R Tutorial 8: Propensity Score Matching

In this tutorial we'll analyze the effect of going to Catholic school, as opposed to public school, on student achievement. Because students who attend Catholic school on average are different from students who attend public school, we will use propensity score matching to get more credible causal estimates of Catholic schooling.

UPDATE: Many people have asked for the data used in this tutorial. To get the dataset used below (ecls.csv), please follow the instructions outlined here (https://github.com/sejdemyr/ecls).

To examine the effect of going to Catholic school ("Treated") versus public school ("Control") on student achievement using matching we will go through the following steps:

1. Estimate the propensity score (the probability of being Treated given a set of pre-treatment covariates).
2. Examine the region of common support.
3. Choose and execute a matching algorithm. In this tutorial we'll use nearest neighbor propensity score matching.
4. Examine covariate balance *after* matching.
5. Estimate treatment effects.

In addition, before we implement a matching method, we'll conduct the following analyses using the non-matched data:

- Examine the difference-in-means between Treated and Control for the outcome variable.
- Examine the difference-in-means between Treated and Control for pre-treatment covariates.

Before we start, load a few packages and read in ecls.csv:

```
library(MatchIt)
library(dplyr)
library(ggplot2)

ecls <- read.csv("ecls.csv")
```

1 Pre-analysis using non-matched data

1.1 Difference-in-means: outcome variable

Here is some basic information about public and catholic school students in terms of math achievement. Note that we're using students' standardized math score (`c5r2mtsc_std`) – with a mean of 0 and standard deviation of 1 – as the outcome variable of interest. The independent variable of interest is `catholic` (1 = student went to catholic school; 0 = student went to public school).

```
ecls %>%
  group_by(catholic) %>%
  summarise(n_students = n(),
            mean_math = mean(c5r2mtsc_std),
            std_error = sd(c5r2mtsc_std) / sqrt(n_students))
```

```
## # A tibble: 2 × 4
##   catholic n_students   mean_math  std_error
##      <int>      <int>       <dbl>      <dbl>
## 1        0       9568 -0.03059583 0.01038536
## 2        1       1510  0.19386817 0.02235282
```

Note that the outcome variable has been standardized (mean = 0, sd = 1). This is common in education research. The summary table above indicates that 3rd grade Catholic school students' average math score is more than 20% of a standard deviation higher than that of public school students. This could have been calculated using the non-standardized outcome variable as follows:

```
ecls %>%
  mutate(test = (c5r2mtsc - mean(c5r2mtsc)) / sd(c5r2mtsc)) %>% #this is how the m
ath score is standardized
  group_by(catholic) %>%
  summarise(mean_math = mean(test))
```

```
## # A tibble: 2 × 2
##   catholic   mean_math
##      <int>       <dbl>
## 1        0 -0.03059583
## 2        1  0.19386817
```

The difference-in-means is statistically significant at conventional levels of confidence (as is also evident from the small standard error above):

```
with(ecls, t.test(c5r2mtsc_std ~ catholic))
```

```
##
##  Welch Two Sample t-test
##
## data:  c5r2mtsc_std by catholic
## t = -9.1069, df = 2214.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2727988 -0.1761292
## sample estimates:
## mean in group 0 mean in group 1
##     -0.03059583      0.19386817
```

1.2 Difference-in-means: pre-treatment covariates

We'll work with the following covariates for now:

- `race_white` : Is the student white (1) or not (0)?
- `p5hmage` : Mother's age
- `w3income` : Family income
- `p5numpla` : Number of places the student has lived for at least 4 months
- `w3momed_hsb` : Is the mother's education level high-school or below (1) or some college or more (0)?

Let's calculate the mean for each covariate by the treatment status:

```
ecls_cov <- c('race_white', 'p5hmage', 'w3income', 'p5numpla', 'w3momed_hsb')
ecls %>%
  group_by(catholic) %>%
  select(one_of(ecls_cov)) %>%
  summarise_all(funs(mean(., na.rm = T)))
```

```
## Adding missing grouping variables: `catholic`
```

```
## # A tibble: 2 × 6
##   catholic race_white  p5hmage w3income p5numpla w3momed_hsb
##      <int>      <dbl>    <dbl>    <dbl>    <dbl>       <dbl>
## 1        0  0.5561246 37.56097 54889.16 1.132669   0.4640918
## 2        1  0.7251656 39.57516 82074.30 1.092701   0.2272069
```

What do you see? Take a moment to reflect on what these differences suggest for the relationship of interest (that between Catholic schooling and student achievement).

