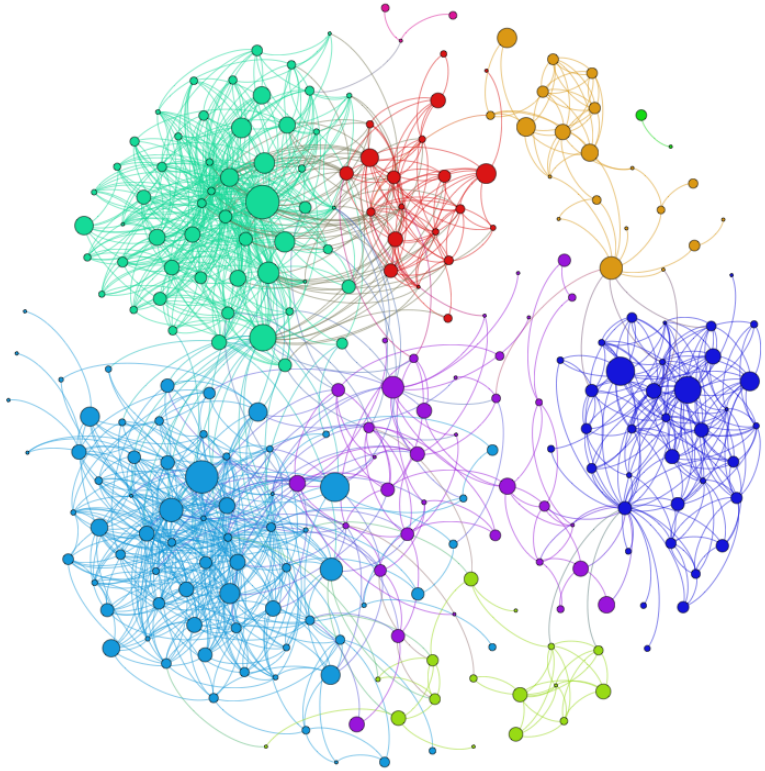
Causal Effect Estimation under Unconfoundedness

Jiaming Mao

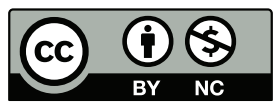
Xiamen University



This version: Spring 2021

Contact: jmao@xmu.edu.cn

Course homepage: jiamingmao.github.io/data-analysis



Causal Effect Estimation Under Unconfoundedness

Suppose our goal is to learn the causal effect of a treatment variable w on an outcome variable y, i.e., $\mathbb{E}[y|\text{do}(w)]$. If there are no *open* back-door paths from w to y[1], then $\mathbb{E}[y|\text{do}(w)] = \mathbb{E}[y|w]$.

[1] according to our causal model $\mathcal{M}$.

Causal Effect Estimation Under Unconfoundedness

When there are open back-door paths from w to y, according to the back-door criterion, if we observe a set of *pre-treatment*[2] variables x such that conditioning on x blocks these paths, then $\mathbb{E}[y|\text{do}(w)]$ is nonparametrically identified:

$$\mathbb{E}[y|\text{do}(w)] = \int \mathbb{E}[y|w,x]\, p(x)\, dx \tag{1}$$

In this case, we say conditional on x, the treatment assignment mechanism is ignorable and w is exogenous to y.

Causal effect estimation under sufficient control for confounding is called *causal effect estimation under unconfoundedness*.

[2] i.e., no variable in x is a descendant of w.

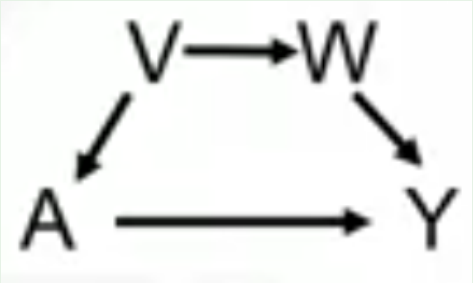
Controlling for Confounding

The back-door criterion shows that we do not need to observe and condition on all confounders, but only a *(minimally) sufficient* set of variables that renders all back-door paths blocked[3].

[3] When all common causes are fully observed, we say there is no unmeasured confounding, or there is selection on observables. As the back-door criterion makes clear, this is sufficient but not necessary for causal effect identification.

Controlling for Confounding

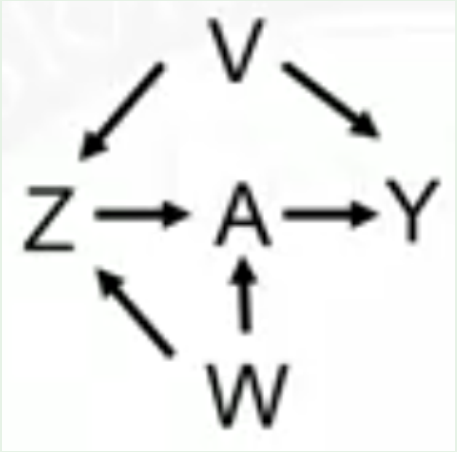
Example 1



- Treatment: A; Outcome: Y
- Sets of variables sufficient to control for confounding: $\{V\}$, $\{W\}$, $\{V, W\}$

Controlling for Confounding

Example 2



- Treatment: A; Outcome: Y
- There is an open back-door path: $A \leftarrow Z \leftarrow V \rightarrow Y$[a].
- Sets of variables sufficient to control for confounding: $\{V\}$, $\{Z, W\}$, $\{Z, V\}$, $\{Z, V, W\}$[b].

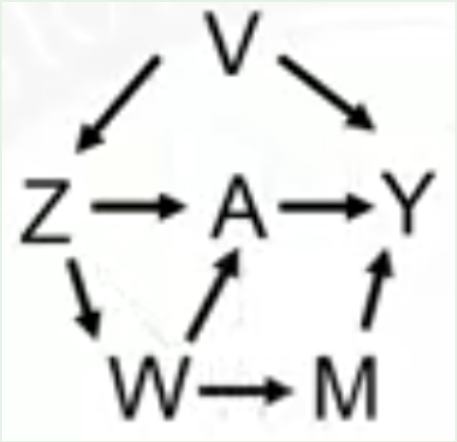
---

[a] The other back-door path $A \leftarrow W \rightarrow Z \leftarrow V \rightarrow Y$ is blocked by the collider Z.

[b] Conditioning only on Z alone would *open* the back-door path $A \leftarrow W \rightarrow Z \leftarrow V \rightarrow Y$.

Controlling for Confounding

Example 3



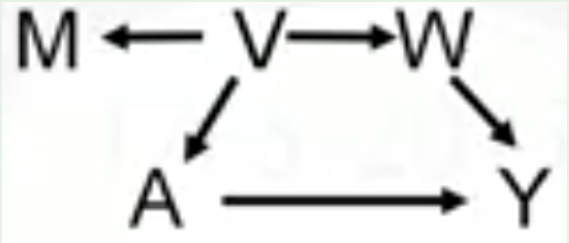
- Treatment: A; Outcome: Y
- Open back-door paths: $A \leftarrow Z \leftarrow V \rightarrow Y$, $A \leftarrow Z \rightarrow W \rightarrow M \rightarrow Y$, $A \leftarrow W \leftarrow Z \leftarrow V \rightarrow Y$, $A \leftarrow W \rightarrow M \rightarrow Y$.
- Sets of variables sufficient to control for confounding: $\{Z, W\}$, $\{Z, M\}$, $\{V, W\}$, $\{V, M\}$, $\{Z, W, M\}$, $\{Z, V, W\}$, $\{Z, V, M\}$, $\{V, M, W\}$, $\{Z, V, W, M\}$.

Disjunctive Cause Criterion

- One method to select what variables to control for confounding among the observed variables is the **disjunctive cause criterion** – control for all (observed) causes of treatment w and outcome y.
- Let V be the set of variables selected based on the disjunctive cause criterion. If there exists a set of observed variables x that satisfies the back-door criterion, then $x \subset V$.
- The disjunctive cause criterion places less knowledge requirement of the underlying causal structure on the investigator.

