# Regression and Matching in R

Pilar Manzi

04/28/2021

## About this workshop/document

Note that this does not cover in-depth explanations of the methods, but instead focuses on how to run them in R, what packages to use, etc. Graphing is also not the focus here, so many figures have not been customized for a better appearance.

## The puzzle and data

Today we'll be investigating whether college educated individuals vote at higher rates? Does college education *cause* an increase in electoral participation?

We'll be working with the last wave of the World Value Survey, which has data for many countries across the world. On the folder, the file is called "WVS_crossnational.dta".

Since it's an R file, we'll be using the 'read_RDS' command.

```
wvs <- readRDS("WVS_crossnational.rds")

# to remove haven (stata) labels
wvs <- zap_labels(wvs)
```

To explore datasets, there are many useful commands: 'head', 'glimpse', 'names', 'str'. Because this one is pretty large, we'll go with 'head':

```
head(wvs)
```

```
## # A tibble: 6 x 547
##   version  doi    A_WAVE A_STUDY B_COUNTRY B_COUNTRY_ALPHA C_COW_NUM C_COW_ALPHA
##   <chr>    <chr>   <dbl>   <dbl>     <dbl> <chr>               <dbl> <chr>
## 1 1-6-0 (~ doi.o~      7       2        20 AND                   232 AND
## 2 1-6-0 (~ doi.o~      7       2        20 AND                   232 AND
## 3 1-6-0 (~ doi.o~      7       2        20 AND                   232 AND
## 4 1-6-0 (~ doi.o~      7       2        20 AND                   232 AND
## 5 1-6-0 (~ doi.o~      7       2        20 AND                   232 AND
## 6 1-6-0 (~ doi.o~      7       2        20 AND                   232 AND
## # ... with 539 more variables: A_YEAR <dbl>, D_INTERVIEW <dbl>,
## #   J_INTDATE <dbl>, FW_END <dbl>, FW_START <dbl>, K_TIME_START <dbl>,
## #   K_TIME_END <dbl>, K_DURATION <dbl>, Q_MODE <dbl>, N_REGION_ISO <dbl>,
## #   N_REGION_WVS <dbl>, N_TOWN <dbl>, G_TOWNSIZE <dbl>, G_TOWNSIZE2 <dbl>,
## #   H_SETTLEMENT <dbl>, H_URBRURAL <dbl>, I_PSU <dbl>, O1_LONGITUDE <dbl>,
## #   O2_LATITUDE <dbl>, S_INTLANGUAGE <dbl>, LNGE_ISO <chr>, E_RESPINT <dbl>,
## #   F_INTPRIVACY <dbl>, E1_LITERACY <dbl>, W_WEIGHT <dbl>, S018 <dbl>,
## #   pwght <dbl>, S025 <dbl>, Q1 <dbl>, Q2 <dbl>, Q3 <dbl>, Q4 <dbl>, Q5 <dbl>,
## #   Q6 <dbl>, Q7 <dbl>, Q8 <dbl>, Q9 <dbl>, Q10 <dbl>, Q11 <dbl>, Q12 <dbl>,
```

```
## #   Q13 <dbl>, Q14 <dbl>, Q15 <dbl>, Q16 <dbl>, Q17 <dbl>, Q18 <dbl>,
## #   Q19 <dbl>, Q20 <dbl>, Q21 <dbl>, Q22 <dbl>, Q23 <dbl>, Q24 <dbl>,
## #   Q25 <dbl>, Q26 <dbl>, Q27 <dbl>, Q28 <dbl>, Q29 <dbl>, Q30 <dbl>,
## #   Q31 <dbl>, Q32 <dbl>, Q33 <dbl>, Q33_3 <dbl>, Q34 <dbl>, Q34_3 <dbl>,
## #   Q35 <dbl>, Q35_3 <dbl>, Q36 <dbl>, Q37 <dbl>, Q38 <dbl>, Q39 <dbl>,
## #   Q40 <dbl>, Q41 <dbl>, Q42 <dbl>, Q43 <dbl>, Q44 <dbl>, Q45 <dbl>,
## #   Q46 <dbl>, Q47 <dbl>, Q48 <dbl>, Q49 <dbl>, Q50 <dbl>, Q51 <dbl>,
## #   Q52 <dbl>, Q53 <dbl>, Q54 <dbl>, Q55 <dbl>, Q56 <dbl>, Q57 <dbl>,
## #   Q58 <dbl>, Q59 <dbl>, Q60 <dbl>, Q61 <dbl>, Q62 <dbl>, Q63 <dbl>,
## #   Q64 <dbl>, Q65 <dbl>, Q66 <dbl>, Q67 <dbl>, Q68 <dbl>, Q69 <dbl>, ...
```

## Explore and recode variables

We'll recode the variables we'll be using, mostly to remove DKs or to group categories.

Votes at the national level: Q222. Group 'always' and 'usually' (1 and 2 v 3, leave 4's out)

```
wvs <- wvs %>% mutate(vote = case_when(
  Q222==1 | Q222==2 ~ 1,
  Q222==3 ~ 0
))
```

College variable: based on Q275 (level of education), group 6-8 (Bachelors and Grad)

```
wvs <- wvs %>% mutate(college = case_when(
 Q275<6 ~0,
 Q275 >5 ~ 1,
))
```

Father's education (Q278): just remove NAs

```
wvs <- wvs %>% mutate(father_educ = case_when(
  Q278>=0 ~ Q278
))
```

Urban/rural: H_URBRURAL (1= urban, 2= rural)

```
wvs <- wvs %>% mutate(urban = case_when(
  H_URBRURAL == 1 ~ 1,
  H_URBRURAL == 2 ~ 0
))
```

Self-reported income scale: Q288, goes from 1 to 10

```
wvs <- wvs %>% mutate(income = case_when(
  Q288>=0 ~ Q288
))
```

Sex: Q260- rename and make into dummy (0,1)

```
wvs <- wvs %>% mutate(sex = case_when(
  Q260 == 1 ~ 1,
  Q260 == 2 ~ 0
))
```

Race: Q290 (1= White, 2= Black, 3= Hispanic)

```
wvs <- wvs %>% mutate(race = case_when (
  Q290 == 840001 ~ 1,
  Q290 == 840002 ~ 2,
```

```
  Q290 == 840004 ~ 3
))
```

Interest in politics: Q199- only rename

```
wvs <- wvs %>% rename(interest = Q199)
```

## Simple t- test

We can start off with a very simple, naive analysis: do college educated individuals vote more than non-college educated individuals?

