Practice Problems in R

Peter Sun

September, 2021

# Practice 1

## Problem 1: Generalized Boosted Regression

### Load Packages and Data

library(tidyverse)  
library(haven)  
library(sjlabelled)  
library(lmtest)  
library(gbm)  
library(modelr)  
library(broom)  
library(flextable)  
library(sandwich)  
library(cobalt)  
library(WeightIt)  
library(Matching)  
select <- dplyr::select  
knitr::opts\_chunk$set(dpi = 300, fig.width = 7)  
  
d <- read\_dta("data/ldw\_exper.dta") %>%  
 haven::zap\_formats() %>%  
 sjlabelled::remove\_all\_labels() %>%  
 as\_tibble()

### Sort Data

set.seed(1000)  
d2 <- d %>%  
 add\_column(runif = runif(nrow(.))) %>%  
 arrange(runif)

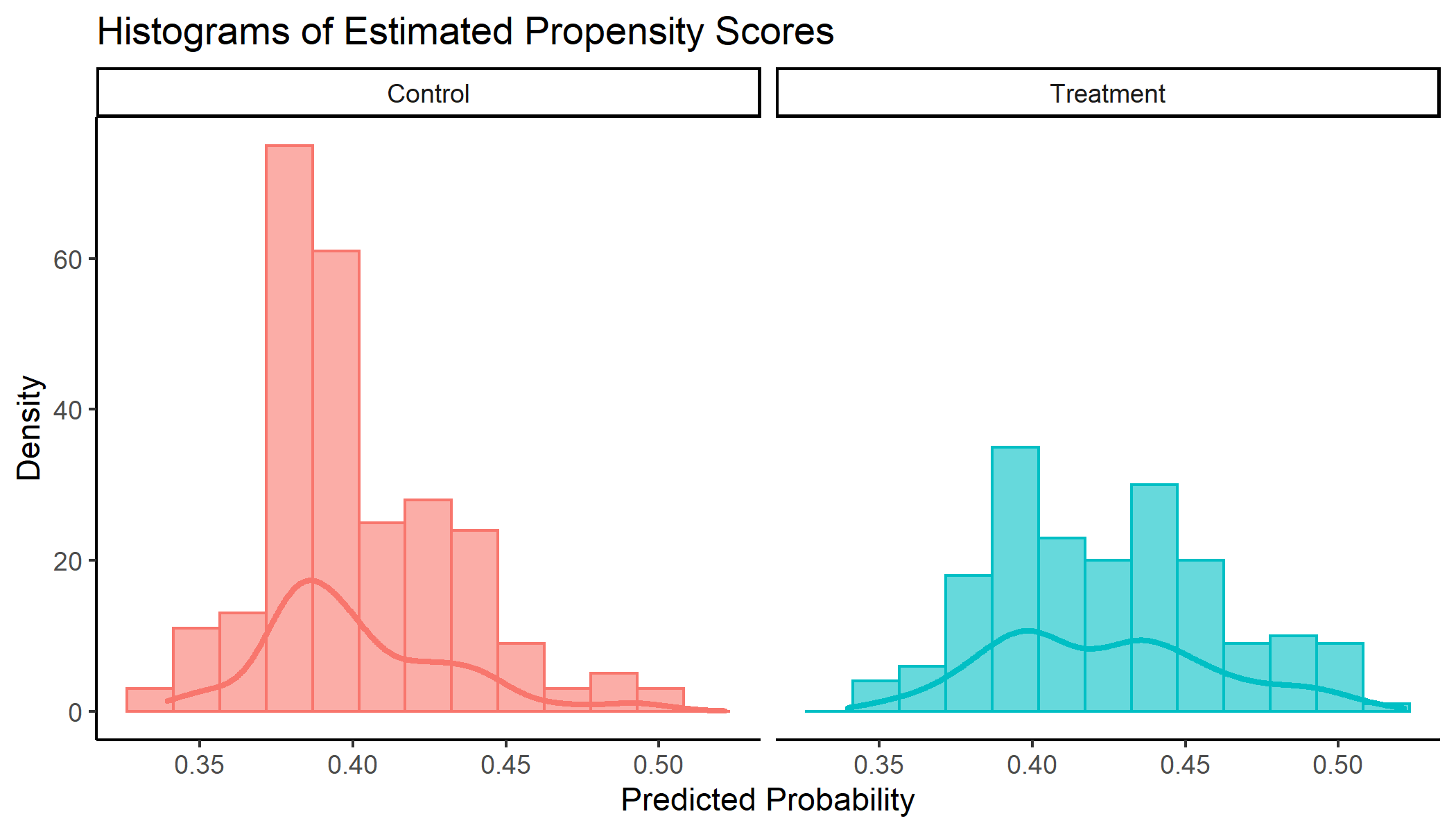
### Generate Propensity Scores

set.seed(1000)  
m1 <- gbm::gbm(formula = t ~ age + educ + black + hisp + married + re74 +  
 re75 + u74 + u75,  
 data = d2,  
 distribution = "bernoulli",  
 n.trees = 1000,  
 train.fraction = 0.8,  
 interaction.depth = 4,  
 shrinkage = 0.0005)  
  
