Visualizing Selection on Observables

Luke Sonnet

May 4, 2016

What is selection on observables? Technically it is

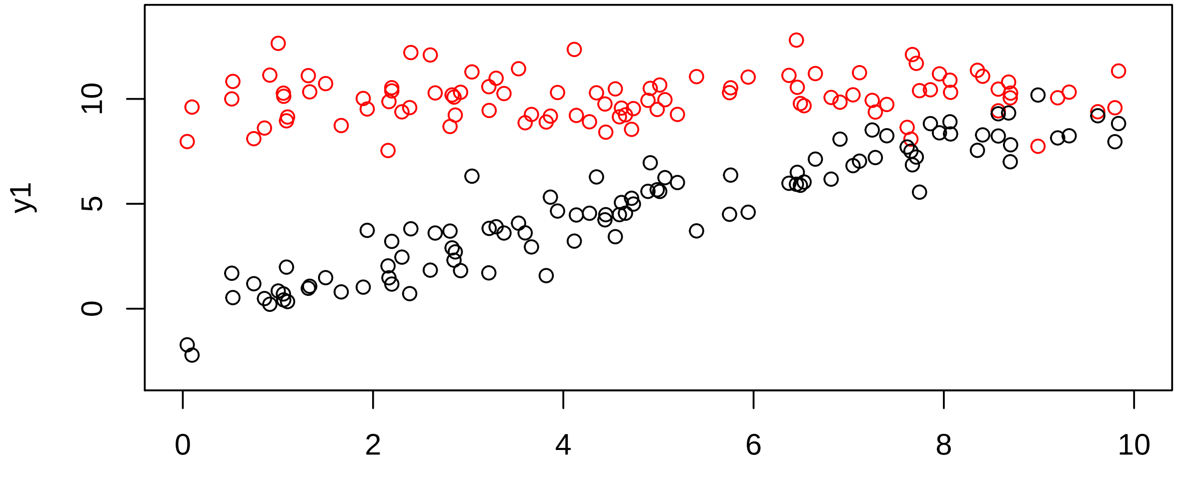
$$\{Y_{0i}, Y_{1i}\} \perp\!\!\!\perp D_i | X_i = x$$

This means that for some value of X_i, which we denote as x, we have a randomized experiment.

Let's look at this via a simulted example. Let's pretend that we can see all of the potential outcomes:

```r
## Create data and some "invisible variable" x
n <- 100
x <- runif(n, 0, 10)

## Create the potential outcomes
y1 <- rnorm(n, mean = 10, sd = 1)
y0 <- rnorm(n, mean = x, sd = 1)
## Plot the potential outcomes
plot(x, y1,
     col = "red", xlab = "",
     xlim = c(0, 10), ylim = c(min(y0) - 1, max(c(y0, y1)) + 1))
points(x, y0)
```



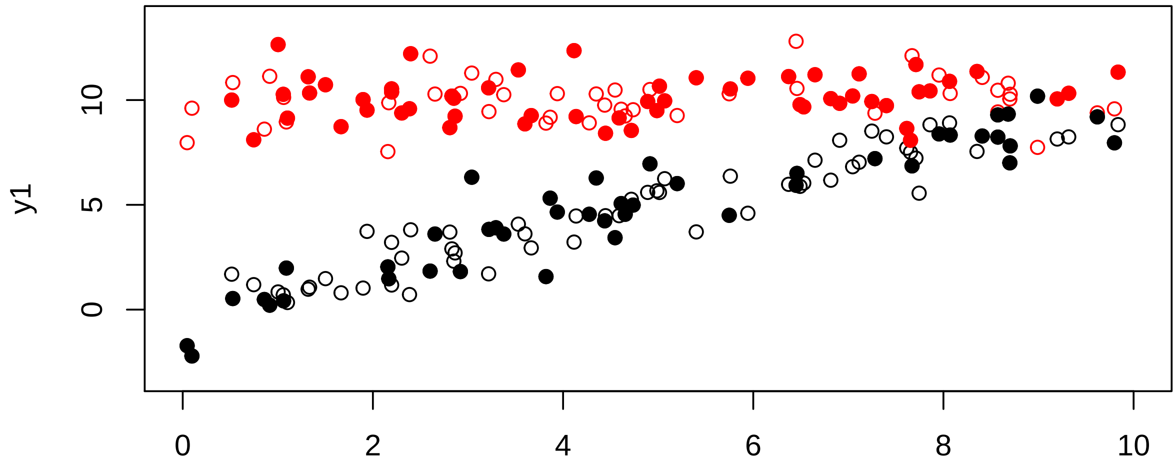
```
## What is the ATE? Not an estimate, the actual ATE
ate <- mean(y1 - y0)
ate
```

```
## [1] 5.401726
```

```
## Can we get the ATT?
```