Disjunctive Cause Criterion

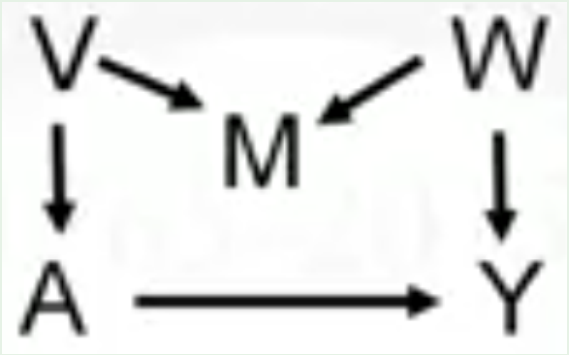
Example 1



- Treatment: A; Outcome: Y
- Observed: $\{A, Y, M, V, W\}$
- Disjunctive cause criterion selects: $\{V, W\}$
- Satisfies back-door criterion? Yes

Disjunctive Cause Criterion

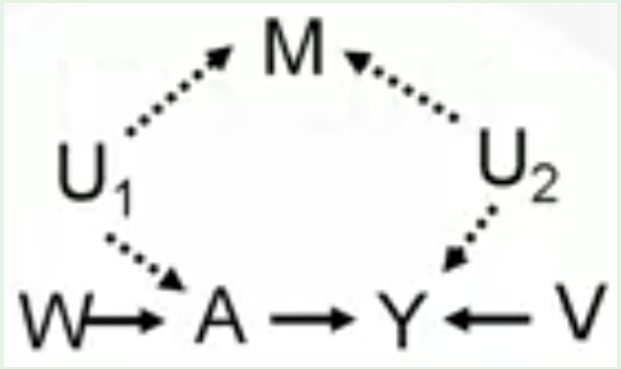
Example 2



- Treatment: A; Outcome: Y
- Observed: $\{A, Y, M, V, W\}$
- Disjunctive cause criterion selects: $\{V, W\}$
- Satisfies back-door criterion? Yes

Disjunctive Cause Criterion

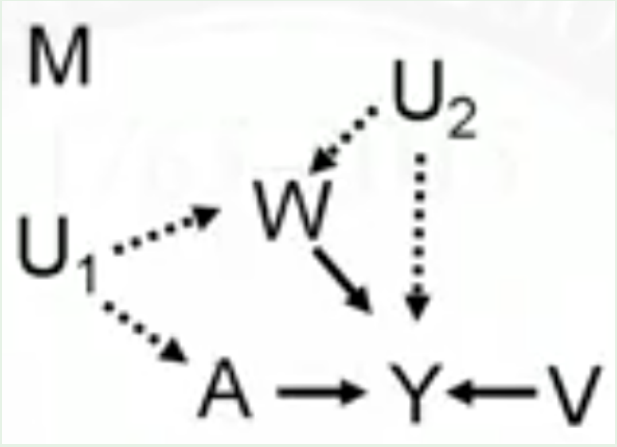
Example 3



- Treatment: A; Outcome: Y
- Observed: $\{A, Y, M, V, W\}$; Unobserved: $\{U_1, U_2\}$
- Disjunctive cause criterion selects: $\{V, W\}$
- Satisfies back-door criterion? Yes

Disjunctive Cause Criterion

Example 4



- Treatment: A; Outcome: Y
- Observed: $\{A, Y, M, V, W\}$; Unobserved: $\{U_1, U_2\}$
- Disjunctive cause criterion selects: $\{V, W\}$
- Satisfies back-door criterion? No. However, here no set of observed variables satisfies the back-door criterion.

Regression for Causal Inference

Assume we observe a set of variables x that satisfies the back-door criterion. Then according to (1), we have:

$$\mathbb{E}\left[y|\text{do}\left(w=a\right)\right]=\int\mathbb{E}\left[y|w=a,x\right]p\left(x\right)dx \tag{2}$$

$$\approx\frac{1}{N}\sum_{i}\mathbb{E}\left[y_{i}|\,w_{i}=a,x_{i}\right] \tag{3}$$

Therefore, if we can estimate $\mathbb{E}\left[y|w,x\right]$, then we have an estimate of $\mathbb{E}\left[y|\text{do}\left(w\right)\right]$[4].

[4] If x is discrete with values $\{1,\ldots,K\}$, then

$$\mathbb{E}\left[y|\text{do}\left(w\right)\right]=\sum_{k=1}^{K}\mathbb{E}\left[y|w,x=k\right]p\left(x=k\right) \tag{4}$$

, which we can estimate by estimating $\mathbb{E}\left[y|w,x=k\right]$ separately for each value of k and then combining them according to (4) using the empirical distribution of x as $p\left(x\right)$. This strategy is called **standardization** and is possible if (1) x is discrete or continuous but can be discretized without loss of much information; (2) x is low-dimensional.

Regression for Causal Inference

Once we have an estimate of $\mathbb{E}[y|\text{do}(w)]$, we can use it to compute the average treatment effect:

$$\tau_{\text{ATE}}(w) = \frac{d\mathbb{E}[y|\text{do}(w)]}{dw}$$

If $w \in \{0, 1\}$ is a binary treatment, then

$$\tau_{\text{ATE}} = \mathbb{E}[y|\text{do}(w = 1)] - \mathbb{E}[y|\text{do}(w = 0)]$$

Regression for Causal Inference

In the potential outcomes framework, the back-door criterion is expressed as the **unconfoundedness** (or, **conditional exchangeability**) condition.

Unconfoundedness

For $w \in \{1, \ldots, A\}$,

$$w \perp (\mathcal{Y}(1), \ldots, \mathcal{Y}(A)) | x \tag{5}$$

Regression for Causal Inference

Under unconfoundedness,

$$\begin{aligned}\mathbb{E}\left[\mathcal{Y}\left(a\right)\right] &= \mathbb{E}\left[\mathbb{E}\left[\mathcal{Y}\left(a\right)|\, x\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\mathcal{Y}\left(a\right)|\, w=a, x\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[y|\, w=a, x\right]\right] \\ &= \mathbb{E}\left[\mu_a\left(x\right)\right]\end{aligned} \tag{6}$$

, where we define $\mu_a\left(x\right) \doteq \mathbb{E}\left[y|\, w=a, x\right]$ as the **conditional regression function**[5,6].

---

[5] (6) is equivalent to (2), since $\mathbb{E}\left[\mathcal{Y}\left(a\right)\right] = \mathbb{E}\left[y|\mathrm{do}\left(w=a\right)\right]$.

[6] Note that

$$\mu_a\left(x=x_i\right) = \mathbb{E}\left[\mathcal{Y}_i\left(a\right)\right]$$

In particular, (6) does *not* imply $\mu_a\left(x\right) = \mathcal{Y}\left(a\right)$. $\mu_a\left(x\right)$ is a function of x, while $\mathcal{Y}\left(a\right)$ is a r.v. itself. This can be made clear by writing (6) as $\mathbb{E}_{\mathcal{Y}(a)}\left[\mathcal{Y}\left(a\right)\right] = \mathbb{E}_x\left[\mu_a\left(x\right)\right]$.

Regression for Causal Inference

When $w \in \{0, 1\}$, we have:

$$\begin{aligned}\tau_{\mathsf{ATE}} &= \mathbb{E}\left[\mathcal{Y}\left(1\right) - \mathcal{Y}\left(0\right)\right] \\ &= \mathbb{E}\left[\mu_1\left(x\right) - \mu_0\left(x\right)\right]\end{aligned}$$

- With a sufficient set of observed variables x that controls for confounding, the causal effect learning problem boils down to the statistical learning problem of estimating $\mu_w\left(x\right) = \mathbb{E}\left[y|w, x\right]$.

- This can be achieved using *any* appropriate parametric, semi-parametric, or non-parametric statistical models.

