Propensity Score Analysis

Meghan Broadbent

An introduction to causal methods using propensity scores

University of Utah, 2017

1. Causality

When conducting a study, the ideal scenario would be to measure the effect of an exposure on an individual patient and compare the results to if that same patient had not received the exposure. However, this is unrealistic, as we cannot observe an outcome under different conditions at the same time for an individual. Instead, we turn to causal inference.

Causal inference is way to give mathematical conditions under which association implies causation. Because you cannot observe an event under different conditions (the counterfactual), causal inference is essentially a missing data problem.

The term *counterfactual* means “contrary-to-fact”. In causality, this implies we are talking about a world where we could observe an outcome had a patient received and not received an exposure. Counterfactuals no live in reality where we observe an outcome for a patient receiving or not receiving an exposure, instead counterfactuals live in a world of “what ifs”.

Let Y^a denote the counterfactual outcome of interest for exposure $A = a$, and let $Y \mid A = a$ be the outcome of interest observed in reality for that exposure. Hence, in observational data, we observe the outcome of interest Y under different exposure exposures, $Y \mid A = 1$ (where they got the exposure) and $Y \mid A = 0$ (where they did not get the exposure). In terms of causality, we instead wish we could observe $Y^{a=1}$ and $Y^{a=0}$, where both the entire population did and did not receive the exposure.

Association is defined by a different risk in two disjoint subsets of the population determined by the patients actual exposure. Whereas causation is defined by a different risk in the same subset of patients under two potential exposures.[1] We say that an exposure has an associative effect if $\mathbb{E}(Y \mid A = 1) \neq \mathbb{E}(Y \mid A = 0)$, and that an exposure has a causal effect if $\mathbb{E}(Y^{a=1}) \neq \mathbb{E}(Y^{a=0})$.

Thus, the goal of causal inference is to produce counterfactual populations, where we can observe the outcome under which the entire population received the exposure ($Y^{a=1}$) or did not receive the exposure ($Y^{a=0}$).

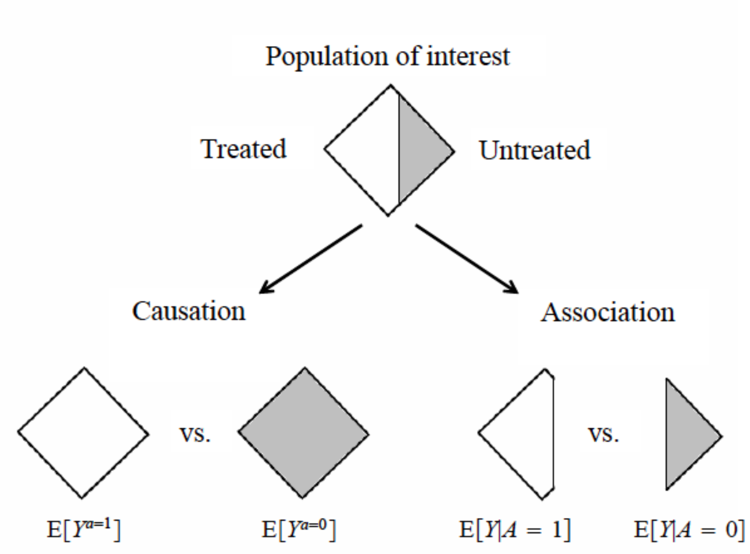


Figure 1: Causation versus Association. Source: Hernán et al. 2017. [2]

How we obtain these counterfactual populations differs depending on the methodology used, such as Propensity Score analyses, Instrumental Variable (IV) analyses, Marginal Structural Models (MSMs), Inverse Probability Weighting (IPW), Difference in Differences analyses (DID), and more. One of the more common approaches is use of the Propensity Score.

2. What is the Propensity Score?

In research, a randomized clinical control trial (RCT) is considered to be the gold standard in terms of estimating the effects of an exposure (or treatment/intervention) on the outcome of interest. The key here is randomization, where the random allocation of an exposure to the patient ensures that the exposure assignment is not confounded by an extraneous factor, such as a patient characteristic. This essentially means that a patient does or does not receive an exposure at random, and that this is not determined by something else. In an RCT, we can directly compare the effect of the exposure on the outcome of interest between the exposed and unexposed groups, because this randomization ensures the exposed and unexposed groups are *exchangeable*.[3]

However, in observational studies, exposure status is not randomly allocated by an investigator. Instead it occurs "naturally", often influenced by a patients own set of characteristics (e.g., BMI, education, gender, age, etc). As a result, baseline characteristics of exposed patients often differ systematically from those of unexposed patients.[3] These systematic differences in the baseline characteristics of the exposed and unexposed patients must be accounted for when estimating the effect of an exposure on the outcome of interest.

Propensity score methods can reduce bias and reduce the likelihood of confounding when estimating an exposure effect in non-randomized, observational data.[4] The propensity score (PS) itself is the predicted probability that a patient would receive (or not receive) the exposure given their baseline characteristics. Hence, it is a conditional probability of exposure assignment, and allows for observational data to be "conditionally randomized". In particular, the PS can be thought of as a balancing score, where the conditional probability of exposure balances the distribution of observed baseline characteristics/covariates between those who were and were not actually exposed in the observed data. This idea of "balance" follows directly from one of the three identifiability conditions that must be met in order to perform a causal analysis.

An important concept to keep in mind is the idea of being *parsimonious*. In a typical regression model with the outcome as the dependent outcome variable, you have to be parsimonious with the number of covariates you put into the model. Meaning, you have to limit how many variables you adjust for, otherwise you run the risk of overfitting. The rule of thumb in a regression model is that no more than '10 covariates' should be adjusted for. However, a propensity score model need not be parsimonious. In fact, you want to capture as much information as you possibly can within a propensity score, as it entails a great deal information into a single score; it reduces dimensionality by compressing a patients entire set of baseline characteristics into a single number – a probability score.

2.1. Identifiability Assumptions

In causal inference, we wish to treat observational data as a conditionally randomized experiment. There are three identifiability conditions to meet for this to be valid: Consistency, Positivity, and Conditional Exchangeability.

Consistency

The exposure is not assigned by the investigator, however it does correspond to a well defined intervention. Meaning, there is no/little risk of misclassification of the exposure and outcome in the study population.

Positivity

Also known as overlap or region of common support, positivity states that all conditional probabilities of exposure have a greater than zero chance of occurring/being observed. Meaning, if someone in the counterfactual population with a certain set of characteristics (e.g., overweight, well educated, diabetic) got the exposure, then there is also someone in the counterfactual world who didn't get the exposure and had the same set of characteristics.

Conditional Exchangeability

What we wish to achieve is $Y^a \amalg A$, meaning the conditional risks of the outcome are equal and independent of the exposure "assignment". It should not matter which group got the exposure and which did not, the same effects should be observed regardless. Hence, $\mathbb{P}(Y = 1 \mid A = 1) = \mathbb{P}(Y = 1 \mid A = 0) = \mathbb{P}(Y^a = 1)$. An assumption for this criteria is no unmeasured confounding.

3. Propensity Score Methods

Let $A = 1$ if a patient received the exposure, and $A = 0$ if a patient did not. To obtain a predicted probability of whether or not a patient were to receive the exposure based on their characteristics, we use a logistic regression with A as the outcome variable,

$$\text{logit}(\mathbb{P}(A = 1 \mid \boldsymbol{x}_i)) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_i + \epsilon_{ik}$$

where $\boldsymbol{x}_i$ denotes a vector of covariates (baseline characteristics).

Suppose for example you were conducting a study of whether or not exposure to tobacco resulted in a higher risk of cardiovascular disease, and that the covariates measured included sex, age, exercise level, dietary habits, and family history of cardiovascular disease. Then the propensity score model would be the following,

$$\text{logit}(\mathbb{P}(Smoke = 1 \mid \boldsymbol{x}_i)) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1 sex_{i1} + \beta_2 age_{i2} + \beta_3 exercise_{i3} + \beta_4 diet_{i4} + \beta 5 history_{i5} + \epsilon_i$$

This will calculate a propensity for exposure (predicted probability of having been exposed to the exposure/intervention) for each patient in the observed population. Once this propensity score (p) is obtained, we can assign p to those who actually got treatment, and $1 - p$ to those who did not. We could then use the propensity score to match patients who were exposed/unexposed, to stratify or reweight the study population, to use as an adjustment variable, and more. The purpose of this exercise is to focus on the 4 main propensity score methods: Matching, Stratification, IPW, and Covariate Adjustment.

3.1. Matching

Propensity score matching consists of calculating a predicted probability of exposure for those who were exposed and those who were not, whereby patients from each group are then matched based on similar or nearly identical propensity scores. The counterfactual population is effectively created by matching an exposed patient with a certain set of characteristics to an unexposed patient (or patients) with a very similar set of characteristics. Because the propensity score is entirely determined by a patients set of observed/measured covariates (e.g., age, gender, clinical presentation, clinical history, geographic region, etc), it can be assumed that the propensity score is a proxy for determining patients who are very similar to one another in respect to their characteristics; thus a comparison can be made between them as if they were the same individual both exposed and unexposed in the counterfactual population (assuming there is no unmeasured confounding).

Because patients are matched together based on their propensity scores, it is crucial to have a good region of common support (i.e., overlap or positivity), as matching typically reduces the sample size of the counterfactual population in the analysis (and is thus a limitation).

3.1.1. One-to-One or One-to-Many?

One-to-One (1:1) matching is as it sounds, matching one patient who got the exposure to one patient who did not get the exposure based on the propensity score. One-to-Many (1:M) however is matching one patient who got the exposure to 2 or more patients who did not get the exposure (all with similar or near identical propensity scores). Although 1:M would retain a larger sample size in terms of matching, 1:1 is more commonly used in practice.

3.1.2. Caliper distance or not?

A caliper distance in terms of matching designates an upper limit on how different two propensity scores can be in order to be considered for a match. The caliper is calculated as $c = d * sd(logit(p))$, where d corresponds to the caliper distance used. For instance, a caliper distance of 0.2 would indicate that the distance between two propensity scores (to be considered for a match) can be no more than 0.2 times the standard deviation of the logit of the propensity score. A caliper of 0.2 is most commonly used, however if you wish to be more strict with the matching criteria a caliper distance could be set at a smaller value (or at a higher value if you wish for your matching criteria to be defined more loosely). Note: the terms “caliper” and “caliper distance” often refer to the same thing.

3.2. Stratification

The stratification of the propensity score groups participants into strata, based on their propensity score.[4] The choice of strata can influence the bias and variance of the estimate. The wider the range is for each strata, the lower the variance and higher the chance of bias – and vice versa. Typically the propensity score is grouped into 5 strata, however 10 deciles are also commonly used.

A variable is created to capture the information of which strata a patient's propensity score falls into, where this strata variable is then adjusted for in the outcome regression (with one of the strata held as reference). The estimate of effect is then calculated for each strata of the propensity score as well as the exposure of interest, however the strata effects can be pooled across all strata for an overall estimate.

3.3. Covariate Adjustment

Once the propensity score is obtained, it can be used as a continuous adjustment variable in your regression model. Because the propensity score it not parsimonious, you can reduce dimensionality by compressing a patients entire set of baseline characteristics this single score. Rather than performing a multivariate regression model where the exposure and set of covariates are regressed onto the outcome of interest, you can instead regress the exposure and propensity score onto the outcome of interest (in the original population – not the matched population), where the propensity score takes place of the covariates you would typically adjust for. Hence it allows you to estimate the exposure effect while adjusting for a patients probability of being exposed, while simultaneously capturing a patients covariate information and thus reduces confounding.

3.4. IPW

Inverse probability weighting (IPW) uses the propensity score to reweight the study population (into a counterfactual population) in order to "unconfound" the exposure and outcome of interest. Each patient is weighted according to the inverse of the predicted probability of exposure status (whether that status is exposed or unexposed).

Weighting options differ between unstabilized and stabilized, whereas additional methods such as truncation can be used to modify the inverse probability weights.

3.4.1. Unstabilized Weighting

In the case of unstabilized weighting, a direct inverse of exposure is calculated as $1/f(A \mid L)$, where A is the exposure and L is the set of adjustment variables (confounders, covariates, etc). Hence, the inverse of the predicted probability of being exposed (IPTW) is $1/p$, where p is the propensity score. The inverse probability of being unexposed is instead $1/1-p$. For example, if a patient's propensity for being exposed is 0.37 and they had the exposure in the observed data, then their weight in the counterfactual study $1/0.37 = 2.7$. Hence, the patient would represent themselves 2.7 in the counterfactual study population. If that same patient instead had not had the exposure in the observed data, their weight in the counterfactual population would instead be 1/(1-0.37) = 1.6.

Unstabilized weighting can lead to extreme weighting. If large/extreme weights do occur, it may warrant the use of stabilized weighting or truncation (or both).