We can carry out t-tests to evaluate whether these means are statistically distinguishable:

```
lapply(ecls_cov, function(v) {
    t.test(ecls[, v] ~ ecls[, 'catholic'])
})
```

```
## [[1]]
## 
##  Welch Two Sample t-test
## 
## data:  ecls[, v] by ecls[, "catholic"]
## t = -13.453, df = 2143.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1936817 -0.1444003
## sample estimates:
## mean in group 0 mean in group 1 
##       0.5561246       0.7251656 
## 
## 
## [[2]]
## 
##  Welch Two Sample t-test
## 
## data:  ecls[, v] by ecls[, "catholic"]
## t = -12.665, df = 2186.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.326071 -1.702317
## sample estimates:
## mean in group 0 mean in group 1 
##        37.56097        39.57516 
## 
## 
## [[3]]
## 
##  Welch Two Sample t-test
## 
## data:  ecls[, v] by ecls[, "catholic"]
## t = -20.25, df = 1825.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -29818.10 -24552.18
## sample estimates:
## mean in group 0 mean in group 1 
##        54889.16        82074.30 
## 
## 
## [[4]]
## 
##  Welch Two Sample t-test
## 
## data:  ecls[, v] by ecls[, "catholic"]
## t = 4.2458, df = 2233.7, p-value = 2.267e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.02150833 0.05842896
## sample estimates:
## mean in group 0 mean in group 1 
##        1.132669        1.092701
##
##
## [[5]]
##
##  Welch Two Sample t-test
##
## data:  ecls[, v] by ecls[, "catholic"]
## t = 18.855, df = 2107.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2122471 0.2615226
## sample estimates:
## mean in group 0 mean in group 1
##       0.4640918       0.2272069
```

If you don't understand the code for the t-tests above, you can also use:

```
with(ecls, t.test(race_white ~ catholic))  #(repeat for each covariate)
```

```
##
##  Welch Two Sample t-test
##
## data:  race_white by catholic
## t = -13.453, df = 2143.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1936817 -0.1444003
## sample estimates:
## mean in group 0 mean in group 1
##       0.5561246       0.7251656
```

2 Propensity score estimation

We estimate the propensity score by running a logit model (probit also works) where the outcome variable is a binary variable indicating treatment status. What covariates should you include? For the matching to give you a causal estimate in the end, you need to include any covariate that is related to both the treatment assignment and potential outcomes. I choose just a few covariates below—they are unlikely to capture all covariates that should be included. You'll be asked to come up with a potentially better model on your own later.

```
ecls <- ecls %>% mutate(w3income_1k = w3income / 1000)
m_ps <- glm(catholic ~ race_white + w3income_1k + p5hmage + p5numpla + w3momed_hsb
,
            family = binomial(), data = ecls)
summary(m_ps)
```

```
## 
## Call:
## glm(formula = catholic ~ race_white + w3income_1k + p5hmage + 
##     p5numpla + w3momed_hsb, family = binomial(), data = ecls)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.1883  -0.6140  -0.4508  -0.3336   2.5659  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -3.2125519  0.2379826 -13.499  < 2e-16 ***
## race_white   0.3145014  0.0700895   4.487 7.22e-06 ***
## w3income_1k  0.0073038  0.0006495  11.245  < 2e-16 ***
## p5hmage      0.0292168  0.0050771   5.755 8.69e-09 ***
## p5numpla    -0.1439392  0.0912255  -1.578    0.115    
## w3momed_hsb -0.6935868  0.0743207  -9.332  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 7701.3  on 9266  degrees of freedom
## Residual deviance: 7168.8  on 9261  degrees of freedom
##   (1811 observations deleted due to missingness)
## AIC: 7180.8
## 
## Number of Fisher Scoring iterations: 5
```

Using this model, we can now calculate the propensity score for each student. It is simply the student's predicted probability of being Treated, given the estimates from the logit model. Below, I calculate this propensity score using `predict()` and create a dataframe that has the propensity score as well as the student's actual treatment status.

```
prs_df <- data.frame(pr_score = predict(m_ps, type = "response"),
                     catholic = m_ps$model$catholic)
head(prs_df)
```