Random experiment

```
## Treatment vector
d <- rbinom(n, 1, 0.5)
## Plot the "observed" values
plot(x, y1,
     col = "red", xlab = "",
     xlim = c(0, 10), ylim = c(min(y0) - 1, max(c(y0, y1)) + 1))
points(x, y0)
points(x[d==1], y1[d==1], pch = 19, col = "red")
points(x[d==0], y0[d==0], pch = 19)
```



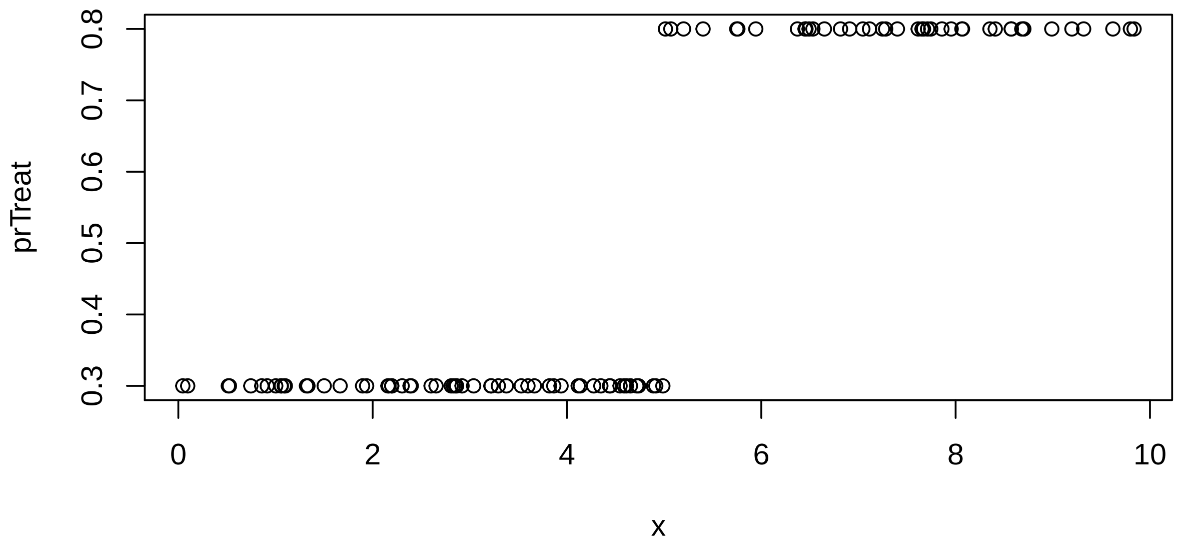
```
## Get our observed Y
y <- ifelse(d, y1, y0)

## What is the difference in means estimator?
dim <- mean(y[d==1]) - mean(y[d==0])
c("ATE" = ate, "DIM" = dim)
```

```
##      ATE      DIM
## 5.401726 5.343321
```