3.4.2. Stabilized Weighting

In stabilized weighting (sw), the predicted probability of exposure (or lack-thereof) is instead considered out of the entire probability of being exposed. Hence, $\text{sw} = f(A)/f(A \mid L)$. For example, suppose that the overall probability of having the exposure is 46%. The patient with a propensity score of 0.37 (who also had the exposure) would then have a stabilized weight of $0.46/0.37 = 1.2$, and would represent 1.2 of themselves in the counterfactual population.

This weighting method reduces Type I error by preserving the original sample size in the counterfactual weighted population, as it keeps a patient's respective weight closer to 1 (reduces extreme weights).

3.4.3. Truncation

If there are extreme weights within the study population, it may skew the counterfactual population and not be as representative as we'd like. For instance, if an exposed patient's predicted probability of exposure is 0.0005, then their weight is $1/0.0005 = 2{,}000$. Meaning, they would be represented 2,000 times in the counterfactual population. To avoid this, truncation can be used at the 1st-percentile and 99th-percentile (or 5th and 95th) to put a bound on how extreme the weights are allowed to be. There is a bias/variance trade-off with this approach, as truncation reduces variability but potentially increases bias.

4. Are your assumptions valid?

When performing propensity score methods (or causal inference overall), we must check to ensure our study population and methodology have met the identifiability assumptions needed to proceed with a causal analysis.

Consistency

We want for the exposure to correspond to a well defined intervention. Meaning, there is no/little risk of misclassification of the exposure and outcome in the study population. This, and the assumption of no unmeasured confounding, begin with the study design itself. Propensity score methods work best when you have measured an ample amount of covariates in the study population. Studies utilizing these methods on observational data can have upwards of 50-100+ variables measured to not only better identify potential confounding, but to also capture as much information about a patient as possible into their propensity for exposure.

In reality, it is not possible to truly quantify whether you have met the assumption of consistency (or no unmeasured confounding for that matter), however the more data you collect the more likely you are to identify well defined assignments of who was exposed and who had the outcome, and those who did not.

Positivity

The propensity score itself should exhibit positivity, meaning that the propensity scores for those exposed and those unexposed have sufficient overlap (also known as the region of common support). One of the first steps in a propensity score analysis is to check this assumption using either a density plot or histogram (or similar). If this assumption is violated, the use of interaction terms and transformation of variables (e.g., age^2 or age*risk of flu) could be useful in achieving a good region of common support. If problems still persist with this assumption, there may be unmeasured confounding warranting further data collection. Note: the distribution of the propensity scores may be different between the exposure groups, however we are looking for the region of overlap - not necessarily similar distributions.

Conditional Exchangeability

In a randomized clinical control trial, randomization of the exposure ensures conditional exchangeability. However in observational studies, causal methods such as the propensity score are needed to meet this assumption. Typically you want the covariates/characteristics of your study population to be similar for those exposed and unexposed, so that comparisons can be made.

To check this assumption, calculate or plot the standardized mean difference (SMD) of the covariates. A general rule of thumb is that you want the SMD to be under a 0.1 difference (10% difference) for each of the covariates being adjusted for in the propensity score model. The SMD itself is calculated as the absolute difference in the means of a covariate between the exposed and unexposed.

5. Step-By-Step

There is a typical “work flow” to follow when using propensity scores in causal inference, outlined below.

1 – Define your causal question.

- What is it that you are trying to answer using causal methods?

2 – Do a literature review, talk with subject matter experts (if possible), and construct a causal diagram.

- You want to understand as much as you can about the relationship not only between your exposure and outcome of interest, but also with outside factors that could influence them and be influenced by them.
- Draw a causal diagram (DAG) to better identify confounding/backdoor-paths that were not obvious based on your understanding of the data. Be able to justify the variables you choose to adjust for.

3 – Create a proxy data set with no outcome.

- Create an ID variable, save your outcome and ID variable to a new dataset, and create a new dataset with the ID variable included but the outcome dropped. Use this full dataset with no outcome variable to do your analysis.
- This is a preventative measure to protect against subconscious “phishing” or “data grunging”.

4 – Identify missing data.

- If you do have missing data, *why*? Is it missing at random? Or is it missing for a specific reason that you should know about?
- Make an informed decision on whether to do a complete case analysis (omission of missing data entirely, usually not an ideal choice), or whether to impute (mean/median/multiple imputation).

5 – Create your propensity score.

- Use a logistic regression model with the exposure variable as your outcome and covariates as your independent variables.
- Play around with the model, use interaction terms and transformations of variables if it means you get a better model. This is one of the only times in which you can get away with “tweaking your model” for a better outcome, as your goal is to produce a propensity model such that your population of exposed and unexposed are exchangeable (i.e., balanced).

6 – Assess positivity (i.e., overlap or region of common support).

- The overlap of the propensity scores for the exposed and unexposed should be similar, as you need to be able to compare exposed patients with a certain propensity score to unexposed patients with a similar propensity score.
- If you haven't achieved a good region of common support, go back to step 5 and tweak your model until you've achieved sufficient overlap. Once you have, proceed to step 7. Note: if you still can't achieve positivity, there may be unmeasured confounding warranting further data collection.

If you choose to use SW/IPW, you should also check the weights for any extremes (warranting truncation), however positivity is typically assess from the propensity score itself (in addition to the assessment of the distribution of weights).

7 – Choose a causal method, merge your outcome, and run the appropriate analysis.

- Merge your outcome back into the data with your propensity scores by the ID variable you created.
- For propensity score matching, run a matched regression. For IPW/SW, run a weighted regression. For covariate adjustment/stratification, run a regression adjusted for the continuous propensity score or the stratified groups.

Unless you are conducting an analysis where the principle investigators have already determined which method to use, it is best to play around with 4-5 different methods to see which produces the best balance (e.g., run propensity score matching and IPW/SW and stratification and adjustment, etc).

8 – Assess conditional exchangeability for the covariates you have measured (i.e., balance)

- For a propensity score matching analysis, check the balance of the covariates in your matched propensity score model. A good way to both quantify and visualize balance is to calculate/plot a standardized mean difference (SMD) for each of the covariates. A rule of thumb is that you want the SMD for each covariate to be under 0.10 (less than a 10% difference). If you haven't achieved good balance, go back to step 5 and tweak the propensity score until you achieve balance.
- For IPW/SW, you will want to check balance (e.g., SMD) as well as the distribution of weights for both the exposed and unexposed groups (plot the weights using a histogram for the exposed and unexposed). For SW, your weights should be close to 1 for both groups, and should have a similar range for IPW. Use truncation if you have extreme weights for either IPW/SW. If you haven't achieved good balance and/or you don't have sufficient weights, go back to step 5 and tweak the propensity score and then reweight your population until this is met.

If your model has good balance/weights, or if you choose to use covariate adjustment/stratification, then proceed to step 9. If you cannot achieve sufficient balance (or weighting), there may be unmeasured confounding warranting further data collection.

Note: another way to assess the balance of your measured covariates is to plot histograms/density plots of the covariates stratified by the exposure of interest. *It is important to remember that the assumption of conditional exchangeability follows the assumption of no unmeasured confounding, however we can only assess the balance of the covariates we measured.*

9 – **Interpret your findings.**

- Based on the causal methods performed, choose which model you think is best for estimating the treatment effect in your study population and explain why.
- Interpret the estimates of effect based on the model you've chosen.

6. National Health Epidemiologic Follow-Up Study (NHEFS)

We will walk through an example for implementing a propensity score model in a variety of methods using the NHEFS as example data. The NHEFS study was a follow up to NHANES, designed to investigate relationships between clinical, nutritional, and behavioral factors from the NHANES study and subsequent morbidity, mortality, hospital utilization, risk factors, and more.[5]

The causal question of interest is "Does quitting smoking cause you to gain weight?", where we will analyze 1,566 tobacco users aged 25-74 with a baseline visit in 1971 and a follow up visit in 1982. The objective is to assess smoking cessation and whether it has an effect on weight gain (or loss). Individuals are classified as exposed if they reported having quit smoking before the follow-up visit, and as unexposed otherwise. Each participants weight change was measured (in kg) as the body weight at the follow-up visit minus the body weight at the baseline visit.

For this analysis the statistical programming language R (version 3.3.3) will be used.

6.1. Introduction to R

To download the latest version of R for windows please visit https://cran.cnr.berkeley.edu/bin/windows/base/, or visit https://cran.cnr.berkeley.edu/bin/macosx/ for a mac. Once R is downloaded, you can begin using it via the terminal or command line. However, for a more intuitive way of working with data using R, you'll want to use an integrated development environment (IDE) such as RStudio. To download RStudio, visit https://www.rstudio.com/products/rstudio/download/#download and select either the Windows Vista/7/8/10 installer (for windows) or the Mac OS X 10.6+ installer (for macs).

For each of these, follow the set up instructions and open RStudio once you have R installed, as you cannot use RStudio unless you have R downloaded.

6.1.1. Packages needed

The following packages will be useful for an exploratory propensity score analysis: gdata, tableone, Matching, MatchIt, dplyr, ipw, ggplot2, survey, and reshape2.

6.2. Analysis

We begin by installing the packages needed and loading the package libraries, defining our working path on the computer, and reading in our dataset.

```r
install.packages("gdata"); library(gdata) ### to read in XLS format data
install.packages("tableone"); library(tableone) ### creates SMDs of covariates (creates a "Table 1")
install.packages("Matching"); library(Matching) ### package for matching propensity scores
install.packages("MatchIt"); library(MatchIt) ### another package for matching (to compare)
install.packages("dplyr"); library(dplyr) ### package for manipulating dataframes (datasets)
install.packages("ipw"); library(ipw) ### to perform IPW
install.packages("ggplot2"); library(ggplot2) ### for plotting
install.packages("survey"); library(survey) ### package for survey design data, but is needed to regress IPW
install.packages("reshape2"); library(reshape2) ### for reshaping data

### set working directory to where the data is located
setwd("/Users/meg/Dropbox/School/PHS7030_Causal/Session2_PropensityScoresIPW/Hw2_Propensities")

### read the xls dataset into an R dataframe (dataset), we denote the data as 'df'
### sheet=1 is needed to tell R which XLS sheet the data is on
### header = T means it's TRUE that there is a header with column names
df = read.xls("NHEFS.xls", sheet=1, header=T)
```

We can check the names of the variables (columns) we have in our data to build a DAG.

```r
### check the column names
colnames(df)
```

```
'seqn' 'qsmk' 'death' 'yrdth' 'modth' 'dadth' 'sbp' 'dbp' 'sex' 'age' 'race' 'income' 'marital' 'school'
'education' 'ht' 'wt71' 'wt82' 'wt82_71' 'birthplace' 'smokeintensity' 'smkintensity82_71' 'smokeyrs'
'asthma' 'bronch' 'tb' 'hf' 'hbp' 'pepticulcer' 'colitis' 'hepatitis' 'chroniccough' 'hayfever'
'diabetes' 'polio' 'tumor' 'nervousbreak' 'alcoholpy' 'alcoholfreq' 'alcoholtype' 'alcoholhowmuch'
'pica' 'headache' 'otherpain' 'weakheart' 'allergies' 'nerves' 'lackpep' 'hbpmed' 'boweltrouble'
'wtloss' 'infection' 'active' 'exercise' 'birthcontrol' 'pregnancies' 'cholesterol' 'hightax82'
'price71' 'price82' 'tax71' 'tax82' 'price71_82' 'tax71_82'
```

For the purpose of this example and for the sake of space, a smaller subset of variables were used (as shown by the DAG), however nearly every one of these variables could be included – as long as it was measured pre-1982. Some of the covariates in the dataset were measured during the follow up time period in 1982. See the codebook for this data for more details.

