HW4 - Propensity Scores

Kenny Mai, George Perrett

YOU MAY WORK IN PAIRS FOR THIS ASSIGNMENT ONLY

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.

R Packages

You will need to use an R package that you may not already have installed, arm.

```
# Load packages
# arm is required
library(arm)
```

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

```
## Loading required package: Matrix
```

```
## Loading required package: lme4
```

```
##
## arm (Version 1.11-2, built: 2020-7-27)
```

```
## Working directory is C:/Users/kenny/Documents/R/2012-Causal-Inference
```

```
# ggplot2 for prettier graphs
library(ggplot2)
# tidyverse for data manipulation
library(tidyverse)
```

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

```
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x dplyr::select() masks MASS::select()
## x tidyr::unpack() masks Matrix::unpack()
```

```
# Hmisc for wtd.var() later
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# MatchIt for matchit() later
library(MatchIt)
```

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) Load the data (you can use the load command since the data are in a .Rdata file) and choose the covariates you want to use as confounders. To make life easier you many want to choose binary indicators of unordered categorical variables (rather than a variable labeled e.g. as 1, 2, 3 for different levels of a categorical variable).

```
# Load data from local folder to workspace as hw4
load("~/R/2012-Causal-Inference/hw4.Rdata")
```

Create a new data frame for analysis that includes the outcome in the 1st column, the treatment indicator in the 2nd column, and the covariates in the remaining columns. Be thoughtful about your choices with respect to the nature of the covariates (e.g. is an unordered categorical being represented as such) and timing (don't control for post-treatment variables!). Provide your code and a list of the variable names for the confounder variables chosen.

Also reduce that data frame to include only observations for children whose birthweight is less than 3000 grams.

```
# Subsetting for only children whose birthweight is less than 3000 grams with tidyverse
df1 <- hw4 %>% filter(bw < 3000) %>% dplyr::select(-momed) %>% dplyr::select(ppvtr.36,treat,everything()
# Personally choosing all covariates, placing them into a vector
cov_names0 <- names(df1[3:length(df1)])
```

Question 2: Estimate the propensity score (Step 2) Estimate the propensity score. That is, fit a propensity score model and save the predicted scores.

```
# Fit GLM model
glm0 <- glm(treat ~ .-1,
            family=binomial(), data=df1)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# Use predict() to generate propensity scores
pscores0 <- predict(glm0, type="response")
```

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?

Estimand of interest is average treatment of 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.

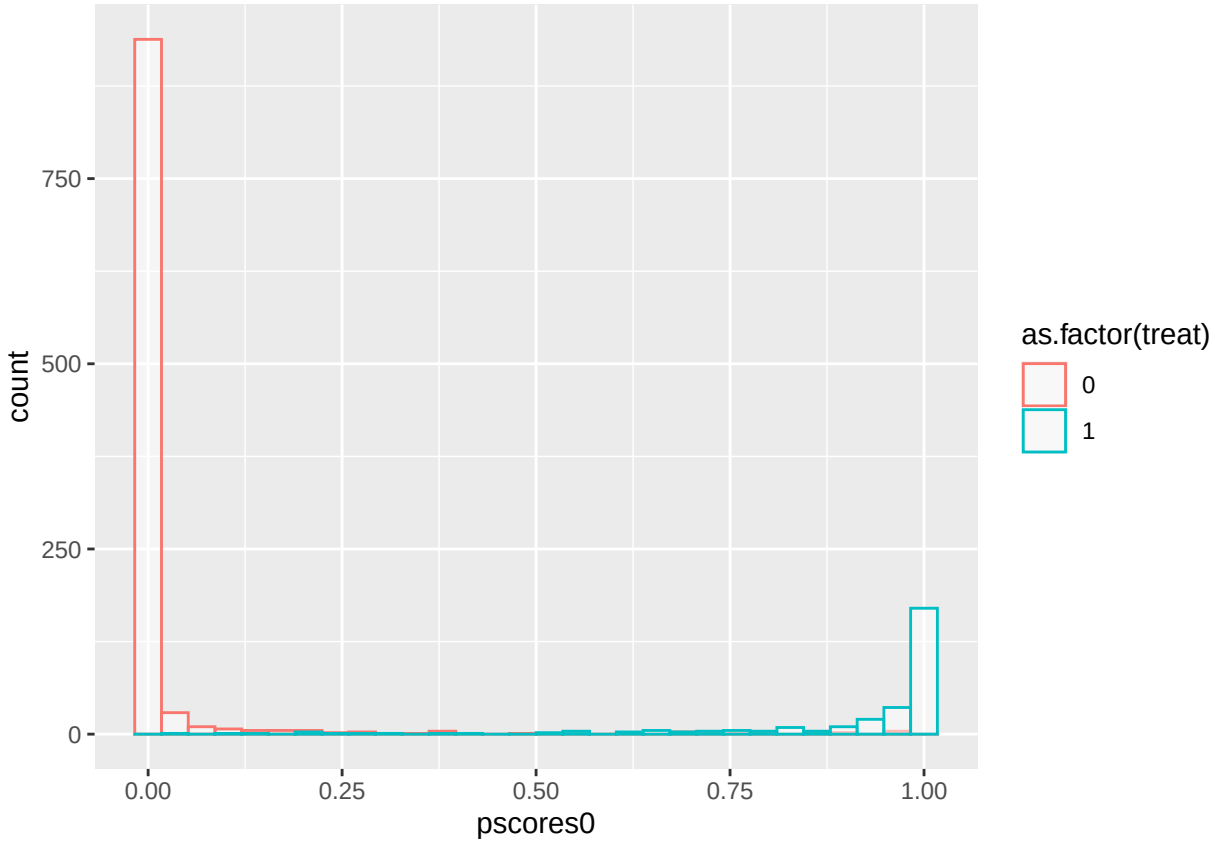
```
# Use matching() from arm to do propensity score matching
match0 <- matching(df1$treat,pscores0,replace=TRUE)
```

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

(a) Examining Overlap. Check overlap on the raw data (that is that data before matching) using some diagnostic plots. Check overlap for the propensity scores as well as two other covariates. Note that it may be necessary to exclude some observations from the plots if they are being obscured in ways similar to the example discussed in class on 10/5.