# Add Predictions to Data  
d3 <- d2 %>%  
 modelr::add\_predictions(m1, var = "psb", type = "response")  
  
# Summary of Propensity Scores  
summary(d3$psb)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3394 0.3855 0.4006 0.4108 0.4346 0.5214

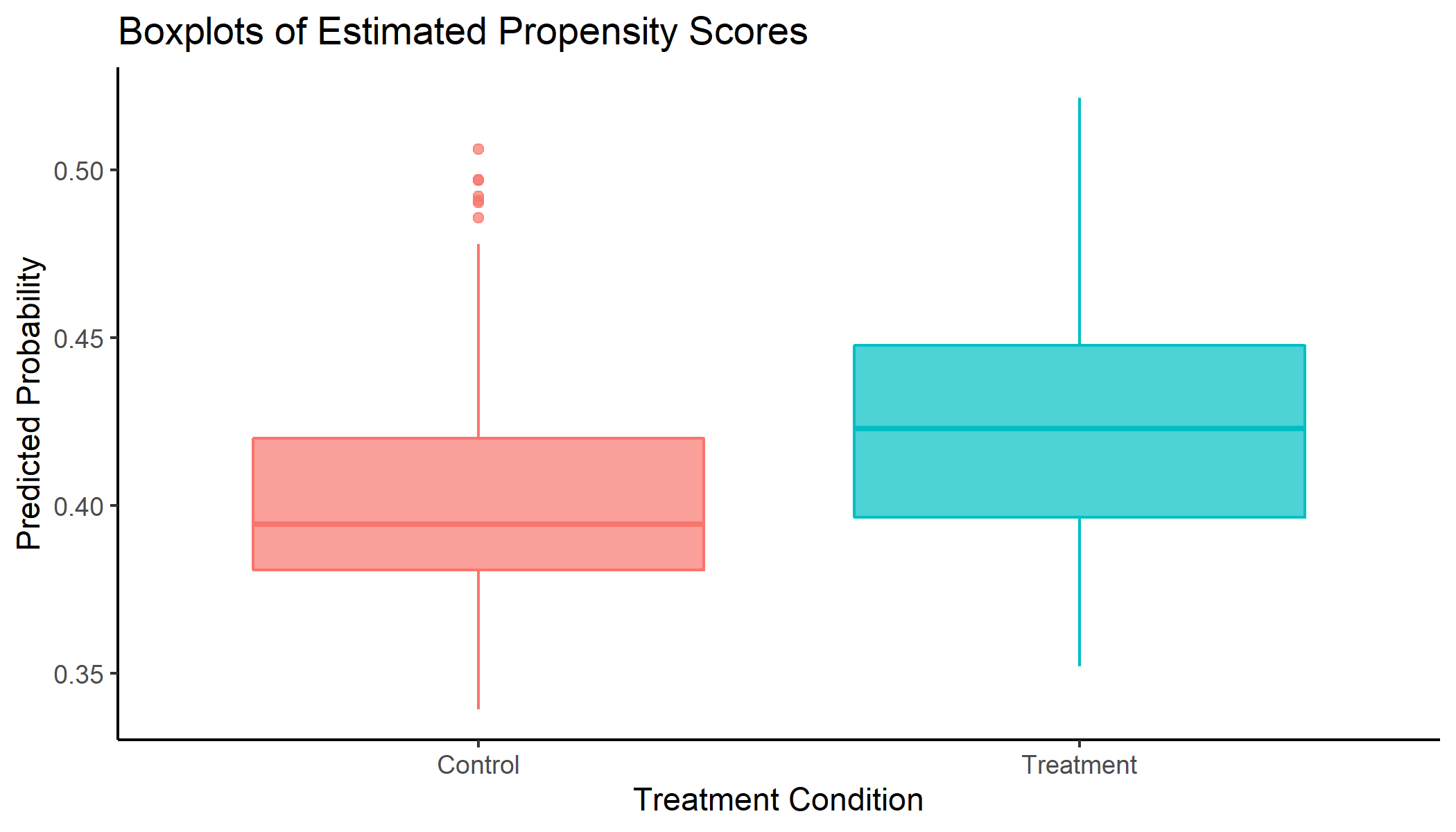
### Plot Histogram

d3 %>%  
 mutate(t = factor(t, labels = c("Control", "Treatment"))) %>%  
 ggplot(aes(x = psb, color = t)) +   
 theme\_classic() +  
 geom\_histogram(aes(fill = t), alpha = 0.6, bins = 13) +  
 geom\_density(size = 1) +   
 labs(x = "Predicted Probability", y = "Density",  
 title = "Histograms of Estimated Propensity Scores") +  
 theme(legend.position = "none") +  
 facet\_wrap(~ t)