Let's take a look at the difference in their means:

```
wvs %>% group_by(college) %>% summarize(mean(vote, na.rm = TRUE))
```

```
## # A tibble: 3 x 2
##   college `mean(vote, na.rm = TRUE)`
##     <dbl>                      <dbl>
## 1       0                      0.837
## 2       1                      0.895
## 3      NA                      0.859
```

But we want to know whether these differences are statistically significant or not. This is where the t-test comes in.

```
 wvs %>%
  t_test(vote ~ college)
```

```
## # A tibble: 1 x 10
##   .y.   group1 group2    n1    n2 statistic     df        p    p.adj
## * <chr> <chr>  <chr>  <int> <int>     <dbl>  <dbl>    <dbl>    <dbl>
## 1 vote  0      1      53800 16473     -19.1 28923. 5.45e-81 5.45e-81
## # ... with 1 more variable: p.adj.signif <chr>
```

The advantage with this package is that you can combine it with dplyr. Perhaps we hypothesize that those that this relationship is not significant among those that are very interested in politics.

```
wvs %>% group_by(interest) %>%  t_test(vote ~ college)
```

```
## # A tibble: 5 x 11
##   interest .y.   group1 group2    n1    n2 statistic     df        p    p.adj
## *    <dbl> <chr> <chr>  <chr>  <int> <int>     <dbl>  <dbl>    <dbl>    <dbl>
## 1        1 vote  0      1       5287  2357     -9.02  5892. 2.46e-19 2.46e-19
## 2        2 vote  0      1      16795  6135    -13.6  12688. 1.38e-41 1.38e-41
## 3        3 vote  0      1      16958  5100     -6.64  8041. 3.28e-11 3.28e-11
## 4        4 vote  0      1      14400  2794     -2.61  3638. 9   e- 3 9   e- 3
## 5       NA vote  0      1        360    87     -1.62   106. 1.08e- 1 1.08e- 1
## # ... with 1 more variable: p.adj.signif <chr>
```

## Regression

But we know we can't simply compare the averages of college/non-college individuals. One quick solution is to control for other variables that we think may matter in explaining this relationship: income (though post-treatment), race, sex, father's education, urban/rural, and political interest.

Since our outcome variable (vote) is binary, we enter into the debate of whether we should use logit or linear probability models. For the sake of simplicity, we'll use a linear model.

*Note, the command to use a logit instead of LPM:*

```
glm(vote ~ college + income + father_educ + sex + urban ,
    data = wvs,
    family = binomial)
```

Some things to keep in mind: we have a few categorical variables in this regression. For now, we can imagine that 'income' is continuous, but political interest is not (very, somewhat, not very, not at all). When we encounter such variables, there is one detail we need to include in the regression to tell R not to consider it as continuous (a "one unit increase" in that variable doesn't have much meaning). Instead, we create a dummy for each level of that variable (one for "Very Interested", one for "Somewhat interested", etc.).

```
lm(vote ~ college + income + father_educ + sex + urban +  factor(interest), data = wvs) %>% summary()
```

```
##
## Call:
## lm(formula = vote ~ college + income + father_educ + sex + urban +
##     factor(interest), data = wvs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9585  0.0907  0.1309  0.1536  0.2633
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.8903027  0.0062874 141.600  < 2e-16 ***
## college              0.0531705  0.0038903  13.668  < 2e-16 ***
## income               0.0004436  0.0007544   0.588   0.5565
## father_educ         -0.0037899  0.0008453  -4.484 7.35e-06 ***
## sex                  0.0062935  0.0030312   2.076   0.0379 *
## urban                0.0055921  0.0032440   1.724   0.0847 .
## factor(interest)2 -0.0239260  0.0051503  -4.646 3.40e-06 ***
## factor(interest)3 -0.0423271  0.0052241  -8.102 5.50e-16 ***
## factor(interest)4 -0.1237483  0.0054540 -22.689  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3512 on 54805 degrees of freedom
##   (16053 observations deleted due to missingness)
## Multiple R-squared:  0.01999,    Adjusted R-squared:  0.01985
## F-statistic: 139.7 on 8 and 54805 DF,  p-value: < 2.2e-16
```

## Fixed effects

Of course, we are only capturing some of the things that may explain voter turnout besides college education. Some of those factors we should also take into account are country-level characteristics. Perhaps people are more generally less likely to vote in one country than in the other because of the quality of institutions, trust in elections, political crises, etc. We can control for these factors by including *country fixed effects*.

One way to do that in R is through the "least squared dummy variable" approach, which simply consist of making a dummy for each country.

```
lm(vote ~ college + factor(interest) + income + father_educ + sex + urban + factor(B_COUNTRY_ALPHA),
    data = wvs) %>% summary()
```

```
##
## Call:
## lm(formula = vote ~ college + factor(interest) + income + father_educ +
##     sex + urban + factor(B_COUNTRY_ALPHA), data = wvs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08507  0.01162  0.08491  0.16560  0.69570
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.8440959  0.0152629  55.304  < 2e-16 ***
## college                      0.0493417  0.0037569  13.134  < 2e-16 ***
## factor(interest)2           -0.0228784  0.0049362  -4.635 3.58e-06 ***
## factor(interest)3           -0.0608552  0.0051113 -11.906  < 2e-16 ***
## factor(interest)4           -0.1295501  0.0053535 -24.199  < 2e-16 ***
## income                       0.0009087  0.0007295   1.246 0.212934
## father_educ                 -0.0116731  0.0009021 -12.941  < 2e-16 ***
## sex                          0.0086977  0.0028710   3.029 0.002451 **
## urban                       -0.0097362  0.0034542  -2.819 0.004824 **
## factor(B_COUNTRY_ALPHA)ARG   0.1779973  0.0186435   9.547  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)AUS   0.1884900  0.0163304  11.542  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)BGD   0.1312608  0.0175076   7.497 6.61e-14 ***
## factor(B_COUNTRY_ALPHA)BOL   0.1851147  0.0160520  11.532  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)BRA   0.1262713  0.0168911   7.476 7.80e-14 ***
## factor(B_COUNTRY_ALPHA)CHL   0.0567943  0.0180717   3.143 0.001675 **
## factor(B_COUNTRY_ALPHA)COL   0.0713318  0.0168609   4.231 2.33e-05 ***
## factor(B_COUNTRY_ALPHA)CYP   0.1427233  0.0181385   7.869 3.65e-15 ***
## factor(B_COUNTRY_ALPHA)DEU   0.1449446  0.0166339   8.714  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)ECU   0.2269759  0.0172386  13.167  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)EGY  -0.1178964  0.0182122  -6.473 9.66e-11 ***
## factor(B_COUNTRY_ALPHA)ETH  -0.1674933  0.0173882  -9.633  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)GRC   0.2080737  0.0173370  12.002  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)HKG  -0.0868497  0.0159267  -5.453 4.97e-08 ***
## factor(B_COUNTRY_ALPHA)IDN   0.1779262  0.0153402  11.599  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)IRN   0.1282315  0.0164079   7.815 5.58e-15 ***
## factor(B_COUNTRY_ALPHA)IRQ  -0.0147290  0.0169756  -0.868 0.385587
## factor(B_COUNTRY_ALPHA)JOR  -0.1109968  0.0171118  -6.487 8.86e-11 ***
## factor(B_COUNTRY_ALPHA)JPN   0.1167182  0.0172995   6.747 1.53e-11 ***
## factor(B_COUNTRY_ALPHA)KAZ   0.0518838  0.0176137   2.946 0.003224 **
## factor(B_COUNTRY_ALPHA)KGZ   0.1160221  0.0172656   6.720 1.84e-11 ***
## factor(B_COUNTRY_ALPHA)KOR   0.1596688  0.0168743   9.462  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)LBN   0.0358888  0.0168482   2.130 0.033166 *
## factor(B_COUNTRY_ALPHA)MEX   0.1355506  0.0163889   8.271  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)MMR  -0.3583612  0.0174670 -20.516  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)MYS  -0.0166556  0.0168061  -0.991 0.321669
## factor(B_COUNTRY_ALPHA)NGA   0.0513451  0.0170812   3.006 0.002649 **
## factor(B_COUNTRY_ALPHA)NIC   0.0123685  0.0176194   0.702 0.482691
## factor(B_COUNTRY_ALPHA)NZL   0.1885109  0.0185362  10.170  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)PAK   0.0268338  0.0162506   1.651 0.098694 .
## factor(B_COUNTRY_ALPHA)PER   0.2118369  0.0166907  12.692  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)PHL   0.1295456  0.0171961   7.533 5.02e-14 ***
## factor(B_COUNTRY_ALPHA)PRI  -0.0712940  0.0187369  -3.805 0.000142 ***
## factor(B_COUNTRY_ALPHA)ROU   0.1712395  0.0174000   9.841  < 2e-16 ***
```

```
## factor(B_COUNTRY_ALPHA)RUS  0.0693587  0.0163803   4.234 2.30e-05 ***
## factor(B_COUNTRY_ALPHA)SRB  0.1605132  0.0183281   8.758  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)THA  0.0306093  0.0170043   1.800 0.071852 .
## factor(B_COUNTRY_ALPHA)TJK  0.1223491  0.0173120   7.067 1.60e-12 ***
## factor(B_COUNTRY_ALPHA)TUN -0.1418719  0.0172746  -8.213  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)TUR  0.1863142  0.0155461  11.985  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)TWN  0.1708968  0.0171099   9.988  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)UKR  0.1545805  0.0177591   8.704  < 2e-16 ***
## factor(B_COUNTRY_ALPHA)USA  0.0880571  0.0156467   5.628 1.83e-08 ***
## factor(B_COUNTRY_ALPHA)VNM -0.1455380  0.0178014  -8.176 3.01e-16 ***
## factor(B_COUNTRY_ALPHA)ZWE  0.0537157  0.0175983   3.052 0.002272 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3314 on 54760 degrees of freedom
##   (16053 observations deleted due to missingness)
## Multiple R-squared:  0.1281, Adjusted R-squared:  0.1272
## F-statistic: 151.7 on 53 and 54760 DF,  p-value: < 2.2e-16
```

