

Propensity Score Diagnostics: Extended Example

Ashley I Naimi

Spring 2024

Simulating Data

Let's gain some intuition behind the idea of the propensity score, and whether it can tell us anything about exchangeability.

We'll start with some simple simulated data, so that we know the true relations, effects, and confounders:

```
pacman::p_load(
  here,
  tidyverse,
  mvtnorm,
  spatstat,
  gridExtra
)

thm <- theme_classic() +
  theme(
    legend.position = "top",
    legend.background = element_rect(fill = "transparent", colour = NA),
    legend.key = element_rect(fill = "transparent", colour = NA)
  )
theme_set(thm)

expit <- function(x){1/(1+exp(-x))}
sample_size <- 100000

set.seed(234)

c <- mvtnorm::rmvnorm(n = sample_size, mean = rep(0, 4), sigma = diag(4))

# true ps model
pi_x <- expit(-2 + log(2)*c[,1] + log(2)*c[,2] + log(2)*c[,3] + log(4)*c[,4])

x <- rbinom(n = sample_size, size = 1, prob = pi_x)

mean(x)

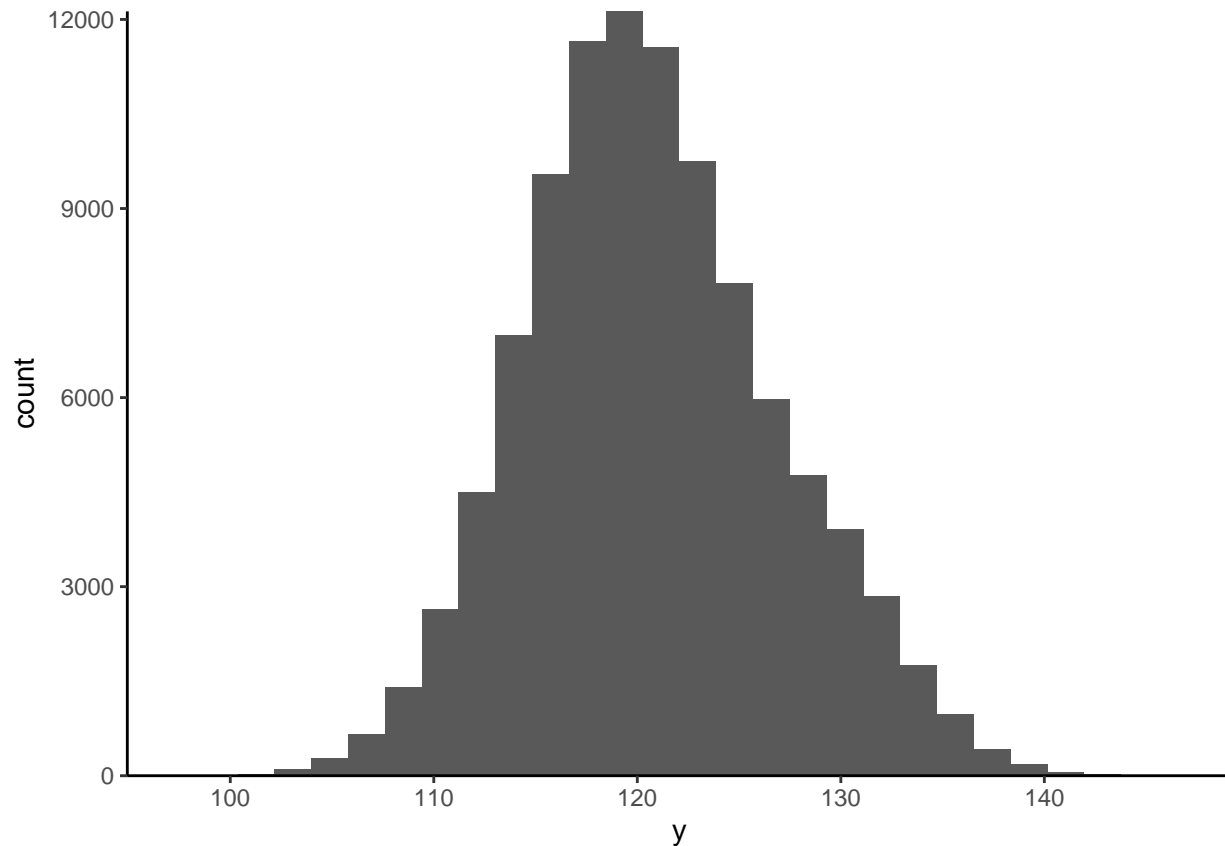
## [1] 0.21205

# true outcome model
y <- rnorm(n = sample_size, mean = 120 + 3.5*x + 2*c[,1] + 2*c[,2] + 2*c[,3] + 4*c[,4], sd = 1)

ggplot() +
  geom_histogram(aes(y)) +
```

```
scale_x_continuous(expand = c(0,0)) +
scale_y_continuous(expand = c(0,0))
```

