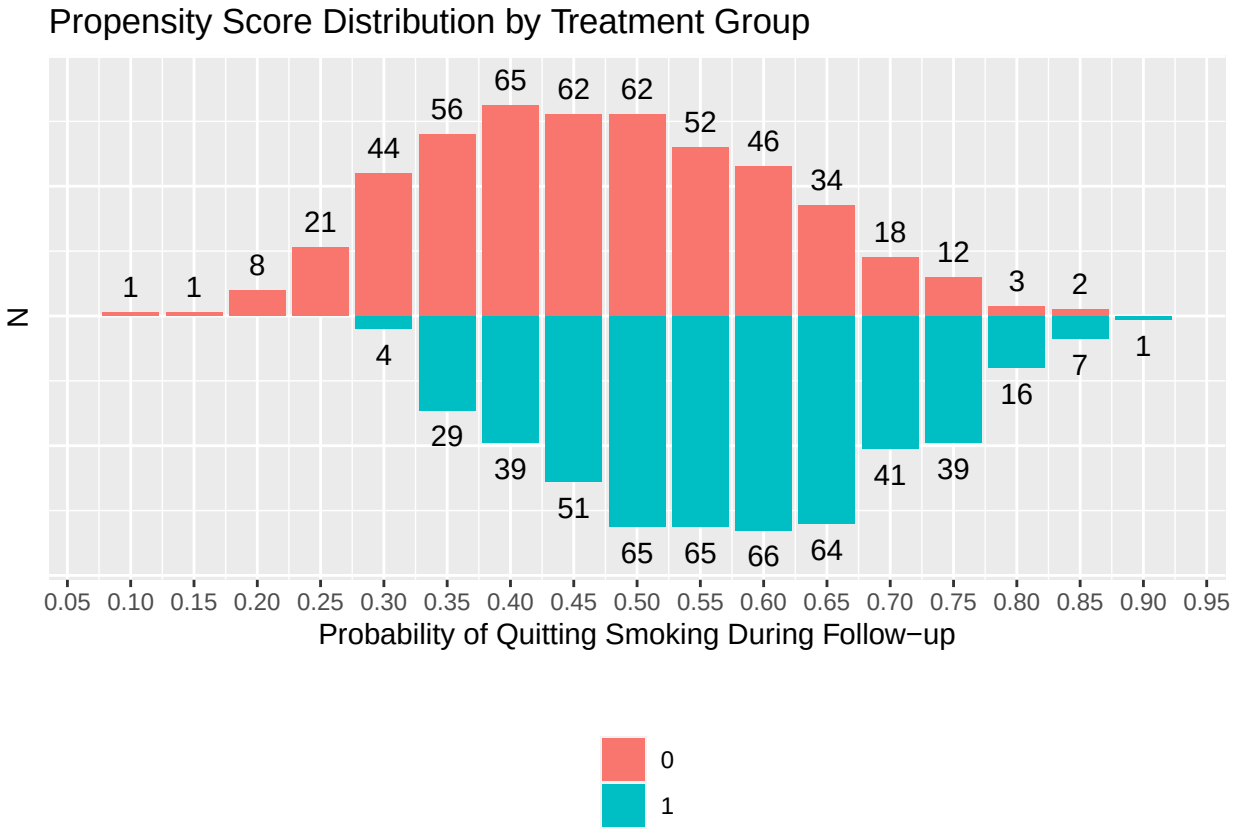

```
matched_df <-
  match.data(matched)
```

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.
```



Empirical standard deviation of the estimated effects

```
empirical_df <-
  matched_df %>% select(A,Y,subclass) %>%
  pivot_wider(values_from = Y, names_from = A) %>%
  group_by(subclass) %>%
  summarise(difference = `1` - `0`)

sd(empirical_df$difference)
```

```
## [1] 1.133379
```

```
summary(glm(Y ~ A + ps, data = matched_df,
                weights = weights))
```

