Regression analysis is one of the most requested machine learning methods in 2019. One group of regression analysis for measuring effects and to evaluate the statistical effect of covariates is Propensity Score Matching (PSM). This method is well suited to investigate if the covariates are changing the effects of the estimates in the regression model. It can, therefore, be used to design the regression model to be more accurate and efficient. In this article, you learn how to do Propensity Score Matching in R.

**Methodology**

The propensity score (PS) is a probability. It is a conditional probability of being exposed given a set of covariates, Pr(E+|covariates). We can calculate a PS for each subject in an observational study regardless of her actual exposure. Once we have a PS for each subject, we then return to the real world of exposed and unexposed. We can match exposed subjects with unexposed subjects with the same (or very similar) PS. Thus, the probability of being exposed is the same as the probability of being unexposed. The exposure is “random.”

Propensity score matching (PSM) arose as a way to achieve exchangeability between exposed and unexposed groups in observational studies without relying on traditional model building. Exchangeability is critical to our causal inference. In experimental studies (e.g. randomized control trials), the probability of being exposed is 0.5. Thus, the probability of being unexposed is also 0.5. The probability of being exposed or unexposed is the same. Therefore, the subject’s actual exposure status is random. This equal probability of exposure makes us feel more comfortable asserting that the exposed and unexposed groups are alike on all factors except their exposure. Therefore, we say that we have exchangeability between groups. One of the biggest challenges with observational studies is that the probability of being in the exposed or unexposed group is not random.

There are several occasions where an experimental study is not feasible or ethical. But we still would like the exchangeability of groups achieved by randomization. PSM helps us to mimic an experimental study using data from an observational study.

**Conducting PSM**

When conducting a PSM the following 5 steps are necessary:

**1. Decide on the set of covariates you want to include**

Decide on the set of covariates you want to include.  
This is the critical step to your PSA. We use these covariates to predict our probability of exposure. We want to include all predictors of the exposure and none of the effects of the exposure. We do not consider the outcome in deciding upon our covariates. We may include confounders and interaction variables. If we are in doubt of the covariate, we include it in our set of covariates (unless we think that it is an effect of the exposure).

**2. Use logistic regression to obtain a PS for each subject**

We use the covariates to predict the probability of being exposed (which is the PS). The more true covariates we use, the better our prediction of the probability of being exposed. We calculate a PS for all subjects, exposed and unexposed.

Using mathematical expressions:  
\( ln(PS/(1-PS))= β\_0+β\_1\*X\_1+…+β\_p\*X\_p \)  
\( PS= (exp(β\_0+β\_1\*X\_1+…+β\_p\*X\_p)) / \)  
\( (1+exp(β0 +β\_1\*X\_1 +…+β\_p\*X\_p)) \)

**3. Match exposed and unexposed subjects on the PS**

We want to match the exposed and unexposed subjects on their probability of being exposed (their PS). If we cannot find a suitable match, then that subject is discarded. Discarding a subject can introduce bias into our analysis.

Several methods for matching exist. Most common is the nearest neighbor within calipers. The nearest neighbor would be the unexposed subject that has a PS nearest to the PS for our exposed subject. We may not be able to find an exact match, so we say that we will accept a PS score within certain caliper bounds. We set an apriori value for the calipers. This value typically ranges from +/-0.01 to +/-0.05. Below 0.01, we can get a lot of variability within the estimate because we have difficulty finding matches and this leads us to discard those subjects (incomplete matching). If we go past 0.05, we may be less confident that our exposed and unexposed are truly exchangeable (inexact matching). Typically, 0.01 is chosen for a cutoff.

The ratio of exposed to unexposed subjects is variable. 1:1 matching may be done, but oftentimes matching with replacement is done instead to allow for better matches. Matching with replacement allows for the unexposed subject that has been matched with an exposed subject to be returned to the pool of unexposed subjects available for matching. There is a trade-off in bias and precision between matching with replacement and without (1:1). Matching with replacement allows for reduced bias because of better matching between subjects. Matching without replacement has better precision because more subjects are used.

**4. Check the balance of covariates in the exposed and unexposed groups after matching on PS**

The substantial overlap in covariates between the exposed and unexposed groups must exist for us to make causal inferences from our data. This is true in all models, but in PSA, it becomes visually very apparent. If there is no overlap in covariates (i.e. if we have no overlap of propensity scores), then all inferences would be made off-support of the data (and thus, conclusions would be model dependent).

We can use a couple of tools to assess our balance of covariates. First, we can create a histogram of the PS for exposed and unexposed groups. Second, we can assess the standardized difference. Third, we can assess the bias reduction.

\( Standardized difference=(100\*(mean(x exposed)- \)  
\( (mean(x unexposed)))/(sqrt((SD^2exposed+ $$SD^2unexposed)/2)) \)

More than 10% difference is considered bad. Our covariates are distributed too different between exposed and unexposed groups for us to feel comfortable assuming exchangeability between groups:  
\( Bias reduction= 1-(|standardized difference matched|/ \)  
\( |standardized difference unmatched|) \)

We would like to see a substantial reduction in bias from the unmatched to the matched analysis. What substantial means is up to you.

**5. Calculate the effect estimate and standard errors with this matched population**

An estimate of the average treatment effect of the treated:  
\( (ATT)=sum(Yexposed- Yunexposed) / no. of matched pairs \)

Standard errors may be calculated using bootstrap resampling methods. The resulting matched pairs can also be analyzed using standard statistical methods, e.g. Kaplan-Meier, Cox proportional hazards models. You can include PS in final analysis model as a continuous measure or create quartiles and stratify.

