R Code for An Evaluation of the Causal Effect of a Promotion Campaign on Software Usage

Febriany & Maysen

April 2024

```
library(boot)
library(tidyverse)
library(haven)
```

```
set.seed(123)
msdata <- read_csv("promotion.csv")
msdata <- msdata %>% filter(Revenue>0)
size<-floor(dim(msdata)[1]*0.5)
samp_idx <-sample(1:dim(msdata)[1],size,FALSE)
first_fold <- msdata[samp_idx,]
second_fold <- msdata[-samp_idx,]
```

Covariate Balance Check

```
#covariate balance check
treatment <- msdata %>% filter(Discount==1)
control <- msdata %>% filter(Discount==0)
#customer's size given by their yearly total revenue
par(mfrow=c(2,2))
hist(log(treatment$Size), main = "Customer Size Under Treatment",
     xlab = "Customer Size")
hist(log(control$Size), main = "Customer Size Under Control",
     xlab = "Customer Size")
#number of employees
par(mfrow=c(2,1))
hist(treatment$`Employee Count`, main = "Employee Count Under Treatment",
     xlab = "Employee Count")
hist(control$`Employee Count`, main = "Employee Count Under Control",
     xlab = "Employee Count")
#whether the customer's business is commercial
par(mfrow=c(2,2))
hist(treatment$`Commercial Flag`, main = "Commercial Flag Under Treatment",
     xlab = "Commercial Flag")
hist(control$`Commercial Flag`, main = "Commercial Flag Under Control",
     xlab = "Commercial Flag")
#whether the customer is a Small Medium Corporation
par(mfrow=c(2,2))
hist(treatment$`SMC Flag`, main = "Small Medium Corporation Flag Under Treatment",
     xlab = "Small Medium Corporation Flag")
hist(control$`SMC Flag`, main = "Small Medium Corporation Flag Under Control",
     xlab = "Small Medium Corporation Flag")
```

```r
#whether the customer is a large consumer in their industry
par(mfrow=c(2,2))
hist(treatment$`Major Flag`, main = "Large Consumer Flag Under Treatment",
     xlab = "Large Consumer Flag")
hist(control$`Major Flag`, main = "Large Consumer Flag Under Control",
     xlab = "Large Consumer Flag")
#whether the customer has global offices
par(mfrow=c(2,2))
hist(treatment$`Global Flag`, main = "Global Offices Flag Under Treatment",
     xlab = "Global Offices Flag")
hist(control$`Global Flag`, main = "Global Offices Flag Under Control",
     xlab = "Global Offices Flag")
```

OR Estimator

```r
set.seed(123)
calculate_ate_bootstrap <- function(data, indices) {
  bootstrap_sample <- data[indices, ]

  treatment_group <- bootstrap_sample %>% filter(Discount == 1)
  control_group <- bootstrap_sample %>% filter(Discount == 0)

    lm_m1 <- lm(Revenue ~ `Global Flag` + `Major Flag` + `SMC Flag` + `Commercial Flag` +
                  `Employee Count` + `Size`, data = treatment_group)
  lm_m0 <- lm(Revenue ~ `Global Flag` + `Major Flag` + `SMC Flag` + `Commercial Flag` +
                `Employee Count` + `Size`, data = control_group)

  mi1 <- mean(predict(lm_m1, second_fold))
  mi0 <- mean(predict(lm_m0, second_fold))
  ate <- mi1 - mi0

  return(ate)
}

set.seed(123)
boot_ate <- boot(data = first_fold, statistic = calculate_ate_bootstrap, R = 1000)
boot_ci <- boot.ci(boot_ate, type = "perc")
print(boot_ate)
print(boot_ci)

#Histogram OR ATE DIstribution
hist(boot_ate$t, breaks = 30, main = "OR ATE Distribution", xlab = "ATE",
     col = "lightblue", border = "black")
abline(v = boot_ate$t0, col = "blue", lwd = 2)
abline(v = boot_ci$percent[4:5], col = "red", lwd = 2)
legend("topright", legend = c("ATE", "Point Estimate", "95% CI"),
       col = c("lightblue", "blue", "red"), lty = c(0, 1, 1), lwd = c(3, 2, 2), bg = "white")
text(5570, 95, "5472.99")
text(6035, 40, "5949")
text(4940, 40, "5023")

# Plot density plot of ATE
plot(density(boot_ate$t), main = "ATE Distribution", xlab = "ATE", col = "blue", lwd = 2)
abline(v = boot_ci$percent[4:5], col = "red", lwd = 2)
```

```
legend("topright", legend = c("ATE", "95% CI"), col = c("blue", "red"), lty = c(1, 1),
       lwd = c(2, 2), bg = "white")
```

