Causal Inference Homework Assignment 4

The role of propensity scores in observational study

Alan Z Chen, Chansoo Song

October 2018

```r
library(arm)
library(ggplot2)
library(rlang)
library(RColorBrewer)
library(stats)
library(Hmisc)
library(caret)
library(randomForest)
library(nnet)
library(dplyr)
library(MatchIt)
library(reshape2)
library(knitr)
library(kableExtra)

# Set global ggplot theme
theme_set(theme_minimal())
# Define global color palettes
col.RdBl.2 <- brewer.pal(3, "RdBu")[-2]
```

Objective

This assignment will give you the opportunity to practice several different propensity score approaches to causal inference. In addition you will be asked to interpret the resulting output and discuss the assumptions necessary for causal inference.

Problem Statement

In this assignment will use data from a constructed observational study. The data and an associated data dictionary are available in this folder. The treatment group for the study that the data are drawn from is the group of children who participated in the IHDP intervention discussed in class. The research question of interest focuses on the effect of the IHDP intervention on age 3 IQ scores for the children that participated in it. The data for the comparison sample of children was pulled from the National Longitudinal Study of Youth during a similar period of time that the data were collected for the IHDP study.

Question 1: Load the data and choose confounders (Step 1)

ANSWER:

```r
load("data/hw4.rdata")

# Pick confounders based on correlation to treatment and
# outcome variables
corr_mat <- cor(hw4)
```

```r
corr_mat <- rbind(corr_mat["treat", ], corr_mat["ppvtr.36", ])
confounders <- names(which((abs(corr_mat[1, ]) > 0.03) * (abs(corr_mat[2,
    ]) > 0.02) > 0))
confounders <- confounders[!(confounders %in% c("treat", "ppvtr.36"))]

# Remove post-treatment variables
confounders <- confounders[!(confounders %in% c("dayskidh", "income"))]

# Don't need Black, Hispanic, AND White
confounders <- confounders[!(confounders %in% c("white"))]

# Subset data for analysis: (a) Reduce data frame to include
# only observations for children whose b.w. < 3000 g (b)
# Include outcome, treatment indicator, and selected
# covariates
dt <- hw4[hw4$bw < 3000, c("ppvtr.36", "treat", confounders)]
```

```r
# New data frame with outcome in 1st column, treatment
# indicator in 2nd column and covariates in the remaining
# columns
head(dt)
```

```
##   ppvtr.36 treat momage b.marr prenatal booze first   bw preterm black
## 1      111     1     33      1        1     0     0 1559      10     0
## 3       81     1     22      0        1     1     0 2240       3     1
## 4       92     1     13      0        1     0     1 1900       6     1
## 5      103     1     25      1        1     0     1 1550       8     1
## 7       81     1     19      0        1     0     0 2270       5     1
## 8       94     1     19      0        1     1     1 1550       4     1
##   hispanic lths hs college st9 st25 st42
## 1        0    0  0       1   0    0    0
## 3        0    1  0       0   0    0    0
## 4        0    1  0       0   0    0    0
## 5        0    0  0       1   0    0    0
## 7        0    1  0       0   0    0    0
## 8        0    0  1       0   0    0    0
```

```r
# List of the variable names for the confounder variables
# chosen
colnames(dt)[3:length(dt)]
```

```
##  [1] "momage"   "b.marr"   "prenatal" "booze"    "first"    "bw"
##  [7] "preterm"  "black"    "hispanic" "lths"     "hs"       "college"
## [13] "st9"      "st25"     "st42"
```

As confounders, we selected covariates that have correlation > 0.03 with treatment and correlation > 0.02 with outcome. We also used Chapter 18 of Gelman and Hill as reference (p. 358 "a set of confounding covariates wthat we think predict both program participation and subsequent test scores"). Our list of confounders is mostly similar. Compared to the text, our list excludes "days in hospital", "sex", "child's age", "some college", "work during pregnancy", "white" and includes "booze" and state indicators.

We excluded post-treatment variables: "dayskidh" and "income" (We determined these are post-treatment because subjects were recruited to the intervention at the time of birth. These two covariates are measured after birth).

Selected confounders:

Child: - child's birth weight (bw) - number of weeks preterm child was born (preterm) - indicator for whether child was born male or female (first)

Mother: - mom age at time of birth(momage) - indicator for whether mom was married at birth (b.marr) - indicator for whether mom received prenatal care (prenatal) - indicator for wehther mom drank alcohol while pregnant (booze) - indicators for child's race/ethnicity (black, hispanic) - indicators for mother's education at time of birth (lths, hs, college) - indicator for state where household resides (st9, st25, st42)

Question 2: Estimate the propensity score (Step 2)

ANSWER:

```
ps.m1 <- glm(treat ~ ., data = dt[, c("treat", confounders)],
    family = binomial(link = "logit"))
dt$psc <- ps.m1$fitted.values
```

Question 3: Restructure your data through matching. [Or at least create the weights variable that will let you to do so in the following steps] (Step 3)

(a) The first thing you need to be clear on before restructuring your data is the estimand. Given the description above about the research question, what is the estimand of interest?

ANSWER:
The research question of interest focuses on the effect of the intervention on the children that participated in it. Since we are interested on the effect of those who received treatment, the estimand of interest is the ATT (average treatment effect for the treated).

(b) First please perform *one-to-one nearest neighbor matching with replacement* using your estimated propensity score from Question 2. Perform this matching using the `matching` command in the arm package. The *"cnts"* variable in the output reflects the number of times each control observation was used as a match (the length is equal to the number of control observations). Use the output of this function to create a weight variable that

- 1) equals one for treated observations and

- 2) equals the number of times used as a match for non- treated observations.

```
matches <- matching(z = dt$treat, score = dt$psc, replace = TRUE)
dt[dt$treat == 0, "wt"] <- matches$cnts
dt[dt$treat == 1, "wt"] <- 1
```

Question 4: Check overlap and balance. (Step 4)

(a) Examining Overlap. Check overlap on the *unmatched* data using some diagnostic plots. Check overlap for the propensity scores as well as two other covariates.

ANSWER:

```
# Paired, inverted histograms: propensity scores
examine_overlap = function(dt, nbins = 40, var = "psc", var_name = "Propensity Score",
    xlab = "Propensity Score", ylim = c(-30, 50), options = NULL) {
```

```
    dt_plot = dt[, c("treat", var)]
    colnames(dt_plot) = c("treat", "X")

    ggplot(dt_plot) + geom_histogram(data = dt_plot[dt_plot$treat ==
        0, ], bins = nbins, fill = "grey", alpha = 0.1, aes(X,
        y = ..count.., color = col.RdBl.2[2])) + geom_histogram(data = dt_plot[dt_plot$treat ==
        1, ], bins = nbins, fill = "grey", alpha = 0.1, aes(X,
        y = -..count.., color = col.RdBl.2[1])) + coord_cartesian(ylim = ylim) +
        scale_color_manual(values = rev(col.RdBl.2), labels = c("Control",
            "Treated"), name = "") + labs(x = xlab, y = "Frequency",
        title = paste("Overlap of ", var_name, " between Groups",
            sep = "")) + options
}

# Overlap of propensity score
examine_overlap(dt, nbins = 40, var = "psc", var_name = "Propensity Score",
    xlab = "Propensity Score", ylim = c(-10, 10))
```
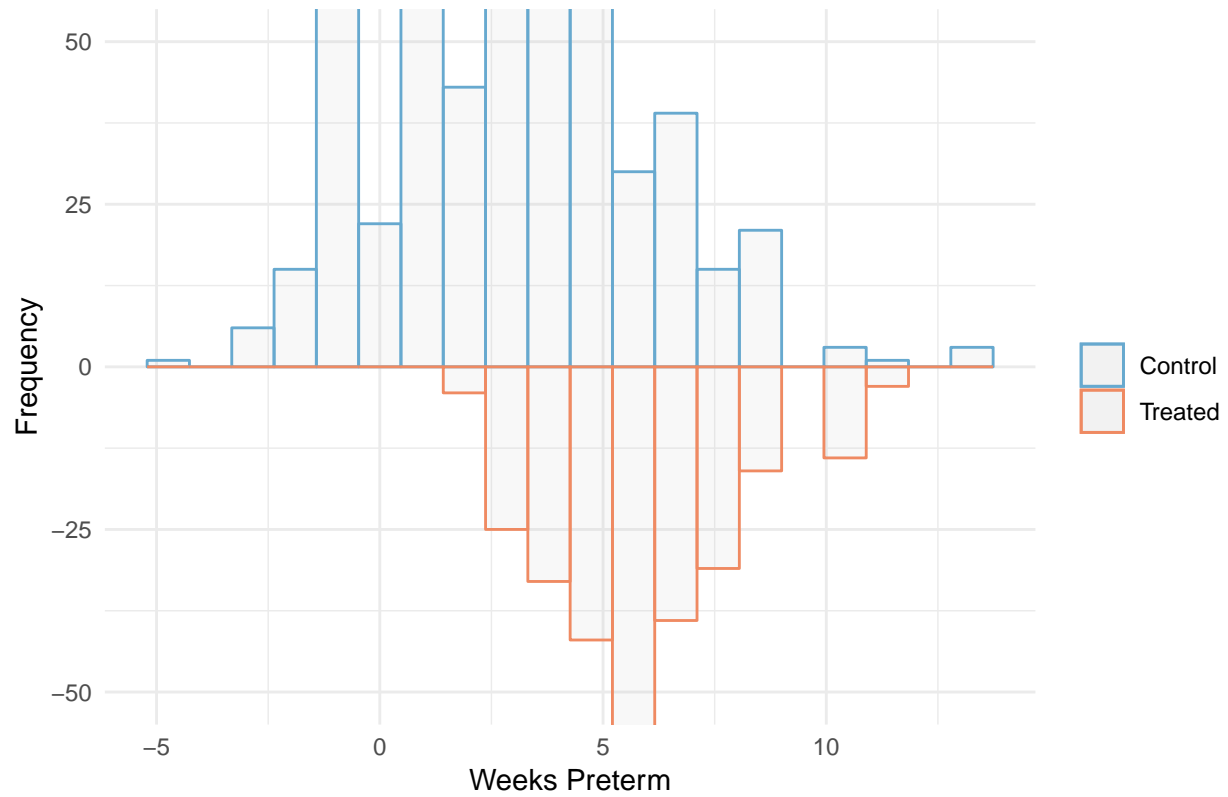