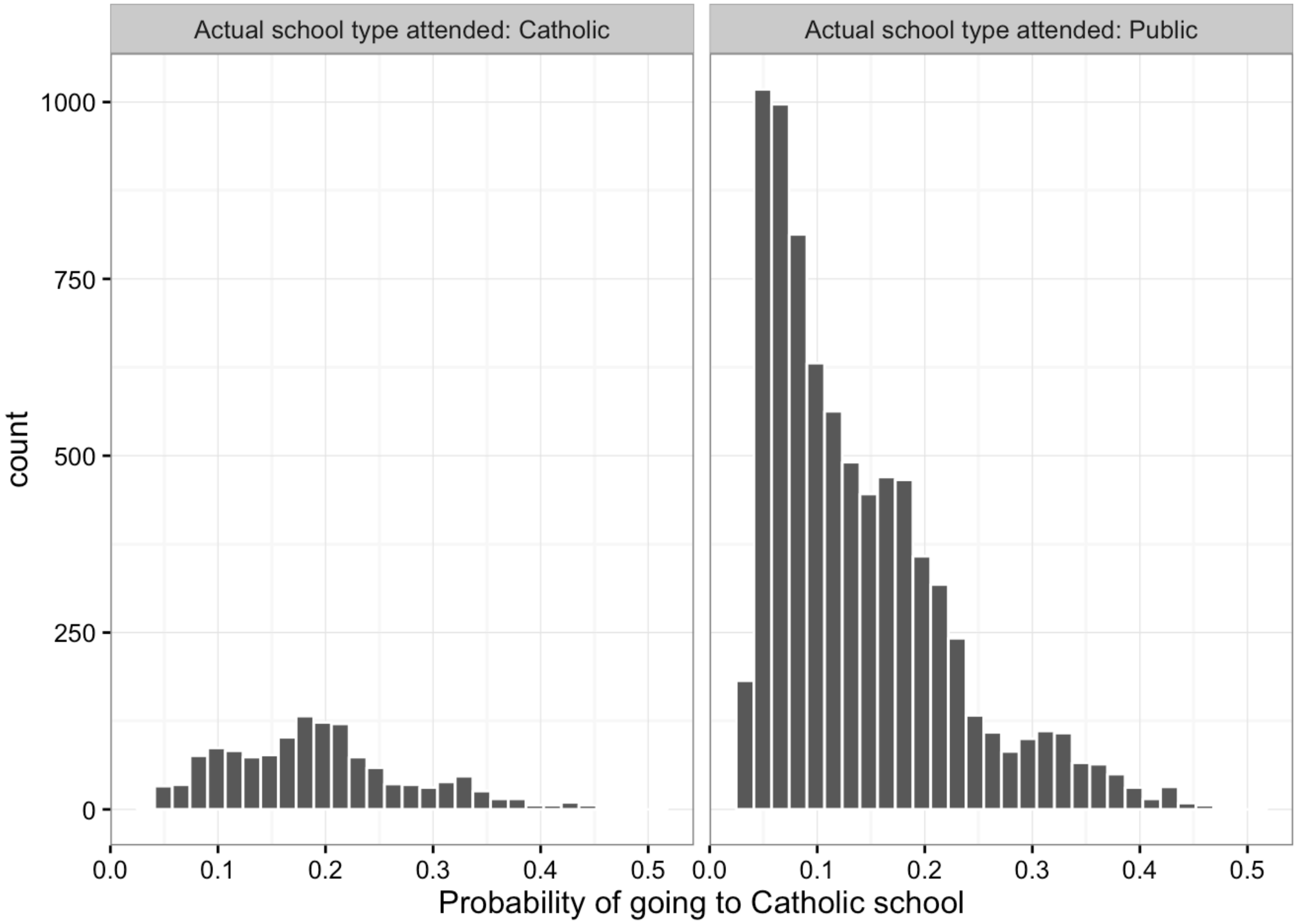
```
##    pr_score catholic
## 1 0.2292928        0
## 2 0.1801360        0
## 4 0.2092957        0
## 5 0.2154022        1
## 6 0.3604931        0
## 7 0.1080608        0
```