*Note: If you are working with panel data, you can use the plm package for more options*

## Interaction terms

Let's return to that hypothesis that perhaps college does not influence vote turnout among people that are very interested in politics. In other words, college has a different effect for people with different levels of interest. For this, we can implement an interaction term between those two variables:

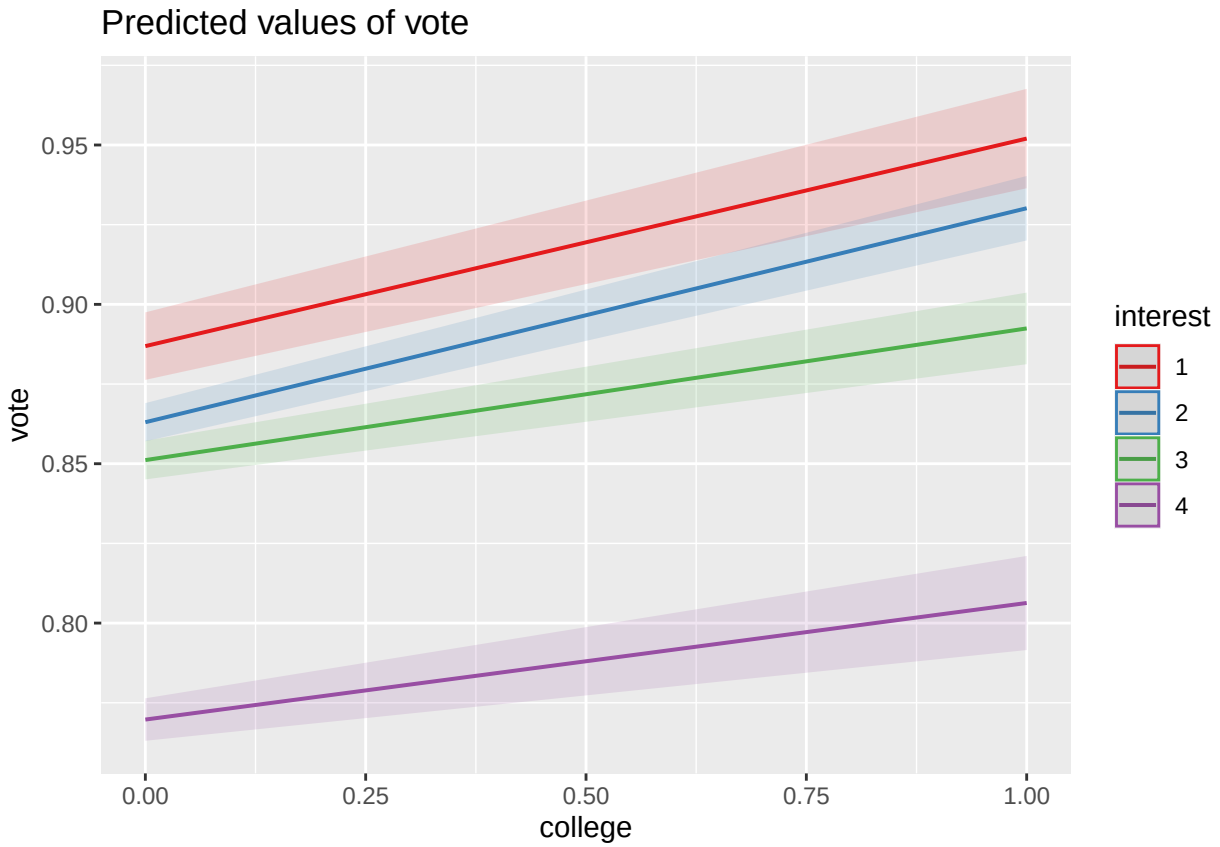
```
mod <- lm(vote ~ college + factor(interest) + income + father_educ + sex + urban + college*factor(intere

summary(mod)
```

```
##
## Call:
## lm(formula = vote ~ college + factor(interest) + income + father_educ +
##     sex + urban + college * factor(interest), data = wvs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96637  0.08274  0.13469  0.15154  0.26020
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.8866706  0.0068488 129.463  < 2e-16 ***
## college                     0.0651253  0.0096521   6.747 1.52e-11 ***
## factor(interest)2          -0.0238823  0.0061864  -3.860 0.000113 ***
## factor(interest)3          -0.0357592  0.0062135  -5.755 8.71e-09 ***
## factor(interest)4          -0.1171684  0.0063654 -18.407  < 2e-16 ***
## income                      0.0004381  0.0007544   0.581 0.561390
## father_educ                -0.0037674  0.0008452  -4.457 8.32e-06 ***
## sex                         0.0060980  0.0030317   2.011 0.044288 *
## urban                       0.0054140  0.0032443   1.669 0.095171 .
## college:factor(interest)2   0.0020145  0.0111514   0.181 0.856645
## college:factor(interest)3  -0.0238119  0.0114366  -2.082 0.037339 *
## college:factor(interest)4  -0.0285511  0.0125259  -2.279 0.022649 *
## ---
```