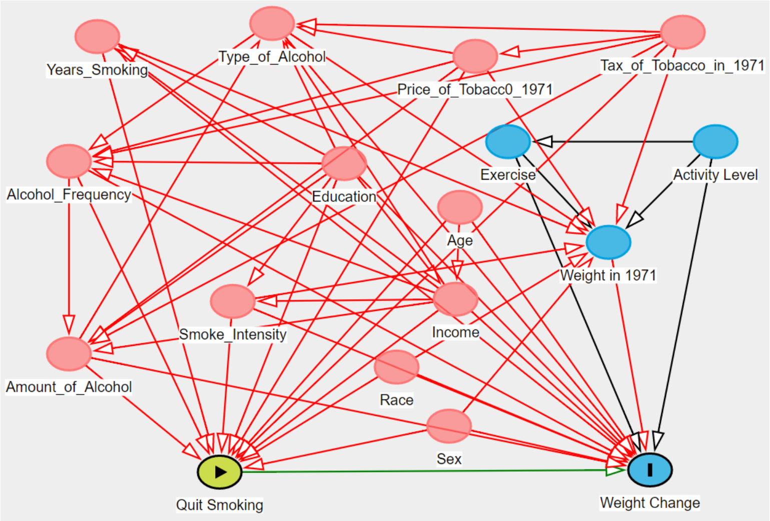


Figure 2: Causal Diagram of the NHEFS Study on Smoking Cessation and Weight Change

The next step would be to create an ID variable, inspect the data, and create vectors containing the variables we want to keep or include. In this analysis, we will reference the entire set of variables we will work with (such as the outcome variable and covariates), as well as the set of adjustment variables (not including the ID nor outcome).

```
### create an ID variable
df$id = seq.int(nrow(df))

### inspect the data, output omitted in example
head(df1)

### create a vector of these column variables names we wish to include in our model
include_covar = c("active", "age", "alcoholfreq", "alcoholhowmuch", "alcoholtype",
                  "education", "exercise", "income", "race", "sex", "smokeintensity",
                  "id", "smokeyrs", "wt71", "tax71", "price71", "qsmk")

### create vector of variables not including outcome or exposure or id
### this for creating a "Table 1"
covars_only = c("active", "age", "alcoholfreq", "alcoholhowmuch", "alcoholtype",
                  "education", "exercise", "income", "race", "sex", "smokeintensity",
                  "smokeyrs", "wt71", "tax71", "price71")
```

We now format the data as needed, such as subsetting to the variables we want, and handling missing data with median imputation. Other methods of imputation, such multiple imputation, should be explored.

```r
### accessing a dataframe follows the format: [row , column], set new data into a dataframe called df_includeCovar and do similar for the outcome and ID
df_includeCovar = df[ , (names(df) %in% include_covar)]
df_outcomeOnly = df[ , (names(df) %in% c("wt82_71", "id"))]

### fix up missing values for income with the rounded median of those groups where income is NA and sex is male/female and education is college or higher (or lower)
### to conditionally assign something, we use the format: df$column[condition] = assignment

### missing income for males with high education
df_includeCovar$income[(is.na(df_includeCovar$income) & df_includeCovar$sex==0 &
    df_includeCovar$education>3)] = round(median(df_includeCovar$income[(df_includeCovar$sex==0 &
    df_includeCovar$education>3)], na.rm=T))
### missing income for males with low education
df_includeCovar$income[(is.na(df_includeCovar$income) & df_includeCovar$sex==0 &
    df_includeCovar$education<=3)] = round(median(df_includeCovar$income[(df_includeCovar$sex==0 &
    df_includeCovar$education<=3)], na.rm=T))
### missing income for females wiht high education
df_includeCovar$income[(is.na(df_includeCovar$income) & df_includeCovar$sex==1 &
    df_includeCovar$education>3)] = round(median(df_includeCovar$income[(df_includeCovar$sex==1 &
    df_includeCovar$education>3)], na.rm=T))
### missing income for females with low education
df_includeCovar$income[(is.na(df_includeCovar$income) & df_includeCovar$sex==1 &
    df_includeCovar$education<=3)] = round(median(df_includeCovar$income[(df_includeCovar$sex==1 &
    df_includeCovar$education<=3)], na.rm=T))

### fix up missing values for alcoholhowmuch when they dont ever drink
### if you dont ever drink, then how much should be 0, however in this case 4 means 'none'
df_includeCovar$alcoholhowmuch[(is.na(df_includeCovar$alcoholhowmuch) & df_includeCovar$alcoholpy==0)] = 4

### fix up missing values for alcoholhowmuch when they do drink (males)
df_includeCovar$alcoholhowmuch[(is.na(df_includeCovar$alcoholhowmuch) & df_includeCovar$sex==0)] = round(median(df_includeCovar$alcoholhowmuch[df_includeCovar$sex==0], na.rm=T))
### females
df_includeCovar$alcoholhowmuch[(is.na(df_includeCovar$alcoholhowmuch) & df_includeCovar$sex==1)] = round(median(df_includeCovar$alcoholhowmuch[df_includeCovar$sex==1], na.rm=T))

### fix up missing values for tobacco prices and taxes on tobacco products in 1971
df_includeCovar$price71[is.na(df_includeCovar$price71)] = round(median(df_includeCovar$price71, na.rm=T))
df_includeCovar$tax71[is.na(df_includeCovar$tax71)] = round(median(df_includeCovar$tax71, na.rm=T))
```

We then check the standardized mean differences of our raw imputed data. For reference, `qsmk` = 1 means a patient quit smoking, and 0 meaning the patient continued to smoke at follow-up. Quite a few variables have a difference greater than 10%, suggesting the observed study population is not exchangeable.

```r
### create table one and print the table
unmatchedTableOne = CreateTableOne(vars=covars_only, strata="qsmk", data=df_includeCovar, test=F)
print(unmatchedTableOne, smd = TRUE)
```

```
                            Stratified by qsmk
                             0             1             SMD   
  n                           1201           428               
  active (mean (sd))          0.64 (0.65)   0.69 (0.66)   0.072
  age (mean (sd))            42.92 (11.89) 46.70 (12.52)  0.309
  alcoholfreq (mean (sd))     1.91 (1.30)   1.95 (1.32)   0.032
  alcoholhowmuch (mean (sd))  3.09 (2.75)   2.91 (2.21)   0.073
  alcoholtype (mean (sd))     2.46 (1.21)   2.52 (1.20)   0.054
  education (mean (sd))       2.69 (1.15)   2.75 (1.28)   0.049
  exercise (mean (sd))        1.18 (0.75)   1.25 (0.72)   0.101
  income (mean (sd))         17.92 (2.67)  18.09 (2.45)   0.066
  race (mean (sd))            0.15 (0.35)   0.09 (0.28)   0.182
  sex (mean (sd))             0.53 (0.50)   0.45 (0.50)   0.172
  smokeintensity (mean (sd)) 21.18 (11.58) 18.79 (12.26)  0.200
  smokeyrs (mean (sd))       24.25 (11.83) 26.61 (13.03)  0.189
  wt71 (mean (sd))           70.49 (15.57) 72.63 (16.08)  0.135
  tax71 (mean (sd))           1.05 (0.21)   1.06 (0.21)   0.005
  price71 (mean (sd))         2.13 (0.22)   2.13 (0.23)   0.021
```

Calculating the Propensity Score

With the data prepared and formatted as needed, we compute the propensity score for the predicted probability of quitting smoking (exposure) as a function of the covariates we've chosen using logistic regression.

Remember, if $A = 1$ for a patient receiving an exposure, and $A = 0$ for no exposure, then to obtain a predicted probability of whether or not a patient receives exposure based on their characteristics we use a logistic regression with exposure A as the outcome variable, rather than our outcome of interest,

$$\text{logit}(\mathbb{P}(A = 1 \mid \boldsymbol{x}_i)) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \epsilon_{ik}$$

where $\boldsymbol{x}_i$ denotes a vector of covariates (baseline characteristics).

```r
### logistic regression of treatment (quitting smoking) as a function of age, alcohol, etc
manualPS = glm(qsmk ~ active + age + alcoholfreq + alcoholhowmuch +
                      alcoholtype + wt71 + education + exercise + income +
                      race + sex + smokeintensity + smokeyrs + tax71 + price71,
               family = binomial(link="logit"),
               data = df_includeCovar)

### get summary of regression
summary(manualPS)
```

```
Call:
glm(formula = qsmk ~ active + age + alcoholfreq + alcoholhowmuch +
    alcoholtype + wt71 + education + exercise + income + race +
    sex + smokeintensity + smokeyrs + tax71 + price71, family = binomial(link = "logit"),
    data = df_includeCovar)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3276  -0.8195  -0.6532   1.1309   2.3610

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.274508   1.222221  -1.861 0.062749 .
active          0.066310   0.092933   0.714 0.475523
age             0.049137   0.009686   5.073 3.91e-07 ***
alcoholfreq     0.050810   0.057525   0.883 0.377094
alcoholhowmuch -0.015391   0.025944  -0.593 0.553028
alcoholtype    -0.013660   0.063787  -0.214 0.830425
wt71            0.006146   0.004168   1.474 0.140353
education       0.070710   0.056861   1.244 0.213666
exercise        0.163862   0.085950   1.906 0.056588 .
income          0.036033   0.026644   1.352 0.176252
race           -0.736842   0.205078  -3.593 0.000327 ***
sex            -0.551271   0.148052  -3.723 0.000196 ***
smokeintensity -0.025012   0.005487  -4.558 5.16e-06 ***
smokeyrs       -0.027918   0.009713  -2.874 0.004049 **
tax71           0.629652   0.923609   0.682 0.495410
price71        -0.751382   0.868139  -0.866 0.386759
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1876.3  on 1628  degrees of freedom
Residual deviance: 1780.0  on 1613  degrees of freedom
AIC: 1812

Number of Fisher Scoring iterations: 4
```

With the the propensity score (p) created from the logistic regression, we create a field in the dataset for p and $1 - p$, and then assign the propensity for quitting smoking (p) for those who actually quit, and assign $1 - p$ for those who did not quit smoking by the follow up time. This new column with the assigned propensities is what we'll use to match or reweight the observed population, and to assess our assumption of positivity.

```r
#setup labels before hand for our plots
labelsCS = paste("Quite Smoking:", c("Yes", "No"))

#pull out the propensity score for the matched data
ps_pred = data.frame(ps_score = predict(manualPS, type="response"),
                     qsmk = manualPS$model$qsmk)

#create a summary of the propensity score (to plot in the corner of histogram)
#do this for those who quit smoking (summary: min max quartiles)
summary_ps_pred = summary(ps_pred$ps_score[ps_pred$qsmk==1])

#paste the names in teh summary and format the results to 2 digits
#collapse the information by the escape character \n for new line
format_summary_ps_pred = paste(names(summary_ps_pred),
                               format(summary_ps_pred, digits=2),
                               collapse="\n ")

#set the annotated information into a dataframe, and attach the correct label
#x and y are the plotting margins, we'll specify them again later
annotations_ps_pred = data.frame(x=0.6, y=75, lab = format_summary_ps_pred,
                                 qsmk = factor(labelsCS[1], levels=c(labelsCS[1], labelsCS[2])))

#check that these look right
format_summary_ps_pred; annotations_ps_pred

#create a summary of those who didn't quit smoking (summary: min max quartiles)
summary_ps_notpred = summary(ps_pred$ps_score[ps_pred$qsmk==0])

#pasting the information from the summary, with digits set at 2
format_summary_ps_notpred = paste(names(summary_ps_notpred),
                          format(summary_ps_notpred, digits=2),
                          collapse="\n ")

#set the annotated information into a dataframe
annotations_ps_notpred = data.frame(x=0.6, y=75, lab = format_summary_ps_notpred,
                        qsmk = factor(labelsCS[2], levels=c(labelsCS[1], labelsCS[2])))

#check that these look right
format_summary_ps_notpred; annotations_ps_notpred

#combine the annotated information into a single data frame, we'll use this to plot
ps_pred_total = rbind(annotations_ps_pred, annotations_ps_notpred)

#check to make sure this is correct
ps_pred_total

#plot from the dataset itself using the pipe function %>%
#we mutate the data so that people with qsmk==1 get a label of quit smoking, else no quit smk
#plot the propensity score as a histogram
#use facet_wrap to plot separate the histograms for smoking status
#use geom_text to add the summary information about the PS to the corner of the plot
ps_pred %>%
  mutate(qsmk = ifelse(qsmk==1, labelsCS[1], labelsCS[2])) %>%
  ggplot(aes(x = ps_score)) +
  geom_histogram(color = "white") +
  facet_wrap(~qsmk, nrow=2, ncol=1) +
  xlab("Probability of Quitting Smoking") +
  ylab("Count") +
  ggtitle("Region of Common Support: Propensity Score") +
  geom_text(data = ps_pred_total, aes(x=0.6, y=75, label=lab, group=NULL)) +
  theme_bw() +
  theme(text = element_text(size=15))
```

Overall, the region of common support (balance) looks good (Fig. 3). There is perhaps a small number of patients with a propensity score that don't have an overlapping score in the contrasting group, however this is okay and manageable. There will never be a perfect overlap in the range of the propensity scores.