Overlap of Propensity Score between Groups
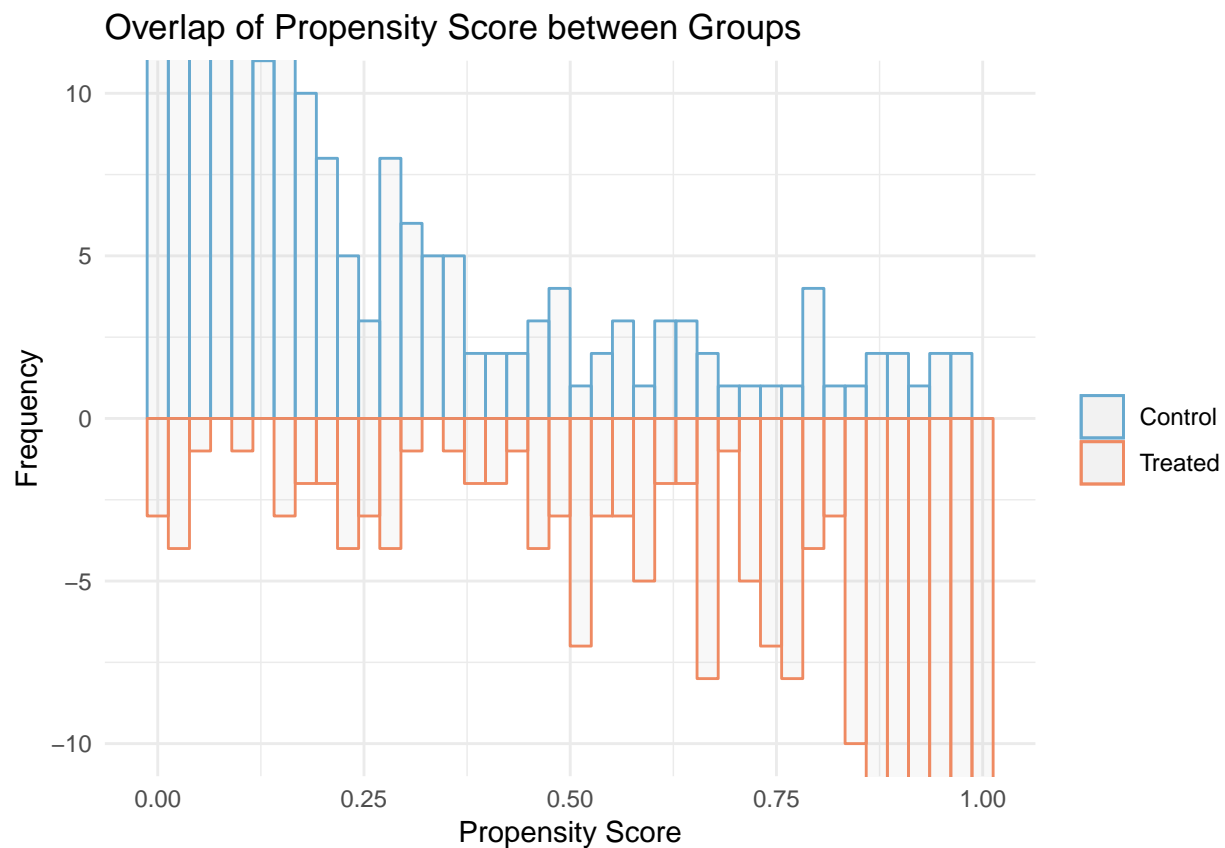
```
# Overlap of preterm
examine_overlap(dt, nbins = 20, var = "preterm", var_name = "Number Weeks Baby Born Preterm",
    xlab = "Weeks Preterm", ylim = c(-50, 50))
```

Overlap of Number Weeks Baby Born Preterm between Groups

```r
# Overlap of birth weight
examine_overlap(dt, nbins = 20, var = "bw", var_name = "Birth weight",
    xlab = "Birth weight", ylim = c(-50, 200))
```

Overlap of Birth weight between Groups
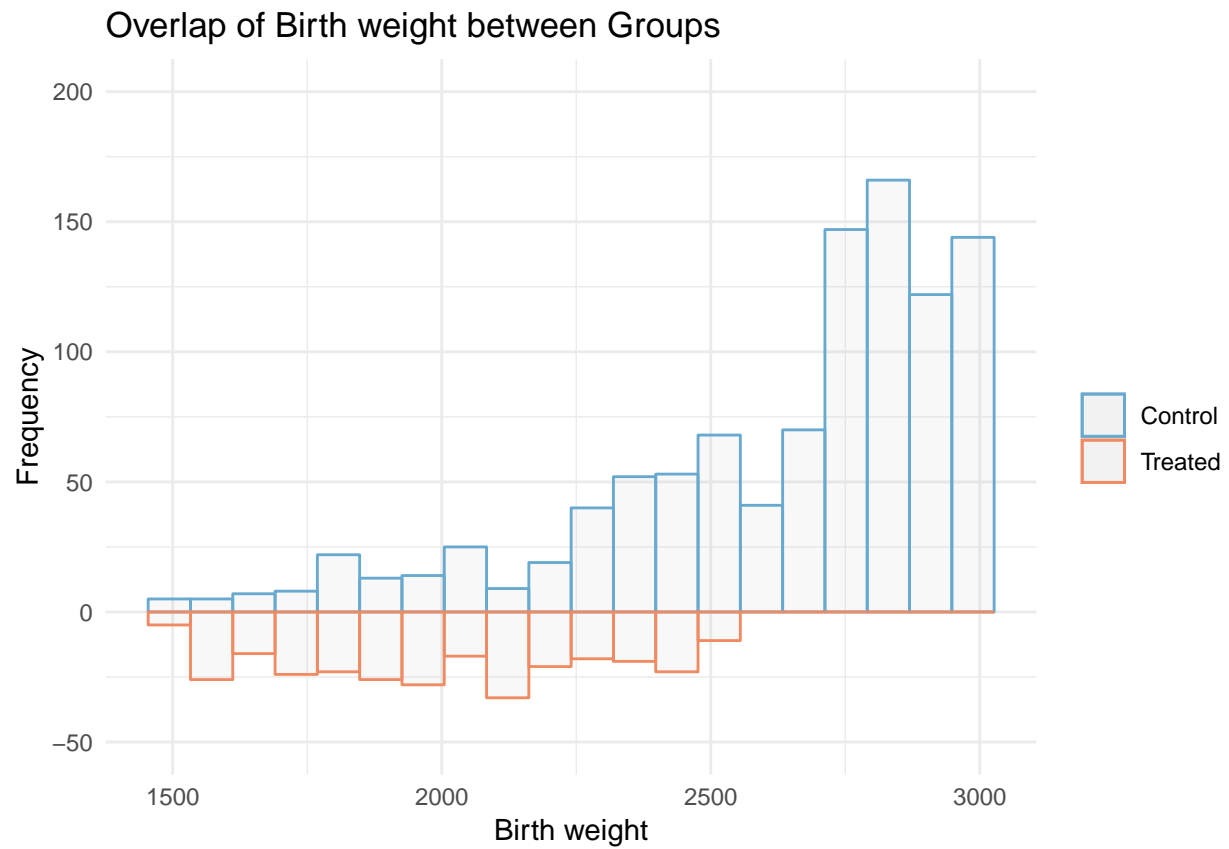

```r
# Overlap of birth weight
examine_overlap(dt, nbins = 20, var = "momage", var_name = "Birth weight",
    xlab = "Birth weight", ylim = c(-50, 100))
```