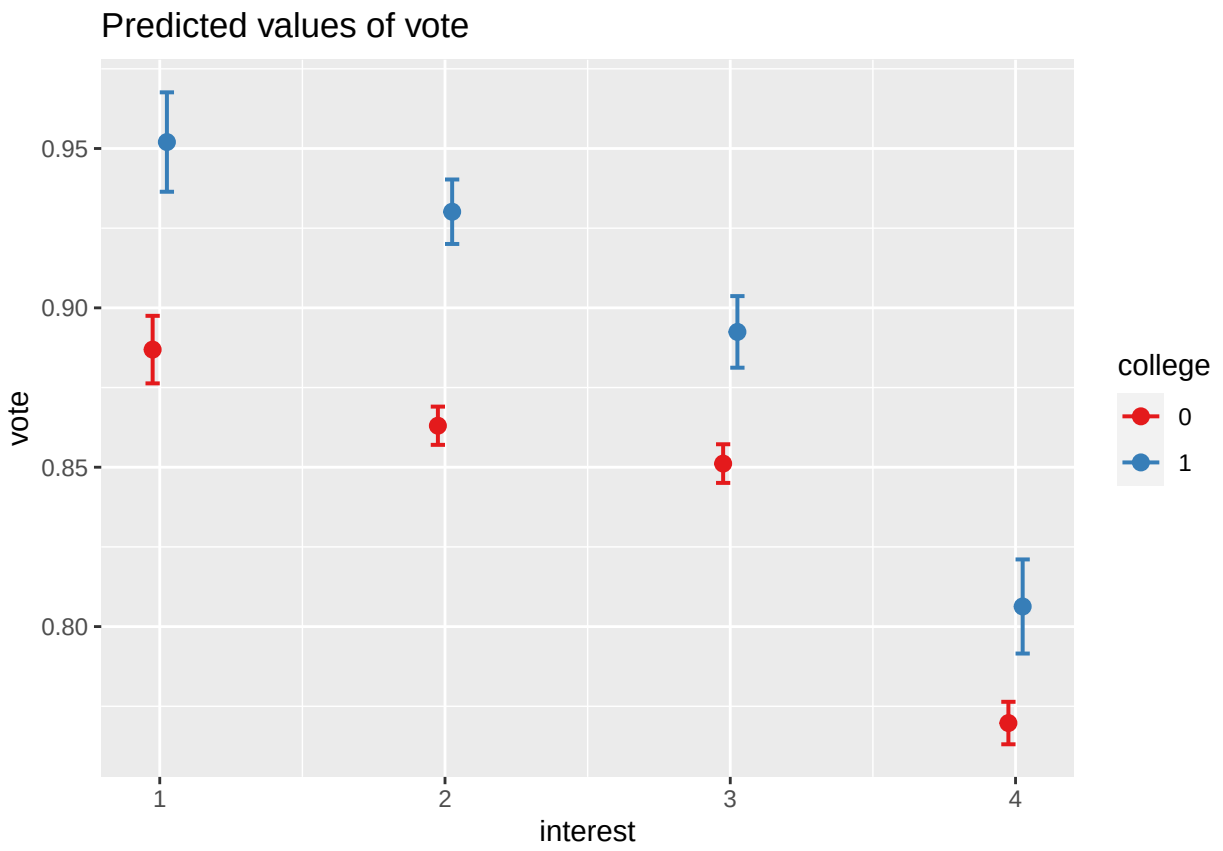
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3512 on 54802 degrees of freedom
##   (16053 observations deleted due to missingness)
## Multiple R-squared:  0.02026,    Adjusted R-squared:  0.02006
## F-statistic:   103 on 11 and 54802 DF,  p-value: < 2.2e-16
```

Regression results of models with interaction terms are not easy to interpret. Instead, we can turn to graphs for some help. We'll use the sjPlot package for this.

```
plot_model(mod, type = "pred", terms = c("college", "interest[1,2,3,4]"))
```




```
plot_model(mod, type = "pred", terms = c("interest[1,2,3,4]", "college"))
```




## Matching

Because regression is not the best tool for causal inference, let's now turn to matching. For this part, we'll work only with the U.S. and keep only the variables we need (and complete observations):

```
wvs_match <- wvs %>% filter(B_COUNTRY_ALPHA =="USA") %>%
  select(vote, college, sex, income, father_educ, race) %>%
  drop_na()
```

With matching, we are trying to replicate an experimental scenario where the only thing that differs between two units is the fact that one is treated (college eduaction) and the other is not. By making them as similar as possible, we can then attribute the difference in voting to the college education. In an ideal world, we find two individuals that are exactly the same to each other (in a set of covariates you think are relevant) but one of those individuals went to college and the other did not. Since they have the same income level, race, and family background, we can then attribute the effect of higher voting rates to college attendance.

How plausible is it to find two people that are exactly the same in many dimensions? Not very plausible. Exact matching is impossible, so we turn to other techniques. One of the most common ones is propensity score matching. The propensity score basically reduces all of the covariates into one dimension, creating an index that indicates your probability of attending college. We then use this index to find matches. People who went to college (*the treatment group*) are matched with others with very similar propensity scores, but who did not go to college (*the control group*). Note, however, that at this stage we still face the same assumptions as OLS: that we are correctly specifying the model that determines college attendance.

Even within the world of propensity score matching, there are many different alternatives (with or without replacement, use of caliper, greedy vs optimal, etc.). In many of these cases, you will be facing a bias-variance tradeoff: one of them increases the amount of observations you use, the other increases the quality of the

matches.

Importantly, it is OK to test out different methods *BEFORE* you estimate your effects. This is done to ensure that you have a balanced dataset before your estimation. A balanced dataset is one where the differences between the treatment and control groups (in your selected covariates) are minimal. One where the treated/control group look very much alike across these selected characteristics. If you have not achieved balance with your selected matching technique/propensity score model, you are expected to go back and tweak it (before peeking at the effects!).

Select covariate for match –> Run matching algorithm –> Balance? If not, return to step 1. If balanced –> Estimate effect

## Balance check

As a first step in our analysis, let's check that the two groups (college and non-college) are indeed different to start with. We can do so by showing that covariates *are not* balanced.

To check balance, run a regression where the treatment (college) is the outcome and the covariates are the predictore (use: income, race, sex, and father_educ).

When our data is *unbalanced*, predictors *should* explain some of the variation.

```
lm(college ~ income + factor(race) + sex + father_educ,
   data = wvs_match) %>% summary()
```

```
##
## Call:
## lm(formula = college ~ income + factor(race) + sex + father_educ,
##     data = wvs_match)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0093 -0.3691 -0.1355  0.4217  1.0981
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.242242   0.035953  -6.738 2.09e-11 ***
## income         0.059342   0.005523  10.745  < 2e-16 ***
## factor(race)2 -0.098853   0.038628  -2.559   0.0106 *
## factor(race)3 -0.059641   0.027219  -2.191   0.0286 *
## sex            0.085137   0.020149   4.225 2.49e-05 ***
## father_educ    0.086454   0.005682  15.216  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4435 on 2025 degrees of freedom
## Multiple R-squared:  0.2037, Adjusted R-squared:  0.2018
## F-statistic: 103.6 on 5 and 2025 DF,  p-value: < 2.2e-16
```

Let's check how the averages differ:

```
wvs_match %>% group_by(college) %>% summarize(mean(income), mean(sex), mean(father_educ))
```

```
## # A tibble: 2 x 4
##   college `mean(income)` `mean(sex)` `mean(father_educ)`
##     <dbl>          <dbl>       <dbl>               <dbl>
## 1       0           4.63       0.487                3.43
## 2       1           5.76       0.631                4.77
```

## 1:1 matching on propensity score

Propensity score matching is the most common method of matching. Propensity scores indicate how likely it is for a unit to be treated. To construt the propensity score, we need to build a model that predicts treatment (in this case, attending college).

Once we've constructed that propensity score, there are several alternatives as to which control unit to pick for each treated unit. In this case, we'll look for the closest match among the control, and we'll use that control only once. This means we'll match *without* replacement. Any control units that aren't matched with a treated unit will be discarded.