Hajek Estimator

```
set.seed(123)
calculate_hajek_bootstrap <- function(data, indices) {
  bootstrap_sample <- data[indices, ]

  prop_model <- glm(Discount ~ `Global Flag` + `Major Flag` + `SMC Flag`
                    + `Commercial Flag` + `Employee Count` + `Size`,
                    family = binomial(link="logit"), data = bootstrap_sample)
  pscore_hat <- predict(prop_model, newdata = second_fold, type="response")

 mi1 <- mean((second_fold$Discount/pscore_hat)/(mean(second_fold$Discount/pscore_hat))
             *second_fold$Revenue)
 mi0 <- mean(((1-second_fold$Discount)/(1-pscore_hat))/
               (mean((1-second_fold$Discount)/(1-pscore_hat)))*second_fold$Revenue)
 hajek <- mi1-mi0

  return(hajek)
}

boot_hajek <- boot(data = first_fold, statistic = calculate_hajek_bootstrap, R = 1000)
boot_ci <- boot.ci(boot_hajek, type = "perc")
print(boot_hajek)
print(boot_ci)

#Histogram Hajek ATE DIstribution
hist(boot_hajek$t, breaks = 30, main = "Hajek ATE Distribution", xlab = "ATE",
     col = "lightblue", border = "black")
abline(v = boot_hajek$t0, col = "blue", lwd = 2)
abline(v = boot_ci$percent[4:5], col = "red", lwd = 2)
legend("topright", legend = c("ATE", "95% CI"), col = c("blue", "red"),
       lty = c(1, 1), lwd = c(2, 2), bg = "white")
text(5520, 120, "5779.70")
text(7150, 60, "6957")
text(4000, 60, "4244")
```

Doubly Robust Estimator

```
set.seed(123)
calculate_dr_bootstrap <- function(data, indices) {
  bootstrap_sample <- data[indices, ]

 prop_model <- glm(Discount ~ `Global Flag` + `Major Flag` + `SMC Flag`
                   + `Commercial Flag` + `Employee Count` + `Size`,
                   family = binomial(link="logit"), data = bootstrap_sample)
  pscore_hat <- predict(prop_model, newdata = second_fold, type="response")

  treatment_group <- bootstrap_sample %>% filter(Discount == 1)
  control_group <- bootstrap_sample %>% filter(Discount == 0)

  lm_m1 <- lm(Revenue ~ `Global Flag` + `Major Flag` + `SMC Flag`
              + `Commercial Flag` + `Employee Count` + `Size`, data = treatment_group)
```

```
  lm_m0 <- lm(Revenue ~ `Global Flag` + `Major Flag` + `SMC Flag`
              + `Commercial Flag` + `Employee Count` +  `Size`, data = control_group)

  mi1 <- predict(lm_m1, second_fold)
  mi0 <- predict(lm_m0, second_fold)

  dr1 <- mean((second_fold$Discount*second_fold$Revenue/pscore_hat)
              -((second_fold$Discount-pscore_hat)/pscore_hat*mi1))
  dr0 <- mean(((1-second_fold$Discount)*second_fold$Revenue
               /(1-pscore_hat)) - ((pscore_hat-second_fold$Discount)
                                   /(1-pscore_hat)*mi0))
 dr <- dr1-dr0

  return(dr)
}

boot_dr <- boot(data = first_fold, statistic = calculate_dr_bootstrap, R = 1000)
boot_ci <- boot.ci(boot_dr, type = "perc")
print(boot_dr)
print(boot_ci)

#Histogram Doubly Robust ATE DIstribution
hist(boot_dr$t, breaks = 30, main = "Doubly Robust ATE Distribution",
     xlab = "ATE", col = "lightblue", border = "black")
abline(v = boot_dr$t0, col = "blue", lwd = 2)
abline(v = boot_ci$percent[4:5], col = "red", lwd = 2)
legend("topright", legend = c("ATE", "95% CI"), col = c("blue", "red"),
       lty = c(1, 1), lwd = c(2, 2), bg = "white")
text(5575, 150, "5527.63")
text(5690, 60, "5661")
text(5380, 60, "5407")
```