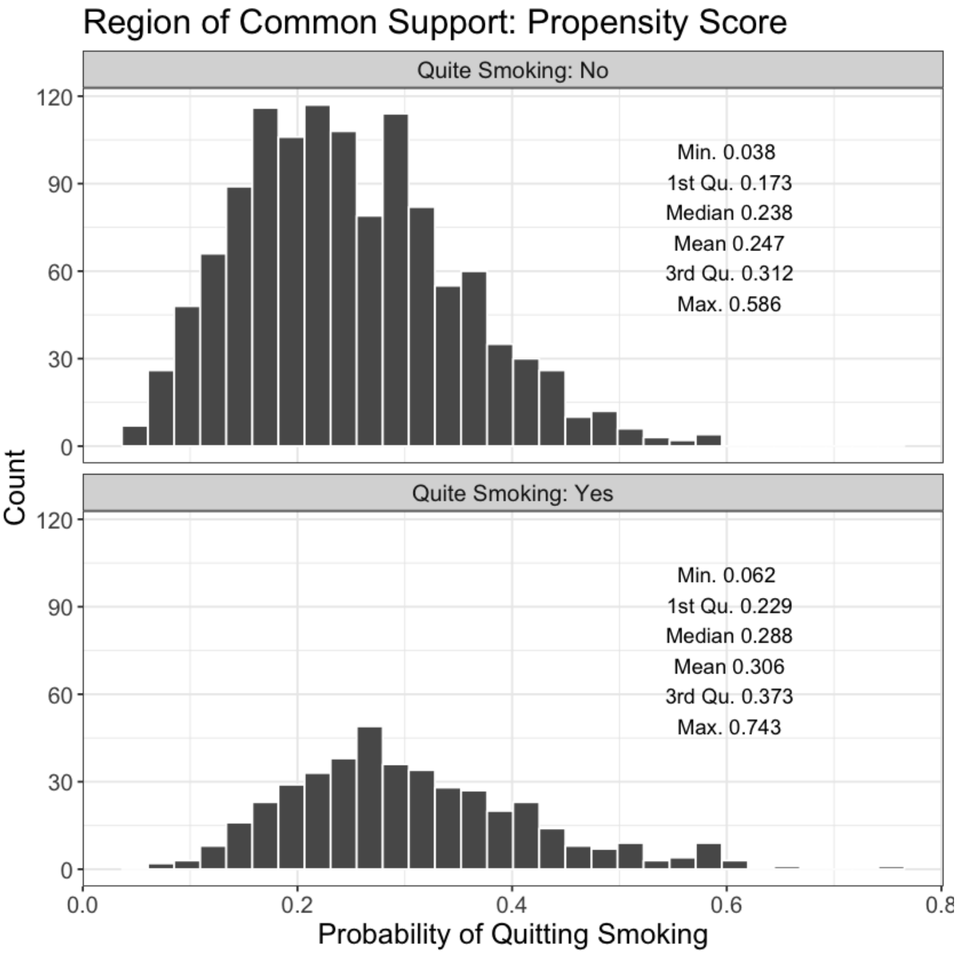


Figure 3: Distribution of the propensity score by exposure group

At this point we use the propensity score in a variety of methods.

Matching manually created propensity scores

After we've created the assigned propensity scores (p for those actually exposed and $1 - p$ for those actually unexposed), we'll use the Matching package to perform 1:1 matching on those who did and did not quit smoking, both with and without a caliper of 0.2.

One-to-One Matching, No Caliper

Using the Matching package, we match the patients based on their propensity scores. We then assess the SMD's of the covariates in this new 1:1 matched population.

```r
### Matching Package - without caliper distance
ps_MatchPkg = Match(Tr = (df_includeCovar$qsmk == 1), ### treatment
                    X = log(df_includeCovar$ps_qsmk/df_includeCovar$ps_notqsmk), ### log(ps / 1-ps)
                    M = 1, ### 1:1 matching
                    caliper = NULL, ### no caliper distance set
                    replace = F, ### no replacement
                    ties = T, ### true that PS's can be tied for treated/untreated
                    version = "fast")

### pull out the matched data by their index as treated and control (given from Match package) by rows
ps_MatchIndex = df_includeCovar[unlist(ps_MatchPkg[c("index.treated","index.control")]) , ]

### CreateTableOne
ps_MatchPkg_T1 = CreateTableOne(vars = covars_only, ### use variables of interest in Table 1
                                strata = "qsmk", ### stratify by treatment exposure
                                data = ps_MatchIndex, ### data is our indexed data from above
                                test = F) ### no hypothesis tests

### print table 1
print(ps_MatchPkg_T1, smd = TRUE)
```

```
                            Stratified by qsmk
                             0             1             SMD
  n                            428           428
  active (mean (sd))          0.69 (0.64)   0.69 (0.66)   0.004
  age (mean (sd))            46.71 (12.28) 46.70 (12.52)  0.001
  alcoholfreq (mean (sd))     1.98 (1.34)   1.95 (1.32)   0.019
  alcoholhowmuch (mean (sd))  2.86 (2.11)   2.91 (2.21)   0.024
  alcoholtype (mean (sd))     2.59 (1.20)   2.52 (1.20)   0.059
  education (mean (sd))       2.74 (1.21)   2.75 (1.28)   0.009
  exercise (mean (sd))        1.26 (0.73)   1.25 (0.72)   0.019
  income (mean (sd))         18.12 (2.58)  18.09 (2.45)   0.014
  race (mean (sd))            0.10 (0.30)   0.09 (0.28)   0.040
  sex (mean (sd))             0.44 (0.50)   0.45 (0.50)   0.005
  smokeintensity (mean (sd)) 18.87 (10.46) 18.79 (12.26)  0.007
  smokeyrs (mean (sd))       26.80 (12.45) 26.61 (13.03)  0.015
  wt71 (mean (sd))           71.91 (14.73) 72.63 (16.08)  0.047
  tax71 (mean (sd))           1.05 (0.20)   1.06 (0.21)   0.026
  price71 (mean (sd))         2.13 (0.22)   2.13 (0.23)   0.001
```

Overall there are no obvious issues with the balance achieved from this matched population. Each of the covariates have an SMD that is well below the 0.10 threshold.

One-to-one Matching, Caliper = 0.2

Similarly, we use the Matching package to match the propensity scores with a maximum allowable caliper distance of 0.2 (caliper of $0.2 * sd(logit(ps))$). We assess the SMD's of the covariates in this 1:1 matched population.

```
### Matching Package - with caliper distance
ps_MatchPkg_C = Match(Tr = (df_includeCovar$qsmk == 1), ### treatment
                      X = log(df_includeCovar$ps_qsmk/df_includeCovar$ps_notqsmk), ### log(ps / 1-ps)
                      M = 1, ### 1:1 matching
                      caliper = 0.2, ### caliper distance of 0.2 * sd(logit(ps))
                      replace = F, ### no replacement
                      ties = T, ### true that PS's can be tied for treated/untreated
                      version = "fast")

### pull out the matched data by their index as treated and control (given from Match package) by rows
ps_MatchIndex_C = df_includeCovar[unlist(ps_MatchPkg_C[c("index.treated","index.control")]) , ]

### CreateTableOne
ps_MatchPkg_T1_C = CreateTableOne(vars = covars_only, ### use variables of interest in Table 1
                                  strata = "qsmk", ### stratify by treatment exposure
                                  data = ps_MatchIndex_C, ### data is our indexed data from above
                                  test = F) ### no hypothesis tests

### print table 1
print(ps_MatchPkg_T1_C, smd = TRUE)
```

```
                            Stratified by qsmk
                             0              1            SMD
  n                            415           415
  active (mean (sd))          0.72 (0.66)   0.68 (0.66)   0.062
  age (mean (sd))            46.01 (12.26) 46.11 (12.20)  0.009
  alcoholfreq (mean (sd))     1.94 (1.33)   1.96 (1.32)   0.009
  alcoholhowmuch (mean (sd))  2.91 (2.51)   2.88 (2.09)   0.011
  alcoholtype (mean (sd))     2.51 (1.20)   2.53 (1.20)   0.020
  education (mean (sd))       2.73 (1.20)   2.77 (1.28)   0.033
  exercise (mean (sd))        1.26 (0.72)   1.24 (0.72)   0.034
  income (mean (sd))         18.00 (2.60)  18.11 (2.45)   0.044
  race (mean (sd))            0.11 (0.31)   0.09 (0.29)   0.064
  sex (mean (sd))             0.46 (0.50)   0.45 (0.50)   0.015
  smokeintensity (mean (sd)) 18.58 (10.64) 19.20 (12.21)  0.054
  smokeyrs (mean (sd))       26.18 (12.37) 26.48 (12.86)  0.024
  wt71 (mean (sd))           72.21 (14.87) 72.48 (16.20)  0.017
  tax71 (mean (sd))           1.04 (0.22)   1.05 (0.21)   0.039
  price71 (mean (sd))         2.12 (0.23)   2.13 (0.23)   0.034
```

The results are fairly similar to that of the non-caliper matching, with all covariates having an SMD < 0.10.

Using MatchIt to Match Patients

For comparison purposes, we use the MatchIt package which will perform a regression, assign the propensity scores as needed, and provide summary results all at once. Using the same approach as before, we explore the results of this package with and without a caliper of 0.2.

One-to-One MatchIt, No Caliper

The `matchit()` function specifies the regression model (rather than just specifying the propensity scores to match). We then assess the SMD's of the covariates in a 1:1 matched population using MatchIt.

```
### MatchIt Package - nearest neighbors without caliper (automatic PS calculation)
ps_MatchItReg = matchit(qsmk ~ active + age + alcoholfreq + alcoholhowmuch +
                               alcoholtype + wt71 + education + exercise + income +
                               race + sex + smokeintensity + smokeyrs + tax71 + price71,
                        method = "nearest",
                        data = df_includeCovar)

### match the scores
ps_MatchIt = match.data(ps_MatchItReg)

### CreateTableOne - matchit without caliper
ps_MatchIt_T1 = CreateTableOne(vars = covars_only,
                              strata = "qsmk",
                              data = ps_MatchIt,
                              test = F)

### print table 1
print(ps_MatchIt_T1, smd = TRUE)
```

```
                            Stratified by qsmk
                             0             1             SMD
  n                            428           428
  active (mean (sd))          0.70 (0.66)   0.69 (0.66)   0.014
  age (mean (sd))            46.40 (12.18) 46.70 (12.52)  0.024
  alcoholfreq (mean (sd))     1.99 (1.35)   1.95 (1.32)   0.028
  alcoholhowmuch (mean (sd))  2.87 (2.30)   2.91 (2.21)   0.020
  alcoholtype (mean (sd))     2.56 (1.20)   2.52 (1.20)   0.033
  education (mean (sd))       2.75 (1.18)   2.75 (1.28)  <0.001
  exercise (mean (sd))        1.24 (0.72)   1.25 (0.72)   0.013
  income (mean (sd))         18.14 (2.52)  18.09 (2.45)   0.022
  race (mean (sd))            0.09 (0.29)   0.09 (0.28)   0.008
  sex (mean (sd))             0.46 (0.50)   0.45 (0.50)   0.019
  smokeintensity (mean (sd)) 18.64 (10.43) 18.79 (12.26)  0.014
  smokeyrs (mean (sd))       26.51 (12.15) 26.61 (13.03)  0.008
  wt71 (mean (sd))           71.95 (14.33) 72.63 (16.08)  0.045
  tax71 (mean (sd))           1.05 (0.22)   1.06 (0.21)   0.022
  price71 (mean (sd))         2.13 (0.23)   2.13 (0.23)   0.001
```

To no surprise, this performs similarly to our previous matching efforts, with all the covariates having an SMD under a 10% difference.

One-to-One MatchIt, Caliper = 0.2

As before, we use the MatchIt package to compare the results of using a caliper of 0.2, and assess the balance (SMD) of the covariates in this model.

```
### MatchIt Package - nearest neighbors with caliper (automatic PS calculation)
ps_MatchItReg_C = matchit(qsmk ~ active + age + alcoholfreq + alcoholhowmuch +
                              alcoholtype + wt71 + education + exercise + income +
                              race + sex + smokeintensity + smokeyrs + tax71 + price71,
                       method = "nearest",
                       caliper = 0.2,
                       data = df_includeCovar)

### match the scores
ps_MatchIt_C = match.data(ps_MatchItReg_C)

### CreateTableOne - matchitwith caliper
ps_MatchIt_T1_C = CreateTableOne(vars = covars_only,
                                 strata = "qsmk",
                                 data = ps_MatchIt_C,
                                 test = F)

### print table 1
print(ps_MatchIt_T1_C, smd = TRUE)
```

```
                            Stratified by qsmk
                             0             1             SMD
  n                            418           418
  active (mean (sd))          0.65 (0.66)   0.67 (0.66)   0.040
  age (mean (sd))            46.08 (11.80) 46.32 (12.36)  0.020
  alcoholfreq (mean (sd))     1.92 (1.36)   1.94 (1.32)   0.021
  alcoholhowmuch (mean (sd))  2.86 (2.10)   2.94 (2.22)   0.035
  alcoholtype (mean (sd))     2.51 (1.23)   2.53 (1.20)   0.018
  education (mean (sd))       2.81 (1.25)   2.75 (1.28)   0.046
  exercise (mean (sd))        1.23 (0.72)   1.24 (0.72)   0.007
  income (mean (sd))         18.10 (2.60)  18.11 (2.44)   0.004
  race (mean (sd))            0.10 (0.30)   0.09 (0.29)   0.032
  sex (mean (sd))             0.45 (0.50)   0.45 (0.50)   0.005
  smokeintensity (mean (sd)) 19.46 (11.21) 18.98 (12.28)  0.041
  smokeyrs (mean (sd))       26.19 (11.93) 26.64 (12.94)  0.036
  wt71 (mean (sd))           72.16 (14.49) 72.45 (16.14)  0.019
  tax71 (mean (sd))           1.07 (0.22)   1.06 (0.22)   0.066
  price71 (mean (sd))         2.14 (0.23)   2.13 (0.23)   0.057
```

All the covariates are well within the allowable 10% range for the SMD, similar to all the previous examples.