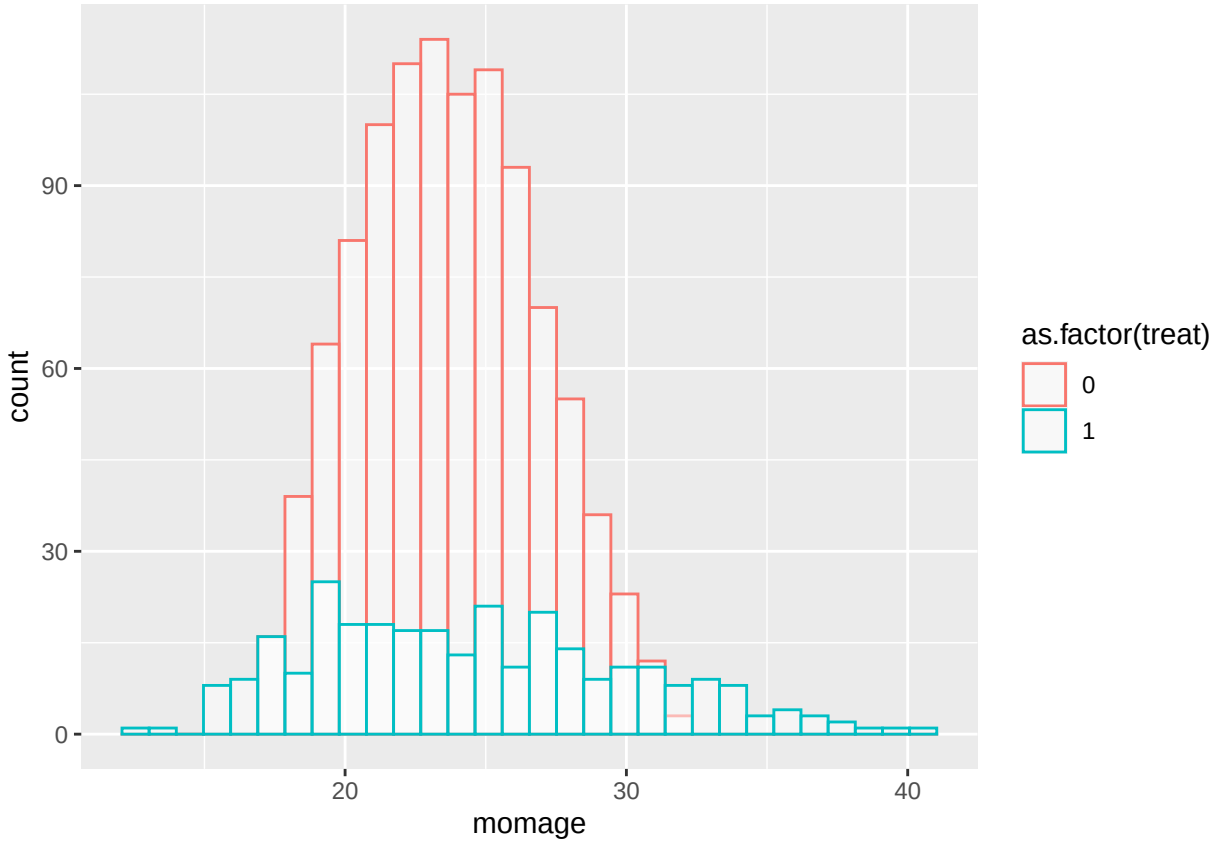
```
# Check overlap of the raw data
ggplot() +
  geom_histogram(data = df1, aes(x=pscores0,color=as.factor(treat)),fill="white",alpha=0.5,position="id
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



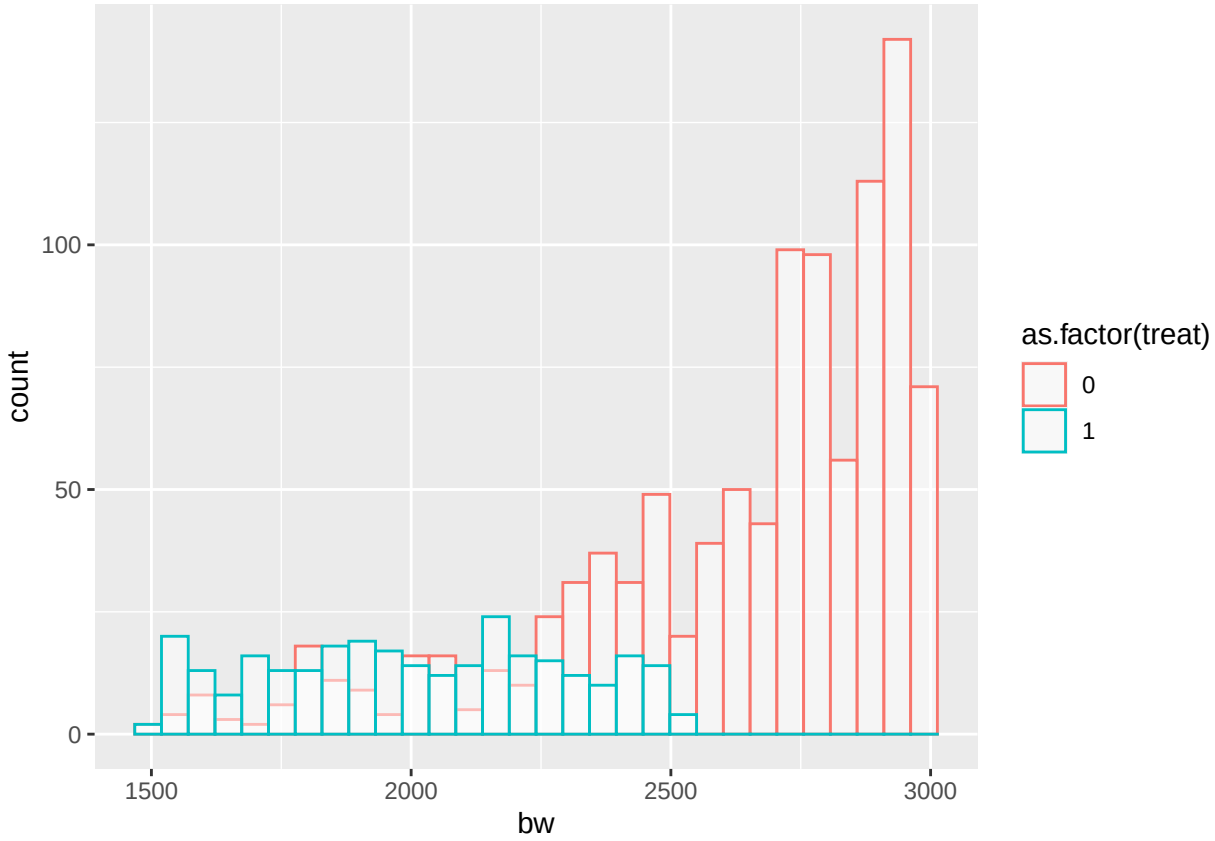

