Project 6: Randomization and Matching

Takun Wang

2024-04-04

1. Introduction

In this project, you will explore the question of whether college education causally affects political participation. Specifically, you will use replication data from Who Matches? Propensity Scores and Bias in the Causal Effects of Education on Participation by former Berkeley PhD students John Henderson and Sara Chatfield. Their paper is itself a replication study of Reconsidering the Effects of Education on Political Participation by Cindy Kam and Carl Palmer. In their original 2008 study, Kam and Palmer argue that college education has no effect on later political participation, and use the propensity score matching to show that pre-college political activity drives selection into college and later political participation. Henderson and Chatfield in their 2011 paper argue that the use of the propensity score matching in this context is inappropriate because of the bias that arises from small changes in the choice of variables used to model the propensity score. They use genetic matching (at that point a new method), which uses an approach similar to optimal matching to optimize Mahalanobis distance weights. Even with genetic matching, they find that balance remains elusive however, thus leaving open the question of whether education causes political participation.

You will use these data and debates to investigate the benefits and pitfalls associated with matching methods. Replication code for these papers is available online, but as you'll see, a lot has changed in the last decade or so of data science! Throughout the assignment, use tools we introduced in lab from the tidyverse and the MatchIt packages. Specifically, try to use dplyr, tidyr, purrr, stringr, and ggplot instead of base R functions. While there are other matching software libraries available, MatchIt tends to be the most up to date and allows for consistent syntax.

2. Data

The data is drawn from the Youth-Parent Socialization Panel Study which asked students and parents a variety of questions about their political participation. This survey was conducted in several waves. The first wave was in 1965 and established the baseline pre-treatment covariates. The treatment is whether the student attended college between 1965 and 1973 (the time when the next survey wave was administered). The outcome is an index that calculates the number of political activities the student engaged in after 1965. Specifically, the key variables in this study are:

- **college**: Treatment of whether the student attended college or not. 1 if the student attended college between 1965 and 1973, 0 otherwise.
- **ppnscal**: Outcome variable measuring the number of political activities the student participated in. Additive combination of whether the student voted in 1972 or 1980 (student_vote), attended a campaign rally or meeting (student_meeting), wore a campaign button (student_button), donated money to a campaign (student_money), communicated with an elected official (student_communicate), attended a demonstration or protest (student_demonstrate), was involved with a local community event (student_community), or some other political participation (student_other)

Otherwise, we also have covariates measured for survey responses to various questions about political attitudes. We have covariates measured for the students in the baseline year, covariates for their parents in the baseline year, and covariates from follow-up surveys. **Be careful here**. In general, post-treatment covariates will be clear from the name (i.e. student_1973Married indicates whether the student was married in the 1973 survey). Be mindful that the baseline covariates were all measured in 1965, the treatment occurred between 1965 and 1973, and the outcomes are from 1973 and beyond. We will distribute the Appendix from Henderson and Chatfield that describes the covariates they used, but please reach out with any questions if you have questions about what a particular variable means.

```
## Load packages
library(tidyverse)
library(cobalt)
library(MatchIt)
library(ggpubr)
library(grid)
library(gridExtra)
```

```
## Load data
ypsps <- read_csv('data/ypsps.csv')
dim(ypsps)
```

```
## [1] 1254  174
```

```
colnames(ypsps)
```

```
##   [1] "interviewid"                "college"
##   [3] "student_vote"               "student_meeting"
##   [5] "student_other"              "student_button"
##   [7] "student_money"              "student_communicate"
##   [9] "student_demonstrate"        "student_community"
##  [11] "student_ppnscal"            "student_PubAff"
##  [13] "student_Newspaper"          "student_Radio"
##  [15] "student_TV"                 "student_Magazine"
##  [17] "student_FamTalk"            "student_FrTalk"
##  [19] "student_AdultTalk"          "student_PID"
##  [21] "student_SPID"               "student_GovtOpinion"
##  [23] "student_GovtCrook"          "student_GovtWaste"
##  [25] "student_TrGovt"             "student_GovtSmart"
##  [27] "student_Govt4All"           "student_Cynic"
##  [29] "student_LifeWish"           "student_GLuck"
##  [31] "student_FPlans"             "student_EgoA"
##  [33] "student_WinArg"             "student_StrOpinion"
##  [35] "student_MChange"            "student_EgoB"
##  [37] "student_TrOthers"           "student_OthHelp"
##  [39] "student_OthFair"            "student_Trust"
##  [41] "student_Senate"             "student_Tito"
##  [43] "student_Court"              "student_Govern"
##  [45] "student_CCamp"              "student_FDR"
##  [47] "student_Knowledge"          "student_NextSch"
##  [49] "student_GPA"                "student_SchOfficer"
##  [51] "student_SchPublish"         "student_Hobby"
##  [53] "student_SchClub"            "student_OccClub"
##  [55] "student_NeighClub"          "student_RelClub"
```

```
##  [57] "student_YouthOrg"             "student_MiscClub"
##  [59] "student_ClubLev"              "student_Phone"
##  [61] "student_Gen"                  "student_Race"
##  [63] "parent_Newspaper"             "parent_Radio"
##  [65] "parent_TV"                    "parent_Magazine"
##  [67] "parent_LifeWish"              "parent_GLuck"
##  [69] "parent_FPlans"                "parent_WinArg"
##  [71] "parent_StrOpinion"            "parent_MChange"
##  [73] "parent_TrOthers"              "parent_OthHelp"
##  [75] "parent_OthFair"               "parent_PID"
##  [77] "parent_SPID"                  "parent_Vote"
##  [79] "parent_Persuade"              "parent_Rally"
##  [81] "parent_OthAct"                "parent_PolClub"
##  [83] "parent_Button"                "parent_Money"
##  [85] "parent_Participate1"          "parent_Participate2"
##  [87] "parent_ActFrq"                "parent_GovtOpinion"
##  [89] "parent_GovtCrook"             "parent_GovtWaste"
##  [91] "parent_TrGovt"                "parent_GovtSmart"
##  [93] "parent_Govt4All"              "parent_Employ"
##  [95] "parent_EducHH"                "parent_EducW"
##  [97] "parent_ChurchOrg"             "parent_FratOrg"
##  [99] "parent_ProOrg"                "parent_CivicOrg"
## [101] "parent_CLOrg"                 "parent_NeighClub"
## [103] "parent_SportClub"             "parent_InfClub"
## [105] "parent_FarmGr"                "parent_WomenClub"
## [107] "parent_MiscClub"              "parent_ClubLev"
## [109] "parent_FInc"                  "parent_HHInc"
## [111] "parent_OwnHome"               "parent_Senate"
## [113] "parent_Tito"                  "parent_Court"
## [115] "parent_Govern"                "parent_CCamp"
## [117] "parent_FDR"                   "parent_Knowledge"
## [119] "parent_Gen"                   "parent_Race"
## [121] "parent_GPHighSchoolPlacebo"   "parent_HHCollegePlacebo"
## [123] "student_1973Married"          "student_1973Military"
## [125] "student_1973Drafted"          "student_1973Unemployed"
## [127] "student_1973NoEmployers"      "student_1973OwnHome"
## [129] "student_1973NoResidences"     "student_1973VoteNixon"
## [131] "student_1973VoteMcgovern"     "student_1973CollegeDegree"
## [133] "student_1973CurrentCollege"   "student_1973CollegeYears"
## [135] "student_1973HelpMinority"     "student_1973Busing"
## [137] "student_1973GovChange"        "student_1973VietnamRight"
## [139] "student_1973VietnamApprove"   "student_1973Trust"
## [141] "student_1973Luck"             "student_1973SureAboutLife"
## [143] "student_1973CurrentSituation" "student_1973FutureSituation"
## [145] "student_1973ThermMilitary"    "student_1973ThermRadical"
## [147] "student_1973ThermDems"        "student_1973ThermRep"
## [149] "student_1973ThermBlack"       "student_1973ThermWhite"
## [151] "student_1973ThermNixon"       "student_1973ThermMcgovern"
## [153] "student_1973Newspaper"        "student_1973PubAffairs"
## [155] "student_1973GovtEfficacy"     "student_1973GovtNoSay"
## [157] "student_1973PartyID"          "student_1973IncSelf"
## [159] "student_1973HHInc"            "student_1973ChurchAttend"
## [161] "student_1973Knowledge"        "student_1973Ideology"
## [163] "student_1982vote76"           "student_1982vote80"
```

```
## [165] "student_1982meeting"          "student_1982other"
## [167] "student_1982button"           "student_1982money"
## [169] "student_1982communicate"      "student_1982demonstrate"
## [171] "student_1982community"        "student_1982IncSelf"
## [173] "student_1982HHInc"            "student_1982College"
```

```
## Select variables (the same vars as Appendix A in Henderson & Chatfield 2009)
ypsps <- ypsps %>% select(
  ## ID, treatment, outcome (3)
  interviewid, college, student_ppnscal,
  ## STU: cognitive ability (7)
  student_PubAff, student_Newspaper, student_Radio, student_Magazine, student_FamTalk,
  student_FrTalk, student_AdultTalk,
  ## STU: external efficacy (6)
  student_GovtOpinion, student_GovtCrook, student_GovtWaste, student_TrGovt, student_GovtSmart,
  student_Govt4All,
  ## STU: personality characteristics (9)
  student_LifeWish, student_GLuck, student_FPlans, student_WinArg, student_StrOpinion,
  student_MChange, student_TrOthers, student_OthHelp, student_OthFair,
  ## STU: civic participation (9)
  student_SchOfficer, student_SchPublish, student_Hobby, student_SchClub, student_OccClub,
  student_NeighClub, student_RelClub, student_YouthOrg, student_MiscClub,
  ## STU: other (7)
  student_SPID, student_Knowledge, student_NextSch, student_Phone, student_Gen,
  student_Race, student_GPA,
  ## PAR: cognitive ability (4)
  parent_Newspaper, parent_Radio, parent_TV, parent_Magazine,
  ## PAR: external efficacy (6)
  parent_GovtOpinion, parent_GovtCrook, parent_GovtWaste, parent_TrGovt, parent_GovtSmart,
  parent_Govt4All,
  ## PAR: personality characteristics (9)
  parent_LifeWish, parent_GLuck, parent_FPlans, parent_WinArg, parent_StrOpinion,
  parent_MChange, parent_TrOthers, parent_OthHelp, parent_OthFair,
  ## PAR: civic participation (11)
  parent_ChurchOrg, parent_FratOrg, parent_ProOrg, parent_CivicOrg, parent_CLOrg,
  parent_NeighClub, parent_SportClub, parent_InfClub, parent_FarmGr, parent_WomenClub,
  parent_MiscClub,
  ## PAR: political participation (7)
  parent_Vote, parent_Persuade, parent_Rally, parent_OthAct, parent_PolClub,
  parent_Button, parent_Money,
  ## PAR: other (6)
  parent_SPID, parent_Knowledge, parent_Employ, parent_EducHH, parent_HHInc,
  parent_OwnHome)

## 84 variables in total
dim(ypsps)
```

```
## [1] 1254   84
```

```
## Convert some vars into factor class (for other non-binary vars, assume the values
## have a meaningful order and that the spacing between consecutive values is the same)
ypsps$student_Race <- as.factor(ypsps$student_Race)
```

