Problem Set 7: Matching and Weighting Estimators

GOV 2003

Due at 11:59 pm (ET) on Nov 10, 2021

Instruction

Before you begin, please read the following instructions **carefully**:

- **No late submission is allowed** without prior approval from the instructors.
- **All answers should be typed up**. We recommend the use of `Rmarkdown`. A `Rmarkdown` template for this problem set is provided. Answers to analytical solutions should also be typed up.
- **A PDF copy of your answer** including your computer code should be uploaded to `Gradescope` before the deadline. **Do not submit the markdown file itself**.
- This problem set includes a bonus question for extra credit. No deduction in the total points will be made from this question. Note that the maximum points of this problem set is 15 points. That is, if the student receives 3 points from the bonus question and 14 points from the other questions, the total points will be 15 points.

Introduction

This problem set consists of two parts. In `Part A`, we will analytically investigate the use of propensity score in matching and weighting. In `Part B`, we will use the data from Aizer et al. (2016) to apply the matching and weighting estimators.

Part A

In this part, we will analytically investigate the use of propensity score in matching (Question 1) and weighting (Question 2).

You can write down the answer in hand for this part if you want.

Setup A

Let D_i denote a binary treatment variable, Y_i an outcome variable, $\mathbf{X}_i$ observed covariate variables, and $\pi(\mathbf{X}_i) = \mathbb{P}(D_i = 1 \mid \mathbf{X}_i)$ the true propensity score. Here, we assume SUTVA, no unmeasured confounders, and overlap/positivity assumptions throughout the question.

Question 1: Matching (1 pt for each)

(a) Recall that covariates are balanced conditional on true propensity score: $D_i \perp\!\!\!\perp \mathbf{X}_i \mid \pi(\mathbf{X}_i)$. Prove this by showing that

$$\mathbb{P}(D_i = 1 \mid \mathbf{X}_i, \pi(\mathbf{X}_i)) = \mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i)).$$

Hints:

- Observe that $\mathbb{P}(D_i = 1 \mid \mathbf{X}_i, \pi(\mathbf{X}_i)) = \pi(\mathbf{X}_i)$.
- Show that the right-hand side is also equal to $\pi(\mathbf{X}_i)$.
- Since D_i is binary (= 0 or 1), $\mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i)) = \mathbb{E}[D_i \mid \pi(\mathbf{X}_i)]$ and $\pi(\mathbf{X}_i) = \mathbb{P}(D_i = 1 \mid \mathbf{X}_i) = \mathbb{E}[D_i \mid \mathbf{X}_i]$
- Use the law of iterated expectations, $\mathbb{E}[X \mid Y] = \mathbb{E}[\mathbb{E}[X \mid Y, Z] \mid Y]$

(b) Recall that given the propensity score, we only need to match/balance on such $\pi(\mathbf{x})$: $(Y_i(0), Y_i(1)) \perp\!\!\!\perp D_i \mid \pi(\mathbf{X}_i)$. Prove that this holds under *no unmeasured confounders* and *overlap* assumptions by showing that

$$\mathbb{P}(D_i = 1 \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)) = \mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i)).$$

Hints:

- Note that we've shown that the right-hand side is $\pi(\mathbf{X}_i)$ in (a). Show that the left-hand side is also equal to $\pi(\mathbf{X}_i)$.
- Since D_i is binary (= 0 or 1), $\mathbb{P}(D_i = 1 \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)) = \mathbb{E}[D_i \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)]$
- Use the law of iterated expectations, $\mathbb{E}[X \mid Y] = \mathbb{E}[\mathbb{E}[X \mid Y, Z] \mid Y]$

(c) [Bonus] When would you prefer to use propensity score matching to exact matching? What may be potentially problematic with this approach in practice?

Answer 1

(a)

$$\begin{aligned}
\mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i)) &= \mathbb{E}[D_i \mid \pi(\mathbf{X}_i)] && \text{[by the third hint]} \\
&= \mathbb{E}[\mathbb{E}[D_i \mid \pi(\mathbf{X}_i), \mathbf{X}_i] \mid \pi(\mathbf{X}_i)] && \text{[by the law of iterated expectations]} \\
&= \mathbb{E}[\mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i), \mathbf{X}_i) \mid \pi(\mathbf{X}_i)] && \text{[by the third hint]} \\
&= \mathbb{E}[\pi(\mathbf{X}_i) \mid \pi(\mathbf{X}_i)] && \text{[by the first hint]} \\
&= \pi(\mathbf{X}_i).
\end{aligned}$$

Since the left-hand side is $\mathbb{P}(D_i = 1 \mid \mathbf{X}_i, \pi(\mathbf{X}_i)) = \pi(\mathbf{X}_i)$ (first **hint**), we have

$$\mathbb{P}(D_i = 1 \mid \mathbf{X}_i, \pi(\mathbf{X}_i)) = \mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i)) \quad \square$$

(b)

$$\begin{aligned}
&\mathbb{P}(D_i = 1 \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)) \\
&\quad = \mathbb{E}[D_i \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)] && \text{[by the third hint]} \\
&\quad = \mathbb{E}[\mathbb{E}[D_i \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i), \mathbf{X}_i] \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)] && \text{[by the law of iterated expectations]} \\
&\quad = \mathbb{E}[\mathbb{E}[D_i \mid \pi(\mathbf{X}_i), \mathbf{X}_i] \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)] && \text{[by no unmeasured confounder]} \\
&\quad = \mathbb{E}[\mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i), \mathbf{X}_i) \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)] && \text{[by the third hint]} \\
&\quad = \mathbb{E}[\pi(\mathbf{X}_i) \mid \pi(\mathbf{X}_i)] && \text{[by the result of (a)]} \\
&\quad = \pi(\mathbf{X}_i).
\end{aligned}$$

Since the right-hand side is $\mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i)) = \pi(\mathbf{X}_i)$ (the result of (a)), we have

$$\mathbb{P}(D_i = 1 \mid Y_i(0), Y_i(1), \pi(\mathbf{X}_i)) = \mathbb{P}(D_i = 1 \mid \pi(\mathbf{X}_i)) \quad \square$$

(c) If the pre-treatment covariates ($\mathbf{X}_i$) is high-dimensional or continuous, it may be easier to match on the scalar value of propensity score. However, the *true* propensity score is not known to researchers, and estimating it may leave researchers with more model dependence and discretion. Thus balancing only on the estimated propensity score may not actually balance $\mathbf{X}_i$. See Secition 4.2 of King and Nielsen (2019) for further discussion.