**Strengths and Limitations of PSM**

**Strengths**

It is possible to include interaction terms in calculating PSM. PSM uses one score instead of multiple covariates in estimating the effect. This allows an investigator to use dozens of covariates, which is not usually possible in traditional multivariable models because of limited degrees of freedom and zero count cells arising from stratifications of multiple covariates.  
PSM can be used for dichotomous and continuous variables (continuous variables has lots of ongoing research).  
Students included in this study may be a more representative sample of “real world” patients than a Randomised Controlled Trial (RCT) would provide. Since we don’t use any information on the outcome when calculating the PS, no analysis based on the PS will bias effect estimation. We avoid off-support inference. We rely less on p-values and other model-specific assumptions. We don’t need to know the causes of the outcome to create exchangeability.

**Limitations of PSM**

The most serious limitation is that PSM only controls for measured covariates. Group overlap must be substantial (to enable appropriate matching). Matching on observed covariates may open backdoor paths in unobserved covariates and exacerbate hidden bias. PSM works best in large samples to obtain a good balance of covariates. If we have missing data, we get a missing PS. Does not take into account clustering (problematic for neighborhood-level research).

**First analysis: Differences in means in R**

We start by loading the packages and [dataset](https://www.researchconnections.org/childcare/studies/4075?q=c5r2mtsc&type=Data+Sets) in R:

# Load packages in R

library(MatchIt)

library(dplyr)

library(ggplot2)

# Load dataset in R

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

First, it is time to look at the difference in means of the outcome variable:

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*

*1 0 9568 -0.03059583 0.01038536*

*2 1 1510 0.19386817 0.02235282*

Let us test if table above with difference in means is statistically significant at conventional levels of confidence:

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*

Now let us look at the difference in means in the 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 us calculate the mean for each covariate by the treatment status:

# Mean for each covariate by the treatment status

ecls\_cov %

group\_by(catholic) %>%

select(one\_of(ecls\_cov)) %>%

summarise\_all(funs(mean(., na.rm = T)))

*A tibble: 2 × 6*

*catholic race\_white p5hmage w3income p5numpla w3momed\_hsb*

*1 0 0.5561246 37.56097 54889.16 1.132669 0.4640918*

*2 1 0.7251656 39.57516 82074.30 1.092701 0.2272069*

We can also 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*

**Propensity Scores in R**

We estimate the propensity score by running a logit model (probit also works) where the outcome variable is a binary variable indicating treatment status. For the matching to give a causal estimate in the end, we need to include any covariate that is related to both the treatment assignment and potential outcomes. Let us choose just a few covariates below—they are unlikely to capture all covariates that should be included.

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 the student’s predicted probability of being Treated, given the estimates from the logit model. Below, we 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*

**Plotting Propensity Scores in R**

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

labs %

mutate(catholic = ifelse(catholic == 1, labs[1], labs[2])) %>%

ggplot(aes(x = pr\_score)) +

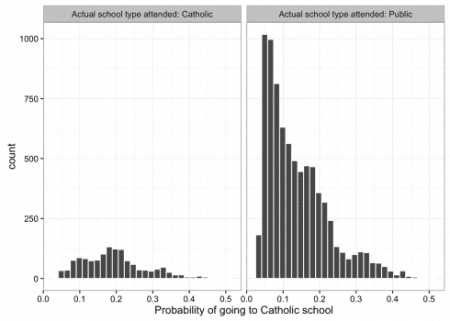
geom\_histogram(color = "white") +

facet\_wrap(~catholic) +

xlab("Probability of going to Catholic school") +

theme\_bw()

*`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.*

The above coding gives us the following histogram:  
[](https://i0.wp.com/datascienceplus.com/wp-content/uploads/2019/07/H1.png?ssl=1)

**Creating a matching algorithm**

A 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.  
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 % # 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 + w3momed\_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). To create a dataframe containing only the matched observations, we 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.

**Examining covariate balance in the matched sample**

We’ll do 2 things to assess covariate balance in the matched sample:

**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 we use a ‘loess smoother’ to estimate the mean of each covariate, by treatment status, at each value of the propensity score.

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"),

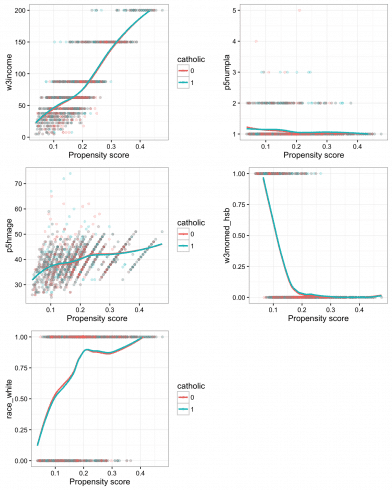
fn\_bal(dta\_m, "p5hmage"),

fn\_bal(dta\_m, "w3momed\_hsb") + theme(legend.position = "none"),

fn\_bal(dta\_m, "race\_white"),

nrow = 3, widths = c(1, 0.8)

)

The above coding gives us the following plot:  
[](https://i0.wp.com/datascienceplus.com/wp-content/uploads/2019/07/H2.png?ssl=1)

**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.

#Adding missing grouping variables: `catholic`

dta\_m %>%

group\_by(catholic) %>%

select(one\_of(ecls\_cov)) %>%

summarise\_all(funs(mean))

*A tibble: 2 × 6*

*catholic race\_white p5hmage w3income p5numpla w3momed\_hsb*

*1 0 0.7470414 39.5503 81403.99 1.076183 0.2152367*

*2 1 0.7411243 39.5932 82568.94 1.091716 0.2233728*

We 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*

**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:

**OLS 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*

**OLS with covariates**

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*