```
ggplot() +
  geom_histogram(data = df1, aes(x=momage,color=as.factor(treat)),fill="white",alpha=0.5,position="iden
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot() +
  geom_histogram(data = df1, aes(x=bw,color=as.factor(treat)),fill="white",alpha=0.5,position="identity
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

The plots above show clear areas where there is lack of overlap. This shows some areas of the treatment group lack common support with the control group. Differences in overlap cannot be adjusted through matching, and will limit the coverage of inference.

(c) Examining Balance. You will build your own function to check balance! This function should take as inputs (at least) 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 pre-match treatment group
2) Mean in the pre-match control group
3) Mean in the matched treatment group*
4) Mean in the matched control group
5) Pre-match 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 pre-match groups (control/treated)
8) Ratio of standard deviations across matched groups (control/treated)

I will provide a "unit test" of this function in a few days so you can help ensure that you are doing the right thing.

```
# Name function balfunc, with 4 inputs: data, treatment vector, weight vector, and covariate vector
balfunc <- function(df,z,weights,cov_names){
  # Define weight_vec as weights for only controll group
  weight_vec <- weights[z == 0]
  # Initialize results list
  results <- list()
```

```r
  # Begin loop through each covariate
  for (i in 1:length(cov_names)) {
    # Treated mean for raw data
    mn1     <- mean(df[z == 1, cov_names[i] ])
    # Control mean for raw data
    mn0     <- mean(df[z == 0, cov_names[i] ])
    # Treated mean for matched data
    mn1.m   <- mean(df[z == 1, cov_names[i] ])
    # Control mean for matched data
    mn0.m   <- weighted.mean(df[z == 0, cov_names[i] ], w = weight_vec)
    # Control variance for matched data
    var0.m  <- wtd.var(df[z == 0, cov_names[i] ], weights = weight_vec)
    # Control variance for raw data
    var0    <- var(df[z == 0, cov_names[i] ])
    # Treated variance for raw and matched data
    var1    <- var(df[z == 1, cov_names[i] ])
    # Logical for whether or not variable is binary
    if(length(unique(df[,cov_names[i] ])) > 2){
      # Difference in means for raw data
      diff      <- (mn1 - mn0)/sqrt(var1)
      # Difference in means for matched data
      diff.m    <- (mn1.m - mn0.m)/sqrt(var1)
      # Ratio of standard deviations across raw data
      ratio     <- sqrt(var0/var1)
      # Ratio of standard deviations across matched data
      ratio.m   <- sqrt(var0.m/var1)
    }
    else {
      # Difference in means for raw data if binary
      diff <- (mn1 - mn0)
      # Difference in means for matched data if binary
      diff.m <- (mn1.m - mn0.m)
      # No need for ratios for binary variables
      ratio <- 0
      ratio.m <- 0
    }
    # Combine results
     results[[i]] <- c(mn1, mn0, mn1.m, mn0.m, diff, diff.m, ratio, ratio.m)
  }
  # Turn results into data frame with new labels
  results <- data.frame(Reduce(rbind, results))
  names(results) <- c("mn1", "mn0", "mn1.m", "mn0.m", "diff", "diff.m", "ratio", "ratio.m")
  rownames(results) <- cov_names
  # Return data frame
  return(results)
}
```

(d) How do you interpret the resulting balance? In particular what are your concerns with regard to covariates that are not well balanced (3-4 sentences at most).

The larger the absolute values of the difference in means, the greater the imbalance between the control and treatment groups. In particularly, the variables with large imbalances could be indicative of systemic differences between the control and treatment groups. Not adjusting for these differences could cause the resulting interpretation to be biased and result in the ATT to be biased.

(e) Show the results of your balance function on a simple example 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.

```
w0 <- match0$cnts
b0 <- round(balfunc(df1,df1$treat,w0,cov_names0),digits = 3)
b0[c(9,2),]
```

```
##             mn1      mn0    mn1.m    mn0.m   diff diff.m ratio ratio.m
## bw     2008.648 2629.482 2008.648 2082.650 -2.191 -0.261 1.175   0.883
## b.marr    0.431    0.595    0.431    0.783 -0.164 -0.352 0.000   0.000
```

This is the unit test.

```
             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  0.000   0.000
```

Question 5: Repeat steps 2-4 within the matching framework. It is rare that your first specification of the propensity score model or choice of matching method is the best. Try at least *3* new approaches. Try to achieve better balance! For continuous variables strive for standardized mean differences less than .1. Try to get ratios of standard deviation closer to 1 than they are for the pre-match data (it may be difficult for some covariates to get the ratio close to 1). For binary variables strive for difference in means (equivalently difference in percentages) less than .05.

Ideas for trying something new in Step 2. You could try a new propensity score specification and then find the corresponding matched sample and calculate balance and overlap. For instance, you could change the inputs to the model (add quadratic terms, transformed versions of variables, or interactions, or delete predictors) or the model/algorithm used to estimate propensity scores (try probit or GAM or GBM or something else!). Alternately you could try a different matching method. A simple switch would be to switch from matching without replacement to matching with replacement. You could try k-1 matching or caliper matching or optimal matching though this will require using another package such as MatchIt. You could also try eliminating observations from the dataset. Importantly though if you eliminate observations from the group that we are trying to make inferences about you will need to profile those who have been removed. If you remove control observations from the comparison group (for instance those in states not represented by the IHDP observations) you do not need to do this.

Save your results (weights and balance) for reporting later.