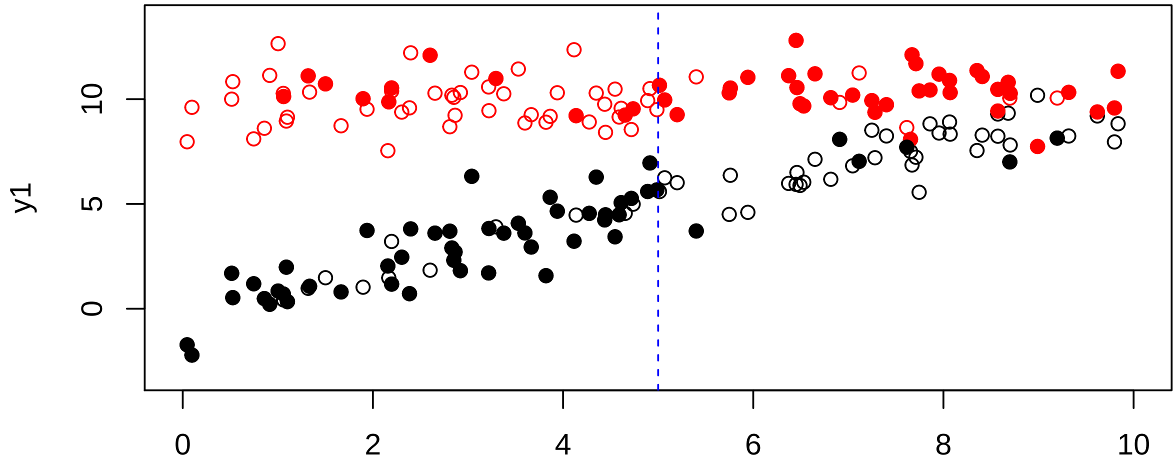
Not bad, but what if we violate the randomization, and instead say that people with higher values of some invisible index are more likely to receive treatment?

```
## Higher probability of treatment for x > 5
prTreat <- ifelse(x > 5, 0.8, 0.3)
plot(x, prTreat)
```



```
d <- rbinom(n, 1, pr = prTreat)

## Plot the potential outcomes
plot(x, y1,
     col = "red", xlab = "",
     xlim = c(0, 10), ylim = c(min(y0) - 1, max(c(y0, y1)) + 1))
points(x, y0)
## Plot the "observed" values
points(x[d==1], y1[d==1], pch = 19, col = "red")
points(x[d==0], y0[d==0], pch = 19)
abline(v = 5, lty = 2, col = "blue")
```



```
## Get our observed Y
y <- ifelse(d, y1, y0)

## What is the difference in means estimator?
dim <- mean(y[d==1]) - mean(y[d==0])
c("ATE" = ate, "DIM" = dim)
```

```
##      ATE      DIM
## 5.401726 7.042836
```

It is biased upwards! Because we are more likely to see the values of y_{0i} when they are lower, it looks like the treatment effect is too large.

Selection on Observables

Let's say the index x is now observed. We know that treatment is as if random when $x > 5$ and when $x \leq 5$. Then we know that

$$\{Y_{0i}, Y_{1i}\} \perp\!\!\!\perp D_i | X_i = x$$

is true; of course in reality you wouldn't know the true data generating process, but you would have to guess and defend it.

Now, we can use the estimators Chad talked about in the SOO section. For now, let's use the subclassification estimator,

$$\sum_{j=1}^{M} \{\overline{Y_{1j}} - \overline{Y_{0j}}\} \frac{n_j}{n}$$

where $\overline{Y_{dj}}$ is the average outcome for observations in group j with treatment status d, and n_j is the number of observations in group j.

```
dim_subclass <- (mean(y[d==1 & x > 5]) - mean(y[d==0 & x > 5])) * (sum(x > 5) / n) +
  (mean(y[d==1 & x <= 5]) - mean(y[d==0 & x <= 5])) * (sum(x <= 5) / n)
c("ATE" = ate, "DIM Subclass" = dim_subclass)
```

```
##          ATE DIM Subclass
##     5.401726     5.767959
```