3. Randomization

Matching is usually used in observational studies to to approximate random assignment to treatment. But could it be useful even in randomized studies? To explore the question do the following:

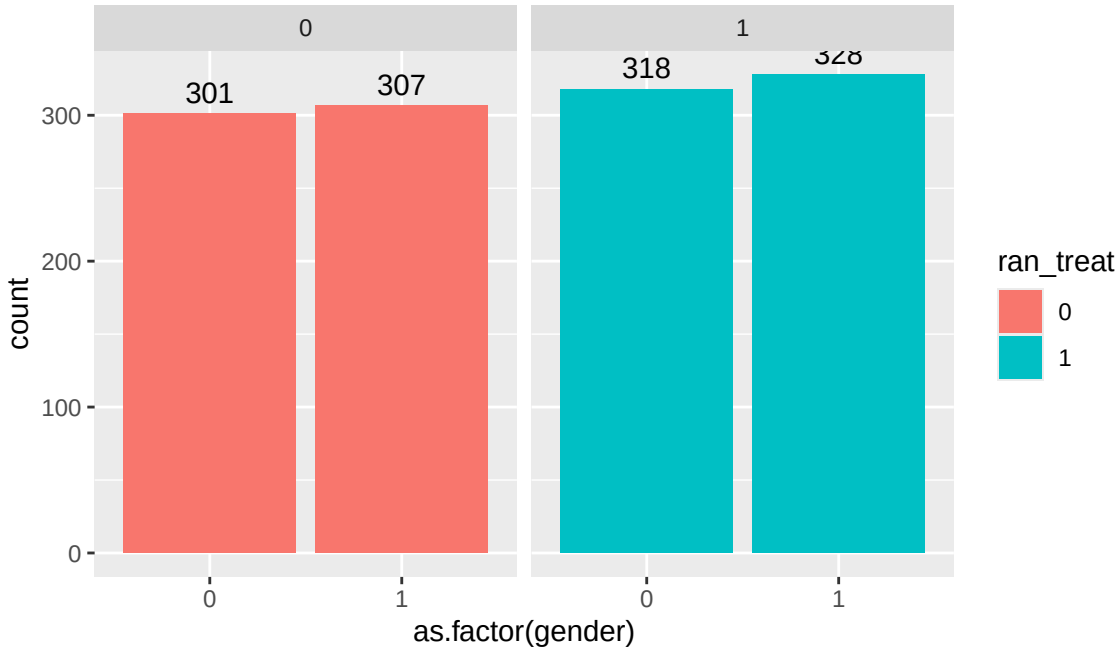
1. Generate a vector that randomly assigns each unit to either treatment or control.

2. Choose a baseline covariate (for either the student or parent). A binary covariate is probably best for this exercise.

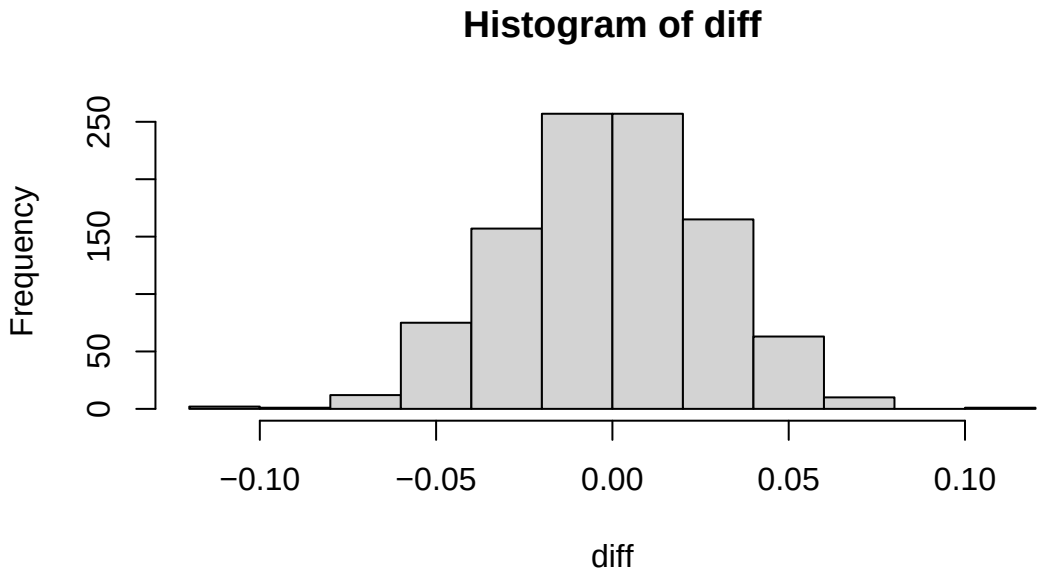
3. Visualize the distribution of the covariate by treatment/control condition. Are treatment and control balanced on this covariate?

4. Simulate the first 3 steps 10,000 times and visualize the distribution of treatment/control balance across the simulations.

```
## Set seed
set.seed(224)

## Generate a random vector and choose a baseline covariate
df <- data.frame(
  ran_treat = as.factor(sample(0:1,
                               size = nrow(ypsps),
                               replace = TRUE,
                               prob = c(0.50,0.50))),
  gender = ypsps$student_Gen)

## Visualize the distribution by treatment/control (ggplot)
ggplot(df, aes(x = as.factor(gender), group = ran_treat, fill = ran_treat)) +
  geom_bar(position = "dodge") +
  geom_text(stat = "count", aes(label = after_stat(count)), vjust = -0.5) +
  facet_grid(cols = vars(ran_treat))
```



```r
## Simulate this 1,000 times
diff <- rep(0, 1000)

for (i in c(1:1000)) {
  df <- data.frame(
    ran_treat = as.factor(sample(0:1,
                                 size = nrow(ypsps),
                                 replace = TRUE,
                                 prob = c(0.50,0.50))),
    gender = ypsps$student_Gen)

  df <- df %>% group_by(ran_treat) %>% summarise(mean = mean(gender))

  diff[i] <- df[[2, "mean"]] - df[[1, "mean"]]
}

## Plot
hist(diff)
```



3.1. Questions

1. What do you see across your simulations? Why does independence of treatment assignment and baseline covariates not guarantee balance of treatment assignment and baseline covariates?

Your Answer: The histogram from the simulations illustrates variability in the distribution of male/female percentages between the two groups across different iterations. This variability arises from the probabilistic nature of random sampling. In essence, while independence ensures that treatment assignment is not systematically biased by baseline characteristics, it does not eliminate the randomness inherent in sampling. This randomness can lead to scenarios where, by chance alone, there are significant differences in baseline

characteristics (such as gender distribution in this context) between treatment groups. Therefore, it's crucial to conduct and review balance checks after randomization and consider statistical techniques or design adjustments (e.g., stratification, matching) to mitigate potential imbalances in key baseline covariates.