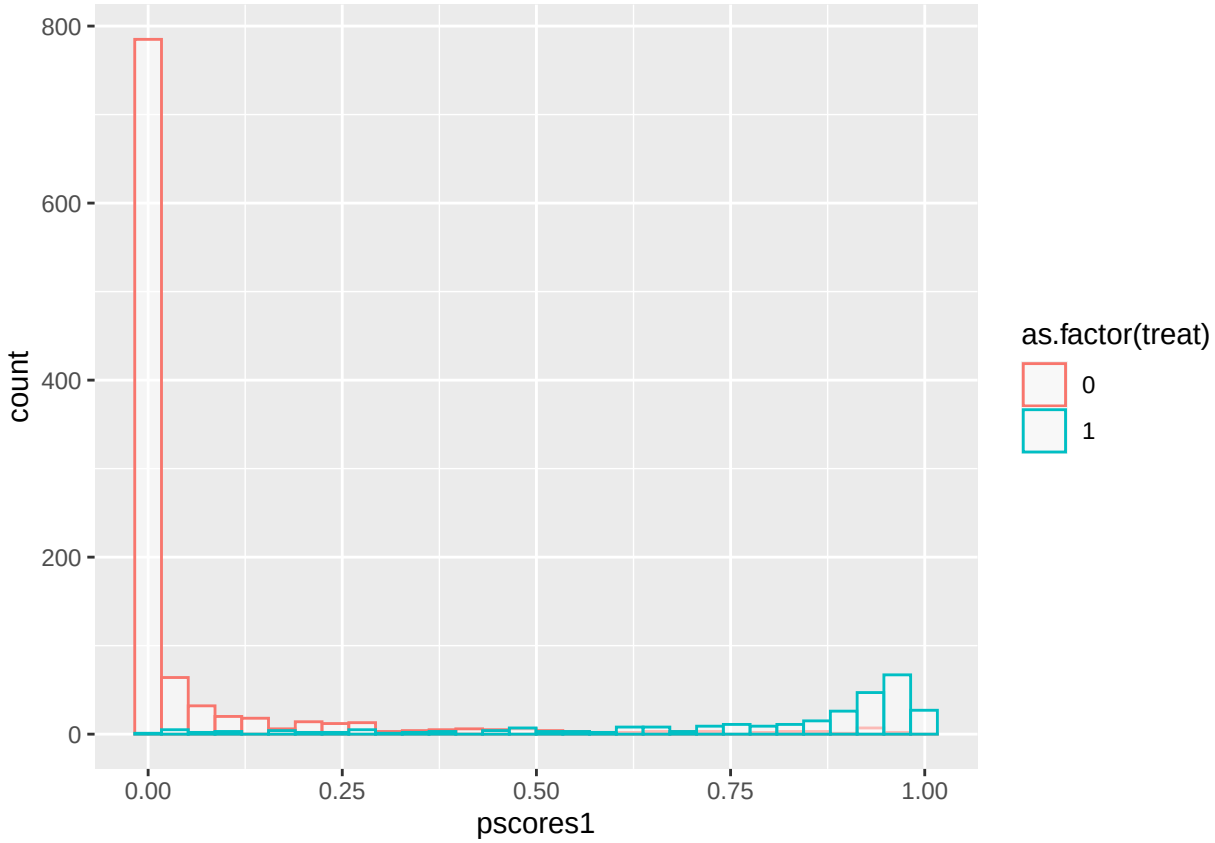
```
# Choose covariates - keeping all of them again
cov_names1 <- names(df1[3:length(df1)])
# Fit GLM model, removed some variables
glm1 <- glm(treat ~
              momage
            + b.marr
            + work.dur
            + prenatal
            + cig
            + booze
            + sex
            + first
            + poly(bw,2)
            + bwg
            + poly(preterm,2)
            + black
            + hispanic
            + white
```

```
          + lths
          + hs
          + ltcoll
          + college
          + dayskidh
          + income
           ,family=binomial(), data=df1)
# Use predict() to generate propensity scores
pscores1 <- predict(glm1, type="response")
# Match
match1 <- matching(df1$treat,pscores1,replace=TRUE)
w1 = match1$cnts
# Use balance function again, save results
b1 = round(balfunc(df1,df1$treat,w1,cov_names1),digits=3)
b1
```

```
##                mn1       mn0     mn1.m     mn0.m   diff diff.m ratio ratio.m
## momage      24.445    23.541    24.445    25.207  0.154 -0.130 0.552   0.546
## b.marr       0.431     0.595     0.431     0.414 -0.164  0.017 0.000   0.000
## work.dur     0.590     0.578     0.590     0.521  0.012  0.069 0.000   0.000
## prenatal     0.955     0.976     0.955     0.993 -0.021 -0.038 0.000   0.000
## cig          0.352     0.428     0.352     0.545 -0.076 -0.193 0.000   0.000
## booze        0.124     0.778     0.124     0.117 -0.654  0.007 0.000   0.000
## sex          0.507     0.544     0.507     0.772 -0.037 -0.266 0.000   0.000
## first        0.483     0.448     0.483     0.579  0.035 -0.097 0.000   0.000
## bw        2008.648  2629.482  2008.648  2033.636 -2.191 -0.088 1.175   0.942
## bwg          0.490     0.928     0.490     0.569 -0.439 -0.079 0.000   0.000
## preterm      6.072     2.406     6.072     5.452  1.908  0.323 1.295   0.833
## black        0.503     0.377     0.503     0.328  0.127  0.176 0.000   0.000
## hispanic     0.093     0.185     0.093     0.100 -0.092 -0.007 0.000   0.000
## white        0.403     0.438     0.403     0.572 -0.034 -0.169 0.000   0.000
## lths         0.434     0.341     0.434     0.283  0.094  0.152 0.000   0.000
## hs           0.283     0.422     0.283     0.521 -0.140 -0.238 0.000   0.000
## ltcoll       0.166     0.187     0.166     0.097 -0.022  0.069 0.000   0.000
## college      0.117     0.050     0.117     0.100  0.068  0.017 0.000   0.000
## dayskidh    14.686     6.021    14.686    13.812  0.768  0.078 0.794   1.705
## st5          0.138     0.016     0.138     0.014  0.122  0.124 0.000   0.000
## st9          0.134     0.021     0.134     0.034  0.113  0.100 0.000   0.000
## st12         0.100     0.054     0.100     0.041  0.046  0.059 0.000   0.000
## st25         0.114     0.015     0.114     0.000  0.099  0.114 0.000   0.000
## st36         0.117     0.041     0.117     0.028  0.076  0.090 0.000   0.000
## st42         0.145     0.039     0.145     0.024  0.106  0.121 0.000   0.000
## st48         0.114     0.071     0.114     0.007  0.043  0.107 0.000   0.000
## st53         0.138     0.011     0.138     0.028  0.127  0.110 0.000   0.000
## st99         0.000     0.733     0.000     0.824 -0.733 -0.824 0.000   0.000
## income   21347.394 27330.257 21347.394 13714.478 -0.287  0.366 3.822   0.886
```

```
# Check overlap of the raw data
ggplot() +
  geom_histogram(data = df1, aes(x=pscores1,color=as.factor(treat)),fill="white",alpha=0.5,position="id
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



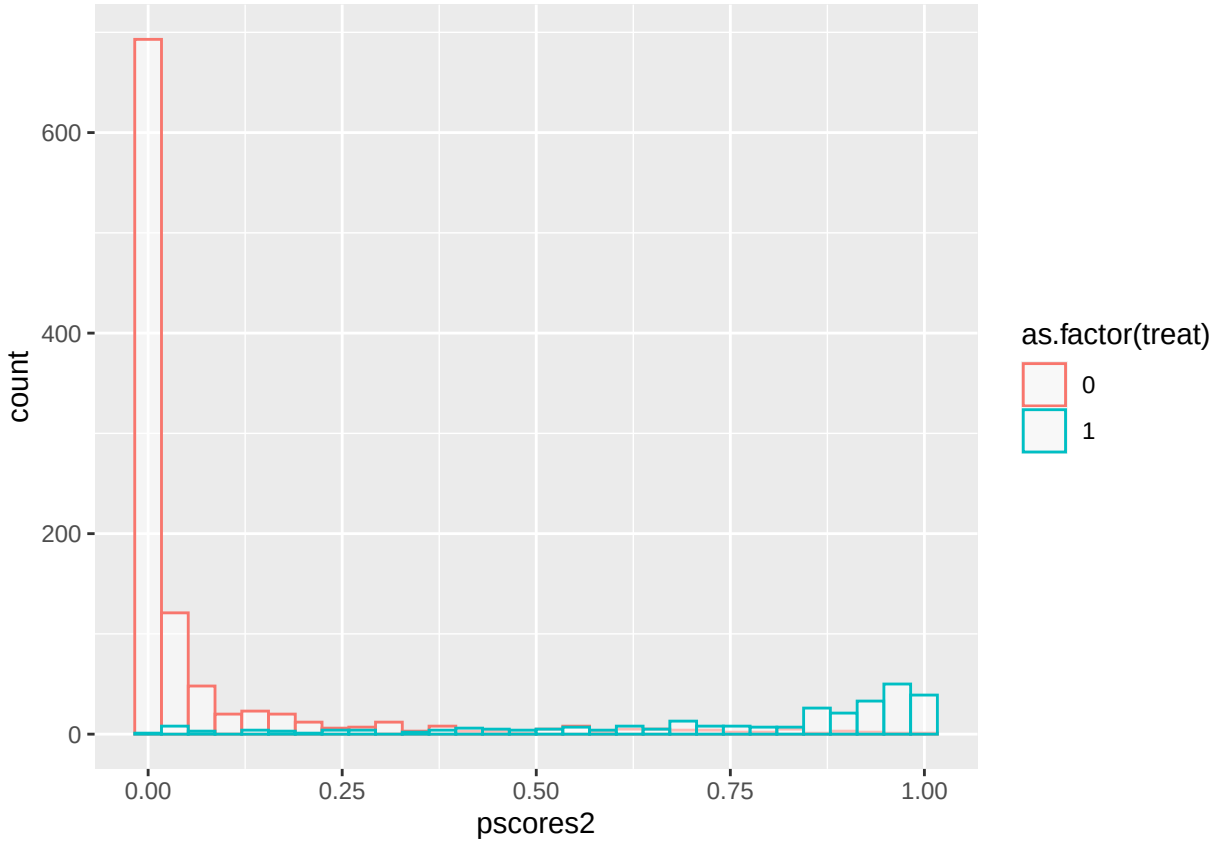