Pretty good! So conditioning on X allows us to recover the ATE because selection on observables holds here. Let's see that explicitly. Note, we can only do this because we are simulating an example where we see all potential outcomes. Unconditionally, is there a relationship between the potential outcomes and the treatment?

```
cor(y0, d)
```

```
## [1] 0.5139241
```

```
cor(y1, d)
```

```
## [1] 0.2427994
```

It looks like `y0` is not independent of d. What about if we look at conditional independence?

```
cor(y0[x > 5], d[x > 5])
```

```
## [1] 0.09738466
```

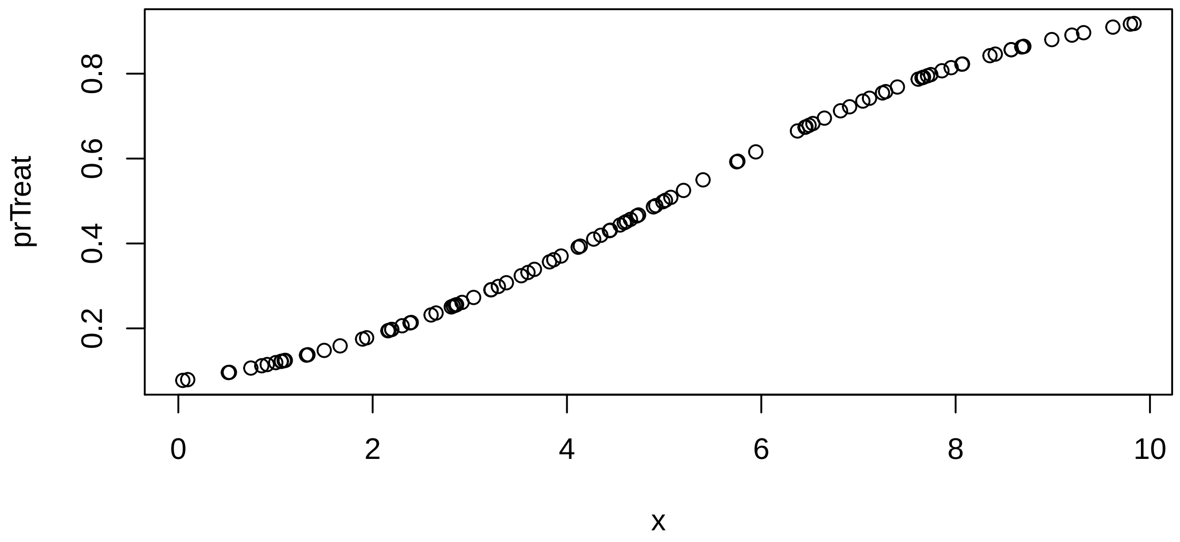
```
cor(y0[x <= 5], d[x <= 5])
```

```
## [1] -0.05363963
```

Of course, we know they are uncorrelated because the probability of treatment is constant in those subgroups.