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



```
## the observed data: NOTE WE'RE MISSING THE STRONGEST CONFOUNDER!!
```

```
dat_ <- data.frame(y, x, c[,1:3])
```

```
names(dat_) <- c("y", "x", paste0("c", 1:3))
```

```
head(dat_)
```

```
##      y x      c1      c2      c3
## 1 119.0790 0  0.66076974 -2.05298300 -1.49920605
## 2 112.3608 0  1.45913853  0.14013904  0.20918439
## 3 120.9611 0 -0.48693413 -1.08786731  0.05785971
## 4 125.1544 0 -0.02561697  0.51484639  0.99005668
## 5 120.1924 0 -0.93007223  0.08403068  0.52677963
## 6 120.3508 0  0.20535184  1.01628335  0.40899897
```

Propensity Score Fitting

Let's now use these data to fit a propensity score model:

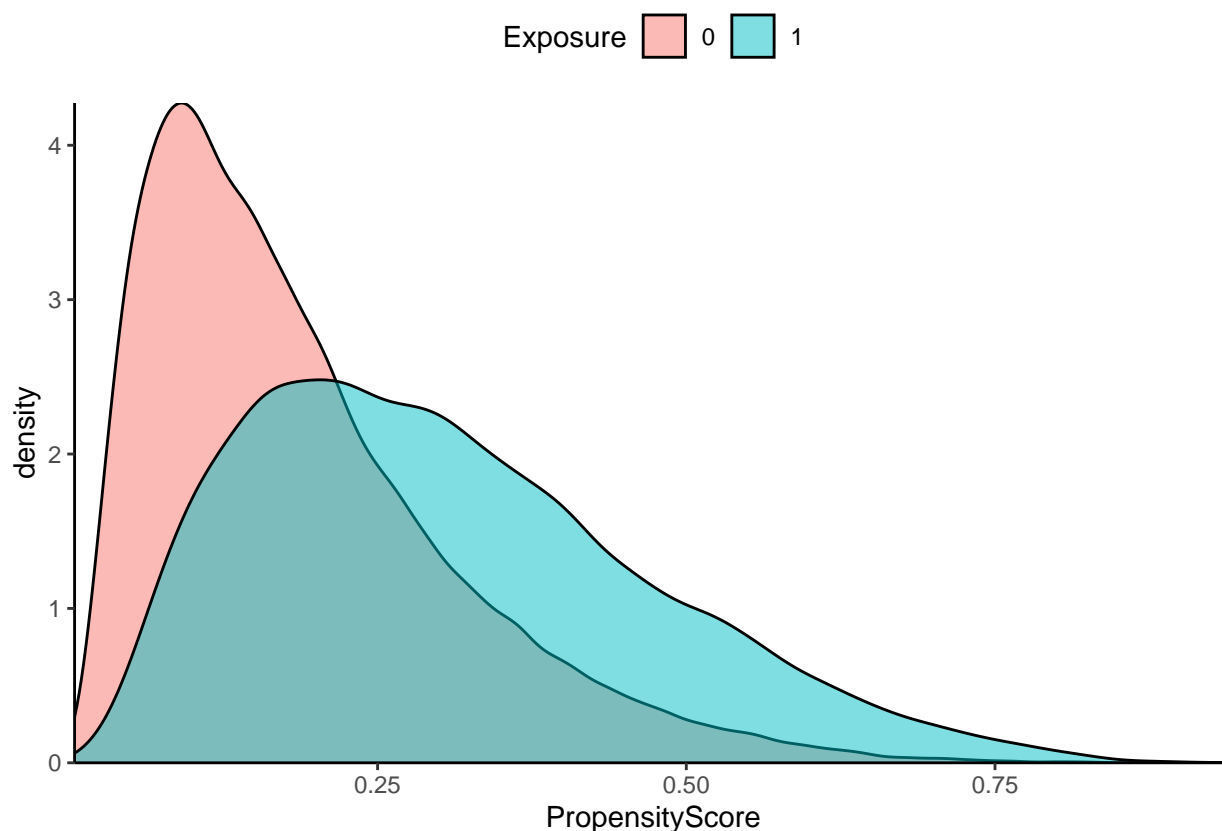
```
propensity_score <- glm(x ~ c1 + c2 + c3, data = dat_,
                        family = binomial("logit"))$fitted.values
```