4. Propensity Score Matching

4.1. One Model

Select covariates that you think best represent the "true" model predicting whether a student chooses to attend college, and estimate a propensity score model to calculate the Average Treatment Effect on the Treated (ATT). Plot the balance of the covariates. Report the balance of the p-scores across both the treatment and control groups, and using a threshold of standardized mean difference of p-score $\leq .1$, report the number of covariates that meet that balance threshold.

```
# (1) Construct a model and calculate the ATT

## Create a logistic regression model
glm(college ~
      student_Gen + student_Race + student_GPA + student_Newspaper + student_WinArg +
      parent_EducHH + parent_HHInc + parent_Employ + parent_OwnHome + parent_Newspaper,
    family = "binomial",
    data = ypsps) %>%
  summary()
```

```
##
## Call:
## glm(formula = college ~ student_Gen + student_Race + student_GPA +
##     student_Newspaper + student_WinArg + parent_EducHH + parent_HHInc +
##     parent_Employ + parent_OwnHome + parent_Newspaper, family = "binomial",
##     data = ypsps)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -0.27270    0.43544  -0.626   0.5311
## student_Gen        0.80479    0.14425   5.579 2.42e-08 ***
## student_Race2      0.62282    0.26150   2.382   0.0172 *
## student_Race3      2.55472    1.11244   2.297   0.0216 *
## student_GPA       -0.81890    0.11105  -7.374 1.65e-13 ***
## student_Newspaper -0.08743    0.05017  -1.743   0.0814 .
## student_WinArg    -0.20973    0.07804  -2.687   0.0072 **
## parent_EducHH      0.47343    0.05748   8.236  < 2e-16 ***
## parent_HHInc       0.15722    0.03466   4.536 5.74e-06 ***
## parent_Employ      0.21995    0.14382   1.529   0.1262
## parent_OwnHome     0.35558    0.17311   2.054   0.0400 *
## parent_Newspaper   0.09864    0.04757   2.073   0.0381 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1638.3  on 1253  degrees of freedom
## Residual deviance: 1302.5  on 1242  degrees of freedom
```

```
## AIC: 1326.5
##
## Number of Fisher Scoring iterations: 4
```

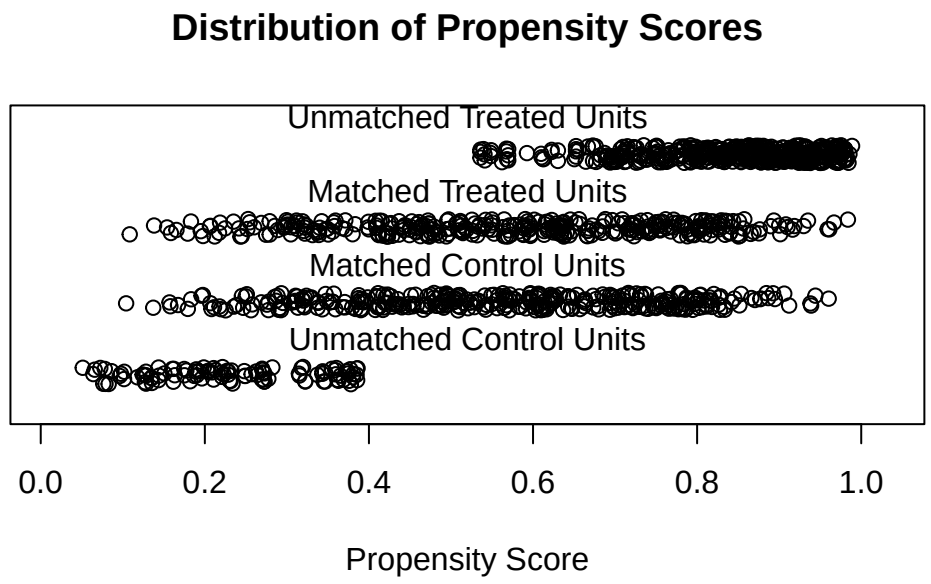
```
## Match it based on the model
m.out <- ypsps %>%
  matchit(college ~
            student_Gen + student_Race + student_GPA + student_Newspaper + student_WinArg +
            parent_EducHH + parent_HHInc + parent_Employ + parent_OwnHome + parent_Newspaper,
          distance = "glm",
          link = "logit",
          method = "nearest",
          caliper = 0.1,
          data = .)
```

```
## Warning: Fewer control units than treated units; not all treated units will get
## a match.
```

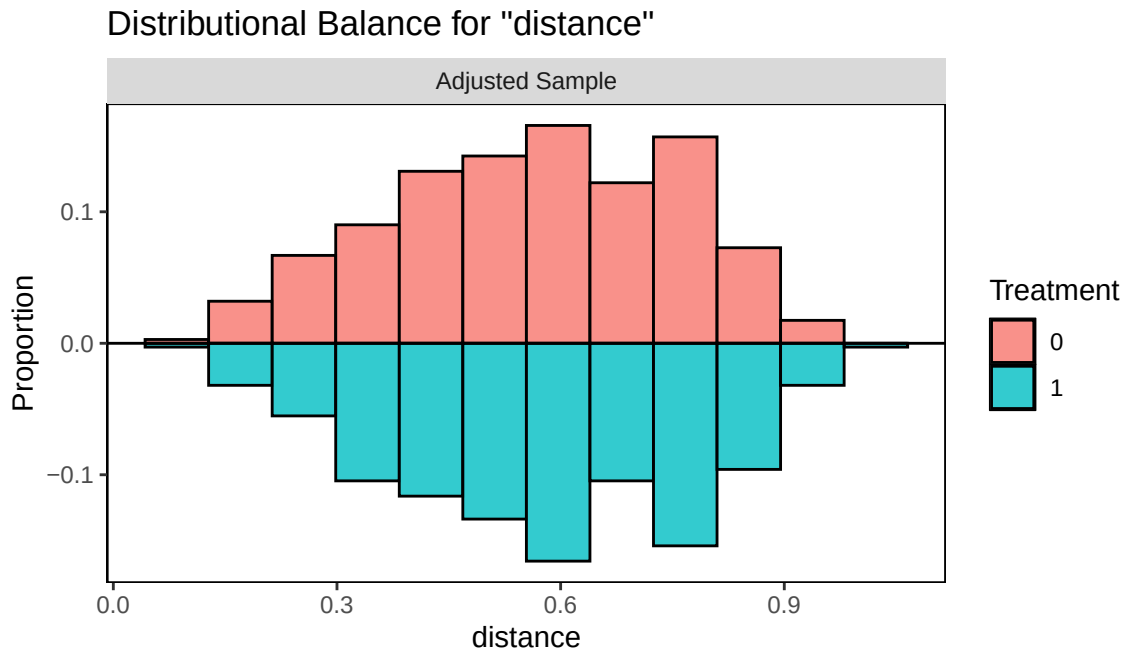
```
m.data <- match.data(m.out)

## Calculate the ATT
Y1 <- m.data %>% filter(college == 1) %>% summarise(mean(student_ppnscal)) %>% pull()
Y0 <- m.data %>% filter(college == 0) %>% summarise(mean(student_ppnscal)) %>% pull()
ATT <- Y1 - Y0
print(ATT)
```

```
## [1] 1.037791
```