Simulation 1

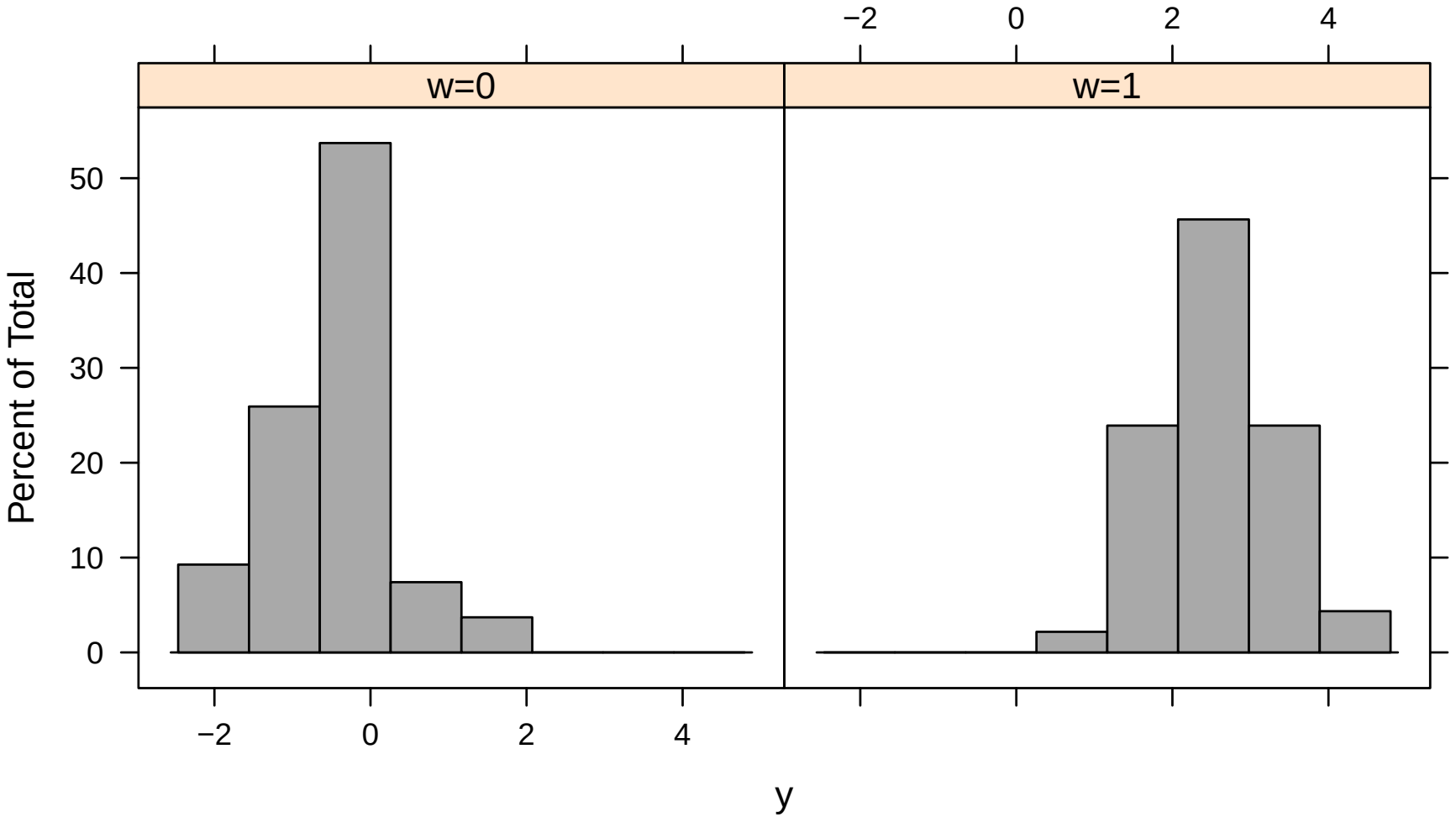
Let's generate some data with binary treatment w and outcome y[7]:

$$
\begin{aligned}
x &\sim \mathcal{N}(0, 4) \\
w &\leftarrow \text{Bernoulli}\left((1 + \exp(-x))^{-1}\right) \\
y &\leftarrow 2w + 0.5x + e, \quad e \sim \mathcal{N}(0, 0.01)
\end{aligned}
$$

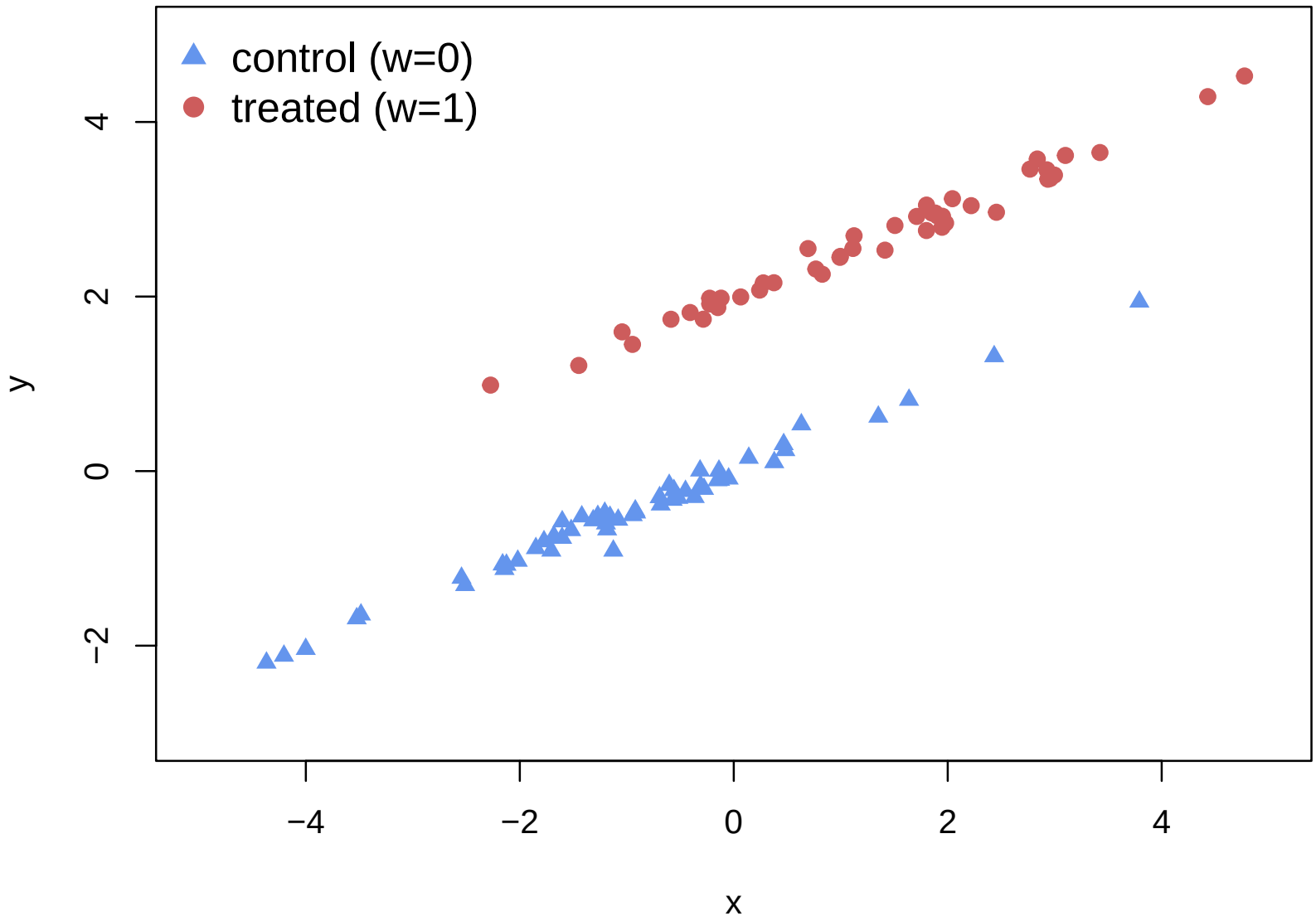
```
# Simulation
require(sigmoid)
n = 100
x = 2*rnorm(n)
w = rbinom(n,1,sigmoid(x))
y = 0.5*x + 2*w + 0.1*rnorm(n)
```

[7] "$\leftarrow$" denotes a structural relationship: "$y \leftarrow x$" means "y is determined by x". See "Foundations of Causal Inference".

Simulation 1



Simulation 1



Simulation 1

Here the true treatment effect is homogeneous: $\tau = 2$.

Naive comparison of $\mathbb{E}[y|w = 1]$ and $\mathbb{E}[y|w = 0]$ gives biased estimate:

```
mean(y[w==1]) - mean(y[w==0])
```

```
## [1] 3.107458
```

Need to control for confounder x ...