```
m.out1 <- matchit(college ~ income + sex + father_educ + factor(race), data = wvs_match, method = "near
```

Let's print out some of the details on the matching method:

```
m.out1
```

```
## A matchit object
##  - method: 1:1 nearest neighbor matching without replacement
##  - distance: Propensity score
##              - estimated with logistic regression
##  - number of obs.: 2031 (original), 1784 (matched)
##  - target estimand: ATT
##  - covariates: income, sex, father_educ, factor(race)
```

## Checking balance

Before estimating the ATT, we need to make sure we have a balanced sample. The whole purpose of matching is to replicate an experiment, where treated units are compared to very similar untreated units. This table allows us to see some of the differences in means between the treat/control across the set of covariates we chose. The first columns are the easiest to intepret, as they show you the averages in the two groups and the standardized mean differences (the closer to zero, the better).

```
summary(m.out1, un = FALSE)
```

```
##
## Call:
## matchit(formula = college ~ income + sex + father_educ + factor(race),
##     data = wvs_match, method = "nearest")
##
## Summary of Balance for Matched Data:
##                  Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance                 0.5559        0.4044          0.6966     1.6043
## income                   5.7578        5.1457          0.3646     1.1422
## sex                      0.6312        0.5460          0.1766          .
## father_educ              4.7679        3.6973          0.5621     1.7305
## `factor(race)`1          0.8307        0.7478          0.2212          .
## `factor(race)`2          0.0471        0.0751         -0.1323          .
## `factor(race)`3          0.1222        0.1771         -0.1677          .
##                  eCDF Mean eCDF Max Std. Pair Dist.
## distance            0.1657   0.3408          0.6967
## income              0.0612   0.2074          0.9014
## sex                 0.0852   0.0852          0.9526
## father_educ         0.1355   0.2623          0.7504
## `factor(race)`1     0.0830   0.0830          0.7354
## `factor(race)`2     0.0280   0.0280          0.4499
## `factor(race)`3     0.0549   0.0549          0.6538
```

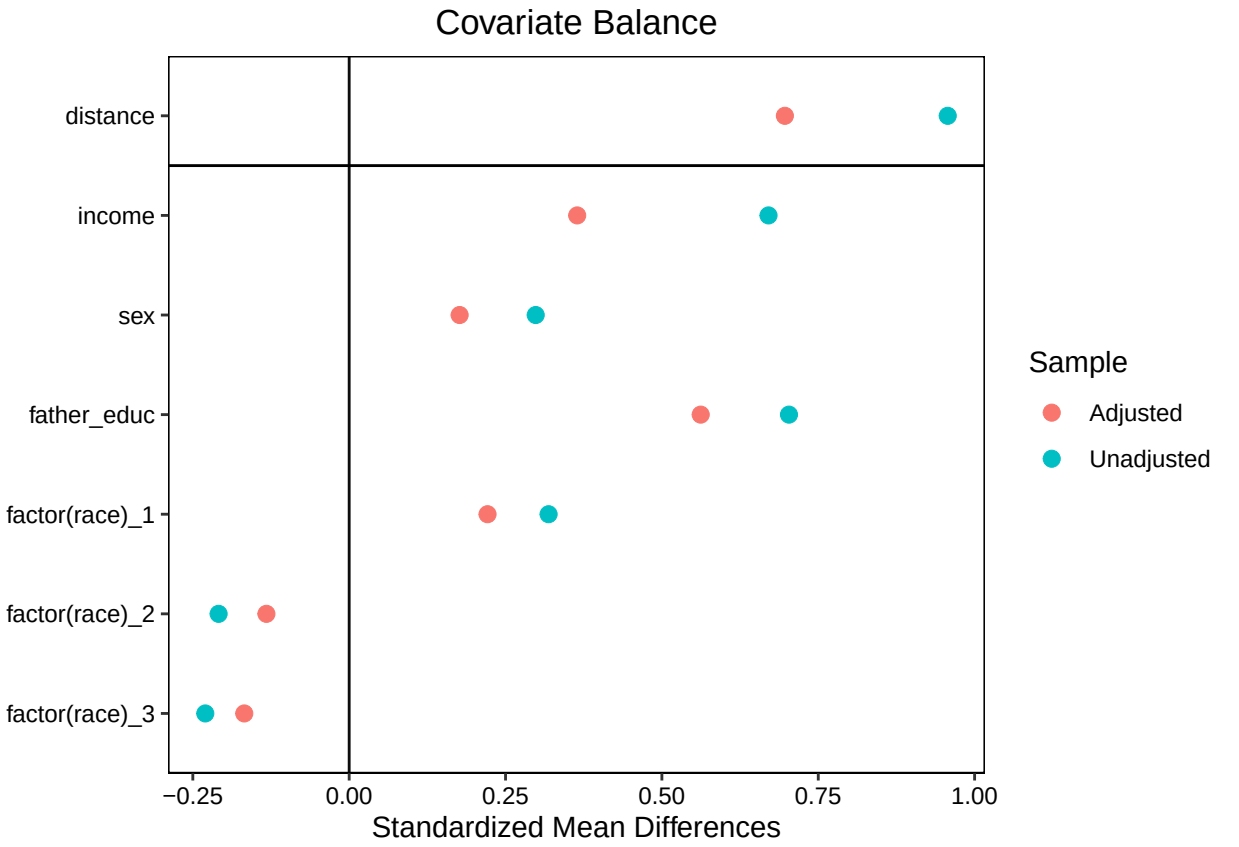
```
##
## Sample Sizes:
##           Control Treated
## All          1139     892
## Matched       892     892
## Unmatched     247       0
## Discarded       0       0
```

Simply by comparing the means, we can see that we haven't done a great job achieving balance. For instance, Whites represent 89% of the treated sample and 74% of the control sample; men represent almost 70% of the treated sample and only 45% of the control.

There are many options to visualize balance checks; the 'cobalt' package is helpful for this. The following graph compares the standardized means before and after matching. Ideally, the red dots (after matching) should be very close to zero. This would mean that the differences in the averages of those variables have dissapeared. In other words, that we have made the treatment and control as similar as possible.

Here, we see that balance has actually become worse in many cases! Notice how the red dots are further away from zero than the blue dots (before matching). This is why it is so important to check balance before running our analysis.

```
love.plot(m.out1, binary = "std")
```




The table at the bottom of the 'summary' command is also important: it tells us how many of our units have been used in the matching. Notice there are 176 units that are left unused. It's important to check that we haven't lost too many observations, and that we still have a decent sized n to carry out the analysis.

### See matched data:

To better understand what's happening behind Matching, it may be useful to visualize some specific cases. Here, we'll take a look at the first two matches (case 7 was matched with case 436). We see they are both men, and both white, yet the level of their father's education is very different.