Inverse Probability Weighting

To assess the performance and results of IPW, we'll use both stabilized and unstabilized weighting, and assess the distribution of weights for any extremes.

Unstabilized Weighting

We perform a logistic regression with the exposure as the outcome. The `ipwpoint()` function will automatically weight the population to the counterfactuals.

```
### ipw unstabilized model
iptw_unstabilized = ipwpoint(exposure = qsmk, family="binomial", link="logit",
                             denominator = ~ active + age + alcoholfreq + alcoholhowmuch +
                                             alcoholtype + wt71 + education + exercise + income +
                                             race + sex + smokeintensity + smokeyrs + tax71 + price71,
                             data=df_includeCovar)

### assign the sipw weights
df_includeCovar$ipw = iptw_unstabilized$ipw.weights
```

Assessment of the weights.

```
### weights for those who quit smoking (summary: min max quartiles)
### pull the weights from the data frame (weights = stipw)
### paste the summary information and set the annotations into a dataframe
ipw1 = summary(df_includeCovar$ipw[df_includeCovar$qsmk==1])
ipw1_format = paste(names(ipw1), format(ipw1, digits=2), collapse="\n ")
annotate_ipw1 = data.frame(x=3.5, y=300, lab = ipw1_format,
                           qsmk = factor(labelsCS[1], levels=c(labelsCS[1], labelsCS[2])))
### check the data
ipw1_format; annotate_ipw1


### weights for those who didnt quit smoking (summary: min max quartiles)
ipw0 = summary(df_includeCovar$ipw[df_includeCovar$qsmk==0])
ipw0_format = paste(names(ipw0), format(ipw0, digits=2), collapse="\n ")
annotate_ipw0 = data.frame(x=3.5, y=300, lab = ipw0_format,
                        qsmk = factor(labelsCS[2], levels=c(labelsCS[1], labelsCS[2])))
### check the data
ipw0_format; annotate_ipw0

### bind the annotated summaries for the weights where qsmk==1 and qsmk==0
ipw_total = rbind(annotate_ipw1, annotate_ipw0); ipw_total

#plot from the dataset itself using the pipe function %>%
#we mutate the data so that people with qsmk==1 get a label of quit smoking, else no quit smk
#plot the weight as a histogram
#use facet_wrap to plot separate the histograms for smoking status
#use geom_text to add the summary information about the weight to the corner of the plot
df_includeCovar %>%
  mutate(qsmk = ifelse(qsmk==1, labelsCS[1], labelsCS[2])) %>%
  ggplot(aes(x = ipw)) +
  geom_histogram(color = "white") +
  facet_wrap(~qsmk, nrow=2, ncol=1) +
  xlab("Weights") +
  ylab("Count") +
  ggtitle("Weights: IPW") +
  geom_text(data = ipw_total, aes(x=4, y=350, label=lab, group=NULL)) +
  theme_bw() +
  theme(text = element_text(size=15))
```

Overall, the weights do have not seriously extreme values. The weights are not as equally dispersed by group, as those who quit smoking have higher weighting comparatively, however we will refrain from truncation (Fig. 4).

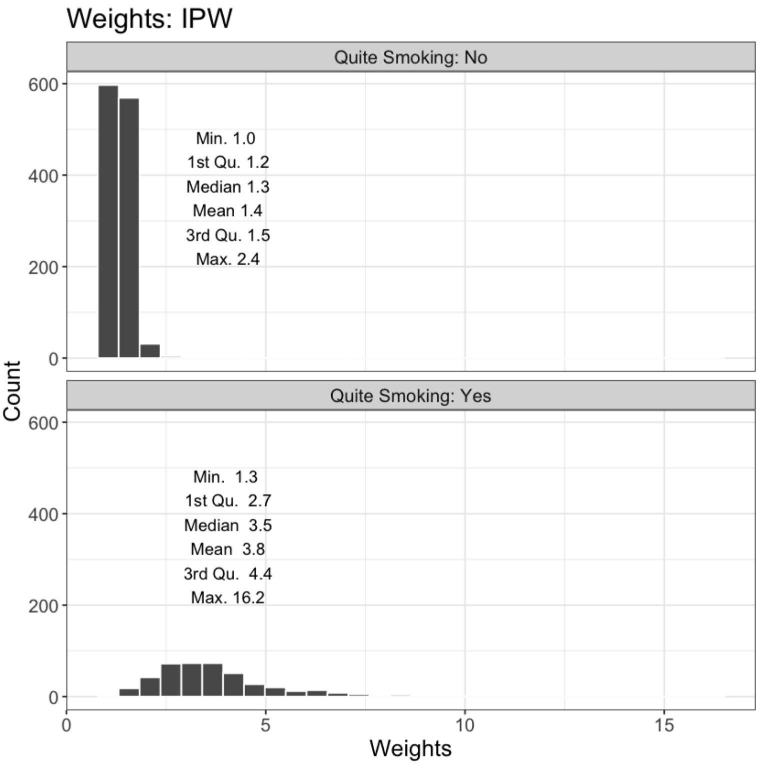


Figure 4: Unstabilized Weighting Distribution

We assess balance in the unstabilized population using the `svydesign()` function, which gives better results for calculating a Table One. As before, each of the covariates exhibit a sufficient SMD.

```
### setup the IPW model with svydesign package (this does a regression using robust SEs)
### we need to run a regression to reweight the population, then we can check balance on this new population
ipw_unstabilized_svy = svydesign(ids = ~ 1, data = df_includeCovar, weights = ~ ipw)

### table 1
ipwTableOne_unstabilized = svyCreateTableOne(vars = covars_only, strata="qsmk",
                                              data=ipw_unstabilized_svy, test=F)

### print table 1
print(ipwTableOne_unstabilized, smd=T)
```

```
                             Stratified by qsmk
                              0                1                SMD
  n                           1627.93          1621.11
  active (mean (sd))             0.65 (0.65)      0.66 (0.64)    0.016
  age (mean (sd))               43.89 (12.10)    44.13 (12.32)   0.019
  alcoholfreq (mean (sd))        1.92 (1.31)      1.92 (1.31)   <0.001
  alcoholhowmuch (mean (sd))     3.04 (2.62)      2.93 (2.16)    0.043
  alcoholtype (mean (sd))        2.48 (1.21)      2.51 (1.19)    0.028
  education (mean (sd))          2.71 (1.17)      2.75 (1.24)    0.032
  exercise (mean (sd))           1.19 (0.74)      1.19 (0.71)    0.002
  income (mean (sd))            17.97 (2.66)     18.04 (2.46)    0.026
  race (mean (sd))               0.13 (0.34)      0.14 (0.34)    0.009
  sex (mean (sd))                0.51 (0.50)      0.51 (0.50)    0.002
  smokeintensity (mean (sd))    20.58 (11.41)    20.40 (12.95)   0.015
  smokeyrs (mean (sd))          24.88 (11.98)    25.05 (12.78)   0.013
  wt71 (mean (sd))              71.02 (15.52)    71.11 (16.65)   0.005
  tax71 (mean (sd))              1.05 (0.21)      1.05 (0.21)    0.021
  price71 (mean (sd))            2.13 (0.23)      2.13 (0.22)    0.019
```

Stabilized Weighting

The ipw package can perform both stabilized and unstabilized weighting. To perform stabilized weighting, the numerator has to be specified as 1. When the numerator is omitted it instead calculates an unstabilized weight. This may seem counterintuitive, as a by-hand calculation tells us the IPW is $1/f(A \mid L)$, and that the SW is $f(A)/f(A \mid L)$. However, the ipw package requires the numerator be set to 1 when performing stabilized weighting. Be aware of this when performing an IPW analysis in R.

```
### ipw model
iptw_stabilized = ipwpoint(exposure = qsmk, family="binomial", link="logit",
                                       numerator = ~ 1, ### expressing this tells R to do stabilized
                                       denominator = ~ active + age + alcoholfreq + alcoholhowmuch +
                                                       alcoholtype + wt71 + education + exercise + income +
                                                       race + sex + smokeintensity + smokeyrs + tax71 +
                                                       price71,
                                       data=df_includeCovar)

### assign the sipw weights
df_includeCovar$sw = iptw_stabilized$ipw.weights
```

Assessment of the weights

```
### weights for those who quit smoking (summary: min max quartiles)
### pull the weights from the data frame (weights = stipw)
### paste the summary information and set the annotations into a dataframe
sw1 = summary(df_includeCovar$sw[df_includeCovar$qsmk==1])
sw1_format = paste(names(sw1), format(sw1, digits=2), collapse="\n ")
annotate_sw1 = data.frame(x=3.5, y=300, lab = sw1_format,
                          qsmk = factor(labelsCS[1], levels=c(labelsCS[1], labelsCS[2])))
### check the data
sw1_format; annotate_sw1


### weights for those who didnt quit smoking (summary: min max quartiles)
sw0 = summary(df_includeCovar$sw[df_includeCovar$qsmk==0])
sw0_format = paste(names(sw0), format(sw0, digits=2), collapse="\n ")
annotate_sw0 = data.frame(x=3.5, y=300, lab = sw0_format,
                          qsmk = factor(labelsCS[2], levels=c(labelsCS[1], labelsCS[2])))
### check the data
sw0_format; annotate_sw0

### bind the annotated summaries for the weights where qsmk==1 and qsmk==0
sw_total = rbind(annotate_sw1, annotate_sw0); sw_total

#plot from the dataset itself using the pipe function %>%
#we mutate the data so that people with qsmk==1 get a label of quit smoking, else no quit smk
#plot the weight as a histogram
#use facet_wrap to plot separate the histograms for smoking status
#use geom_text to add the summary information about the weight to the corner of the plot
df_includeCovar %>%
  mutate(qsmk = ifelse(qsmk==1, labelsCS[1], labelsCS[2])) %>%
  ggplot(aes(x = sw)) +
  geom_histogram(color = "white") +
  facet_wrap(~qsmk, nrow=2, ncol=1) +
  xlab("Weights") +
  ylab("Count") +
  ggtitle("Weights: SW") +
  geom_text(data = sw_total, aes(x=4, y=350, label=lab, group=NULL)) +
  theme_bw() +
  theme(text = element_text(size=15))
```

Overall, the weights are far more evenly dispersed compared to unstabilized IPW, where the weighting for each exposure group has a median and mean near 1 (as is expected in stabilized weighting) (Fig. 5).