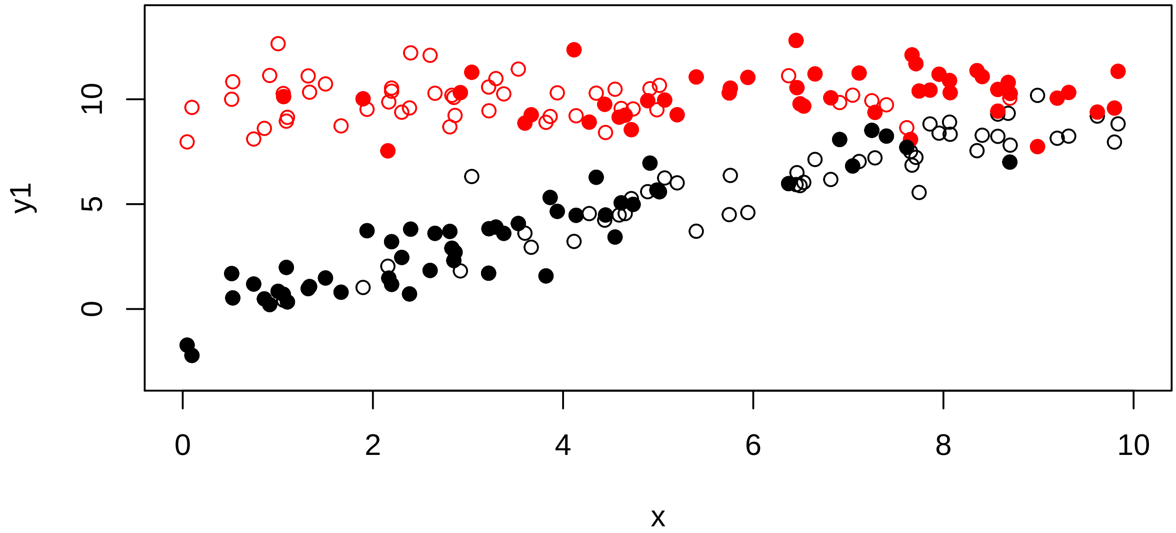
What if our probability of treatment directly corresponded to our x variable? For example:

```
## Higher probability of treatment as x increases
prTreat <- 1 / (1 + exp(-(-2.5 + x/2)))
plot(x, prTreat)
```



```r
d <- rbinom(n, 1, pr = prTreat)

## Plot the potential outcomes
plot(x, y1,
     col = "red",
     xlim = c(0, 10), ylim = c(min(y0) - 1, max(c(y0, y1)) + 1))
points(x, y0)
## Plot the "observed" values
points(x[d==1], y1[d==1], pch = 19, col = "red")
points(x[d==0], y0[d==0], pch = 19)
```



```
## Get our observed Y
y <- ifelse(d, y1, y0)

## What is the difference in means estimator?
dim <- mean(y[d==1]) - mean(y[d==0])
c("ATE" = ate, "DIM" = dim)
```

```
##      ATE      DIM
## 5.401726 6.922247
```

Also biased upwards. Here is a case where we could use inverse propensity score weighting. But instead of using the true probability of treatment, we have to estimate it.

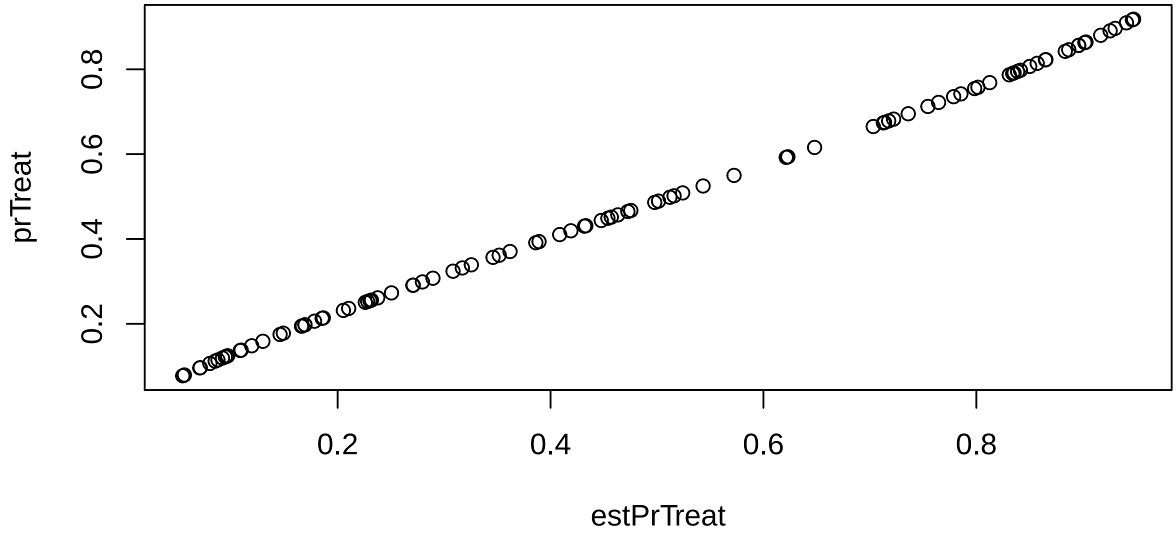
```
## Logit to estimate treatment model (will be correct model in this case)
treatModel <- glm(d ~ x, family = binomial)
## Did it recover the true treatment assignemnt model?
cbind(treatModel$coefficients, c(-2.5, 0.5))
```

```
##                   [,1] [,2]
## (Intercept) -2.8813393 -2.5
## x            0.5875268  0.5
```

```
estPrTreat <- predict(treatModel, type = "response")
```

