The objective of the matching estimator is to compare the effects of treated and untreated observations that have the same *propensity* for being treated, if treatment is random after conditioning on a set of observables. If for every treated observation, we are able to find a similar observation that was not treated, then we will be able to find an estimate of $\tau(x) = \mathbb{E}(Y_i(1) - Y_i(0)|X_i = x)$. If possible, we could potentially compare the outcomes of two similar individual observations, one with treatment and one without; but there may be a way to make use of more information by compositing the effects of different groups. The method of compositing is the distinguishing feature of the weighted and blocked propensity scores presented in lecture. The choice of method may be a matter of style or it could be driven by sparse data, and the need to interpolate the scores. The objective of this section is to review a few of the alternatives.

First, we should probably create the data that will be used in each of the methods. Let D_i be the indicator of treatment for observation $i = 1, 2, \ldots, N$; let Y_i be the outcome variable; and let X_i be the vector of observable characteristics, which affect the propensity for receiving treatment:

$$Y_i = \delta D_i + \beta X_i + \epsilon_i, \quad \text{with} \;\; \epsilon_i \sim N(0,1) \tag{1}$$

Assume further that there are three observable characteristics $x_1, x_2, x_3 \sim Unif(0,1)$ and that treatment is determined by the following rule:

$$D_i = \begin{cases} 1 & \text{if } 2(x_{1i} + x_{2i} + x_{3i}) + u_i > 2 \\ 0 & \text{otherwise} \end{cases}$$

where $u_i \sim N(0,1)$. Note that if we run a linear regression without conditioning on X_i, the treatment effect will be biased, since the composite error term will be correlated with both treatment and outcome. With this framework, we can construct a data set of size $N = 5000$ in order to examine the behavior of various estimation techniques.

```
N <- 5000; eps <- rnorm(N); u <- rnorm(N)
x1 <- runif(N); x2 <- runif(N); x3 <- runif(N)
D <- ifelse(2*(x1 + x2 + x3) + u > 4, 1, 0)
Y <- D + x1 + x2 + x3 + eps
summary(D); summary(Y)
```

```
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.0000  0.0000  0.0000  0.2442  0.0000  1.0000
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -2.2320  0.8454  1.7050  1.7570  2.6150  6.4010
```

Roughly one quarter of the observations received treatment, and the outcome variable has about a ten unit spread, centered around 1.5 or 2. (This is subject to some uncertainty. Each time this document is compiled to LaTeX, the R code is run again.)

Ordinary least squares

For reference, we estimate to basic, linear models by ordinary least squares. First, we do not condition on the X covariates, which will yield biased estimates of the treatment effect — which is known. We bootstrap the distribution of the estimated treatment effect. We sample $n = 500$ observations from the distribution, estimate the impact effect, and repeat for $B = 5000$ iterations. Note that we do not iterate using a `for` loop, but rather by applying the `ols` function, defined below, to a range of indices using `sapply` to keep the code compact and readable.

```
n <- 500; B <- 5000
X <- cbind(1, D)
```

```
ols <- function(i) {
  idx <- sample.int(N,n)
  Xs <- X[idx,]
  b <- solve(t(Xs) %*% Xs) %*% t(Xs) %*% Y[idx]
  b[2]
}

res.ols <- data.frame(impact=sapply(1:B, ols), method=c("ols"))
```

Before we graph the distribution, let's perform the same process for the estimated impact, conditioning on X. This should yield a consistent estimator for the treatment effect δ, since by construction there is no three-way covariation between the error, outcome, *and* treatment, after conditioning on the observables.

```
X.ext <- cbind(1, D, x1, x2, x3)

mult.ols <- function(i) {
  idx <- sample.int(N,n)
  Xs <- X.ext[idx,]
  b <- solve(t(Xs) %*% Xs) %*% t(Xs) %*% Y[idx]
  b[2]
}

res.mult <- data.frame(impact=sapply(1:B, mult.ols), method=c("mult.ols"))
total.res <- rbind(res.ols, res.mult)
```