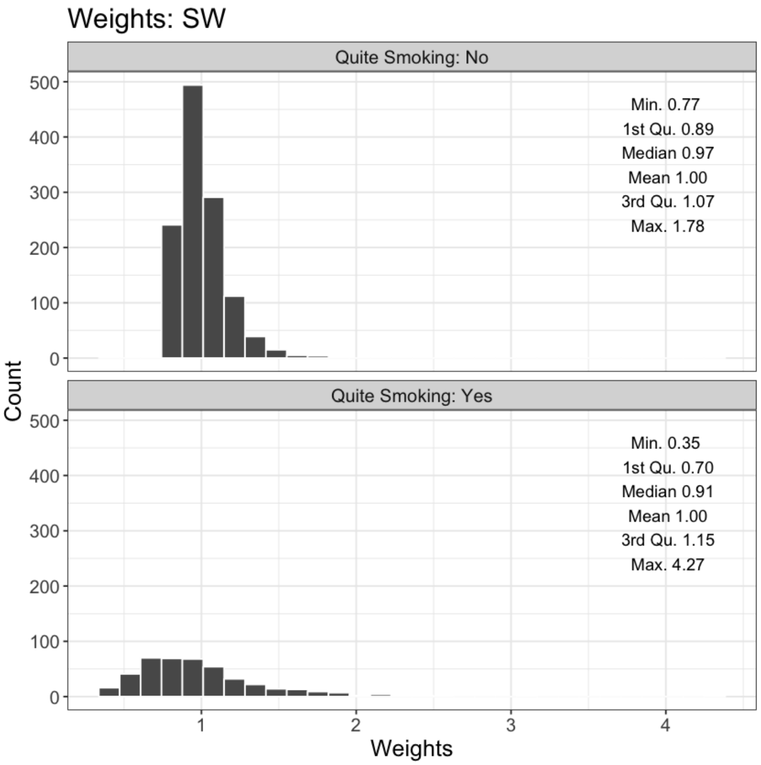


Figure 5: Unstabilized Weighting Distribution

The balance in the stabilized IPW is identical (when rounded) to unstabilized IPW, suggesting that either of these two models is sufficient with no major differences in performance (even though there are minor deviations in the distributions of weights).

```
### regression model
ipw_stabilized_svy = svydesign(ids = ~ 1, data = df_includeCovar, weights = ~ sw)

### table 1
ipwTableOne_stabilized = svyCreateTableOne(vars = covars_only, strata="qsmk",
                                           data=ipw_stabilized_svy, test=F)

### print table 1
print(ipwTableOne_stabilized, smd=T)
```

```
                           Stratified by qsmk
                            0              1              SMD
  n                         1200.21        425.93
  active (mean (sd))           0.65 (0.65)    0.66 (0.64)   0.016
  age (mean (sd))             43.89 (12.10)  44.13 (12.32)  0.019
  alcoholfreq (mean (sd))      1.92 (1.31)    1.92 (1.31)  <0.001
  alcoholhowmuch (mean (sd))   3.04 (2.62)    2.93 (2.16)   0.043
  alcoholtype (mean (sd))      2.48 (1.21)    2.51 (1.19)   0.028
  education (mean (sd))        2.71 (1.17)    2.75 (1.24)   0.032
  exercise (mean (sd))         1.19 (0.74)    1.19 (0.71)   0.002
  income (mean (sd))          17.97 (2.66)   18.04 (2.46)   0.026
  race (mean (sd))             0.13 (0.34)    0.14 (0.34)   0.009
  sex (mean (sd))              0.51 (0.50)    0.51 (0.50)   0.002
  smokeintensity (mean (sd))  20.58 (11.41)  20.40 (12.95)  0.015
  smokeyrs (mean (sd))        24.88 (11.98)  25.05 (12.78)  0.013
  wt71 (mean (sd))            71.02 (15.52)  71.11 (16.65)  0.005
  tax71 (mean (sd))            1.05 (0.21)    1.05 (0.21)   0.021
  price71 (mean (sd))          2.13 (0.23)    2.13 (0.22)   0.019
```

Overall assessment of balance

A useful depiction of conditional exchangeability (given no unmeasured confounding) is to combine all the results of the standardized mean differences for each method into a single plot. This consists of a line graph, with a reference held at the 10% mark to identify which covariates are within range for each of the causal methods.

```
### pull the information from each of the tables and set it into a dataframe using extractsmd()
smdPlot = data.frame(Covariates = names(ExtractSmd(unmatchedTableOne)),
                     Unmatched = ExtractSmd(unmatchedTableOne),
                     PS.MatchIt = ExtractSmd(ps_MatchIt_T1),
                     PS.MatchIt.C = ExtractSmd(ps_MatchIt_T1_C),
                     PS.Matched = ExtractSmd(ps_MatchPkg_T1),
                     PS.Matched.C = ExtractSmd(ps_MatchPkg_T1_C),
                     IPW = ExtractSmd(ipwTableOne_unstabilized),
                     SW = ExtractSmd(ipwTableOne_stabilized))

### use the melt function to change/transform the data into something we can plot
### melt is the same function you use to transform from wide to long formats (longitudinal data)
smdPlotMelt = melt(data = smdPlot,
                   id.vars = c("Covariates"),
                   variable.name = "Method",
                   value.name = "SMD")

### set the variable names and order them by what's in the unmatched data (defined in changePlt)
var.Names = as.character(smdPlot$Covariates[order(smdPlot$Unmatched)])

### set up the factors of the variables for plotting (pulls each var out of the list)
smdPlotMelt$Covariates = factor(smdPlotMelt$Covariates, levels=var.Names)

### add more levels to factor
levels(smdPlotMelt$Method) <- c(levels(smdPlotMelt$Method), c("MatchIt Pkg (no caliper)",
                                                              "MatchIt Pkg (caliper=0.2)",
                                                              "PS Matching (no caliper)",
                                                              "PS Matching (caliper=0.2)"))

### rename Method values to look better in plot
smdPlotMelt$Method[smdPlotMelt$Method=="PS.MatchIt"] = "MatchIt Pkg (no caliper)"
smdPlotMelt$Method[smdPlotMelt$Method=="PS.MatchIt.C"] = "MatchIt Pkg (caliper=0.2)"
smdPlotMelt$Method[smdPlotMelt$Method=="PS.Matched"] = "PS Matching (no caliper)"
smdPlotMelt$Method[smdPlotMelt$Method=="PS.Matched.C"] = "PS Matching (caliper=0.2)"

### plot
options(jupyter.plot_mimetypes = 'image/png')

smdPlotMelt %>%
    ggplot(aes(x = Covariates, y = SMD, group = Method, color = Method, shape=Method)) +
    geom_line() +
    geom_point() +
    geom_hline(yintercept = 0.1, color = "black", size = 0.1) +
    coord_flip() +
    theme_bw() +
    ggtitle("Comparison of Propensity Score\nMethods and Implementation") +
    xlab("Covariates\n") +
    ylab("\nStandardized Mean Difference") +
    theme(legend.key = element_blank(),
          text = element_text(size=12),
          axis.title = element_text(size = 12, color="black"))
```

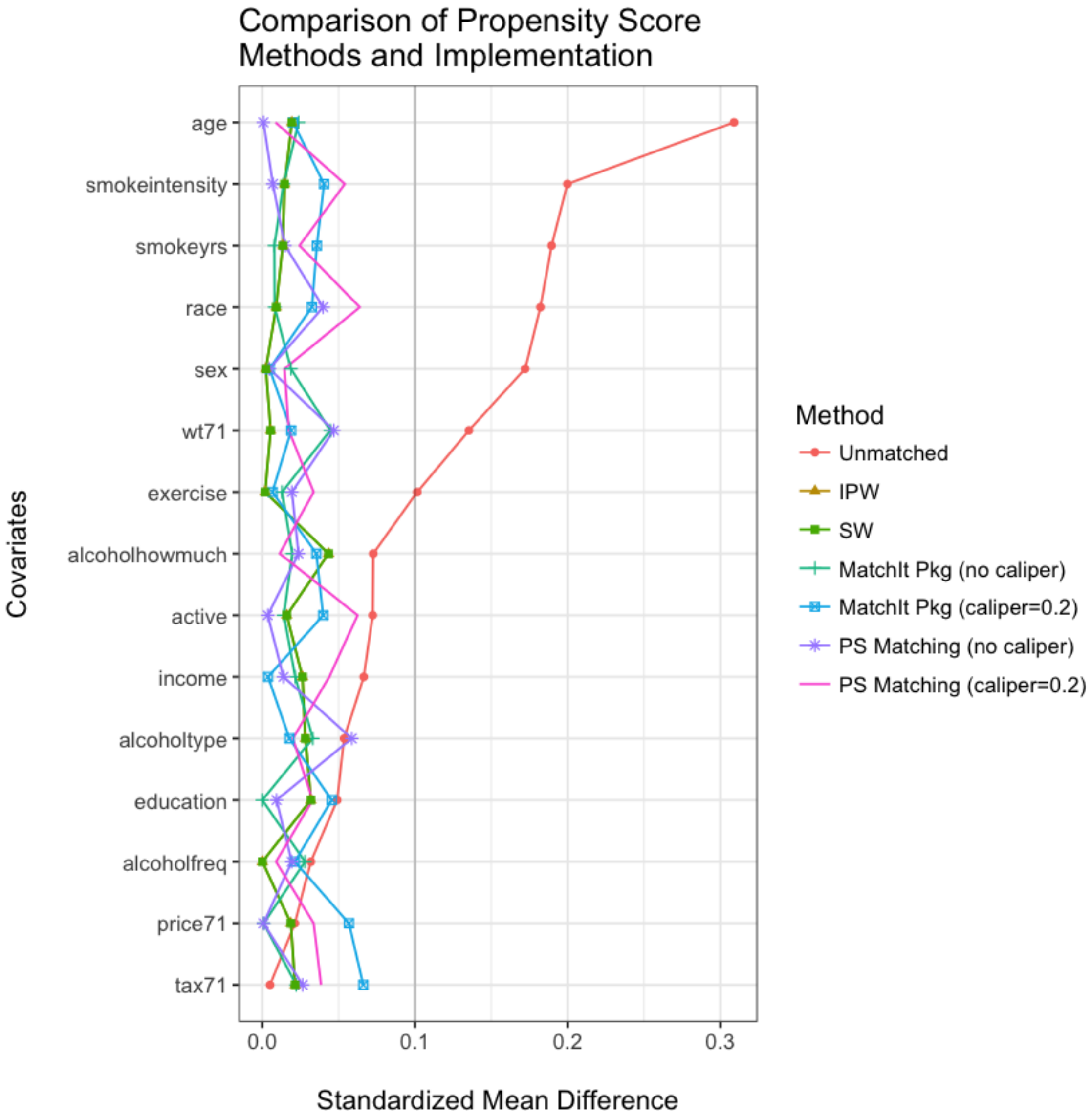


Figure 6: Comparison of the SMD for each of the causal methods used. Pkg=Package.

When compared to the original population, each of the models perform exceptionally well without any further adjustment needed. It is important to remember that if we instead did not achieve balance on a particular covariate (or several) for one or more of these methods, we could try testing out interactions of variables (e.g., perhaps there is an interaction between age and income, or alcohol frequency and how much alcohol is drank in a sitting, etc). However in this case it appears that each of the covariates are well under the 10% rule of thumb, with stabilized weighting, IPW, and PS Matching (with no caliper) performing slightly better compared to the other methods. Note: the line for IPW is underneath the SW line as they are nearly identical.

Estimate the outcome

To estimate the outcome in a regression model based on these causal methods, we first merge the outcome with the data we've been working with by ID. We'll then use this population to reweight in IPW/SW, and to implement covariate adjustment and stratification. We'll have to do similar for the special matched populations (as these are different from the original population).

```
#merge data back together
merged_df = merge(df_outcomeOnly, df_includeCovar, by="id")
```

Now, we'll perform a regression model for each of methods used, and compare the results all at once afterward.

PS Matching Results (no caliper)

For the matched propensity score population (with no caliper), we merge the outcome with the matched population we created and perform a regression, with the model as $\mathbb{E}(Y^a) = \beta_0 + \beta_1 qsmk_{i1} + \epsilon_i$, where Y^a exists in the counterfactual (non-caliper defined) matched population for the exposure a (smoking cessation).

```
### merge data for ps matched data
matchingPopulation = merge(df_outcomeOnly, ps_MatchIndex, by="id")

### perform regression
ps_regression_matching = glm(wt82_71 ~ qsmk, data=matchingPopulation)

### summary and confidence intervals
summary(ps_regression_matching); confint(ps_regression_matching)
```

```
Call:
glm(formula = wt82_71 ~ qsmk, data = matchingPopulation)

Deviance Residuals:
    Min       1Q    Median       3Q      Max
-31.912   -4.073   -0.052    4.488   42.986

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.4104     0.3959   3.563 0.000388 ***
qsmk          3.1147     0.5623   5.540  4.1e-08 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for gaussian family taken to be 64.24959)

    Null deviance: 54078  on 812  degrees of freedom
Residual deviance: 52106  on 811  degrees of freedom
  (43 observations deleted due to missingness)
AIC: 5695.5

Number of Fisher Scoring iterations: 2

                  Conf. Int
                2.5 %   97.5 %
(Intercept) 0.6344852   2.186234
qsmk        2.0127132   4.216726
```

PS Matching Results (caliper = 0.2)

We do similar for the matched population with a caliper of 0.2. Here, the model is represented as

$$\mathbb{E}(Y^a) = \beta_0 + \beta_1 qsmk_{i1} + \epsilon_i$$

where Y^a exists in the counterfactual (caliper defined) matched population for the exposure a (smoking cessation).

```
### merge data for ps matched data with caliper
matchingPopulation_C = merge(df_outcomeOnly, ps_MatchIndex_C, by="id")

### perform regression
ps_regression_matching_C = glm(wt82_71 ~ qsmk, data=matchingPopulation_C)

### summary and confidence intervals
summary(ps_regression_matching_C); confint(ps_regression_matching_C)
```

```
Call:
glm(formula = wt82_71 ~ qsmk, data = matchingPopulation_C)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-31.680   -4.208   -0.131    4.522   42.861

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.6302     0.3923   4.156 3.59e-05 ***
qsmk          3.0200     0.5576   5.416 8.08e-08 ***
---
Signif. codes:  0   ***   0.001   **   0.01   *   0.05   .   0.1        1

(Dispersion parameter for gaussian family taken to be 61.55188)

    Null deviance: 50432  on 791  degrees of freedom
Residual deviance: 48626  on 790  degrees of freedom
  (38 observations deleted due to missingness)
AIC: 5514.5

Number of Fisher Scoring iterations: 2

                  Conf. Int
                2.5 %   97.5 %
(Intercept) 0.8613969   2.399087
qsmk        1.9271773   4.112866
```