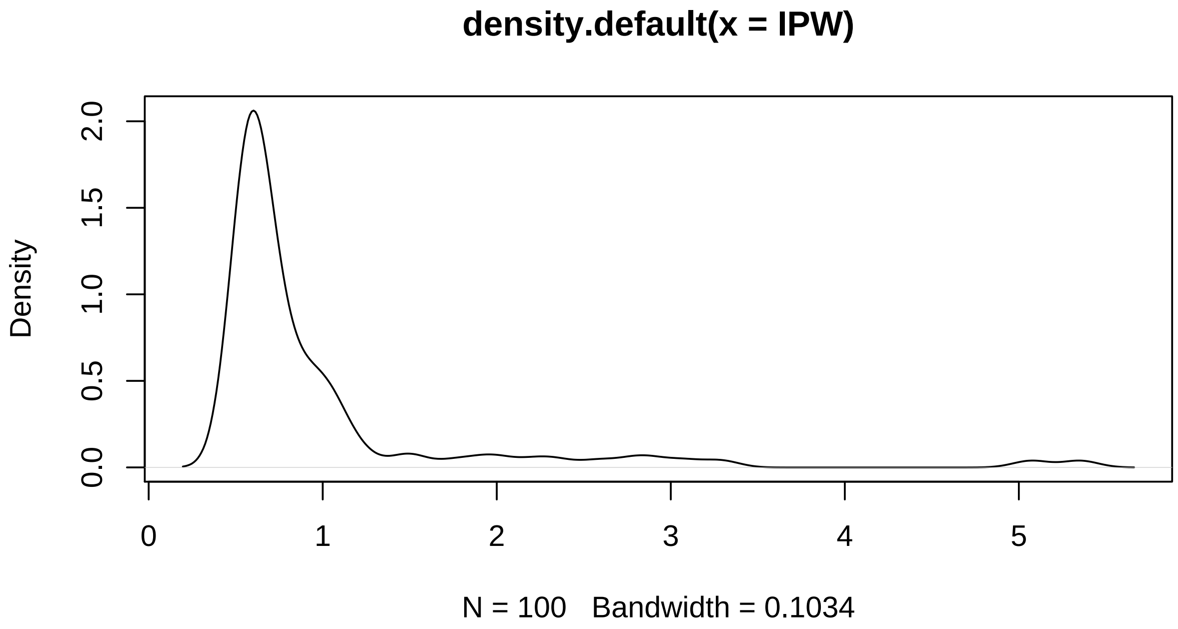
Now we can see how well our model for estimating the propensity scores did:

```
plot(estPrTreat, prTreat)
```



It looks like we fit the model pretty well! Now let's do our weighting:

```
prD <- mean(d)
IPW <- (d * prD + (1 - d) * (1 - prD)) /
  (d * estPrTreat + (1 - d) * (1 - estPrTreat))
plot(density(IPW))
```



There are some extreme examples, which might not be desirable. Nonetheless, let's use weighting and compute our difference in means estimator:

```
lmAte <- lm(y ~ d, weight = IPW)
dim_ipw <- lmAte$coefficients[2]
c("ATE" = ate, "DIM IPW Weight" = dim_ipw)
```

```
##              ATE DIM IPW Weight.d
##         5.401726         5.294993
```

Much better!