Propensity Score Overlap

```
# ps overlap plot

plot_dat <- data.frame(Exposure = factor(dat_$x), PropensityScore = propensity_score)

ggplot(plot_dat) +
  geom_density(aes(x = PropensityScore, group = Exposure, fill = Exposure), alpha = .5) +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0))
```



PS overlap is not terrible. Let's look at stabilized weights:

Stabilized Weight Distribution

```
sw <- (mean(dat_$x)/propensity_score)*dat_$x +
  ((1 - mean(dat_$x))/(1 - propensity_score))*(1 - dat_$x)

summary(sw)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.2251  0.8487   0.9235  0.9992  1.0608 16.6896
```

Distribution of stabilized weights look good!

Covariate “Balance” Across Exposure (via logit regression)

Let's try even more diagnostics of the propensity score to see if we've achieved “balance”:

We can look at odds ratios for each confounder-exposure association:

```
# pre - balance

summary(glm(x ~ c1 + c2 + c3, data = dat_, family = binomial("logit")))$coefficients

##              Estimate Std. Error    z value Pr(>|z|)
## (Intercept) -1.5329944 0.009187248 -166.86111    0
## c1           0.5275083 0.008662548  60.89528    0
## c2           0.5277242 0.008626033  61.17809    0
## c3           0.5546114 0.008705674  63.70689    0

# note the difference
```