MatchIt Results (no caliper)

For the MatchIt data, we merge the outcome with the matched population and perform the outcome regression. Here, the model is represented as

$$\mathbb{E}(Y^a) = \beta_0 + \beta_1 qsmk_{i1} + \epsilon_i$$

where Y^a exists in the counterfactual (non-caliper defined) matched population for the exposure a (smoking cessation) using MatchIt.

```
### merge data for ps matched data with caliper
matchitPopulation = merge(df_outcomeOnly, ps_MatchIt, by="id")

### perform regression
ps_regression_matchit = glm(wt82_71 ~ qsmk, data=matchitPopulation)

### summary and confidence intervals
summary(ps_regression_matchit); confint(ps_regression_matchit)
```

```
Call:
glm(formula = wt82_71 ~ qsmk, data = matchitPopulation)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-31.990   -4.003   -0.010    4.417   42.986

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.4877     0.3871   3.843 0.000131 ***
qsmk          3.0374     0.5495   5.528 4.37e-08 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for gaussian family taken to be 61.28578)

    Null deviance: 51514  on 811  degrees of freedom
Residual deviance: 49641  on 810  degrees of freedom
  (44 observations deleted due to missingness)
AIC: 5650.2

Number of Fisher Scoring iterations: 2

                  Conf. Int
                2.5 %   97.5 %
(Intercept) 0.7290333   2.246420
qsmk        1.9604116   4.114293
```

MatchIt Results (caliper = 0.2)

Again, for the MatchIt data with a caliper, we merge the outcome and perform the outcome regression. Here, the model is represented as

$$\mathbb{E}(Y^a) = \beta_0 + \beta_1 qsmk_{i1} + \epsilon_i$$

where Y^a exists in the counterfactual (caliper defined) matched population for the exposure a (smoking cessation) using MatchIt.

```
### merge data for ps matched data with caliper
matchitPopulation_C = merge(df_outcomeOnly, ps_MatchIt_C, by="id")

### perform regression
ps_regression_matchit_C = glm(wt82_71 ˜ qsmk, data=matchitPopulation_C)

### summary and confidence intervals
summary(ps_regression_matchit_C); confint(ps_regression_matchit_C)
```

```
Call:
glm(formula = wt82_71 ˜ qsmk, data = matchitPopulation_C)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-27.629   -4.153   -0.072    4.582   42.859

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.6588     0.4008   4.139 3.86e-05 ***
qsmk          2.9937     0.5693   5.258 1.87e-07 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for gaussian family taken to be 64.5754)

    Null deviance: 53123  on 796  degrees of freedom
Residual deviance: 51337  on 795  degrees of freedom
  (39 observations deleted due to missingness)
AIC: 5587.5

Number of Fisher Scoring iterations: 2

                  Conf. Int
                2.5 %   97.5 %
(Intercept) 0.8733068   2.444388
qsmk        1.8778945   4.109561
```

IPW

For inverse probability weights, we use the survey design package to run the regression using the appropriate weight. Here, the model is represented as

$$\mathbb{E}(Y^a) = \beta_0 + \beta_1 qsmk_{i1} + \epsilon_i$$

where Y^a exists in the counterfactual population for the exposure a (smoking cessation), using an unstabilized weight to reweight the original population.

```r
### have to use the survey design package for weighted data
ipw_rg = (svyglm(wt82_71 ~ qsmk, design = svydesign(~1, weights=~ipw,
          data=merged_df)))

### summary and confidence intervals
summary(ipw_rg); confint(ipw_rg)
```

```
Call:
svyglm(formula = wt82_71 ~ qsmk, design = svydesign(~1, weights = ~ipw,
    data = merged_df))

Survey design:
svydesign(~1, weights = ~ipw, data = merged_df)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.7882     0.2222   8.049 1.64e-15 ***
qsmk          3.2703     0.5220   6.265 4.80e-10 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for gaussian family taken to be 64.7786)

Number of Fisher Scoring iterations: 2

                  Conf. Int
                2.5 %   97.5 %
(Intercept) 1.352710    2.223604
qsmk        2.247239    4.293301
```

SW

Similarly to IPW, we use the survey design package to run the appropriate regression with the stabilized weights. Here, the model is represented as

$$\mathbb{E}(Y^a) = \beta_0 + \beta_1 qsmk_{i1} + \epsilon_i$$

where Y^a exists in the counterfactual population for the exposure a (smoking cessation), using a stabilized weight to reweight the original population.

```
### have to use the survey design package for weighted data
sw_rg = (svyglm(wt82_71 ˜ qsmk, design = svydesign(˜1, weights=˜sw,
          data=merged_df)))

### summary and confidence intervals
summary(sw_rg); confint(sw_rg)
```

```
Call:
svyglm(formula = wt82_71 ˜ qsmk, design = svydesign(˜1, weights = ˜sw,
    data = merged_df))

Survey design:
svydesign(˜1, weights = ˜sw, data = merged_df)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.7882     0.2222   8.049 1.64e-15 ***
qsmk          3.2703     0.5220   6.265 4.80e-10 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for gaussian family taken to be 60.66354)

Number of Fisher Scoring iterations: 2

                  Conf. Int
                2.5 %   97.5 %
(Intercept) 1.352710    2.223604
qsmk        2.247239    4.293301
```

Stratification

For a stratified propensity score model, we typically use deciles to create an even grouping of the propensity score. Five strata are commonly used, however we'll use 10 deciles. The goal is to create a strata variable that captures the level of the propensity score that we can then adjust for in a regression model (with one of the strata's held as reference). Here, the model is represented as

$$\mathbb{E}(y_i) = \beta_0 + \beta_1 qsmk_{i1} + \beta_2 strata_{i2} + \epsilon_i$$

where *strata* represents the information about the stratified propensity score .

Interestingly, the propensity score does not exist at higher levels of strata, and thus we only have information on the propensity score strata up to the 6th-decile (with the 1st-decile held as reference) (below).

```
### define a stratum for the PS to take on deciles (as a vector)
strat = quantile(merged_df$pAssign, probs=c(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9))

### set the stratums range of PS
strat1 = merged_df$ps_qsmk <= strat[1]
strat2 = merged_df$ps_qsmk > strat[1] & merged_df$ps_qsmk <= strat[2]
strat3 = merged_df$ps_qsmk > strat[2] & merged_df$ps_qsmk <= strat[3]
strat4 = merged_df$ps_qsmk > strat[3] & merged_df$ps_qsmk <= strat[4]
strat5 = merged_df$ps_qsmk > strat[4] & merged_df$ps_qsmk <= strat[5]
strat6 = merged_df$ps_qsmk > strat[5] & merged_df$ps_qsmk <= strat[6]
strat7 = merged_df$ps_qsmk > strat[6] & merged_df$ps_qsmk <= strat[7]
strat8 = merged_df$ps_qsmk > strat[7] & merged_df$ps_qsmk <= strat[8]
strat9 = merged_df$ps_qsmk > strat[8] & merged_df$ps_qsmk <= strat[9]
strat10 = merged_df$ps_qsmk > strat[9]

### setup of the a single variable to take on categorical values
merged_df$stratvar = numeric(length(merged_df$qsmk))

### assign the categorical values through a forloop
### for i = 1 until the max length of the dataset, assign values
for (i in 1:length(merged_df$qsmk)) {
    if (strat1[i]==T) {merged_df$stratvar[i] = 1}
    else if (strat2[i]==T) {merged_df$stratvar[i] = 2}
    else if (strat3[i]==T) {merged_df$stratvar[i] = 3}
    else if (strat4[i]==T) {merged_df$stratvar[i] = 4}
    else if (strat5[i]==T) {merged_df$stratvar[i] = 5}
    else if (strat6[i]==T) {merged_df$stratvar[i] = 6}
    else if (strat7[i]==T) {merged_df$stratvar[i] = 7}
    else if (strat8[i]==T) {merged_df$stratvar[i] = 8}
    else if (strat9[i]==T) {merged_df$stratvar[i] = 9}
    else merged_df$stratvar[i] = 10
}

### regression as deciles
strmodel1 = glm(wt82_71 ~ qsmk + as.factor(stratvar), data = merged_df)
summary(strmodel1); confint(strmodel1)
```

```
Call:
glm(formula = wt82_71 ~ qsmk + as.factor(stratvar), data = merged_df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-42.261   -4.151   -0.092    4.082   47.558

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)            2.9521     0.2798  10.552  < 2e-16 ***
qsmk                   3.2383     0.4551   7.116 1.69e-12 ***
as.factor(stratvar)2  -1.9717     0.4322  -4.562 5.47e-06 ***
as.factor(stratvar)3  -3.2893     0.6078  -5.412 7.22e-08 ***
as.factor(stratvar)4  -6.9530     2.4643  -2.822 0.004840 **
as.factor(stratvar)6 -27.2856     7.6923  -3.547 0.000401 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for gaussian family taken to be 58.96332)

    Null deviance: 97176  on 1565  degrees of freedom
Residual deviance: 91983  on 1560  degrees of freedom
  (63 observations deleted due to missingness)
AIC: 10837

Number of Fisher Scoring iterations: 2

                              Conf. Int
                            2.5 %   97.5 %
(Intercept)               2.403744   3.500365
qsmk                      2.346354   4.130296
as.factor(stratvar)2     -2.818832  -1.124517
as.factor(stratvar)3     -4.480528  -2.097992
as.factor(stratvar)4    -11.782951  -2.123103
as.factor(stratvar)6    -42.362214  -12.208987
```

Continuous Adjustment

For a continuous adjustment of the propensity score, we perform a regression with the outcome of interest as the dependent variable, and the exposure and propensity score as the independent variables. This is a way in which we can capture all the covariate information whilst reducing dimensionality. Here, the model is represented as $\mathbb{E}(y_i) = \beta_0 + \beta_1 qsmk_{i1} + \beta_2 p_{i2} + \epsilon_i$, where p represents the continuous propensity score.

```r
### regression as continuous - this is done on the original dataset
### its a like a unique form of multivariate regression
### where the PS captures all the covariate information
strmodel = glm(wt82_71 ~ qsmk + ps_qsmk, data=merged_df)
summary(strmodel); confint(strmodel)
```

```
Call:
glm(formula = wt82_71 ~ qsmk + ps_qsmk, data = merged_df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-42.721   -4.116   -0.117    4.211   48.037

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.4746     0.5169  10.592  < 2e-16 ***
qsmk          3.3593     0.4566   7.358 3.01e-13 ***
ps_qsmk     -14.1476     1.8865  -7.500 1.07e-13 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1        1

(Dispersion parameter for gaussian family taken to be 58.81996)

    Null deviance: 97176  on 1565  degrees of freedom
Residual deviance: 91936  on 1563  degrees of freedom
  (63 observations deleted due to missingness)
AIC: 10830

Number of Fisher Scoring iterations: 2

                  Conf. Int
                2.5 %   97.5 %
(Intercept) 4.461547    6.487620
qsmk        2.464440    4.254129
ps_qsmk   -17.844953    -10.450166
```

Multivariate Regression

We will perform a multivariate regression for comparison purposes. This regression model will clearly violate the "10-covariate" rule, and thus runs the risk of overfitting (although in this setting the non-significant covariates could then be removed). Here, the model is represented as

$$\begin{aligned}
\mathbb{E}(y_i) =& \beta_0 + \beta_1 qsmk_{i1} + \beta_2 active_{i2} + \beta_3 age_{i3} + \beta_4 alcoholfreq_{i4} + \beta_5 alcoholhowmuch_{i5} + \\
& \beta_6 alcoholtype_{i6} + \beta_7 wt71_{i7} + \beta_8 education_{i8} + \beta_9 exercise_{i9} + \beta_{10} income_{i10} + \\
& \beta_{11} race_{i11} + \beta_{12} sex_{i12} + \beta_{13} smokeintensity_{i13} + \beta_{14} smokeyrs_{i14} + \\
& \beta_{15} price71_{i15} + \beta_{16} tax71_{i16} + \epsilon_i
\end{aligned}$$