Overlap of Birth weight between Groups
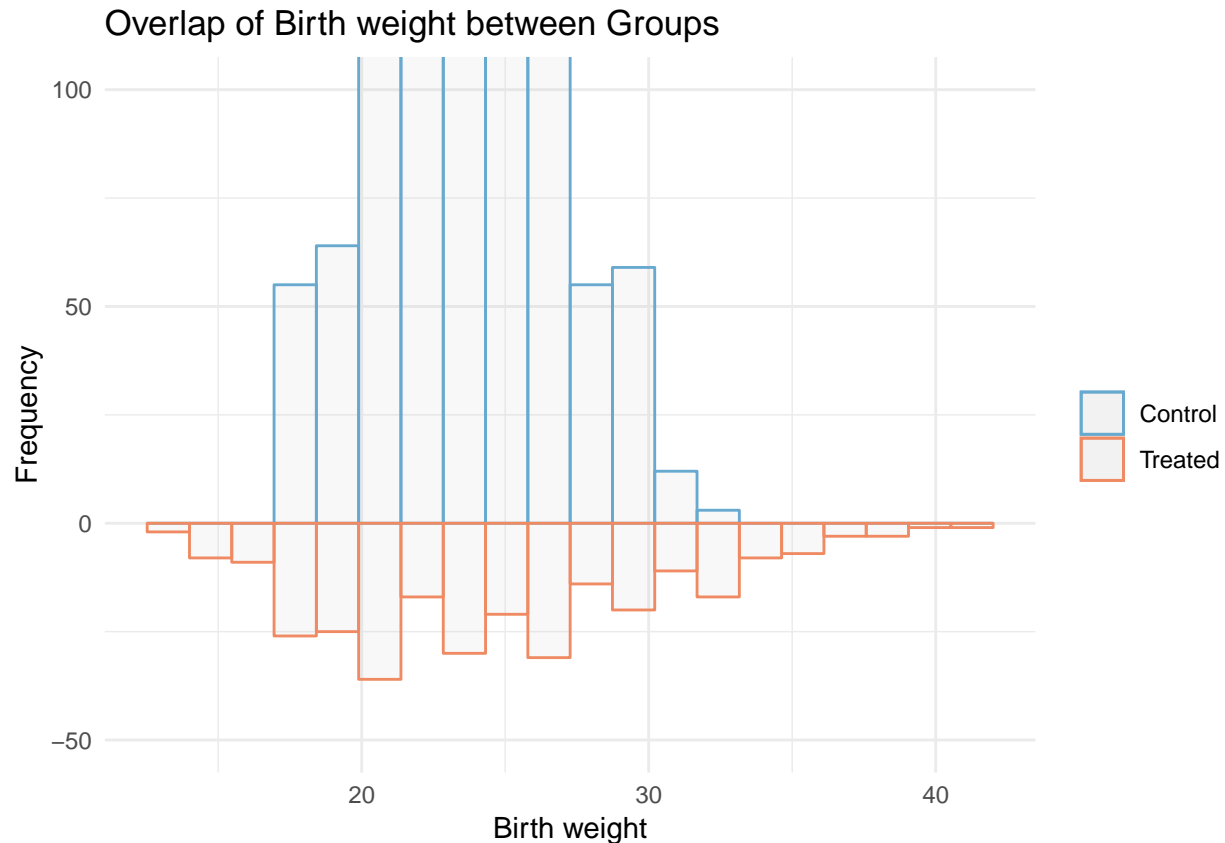
(b) Interpreting Overlap. What do these plots reveal about the overlap required to estimate our estimand of interest.

ANSWER:

These plots reveal we have enough overlap to estimate the average treatment effect on the treated. Between the treatment and control groups, we examined the overlap in propensity scores, preterm, and birth weight covariates. A couple notes: One, due to the imbalance in number of observations between treatment and control groups, comparing paired histograms can misleadingly imply lack of overlap. We addressed this issue by limiting the range of y-axis in the plots, effectively "zooming in", so that all histogram bins are clearly visible. Two, notice in the paired histograms for the propensity score that there are no control observations in the highest bin group of propensity scores (0.975-1.00). This may suggest there is not complete overlap, however, it is also an artifact of the histogram bin size. Doubling the bin size from 0.025 to 0.05 propensity score units eliminates this lack of overlap.

(c) Examining Balance. You will build your own function to check balance! This function should take as inputs the data frame created in Question 1, the vector with the covariate names chosen in Question 1, and the weights created in Question 2. It should output the following:

- 1) Mean in the unmatched treatment group

- 2) Mean in the unmatched control group

- 3) Mean in the matched treatment group

- 4) Mean in the matched control group

- 5) Unmatched mean difference (standardized for continuous variables, not standardized for binary variables)

- 6) Matched mean difference (standardized for continuous variables, not standardized for binary variables)

- 7) Ratio of standard deviations across unmatched groups (control/treated)

- 8) Ratio of standard deviations across matched groups (control/treated)

ANSWER:

```r
checkBalance = function(df, confounders, wt_var) {

    # Subset data to confounders only
    df2 <- df[, confounders]

    binary = apply(df2, 2, function(x) all(x %in% 0:1))  # Binary Variable Indicator
    trt = df$treat == 1  # Treatment Indicator
    ctr = df$treat == 0  # Control Indicator
    n_trt = sum(trt)  # Treatment Sample Size
    n_ctrl = sum(ctr)  # Control Sample Size
    wts.trt = df[trt, wt_var]  # Set treatment weights
    wts.ctr = df[ctr, wt_var]  # Set control weights

    # Means
    trt.means = apply(df2[trt, ], 2, mean)
    trt.means_w = apply(df2[trt, ], 2, wtd.mean, wts.trt)
    ctr.means = apply(df2[ctr, ], 2, mean)
    ctr.means_w = apply(df2[ctr, ], 2, wtd.mean, wts.ctr)

    # Variances
    trt.var = apply(df2[trt, ], 2, var)
    trt.var_w = apply(df2[trt, ], 2, function(x) sum(wts.trt *
        (x - wtd.mean(x, wts.trt))^2)/(sum(wts.trt) - 1))
    ctr.var = apply(df2[ctr, ], 2, var)
    ctr.var_w = apply(df2[ctr, ], 2, function(x) sum(wts.ctr *
        (x - wtd.mean(x, wts.ctr))^2)/(sum(wts.ctr) - 1))

    # Standardized Mean Differences
    mean.diff = (trt.means - ctr.means)/sqrt(trt.var)
    mean.diff.bin = (trt.means - ctr.means)
    diff = mean.diff * (1 - binary) + mean.diff.bin * binary

    mean.diff_w = (trt.means_w - ctr.means_w)/sqrt(trt.var_w)
    mean.diff.bin_w = (trt.means_w - ctr.means_w)
    diff.m = mean.diff_w * (1 - binary) + mean.diff.bin_w * binary

    # Ratios of standard deviations
    ratio = sqrt(ctr.var)/sqrt(trt.var)
    ratio[binary] = NA
```

```
        ratio.m = sqrt(ctr.var_w)/sqrt(trt.var_w)
        ratio.m[binary] = NA

        # Return results
        result = cbind(trt.means, ctr.means, trt.means, ctr.means_w,
            diff, diff.m, ratio, ratio.m)
        colnames(result) = c("mn1", "mn0", "mn1.m", "mn0.m", "diff",
            "diff.m", "ratio", "ratio.m")
        return(result)
}
```

(d) How do you interpret the resulting balance? In particular what are your concerns with regard to covariates that are not well balanced (write about 5 or 6 sentences).

ANSWER:

We focus on the balance of statistics for the matched sample, given that our estimand of interest is the `ATT`. First, we examine the difference in means (standardized for continuous covariates) in the matched sample, and notice similar group means (mean difference < 0.10) in nearly half of the covariates (7 of 15). The fact that the means of several mother-related covariates (momage, b.marr, first, black, hispanic) are dissimilar between the treated and control groups in the matched sample is of concern. Additionally, the ratio of standard deviations in the matched sample is only close to 1.0 for 1 of 3 continuous covariates. Particularly concerning is the greater variance in mother's age at birth in the treatment group compared to the control group.

```
ps.m1.bal <- checkBalance(df = dt, confounders = confounders,
    wt_var = "wt")

# Rounded to two decimal places
round(ps.m1.bal, 2)
```

```
##              mn1     mn0    mn1.m    mn0.m  diff diff.m ratio ratio.m
## momage     24.44   23.54    24.44    23.38  0.15   0.18  0.55    0.49
## b.marr      0.43    0.60     0.43     0.64 -0.16  -0.21    NA      NA
## prenatal    0.96    0.98     0.96     0.98 -0.02  -0.02    NA      NA
## booze       0.12    0.78     0.12     0.15 -0.65  -0.03    NA      NA
## first       0.48    0.45     0.48     0.28  0.04   0.21    NA      NA
## bw       2008.65 2629.48  2008.65  1989.48 -2.19   0.07  1.18    1.05
## preterm     6.07    2.41     6.07     6.25  1.91  -0.09  1.29    1.19
## black       0.50    0.38     0.50     0.28  0.13   0.23    NA      NA
## hispanic    0.09    0.19     0.09     0.45 -0.09  -0.36    NA      NA
## lths        0.43    0.34     0.43     0.60  0.09  -0.16    NA      NA
## hs          0.28    0.42     0.28     0.31 -0.14  -0.02    NA      NA
## college     0.12    0.05     0.12     0.04  0.07   0.07    NA      NA
## st9         0.13    0.02     0.13     0.37  0.11  -0.23    NA      NA
## st25        0.11    0.01     0.11     0.05  0.10   0.06    NA      NA
## st42        0.14    0.04     0.14     0.02  0.11   0.13    NA      NA
```

```
# Plot standardized mean Differences
plotMeanDiff = function(balance_object, name) {
    df = data.frame(balance_object[, c(5, 6)])
    df$var = as.factor(row.names(df))
    colnames(df) = c("unmatched", "matched", "var")
    df = melt(df)
    ggplot(df, aes(x = value, y = var, shape = variable, color = variable)) +
        geom_point() + geom_vline(xintercept = 0, color = "blue") +
```

```r
        geom_vline(xintercept = -0.05, color = "orange", linetype = "dotted") +
        geom_vline(xintercept = 0.05, color = "orange", linetype = "dotted") +
        geom_vline(xintercept = -0.1, color = "red", linetype = "dotted") +
        geom_vline(xintercept = 0.1, color = "red", linetype = "dotted") +
        coord_cartesian(xlim = c(-0.5, 0.5)) + labs(x = "Standardized Difference in Means",
        y = "predictor", title = name, caption = "Orange dotted line indicates +/-0.05 bounds and red li
}

# Plot Ratios of Standard Deviations
plotSDRatios = function(balance_object, name) {
    df = data.frame(balance_object[, c(7, 8)])
    df = df[!is.na(df)[, 1], ]
    df$var = as.factor(row.names(df))
    colnames(df) = c("unmatched", "matched", "var")
    df = melt(df)
    ggplot(df, aes(x = value, y = var, shape = variable, color = variable)) +
        geom_point() + geom_vline(xintercept = 1, color = "blue") +
        labs(x = "Ratios of Standard Deviations", y = "predictor",
            title = name)
}

plotMeanDiff(ps.m1.bal, "Propensity Score Model: Logistic Regression")
```
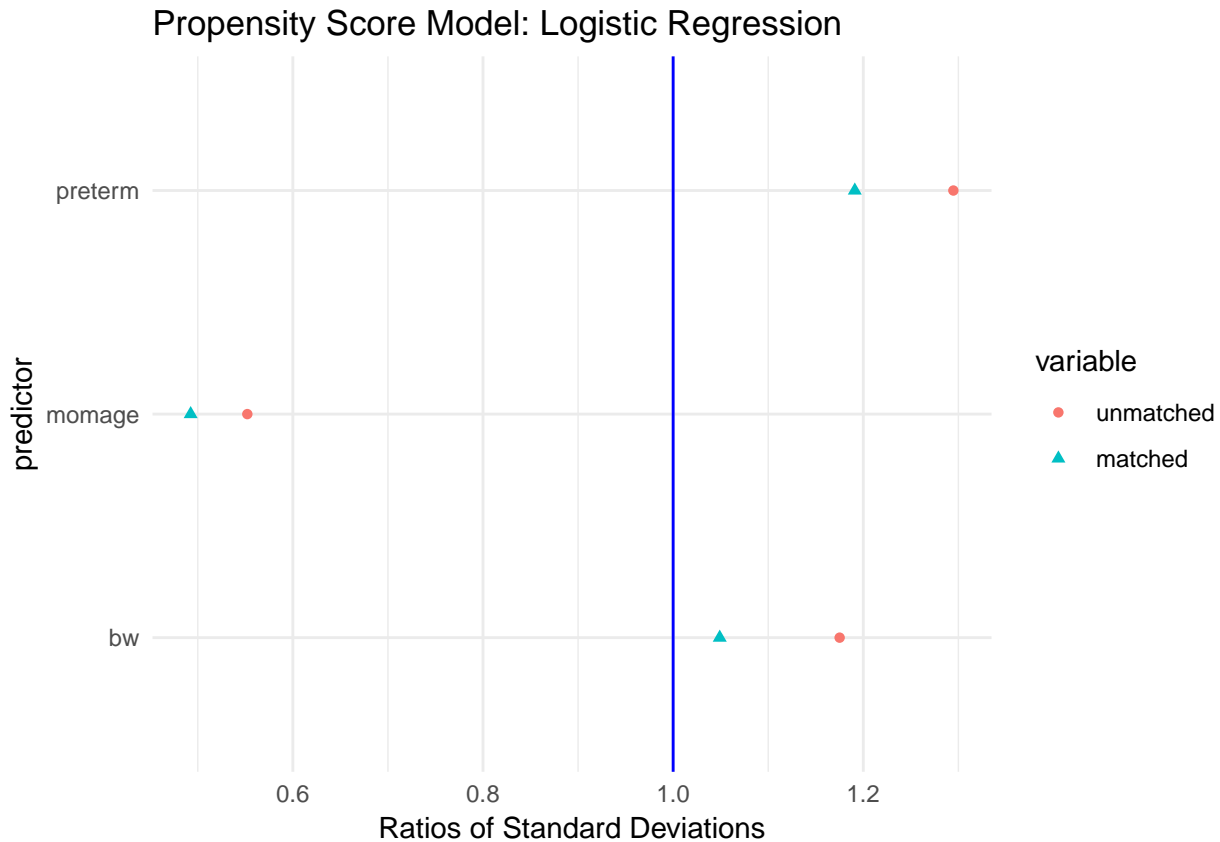
Using var as id variables

Propensity Score Model: Logistic Regression

Orange dotted line indicates +/−0.05 bounds and red lines indicate +/−0.1 bounds.
Values outside of range x=[−0.5,0.5] not displayed.

```
plotSDRatios(ps.m1.bal, "Propensity Score Model: Logistic Regression")
```

```
## Using var as id variables
```


Propensity Score Model: Logistic Regression

(e) Unit test. Show the results of your balance function on a simple example with the same sample as above (that is, limited to children with birth weight less than 3000) where the propensity score is fit using logistic regression on "bw" and "b.marr" and the matching is performed using 1-1 nearest neighbor matching with replacement. The output of your balance function should match the following (when rounded to 3 decimal places):

ANSWER:

```
# Subset data
temp <- hw4[hw4$bw < 3000, c("treat", "bw", "b.marr")]

# Fit propensity score model (logistic)
ps.temp <- glm(treat ~ bw + b.marr, data = temp, family = binomial(link = "logit"))

# Generate propensities scores
temp$psc <- ps.temp$fitted.values

# 1-1 nearest neighbor matching with replacement
ps.temp.matches <- matching(z = temp$treat, score = temp$psc,
    replace = TRUE)
temp[temp$treat == 0, "wt"] <- ps.temp.matches$cnts
temp[temp$treat == 1, "wt"] <- 1
```

```
# Rounded to 3 decimal places
round(checkBalance(temp, c("bw", "b.marr"), "wt"), 3)
```

```
##              mn1      mn0    mn1.m    mn0.m   diff diff.m ratio ratio.m
## bw      2008.648 2629.482 2008.648 2001.838 -2.191  0.024 1.175   1.044
## b.marr     0.431    0.595    0.431    0.486 -0.164 -0.055    NA      NA
```

Question 5: Repeat steps 2-4 within the matching framework.

ANSWER:

We use the following propensity score models to try to achieve better balance:
- Probit regression

We use the following matching methods to try to achieve better balance:
- Mahalanobis Distance Matching
- Matching without Replacement (Using propensity scores from original logistic regression)
- Nearest Neighbor Matching with Replacement

Brief descriptions of each of the methods:

(1) Probit Regression

A probit regression is equivalent to the logistic regression except that it replaces the logistic with the normal distribution. So:

$$Pr(y_i = 1) = \Phi(X_i B)$$

Where Φ is the normal cumulative distribution function. There is strong overlap between the control and treatment distributions over the propensity score so no observations were dropped. We using matching with replacement

(2) Mahalanobis Matching

Mahalanobis matching uses a distance measure in multivariate space that takes into account variances of variables as well as covariances between them.

(3) Matching without Replacement (Using propensity scores from original logistic regression)

We use the original propensity scores Q2. This time, we match without replacement.

(4) Nearest Neighbor Matching with Replacement

We use K-closest matching with k = 2. This matching method selects the 2 control subjects with the closest distance to the treated subject (using `default=logit`). Matches are chosen for treatment subjects one at a time using the default order (largest to smallest). We allow for replacement.