```
head(m.out1$match.matrix)
```

```
##    [,1]
## 11 "33"
## 12 "1099"
## 16 "1732"
## 27 "1768"
## 29 "66"
## 32 "546"
```

```
i <- rownames(wvs_match) %in% c(11,33)
wvs_match[i,]
```

```
## # A tibble: 2 x 6
##    vote college   sex income father_educ  race
##   <dbl>   <dbl> <dbl>  <dbl>       <dbl> <dbl>
## 1     1       1     0      4           4     3
## 2     0       0     1      2           4     3
```

### Caliper

But what if the closest neighbor is actually quite different from the treated unit? To avoid this, we can use caliper: we impose a restriction on how far away the match can be. We establish that the propensity score of the match can be no more than x standard deviations away. Some researchers recommend using a caliper of 0.2 standard deviations. (If a control's propensity score is more than 0.2 standard deviations away from the treatment's propensity score, then they won't match). By doing this, we are ensuring better matches, but we may be losing some observations: if there are no control units less than 0.2 SD's away, then that observations is dropped.

The procedure is almost the same as above, except we include the option of caliper at the end:

```
m.out.caliper <- matchit(college ~ income + factor(race) + sex + father_educ, data = wvs_match, method =
```

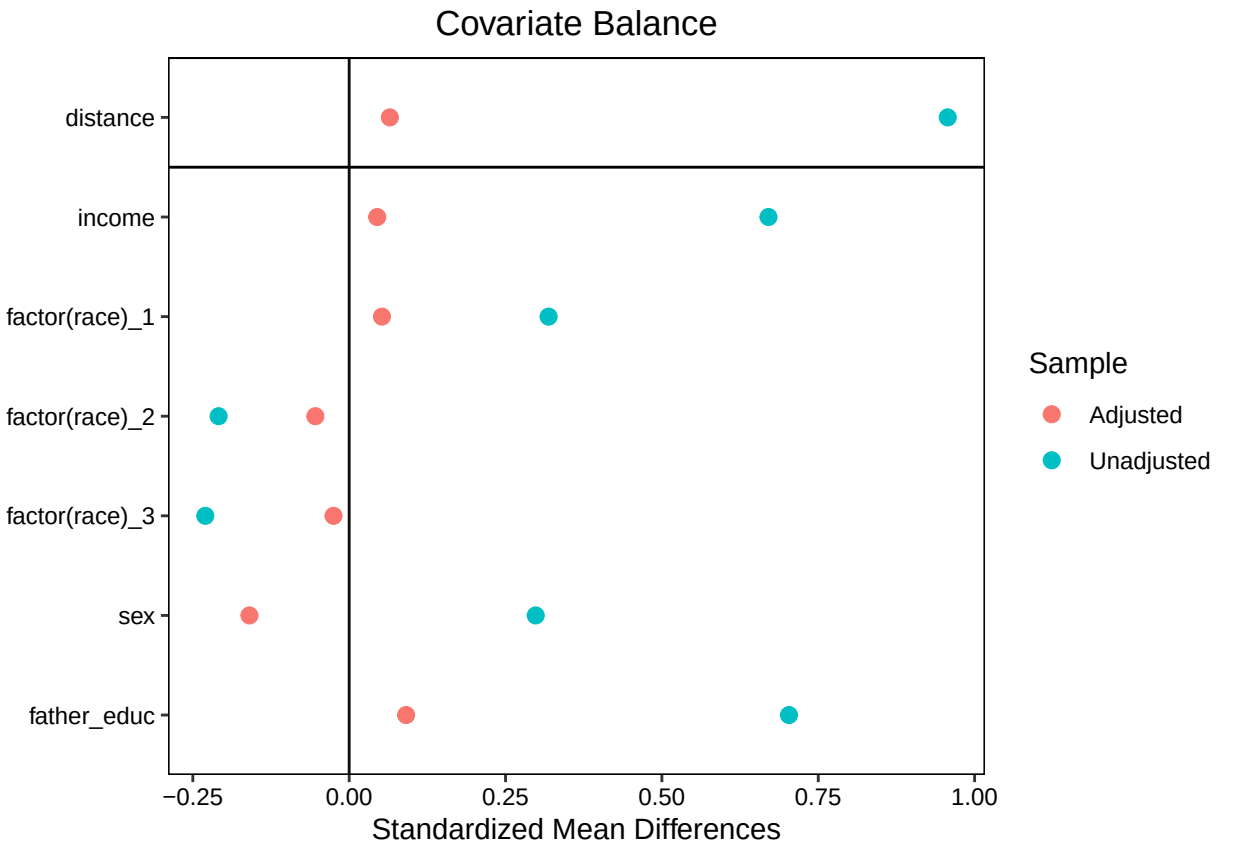
```
summary(m.out.caliper, un = FALSE)
```

```
##
## Call:
## matchit(formula = college ~ income + factor(race) + sex + father_educ,
##     data = wvs_match, method = "nearest", distance = "glm", caliper = 0.2)
##
## Summary of Balance for Matched Data:
##                 Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance               0.4691        0.4549          0.0650     1.1698
## income                 5.3863        5.3110          0.0448     0.9639
## `factor(race)`1        0.7774        0.7578          0.0524          .
## `factor(race)`2        0.0671        0.0786         -0.0541          .
## `factor(race)`3        0.1555        0.1637         -0.0250          .
## sex                    0.5270        0.6039         -0.1594          .
## father_educ            4.2029        4.0295          0.0911     1.2462
##                 eCDF Mean eCDF Max Std. Pair Dist.
## distance           0.0150   0.0638          0.0652
## income             0.0079   0.0262          0.5342
```

```
## `factor(race)`1    0.0196   0.0196          0.3666
## `factor(race)`2    0.0115   0.0115          0.3168
## `factor(race)`3    0.0082   0.0082          0.2948
## sex                0.0769   0.0769          0.5665
## father_educ        0.0276   0.0622          0.3866
##
## Sample Sizes:
##           Control Treated
## All          1139     892
## Matched       611     611
## Unmatched     528     281
## Discarded       0       0
```

This is looking much better!

```
love.plot(m.out.caliper, binary = "std")
```




## Full Matching

Another alternative matching technique is the *full matching*, where each treated unit is matched with at least one other control unit, and every control to a treated unit. From Stuart and Green (2008): "full matching occurs"wherein all units, both treatment and control (i.e., the "full" sample), are assigned to a subclass and receive at least one match. The matching is optimal in the sense that that sum of the absolute distances between the treated and control unitsin each subclass are as small as possible"."Full matching, first developed by Rosenbaum (1991) and illustrated by Hansen (2004), uses all available individuals in the data by grouping the individuals into a series of matched sets (subclasses), with each matched set containing at least 1 treated individual (who received the treatment of interest) and at least 1 comparison individual (who

did not). Full matching forms these matched sets in an optimal way, such that treated individuals who have many comparison individuals who are similar (on the basis of the propensity score) will be grouped with many comparison individuals, whereas treated individuals with few similar comparison individuals will be grouped with relatively fewer comparison individuals. The method is thus more flexible than traditional k:1 matching, in which each treated individual is required to be matched with the same number of comparison individuals (k), regardless of whether each individual actually has k good matches (Ming & Rosenbaum, 2000)." .footnote[https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5784842/]

Again, we set this difference in the specifications after the model:

```
m.out2 <- matchit(college ~ income + factor(race) + sex + father_educ, data = wvs_match,
method = "full", distance = "glm")
```

Let's get the basics:

```
summary(m.out2, un= FALSE)
```

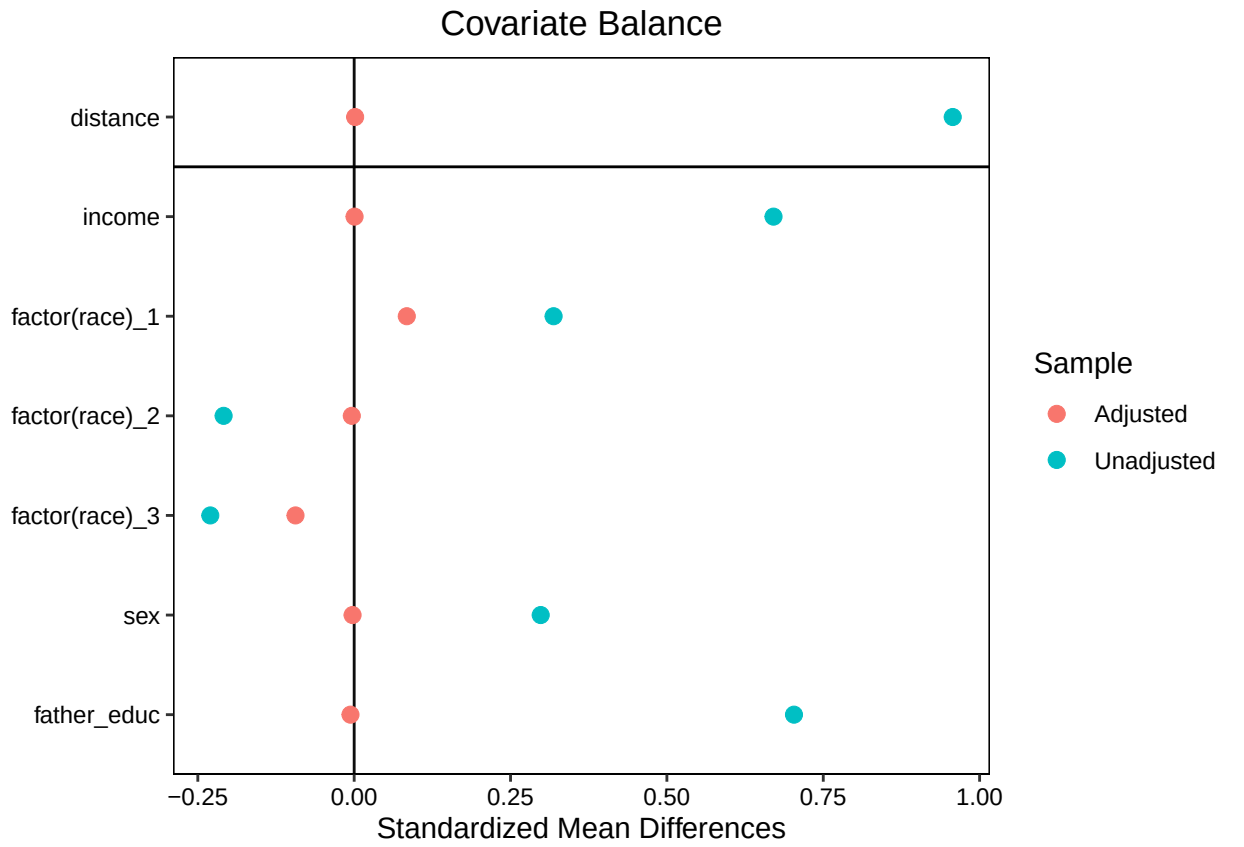
```
##
## Call:
## matchit(formula = college ~ income + factor(race) + sex + father_educ,
##     data = wvs_match, method = "full", distance = "glm")
##
## Summary of Balance for Matched Data:
##                  Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance                 0.5559        0.5556          0.0015     0.9979
## income                   5.7578        5.7568          0.0006     0.9660
## `factor(race)`1          0.8307        0.7992          0.0841          .
## `factor(race)`2          0.0471        0.0479         -0.0038          .
## `factor(race)`3          0.1222        0.1529         -0.0938          .
## sex                      0.6312        0.6324         -0.0026          .
## father_educ              4.7679        4.7792         -0.0059     1.0021
##                  eCDF Mean eCDF Max Std. Pair Dist.
## distance            0.0015   0.0235          0.0039
## income              0.0056   0.0254          0.2075
## `factor(race)`1     0.0315   0.0315          0.5663
## `factor(race)`2     0.0008   0.0008          0.4218
## `factor(race)`3     0.0307   0.0307          0.4764
## sex                 0.0012   0.0012          0.8803
## father_educ         0.0087   0.0234          0.3024
##
## Sample Sizes:
##               Control Treated
## All           1139.       892
## Matched (ESS)  136.04     892
## Matched       1139.       892
## Unmatched        0.         0
## Discarded        0.         0
```

Balance is much better here. And, as the bottom table suggests, we've used *all* the observations. Some control units have been used more than once.

*Note on ESS (effective sample size): "Units may be weighted in such a way that they contribute less to the sample than would unweighted units, so the effective sample size (ESS) of the full matching weighted sample may be lower than even that of 1:1 pair matching."

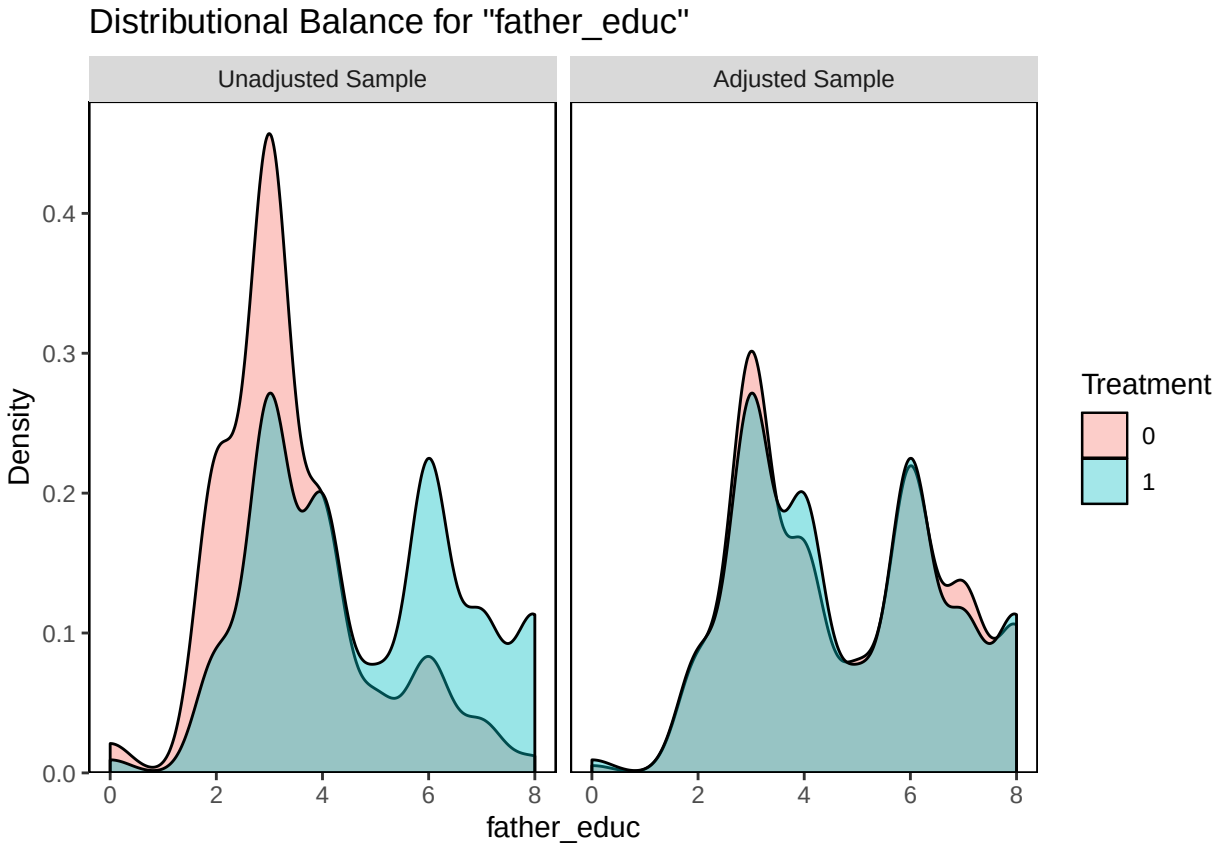
Let's visualize our balance with new graphs, again from the 'cobalt' package:

```
love.plot(m.out2, binary = "std")
```

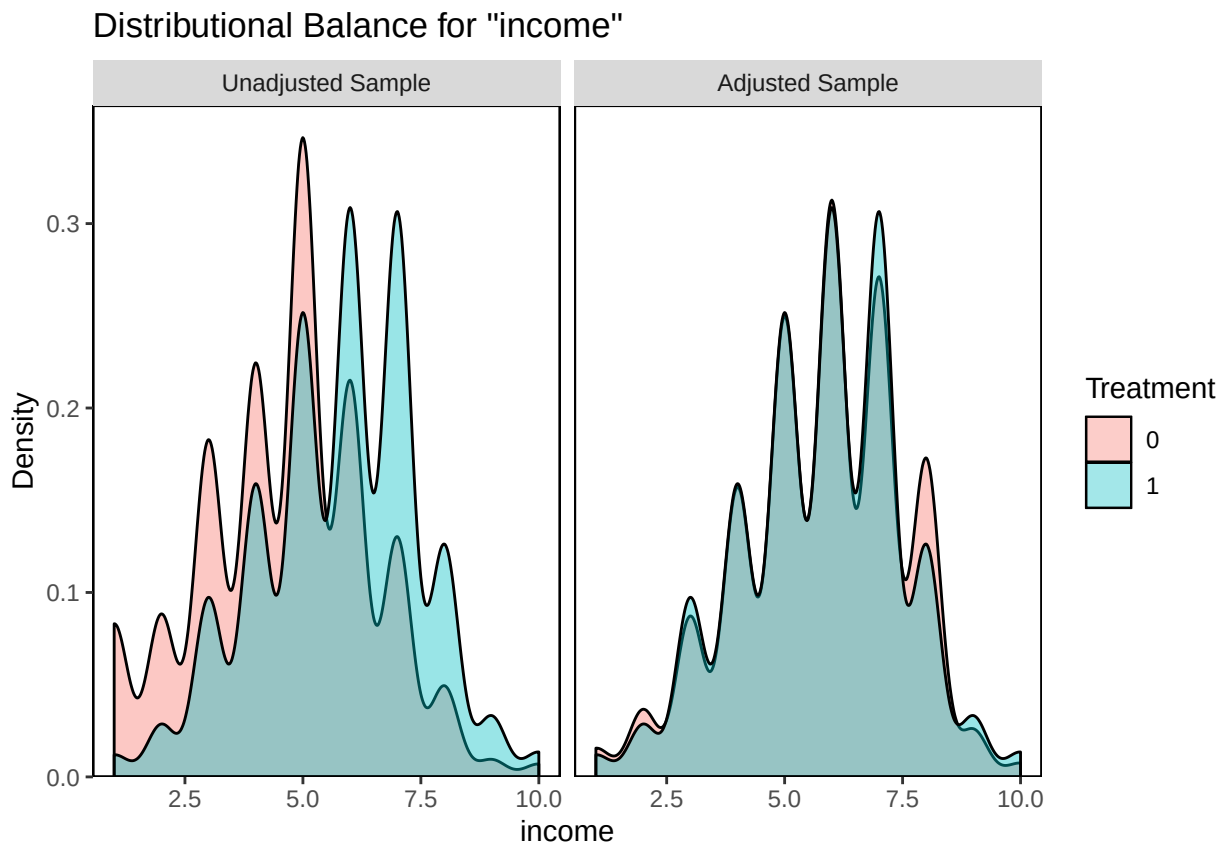



And we can plot individual variables to see how they changed before and after matching:

```
bal.plot(m.out2, var.name = "father_educ", which = "both")
```




```
bal.plot(m.out2, var.name = "income", which = "both")
```




### Estimating the effect of college attendance

Now that we are more confident that our matching technique worked, we can proceed to estimate the effect of college education on voting. Recall that in most cases, we will be estimating the Average Treatment Effect among the Treated. What this means is that we are compare the effect of going to college **among those that would have possibly gone to college**.

The only thing we need to do before running the regression is extracting the matched dataset.

```
m.data <- match.data(m.out2)
```

Let's take a look at the dataset. Notice it has a few extra variables (like weights).

```
glimpse(m.data)
```

```
## Rows: 2,031
## Columns: 9
## $ vote        <dbl> 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1~
## $ college     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0~
## $ sex         <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0~
## $ income      <dbl> 1, 1, 5, 6, 10, 6, 2, 5, 1, 8, 4, 5, 1, 5, 5, 6, 2, 4, 2, ~
## $ father_educ <dbl> 2, 3, 5, 6, 3, 3, 6, 3, 3, 3, 4, 2, 4, 2, 3, 7, 4, 4, 3, 3~
## $ race        <dbl> 1, 1, 1, 2, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 3, 1, 2, 2, 1, 1~
## $ distance    <dbl> 0.07379312, 0.10813605, 0.48665630, 0.63589549, 0.65196621~
## $ weights     <dbl> 0.11608235, 0.06720557, 0.31922646, 7.66143498, 1.27690583~
## $ subclass    <fct> 161, 340, 319, 358, 151, 521, 356, 536, 309, 563, 329, 7, ~
```

We can do this simply by running a regression and looking at the coefficient for the treatment variable. We

will include the same covariates we included in the propensity score estimation to increase the precision of our estimate and correct for any remaining differences. But, in theory, we have already "controlled" for those covariates in our matching process. The only difference with this regression is that we are including weights.

```
lm(vote ~ college + factor(race) + income + father_educ + sex , data = m.data, weights = weights) %>% su
```

```
##
## Call:
## lm(formula = vote ~ college + factor(race) + income + father_educ +
##     sex, data = m.data, weights = weights)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6546  0.0060  0.0487  0.0953  0.7506
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.764418   0.025696  29.749  < 2e-16 ***
## college        0.058438   0.011815   4.946 8.20e-07 ***
## factor(race)2 -0.028270   0.027751  -1.019   0.3085
## factor(race)3 -0.036274   0.017089  -2.123   0.0339 *
## income         0.022098   0.003535   6.252 4.94e-10 ***
## father_educ   -0.005693   0.003131  -1.818   0.0692 .
## sex            0.058382   0.012349   4.728 2.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.264 on 2024 degrees of freedom
## Multiple R-squared:  0.04971,    Adjusted R-squared:  0.0469
## F-statistic: 17.65 on 6 and 2024 DF,  p-value: < 2.2e-16
```

The results indicate that going to college does indeed have a positive and significant effect on voting.