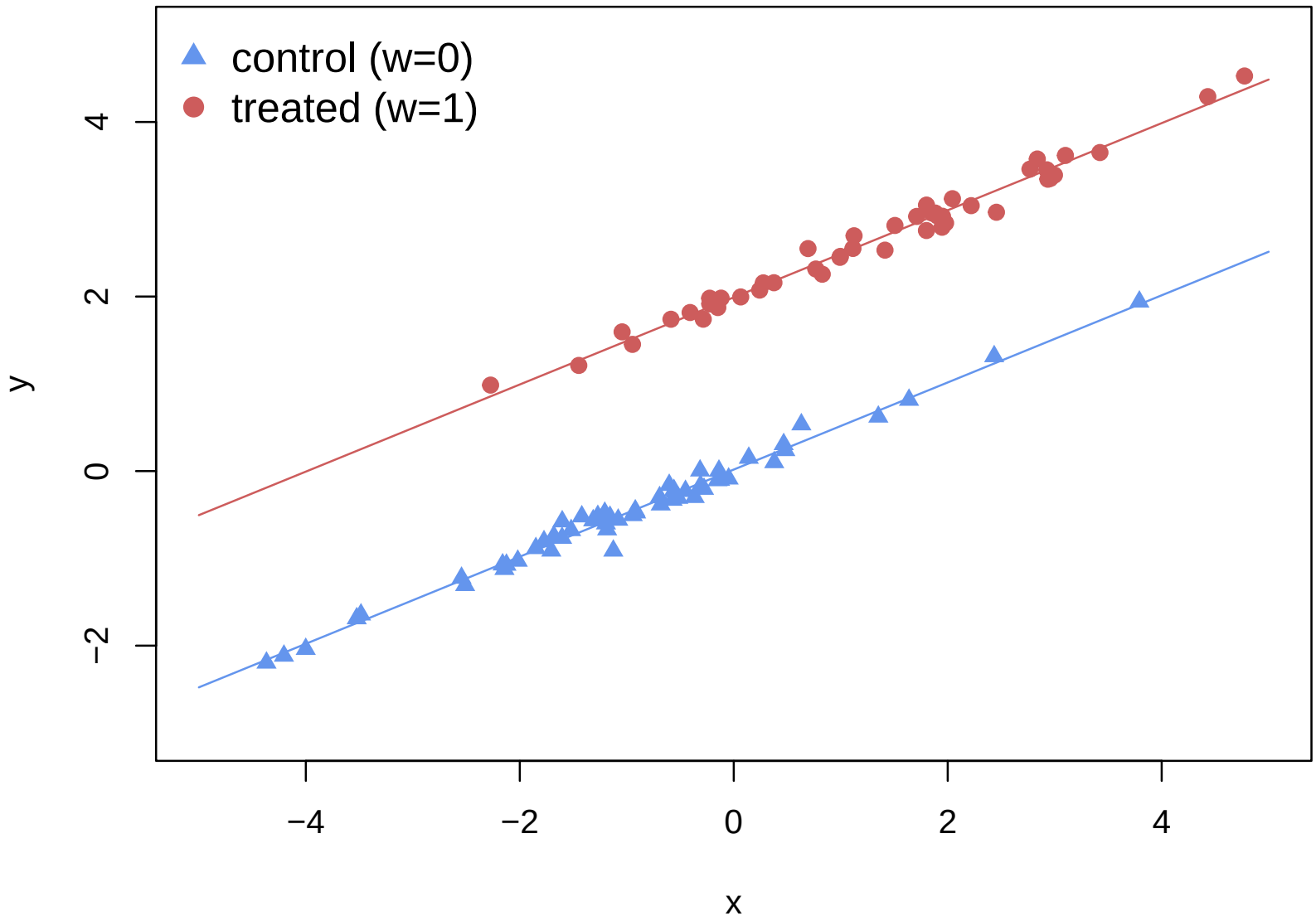
Simulation 1

Linear regression:

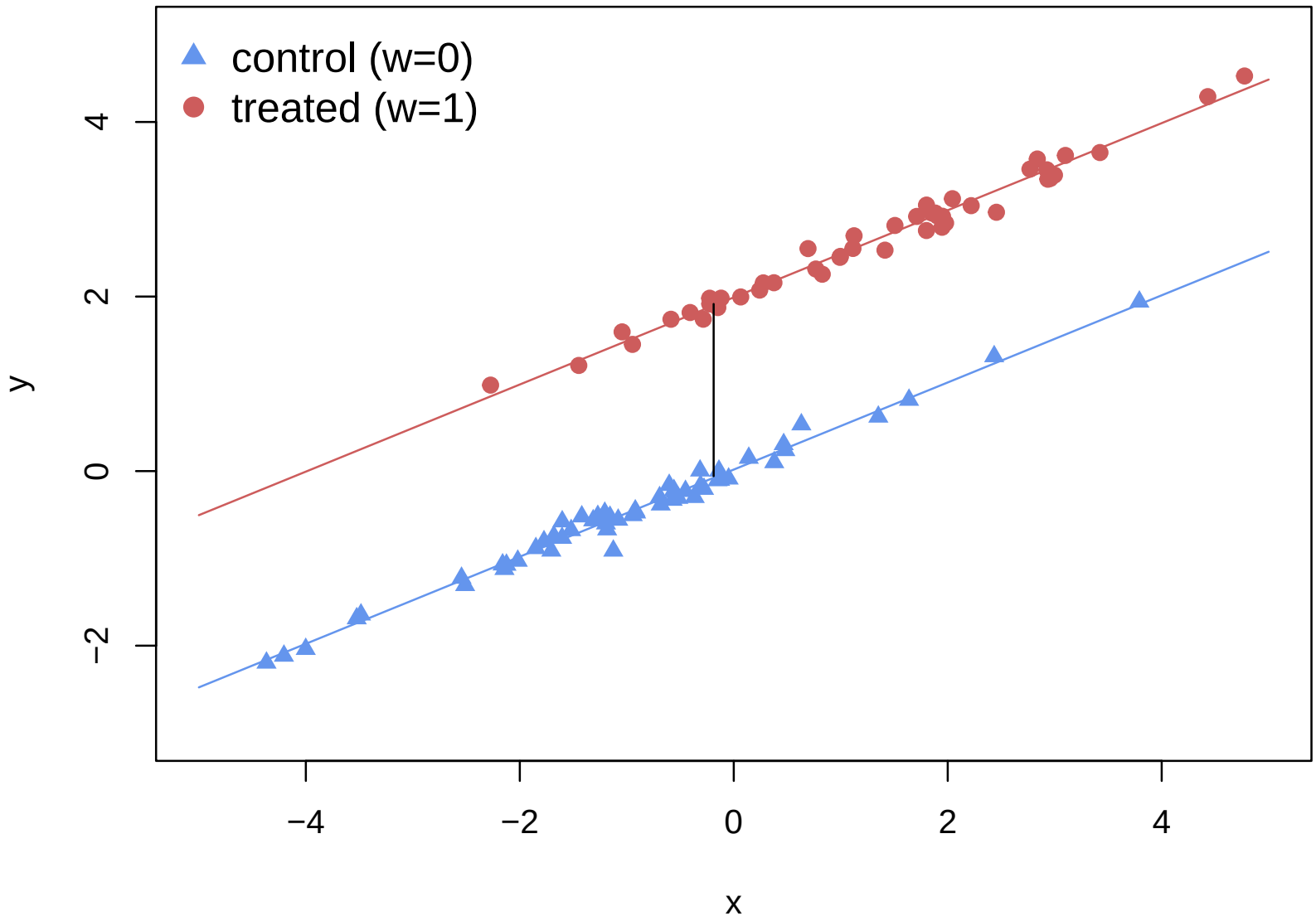
$$y = \beta_0 + \beta_1 w + \beta_2 x + e \tag{7}$$

```
require(AER)
data = data.frame(y=y,w=w,x=x)
fit = lm(y~.,data)
coeftest(fit)

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0177336  0.0149535  1.1859   0.2386
## w           1.9727686  0.0248930 79.2499   <2e-16 ***
## x           0.4990824  0.0065954 75.6710   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simulation 1



Simulation 1



Simulation 1

In model (7), $\widehat{\beta_1}$ *is* our estimate of the ATE:

$$\begin{aligned}
\widehat{\tau}_{\mathsf{ATE}} &= \widehat{\mathbb{E}}\left[y|\mathsf{do}\left(w=1\right)\right] - \widehat{\mathbb{E}}\left[y|\mathsf{do}\left(w=0\right)\right] \\
&= \int \left(\widehat{\mathbb{E}}\left[y|w=1,x\right] - \widehat{\mathbb{E}}\left[y|w=0,x\right]\right) p\left(x\right) dx \\
&= \int \left[\left(\widehat{\beta_0} + \widehat{\beta_1} + \widehat{\beta_2} x\right) - \left(\widehat{\beta_0} + \widehat{\beta_2} x\right)\right] p\left(x\right) dx = \widehat{\beta_1}
\end{aligned}$$

Simulation 1

In general, if the regression model for $\mathbb{E}[y|w,x]$ is additive in w and x:

$$\mathbb{E}[y|w,x] = \phi_1(w) + \phi_2(x) \tag{8}$$

, i.e., if we assume *no interaction* between w and x, then

$$\begin{aligned}
\tau_{\mathsf{ATE}}(w) &= \frac{d\mathbb{E}[y|\mathsf{do}(w)]}{dw} \\
&= \int \frac{\partial \mathbb{E}[y|w,x]}{\partial w} p(x)\, dx \\
&= \int \phi_1'(w)\, p(x)\, dx = \phi_1'(w)
\end{aligned}$$

Homogeneous and Heterogeneous Treatment Effects

Consider a binary treatment $w \in \{0, 1\}$. Let $\tau_i = \mathcal{Y}_i^1 - \mathcal{Y}_i^0$ be the individual treatment effect on individual i.

- In the special case that $\tau_i = \tau \ \forall i$, we say the treatment effect (in this population) is homogeneous.
- If x is a variable that describes individual characteristics, then homogenous treatment effect implies that $\mathbb{E}[y|w = 1, x] - \mathbb{E}[y|w = 0, x] = \tau$.
- In general, however, we should expect individual treatment effects to be heterogeneous in any given population.
- Under heterogeneous treatment effects, individual characteristics *could* be effect modifiers: it is possible for $\mathbb{E}[y|w = 1, x] - \mathbb{E}[y|w = 0, x] = \tau(x)$ to vary by x.

Simulation 2

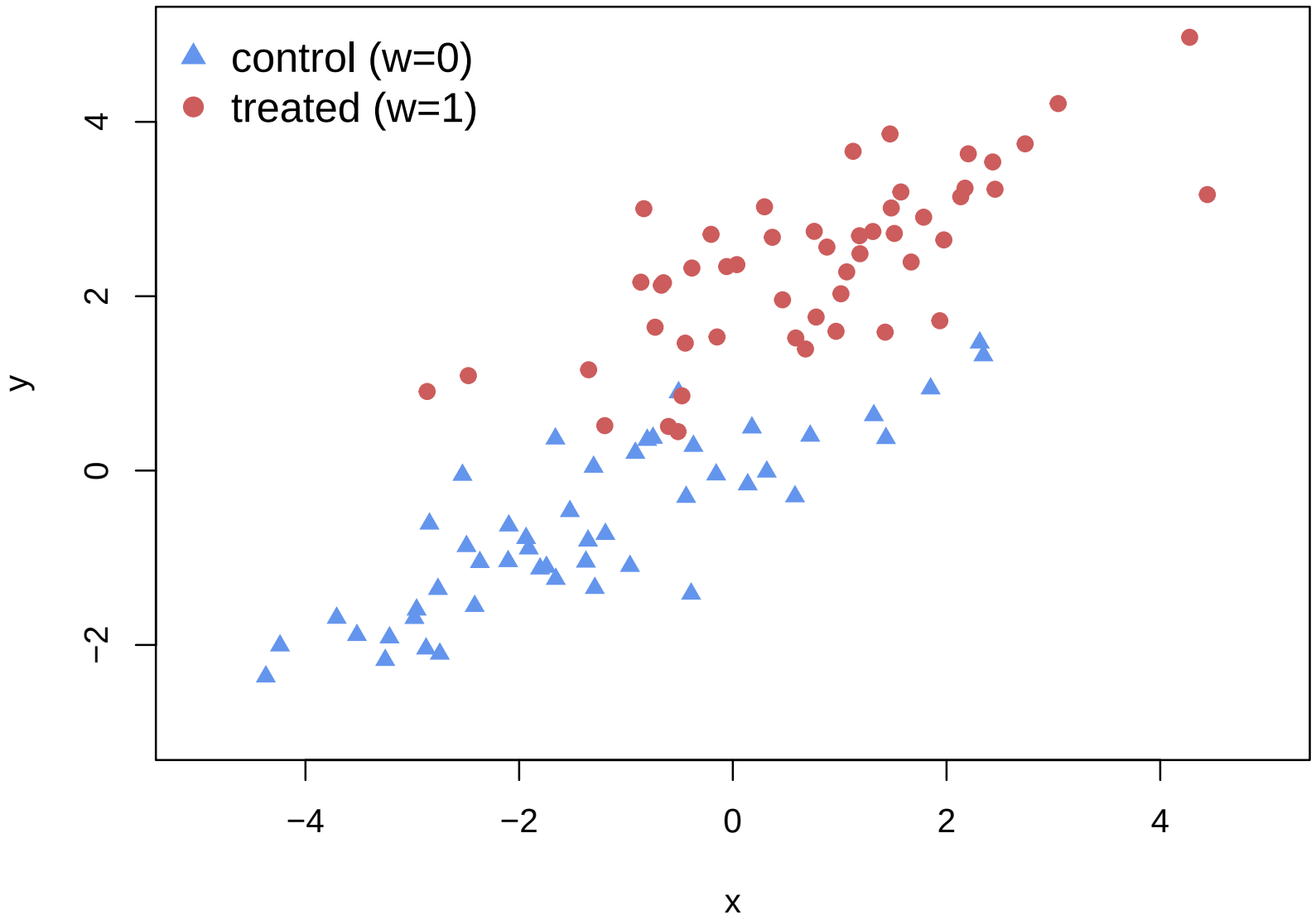
$$x \sim \mathcal{N}(0, 4)$$

$$w \leftarrow \text{Bernoulli}\left((1 + \exp(-x))^{-1}\right)$$

$$y \leftarrow \alpha \cdot w + 0.5x + e, \quad \alpha \sim \mathcal{N}(2, 0.25), \quad e \sim \mathcal{N}(0, 0.25)$$