[Bonus] Question 2: Weighting (1 pt)

Now, we move to the Horvitz-Thompson estimator which weights the units by inverse propensity score. Here, we will prove the unbiasedness of the HT estimator for the average treatment effect for the treated (ATT) given the true propensity score: $\mathbb{E}[\widehat{\tau}_{ipw,t}] = \mathbb{E}[Y_i(1) - Y_i(0) \mid D_i = 1]$ where $\widehat{\tau}_{ipw,t} = \frac{1}{n_1}\sum_{i=1}^{n}\left(D_iY_i - \frac{\pi(\mathbf{X}_i)(1-D_i)Y_i}{1-\pi(\mathbf{X}_i)}\right)$. Prove the unbiasedness under *no unmeasured confounders* and *overlap* assumptions by showing that

$$\mathbb{E}\left[\frac{\pi(\mathbf{X}_i)(1-D_i)Y_i}{1-\pi(\mathbf{X}_i)}\right] = \mathbb{E}[Y_i(0) \mid D_i = 1]\mathbb{P}(D_i = 1)$$

Hints:

- Use the fact that $\mathbb{P}(D_i = 1 \mid \mathbf{X}_i) = \frac{\mathbb{P}(\mathbf{X}_i \mid D_i=1)\mathbb{P}(D_i=1)}{\mathbb{P}(X_i)}$.
- For finite and countable Y, $\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]] = \sum_y \mathbb{E}[X \mid Y = y]\mathbb{P}(Y = y)$

Answer 2

$$\begin{aligned}
&\mathbb{E}\left[\frac{\pi(\mathbf{X}_i)(1-D_i)Y_i}{1-\pi(\mathbf{X}_i)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{\pi(\mathbf{X}_i)(1-D_i)Y_i}{1-\pi(\mathbf{X}_i)} \mid \mathbf{X}_i\right]\right] && \text{[by the l.i.e.]} \\
&= \mathbb{E}\left[\mathbb{E}\left[(1-D_i)Y_i \mid \mathbf{X}_i\right]\frac{\pi(\mathbf{X}_i)}{1-\pi(\mathbf{X}_i)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[(1-D_i)Y_i(0) \mid \mathbf{X}_i\right]\frac{\pi(\mathbf{X}_i)}{1-\pi(\mathbf{X}_i)}\right] && \text{[by the consistency]} \\
&= \mathbb{E}\left[\mathbb{E}\left[Y_i(0) \mid \mathbf{X}_i\right]\mathbb{E}\left[1-D_i \mid \mathbf{X}_i\right]\frac{\pi(\mathbf{X}_i)}{1-\pi(\mathbf{X}_i)}\right] && \text{[by the n.u.c.]} \\
&= \mathbb{E}\left[\mathbb{E}\left[Y_i(0) \mid D_i = 1, \mathbf{X}_i\right](1-\pi(\mathbf{X}_i))\frac{\pi(\mathbf{X}_i)}{1-\pi(\mathbf{X}_i)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Y_i(0) \mid D_i = 1, \mathbf{X}_i\right]\pi(\mathbf{X}_i)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Y_i(0) \mid D_i = 1, \mathbf{X}_i\right]\frac{\mathbb{P}(\mathbf{X}_i \mid D_i = 1)\mathbb{P}(D_i = 1)}{\mathbb{P}(\mathbf{X}_i)}\right] && \text{[by the first hint]} \\
&= \sum_{\mathbf{x}} \mathbb{E}\left[Y_i(0) \mid D_i = 1, \mathbf{X}_i = \mathbf{x}\right]\frac{\mathbb{P}(\mathbf{X}_i = \mathbf{x} \mid D_i = 1)\mathbb{P}(D_i = 1)}{\mathbb{P}(\mathbf{X}_i = \mathbf{x})}\mathbb{P}(\mathbf{X}_i = \mathbf{x}) && \text{[by the second hint]} \\
&= \sum_{\mathbf{x}} \mathbb{E}\left[Y_i(0) \mid D_i = 1, \mathbf{X}_i = \mathbf{x}\right]\mathbb{P}(\mathbf{X}_i = \mathbf{x} \mid D_i = 1)\mathbb{P}(D_i = 1) \\
&= \mathbb{E}\left[\mathbb{E}\left[Y_i(0) \mid D_i = 1, \mathbf{X}_i\right] \mid D_i = 1\right]\mathbb{P}(D_i = 1) && \text{[by the second hint]} \\
&= \mathbb{E}\left[Y_i(0) \mid D_i = 1\right]\mathbb{P}(D_i = 1) && \text{[by the l.i.e.]} \quad \square
\end{aligned}$$

Part B

In this part, we will use the data from Aizer et al. (2016) to apply the matching and weighting estimators.

Setup B

Aizer et al. (2016) examine whether cash transfers to poor families improves children's longevity. The authors analyze individual-level data from the first government welfare program in American history: the Mother's Pension Program, which gave cash to impoverished mothers with no male partner in their household. Using the original applications for the program and Census records, the authors evaluate the impact of the program on their life and health outcomes as men. (Typically, men are much easier to track through administrative records because they rarely change their last names.) As a control group, the authors examine boys whose mothers applied for the Mother's Pension but whose applications were rejected. Thus, all observations in the sample had mothers who were eligible to apply; after applying, some were accepted, and others were rejected. We will be examining the impact of receiving this cash transfer as children on the participants' lifespan.