Now let's get weighted odds ratios for each confounder-exposure association:

```
# post - balance

dat_$sw <- sw

summary(glm(x ~ c1 + c2 + c3, data = dat_, weights = sw, family = binomial("logit")))$coefficients

##              Estimate Std. Error    z value Pr(>|z|)
## (Intercept) -1.318800247 0.007753512 -170.0906899 0.0000000
## c1           0.008074501 0.007788143   1.0367685 0.2998438
## c2           0.008597049 0.007767283   1.1068284 0.2683681
## c3           0.006879315 0.007768947   0.8854888 0.3758931

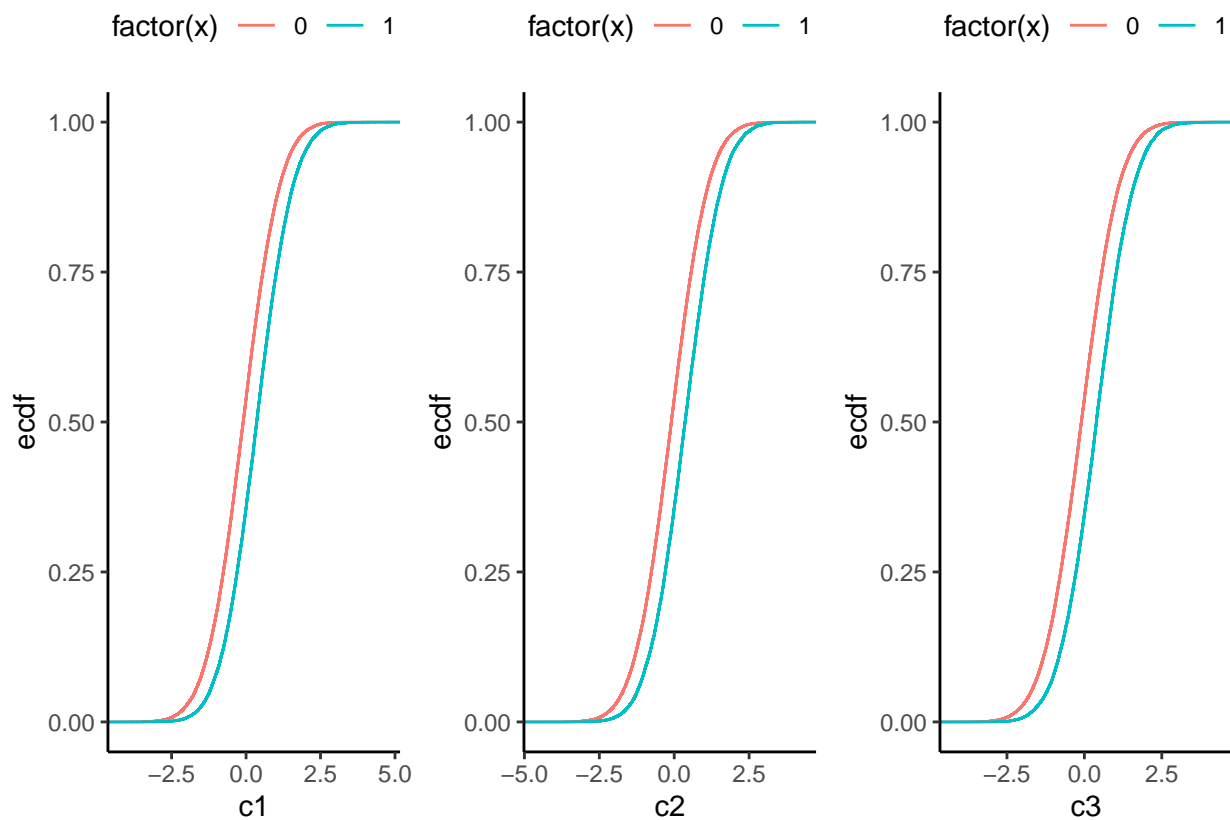
# note the similarity!
```

Unweighted and Weighted eCDFs

Let's look at the cumulative distribution functions for each confounder stratified by the exposure:

```
p1 <- ggplot(dat_, aes(c1, colour = factor(x))) + stat_ecdf()
p2 <- ggplot(dat_, aes(c2, colour = factor(x))) + stat_ecdf()
p3 <- ggplot(dat_, aes(c3, colour = factor(x))) + stat_ecdf()

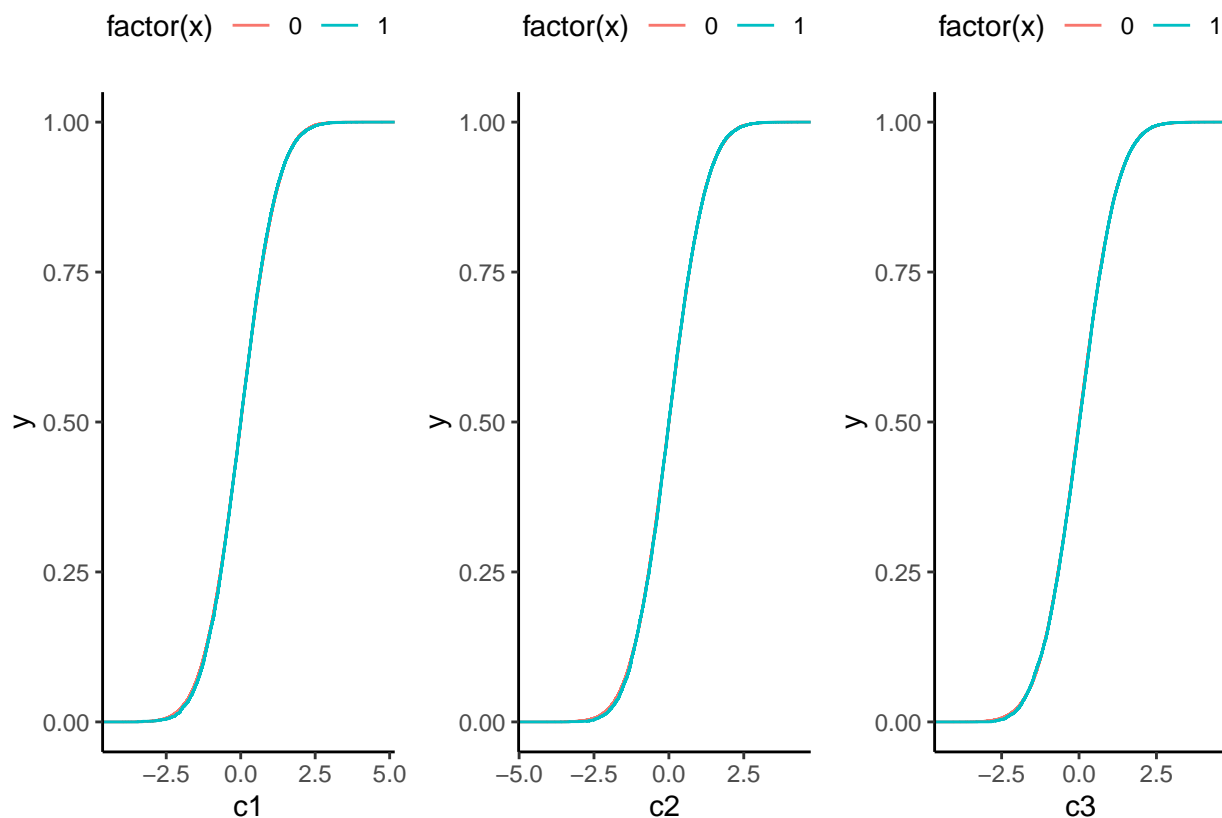
grid.arrange(p1,p2,p3, nrow = 1)
```