```r
# Simulation
require(sigmoid)
n = 100
x = 2*rnorm(n)
w = rbinom(n,1,sigmoid(x))
y = (2+0.5*rnorm(n))*w + 0.5*x + 0.5*rnorm(n)
```

Simulation 2



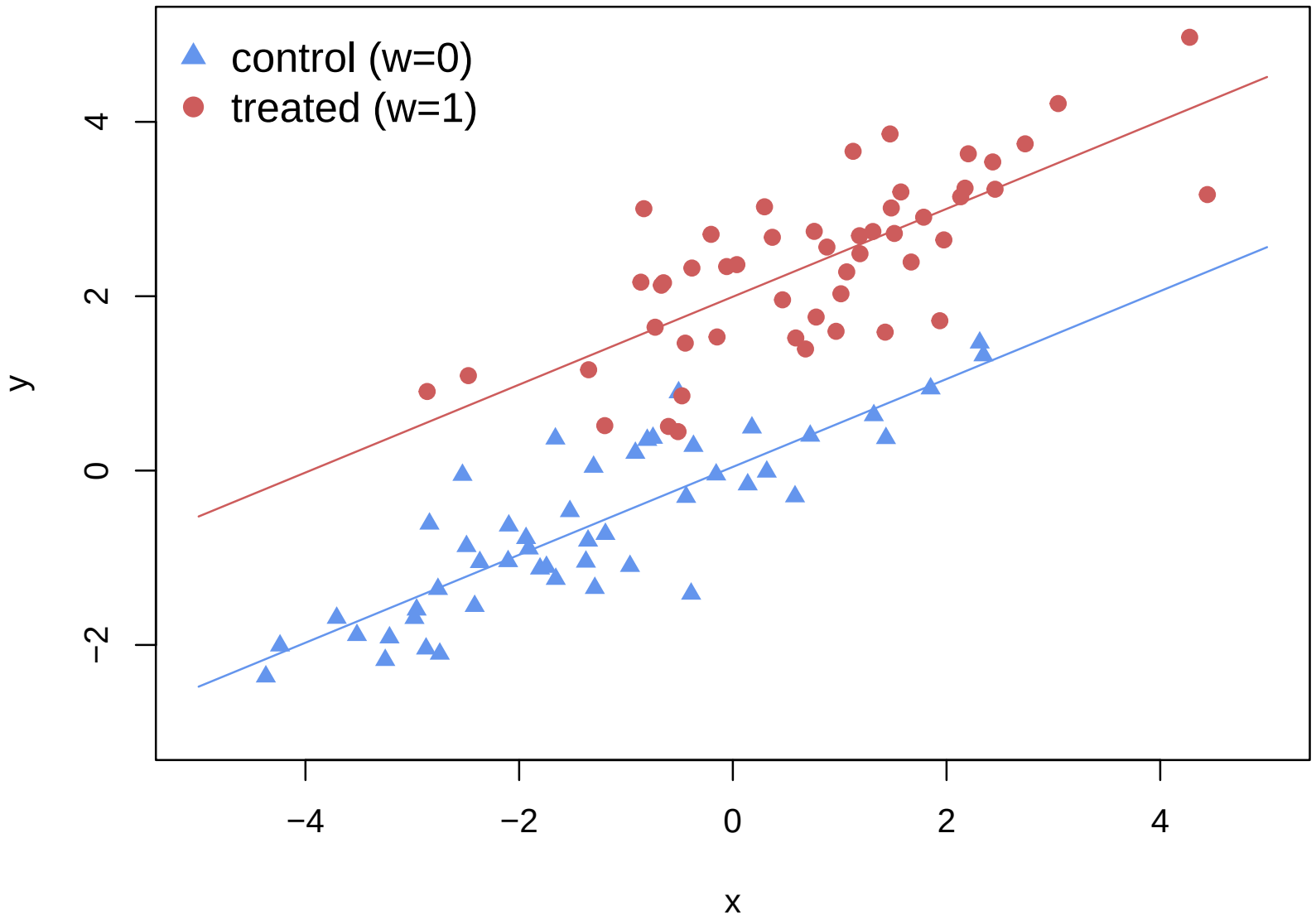
Simulation 2

Linear regression:

$$y = \beta_0 + \beta_1 w + \beta_2 x + e \tag{9}$$

```
require(AER)
data = data.frame(y=y,w=w,x=x)
fit = lm(y~.,data)
coeftest(fit)

##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.042018   0.100419  0.4184   0.6766
## w           1.951974   0.145319 13.4323   <2e-16 ***
## x           0.504104   0.038785 12.9975   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simulation 2



Simulation 2

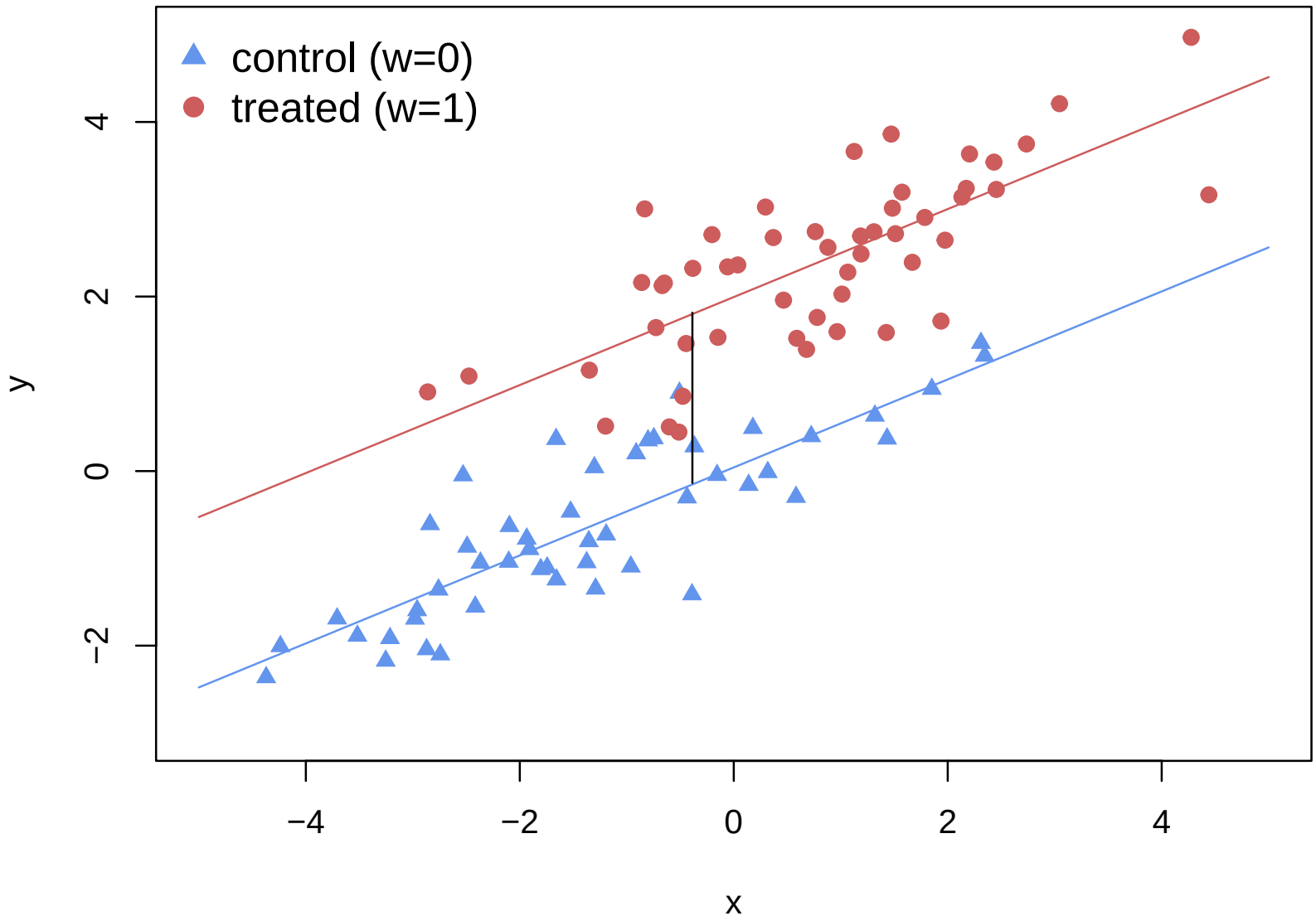
Here the treatment effect is heterogeneous: $\tau_i = \alpha_i$, but does not vary with x.

The average treatment effect

$$\begin{aligned}\tau_{\mathsf{ATE}} &= \mathbb{E}\left[\tau_i\right] = \mathbb{E}\left[\alpha_i\right] \\ &= \mathbb{E}\left[y|w=1,x\right] - \mathbb{E}\left[y|w=0,x\right]\end{aligned}$$

Hence in (9), $\widehat{\beta_1}$ is again our estimate of the ATE.