The variables in a simplified version of the dataset (`pensions.RData`) that we analyze in this part are described below:

Variable	Description
Treatment variable	
`accepted`	A binary treatment indicating whether the subject's mother was accepted into the program and given money (D_i)
Pre-treatment covariates	
`divorced`	Whether subject's mother was divorced when she applied
`childageyears`	How old the subject was when his mother applied
`length_name`	Length of family name (may provide information about family background)
`numkids`	Number of children in family when mother applied
`famearn`	Estimated family income at time of application
`famearn_2`	Estimated family income^2
`maxage`	Age of oldest child in family when mother applied
`minage`	Age of youngest child in family when mother applied
`born.1901`	Applicant born 1901-1910
`born.1911`	Applicant born 1911-1920
`born.1921`	Applicant born post-1920
Outcome variable	
`ageatdeath`	Age at death (Y_i)

Question 3: Matching (1 pt for (a) and (d); 4 pts for (b) and (c))

In this question, we will apply the matching methods to estimate the ATT.

(a) Check the balance of the treated and control group using the standardized mean differences between the two groups. Which variables are particularly imbalanced?

Hint: You may use `cobalt::bal.tab()` function.

(b) Conduct one-to-one nearest neighbor matching (with replacement) based on Mahalanobis distance. Show the index of the matched row for the 1, 3, 4, 5, and 6-th rows (which are all treated) and the total number of matched control units. Assess the balance of covariates after matching in terms of the standardized mean differences. Finally, estimate the ATT and its standard error. Briefly discuss the results in words.

Hint: You may use `Matching::Match()` function to estimate the standard error based on Abadie and Imbens (2006) estimator.

(c) Estimate the propensity score using the logistic regression with all pre-treatment covariates as linear predictors (you may exclude `born.1901` to avoid collinearity). How balanced does the propensity score look? Conduct one-to-one nearest neighbor propensity score matching (with replacement) for the estimated propensity scores. Show the index of the matched row for the 1, 3, 4, 5, and 6-th rows (which are all treated) and the total number of matched control units. How balanced does the propensity score look after matching? Make sure to assess the balance of covariates after matching in terms of the standardized mean differences. Finally, estimate the ATT and its standard error. Briefly discuss the results in words.

Hints:

- You may use `MatchIt::matchit()` function. Check `your-matchitobject$match.matrix` for the matched rows and `your-matchitobject$nn` for the number of matched units (use `?matchit` to see more details).
- Or you may use `Matching::Match()` function. Check `your-Matchobject$index.control` for the matched rows and use `length(unique(your-Matchobject$index.control))` for the number of matched units (use `?Match` to see more details).

(d) [Bonus] Choose an alternative way (either statistical model or machine learning method) to estimate the propensity score. Repeat (c) using this propensity score. You may also use alternative matching method (e.g., using optimal matching or increasing the number of matches). Did you have to make any trade-offs, either in terms of sample size or between the balances of the various variables?

Answer 3

(a)

```r
load("pensions.RData")
library(cobalt)
```

```
##  cobalt (Version 4.2.4, Build Date: 2020-11-05 17:30:21 UTC)
```

```r
bal.tab(x = pensions,
        treat = pensions$accepted)
```

```
## Note: 's.d.denom' not specified; assuming pooled.
```

```
## Balance Measures
##                  Type Diff.Un
## ageatdeath    Contin.  0.0869
## divorced       Binary  0.0012
## childageyears Contin. -0.1246
## length_name   Contin.  0.0095
## numkids       Contin.  0.1048
## famearn       Contin. -0.1100
## famearn_2     Contin. -0.0721
## maxage        Contin. -0.0925
## minage        Contin. -0.2046
## born.1901      Binary -0.1015
## born.1911      Binary  0.0902
## born.1921      Binary  0.0113
##
## Sample sizes
##     Control Treated
## All     983    6876
```

(b)

```r
# Conduct one-to-one nearest neighbor mahalanobis distance matching
## Matching package
library(Matching)
```

```
## Loading required package: MASS
```

```
## ##
## ##  Matching (Version 4.9-7, Build Date: 2020-02-05)
## ##  See http://sekhon.berkeley.edu/matching for additional documentation.
## ##  Please cite software as:
## ##   Jasjeet S. Sekhon. 2011. ``Multivariate and Propensity Score Matching
## ##   Software with Automated Balance Optimization: The Matching package for R.''
## ##   Journal of Statistical Software, 42(7): 1-52.
## ##
```