Now let's look at the same CDFs for each confounder, but weighted by sw:

```
## note: we need to modify the stat_ecdf function to include weights.
## https://github.com/NicolasWoloszko/stat_ecdf_weighted
p1 <- ggplot(dat_, aes(c1, colour = factor(x), weight = sw)) + stat_ecdf()
p2 <- ggplot(dat_, aes(c2, colour = factor(x), weight = sw)) + stat_ecdf()
p3 <- ggplot(dat_, aes(c3, colour = factor(x), weight = sw)) + stat_ecdf()

grid.arrange(p1,p2,p3, nrow = 1)
```



Despite Balance, Still Bias

Clearly, all these metrics suggest to us that there is good covariate balance across the exposed and unexposed group when we weight the analysis. This is true, even though we've failed to include an important confounder in our propensity score model.

What happens when we conduct a weighted analysis using this mis-specified propensity score model?:

```
mod_sw <- lm(y ~ x, weights = sw, data = dat_)

summary(mod_sw)$coefficients # let's ignore SE estimation
```

```
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 119.08861 0.01678295 7095.8075     0
## x           8.01321 0.03653354  219.3384     0
```

Note that in the simulation code above, we defined the true effect of X on Y to be 3.5. But our weighted analysis suggests this effect is actually 8.01, which is nowhere near the true effect. This is entirely because we left out C_4 from our propensity score model, which was defined as a strong confounder in the simulation.

This example should provide you with some insight as to why propensity score overlap, balance, and other metrics defined on the basis of the propensity score say nothing about exchangeability.