Now we can plot the two distributions of impact estimates, based on the method of estimation. The vertical line in Figure 1 indicates the true, known impact effect. It is clear that the OLS estimates with omitted variables overstate the treatment effect, since there is selection into the treatment group.

```
library(ggplot2)
p <- ggplot(total.res, aes(x=impact, colour=method)) + geom_density()
p + geom_vline(xintercept = 1)
```

Conditioning on the propensity score

We have shown in lecture that D_i is independent of $Y_i(\cdot)$, after conditioning on $p(X_i)$. We should therefore get a consistent estimate for the treatment effect, after conditioning on the propensity score. The model we wish to estimate, then, is given as $y_i = \alpha + \delta D_i + \gamma \hat{p}_i + \epsilon_i$:

```
data <- data.frame(cbind(Y, D, x1, x2, x3))

psm <- function(i) {
  idx <- sample.int(N,n)
  d <- data[idx,]
  logit <- glm(D ~ x1 + x2 + x3, data = d, family = "binomial")
  d$p <- logit$fitted.values
  ols <- lm(Y ~ D + p, data = d)
  ols$coefficients[["D"]]
}

res.psm <- data.frame(impact=sapply(1:B, psm), method=c("psm"))
```

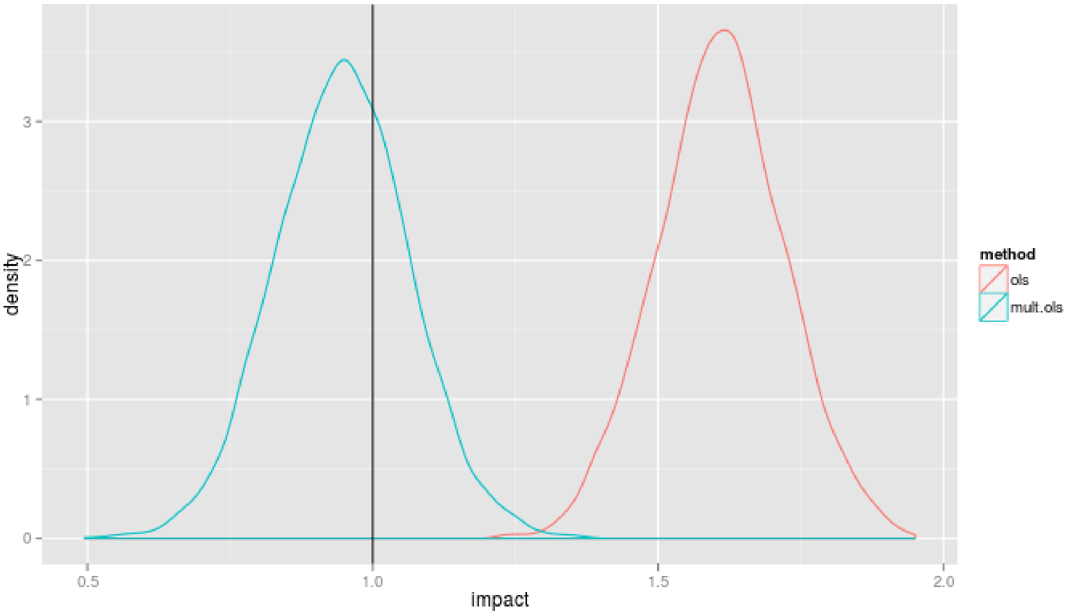

Figure 1: Estimated impacts based on OLS regression

Indeed, the distribution seems to be centered around the true effect, suggesting that the estimator is consistent. The problem with this method, however, is that it is computationally intensive, relative to the simple linear regression. Any two step estimator — especially one with a nonlinear model — will take much longer, without much gain in this case, since we know that multiple regression by OLS will yield a *best* linear unbiased estimator.

Propensity score weighting