### Plot Boxplot

d3 %>%  
 mutate(t = factor(t, labels = c("Control", "Treatment"))) %>%  
 ggplot(aes(x = t, y = psb, color = t, fill = t)) +   
 theme\_classic() +  
 geom\_boxplot(alpha = 0.7) +  
 labs(x = "Treatment Condition",  
 y = "Predicted Probability",  
 title = "Boxplots of Estimated Propensity Scores") +  
 theme(legend.position = "none")



## Problem 2: Propensity Score Weighting

### Create ATE and ATT Weights

d4 <- d3 %>%  
 mutate(ate\_w = ifelse(t == 0, 1/(1-psb), 1/psb),  
 att\_w = ifelse(t == 0, psb/(1-psb), 1))  
  
# Import Stata-generated weights to replicate results  
stata\_weights <- read\_dta("data/ldw1.dta") %>%  
 haven::zap\_formats() %>%  
 sjlabelled::remove\_all\_labels() %>%  
 as\_tibble() %>%  
 mutate(stata\_ate\_w = ifelse(t == 0, 1/(1-psb), 1/psb),  
 stata\_att\_w = ifelse(t == 0, psb/(1-psb), 1)) %>%  
 select(id, stata\_ate\_w, stata\_att\_w)  
d5 <- d4 %>%  
 arrange(id) %>%  
 left\_join(stata\_weights, by = "id")

### PSW With ATE and ATT Weights

# Define outcome formula  
f = as.formula(re78 ~ t + age + educ + black + hisp + married + re74 + re75 +   
 u74 + u75)

#### R-Generated Propensity Scores

m2 <- lm(f, data = d5, weights = ate\_w)  
lmtest::coeftest(m2, vcov. = vcovHC(m2, "HC1")) # ATE

##   
## t test of coefficients:  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.7800472 2.9708349 -0.2626 0.793008   
## t 1.6486269 0.6551563 2.5164 0.012216 \*   
## age 0.0645287 0.0424837 1.5189 0.129515   
## educ 0.4413544 0.1694931 2.6040 0.009531 \*\*  
## black -1.9904522 1.0830659 -1.8378 0.066776 .   
## hisp 0.5264386 1.5647002 0.3364 0.736696   
## married -0.0775809 0.8957555 -0.0866 0.931022   
## re74 0.1443565 0.1390312 1.0383 0.299707   
## re75 0.0092717 0.1355125 0.0684 0.945483   
## u74 1.9875514 1.6760984 1.1858 0.236342   
## u75 -1.4102696 1.5517788 -0.9088 0.363956   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

m3 <- lm(f, data = d5, weights = att\_w)   
lmtest::coeftest(m3, vcov. = vcovHC(m3, "HC1")) # ATT

##   
## t test of coefficients:  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.100183 3.027042 -0.3635 0.716445   
## t 1.723694 0.663051 2.5996 0.009650 \*\*  
## age 0.061982 0.042685 1.4521 0.147207   
## educ 0.453051 0.172495 2.6265 0.008933 \*\*  
## black -1.924978 1.100434 -1.7493 0.080948 .   
## hisp 0.545830 1.571080 0.3474 0.728442   
## married -0.045077 0.915577 -0.0492 0.960756   
## re74 0.157119 0.144833 1.0848 0.278597   
## re75 0.011515 0.135862 0.0848 0.932494   
## u74 2.165300 1.704711 1.2702 0.204699   
## u75 -1.404330 1.570928 -0.8939 0.371845   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#### Stata-Generated Propensity Scores (Identical Results)

m2.stata <- lm(f, data = d5, weights = stata\_ate\_w)   
lmtest::coeftest(m2.stata, vcov. = vcovHC(m2.stata, "HC1")) # ATE

##   
## t test of coefficients:  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.465769 2.910605 -0.1600 0.872936   
## t 1.626429 0.655889 2.4797 0.013527 \*   
## age 0.062082 0.042087 1.4751 0.140915   
## educ 0.429798 0.165495 2.5970 0.009722 \*\*  
## black -2.012326 1.075461 -1.8711 0.062000 .   
## hisp 0.504625 1.528379 0.3302 0.741431   
## married -0.084147 0.891903 -0.0943 0.924878   
## re74 0.133469 0.131633 1.0139 0.311174   
## re75 0.011185 0.137145 0.0816 0.935038   
## u74 1.888649 1.658767 1.1386 0.255504   
## u75 -1.424604 1.550877 -0.9186 0.358825   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

m3.stata <- lm(f, data = d5, weights = stata\_att\_w)   
lmtest::coeftest(m3.stata, vcov. = vcovHC(m3.stata, "HC1")) # ATT