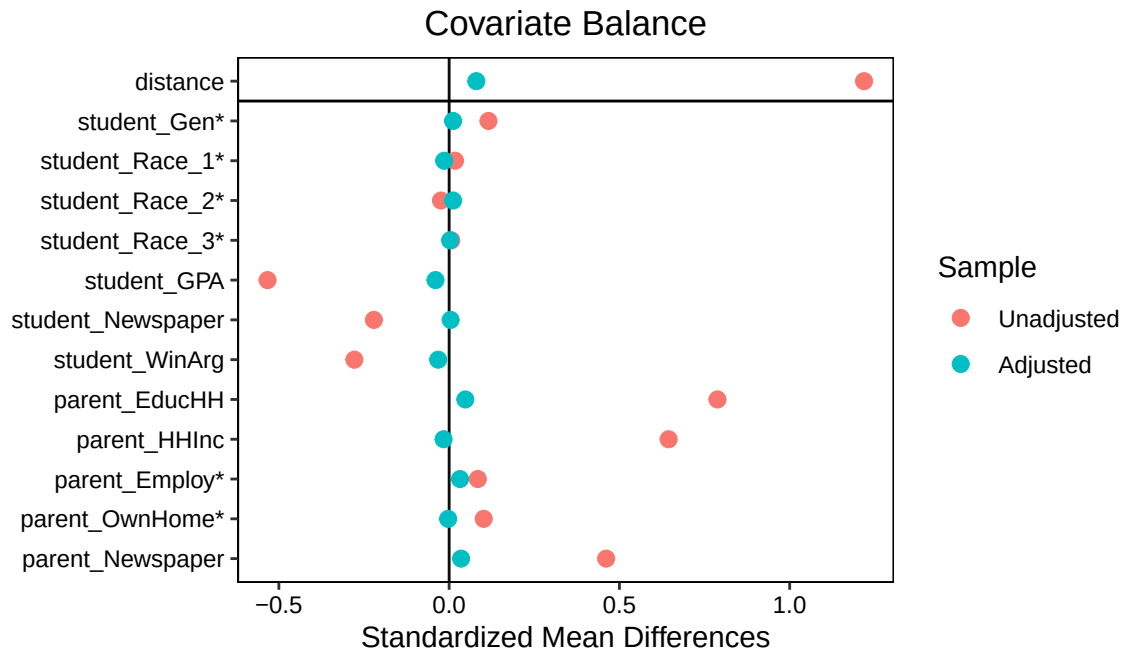
```
# (2) Check

## Check region of common support
plot(m.out, type = "jitter", interactive = FALSE)
```



```r
## Report the balance of the p-scores
bal.plot(m.out, var.name = "distance",
         mirror = TRUE, type = "histogram")
```




```r
## Plot the balance for the covariates
love.plot(m.out, stars = "raw")
```



```r
## Create covariate balance table
tb <- bal.tab(m.out, un = TRUE, disp = c("means"))
tb
```

```
## Balance Measures
##                       Type M.0.Un M.1.Un Diff.Un M.0.Adj M.1.Adj Diff.Adj
## distance          Distance 0.4841 0.7281  1.2184  0.5618  0.5778   0.0799
## student_Gen         Binary 0.4324 0.5479  0.1156  0.4738  0.4855   0.0116
## student_Race_1      Binary 0.9091 0.9265  0.0174  0.9215  0.9070  -0.0145
## student_Race_2      Binary 0.0887 0.0648 -0.0239  0.0756  0.0872   0.0116
## student_Race_3      Binary 0.0022 0.0087  0.0065  0.0029  0.0058   0.0029
## student_GPA        Contin. 2.6231 2.2740 -0.5332  2.5203  2.4942  -0.0400
## student_Newspaper  Contin. 2.2328 1.9427 -0.2209  2.1395  2.1453   0.0044
## student_WinArg     Contin. 2.3880 2.1370 -0.2782  2.3343  2.3052  -0.0322
## parent_EducHH      Contin. 2.1685 3.3238  0.7880  2.4070  2.4767   0.0476
## parent_HHInc       Contin. 5.9047 7.2864  0.6445  6.4448  6.4099  -0.0163
## parent_Employ       Binary 0.6120 0.6961  0.0842  0.6366  0.6686   0.0320
## parent_OwnHome      Binary 0.7428 0.8443  0.1015  0.7936  0.7907  -0.0029
## parent_Newspaper   Contin. 2.7938 3.3674  0.4612  3.0291  3.0727   0.0351
##
## Sample sizes
##           Control Treated
## All           451     803
## Matched       344     344
## Unmatched     107     459
```

```r
## Report the overall balance and the proportion of covariates that meet the balance threshold
SMD <- tb[[1]] %>% select(Diff.Adj) %>% slice(2:n()) %>% pull()
cat("Out of the", length(SMD), "covariates,", sum(abs(SMD) <= 0.1), "of them have SMD less or equal to 0
```

```
## Out of the 12 covariates, 12 of them have SMD less or equal to 0.1
```

```r
cat("The proportion is:", sum(abs(SMD) <= 0.1) / length(SMD) * 100, "%")
```

```
## The proportion is: 100 %
```

```r
## Calculate the mean percent improvement in the standardized mean diff.
imp_pct <- tb[[1]] %>% slice(2:n()) %>%
  summarise(Diff.Un = mean(abs(Diff.Un)), Diff.Adj = mean(abs(Diff.Adj))) %>%
  mutate(imp = -(Diff.Adj - Diff.Un)) %>%
  mutate(imp_pct = imp / Diff.Un) %>%
  pull(imp_pct)

cat("The mean percent improve in the SMD is:", round(imp_pct * 100, 2), "%")
```

```
## The mean percent improve in the SMD is: 92.33 %
```

4.2. Simulations

Henderson/Chatfield argue that an improperly specified propensity score model can actually *increase* the bias of the estimate. To demonstrate this, they simulate 800,000 different propensity score models by choosing different permutations of covariates. To investigate their claim, do the following:

1. Using as many simulations as is feasible (at least 10,000 should be ok, more is better!), randomly select the number of and the choice of covariates for the propensity score model.

2. For each run, store the ATT, the proportion of covariates that meet the standardized mean difference $\leq .1$ threshold, and the mean percent improvement in the standardized mean difference. You may also wish to store the entire models in a list and extract the relevant attributes as necessary.

3. Plot all of the ATTs against all of the balanced covariate proportions. You may randomly sample or use other techniques like transparency if you run into overplotting problems. Alternatively, you may use plots other than scatterplots, so long as you explore the relationship between ATT and the proportion of covariates that meet the balance threshold.

4. Finally choose 10 random models and plot their covariate balance plots (you may want to use a library like gridExtra to arrange these)

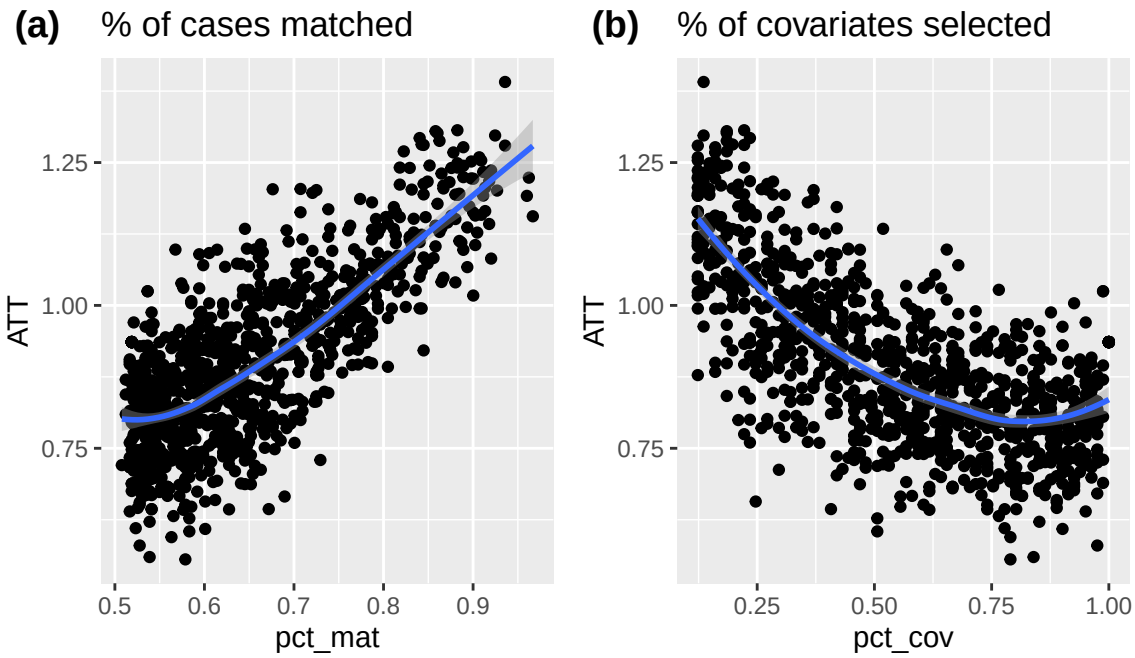
```
# (1) Simulation

## Set seed
set.seed(224)

## Create a dataframe to save results
n <- 1000
df <- data.frame(ATT     = c(1:n),
                 n_mat   = c(1:n),
                 pct_mat = c(1:n),
                 n_cov   = c(1:n),
                 pct_cov = c(1:n),
                 bal_pct = c(1:n),
                 imp_pct = c(1:n))

## Run the loop
for (i in c(1:n)) {

  ## Randomly select features
  col <- colnames(ypsps)[4:84]  # vars to choose from
  n_cov <- sample(10:81, 1)     # number of vars to select
  col <- sample(col, n_cov)         # sample n vars

  ## Generate data
  data <- ypsps[, c("student_ppnscal", "college", col)]

  ## Match data
  m.out <- matchit(college ~ . - student_ppnscal,
                   distance = "glm",
                   link = "logit",
                   method = "nearest",
                   caliper = 0.1,
                   data = data)
  m.data <- match.data(m.out)

  ## Calculate the ATT
  Y1 <- m.data %>% filter(college == 1) %>% summarise(mean(student_ppnscal)) %>% pull()
  Y0 <- m.data %>% filter(college == 0) %>% summarise(mean(student_ppnscal)) %>% pull()
  ATT <- Y1 - Y0
```

```r
  ## Cases matched
  tb <- bal.tab(m.out, un = TRUE)
  n_mat <- tb[[2]]["Matched (ESS)", "Treated"]
  pct_mat <- n_mat / nrow(data %>% filter(college == 0))

  ## Covariate balance
  SMD <- tb[[1]] %>% select(Diff.Adj) %>% slice(2:n()) %>% pull()
  imp_pct <- tb[[1]] %>% slice(2:n()) %>%
    summarise(Diff.Un = mean(abs(Diff.Un)), Diff.Adj = mean(abs(Diff.Adj))) %>%
    mutate(imp = -(Diff.Adj - Diff.Un)) %>%
    mutate(imp_pct = imp / Diff.Un) %>%
    pull(imp_pct)

  ## Save the results of each iteration
  df$ATT[i] <- ATT                                    # ATT
  df$n_mat[i] <- n_mat                                # number of cases successfully matched
  df$pct_mat[i] <- pct_mat                            # proportion of cases successfully matched
  df$n_cov[i] <- n_cov                                # number of covariates selected
  df$pct_cov[i] <- n_cov / 81                         # proportion of covariates selected
  df$bal_pct[i] <- sum(abs(SMD) <= 0.1) / length(SMD) # proportion of balanced covariates
  df$imp_pct[i] <- imp_pct                            # mean percent balance improvement
}

head(df)
```