```
##
## Call:
## glm(formula = Y ~ A + ps, data = matched_df, weights = weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9065  -0.5232  -0.0199   0.5371   2.8169
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.21996    0.09899  12.324  < 2e-16 ***
## A            1.49155    0.05477  27.233  < 2e-16 ***
## ps           1.50497    0.19575   7.688 3.65e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.6408273)
##
##     Null deviance: 1314.25  on 973  degrees of freedom
## Residual deviance:  622.24  on 971  degrees of freedom
## AIC: 2335.7
##
## Number of Fisher Scoring iterations: 2
```

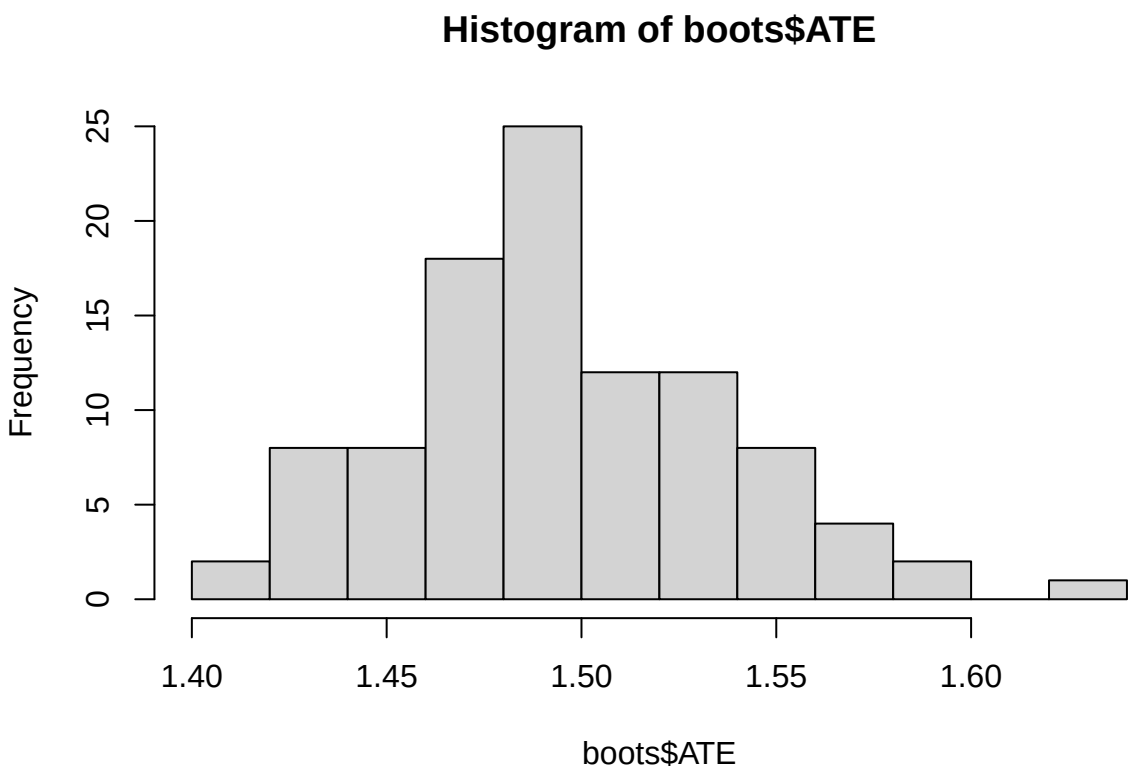
simple bootstrap

```
nboot <- 100
# set up a matrix to store results
boots <- data.frame(i = 1:nboot,
                    se = NA,
                    mean1 = NA,
                    mean0 = NA,
                    beta1 = NA,
                    ATE = 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)

  # 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)

  sampl.untreated$pred.y <-
    predict(bootmod, sampl.untreated)