##   
## t test of coefficients:  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.879920 3.004629 -0.2929 0.769773   
## t 1.696293 0.665212 2.5500 0.011115 \*   
## age 0.059006 0.042607 1.3849 0.166800   
## educ 0.445827 0.170320 2.6176 0.009165 \*\*  
## black -1.904275 1.087680 -1.7508 0.080692 .   
## hisp 0.541277 1.533801 0.3529 0.724335   
## married -0.034142 0.914053 -0.0374 0.970222   
## re74 0.150765 0.143163 1.0531 0.292881   
## re75 0.013113 0.137456 0.0954 0.924044   
## u74 2.103916 1.701726 1.2363 0.217000   
## u75 -1.414714 1.565945 -0.9034 0.366801   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### Check Imbalance

Note that a custom function by Jorge Cimentada is used to replicate the robust standard errors in Stata.[[1]](#footnote-32)

# Replicate Stata robust standard errors  
robustse <- function(x, coef = c("logit", "odd.ratio", "probs")) {  
 suppressMessages(suppressWarnings(library(lmtest)))  
 suppressMessages(suppressWarnings(library(sandwich)))  
  
 sandwich1 <- function(object, ...) {  
 sandwich(object) \*  
 nobs(object) / (nobs(object) - 1)  
 }  
 # Function calculates SE's  
 mod1 <- coeftest(x, vcov = sandwich1)  
 # apply the function over the variance-covariance matrix  
  
 if (coef == "logit") {  
 return(mod1) # return logit with robust SE's  
 } else if (coef == "odd.ratio") {  
 mod1[, 1] <- exp(mod1[, 1]) # return odd ratios with robust SE's  
 mod1[, 2] <- mod1[, 1] \* mod1[, 2]  
 return(mod1)  
 } else {  
 mod1[, 1] <- (mod1[, 1] / 4) # return probabilites with robust SE's  
 mod1[, 2] <- mod1[, 2] / 4  
 return(mod1)  
 }  
}  
  
# Function to Check Imbalance  
check\_bal <- function(var, weight, type) {  
 if(type == "categorical") {  
 m <- glm(as.formula(paste0(var, "~t")),  
 family = quasibinomial,  
 data = d5,  
 weights = weight  
 )  
 m %>%  
 tidy() %>%  
 mutate(odds.ratio = exp(estimate), variable = var) %>%  
 mutate(or.se = robustse(m, coef = "odd.ratio")[,2]) %>%  
 mutate(statistic = robustse(m, coef = "odd.ratio")[,3]) %>%  
 mutate(p.value = robustse(m, coef = "odd.ratio")[,4]) %>%  
 select(variable, term, odds.ratio, or.se, statistic, p.value)  
 } else if(type == "continuous") {  
 m <- lm(as.formula(paste0(var, "~t")),  
 data = d5,  
 weights = weight)  
 lmtest::coeftest(m, vcov. = vcovHC(m, "HC1")) %>%  
 tidy() %>%  
 add\_column(var, .before = "term")  
 }  
}  
format\_bal <- function(df) {  
 df %>%  
 flextable() %>%  
 flextable::set\_table\_properties(width = 1, layout = "autofit") %>%  
 flextable::colformat\_double(digits = 7)  
}  
  
# Categorical Variables  
cat\_vars <- c("black", "hisp", "married", "u74", "u75")  
format\_bal(map\_dfr(cat\_vars, check\_bal, d5$stata\_ate\_w, "categorical"))

| variable | term | odds.ratio | or.se | statistic | p.value |
| --- | --- | --- | --- | --- | --- |
| black | (Intercept) | 4.8628405 | 0.7986511 | 9.6302118 | 0.0000000 |
| black | t | 1.0997119 | 0.2872935 | 0.3638291 | 0.7159856 |
| hisp | (Intercept) | 0.1167194 | 0.0233856 | -10.7207574 | 0.0000000 |
| hisp | t | 0.5680117 | 0.2106900 | -1.5248703 | 0.1272914 |
| married | (Intercept) | 0.1838332 | 0.0316686 | -9.8319094 | 0.0000000 |
| married | t | 1.2388627 | 0.3163250 | 0.8388736 | 0.4015402 |
| u74 | (Intercept) | 2.9214670 | 0.4193075 | 7.4696105 | 0.0000000 |
| u74 | t | 0.8907275 | 0.1929677 | -0.5341418 | 0.5932434 |
| u75 | (Intercept) | 2.0816176 | 0.2783919 | 5.4819414 | 0.0000000 |
| u75 | t | 0.7896914 | 0.1590241 | -1.1725044 | 0.2409946 |