Fit new propensity score models

```
# Model 2: Probit
ps.m2 <- glm(treat ~ ., data = dt[, c("treat", confounders)],
    family = binomial(link = "probit"))
dt$psc_probit <- ps.m2$fitted.values
confusionMatrix(factor(ifelse(dt$psc_probit > 0.5, 1, 0)), factor(dt$treat))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
```

```
##          0 995  40
##          1  35 250
##
##                Accuracy : 0.9432
##                  95% CI : (0.9293, 0.9551)
##     No Information Rate : 0.7803
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8332
##  Mcnemar's Test P-Value : 0.6442
##
##             Sensitivity : 0.9660
##             Specificity : 0.8621
##          Pos Pred Value : 0.9614
##          Neg Pred Value : 0.8772
##              Prevalence : 0.7803
##          Detection Rate : 0.7538
##    Detection Prevalence : 0.7841
##       Balanced Accuracy : 0.9140
##
##        'Positive' Class : 0
##
```

Matching with new models

```r
# Rematching

# Probit
matches <- matching(z = dt$treat, score = dt$psc_probit, replace = TRUE)
dt[dt$treat == 1, "w_probit"] <- 1
dt[dt$treat == 0, "w_probit"] <- matches$cnt

# Matching without Replacement using Original Logit Model
matches_nr <- matching(z = dt$treat, score = dt$psc, replace = FALSE)
matches_nr_idx <- c(which(dt$treat == 1), matches_nr$matched[1:290])
dt[matches_nr_idx, "w_logit_no_rep"] = 1
dt$w_logit_no_rep[is.na(dt$w_logit_no_rep)] = 0
```

Calculate the Mahalanobis distance for matching:

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}$$

We constructed our own Mahalanobis Distance function. As shown below, it gets the same results as the `MatchIt` package.

```r
# Mahalanobis Distance function, d(x,y)
myMH = function(trt, ctr, inv.cov, data) {
    x = as.matrix(data[trt, ])
    y = as.matrix(data[ctr, ])
    diff = x - y
    sqrt(rowSums((diff %*% inv.cov) * diff))
}

inv_cov_mat = solve(cov(dt[, confounders]))
trt = row.names(dt[dt$treat == 1, ])
ctr = row.names(dt[dt$treat == 0, ])
```

```r
# Compute distances for all pairwise distances between
# treated and control
mahalo_dist = outer(trt, ctr, FUN = myMH, inv.cov = inv_cov_mat,
    data = dt[, confounders])

# Find Matches (Smallest Distance)
matches = apply(mahalo_dist, 1, function(x) min(which(x == min(x))))

# Get Weights
m_freq = data.frame(table(matches))
ctr.idx = data.frame(idx = seq(1:sum(dt$treat == 0)))
matches = merge(ctr.idx, m_freq, by.x = "idx", by.y = "matches",
    all.x = T)

# Set Weights
dt[dt$treat == 0, "wt_mh"] = matches$Freq * 0.5241379
dt[is.na(dt$wt_mh), "wt_mh"] = 0
dt[dt$treat == 1, "wt_mh"] = 1

# Check against MatchIt function
df = dt[, c("treat", confounders)]
zz <- matchit(treat ~ momage + b.marr + prenatal + booze + first +
    bw + preterm + black + hispanic + lths + hs + college + st9 +
    st25 + st42, data = df, method = "nearest", distance = "mahalanobis",
    replace = TRUE)

# Check weights derived from our Mahalanobis matches to the
# MatchIt matches (up to 5 decimal places)
all(round(dt$wt_mh, 5) == round(zz$weights, 5))
```

```
## [1] TRUE
```

K-2 Matching with Replacement

```r
# Check against MatchIt function
df = dt[, c("treat", confounders)]
k2 <- matchit(treat ~ momage + b.marr + prenatal + booze + first +
    bw + preterm + black + hispanic + lths + hs + college + st9 +
    st25 + st42, data = df, method = "nearest", ratio = 2, replace = T)
dt$wt_k2 = k2$weights
```

Check overlap of matched sample

```r
title <- ggtitle("Propensity Score Model: Probit Regression")
examine_overlap(dt[dt$w_probit != 0, ], nbins = 40, var = "psc_probit",
    var_name = "Propensity Score", xlab = "Propensity Score",
    ylim = c(-10, 10), title)
```

## Propensity Score Model: Probit Regression


```
title <- ggtitle("Matching Method: Logistic w/o Replacement")
examine_overlap(dt[dt$w_logit_no_rep != 0, ], nbins = 40, var = "psc",
    var_name = "Propensity Score", xlab = "Propensity Score",
    ylim = c(-10, 10), title)
```

Matching Method: Logistic w/o Replacement


```r
title <- ggtitle("Matching Method: Mahalanobis")
examine_overlap(dt[dt$wt_mh != 0, ], nbins = 20, var = "preterm",
    var_name = "Number Weeks Baby Born Preterm", xlab = "Weeks Preterm",
    ylim = c(-50, 50), title)
```
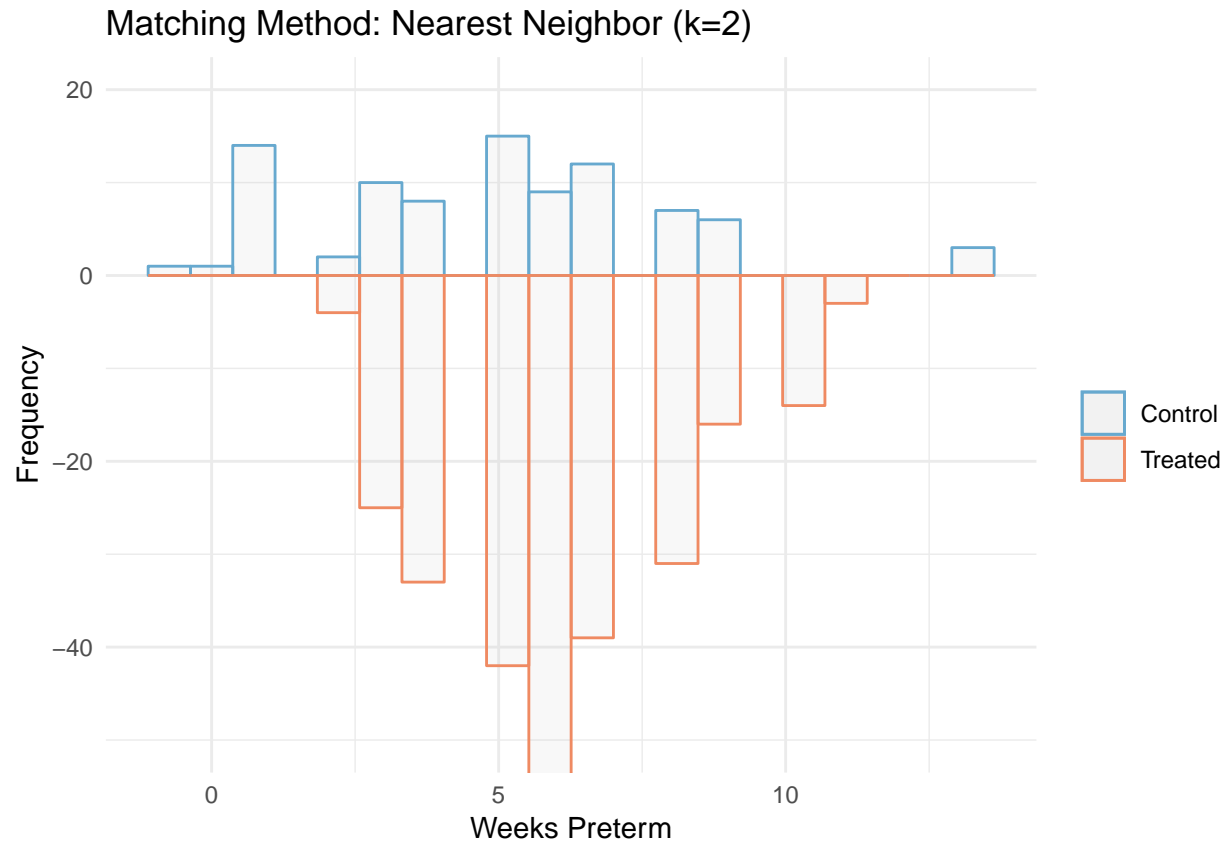
Matching Method: Mahalanobis
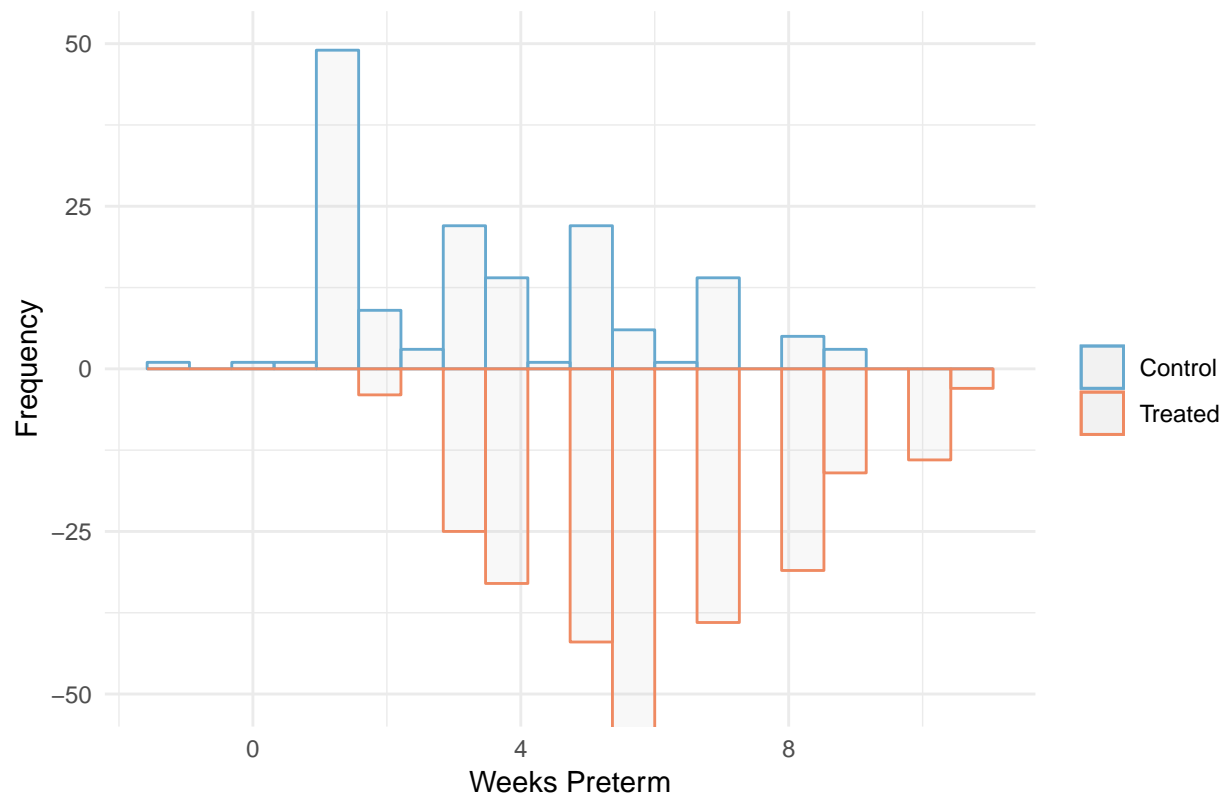

```
title <- ggtitle("Matching Method: Nearest Neighbor (k=2)")
examine_overlap(dt[dt$wt_k2 != 0, ], nbins = 20, var = "preterm",
    var_name = "Number Weeks Baby Born Preterm", xlab = "Weeks Preterm",
    ylim = c(-50, 20), title)
```