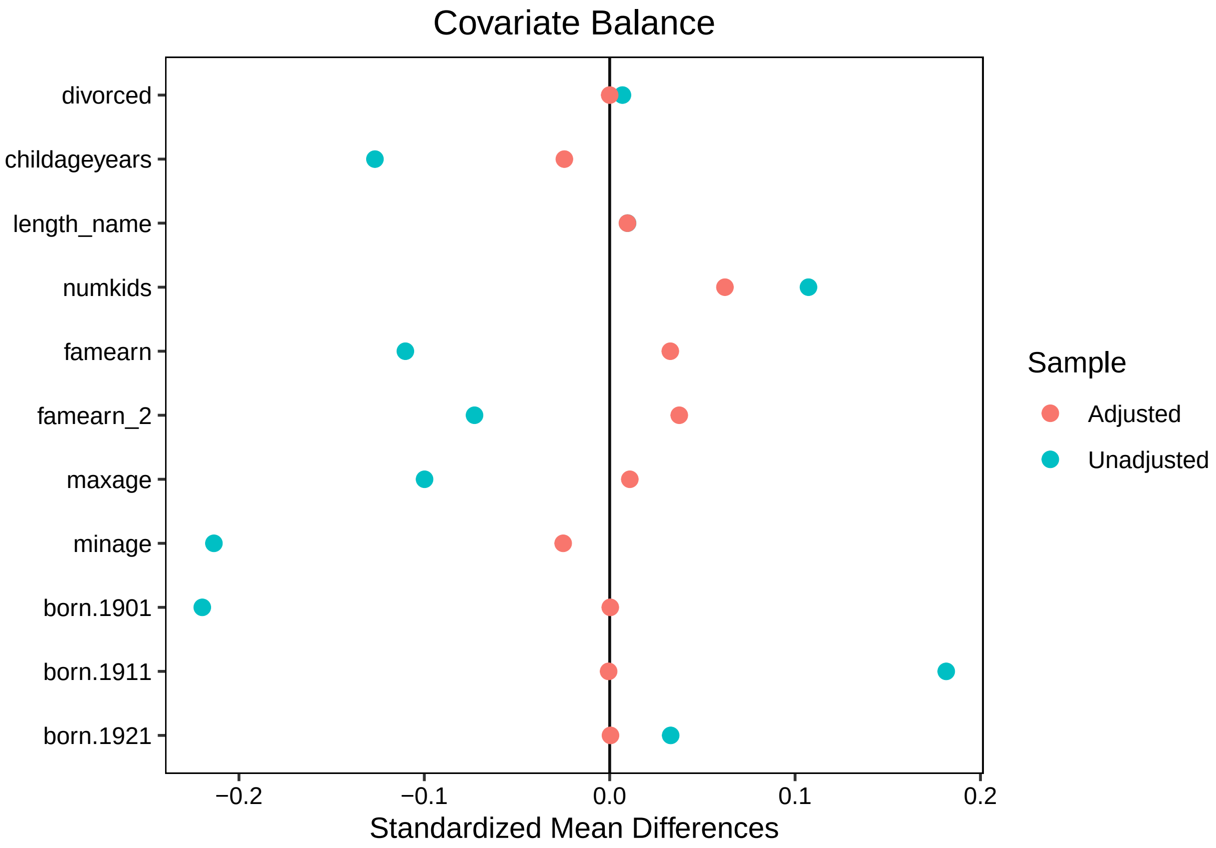
```r
match_md  <- Match(Y=pensions$ageatdeath,
                   Tr=pensions$accepted,
                   X=pensions[,c("divorced", "childageyears", "length_name",
                                 "numkids", "famearn", "famearn_2",
                                 "maxage", "minage",
                                 "born.1901", "born.1911", "born.1921")],
                   M=1, replace = TRUE, ties = FALSE,
                   Weight = 2)
match_md$index.control[1:6]
```

```
## [1] 6661  791 6331 2880  926 6306
```

```r
# Note that matching result may be different from that of MachIt because of ties.
length(unique(match_md$index.control))
```

```
## [1] 936
```

```r
# Balance
love.plot(accepted ~ divorced + childageyears +
            length_name + numkids + famearn + famearn_2 +
            maxage + minage + born.1901 + born.1911 + born.1921,
          data = pensions,
          stats = "mean.diffs",
          weights = data.frame(Matched = get.w(match_md)),
          method = c("matching"), binary = "std")
```



```r
# ATT
summary(match_md)
```

```
##
## Estimate...  1.934
## SE.........  0.19971
## T-stat.....  9.6838
## p.val......  < 2.22e-16
##
## Original number of observations..............  7859
## Original number of treated obs...............  6876
## Matched number of observations...............  6876
## Matched number of observations  (unweighted).  6876
```

(c)

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

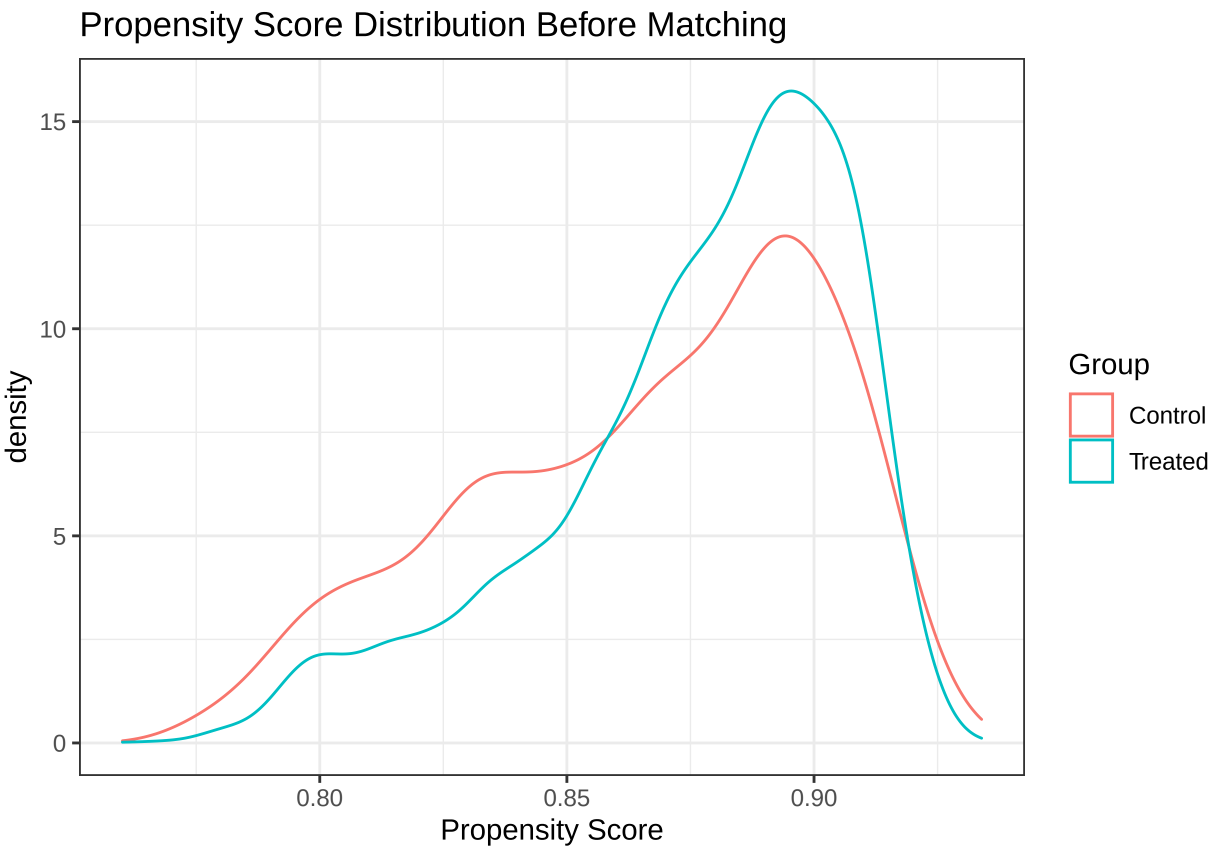
```r
library(MatchIt)
```

```
##
## Attaching package: 'MatchIt'

## The following object is masked from 'package:cobalt':
##
##     lalonde
```

```r
# Estimate propensity score using logistic regression
# Propensity score
ps_logistic <- glm(accepted ~ divorced + childageyears +
      length_name + numkids + famearn + famearn_2 +
      maxage + minage + born.1911 + born.1921,
    family = binomial(logit),
    data = pensions)

pensions %>%
  mutate(pscore = predict(ps_logistic, type = "response")) %>%
  mutate(Group = ifelse(accepted == 1, 'Treated', 'Control')) %>%
  ggplot(aes(x = pscore)) +
geom_density(aes(color = Group)) +
ggtitle("Propensity Score Distribution Before Matching") + labs(x = "Propensity Score") +
theme_bw()
```



```
# Conduct one-to-one nearest neighbor propensity score matching
## MatchIt package
match_logistic <- matchit(accepted ~ divorced + childageyears +
                            length_name + numkids + famearn + famearn_2 +
                            maxage + minage + born.1911 + born.1921,
                          data = pensions,
                          method = "nearest",
                          distance = "logit",
                          replace = T)
row.names(match_logistic$match.matrix)[1:6]
```

```
## [1] "1" "3" "4" "5" "6" "7"
```

```
match_logistic$match.matrix[1:6]
```

```
## [1] "2849" "5588" "7843" "33"   "3640" "6535"
```