format\_bal(map\_dfr(cat\_vars, check\_bal, d5$stata\_att\_w, "categorical"))

| variable | term | odds.ratio | or.se | statistic | p.value |
| --- | --- | --- | --- | --- | --- |
| black | (Intercept) | 4.9877646 | 0.8221686 | 9.7489458 | 0.0000000 |
| black | t | 1.0785012 | 0.2815619 | 0.2894740 | 0.7722186 |
| hisp | (Intercept) | 0.1111411 | 0.0223087 | -10.9451279 | 0.0000000 |
| hisp | t | 0.5688118 | 0.2106633 | -1.5234109 | 0.1276559 |
| married | (Intercept) | 0.1867004 | 0.0323266 | -9.6926309 | 0.0000000 |
| married | t | 1.2497739 | 0.3193590 | 0.8725380 | 0.3829149 |
| u74 | (Intercept) | 2.8153157 | 0.4059147 | 7.1789995 | 0.0000000 |
| u74 | t | 0.8616888 | 0.1868079 | -0.6866516 | 0.4923023 |
| u75 | (Intercept) | 1.9635837 | 0.2636932 | 5.0246644 | 0.0000005 |
| u75 | t | 0.7639094 | 0.1539378 | -1.3364198 | 0.1814121 |

# Continuous Variables  
cont\_vars <- c("age", "educ", "re74", "re75")  
format\_bal(map\_dfr(cont\_vars, check\_bal, d5$stata\_ate\_w, "continuous"))

| var | term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- | --- |
| age | (Intercept) | 25.1403511 | 0.4440221 | 56.6195931 | 0.0000000 |
| age | t | 0.5085618 | 0.6801859 | 0.7476806 | 0.4550495 |
| educ | (Intercept) | 10.1174367 | 0.1020983 | 99.0950541 | 0.0000000 |
| educ | t | 0.1629619 | 0.1762747 | 0.9244774 | 0.3557411 |
| re74 | (Intercept) | 2.1421666 | 0.3557791 | 6.0210583 | 0.0000000 |
| re74 | t | -0.1742469 | 0.4893238 | -0.3560973 | 0.7219372 |
| re75 | (Intercept) | 1.3007971 | 0.1948838 | 6.6747307 | 0.0000000 |
| re75 | t | 0.1288075 | 0.2977874 | 0.4325485 | 0.6655533 |

format\_bal(map\_dfr(cont\_vars, check\_bal, d5$stata\_att\_w, "continuous"))

| var | term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- | --- |
| age | (Intercept) | 25.2629379 | 0.4537770 | 55.6725839 | 0.0000000 |
| age | t | 0.5532783 | 0.6945403 | 0.7966107 | 0.4261038 |
| educ | (Intercept) | 10.1584976 | 0.1051669 | 96.5940992 | 0.0000000 |
| educ | t | 0.1874483 | 0.1813631 | 1.0335526 | 0.3019093 |
| re74 | (Intercept) | 2.1919636 | 0.3609099 | 6.0734365 | 0.0000000 |
| re74 | t | -0.0963895 | 0.5091306 | -0.1893218 | 0.8499273 |
| re75 | (Intercept) | 1.3488199 | 0.1985179 | 6.7944480 | 0.0000000 |
| re75 | t | 0.1832357 | 0.3088324 | 0.5933178 | 0.5532713 |

Similar results can be obtained using the R-generated propensity score weights:

# With R-generated weights  
format\_bal(map\_dfr(cat\_vars, check\_bal, d5$ate\_w, "categorical"))  
format\_bal(map\_dfr(cat\_vars, check\_bal, d5$att\_w, "categorical"))  
format\_bal(map\_dfr(cont\_vars, check\_bal, d5$ate\_w, "continuous"))  
format\_bal(map\_dfr(cont\_vars, check\_bal, d5$att\_w, "continuous"))