Matching Method: Nearest Neighbor (k=2)

Check balance

```
logit.bal <- checkBalance(dt, confounders, "wt")
probit.bal <- checkBalance(dt, confounders, "w_probit")
logit_nr.bal <- checkBalance(dt, confounders, "w_logit_no_rep")
mh.bal <- checkBalance(dt, confounders, "wt_mh")
mk2.bal <- checkBalance(dt, confounders, "wt_k2")

plotMeanDiff(logit.bal, "Propensity Score Model: Logistic Regression")
```

```
## Using var as id variables
```

Propensity Score Model: Logistic Regression

Orange dotted line indicates +/−0.05 bounds and red lines indicate +/−0.1 bounds.
Values outside of range x=[−0.5,0.5] not displayed.

```
plotMeanDiff(probit.bal, "Propensity Score Model: Probit Regression")
```

```
## Using var as id variables
```

Propensity Score Model: Probit Regression

Orange dotted line indicates +/−0.05 bounds and red lines indicate +/−0.1 bounds.
Values outside of range x=[−0.5,0.5] not displayed.
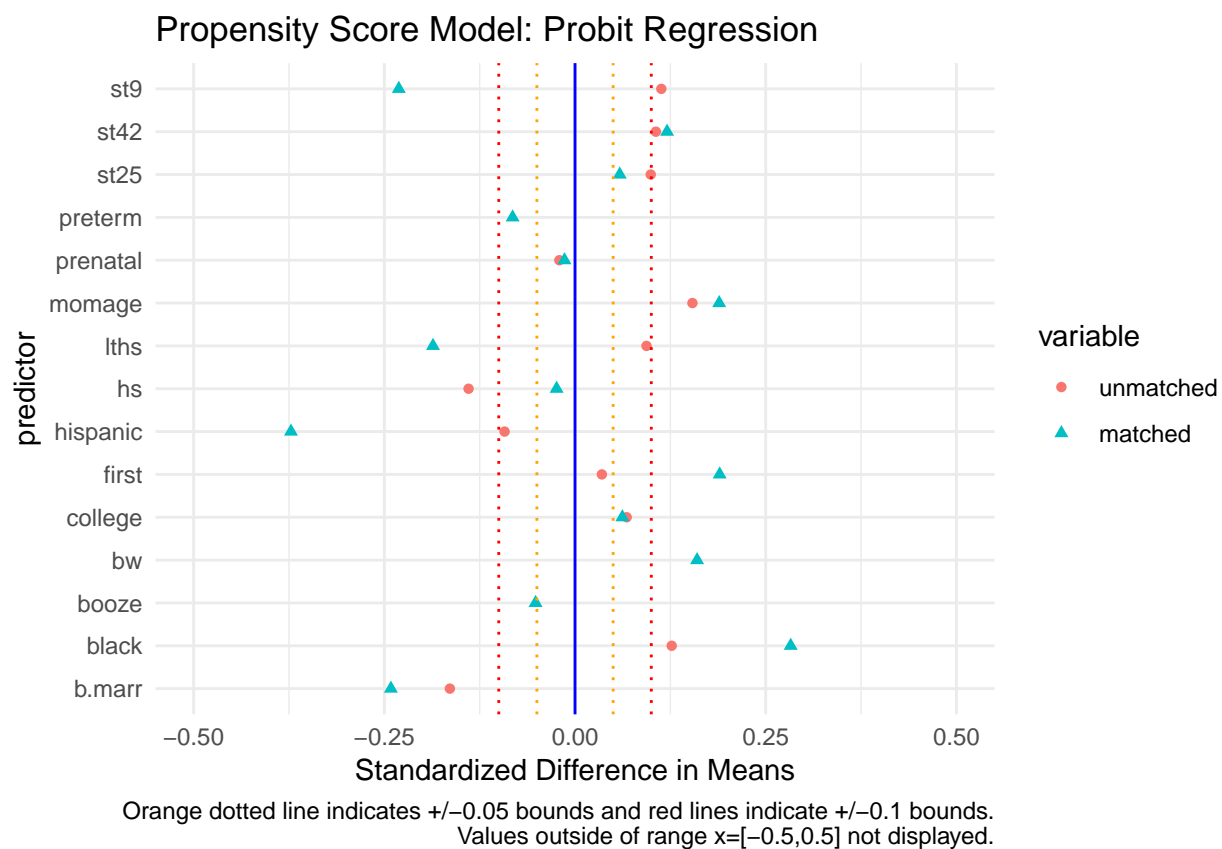
```
plotMeanDiff(logit_nr.bal, "Matching Method: Logistic w/o Replacement")
```
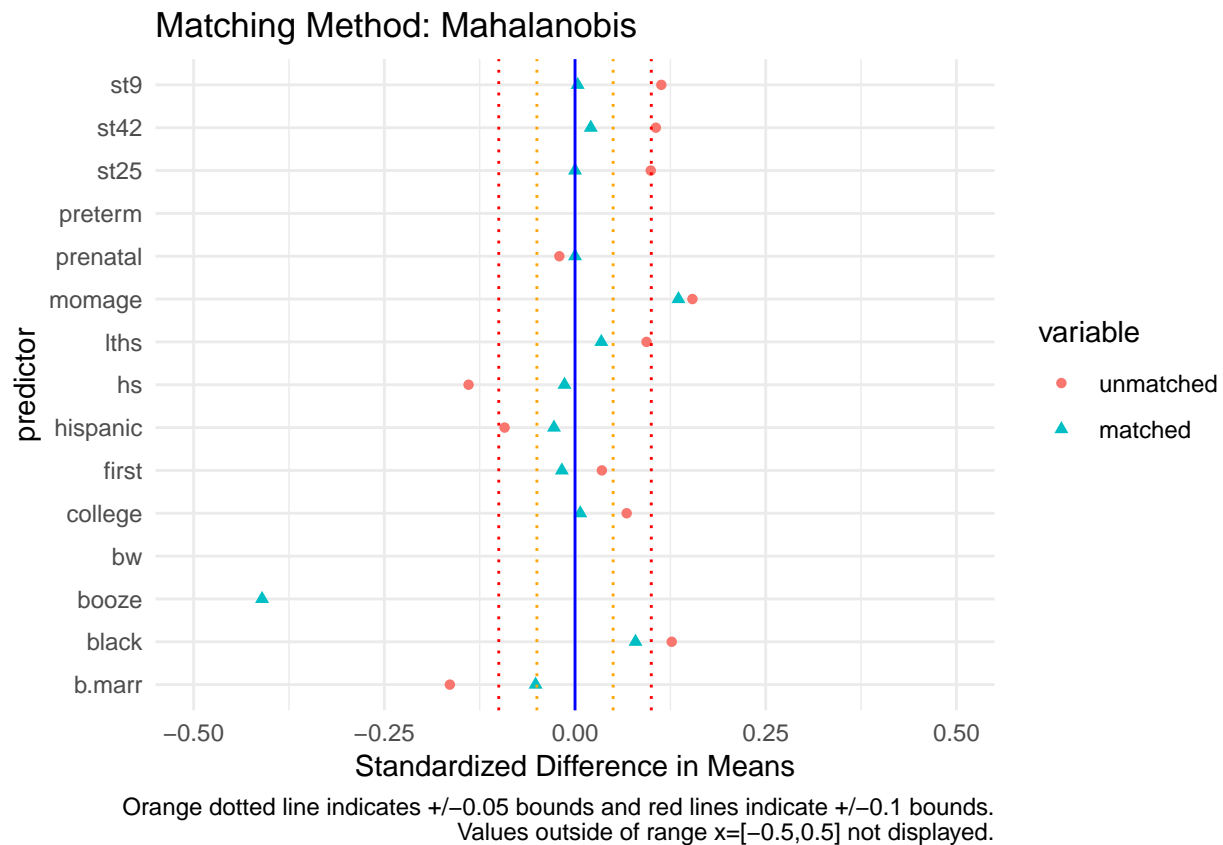
```
## Using var as id variables
```