Sensitivity Analysis

```
set.seed(123)

calculate_ate_bootstrap <- function(data, indices, E0, E1) {
  bootstrap_sample <- data[indices, ]

  treatment_group <- bootstrap_sample %>% filter(Discount == 1)
  control_group <- bootstrap_sample %>% filter(Discount == 0)

  ate_values <- numeric(length(E0))

  for (i in seq_along(E0)) {
  lm_m1 <- lm(Revenue ~ `Global Flag` + `Major Flag` + `SMC Flag` +
                `Commercial Flag` + `Employee Count` + `Size`, data = treatment_group)
  lm_m0 <- lm(Revenue ~ `Global Flag` + `Major Flag` + `SMC Flag`
              + `Commercial Flag` + `Employee Count` + `Size`, data = control_group)

  mi1 <- mean(predict(lm_m1, second_fold))
  mi0 <- mean(predict(lm_m0, second_fold))

  ate_values[i] <- mean(second_fold$Discount * mi1) +
```

```
    mean((1 - second_fold$Discount) * (mi1 / E1[i])) -
    mean(second_fold$Discount * mi0 * E0[i]) - mean((1 - second_fold$Discount) * mi0)}

  return(ate_values)
}


# Initialize vectors for E0 and E1
E1 <- c(1/2, 1/1.7, 1/1.5, 1/1.3, 1, 1.3, 1.5, 1.7, 2)
E0 <- c(1/2, 1/1.7, 1/1.5, 1/1.3, 1, 1.3, 1.5, 1.7, 2)

# Initialize matrices to store ATE and SE values
ate_values <- matrix(NA, nrow = length(E0), ncol = length(E1))
se_values <- matrix(NA, nrow = length(E0), ncol = length(E1))

# Loop through each combination of E0 and E1
for (i in seq_along(E0)) {
  for (j in seq_along(E1)) {
    boot_ate <- boot(data = msdata, statistic = calculate_ate_bootstrap,
                     R = 1000, E0 = E0[i], E1 = E1[j])
    ate_values[i, j] <- mean(boot_ate$t)
    se_values[i, j] <- sd(boot_ate$t)
  }
}

print(ate_values)
```