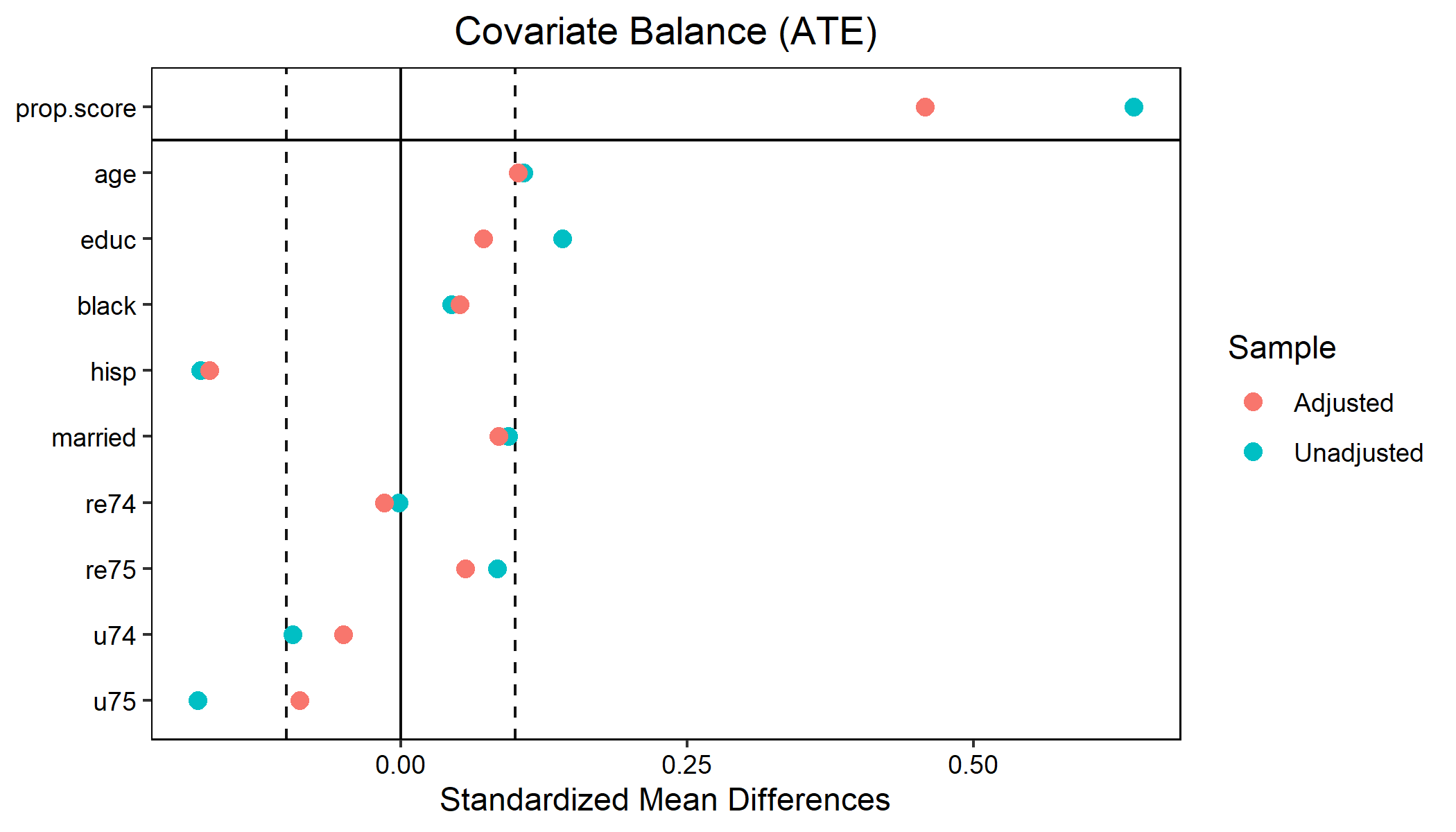
### Alternative Solution with WeightIt

#### Use GBM to Generate ATE and ATT

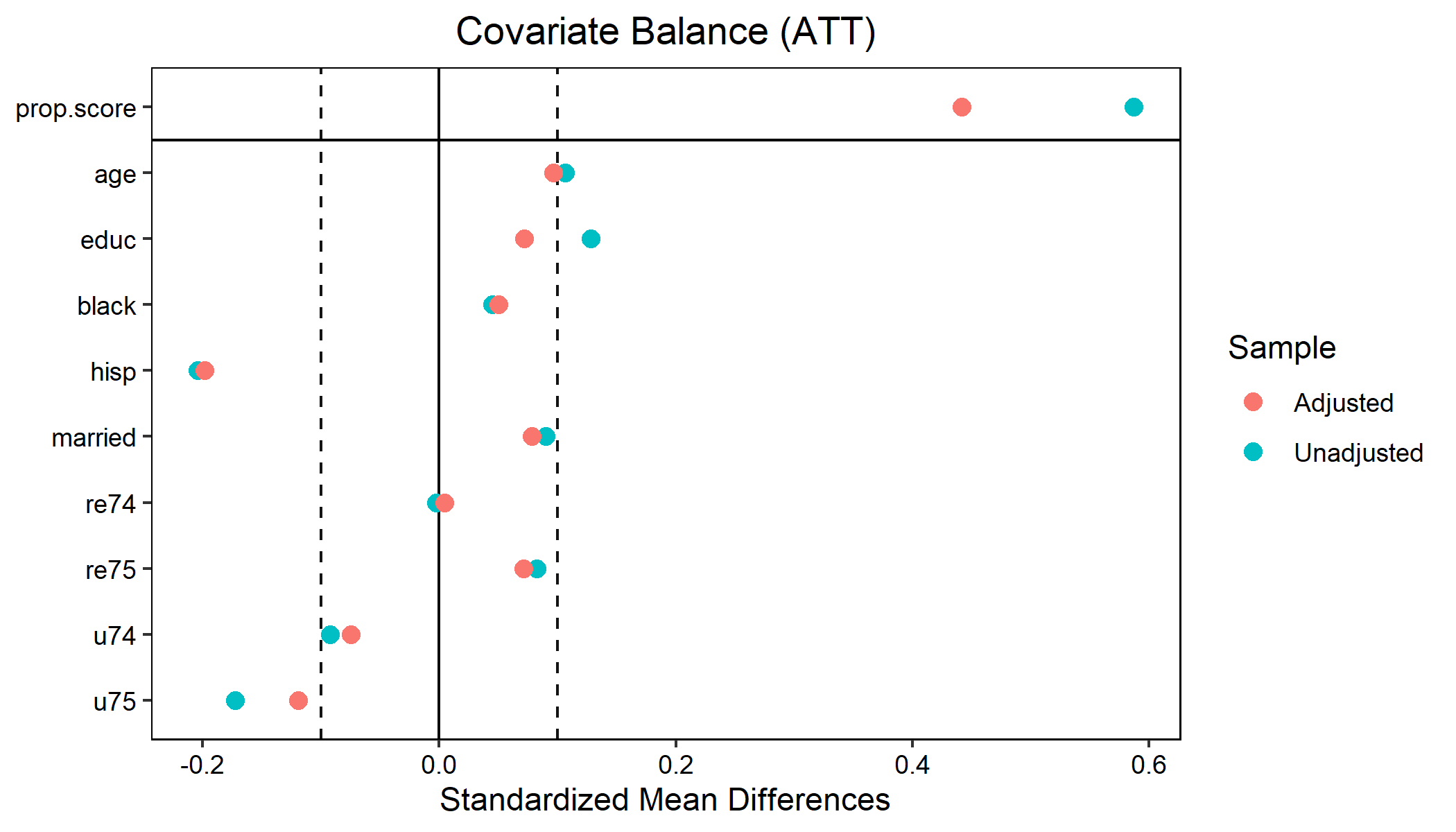
w1.out <- WeightIt::weightit(  
 formula = t ~ age + educ + black + hisp + married + re74 +  
 re75 + u74 + u75,  
 data = d5,  
 method = "gbm",  
 distribution = "bernoulli",  
 stop.method = "es.mean",  
 n.trees = 1000,  
 interaction.depth = 4,  
 shrinkage = 0.0005,  
 estimand = "ATE")  
w2.out <- WeightIt::weightit(  
 formula = t ~ age + educ + black + hisp + married + re74 +  
 re75 + u74 + u75,  
 data = d5,  
 method = "gbm",  
 distribution = "bernoulli",  
 stop.method = "es.mean",  
 n.trees = 1000,  
 interaction.depth = 4,  
 shrinkage = 0.0005,  
 estimand = "ATT")

#### Assess Balance with the Cobalt Package

cobalt::love.plot(w1.out, thresholds = c(m = .1), binary = "std") +  
 labs(title = "Covariate Balance (ATE)")



cobalt::love.plot(w2.out, thresholds = c(m = .1), binary = "std") +  
 labs(title = "Covariate Balance (ATT)")



#### Inference

The ATE and ATT weights can be obtained with w1.out$weights (ATE) and w2.out$weights (ATT).

m2.weightit <- lm(f, data = d5, weights = w1.out$weights)   
lmtest::coeftest(m2.weightit, vcov. = vcovHC(m2.weightit, "HC1"))

##   
## t test of coefficients:  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.537536 2.924890 -0.1838 0.854272   
## t 1.591101 0.648351 2.4541 0.014517 \*   
## age 0.061798 0.042806 1.4437 0.149550   
## educ 0.435506 0.165826 2.6263 0.008937 \*\*  
## black -2.037768 1.087347 -1.8741 0.061592 .   
## hisp 0.567686 1.534745 0.3699 0.711645   
## married -0.010996 0.894991 -0.0123 0.990203   
## re74 0.141330 0.134181 1.0533 0.292801   
## re75 0.002692 0.132501 0.0203 0.983800   
## u74 1.826432 1.619647 1.1277 0.260081   
## u75 -1.340307 1.490193 -0.8994 0.368929   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

m3.weightit <- lm(f, data = d5, weights = w2.out$weightit)   
lmtest::coeftest(m3.weightit, vcov. = vcovHC(m3.weightit, "HC1"))