Matching Method: Logistic w/o Replacement

Orange dotted line indicates +/−0.05 bounds and red lines indicate +/−0.1 bounds.
Values outside of range x=[−0.5,0.5] not displayed.
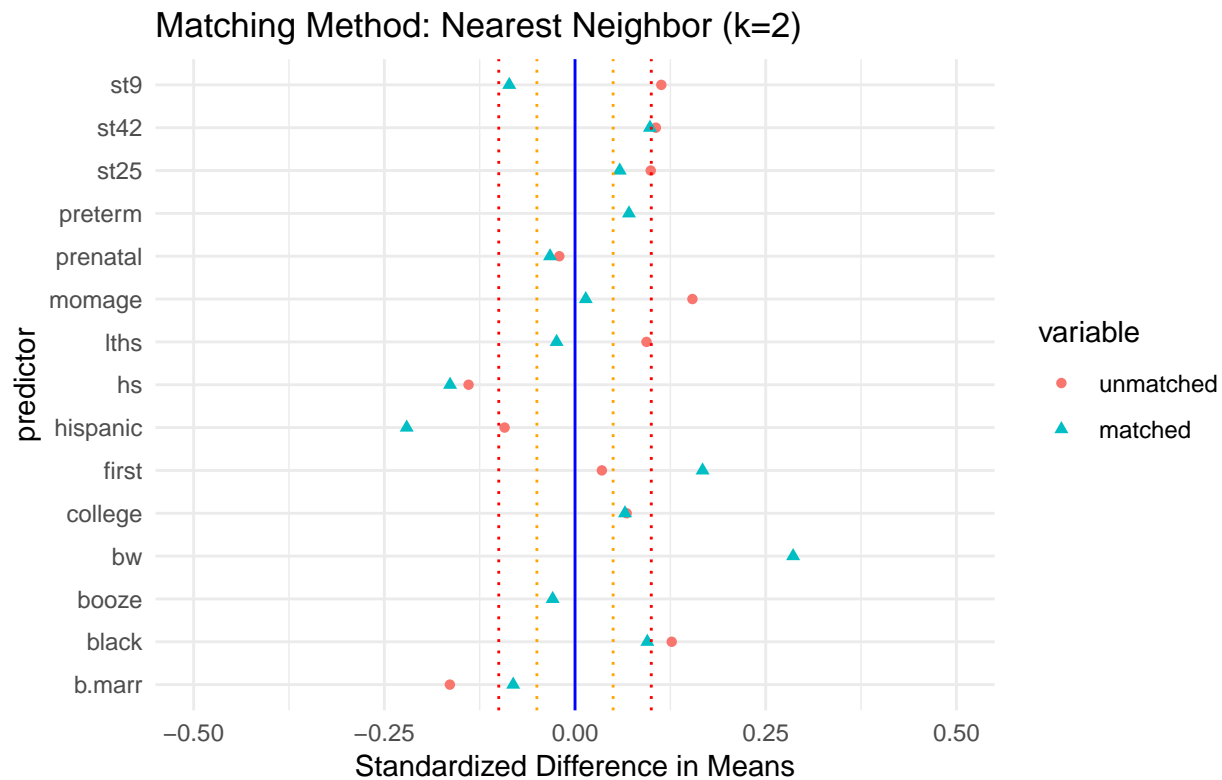
```
plotMeanDiff(mh.bal, "Matching Method: Mahalanobis")
```

```
## Using var as id variables
```

Matching Method: Mahalanobis

Orange dotted line indicates +/−0.05 bounds and red lines indicate +/−0.1 bounds.
Values outside of range x=[−0.5,0.5] not displayed.

```
plotMeanDiff(mk2.bal, "Matching Method: Nearest Neighbor (k=2)")
```

```
## Using var as id variables
```

Matching Method: Nearest Neighbor (k=2)

Orange dotted line indicates +/−0.05 bounds and red lines indicate +/−0.1 bounds.
Values outside of range x=[−0.5,0.5] not displayed.

```r
# Plot SD Ratios
temp = rbind(logit.bal, probit.bal, logit_nr.bal, mh.bal, mk2.bal)

temp = temp[order(rownames(temp)), ]
row.names(temp) = paste(row.names(temp), rep(c("logit", "probit",
    "logit_no_replace", "mahalanobis", "neighbor"), times = 15),
    sep = "_")
plotSDRatios(temp, "Comparing SD Ratios of All Propensity Score Models\n and Matching Methods Used")

## Using var as id variables
```
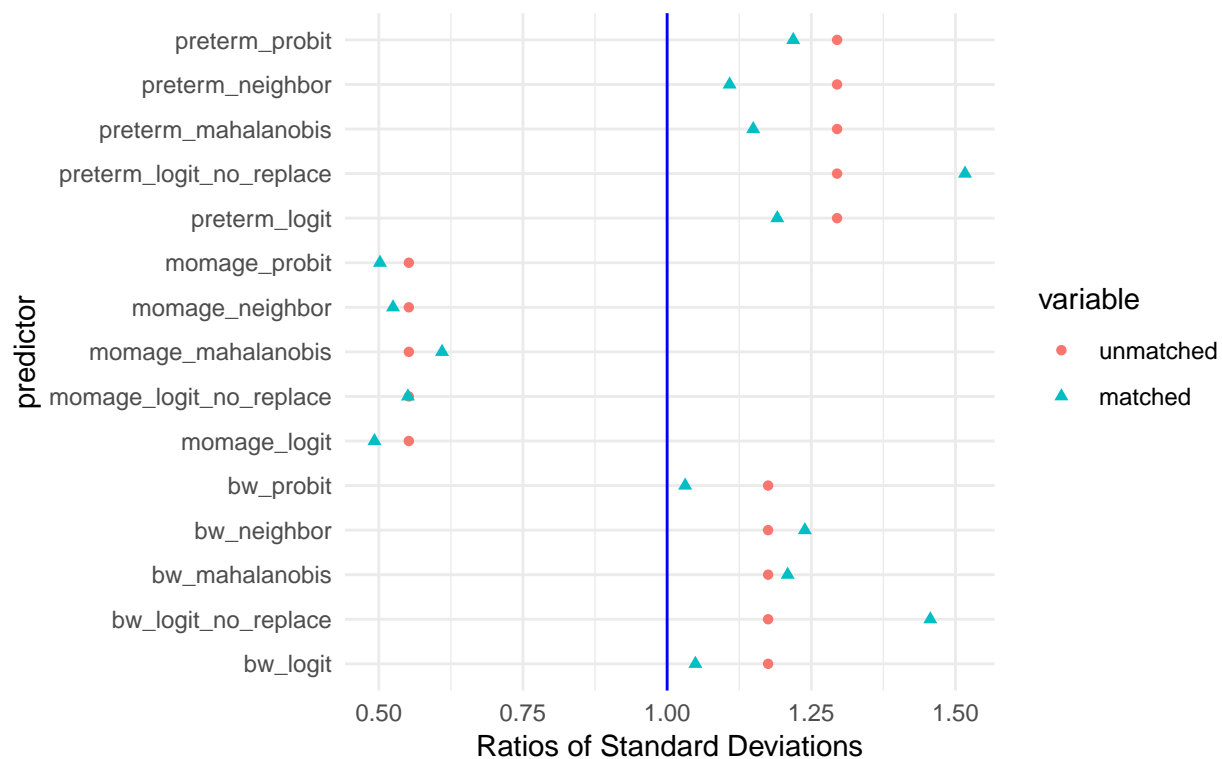
Comparing SD Ratios of All Propensity Score Models and Matching Methods Used

Question 6: Repeat steps 2-4, but this time using IPTW.

ANSWER:

```
ps.IPTW <- glm(treat ~ ., data = dt[, c("treat", confounders)],
    family = binomial(link = "logit"))
dt$psc_iptw <- ps.IPTW$fitted.values
```

For estimating the ATT, define the weight as follows:

$$\omega(Z, x) = Z + (1 - Z) * \frac{\hat{\epsilon}(x)}{1 - \hat{\epsilon}(x)}$$

```
dt$r_att = dt$psc + (1 - dt$treat) * dt$psc/(1 - dt$psc)
```

```
iptw.balance <- checkBalance(dt, confounders, "r_att")
```

Question 7: Comparative balance table. Create a table with columns 6 and 8 from your function for each of the matching and weighting methods performed above. Which approach would you choose and why? (1-2 paragraphs at most)

ANSWER:

```r
compareBalance <- function(model_bals, model_names, col) {
    # model_bals: needs to be a list() of outputs from
    # checkBalance model_names: vector of model names in same
    # order as bals col: col to extract for output: [1,8]

    # number of models
    n_models <- length(model_bals)
    # subset out columns 6 & 8
    temp.ls <- lapply(model_bals, function(bal) bal[, col])
    # cbind list of df's into one df
    temp.df <- data.frame(do.call("cbind", temp.ls))
    # kable output
    names(temp.df) <- model_names
    if (interactive()) {
        return(round(temp.df, 2))
    } else {
        out <- kable(temp.df, digits = 2, align = "c", format = "latex")
        if (col == 6) {
            out <- out %>% add_header_above(c(` ` = 1, `Mean Difference (matched)` = n_models))
        } else {
            out <- out %>% add_header_above(c(` ` = 1, `Ratio sd (matched)` = n_models))
        }
        return(out)
    }
}


model_bals <- list(logit.bal, probit.bal, logit_nr.bal, mh.bal,
    mk2.bal, iptw.balance)
model_names <- c("Logistic", "Probit", "Logistic (no replacement)",
    "Mahalanobis Matching", "K=2 Nearest Neighbors", "IPTW")

compareDiffs <- compareBalance(model_bals, model_names, 6)
compareRatios <- compareBalance(model_bals, model_names, 8)
```