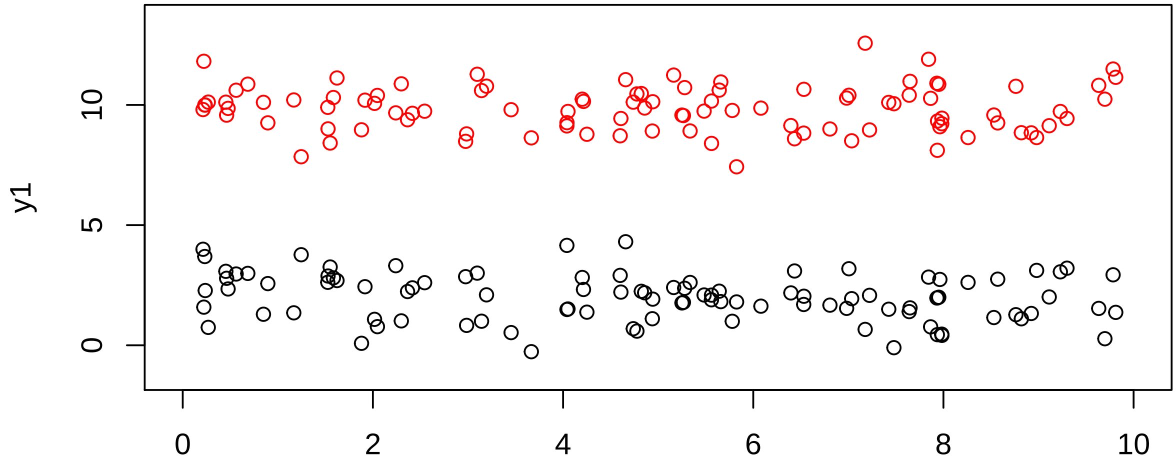
Non-constant Probability of Treatment

Now, is it the fact that there was a non-constant probability of treatment that was a problem or was it the fact that there was non-constant probabilit of treatment that was related to the potential outcomes? This is a subtle point.

Let's look at the following example:

```
## Create data and some "invisible variable" x
n <- 100
x <- runif(n, 0, 10)

## Create the potential outcomes
y1 <- rnorm(n, mean = 10, sd = 1)
y0 <- rnorm(n, mean = 2, sd = 1)
## Plot the potential outcomes
plot(x, y1,
     col = "red", xlab = "",
     xlim = c(0, 10), ylim = c(min(y0 - 1), max(y1 + 1)))
points(x, y0)
```

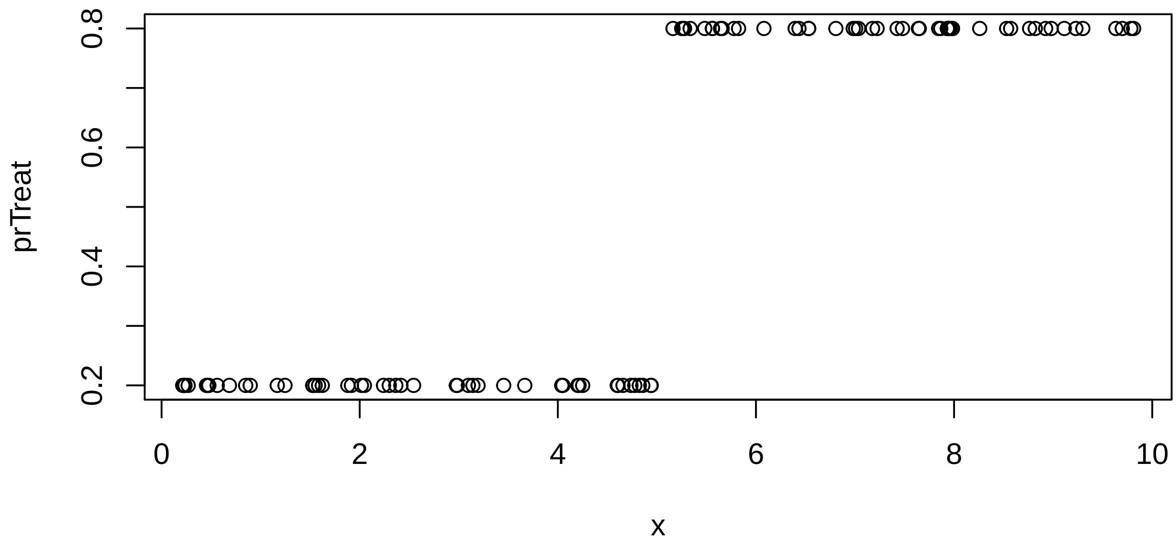


```
## What is the ATE? Not an estimate, the actual ATE
ate <- mean(y1 - y0)
ate
```

```
## [1] 7.851311
```