##   
## t test of coefficients:  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.221429 2.824293 0.0784 0.93754   
## t 1.672042 0.661797 2.5265 0.01187 \*  
## age 0.053668 0.040388 1.3288 0.18461   
## educ 0.402947 0.161092 2.5013 0.01274 \*  
## black -2.039466 1.038581 -1.9637 0.05020 .  
## hisp 0.424649 1.427471 0.2975 0.76624   
## married -0.146662 0.864040 -0.1697 0.86529   
## re74 0.123573 0.127147 0.9719 0.33165   
## re75 0.019458 0.140630 0.1384 0.89001   
## u74 1.380999 1.554643 0.8883 0.37487   
## u75 -1.071817 1.408301 -0.7611 0.44703   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Practice 2: Matching Estimators

## Load Data

p2.d <- read\_dta("data/prac2.dta") %>%  
 haven::zap\_formats() %>%  
 sjlabelled::remove\_all\_labels() %>%  
 as\_tibble()

## Breusch-Pagan Test for Heteroskedasticity

The homoscedasticity assumption is not valid (e.g., p-value of the test for age97 is < .05), indicating that the conditional variance of the outcome variable was not constant across levels of child’s age, therefore a robust estimation of variance is warranted.

p2.m0 <- lm(lwss97 ~ kuse + male + black + age97 + pcged97 + mratio96 + pcg\_adc, data = p2.d)  
get\_bptest <- function(data, lm.model, var) {  
 b <- lmtest::bptest(lm.model, as.formula(paste0("~", var)),   
 data = data, studentize = F)  
 return(tibble(variable = var, statistic = b$statistic,   
 df = b$parameter, p.value = b$p.value))  
}  
map\_dfr(c("kuse", "male", "black", "age97", "pcged97", "mratio96", "pcg\_adc"),   
 get\_bptest, data = p2.d, lm.model = p2.m0) %>%  
 flextable() %>%  
 flextable::set\_table\_properties(width = 1, layout = "autofit") %>%  
 flextable::colformat\_double(digits = 2)

| variable | statistic | df | p.value |
| --- | --- | --- | --- |
| kuse | 1.78 | 1.00 | 0.18 |
| male | 0.86 | 1.00 | 0.35 |
| black | 1.15 | 1.00 | 0.28 |
| age97 | 8.55 | 1.00 | 0.00 |
| pcged97 | 4.43 | 1.00 | 0.04 |
| mratio96 | 6.85 | 1.00 | 0.01 |
| pcg\_adc | 0.60 | 1.00 | 0.44 |

## Matching Estimators

### Define Outcome (Y), Treatment Index (Tr), and Variables to Match On (X)

Y <- p2.d$lwss97  
Tr <- p2.d$kuse  
X <- select(p2.d, male, black, age97, pcged97, mratio96, pcg\_adc)

### Get All Estimators

get\_match <- function(estimand, sample) {  
 m <- Matching::Match(Y = Y, Tr = Tr, X = X, M = 4, BiasAdjust = T, Var.calc = 4,  
 estimand = estimand, sample = sample)  
 return(list(  
 est = m$est[,1],  
 se = m$se,  
 t.stat = m$est[,1]/m$se,  
 p = (1 - pnorm(abs(m$est[,1]/m$se))) \* 2  
 ))  
}  
tribble(  
 ~estimator, ~estimand, ~sample,  
 "SATE", "ATE", T,  
 "PATE", "ATE", F,  
 "SATT", "ATT", T,  
 "PATT", "ATT", F,  
 "SATC", "ATC", T,  
 "PATC", "ATC", F  
) %>%  
 rowwise() %>%  
 mutate(match = list(get\_match(estimand, sample))) %>%  
 tidyr::unnest\_wider(match) %>%  
 select(-estimand, -sample) %>%  
 flextable::flextable() %>%  
 flextable::set\_table\_properties(width = 1, layout = "autofit")

| estimator | est | se | t.stat | p |
| --- | --- | --- | --- | --- |
| SATE | -5.448863 | 1.646936 | -3.3084850 | 0.0009380219 |
| PATE | -5.448863 | 1.652232 | -3.2978811 | 0.0009741739 |
| SATT | -1.277287 | 1.683284 | -0.7588067 | 0.4479681672 |
| PATT | -1.277287 | 1.695820 | -0.7531973 | 0.4513313553 |
| SATC | -7.016781 | 1.965677 | -3.5696503 | 0.0003574581 |
| PATC | -7.016781 | 1.969424 | -3.5628594 | 0.0003668372 |

1. <https://cimentadaj.github.io/blog/2016-09-19-obtaining-robust-standard-errors-and-odds-ratios/obtaining-robust-standard-errors-and-odds-ratios-for-logistic-regression-in-r/> [↑](#footnote-ref-32)