```
# Choose covariates - keeping all of them again
cov_names2 <- names(df1[3:length(df1)])
# Fit GLM model
glm2 <- glm(treat ~
              momage
            + b.marr
            + work.dur
            + prenatal
            + cig
            + booze
            + sex
            + first
            + bw
            + bwg
            + preterm
            + black
            + hispanic
            + white
            + lths
            + hs
            + ltcoll
            + college
            + dayskidh
            + income
             ,family=binomial(), data=df1)
# Use predict() to generate propensity scores
```

```r
pscores2 <- predict(glm2, type="response")
# Try matching without replacement - probably a bad idea
match2 <- matching(df1$treat,pscores2,replace=FALSE)
w2 = match2$cnts
# Use balance function again, save results
b2 = round(balfunc(df1,df1$treat,w2,cov_names2),digits=3)
b2
```

```
##                mn1       mn0     mn1.m     mn0.m   diff diff.m ratio ratio.m
## momage      24.445    23.541    24.445    23.914  0.154  0.090 0.552   0.561
## b.marr       0.431     0.595     0.431     0.497 -0.164 -0.066 0.000   0.000
## work.dur     0.590     0.578     0.590     0.603  0.012 -0.014 0.000   0.000
## prenatal     0.955     0.976     0.955     0.979 -0.021 -0.024 0.000   0.000
## cig          0.352     0.428     0.352     0.434 -0.076 -0.083 0.000   0.000
## booze        0.124     0.778     0.124     0.479 -0.654 -0.355 0.000   0.000
## sex          0.507     0.544     0.507     0.510 -0.037 -0.003 0.000   0.000
## first        0.483     0.448     0.483     0.407  0.035  0.076 0.000   0.000
## bw        2008.648  2629.482  2008.648  2319.280 -2.191 -1.096 1.175   1.350
## bwg          0.490     0.928     0.490     0.772 -0.439 -0.283 0.000   0.000
## preterm      6.072     2.406     6.072     4.241  1.908  0.953 1.295   1.505
## black        0.503     0.377     0.503     0.428  0.127  0.076 0.000   0.000
## hispanic     0.093     0.185     0.093     0.128 -0.092 -0.034 0.000   0.000
## white        0.403     0.438     0.403     0.445 -0.034 -0.041 0.000   0.000
## lths         0.434     0.341     0.434     0.397  0.094  0.038 0.000   0.000
## hs           0.283     0.422     0.283     0.369 -0.140 -0.086 0.000   0.000
## ltcoll       0.166     0.187     0.166     0.169 -0.022 -0.003 0.000   0.000
## college      0.117     0.050     0.117     0.066  0.068  0.052 0.000   0.000
## dayskidh    14.686     6.021    14.686     9.730  0.768  0.439 0.794   1.216
## st5          0.138     0.016     0.138     0.010  0.122  0.128 0.000   0.000
## st9          0.134     0.021     0.134     0.021  0.113  0.114 0.000   0.000
## st12         0.100     0.054     0.100     0.069  0.046  0.031 0.000   0.000
## st25         0.114     0.015     0.114     0.007  0.099  0.107 0.000   0.000
## st36         0.117     0.041     0.117     0.031  0.076  0.086 0.000   0.000
## st42         0.145     0.039     0.145     0.048  0.106  0.097 0.000   0.000
## st48         0.114     0.071     0.114     0.069  0.043  0.045 0.000   0.000
## st53         0.138     0.011     0.138     0.010  0.127  0.128 0.000   0.000
## st99         0.000     0.733     0.000     0.734 -0.733 -0.734 0.000   0.000
## income   21347.394 27330.257 21347.394 28163.394 -0.287 -0.327 3.822   4.501
```

```r
# Check overlap of the raw data
ggplot() +
  geom_histogram(data = df1, aes(x=pscores2,color=as.factor(treat)),fill="white",alpha=0.5,position="id
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Choose covariates - still keeping all
cov_names3 <- names(df1[3:length(df1)])
# Trying something new, using matchit()
pscores3 <- matchit(treat ~ bw + income + hs + lths + b.marr + work.dur + preterm + dayskidh + sex + his
```

```
## Loading required namespace: mgcv
```

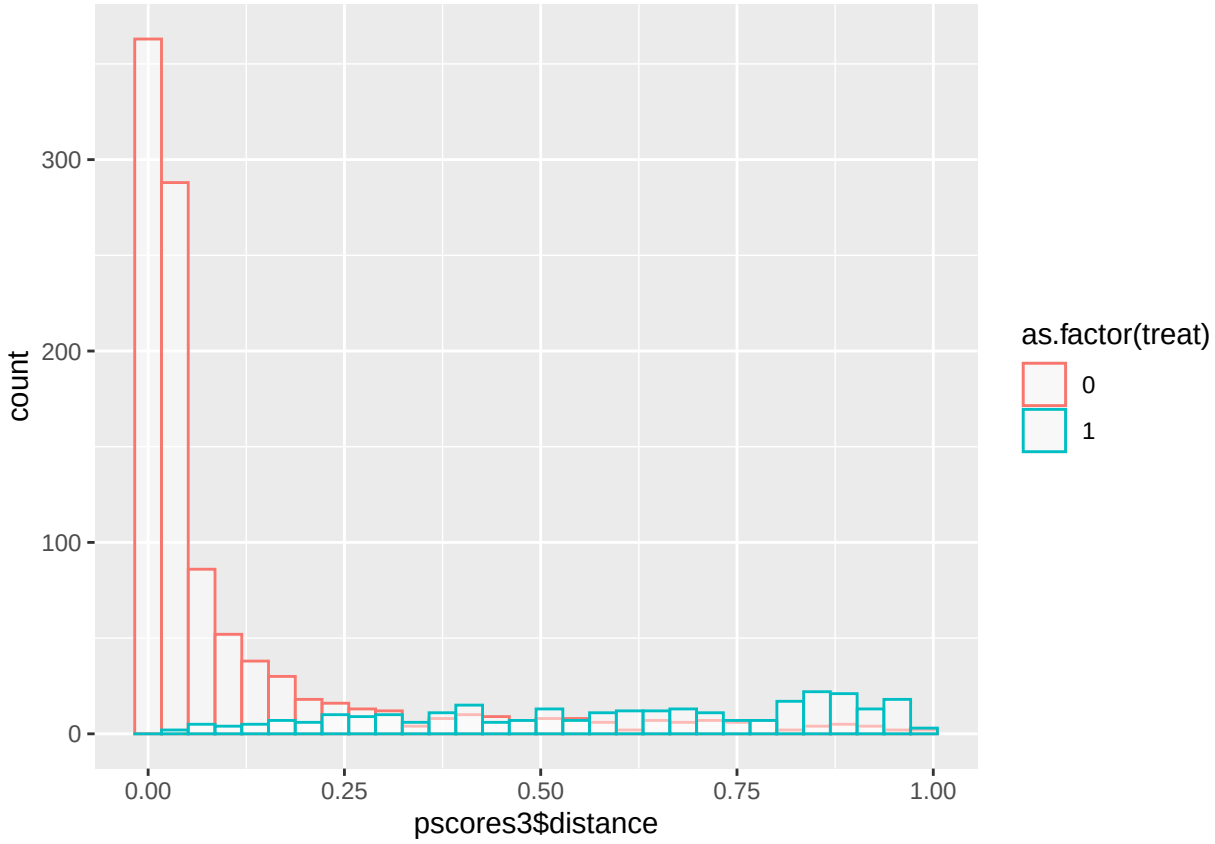
```
w3 = pscores3$weights
# Use balance function again, save results
b3 = round(balfunc(df1,df1$treat,w3,cov_names3),digits=3)
b3
```