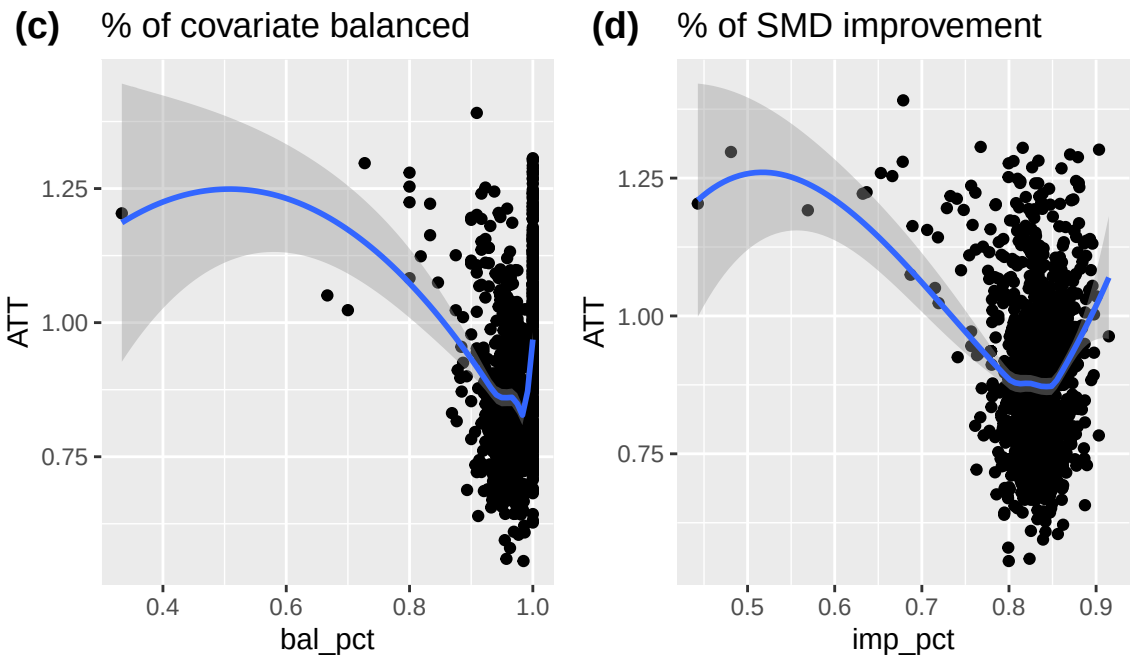
ATT	n_mat	pct_mat	n_cov	pct_cov	bal_pct	imp_pct
0.7044534	247	0.5476718	62	0.7654321	0.9354839	0.8230474
1.2032787	305	0.6762749	21	0.2592593	1.0000000	0.8586330
0.9212598	381	0.8447894	16	0.1975309	1.0000000	0.8264522
0.7857143	294	0.6518847	24	0.2962963	1.0000000	0.8768257
0.7914894	235	0.5210643	79	0.9753086	0.9753086	0.8153927
0.7491166	283	0.6274945	67	0.8271605	0.9855072	0.8437649

```r
# (2) Plot ATT

## ATT vs. proportion of cases successfully matched
g11 <- df %>% ggplot(aes(y = ATT, x = pct_mat), data = .) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "loess") +
  labs(title = '% of cases matched')

## ATT vs. proportion of covariates selected
g12 <- df %>% ggplot(aes(y = ATT, x = pct_cov), data = .) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "loess") +
  labs(title = '% of covariates selected')

## ATT vs. proportion of balanced covariates
g13 <- df %>% ggplot(aes(y = ATT, x = bal_pct), data = .) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "loess") +
  labs(title = '% of covariate balanced')
```

```
## ATT vs. proportion of mean percent balance improvement
g14 <- df %>% ggplot(aes(y = ATT, x = imp_pct), data = .) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "loess") +
  labs(title = '% of SMD improvement')

## Arrange all plots
ggarrange(g11, g12, labels = c('(a)', '(b)'))
```




```
ggarrange(g13, g14, labels = c('(c)', '(d)'))
```



4.3. Questions

1. How many simulations resulted in models with a higher proportion of balanced covariates? Do you have any concerns about this?

Your Answer: In the set of 1,000 simulations, 337 resulted in fully balanced covariates. However, a notable observation from graph (c) is the significant variability in estimated ATTs when all of the covariates appear balanced. This variability could be attributed to the randomly selected number of covariates in each iteration; notably, complete balance can be easier to be achieved with a minimal selection of covariates. This raises a concern illustrated by graph (b): fewer covariates tend to yield higher ATT estimates, indicating the significance of the quantity of covariates used in matching methods on the reliability of ATT estimations.

```
## Number of simulations resulted in all covariates being balanced
df %>% filter(bal_pct == 1) %>% nrow()
```

```
## [1] 337
```

```
## Simulations with bad proportion of SMD improvement
df %>% filter(imp_pct <= 0.65)
```

ATT	n_mat	pct_mat	n_cov	pct_cov	bal_pct	imp_pct
1.191686	433	0.9600887	10	0.1234568	0.9000000	0.5691739
1.224439	401	0.8891353	10	0.1234568	0.8000000	0.6366162
1.221675	406	0.9002217	10	0.1234568	0.8333333	0.6322123
1.297362	417	0.9246120	11	0.1358025	0.7272727	0.4809286
1.203762	319	0.7073171	12	0.1481481	0.3333333	0.4429117

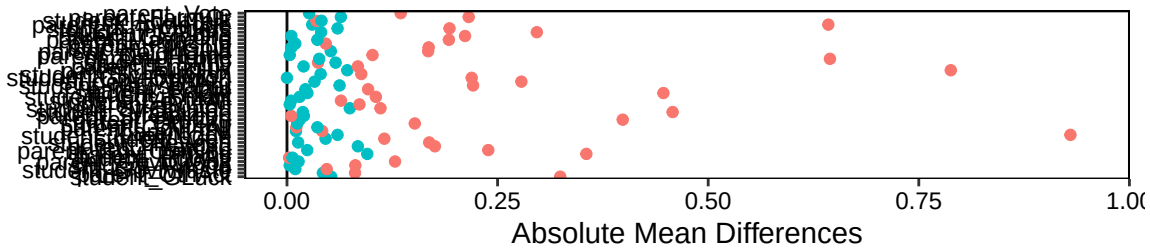
2. Analyze the distribution of the ATTs. Do you have any concerns about this distribution?

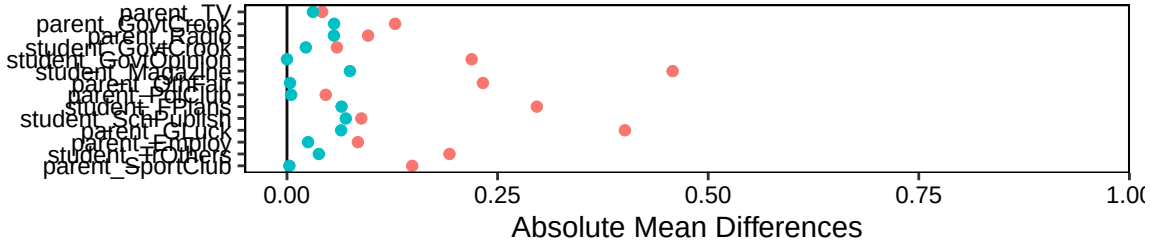
Your Answer: The observation that fewer selected covariates tend to result in higher ATT estimates suggests a need for further investigation. Simulations should be performed with a consistent number of selected covariates, albeit choosing different sets in each run. This approach would allow for a clearer understanding of the relationship between the proportion of balanced covariates, the mean percent of balance improvement, and their influence on ATT estimations. By standardizing the number of covariates across simulations, we can more accurately assess the effects of covariate balance on ATT variability.