	Mean Difference (matched)					
	Logistic	Probit	Logistic (no replacement)	Mahalanobis Matching	K=2 Nearest Neighbors	IPTW
momage	0.18	0.19	0.07	0.14	0.01	0.05
b.marr	-0.21	-0.24	-0.08	-0.05	-0.08	-0.11
prenatal	-0.02	-0.01	-0.03	0.00	-0.03	-0.04
booze	-0.03	-0.05	-0.29	-0.41	-0.03	-0.15
first	0.21	0.19	0.06	-0.02	0.17	0.17
bw	0.07	0.16	-1.31	-1.58	0.29	-0.02
preterm	-0.09	-0.08	1.07	1.34	0.07	0.25
black	0.23	0.28	0.09	0.08	0.09	0.16
hispanic	-0.36	-0.37	-0.05	-0.03	-0.22	-0.25
lths	-0.16	-0.19	0.06	0.03	-0.02	-0.02
hs	-0.02	-0.02	-0.10	-0.01	-0.16	-0.13
college	0.07	0.06	0.06	0.01	0.07	0.04
st9	-0.23	-0.23	0.08	0.00	-0.09	-0.04
st25	0.06	0.06	0.07	0.00	0.06	0.05
st42	0.13	0.12	0.08	0.02	0.10	0.10

	Ratio sd (matched)					
	Logistic	Probit	Logistic (no replacement)	Mahalanobis Matching	K=2 Nearest Neighbors	IPTW
momage	0.49	0.50	0.55	0.61	0.52	0.52
b.marr	NA	NA	NA	NA	NA	NA
prenatal	NA	NA	NA	NA	NA	NA
booze	NA	NA	NA	NA	NA	NA
first	NA	NA	NA	NA	NA	NA
bw	1.05	1.03	1.46	1.21	1.24	1.37
preterm	1.19	1.22	1.52	1.15	1.11	1.22
black	NA	NA	NA	NA	NA	NA
hispanic	NA	NA	NA	NA	NA	NA
lths	NA	NA	NA	NA	NA	NA
hs	NA	NA	NA	NA	NA	NA
college	NA	NA	NA	NA	NA	NA
st9	NA	NA	NA	NA	NA	NA
st25	NA	NA	NA	NA	NA	NA
st42	NA	NA	NA	NA	NA	NA

We compare the mean differences and ratios of standard deviations among confounders between the treatment and control groups to assess which propensity and/or matching approach results in the best balance. We aim to reduce the mean differences to below 0.10 for continuous variables and 0.05 for binary variables. For the ratio of standard deviations, we aim to get as close to 1.0 as possible. Considering the number of confounders and approaches, the above tables can make assessments of which approach is best due to the number of values to compare. The plots from Question 5 aid in comparing mean differences and ratios of standard deviations. We prioritize balancing mean differences over ratios because we only consider 3 continuous confounders, and none of the approaches are significantly better in balancing the ratios. The Mahalanobis matching approach achieved the best balance in mean differences, and thus we choose this approach.

Question 8: Estimate the treatment effect for the restructured datasets implied by Questions 4-6 (Step 5)

Estimate the effect of the treatment on the treated for each of your five approaches by fitting a regression with weights equal to the number of times each observation appears in the matched sample (that is, use your weights variable from above) or using IPTW weights. Report the treatment effect and standard error for each approach.

ANSWER:

```
temp <- dt[, c("ppvtr.36", "treat", confounders)]

mod1 = summary(lm(ppvtr.36 ~ ., data = temp, weights = dt$wt))$coefficients
mod1.out = c(mod1["treat", "Estimate"], mod1["treat", "Std. Error"])

mod2 = summary(lm(ppvtr.36 ~ ., data = temp, weights = dt$w_probit))$coefficients
mod2.out = c(mod2["treat", "Estimate"], mod2["treat", "Std. Error"])

mod3 = summary(lm(ppvtr.36 ~ ., data = temp, weights = dt$w_logit_no_rep))$coefficients
mod3.out = c(mod3["treat", "Estimate"], mod3["treat", "Std. Error"])

mod4 = summary(lm(ppvtr.36 ~ ., data = temp, weights = dt$wt_mh))$coefficients
mod4.out = c(mod4["treat", "Estimate"], mod4["treat", "Std. Error"])

mod5 = summary(lm(ppvtr.36 ~ ., data = temp, weights = dt$wt_k2))$coefficients
```

```
mod5.out = c(mod5["treat", "Estimate"], mod5["treat", "Std. Error"])

mod6 = summary(lm(ppvtr.36 ~ ., data = temp, weights = dt$r_att))$coefficients
mod6.out = c(mod5["treat", "Estimate"], mod5["treat", "Std. Error"])

q8_results = rbind(mod1.out, mod2.out, mod3.out, mod4.out, mod5.out,
    mod6.out)
row.names(q8_results) = c("logit (w/ replacement)", "probit (w/ replacement)",
    "logit (no replacement)", "Mahalanobis Matching", "K-2 Nearest Neighbors",
    "IPTW")
colnames(q8_results) = c("Treatment Effect", "Std. Error")
kable(q8_results, digits = 2, align = "c", format = "markdown",
    caption = "Treatment Effect Estimation by Propensity Score Model")
```

	Treatment Effect	Std. Error
logit (w/ replacement)	2.74	2.29
probit (w/ replacement)	3.88	2.25
logit (no replacement)	11.25	1.95
Mahalanobis Matching	13.12	2.13
K-2 Nearest Neighbors	3.44	2.22
IPTW	3.44	2.22

Question 9: Assumptions What assumptions are necessary to interpret the estimates from the propensity score approaches causally? List and describe briefly.

ANSWER:

(a) Ignorability

Ignorability holds if the covariates in the propensity score model are the only confounding covariates and we match on the propensity score. So we need to assume that we have controlled for all the potential confounders. Then we can unbiasedly estimate $E[Y(1)|Z = 1, e(X)]$ with $\bar{Y}_{Z=1,e(X)}$ and $E[Y(0)|Z = 1, e(X)]$ with $\bar{Y}_{Z=0,e(X)}$.

(b) Sufficient Overlap

If we are interested in the effect of the treatment on the treated, we want to make sure that for each treatment group member there is a control group member that is sufficiently similar. Then, we can use this control group member as an empirical counterfactual.

Similarly, if we are interested in the effect of the treatment on the control, we want to make sure that for each control group member there is a treatment group member that is sufficiently similar. Then, we can use this treatment group member as an empirical counterfactual.

If we are interested in the average treatment effect, we want to make sure that for each subject in either treatment or control, there is an empirical counterfactual (i.e. a sufficiently similar control subject for a treatment subject and vise versa).

(c) Appropriate specification of the propensity score model / balance achieved

An appropriate specification of the propensity score model ensures that we have good overlap and balance. Overlap is important with regard to drawing causal inference due to reasons listed above. Balance is important because imbalance forces us to rely more on the correct functional form of our model. Incorrect functional forms would lead to biased estimates of the treatment effect.

(d) SUTVA

Stable unit treatment value assumption states that the treatment effect does not depend on the particular configuration of treatment assignment. That is, there is no dilution or concentration of the treatment effect.

Question 10: Causal Interpretation Provide a causal interpretation of *one* of your estimates above. Remember to specify the counterfactual and to be clear about whom you are making inferences. Also make sure to use causal (counterfactual) language.

ANSWER:
For children who participated in the IDHP intervention, their IQ scores at age 3 were 13.1 points higher than had they not participated in the intervention.

Question 11: Comparison to linear regression Fit a regression of your outcomes to the treatment indicator and covariates.

(a) Report your estimate and standard error.

ANSWER:

```
lm1 <- lm(ppvtr.36 ~ ., data = dt[, c(confounders, "ppvtr.36",
    "treat")])
round(summary(lm1)$coefficients["treat", c("Estimate", "Std. Error"),
    drop = FALSE], 2)
```

```
##       Estimate Std. Error
## treat    11.54       1.76
```

(b) Interpret your results non-causally

ANSWER:
A group of children who participate in the IHDP intervention will expect, on average, an 11.54 point increase in their age 3 IQ scores compared to a group of children who did not participate in the intervention, controlling for all other covariates.

(c) Why might we prefer the results from the propensity score approach to the linear regression results in terms of identifying a causal effect?

ANSWER:
In terms of estimating causal effects, propensity score approaches are preferred over linear regression to account for the systematic differences between the treatment and control groups in key covariates that can affect the outcome. While linear regression can account for some group differences, the method is highly dependent on correct model specificity in order avoid treatment effect estimation bias. Propensity score approaches decrease this bias by restructuring the sample of controls and/or treated observations so that there is overlap in the common support of the covariates between groups, and improved balance in distributions of the covariates between groups. Propensity score approaches decrease the treatment effect estimation bias caused by extrapolating into areas of the covariate space outside of the support of the data, and incorrect model specification of the treatment and covariates on the outcome.