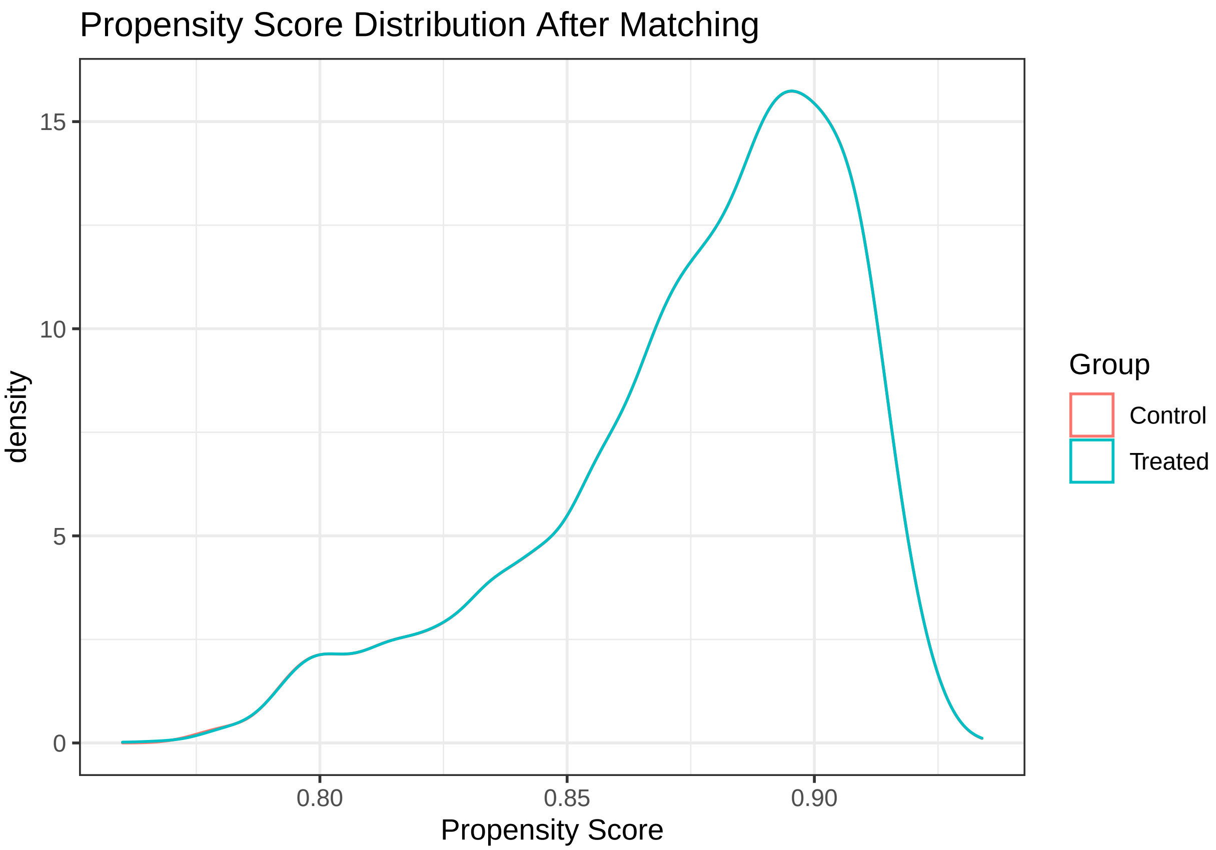
```
match_logistic$nn
```

```
##           Control Treated
## All           983    6876
## Matched       927    6876
## Unmatched      56       0
## Discarded       0       0
```