The resulting estimate of effect is similar to the previous causal models.

```
### multivariate regression
rg = glm(wt82_71 ~ qsmk + active + age + alcoholfreq +
                   alcoholhowmuch + alcoholtype + wt71 + education +
                   exercise + income + race + sex + smokeintensity +
                   smokeyrs + price71 + tax71,
         data=merged_df)

### print summary and confidence intervals
summary(rg); confint(rg)
```

```
Call:
glm(formula = wt82_71 ~ qsmk + active + age + alcoholfreq + alcoholhowmuch +
    alcoholtype + wt71 + education + exercise + income + race +
    sex + smokeintensity + smokeyrs + price71 + tax71, data = merged_df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-41.789   -4.192   -0.288    3.994   45.752

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    16.65028    3.95351   4.212 2.68e-05 ***
qsmk            3.31917    0.44194   7.511 9.90e-14 ***
active         -0.55953    0.30471  -1.836  0.06651 .
age            -0.20640    0.03317  -6.223 6.27e-10 ***
alcoholfreq     0.34832    0.18728   1.860  0.06310 .
alcoholhowmuch -0.08624    0.07530  -1.145  0.25224
alcoholtype    -0.31147    0.20492  -1.520  0.12874
wt71           -0.09982    0.01373  -7.268 5.76e-13 ***
education       0.04337    0.18791   0.231  0.81748
exercise        0.21048    0.27678   0.760  0.44709
income          0.05186    0.08428   0.615  0.53846
race            0.69129    0.60143   1.149  0.25057
sex            -1.33694    0.47687  -2.804  0.00512 **
smokeintensity  0.02703    0.01721   1.571  0.11637
smokeyrs        0.05534    0.03359   1.648  0.09963 .
price71        -1.16504    2.81111  -0.414  0.67861
tax71           1.70745    2.99766   0.570  0.56903
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for gaussian family taken to be 55.12124)

    Null deviance: 97176  on 1565  degrees of freedom
Residual deviance: 85383  on 1549  degrees of freedom
  (63 observations deleted due to missingness)
AIC: 10742

Number of Fisher Scoring iterations: 2

                          Conf. Int
                         2.5 %   97.5 %
(Intercept)      8.901542455    24.39901422
qsmk             2.452992087    4.18534959
active          -1.156743425    0.03769001
age             -0.271406540    -0.14139168
alcoholfreq     -0.018749793    0.71538586
alcoholhowmuch  -0.233819338    0.06133805
alcoholtype     -0.713111095    0.09017703
wt71            -0.126732329    -0.07289891
education       -0.324920730    0.41166875
exercise        -0.331992215    0.75294838
income          -0.113335990    0.21705156
race            -0.487496058    1.87006738
sex             -2.271587216    -0.40229651
smokeintensity  -0.006691344    0.06075462
smokeyrs        -0.010489905    0.12116200
price71         -6.674703744    4.34463307
tax71           -4.167850933    7.58276057
```

Crude Regression

It is useful to compare each of these methods we've explored with the crude analysis. This model consists of only the exposure being regressed onto the outcome of interest,

$$\mathbb{E}(y_i) = \beta_0 + \beta_1 qsmk_{i1} + \epsilon_i$$

```r
### crude regression
crd = glm(wt82_71 ~ qsmk, data=merged_df)

### summary and confidence intervals
summary(crd); confint(crd)
```

```
Call:
glm(formula = wt82_71 ~ qsmk, data = merged_df)

Deviance Residuals:
    Min       1Q    Median       3Q      Max
-43.265   -4.023    0.033    4.248   46.554

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.9845     0.2288   8.672  < 2e-16 ***
qsmk          2.5406     0.4511   5.632 2.11e-08 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for gaussian family taken to be 60.89757)

    Null deviance: 97176  on 1565  degrees of freedom
Residual deviance: 95244  on 1564  degrees of freedom
  (63 observations deleted due to missingness)
AIC: 10883

Number of Fisher Scoring iterations: 2

                  Conf. Int
                2.5 %   97.5 %
(Intercept) 1.536002    2.432993
qsmk        1.656481    3.424682
```

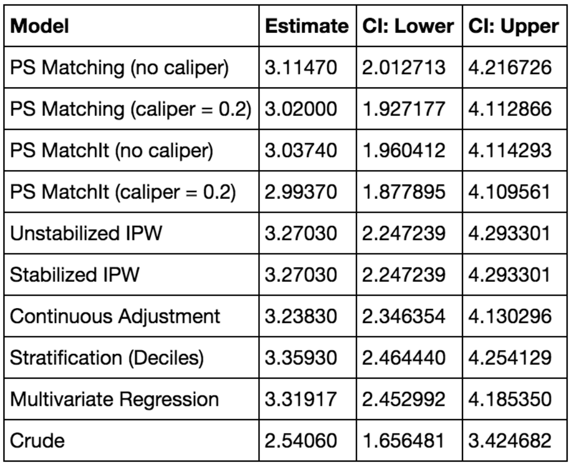
This model will have obvious confounding that is unaccounted for, and we can see that compared to previous models, it under estimates the effect of smoking cessation on the average change in weight.

Comparison

We then compare each of the causal and associative methods we've explored and their resulting estimates of effect (and 95% confidence intervals), followed by a visualization of these model estimates.

```r
### create dataframe with results so we can print out and plot
model = c("PS Matching (no caliper)", "PS Matching (caliper = 0.2)","PS MatchIt (no caliper)", "PS MatchIt (caliper = 0.2)", "Unstabilized IPW", "Stabilized IPW", "Continuous Adjustment", "Stratification (Deciles)", "Multivariate Regression", "Crude")
estimates = c(3.1147, 3.0200, 3.0374, 2.9937, 3.2703, 3.2703, 3.2383, 3.3593, 3.31917, 2.5406)
lower = c(2.0127132, 1.9271773, 1.9604116, 1.8778945, 2.247239, 2.247239, 2.346354, 2.464440, 2.452992087, 1.656481)
upper = c(4.216726, 4.112866, 4.114293, 4.109561, 4.293301, 4.293301, 4.130296, 4.254129, 4.18534959, 3.424682)

### bind the information together that we created
estimates_list = list(model, estimates, lower, upper)
estimates_df = do.call(cbind, estimates_list)

### cast to a dataframe and set column names
estimates_df = data.frame(estimates_df)
names(estimates_df) = c("Model", "Estimate", "CI: Lower", "CI: Upper")

### format the data types as needed
estimates_df$Estimate = as.numeric(as.character(estimates_df$Estimate))
estimates_df$'CI: Lower' = as.numeric(as.character(estimates_df$'CI: Lower'))
estimates_df$'CI: Upper' = as.numeric(as.character(estimates_df$'CI: Upper'))

### printout
estimates_df
```

```r
### create factor with levels
estimates_df$Model = factor(estimates_df$Model, levels=c("PS Matching (no caliper)", "PS Matching (caliper = 0.2)", "PS MatchIt (no caliper)", "PS MatchIt (caliper = 0.2)", "Unstabilized IPW", "Stabilized IPW", "Continuous Adjustment", "Stratification (Deciles)", "Multivariate Regression", "Crude"))

### create plot of the estimates and confidence intervals
estimates_df %>%
    ggplot(aes(x=Estimate, y=Model)) +
    geom_point() +
    geom_errorbarh(mapping=aes(xmin=estimates_df$'CI: Lower', xmax=estimates_df$'CI: Upper'), height=0.2) +
    theme_bw() + theme(axis.text.x = element_text(color="black", size=12),
    axis.text.y = element_text(color="black", size=12), plot.title = element_text(hjust = 0.5)) +
    ylab("") + xlab("\nEstimate of Effect") +
    ggtitle("Estimate of Quitting Smoking and Weight Change\nwith Confidence Intervals by Model\n") + xlim(0,5)
```



Model	Estimate	CI: Lower	CI: Upper
PS Matching (no caliper)	3.11470	2.012713	4.216726
PS Matching (caliper = 0.2)	3.02000	1.927177	4.112866
PS MatchIt (no caliper)	3.03740	1.960412	4.114293
PS MatchIt (caliper = 0.2)	2.99370	1.877895	4.109561
Unstabilized IPW	3.27030	2.247239	4.293301
Stabilized IPW	3.27030	2.247239	4.293301
Continuous Adjustment	3.23830	2.346354	4.130296
Stratification (Deciles)	3.35930	2.464440	4.254129
Multivariate Regression	3.31917	2.452992	4.185350
Crude	2.54060	1.656481	3.424682

Overall, we find a statistically significant estimate of effect between smoking cessation and change of weight; suggesting that on average, smoking cessation causes an increase in weight of approximately 3kg. Each of the models explored suggest a higher impact of smoking cessation on weight change compared to the crude model. The propensity score matching methods have slightly higher variance in their 95% confidence intervals as compared to the rest of the models – with multivariate regression, stratification of the propensity score, and continuous propensity score adjustment having the least variability in the estimate of effect. Regardless, each of the models produce similar results. However further investigation is warranted for the purpose of generalizability as well as the assurance that confounding and/or risk factors have been properly addressed within the study design and analysis phase.

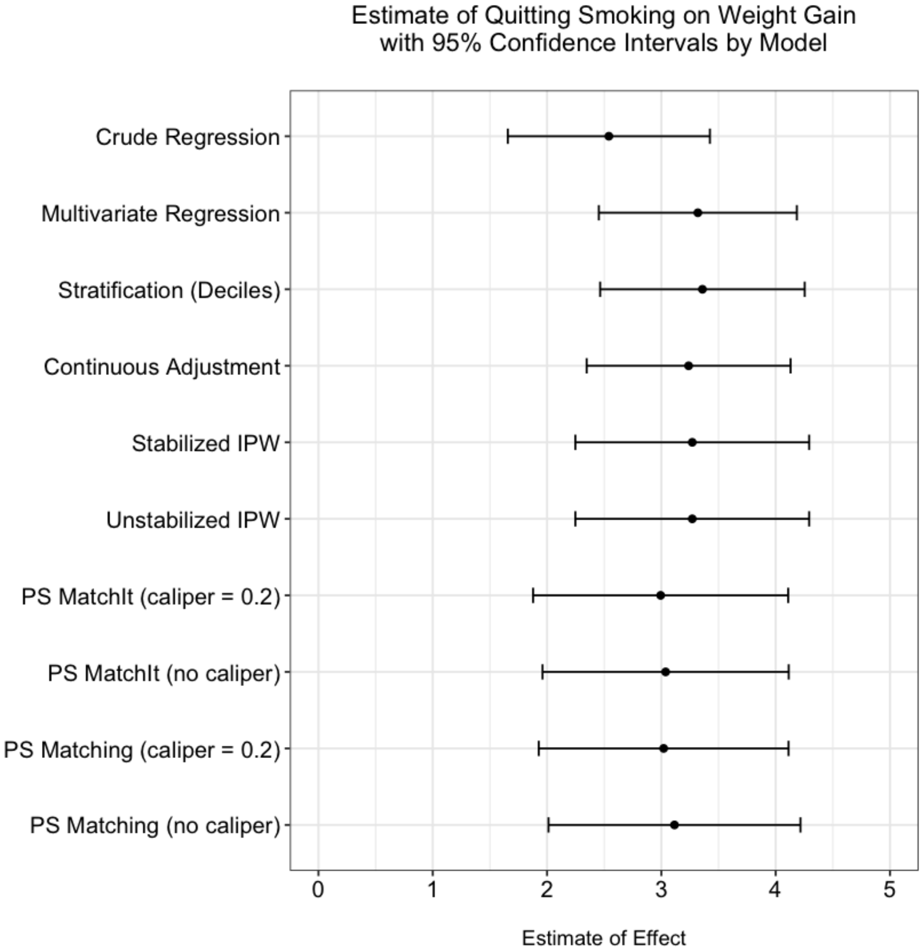


Figure 7: Comparison of the causal model estimates and corresponding confidence intervals.

7. Conclusion

Propensity score methods can help reduce bias in observational data and reduce the likelihood of confounding. By estimating a predicted probability of being exposed (or unexposed), the propensity score can be used to create a counterfactual population (via matching or IPW), to reduce dimensionality (continuous adjustment or stratification), and more.

Overall, propensity scores allow observational data to be conditionally randomized for causal inference – and although propensity score methods are useful in estimating a causal effect, they are not the only method to do so. Other methods such as Instrumental Variable analysis, Difference in Differences, and Marginal Structural Models are particularly useful in causal inference.

To read further on propensity score methods and their uses in practice, please see the references.

References

[1] Hernán MA. A definition of causal effect for epidemiological research Journal of Epidemiology & Community Health. 2004;58:265-271. doi:10.1136/jech.2002.006361.

[2] Hernán MA, Robins JM (2018). Causal Inference. Boca Raton: Chapman & Hall/CRC, forthcoming.

[3] Austin PC. An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies. Multivariate Behavioral Research. 2011;46(3):399-424. doi:10.1080/00273171.2011.568786.

[4] Haukoos JS, Lewis RJ. The Propensity Score. Jama. 2015;314(15):1637-1638. doi:10.1001/jama.2015.13480.

[5] NHANES I Epidemiologic Followup Study (NHEFS). Centers for Disease Control and Prevention. https://wwwn.cdc.gov/nchs/nhanes/nhefs/default.aspx/. Accessed October 21, 2017.