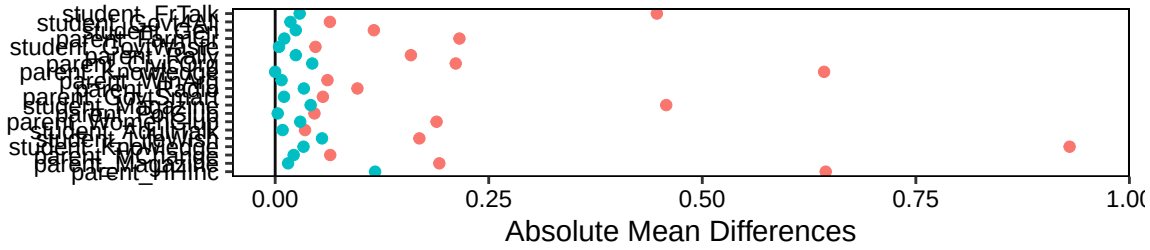
3. Do your 10 randomly chosen covariate balance plots produce similar numbers on the same covariates? Is it a concern if they do not?

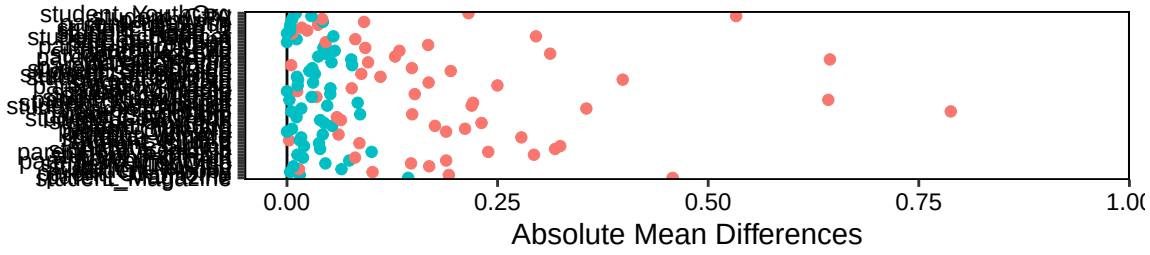
Your Answer: The ten randomly chosen covariate balance plots do not consistently yield similar metrics for the same covariates. This inconsistency is a potential concern, as it may indicate that certain covariates remain unbalanced post-matching. The ability to balance a covariate effectively can vary based on the matching algorithm employed, the interplay with other covariates, and additional factors. This variability underscores the importance of routinely checking covariate balance as a part of the matching process to ensure the reliability and validity of the treatment effect estimations.

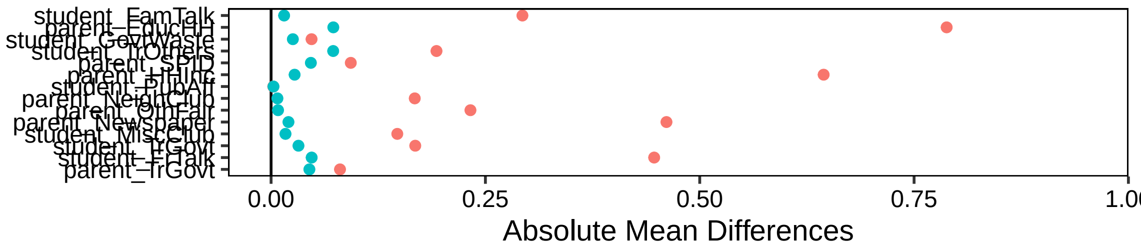
```
## Run a loop
g <- list()

for (i in c(1:10)) {

  ## Randomly select features
  col <- colnames(ypsps)[4:84]  # vars to choose from
  n_cov <- sample(10:81, 1)     # number of vars to select
  col <- sample(col, n_cov)     # sample vars

  ## Match data
  data <- ypsps[, c("student_ppnscal", "college", col)]
  m.out <- matchit(college ~ . - student_ppnscal,
                   distance = "glm",
                   link = "logit",
                   method = "nearest",
                   caliper = 0.1,
                   data = data)

  ## Plot
  g[[i]] <- love.plot(m.out,
                      drop.distance = TRUE,
                      abs = TRUE,
                      limits = list(m = c(0, 1)),
                      size = 2,
                      position = "none",
                      title = "")
}

## Arrange all plots
marrangeGrob(g, nrow = 2, ncol = 1, top = "")
```

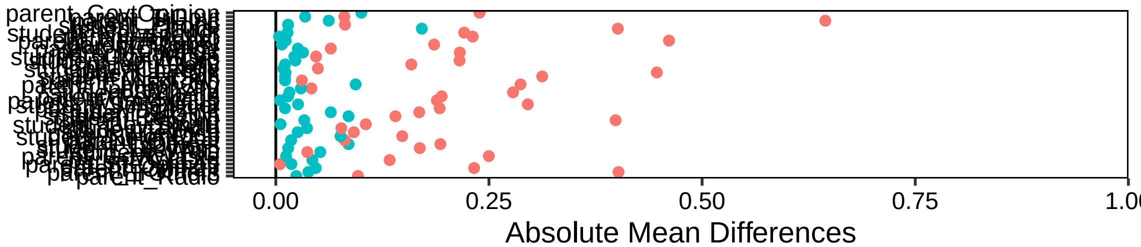

0.00
0.25
0.50
0.75
1.00
Absolute Mean Differences

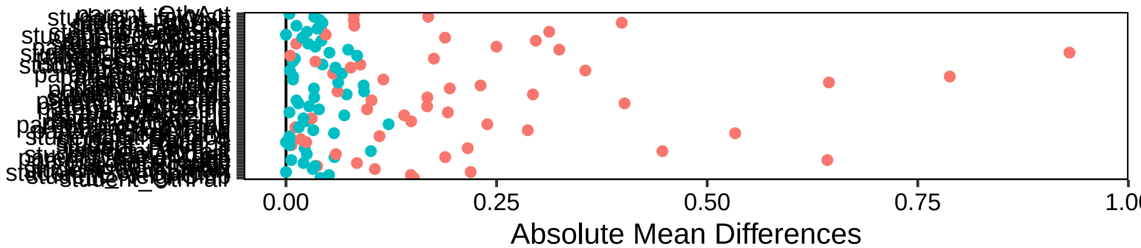

parent_TV
parent_GovTextbook
parent_Radio
student_GovTextbook
student_GovOpinion
student_Magazine
parent_SchFair
parent_PTAClub
student_Friends
student_SchPunish
parent_SchTuck
parent_Employ
student_HOthers
parent_SportClub
0.00
0.25
0.50
0.75
1.00
Absolute Mean Differences

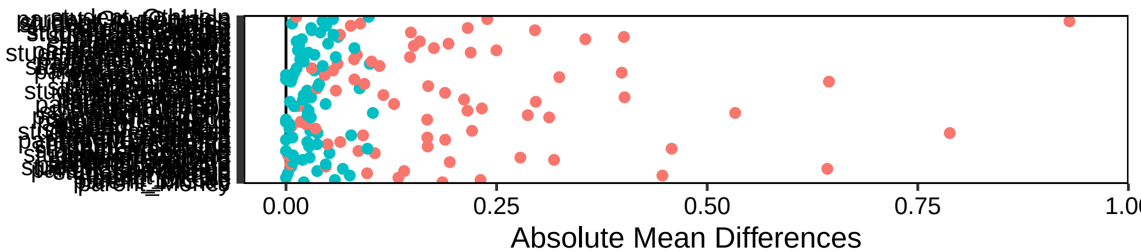

student_FrTalk
parent_Magazine
parent_HHline
0.00
0.25
0.50
0.75
1.00
Absolute Mean Differences