   # output results
```

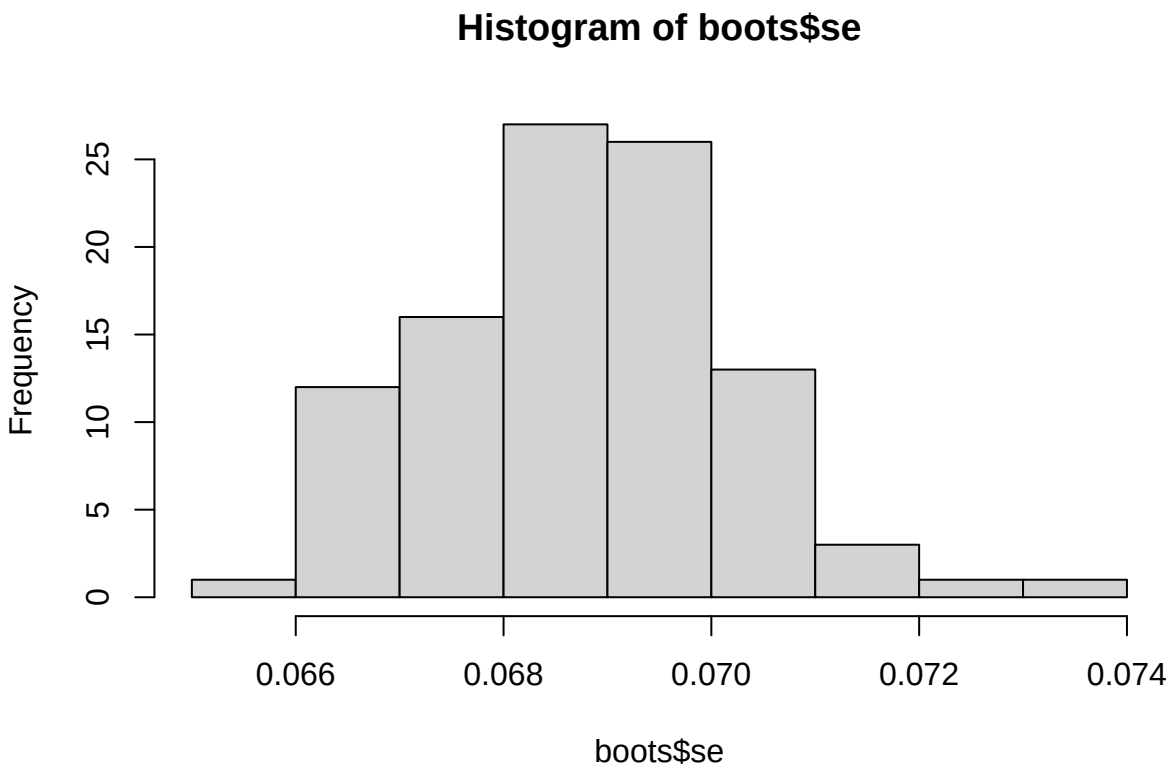
```
  boots[i, "beta1"] <- summary(bootmod)$coeff[2,1]
  boots[i, "se"] <- summary(bootmod)$coeff[2,2]
  boots[i, "mean1"] <- mean(sampl.treated$pred.y)
  boots[i, "mean0"] <- mean(sampl.untreated$pred.y)
  boots[i, "ATE"] <- boots[i, "mean1"] - boots[i, "mean0"]

  Empirical_sd <- sd(boots$ATE)

  ATE <- mean(boots$ATE)

  mean_se <- mean(boots$se)

  # once loop is done, print the results
  if (i == nboot) {
    cat("ATE:")
    cat(ATE)
    cat("\n")
    cat("\n")
    cat("Empirical_sd:")
    cat(Empirical_sd)
    cat("\n")
    cat("\n")
    cat("mean_se:")
    cat(mean_se)
    cat("\n")
    cat("\n")
    cat("95% CI for ATE:")
    cat(ATE - 1.96*mean_se,
        ",",
        ATE + 1.96*mean_se)
    cat("\n")
    cat("\n")
    cat("ATE from beta:") #checking if our computation is correct
    cat(mean(boots$beta1))

  }
}
```

```
## ATE:1.495169
##
## Empirical_sd:0.04175198
##
## mean_se:0.06871705
##
## 95% CI for ATE:1.360483 , 1.629854
##
## ATE from beta:1.495169
```

```
hist(boots$ATE)
```




```
hist(boots$se)
```


Histogram of boots$se
Frequency
0
5
10
15
20
25
0.066
0.068
0.070
0.072
0.074
boots$se