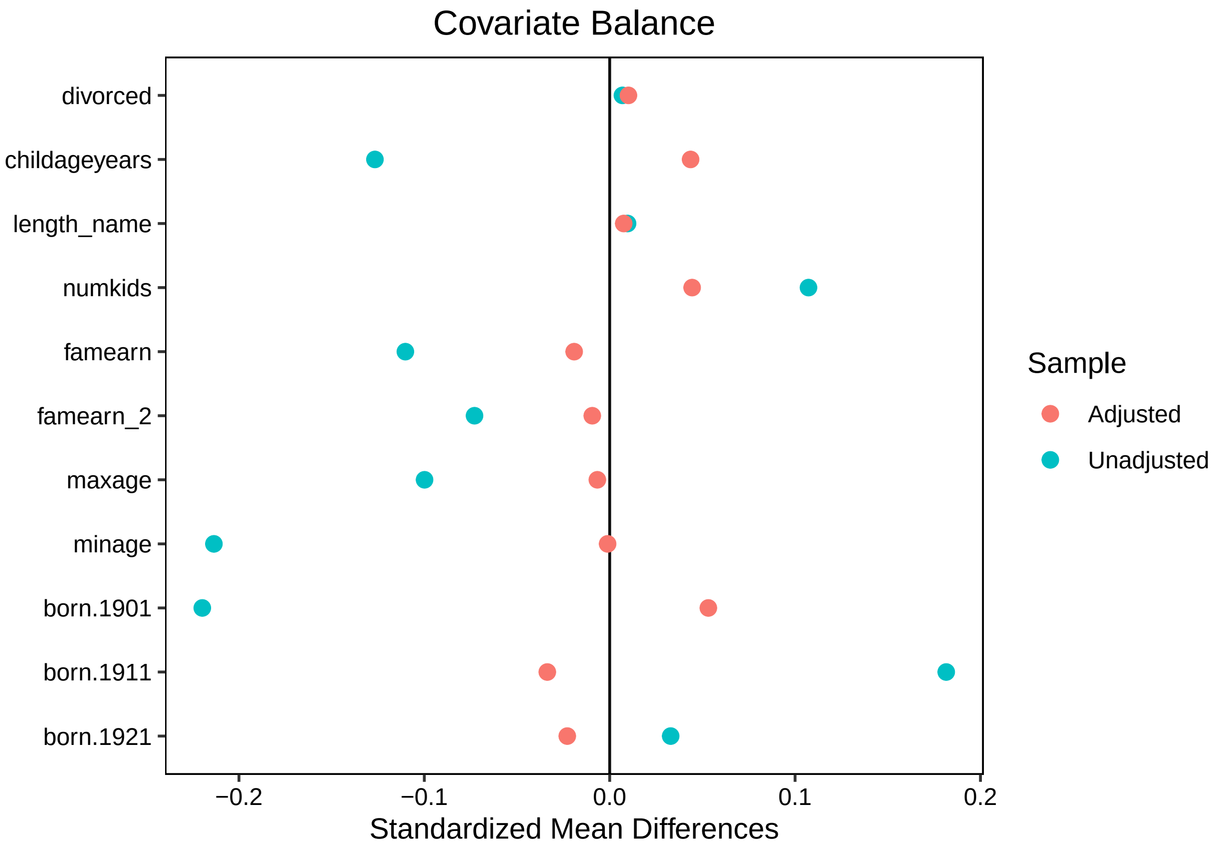
```r
match_logistic$model
```

```
##
## Call:  glm(formula = formula, family = binomial(logit), data = data)
##
## Coefficients:
##   (Intercept)       divorced  childageyears    length_name        numkids
##     1.820e+00      4.956e-02      1.882e-02      7.626e-03      4.214e-02
##       famearn      famearn_2         maxage         minage      born.1911
##    -3.359e-04      1.894e-07     -1.851e-02     -3.500e-02      4.118e-01
##     born.1921
##     2.658e-01
##
## Degrees of Freedom: 7858 Total (i.e. Null);  7848 Residual
## Null Deviance:       5925
## Residual Deviance: 5855  AIC: 5877
```

```r
# Balance
# Propensity score
pensions_ps <- pensions %>%
  mutate(pscore = predict(ps_logistic, type = "response"))
rbind(pensions_ps[pensions_ps$accepted == 1,] %>%
  mutate(Group = "Treated"),
  pensions_ps[match_logistic$match.matrix,] %>%
  mutate(Group = "Control")) %>%
  ggplot(aes(x = pscore)) +
geom_density(aes(color = Group)) +
ggtitle("Propensity Score Distribution After Matching") + labs(x = "Propensity Score") +
theme_bw()
```



```
# Covariates
love.plot(accepted ~ divorced + childageyears +
                            length_name + numkids + famearn + famearn_2 +
                            maxage + minage + born.1901 + born.1911 + born.1921,
  data = pensions,
  stats = "mean.diffs",
  weights = data.frame(Matched = get.w(match_logistic)),
  method = c("matching"), binary = "std")
```



```
# ATT
## Matching package
library(Matching)
match_logistic2  <- Match(Y=pensions$ageatdeath, Tr=pensions$accepted,
                         X=ps_logistic$fitted, M=1, replace = TRUE, ties = FALSE)
match_logistic2$index.control[1:6]
```

```
## [1] 2849 5588 4653   33 3826 6535
```

```
# Note that matching result may be different from that of MachIt because of ties.
length(unique(match_logistic2$index.control))
```

```
## [1] 951
```

```
summary(match_logistic2)
```

```
##
## Estimate...  1.5893
## SE.........  0.19669
## T-stat.....  8.0801
## p.val......  6.6613e-16
##
## Original number of observations..............  7859
## Original number of treated obs...............  6876
## Matched number of observations...............  6876
## Matched number of observations  (unweighted).  6876
```

(d)

Answers will depend.

Question 4: Weighting (4 pts for (a); 1 pt for (b))

In this question, we will apply the weighting methods to estimate the ATT.

(a) Use your estimated propensity score from Q3 (c) and compute an appropriate weight for each of the control units such that their pre-treatment covariate distribution becomes closer to that of the treated group. Be sure to write down the formal expression of the weights you are using. Compute the standardized mean difference of each covariate with and without weights and evaluate the effectiveness of this weighting strategy. Finally, using these weights, estimate the ATT and its standard error using the inverse probability weighting estimator. Report the point estimate and the standard error. Briefly discuss the results in words.

- **Hint**: Use boostrap to estimate the standard error.

(b) [Bonus] Write down a mathematical expression of the augmented inverse probability weighting estimator for the ATT. Using the weights from Q3 (c), estimate the ATT and its standard error with this estimator. Briefly discuss the results.