```
##                mn1       mn0     mn1.m     mn0.m   diff diff.m ratio ratio.m
## momage      24.445    23.541    24.445    23.269  0.154  0.200 0.552   0.559
## b.marr       0.431     0.595     0.431     0.532 -0.164 -0.101 0.000   0.000
## work.dur     0.590     0.578     0.590     0.565  0.012  0.025 0.000   0.000
## prenatal     0.955     0.976     0.955     0.991 -0.021 -0.036 0.000   0.000
## cig          0.352     0.428     0.352     0.345 -0.076  0.007 0.000   0.000
## booze        0.124     0.778     0.124     0.819 -0.654 -0.694 0.000   0.000
## sex          0.507     0.544     0.507     0.500 -0.037  0.007 0.000   0.000
## first        0.483     0.448     0.483     0.377  0.035  0.106 0.000   0.000
## bw        2008.648  2629.482  2008.648  1987.289 -2.191  0.075 1.175   1.102
## bwg          0.490     0.928     0.490     0.437 -0.439  0.053 0.000   0.000
## preterm      6.072     2.406     6.072     6.301  1.908 -0.119 1.295   1.391
## black        0.503     0.377     0.503     0.395  0.127  0.108 0.000   0.000
## hispanic     0.093     0.185     0.093     0.140 -0.092 -0.047 0.000   0.000
## white        0.403     0.438     0.403     0.465 -0.034 -0.062 0.000   0.000
```

```
## lths          0.434     0.341     0.434     0.402  0.094  0.033 0.000   0.000
## hs            0.283     0.422     0.283     0.348 -0.140 -0.065 0.000   0.000
## ltcoll        0.166     0.187     0.166     0.205 -0.022 -0.039 0.000   0.000
## college       0.117     0.050     0.117     0.045  0.068  0.072 0.000   0.000
## dayskidh     14.686     6.021    14.686    14.843  0.768 -0.014 0.794   1.209
## st5           0.138     0.016     0.138     0.013  0.122  0.125 0.000   0.000
## st9           0.134     0.021     0.134     0.009  0.113  0.126 0.000   0.000
## st12          0.100     0.054     0.100     0.051  0.046  0.049 0.000   0.000
## st25          0.114     0.015     0.114     0.021  0.099  0.093 0.000   0.000
## st36          0.117     0.041     0.117     0.036  0.076  0.081 0.000   0.000
## st42          0.145     0.039     0.145     0.044  0.106  0.101 0.000   0.000
## st48          0.114     0.071     0.114     0.082  0.043  0.032 0.000   0.000
## st53          0.138     0.011     0.138     0.003  0.127  0.134 0.000   0.000
## st99          0.000     0.733     0.000     0.740 -0.733 -0.740 0.000   0.000
## income    21347.394 27330.257 21347.394 19015.808 -0.287  0.112 3.822   2.694
```

```
# Check overlap of the raw data
ggplot() +
  geom_histogram(data = df1, aes(x=pscores3$distance,color=as.factor(treat)),fill="white",alpha=0.5,pos
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Question 6: Repeat steps 2-4, but this time using IPTW. Save your results (weights and balance) – do not display them here. Make sure that you use weights specific to the effect of the treatment on the treated. In this section simply include your code for estimating the pscores and your code for creating the IPTW weights.

```r
# Choose covariates - keeping all of them again
cov_names4 <- names(df1[3:length(df1)])
# Fit GLM model, removed more variables
glm4 <- glm(treat ~ .-1, family = "binomial", data = df1)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
# Predict for propensity scores
pscores4 <- predict(glm4, type="response")
# Inverse for ATT
iptw <- tibble(pscores = pscores4, treat = df1$treat)
iptw <- iptw %>% mutate(wgt = if_else(treat==1,1,pscores4/(1-pscores4)))
w4 <- iptw$wgt
b4 <- round(balfunc(df1,df1$treat,w4,cov_names0),digits = 3)
```

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)

```r
# Original fitting
b0[c(6,8)]
```

```
##          diff.m ratio.m
## momage   -0.585   0.580
## b.marr   -0.352   0.000
## work.dur -0.183   0.000
## prenatal -0.045   0.000
## cig       0.224   0.000
## booze    -0.076   0.000
## sex      -0.317   0.000
## first    -0.252   0.000
## bw       -0.261   0.883
## bwg      -0.276   0.000
## preterm   1.193   1.058
## black     0.293   0.000
## hispanic -0.007   0.000
## white    -0.286   0.000
## lths      0.203   0.000
## hs       -0.445   0.000
## ltcoll    0.128   0.000
## college   0.114   0.000
## dayskidh  0.665   0.897
## st5       0.045   0.000
## st9       0.093   0.000
## st12      0.045   0.000
## st25      0.086   0.000
## st36      0.007   0.000
## st42     -0.493   0.000
## st48      0.083   0.000
## st53      0.134   0.000
## st99      0.000   0.000
## income   -0.392   0.725
```

```
# Method 1: removing st variables, squaring bw and preterm, with replacement
b1[c(6,8)]
```

```
##          diff.m ratio.m
## momage   -0.130   0.546
## b.marr    0.017   0.000
## work.dur  0.069   0.000
## prenatal -0.038   0.000
## cig      -0.193   0.000
## booze     0.007   0.000
## sex      -0.266   0.000
## first    -0.097   0.000
## bw       -0.088   0.942
## bwg      -0.079   0.000
## preterm   0.323   0.833
## black     0.176   0.000
## hispanic -0.007   0.000
## white    -0.169   0.000
## lths      0.152   0.000
## hs       -0.238   0.000
## ltcoll    0.069   0.000
## college   0.017   0.000
## dayskidh  0.078   1.705
## st5       0.124   0.000
## st9       0.100   0.000
## st12      0.059   0.000
## st25      0.114   0.000
## st36      0.090   0.000
## st42      0.121   0.000
## st48      0.107   0.000
## st53      0.110   0.000
## st99     -0.824   0.000
## income    0.366   0.886
```

```
# Method 2: removing st variables, squaring bw and preterm, without replacement
b2[c(6,8)]
```

```
##          diff.m ratio.m
## momage    0.090   0.561
## b.marr   -0.066   0.000
## work.dur -0.014   0.000
## prenatal -0.024   0.000
## cig      -0.083   0.000
## booze    -0.355   0.000
## sex      -0.003   0.000
## first     0.076   0.000
## bw       -1.096   1.350
## bwg      -0.283   0.000
## preterm   0.953   1.505
## black     0.076   0.000
## hispanic -0.034   0.000
## white    -0.041   0.000
## lths      0.038   0.000
## hs       -0.086   0.000
## ltcoll   -0.003   0.000
```

```
## college   0.052   0.000
## dayskidh  0.439   1.216
## st5       0.128   0.000
## st9       0.114   0.000
## st12      0.031   0.000
## st25      0.107   0.000
## st36      0.086   0.000
## st42      0.097   0.000
## st48      0.045   0.000
## st53      0.128   0.000
## st99     -0.734   0.000
## income   -0.327   4.501
```

```
# Method 3: using MatchIt
b3[c(6,8)]
```

```
##          diff.m ratio.m
## momage    0.200   0.559
## b.marr   -0.101   0.000
## work.dur  0.025   0.000
## prenatal -0.036   0.000
## cig       0.007   0.000
## booze    -0.694   0.000
## sex       0.007   0.000
## first     0.106   0.000
## bw        0.075   1.102
## bwg       0.053   0.000
## preterm  -0.119   1.391
## black     0.108   0.000
## hispanic -0.047   0.000
## white    -0.062   0.000
## lths      0.033   0.000
## hs       -0.065   0.000
## ltcoll   -0.039   0.000
## college   0.072   0.000
## dayskidh -0.014   1.209
## st5       0.125   0.000
## st9       0.126   0.000
## st12      0.049   0.000
## st25      0.093   0.000
## st36      0.081   0.000
## st42      0.101   0.000
## st48      0.032   0.000
## st53      0.134   0.000
## st99     -0.740   0.000
## income    0.112   2.694
```

```
# IPTW
b4[c(6,8)]
```

```
##          diff.m ratio.m
## momage   -0.161   0.615
## b.marr   -0.160   0.000
## work.dur  0.024   0.000
## prenatal -0.045   0.000
## cig       0.166   0.000
```

```
## booze    -0.236   0.000
## sex       0.011   0.000
## first     0.116   0.000
## bw        0.115   1.096
## bwg       0.025   0.000
## preterm   0.535   1.298
## black     0.162   0.000
## hispanic -0.227   0.000
## white     0.065   0.000
## lths     -0.106   0.000
## hs       -0.120   0.000
## ltcoll    0.117   0.000
## college   0.109   0.000
## dayskidh  0.334   1.126
## st5       0.024   0.000
## st9      -0.078   0.000
## st12      0.023   0.000
## st25      0.058   0.000
## st36     -0.111   0.000
## st42     -0.095   0.000
## st48      0.058   0.000
## st53      0.121   0.000
## st99      0.000   0.000
## income   -0.257   0.948
```