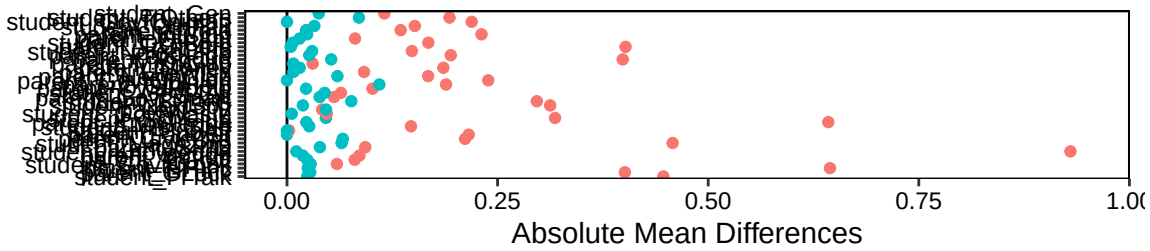

0.00
0.25
0.50
0.75
1.00
Absolute Mean Differences

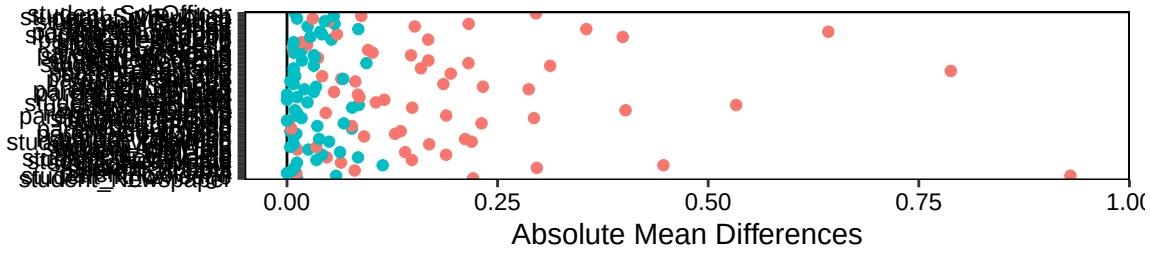

student_FamTalk
parent_EducHH
student_GovtWaste
student_TrOthers
parent_SPID
parent_HHInt
student_PubAff
parent_NeighClub
parent_OthPart
parent_Newspaper
student_MiscClub
student_TrGovt
student_FTalk
parent_TrGovt
0.00
0.25
0.50
0.75
Absolute Mean Differences


parent_GovtOpinion
parent_Radio
0.00
0.25
0.50
0.75
Absolute Mean Differences


0.00
0.25
0.50
0.75
Absolute Mean Differences


parent_Money
0.00
0.25
0.50
0.75
Absolute Mean Differences





5. Matching Algorithm of Your Choice

5.1. Simulate Alternative Model

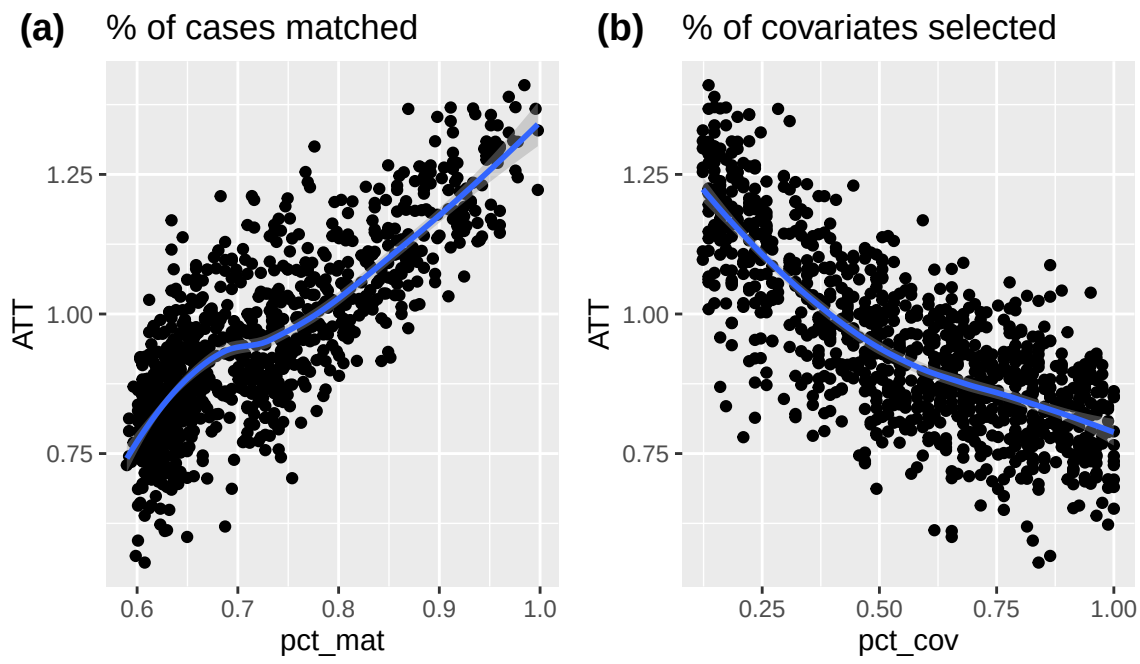
Henderson/Chatfield propose using genetic matching to learn the best weights for Mahalanobis distance matching. Choose a matching algorithm other than the propensity score (you may use genetic matching if you wish, but it is also fine to use the greedy or optimal algorithms we covered in lab instead). Repeat the same steps as specified in Section 4.2 and answer the following questions:

```
# (1) Simulation

## Set seed
set.seed(224)

## Create a dataframe to save results
n <- 1000
df2 <- data.frame(ATT     = c(1:n),
                  n_mat   = c(1:n),
                  pct_mat = c(1:n),
                  n_cov   = c(1:n),
                  pct_cov = c(1:n),
                  bal_pct = c(1:n),
                  imp_pct = c(1:n))

## Run the loop
for (i in c(1:n)) {

  ## Randomly select features
  col <- colnames(ypsps)[4:84]  # vars to choose from
  n_cov <- sample(10:81, 1)     # number of vars to select
```

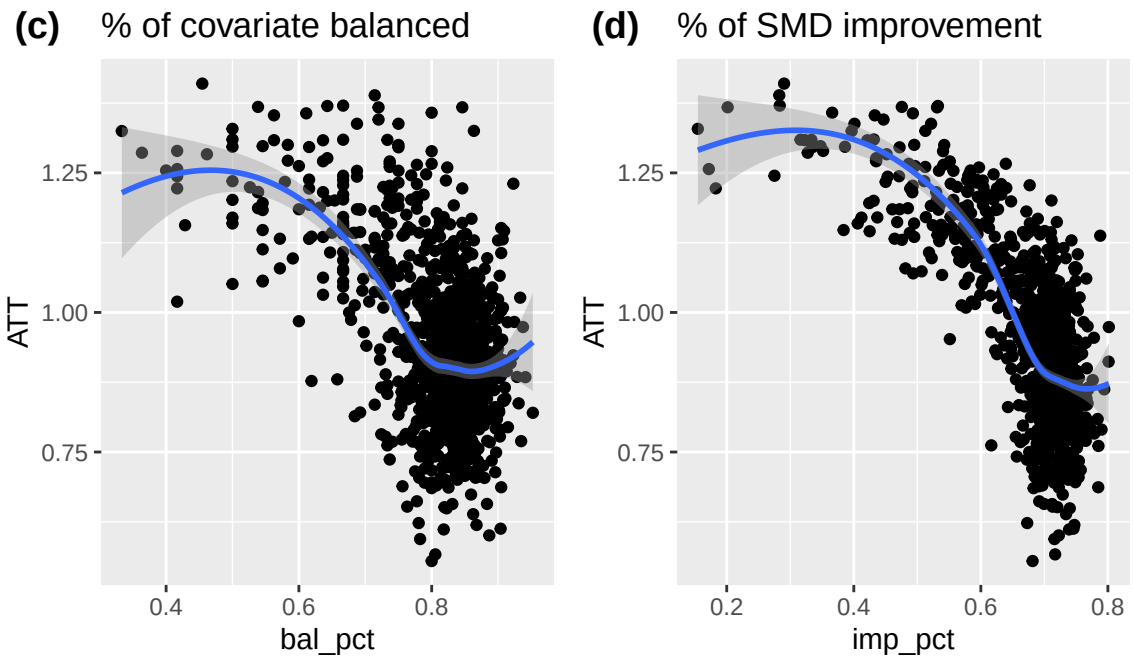
```r
  col <- sample(col, n_cov)     # sample vars

  ## Generate data
  data <- ypsps[, c("student_ppnscal", "college", col)]

  ## Match data
  m.out <- matchit(college ~ . - student_ppnscal,
                   distance = "randomforest",
                   method = "nearest",
                   caliper = 0.1,
                   data = data)
  m.data <- match.data(m.out)

  ## Calculate the ATT
  Y1 <- m.data %>% filter(college == 1) %>% summarise(mean(student_ppnscal)) %>% pull()
  Y0 <- m.data %>% filter(college == 0) %>% summarise(mean(student_ppnscal)) %>% pull()
  ATT <- Y1 - Y0

  ## Cases matched
  tb <- bal.tab(m.out, un = TRUE)
  n_mat <- tb[[2]]["Matched (ESS)", "Treated"]
  pct_mat <- n_mat / nrow(data %>% filter(college == 0))

  ## Covariate balance
  SMD <- tb[[1]] %>% select(Diff.Adj) %>% slice(2:n()) %>% pull()
  imp_pct <- tb[[1]] %>% slice(2:n()) %>%
    summarise(Diff.Un = mean(abs(Diff.Un)), Diff.Adj = mean(abs(Diff.Adj))) %>%
    mutate(imp = -(Diff.Adj - Diff.Un)) %>%
    mutate(imp_pct = imp / Diff.Un) %>%
    pull(imp_pct)

  ## Save the results of each iteration
  df2$ATT[i] <- ATT                                    # ATT
  df2$n_mat[i] <- n_mat                                # number of cases successfully matched
  df2$pct_mat[i] <- pct_mat                            # proportion of cases successfully matched
  df2$n_cov[i] <- n_cov                                # number of covariates selected
  df2$pct_cov[i] <- n_cov / 81                         # proportion of covariates selected
  df2$bal_pct[i] <- sum(abs(SMD) <= 0.1) / length(SMD)  # proportion of balanced covariates
  df2$imp_pct[i] <- imp_pct                            # mean percent balance improvement
}

head(df2)
```