2.1 Examining the region of common support

After estimating the propensity score, it is useful to plot histograms of the estimated propensity scores by treatment status:

```
labs <- paste("Actual school type attended:", c("Catholic", "Public"))
prs_df %>%
  mutate(catholic = ifelse(catholic == 1, labs[1], labs[2])) %>%
  ggplot(aes(x = pr_score)) +
  geom_histogram(color = "white") +
  facet_wrap(~catholic) +
  xlab("Probability of going to Catholic school") +
  theme_bw()
```

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



3 Executing a matching algorithm

A simple method for estimating the treatment effect of Catholic schooling is to restrict the sample to observations within the region of common support, and then to divide the sample within the region of common support into 5 quintiles, based on the estimated propensity score. Within each of these 5 quintiles, we can then estimate the mean difference in student achievement by treatment status. Rubin and others have argued that this is sufficient to eliminate 95% of the bias due to confounding of treatment status with a covariate.

However, most matching algorithms adopt slightly more complex methods. The method we use below is to find pairs of observations that have very similar propensity scores, but that differ in their treatment status. We use the package `MatchIt` for this. This package estimates the propensity score in the background and then matches observations based on the method of choice ("nearest" in this case).

```
ecls_nomiss <- ecls %>%  # MatchIt does not allow missing values
  select(c5r2mtsc_std, catholic, one_of(ecls_cov)) %>%
  na.omit()

mod_match <- matchit(catholic ~ race_white + w3income + p5hmage + p5numpla + w3mom
ed_hsb,
                     method = "nearest", data = ecls_nomiss)
```

We can get some information about how successful the matching was using `summary(mod_match)` and `plot(mod_match)` (try this yourself).

To create a dataframe containing only the matched observations, use the `match.data()` function:

```
dta_m <- match.data(mod_match)
dim(dta_m)
```

```
## [1] 2704    9
```

Note that the final dataset is smaller than the original: it contains 2704 observations, meaning that 1352 pairs of treated and control observations were matched. Also note that the final dataset contains a variable called `distance` , which is the propensity score.

4 Examining covariate balance in the matched sample

We'll do three things to assess covariate balance in the matched sample:

1. visual inspection
2. t-tests of difference-in-means
3. computation of the average absolute standardized difference ("standardized imbalance")