I would choose the approach using built in GLM, balance and weight data b1 and w1, dropping all st variables and squaring the bw and preterm variables. This approach provides good balance without throwing any one variable completely off balance. This approach is also preferred, as overfitting the model would misrepresent the counterfactual. Also, because of the relatively smaller number of treated vs. control, using the matching WITHOUT replacement would reduce the overall sample size and introduce bias into the estimate.

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 datasets 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.

```
# Original fitting
summary(lm(ppvtr.36 ~ ., data = df1, weights = w0))$coefficients[2,]
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##  0.2708564  2.2229636  0.1218447  0.9031061
```

```
# Method 1: removing st variables, squaring bw and preterm, with replacement
summary(lm(ppvtr.36 ~ ., data = df1, weights = w1))$coefficients[2,]
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##  5.2580544  3.6051698  1.4584762  0.1456633
```

```
# Method 2: removing st variables, squaring bw and preterm, without replacement
summary(lm(ppvtr.36 ~ ., data = df1, weights = w2))$coefficients[2,]
```

```
##    Estimate  Std. Error     t value    Pr(>|t|)
## 7.810996653 2.562329244 3.048396950 0.002410863
```

```
# Method 3: using MatchIt
summary(lm(ppvtr.36 ~ ., data = df1, weights = w3))$coefficients[2,]
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
```

```
##  3.4534532  2.9663216  1.1642208  0.2448942
```

```
# IPTW
summary(lm(ppvtr.36 ~ ., data = df1, weights = w4))$coefficients[2,]
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
## -1.9818472  1.0751456 -1.8433290  0.0655099
```

Question 9: Assumptions What assumptions are necessary to interpret the estimates from the propensity score approaches causally?

The primary assumption required for interpreting estimates from propensity score matching is ignoribility. If this assumption is not satisfied, then p-scores will be mispecified and the resulting model will misrepresent the counterfactual. SUFTA also should be satisfied, as spillover effects could effect treatment estimate.

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 about. Also make sure to use causal language.

The estimate of the treatment effect on the treated from data created through matching with replacement is a 5.26 point increase. The causal interpretation would be that children who participated in the IHDP intervention had test scores that were, on average, 5.26 points higher than what they would have been 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.

```
# Simple linear regression
round(summary(lm(ppvtr.36 ~ .,data=df1))$coefficient[2,],5)
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##   10.03743    1.97514    5.08189    0.00000
```

(b) Interpret your results non-causally.

Holding all other covariates constant, a group of children given treatment (enrolled in IHDP) can be expected, on average, have an IQ test score that is 10.03 points higher than a different group of children who were not given treatment.

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

The propensity score approach considers a counterfactual and allows one to make a statement about the IQ test scores of specific individuals in the data. In contrast, linear regression only allows the comparison of averages across different groups. Propensity scores allow making statements about individuals and what would have happened had they not received treatment. Linear regression forces the comparison of different groups of subjects within the data and does not allow inferences about individuals or their respective counterfactuals.

Challenge Question: Improve the standard errors. We know the standard errors for the regression in question 11 aren't quite right because they are acting as if the observations are iid. Find a way to fit this regression that appropriate adjusts the standard errors for the weights used.

```
# GLM function in Survey package should account for the weights used
library(survey)
```

```
## Loading required package: grid
```

```
##
## Attaching package: 'survey'
```

```
## The following object is masked from 'package:Hmisc':
##
##     deff
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
# Creating design with weights in vector w1
design <- svydesign(ids = ~1, weights = w1, data = df1)
# Running regression with all variables, using the weights from w1
reg <- svyglm(ppvtr.36 ~ treat + momage + b.marr + work.dur + prenatal + cig + booze + sex + first + bw
```

```
## Warning in summary.glm(g): observations with zero weight not used for
## calculating dispersion
```

```
## Warning in summary.glm(glm.object): observations with zero weight not used for
## calculating dispersion
```

```
# Estimate, std error and p value of the treatment estimate
summary(reg)$coefficients[2,]
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##  5.2580544  4.8356151  1.0873600  0.2776737
```