ATT	n_mat	pct_mat	n_cov	pct_cov	bal_pct	imp_pct
0.7509158	273	0.6053215	62	0.7654321	0.7903226	0.7333968
0.9726776	366	0.8115299	27	0.3333333	0.7777778	0.6620790
0.9204152	289	0.6407982	65	0.8024691	0.7692308	0.6765556
0.8602151	279	0.6186253	79	0.9753086	0.8765432	0.7194292
1.3578199	422	0.9356984	18	0.2222222	0.8000000	0.3659632
0.9716981	318	0.7050998	65	0.8024691	0.8059701	0.6874206

```r
# (2) Plot ATT

## ATT vs. proportion of cases successfully matched
g21 <- df2 %>% ggplot(aes(y = ATT, x = pct_mat), data = .) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "loess") +
  labs(title = '% of cases matched')

## ATT vs. proportion of covariates selected
g22 <- df2 %>% ggplot(aes(y = ATT, x = pct_cov), data = .) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "loess") +
  labs(title = '% of covariates selected')

## ATT vs. proportion of balanced covariates
g23 <- df2 %>% ggplot(aes(y = ATT, x = bal_pct), data = .) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "loess") +
  labs(title = '% of covariate balanced')

## ATT vs. proportion of mean percent balance improvement
g24 <- df2 %>% ggplot(aes(y = ATT, x = imp_pct), data = .) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "loess") +
  labs(title = '% of SMD improvement')

## Arrange all plots
ggarrange(g21, g22, labels = c('(a)', '(b)'))
```




```r
ggarrange(g23, g24, labels = c('(c)', '(d)'))
```



5.2. Questions

1. Does your alternative matching method have more runs with higher proportions of balanced covariates?

Your Answer: The alternative matching method utilizing random forest did not achieve a higher proportion of balanced covariates compared to the baseline approach. Specifically, while the baseline method achieved a 100% balance in 337 out of 1,000 simulations, the random forest method resulted in only 51 out of 1,000 simulations achieving at least 90% balance. This reduced efficacy in achieving covariate balance with the random forest method might stem from its inherent characteristics, which typically require larger sample sizes to function optimally. Despite this, the trend that fewer selected covariates are associated with higher ATT estimates persisted, as demonstrated in the data where iterations with lesser improvements in balance also selected fewer covariates and reported higher ATT estimates.

```
## Number of simulations resulted in all covariates being balanced
df2 %>% filter(bal_pct >= 0.9) %>% nrow()
```

```
## [1] 51
```

```
## Simulations with bad proportion of SMD improvement
df2 %>% filter(imp_pct <= 0.3)
```

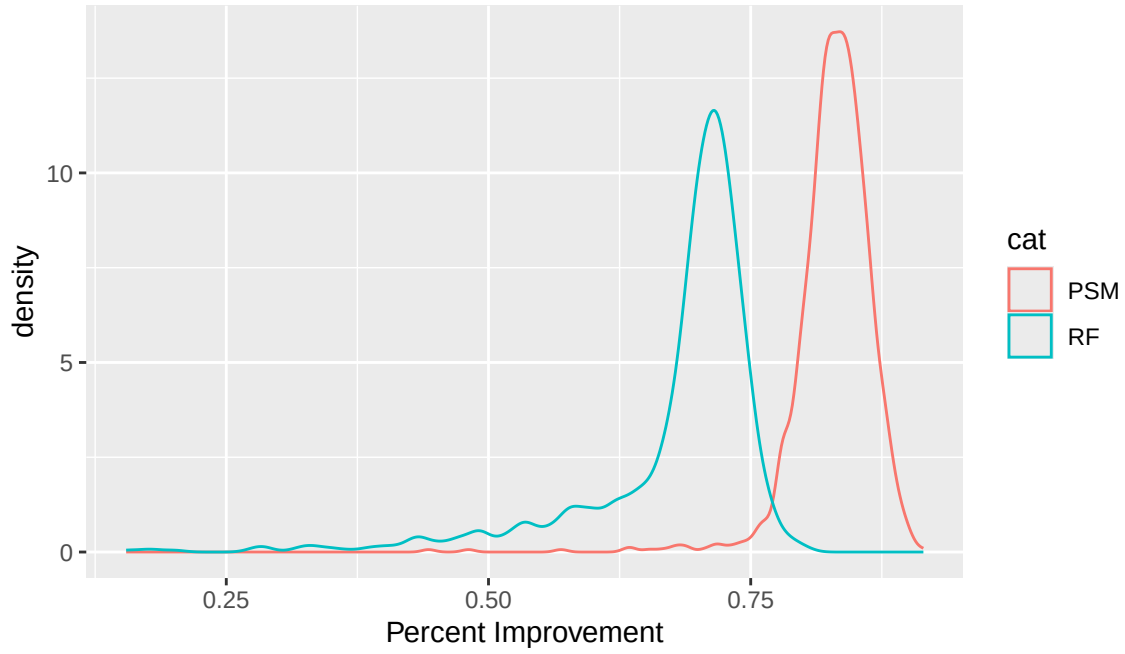
ATT	n_mat	pct_mat	n_cov	pct_cov	bal_pct	imp_pct
1.370454	440	0.9756098	12	0.1481481	0.6666667	0.2834548
1.256818	440	0.9756098	12	0.1481481	0.4166667	0.1718653
1.328889	450	0.9977827	10	0.1234568	0.5000000	0.1545288
1.367483	449	0.9955654	11	0.1358025	0.8461538	0.2014989
1.222222	450	0.9977827	12	0.1481481	0.4166667	0.1828105
1.409910	444	0.9844789	11	0.1358025	0.4545455	0.2902872

ATT	n_mat	pct_mat	n_cov	pct_cov	bal_pct	imp_pct
1.244898	441	0.9778271	15	0.1851852	0.6666667	0.2750973
1.389016	437	0.9689579	12	0.1481481	0.7142857	0.2827268

2. Use a visualization to examine the change in the distribution of the percent improvement in balance in propensity score matching vs. the distribution of the percent improvement in balance in your new method. Which did better? Analyze the results in 1-2 sentences.

Your Answer: Visual comparison of the percent improvement in balance between propensity score matching and the new method reveals that the logistic regression-based propensity score matching yields superior covariate balance between treatment and control groups. The machine learning-based method, here random forest, demonstrated less improvement in balance, likely due to its higher data requirements. This is particularly evident in smaller sample sizes, where logistic regression maintains its effectiveness in improving covariate balance, as highlighted by the comparative graph below.

```
# Visualize distributions of percent improvement
df$cat <- "PSM"
df2$cat <- "RF"
df3 <- rbind(df, df2)

ggplot(df3) +
  geom_density(aes(x = imp_pct, color = cat)) +
  labs(x = "Percent Improvement")
```



6. Discussion Questions

1. Why might it be a good idea to do matching even if we have a randomized or as-if-random design?

Your Answer: Matching in the context of randomized or as-if-random designs can be particularly beneficial despite the fundamental assumption of randomization ensuring unbiased treatment assignment. The key reason lies in the potential for random sampling variability to produce imbalances in baseline covariates across groups purely by chance. Implementing matching techniques in these scenarios serves to mitigate such imbalances, thereby enhancing the precision of the estimated effects by reducing the standard errors. This approach ensures a more accurate and reliable assessment of the treatment effect, capitalizing on the strengths of both randomization and matching to address inherent sampling randomness.

2. The standard way of estimating the propensity score is using a logistic regression to estimate probability of treatment. Given what we know about the curse of dimensionality, do you think there might be advantages to using other machine learning algorithms (decision trees, bagging/boosting forests, ensembles, etc.) to estimate propensity scores instead?

Your Answer: Reflecting on the simulation results, three critical considerations emerge. First, the efficacy of machine learning-based methods is heavily contingent on the availability of large datasets. In scenarios where sample sizes are small, these algorithms might not perform optimally due to their complexity and the curse of dimensionality. Second, the decision to employ machine learning algorithms over logistic regression can and should be grounded in empirical evidence. By assessing whether these algorithms indeed facilitate better covariate balance between treatment and control groups, researchers can make informed choices. Last, the choice also hinges on the domain-specific understanding of the treatment assignment mechanism. Logistic regression offers a high degree of interpretability by relying on specified functional forms that mirror theoretical understandings of the assigning process. Machine learning algorithms, in contrast, often trade off interpretability for flexibility, adapting to complex patterns in the data without predefined assumptions about the nature of the relationships between covariates and treatment assignment.