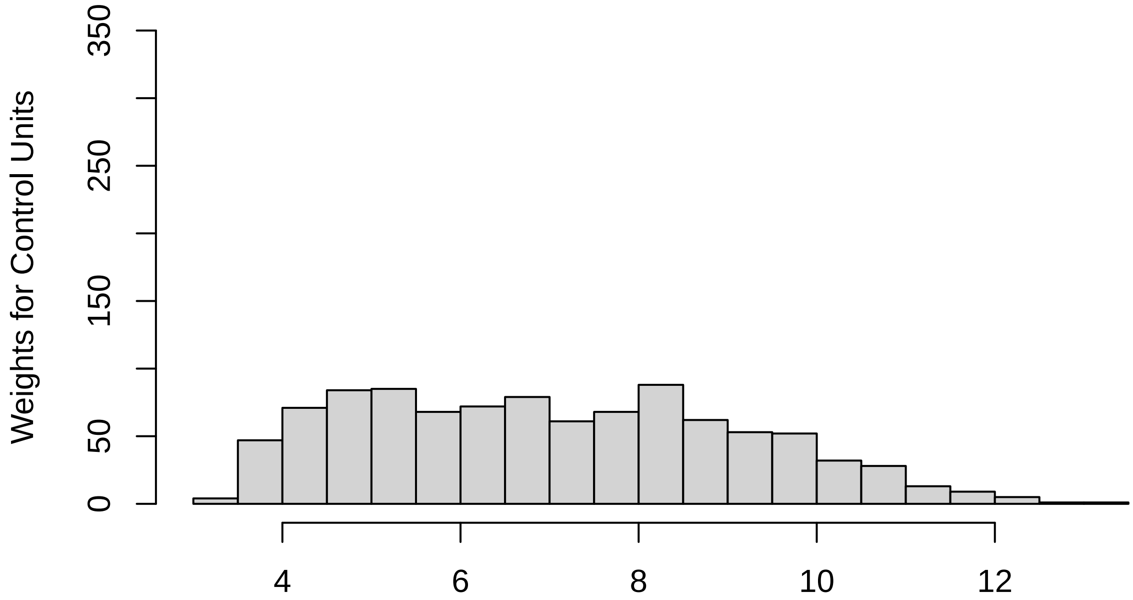
Answer 4

(a)

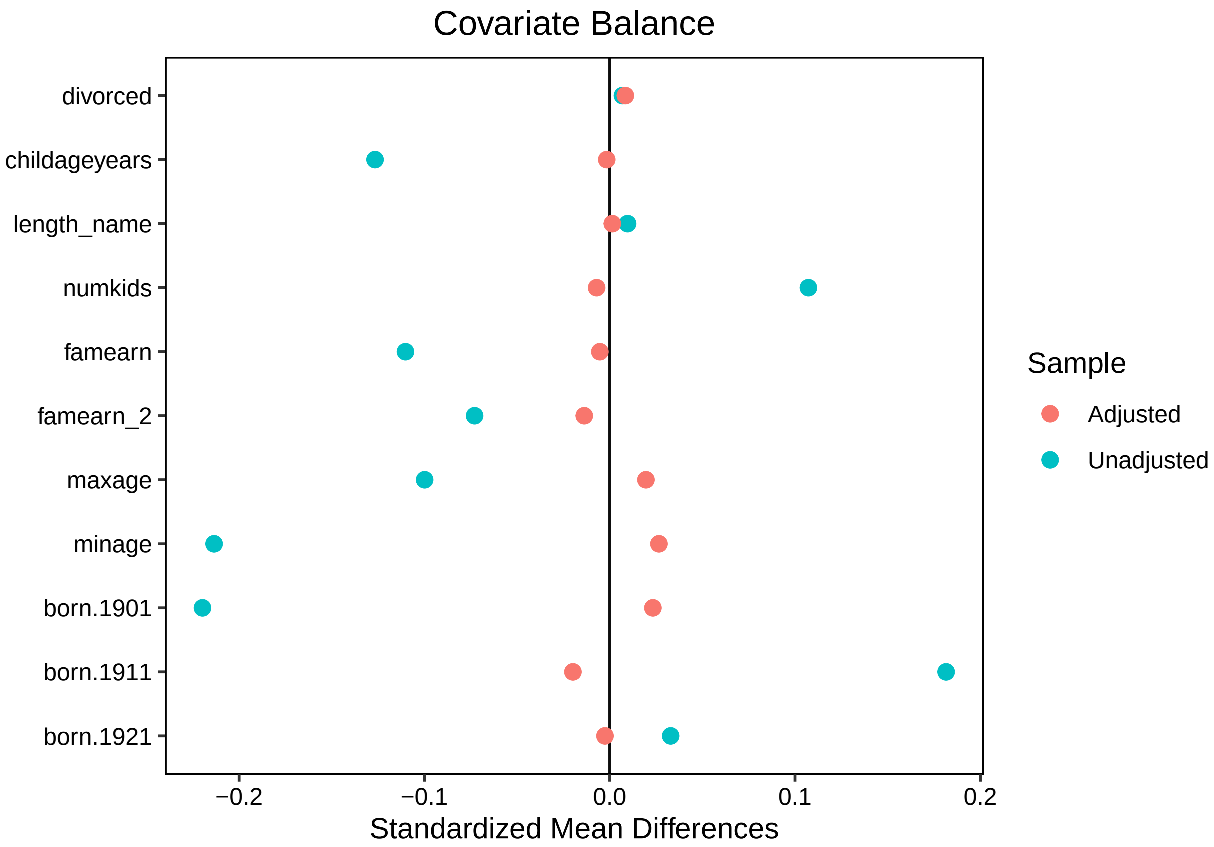
Horvitz-Thompson estimator for treatment effects:

$$\widehat{\text{ATT}} = \widehat{\tau}_{ipw,t} = \frac{1}{n_1}\sum_{i=1}^{n}\left(D_iY_i - \frac{\widehat{\pi}(\mathbf{X}_i)(1-D_i)Y_i}{1-\widehat{\pi}(\mathbf{X}_i)}\right)$$

```r
pi.hat = ps_logistic$fitted.values[pensions$accepted==0]
weights = pi.hat/(1-pi.hat)
# Check extreme weights (none in this case)
hist(weights, main = "", breaks = 15,
    ylab = "Weights for Control Units", xlab = "", ylim = c(0, 350))
```



```
# append estimated weights
weights.full = rep(1, nrow(pensions))
weights.full[pensions$accepted==0] = weights
pensions.w <- pensions %>% mutate(weights = weights.full)
# Balance
love.plot(accepted ~ divorced + childageyears +
                            length_name + numkids + famearn + famearn_2 +
                            maxage + minage + born.1901 + born.1911 + born.1921,
  data = pensions,
  stats = "mean.diffs",
  weights = data.frame(Weighted = pensions.w$weights),
  method = c("matching"), binary = "std")
```



```r
## ATT
# function for att
att_ipw <- function(dat, weights) {
  # weights   <- dat %>% filter(accepted == 0) %>% pull(weights)
  reweights <- weights / sum(weights)
  Y1 <- dat %>% filter(accepted == 1) %>% pull(ageatdeath)
  Y0 <- dat %>% filter(accepted == 0) %>% pull(ageatdeath)
  att_ht  <- (sum(Y1) - sum(Y0 * weights)) / sum(dat$accepted)
  att_hjk <- mean(Y1) - sum(reweights * Y0)
  return(c(att_ht, att_hjk))
}
# point estimate
att_ipw(pensions, weights = weights)
```

```
## [1] 0.9180936 1.2868142
```

```r
# standard error
set.seed(02138); sims<-1000; tau_hat_draws<-rep(NA, sims)
for (i in 1:sims) { # Repeat the following several times
  # 1. Randomly resample n rows of the data with replacement
  sample_boot <- dplyr::slice_sample(pensions, n = nrow(pensions),
                                     replace = TRUE)
  # 2. Estimate ps using the bootstrapped data
  ps_logistic_boot <- glm(accepted ~ divorced + childageyears +
      length_name + numkids + famearn + famearn_2 +
      maxage + minage + born.1911 + born.1921,
    family = binomial(logit),
    data = sample_boot)
  pi.hat_boot = ps_logistic_boot$fitted.values[sample_boot$accepted==0]
  weights_boot = pi.hat_boot/(1-pi.hat_boot)
  # 3. Calculate tau_hat in each bootstrap
  tau_hat_draws[i]  <- att_ipw(sample_boot, weights = weights_boot)[1]
}
# 4. Use empirical variance of the bootstraps
sd(tau_hat_draws)
```

```
## [1] 0.4559426
```

(b)

Observe that $\frac{1}{n_1}\sum_{i=1}^{n} D_i Y_i$ is a consistent estimator for $\mathbb{E}[Y_i(1) \mid D_i = 1]$. Meanwhile, consider the following estimator $\hat{\nu}_{0,\text{aipw}}$ for $\mathbb{E}[Y_i(0) \mid D_i = 1]$

$$
\begin{aligned}
\hat{\nu}_{0,\text{aipw}} &= \frac{1}{n_1}\sum_{i=1}^{n}\left\{\frac{\widehat{\pi}(\mathbf{X}_i)(1-D_i)}{1-\widehat{\pi}(\mathbf{X}_i)}Y_i + \frac{D_i - \widehat{\pi}(\mathbf{X}_i)}{1-\widehat{\pi}(\mathbf{X}_i)}\widehat{\mu}_0(\mathbf{X}_i)\right\} \\
&= \frac{1}{n_1}\sum_{i=1}^{n}\left\{D_i\widehat{\mu}_0(\mathbf{X}_i) + \frac{\widehat{\pi}(\mathbf{X}_i)(1-D_i)(Y_i - \widehat{\mu}_0(\mathbf{X}_i))}{1-\widehat{\pi}(\mathbf{X}_i)}\right\}
\end{aligned}
$$

Observe that $\hat{\nu}_{0,\text{aipw}}$ is consistent if either the propensity score model or the outcome model is correct, thus doubly robust.

Hence, a doubly robust **Augmented** IPW estimator which combines regression and weighting for ATT is:

$$
\widehat{\tau}_{\text{aipw,t}} = \frac{1}{n_1}\sum_{i=1}^{n}\left[D_iY_i - \left\{\frac{\widehat{\pi}(\mathbf{X}_i)(1-D_i)}{1-\widehat{\pi}(\mathbf{X}_i)}Y_i + \frac{D_i - \widehat{\pi}(\mathbf{X}_i)}{1-\widehat{\pi}(\mathbf{X}_i)}\widehat{\mu}_0(\mathbf{X}_i)\right\}\right]
$$

Using a simple linear regression for $\widehat{\mu}_0(\mathbf{X}_i)$:

```r
## ATT
# function for att
att_aipw <- function(dat, pi_hat, mu0_hat) {
  D <- dat %>% pull(accepted)
  Y <- dat %>% pull(ageatdeath)
  att_aipw  <- (sum(D*Y) - sum((pi_hat*(1-D)*Y+ (D-pi_hat)*mu0_hat)/(1-pi_hat))) / sum(dat$acce
  return(att_aipw)
}
# point estimate
mu0 <- lm(ageatdeath ~ divorced + childageyears +
      length_name + numkids + famearn + famearn_2 +
      maxage + minage + born.1911 + born.1921, subset = accepted == 0, data = pensions)
att_aipw(pensions, predict(ps_logistic, newdata = pensions), predict(mu0, newdata = pensions))
# standard error
set.seed(02138); sims<-500; tau_hat_draws<-rep(NA, sims)
```

```r
for (i in 1:sims) { # Repeat the following several times
  # 1. Randomly resample n rows of the data with replacement
  sample_boot <- dplyr::slice_sample(pensions, n = nrow(pensions),
                                     replace = TRUE)
  # 2. Estimate ps using the bootstrapped data
  ps_logistic_boot <- glm(accepted ~ divorced + childageyears +
      length_name + numkids + famearn + famearn_2 +
      maxage + minage + born.1911 + born.1921,
    family = binomial(logit),
    data = sample_boot)
  pi.hat_boot = ps_logistic_boot$fitted.values
  # 2. Estimate mu_0
  mu0_boot <- lm(ageatdeath ~ divorced + childageyears +
      length_name + numkids + famearn + famearn_2 +
      maxage + minage + born.1911 + born.1921, subset = accepted == 0, data = pensions)
  mu0.hat_boot = predict(mu0_boot, newdata = pensions)
  # 3. Calculate tau_hat in each bootstrap
  tau_hat_draws[i]  <- att_aipw(sample_boot, pi.hat_boot, mu0.hat_boot)[1]
}
# 4. Use empirical variance of the bootstraps
sd(tau_hat_draws)
```

References

Aizer, A., Eli, S., Ferrie, J., and Lleras-Muney, A. (2016). The long-run impact of cash transfers to poor families. *American Economic Review*, 106(4):935–71.