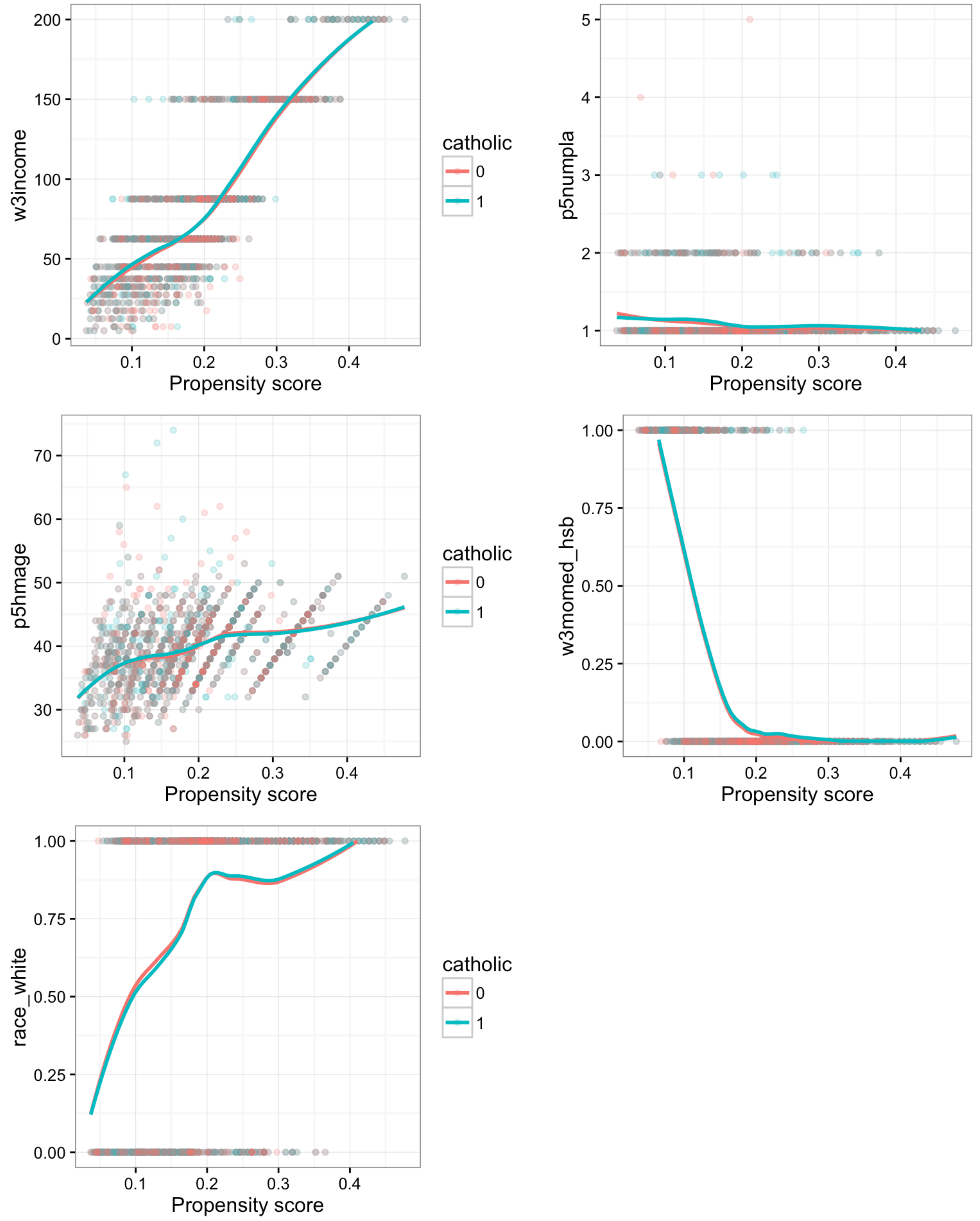
4.1 Visual inspection

It is useful to plot the mean of each covariate against the estimated propensity score, separately by treatment status. If matching is done well, the treatment and control groups will have (near) identical means of each covariate at each value of the propensity score.

Below is an example using the four covariates in our model. Here I use a loess smoother to estimate the mean of each covariate, by treatment status, at each value of the propensity score.

```r
fn_bal <- function(dta, variable) {
  dta$variable <- dta[, variable]
  if (variable == 'w3income') dta$variable <- dta$variable / 10^3
  dta$catholic <- as.factor(dta$catholic)
  support <- c(min(dta$variable), max(dta$variable))
  ggplot(dta, aes(x = distance, y = variable, color = catholic)) +
    geom_point(alpha = 0.2, size = 1.3) +
    geom_smooth(method = "loess", se = F) +
    xlab("Propensity score") +
    ylab(variable) +
    theme_bw() +
    ylim(support)
}

library(gridExtra)
grid.arrange(
   fn_bal(dta_m, "w3income"),
   fn_bal(dta_m, "p5numpla") + theme(legend.position = "none"),
   fn_bal(dta_m, "p5hmage"),
   fn_bal(dta_m, "w3momed_hsb") + theme(legend.position = "none"),
   fn_bal(dta_m, "race_white"),
   nrow = 3, widths = c(1, 0.8)
)
```



4.2 Difference-in-means

The means below indicate that we have attained a high degree of balance on the five covariates included in the model.

```
dta_m %>%
  group_by(catholic) %>%
  select(one_of(ecls_cov)) %>%
  summarise_all(funs(mean))
```

```
## Adding missing grouping variables: `catholic`
```

```
## # A tibble: 2 × 6
##   catholic race_white p5hmage w3income p5numpla w3momed_hsb
##      <int>      <dbl>   <dbl>    <dbl>    <dbl>       <dbl>
## 1        0  0.7470414 39.5503 81403.99 1.076183   0.2152367
## 2        1  0.7411243 39.5932 82568.94 1.091716   0.2233728
```

You can test this more formally using t-tests. Ideally, we should *not* be able to reject the null hypothesis of no mean difference for each covariate:

```
lapply(ecls_cov, function(v) {
    t.test(dta_m[, v] ~ dta_m$catholic)
})
```

```
## [[1]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = 0.35243, df = 2701.8, p-value = 0.7245
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02700440  0.03883872
## sample estimates:
## mean in group 0 mean in group 1
##       0.7470414       0.7411243
##
##
## [[2]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = -0.21331, df = 2702, p-value = 0.8311
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4372485  0.3514496
## sample estimates:
## mean in group 0 mean in group 1
##         39.5503         39.5932
##
##
## [[3]]
##
##  Welch Two Sample t-test
```

```
##
## data:  dta_m[, v] by dta_m$catholic
## t = -0.64787, df = 2701.9, p-value = 0.5171
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4690.731  2360.845
## sample estimates:
## mean in group 0 mean in group 1
##        81403.99        82568.94
##
##
## [[4]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = -1.339, df = 2699.5, p-value = 0.1807
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.038278301  0.007213213
## sample estimates:
## mean in group 0 mean in group 1
##        1.076183        1.091716
##
##
## [[5]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$catholic
## t = -0.51108, df = 2701.5, p-value = 0.6093
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.03935185  0.02307966
## sample estimates:
## mean in group 0 mean in group 1
##       0.2152367       0.2233728
```