Correctly Specified PS Model

To confirm that the problem is, in fact, due to the exclusion of C_4 , let's fit a weighted outcome model with the correct PS:

```

dat_ <- data.frame(y, x, c[,1:4])

names(dat_) <- c("y", "x", paste0("c", 1:4))

head(dat_)

##           y x          c1          c2          c3          c4
## 1 119.0790 0  0.66076974 -2.05298300 -1.49920605  1.47123312
## 2 112.3608 0  1.45913853  0.14013904  0.20918439 -3.03608982
## 3 120.9611 0 -0.48693413 -1.08786731  0.05785971  1.10397550
## 4 125.1544 0 -0.02561697  0.51484639  0.99005668  0.30345432
## 5 120.1924 0 -0.93007223  0.08403068  0.52677963  0.01586862
## 6 120.3508 0  0.20535184  1.01628335  0.40899897 -0.70523547

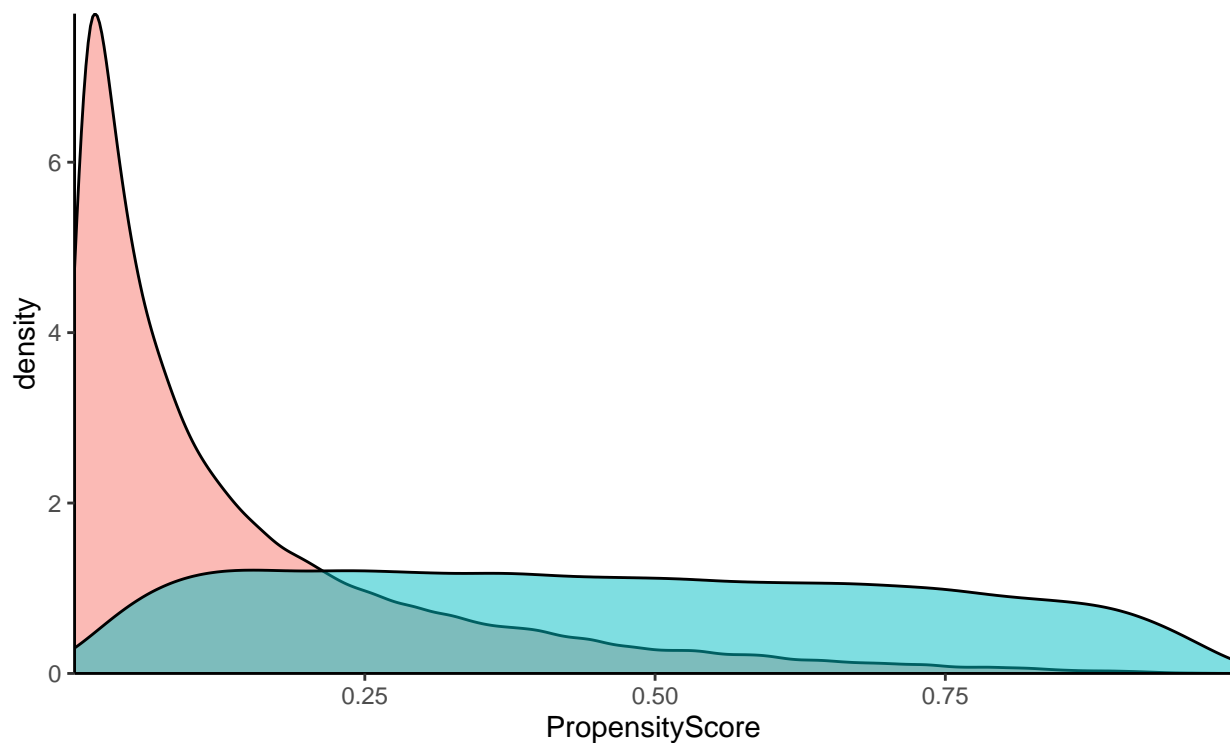
propensity_score <- glm(x ~ c1 + c2 + c3 + c4, data = dat_,
                        family = binomial("logit"))$fitted.values

# ps overlap plot
plot_dat <- data.frame(Exposure = factor(dat_$x), PropensityScore = propensity_score)

ggplot(plot_dat) +
  geom_density(aes(x = PropensityScore, group = Exposure, fill = Exposure), alpha = .5) +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0))

```

Exposure ■ 0 ■ 1



```

dat_$sw <- (mean(dat_$x)/propensity_score)*dat_$x +
  ((1 - mean(dat_$x))/(1 - propensity_score))*(1 - dat_$x)

```

```
summary(dat_$sw)
```

```
##      Min.  1st Qu.   Median     Mean  3rd Qu.    Max.
##    0.2127  0.7979   0.8409   1.0004   0.9781 193.3716
```

```
# weighted outcome model:
```

```
mod_sw <- lm(y ~ x, weights = sw, data = dat_)
```

```
summary(mod_sw)$coefficients # let's ignore SE estimation
```

```
##              Estimate Std. Error    t value Pr(>|t|)
## (Intercept) 119.993205 0.01916389 6261.42116      0
## x           3.479034 0.04159545  83.63978      0
```