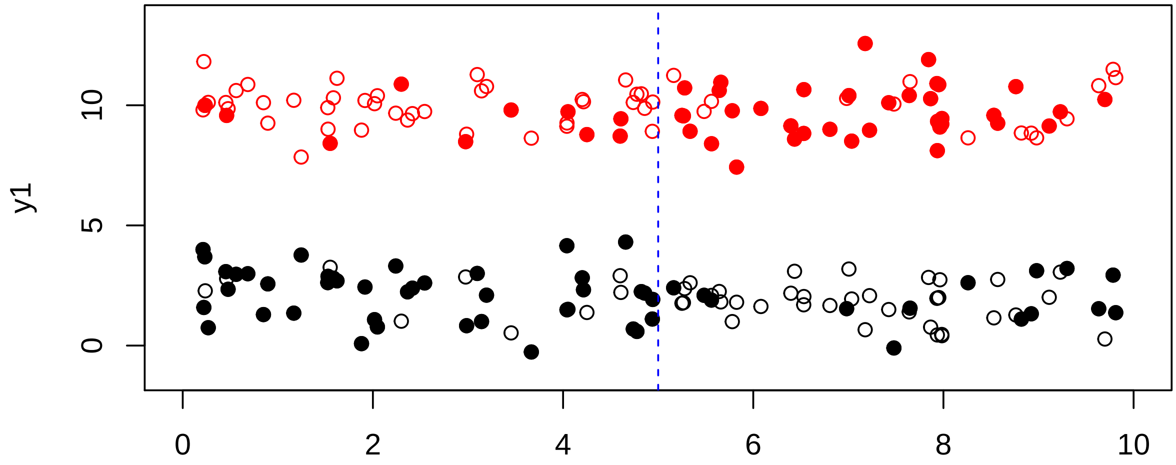
Now let’s again have non-constant treatment:

```
## Higher probability of treatment for x > 5
prTreat <- ifelse(x > 5, 0.8, 0.2)
plot(x, prTreat)
```



```r
d <- rbinom(n, 1, pr = prTreat)

## Plot the potential outcomes
plot(x, y1,
     col = "red", xlab = "",
     xlim = c(0, 10), ylim = c(min(y0) - 1, max(y1) + 1))
points(x, y0)
## Plot the "observed" values
points(x[d==1], y1[d==1], pch = 19, col = "red")
points(x[d==0], y0[d==0], pch = 19)
abline(v = 5, lty = 2, col = "blue")
```



```
## Get our observed Y
y <- ifelse(d, y1, y0)

## What is the difference in means estimator?
dim <- mean(y[d==1]) - mean(y[d==0])
c("ATE" = ate, "DIM" = dim)
```

```
##      ATE      DIM
## 7.851311 7.567197
```

All good! Again, even though there is non-constant probability of treatment, that does not correlate with our potential outcomes. If it did, and we observed x, we could use SOO to estimate our ATE.