4.3 Average absolute standardized difference

As a measure of the average imbalance, we can calculate the following:

$$\left|\frac{\bar{\beta}}{\sigma}\right| = \frac{1}{k} \sum_x \frac{|\beta_x|}{\sigma_x}$$

where β_x is the difference between the covariate means in the treated and control groups in the matched sample. An average absolute standardized difference that is close to 0 is preferable, since that indicates small differences between the control and treatment groups in the matched sample.

Try to implement a function that calculates the absolute standardized difference for any covariate in the matched sample. Then take the average for all the covariates.

5 Estimating treatment effects

Estimating the treatment effect is simple once we have a matched sample that we are happy with. We can use a t-test:

```
with(dta_m, t.test(c5r2mtsc_std ~ catholic))
```

```
## 
##  Welch Two Sample t-test
## 
## data:  c5r2mtsc_std by catholic
## t = 4.1788, df = 2670.1, p-value = 3.025e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.07834911 0.21688388
## sample estimates:
## mean in group 0 mean in group 1 
##       0.3572844       0.2096679
```

Or we can use OLS with or without covariates:

```
lm_treat1 <- lm(c5r2mtsc_std ~ catholic, data = dta_m)
summary(lm_treat1)
```

```
## 
## Call:
## lm(formula = c5r2mtsc_std ~ catholic, data = dta_m)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.5089 -0.5754  0.0431  0.6167  3.0764 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.35728    0.02498  14.304  < 2e-16 ***
## catholic    -0.14762    0.03533  -4.179 3.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9185 on 2702 degrees of freedom
## Multiple R-squared:  0.006421,   Adjusted R-squared:  0.006054 
## F-statistic: 17.46 on 1 and 2702 DF,  p-value: 3.024e-05
```

```
lm_treat2 <- lm(c5r2mtsc_std ~ catholic + race_white + p5hmage +
                  I(w3income / 10^3) + p5numpla + w3momed_hsb, data = dta_m)
summary(lm_treat2)
```

```
## 
## Call:
## lm(formula = c5r2mtsc_std ~ catholic + race_white + p5hmage + 
##     I(w3income/10^3) + p5numpla + w3momed_hsb, data = dta_m)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.6234 -0.5464  0.0336  0.6106  3.1875 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      -0.4973086  0.1490892  -3.336 0.000863 ***
## catholic         -0.1452595  0.0333418  -4.357 1.37e-05 ***
## race_white        0.2843161  0.0387899   7.330 3.03e-13 ***
## p5hmage           0.0151427  0.0032496   4.660 3.32e-06 ***
## I(w3income/10^3)  0.0030040  0.0003774   7.959 2.53e-15 ***
## p5numpla         -0.1106781  0.0555234  -1.993 0.046323 *  
## w3momed_hsb      -0.3815903  0.0415457  -9.185  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.8664 on 2697 degrees of freedom
## Multiple R-squared:  0.1175, Adjusted R-squared:  0.1155 
## F-statistic: 59.85 on 6 and 2697 DF,  p-value: < 2.2e-16
```

Interpret these estimates!

6 Exercise

In groups of 2 or 3, come up with a different model to estimate the propensity score. Evaluate whether it is better or worse than the model we have worked with so far by comparing the average absolute standardized difference (average imbalance) from the two models. Once you have settled on a model, report the following:

1. The average imbalance
2. The region of common support
3. Assessments of the covariance balance in your matched sample (visual inspections and t-tests)
4. The estimated treatment effect of going to catholic school (along with your interpretation of it)

If we have time, repeat the exercise using a different matching algorithm than nearest-neighbor.

The code used in this tutorial is available here (matching.R).