Weighted Balancing Property Check

```
#check if propensity score model satisfies weighted balancing property
global_prop <- function(data, indices){
  data2 <- data[indices,]
  set.seed(123)
  inds <- sample(seq_len(nrow(data2)), size = nrow(data2)/2)
  #split data into two folds
  nuisance <- data2[inds,]
  effect <- data2[-inds,]

  #estimate propensity score on nuisance fold
  #model for the propensity score (logistic regression of treatment on all covariates)
  prop_mod <- glm(Discount ~ .-Revenue, family = binomial(link="logit"), data = nuisance)
  #predict propensity scores on effect fold
  pi_hat <- predict(prop_mod, effect, type = "response")
  #check weighted balancing property on effect fold
  wbp <- mean(((effect$Discount/pi_hat)-1)*effect$`Global Flag`)
  return(wbp)
}

bs <- suppressWarnings(boot(msdata, global_prop, R = 1000))
quantile(bs$t, probs = c(0.025, 0.975))

major_prop <- function(data, indices){
  data2 <- data[indices,]
  set.seed(123)
```

```r
  inds <- sample(seq_len(nrow(data2)), size = nrow(data2)/2)
  #split data into two folds
  nuisance <- data2[inds,]
  effect <- data2[-inds,]

  #estimate propensity score on nuisance fold
  #model for the propensity score (logistic regression of treatment on all covariates)
  prop_mod <- glm(Discount ~ .-Revenue, family = binomial(link="logit"), data = nuisance)
  #predict propensity scores on effect fold
  pi_hat <- predict(prop_mod, effect, type = "response")
  #check weighted balancing property on effect fold
  wbp <- mean(((effect$Discount/pi_hat)-1)*effect$`Major Flag`)
  return(wbp)
}

bs <- suppressWarnings(boot(msdata, major_prop, R = 1000))
quantile(bs$t, probs = c(0.025, 0.975))

SMC_prop <- function(data, indices){
  data2 <- data[indices,]
  set.seed(123)
  inds <- sample(seq_len(nrow(data2)), size = nrow(data2)/2)
  #split data into two folds
  nuisance <- data2[inds,]
  effect <- data2[-inds,]

  #estimate propensity score on nuisance fold
  #model for the propensity score (logistic regression of treatment on all covariates)
  prop_mod <- glm(Discount ~ .-Revenue, family = binomial(link="logit"), data = nuisance)
  #predict propensity scores on effect fold
  pi_hat <- predict(prop_mod, effect, type = "response")
  #check weighted balancing property on effect fold
  wbp <- mean(((effect$Discount/pi_hat)-1)*effect$`SMC Flag`)
  return(wbp)
}

bs <- suppressWarnings(boot(msdata, SMC_prop, R = 1000))
quantile(bs$t, probs = c(0.025, 0.975))

com_prop <- function(data, indices){
  data2 <- data[indices,]
  set.seed(123)
  inds <- sample(seq_len(nrow(data2)), size = nrow(data2)/2)
  #split data into two folds
  nuisance <- data2[inds,]
  effect <- data2[-inds,]

  #estimate propensity score on nuisance fold
  #model for the propensity score (logistic regression of treatment on all covariates)
  prop_mod <- glm(Discount ~ .-Revenue, family = binomial(link="logit"), data = nuisance)
  #predict propensity scores on effect fold
  pi_hat <- predict(prop_mod, effect, type = "response")
  #check weighted balancing property on effect fold
```

```
  wbp <- mean(((effect$Discount/pi_hat)-1)*effect$`Commercial Flag`)
  return(wbp)
}

bs <- suppressWarnings(boot(msdata, com_prop, R = 1000))
quantile(bs$t, probs = c(0.025, 0.975))

emp_prop <- function(data, indices){
  data2 <- data[indices,]
  set.seed(123)
  inds <- sample(seq_len(nrow(data2)), size = nrow(data2)/2)
  #split data into two folds
  nuisance <- data2[inds,]
  effect <- data2[-inds,]

  #estimate propensity score on nuisance fold
  #model for the propensity score (logistic regression of treatment on all covariates)
  prop_mod <- glm(Discount ~ .-Revenue, family = binomial(link="logit"), data = nuisance)
  #predict propensity scores on effect fold
  pi_hat <- predict(prop_mod, effect, type = "response")
  #check weighted balancing property on effect fold
  wbp <- mean(((effect$Discount/pi_hat)-1)*effect$`Employee Count`)
  return(wbp)
}

bs <- suppressWarnings(boot(msdata, emp_prop, R = 1000))
quantile(bs$t, probs = c(0.025, 0.975))

size_prop <- function(data, indices){
  data2 <- data[indices,]
  set.seed(123)
  inds <- sample(seq_len(nrow(data2)), size = nrow(data2)/2)
  #split data into two folds
  nuisance <- data2[inds,]
  effect <- data2[-inds,]

  #estimate propensity score on nuisance fold
  #model for the propensity score (logistic regression of treatment on all covariates)
  prop_mod <- glm(Discount ~ .-Revenue, family = binomial(link="logit"), data = nuisance)
  #predict propensity scores on effect fold
  pi_hat <- predict(prop_mod, effect, type = "response")
  #check weighted balancing property on effect fold
  wbp <- mean(((effect$Discount/pi_hat)-1)*effect$Size)
  return(wbp)
}

bs <- suppressWarnings(boot(msdata, size_prop, R = 1000))
quantile(bs$t, probs = c(0.025, 0.975))
```

Heatmap for Sensitivity Analysis

```
ate_df <- data.frame(E0 = rep(E0, each = length(E1)),
                     E1 = rep(E1, length(E0)),
                     ATE = as.vector(ate_values))
```

```r
# Define the labels
x_labels <- c("1/2", "1/1.7", "1/1.5", "1/1.3", "1", "1.3", "1.5", "1.7", "2")
y_labels <- c("1/2", "1/1.7", "1/1.5", "1/1.3", "1", "1.3", "1.5", "1.7", "2")


ate_df %>% ggplot(aes(x=as.factor(E0), y= as.factor(E1), fill = ATE)) +
  geom_tile(color = "black") +
  geom_tile(color = "white", fill = NA) +
  geom_tile(color = "white", fill = NA) +
  geom_text(aes(label = sprintf("%.0f", ATE)), color = "black", size = 3.4) +
  scale_fill_gradient2() +
  theme_minimal() +
  theme(panel.grid = element_blank())+
  labs(x = "eta0", y = "eta1") +
  theme(panel.grid = element_blank(),
        axis.text.y = element_text(size = 11),
        axis.title.y = element_text(face = "bold", size = 13),
        axis.title.x = element_text(face = "bold", size = 13)
        ) +
  scale_y_discrete(expand = c(0, 0),
                   limits = rev(levels(factor(ate_df$E1))),
                   labels = rev(y_labels)) +
  scale_x_discrete(expand = c(0, 0),
                   position = 'top',
                   labels = x_labels)
```