Simulation 2

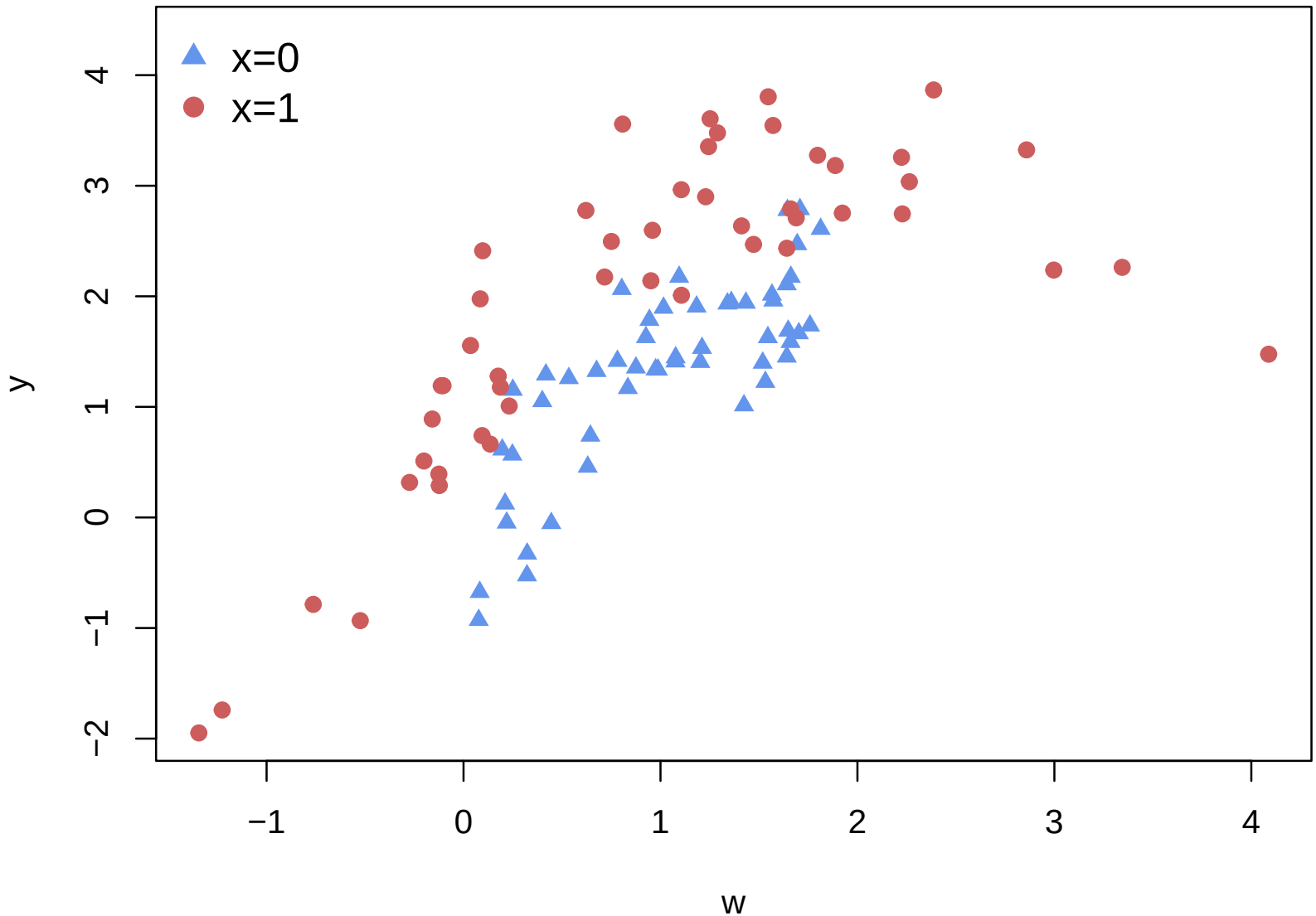


Simulation 3

$$
\begin{aligned}
x &\sim \text{Bernoulli}\,(0.5) \\
w &\leftarrow x \cdot \mathcal{N}\,(0,1) + U\,(0,2) \\
y &\leftarrow 2w - 0.5w^2 + x + e, \quad e \sim \mathcal{N}\,(0, 0.25)
\end{aligned}
$$

```
# Simulation
n = 100
x = rbinom(n,1,0.5)
w = 2*runif(n) + x*rnorm(n)
y = 2*w - 0.5*w^2 + x + 0.5*rnorm(n)
```

Simulation 3



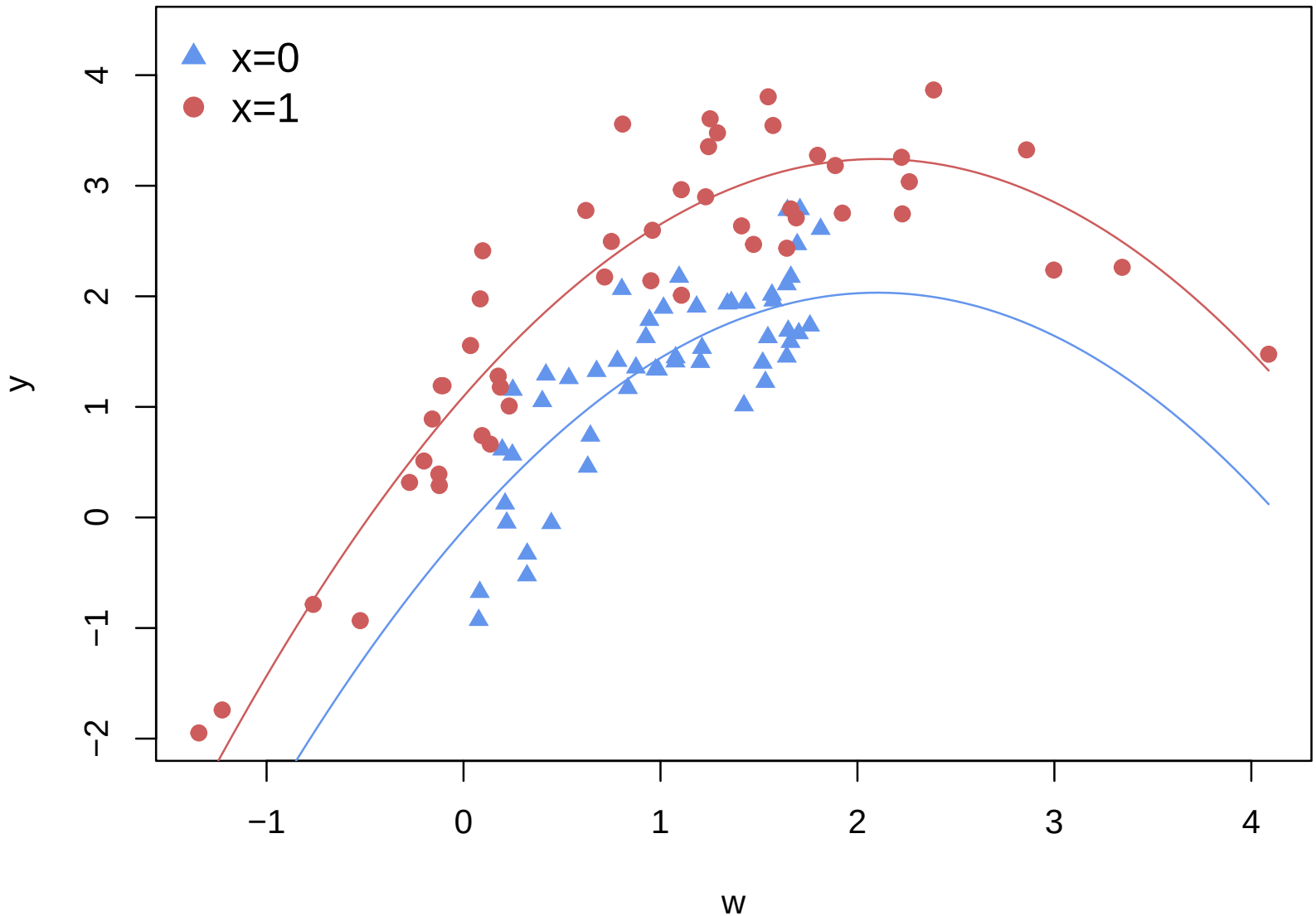
Simulation 3

Polynomial regression in w:

$$y = \beta_0 + \beta_1 w + \beta_2 w^2 + \beta_3 x + e \tag{10}$$

```
require(AER)
fit = lm(y ~ poly(w,2,raw=T) + x)
coeftest(fit)

##
## t test of coefficients:
##
##                          Estimate Std. Error  t value Pr(>|t|)
## (Intercept)             -0.114061   0.098290  -1.1605   0.2487
## poly(w, 2, raw = T)1     2.042029   0.105225  19.4062   <2e-16 ***
## poly(w, 2, raw = T)2    -0.485534   0.039532 -12.2820   <2e-16 ***
## x                        1.208719   0.108421  11.1484   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simulation 3



Simulation 3

Here the treatment effect is homogeneous but non-constant (in w).

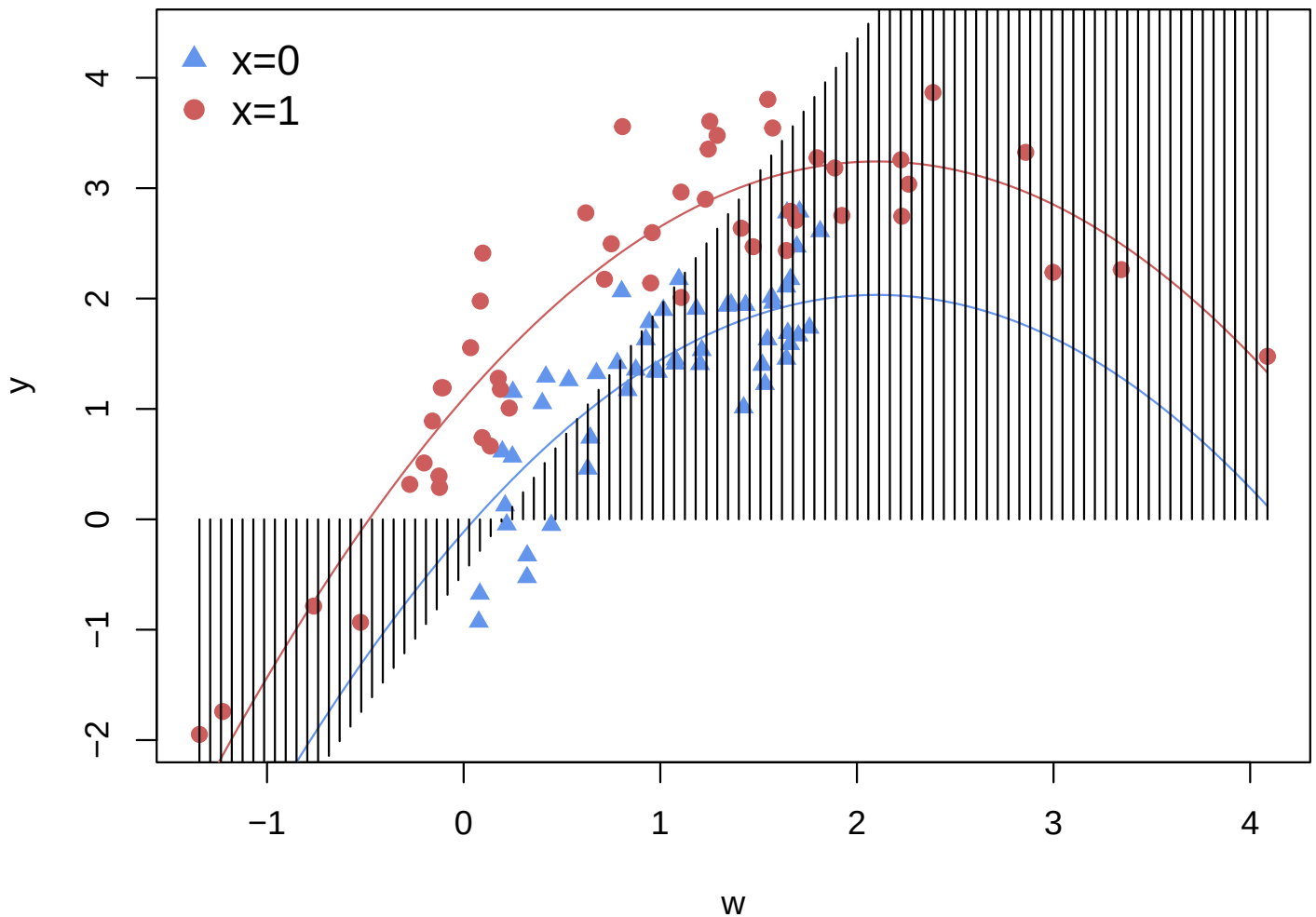
Given the additive structure of (8),

$$\tau_{\text{ATE}}(w) = \phi_1'(w) = 2 - w$$

Given (10),

$$\widehat{\tau}_{\text{ATE}}(w) = \widehat{\beta_1} + 2\widehat{\beta_2} w$$

Simulation 3



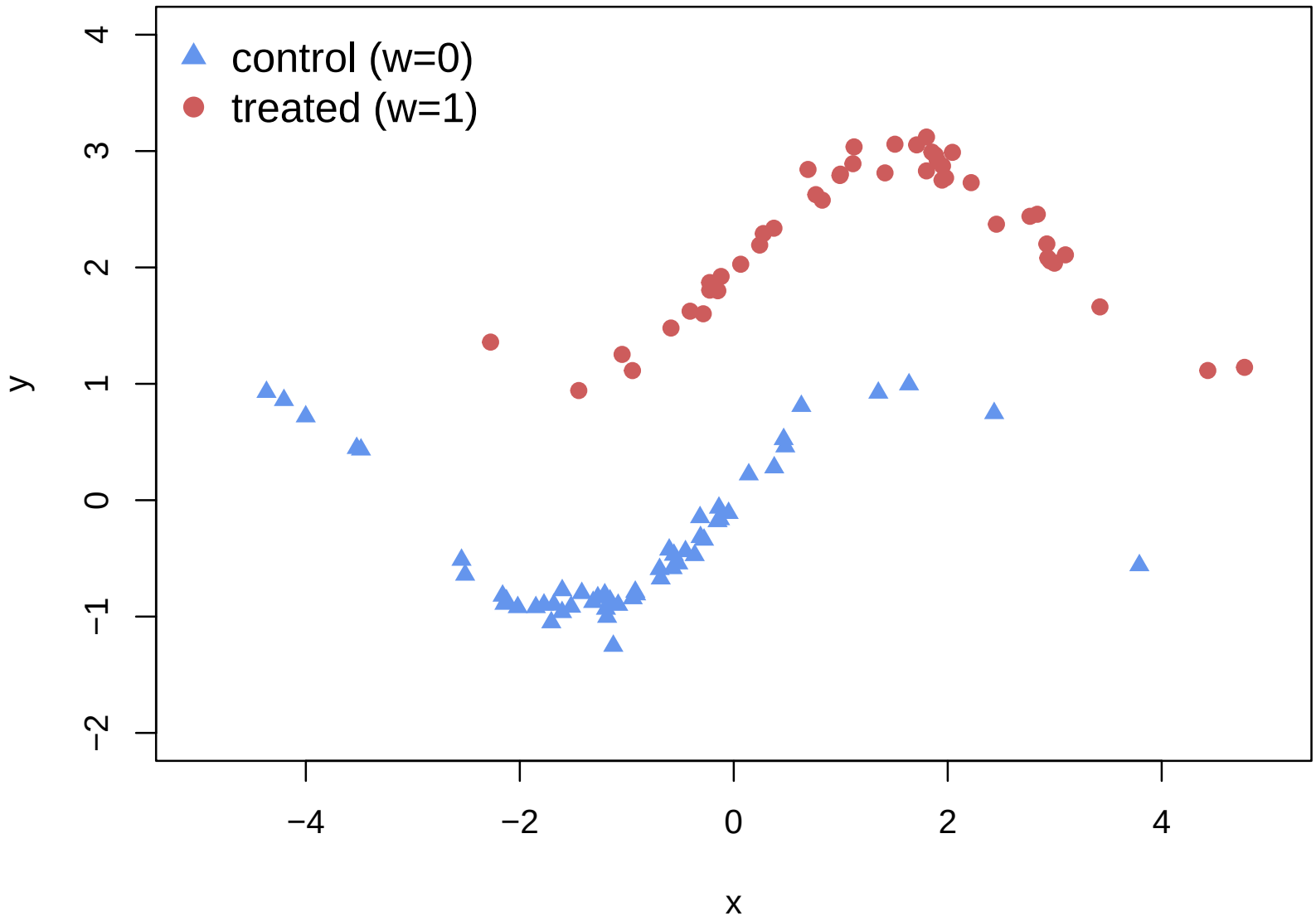
Vertical lines represent $\widehat{\beta}_1 + 2\widehat{\beta}_2 w$: the estimated ATE.

Simulation 4

$$
\begin{aligned}
x &\sim \mathcal{N}(0, 4) \\
w &\leftarrow \text{Bernoulli}\left((1 + \exp(-x))^{-1}\right) \\
y &\leftarrow 2w + \sin(x) + e, \quad e \sim \mathcal{N}(0, 0.01)
\end{aligned}
$$

```r
# Simulation
require(sigmoid)
n = 100
x = 2*rnorm(n)
w = rbinom(n,1,sigmoid(x))
y = sin(x) + 2*w + 0.1*rnorm(n)
```

Simulation 4



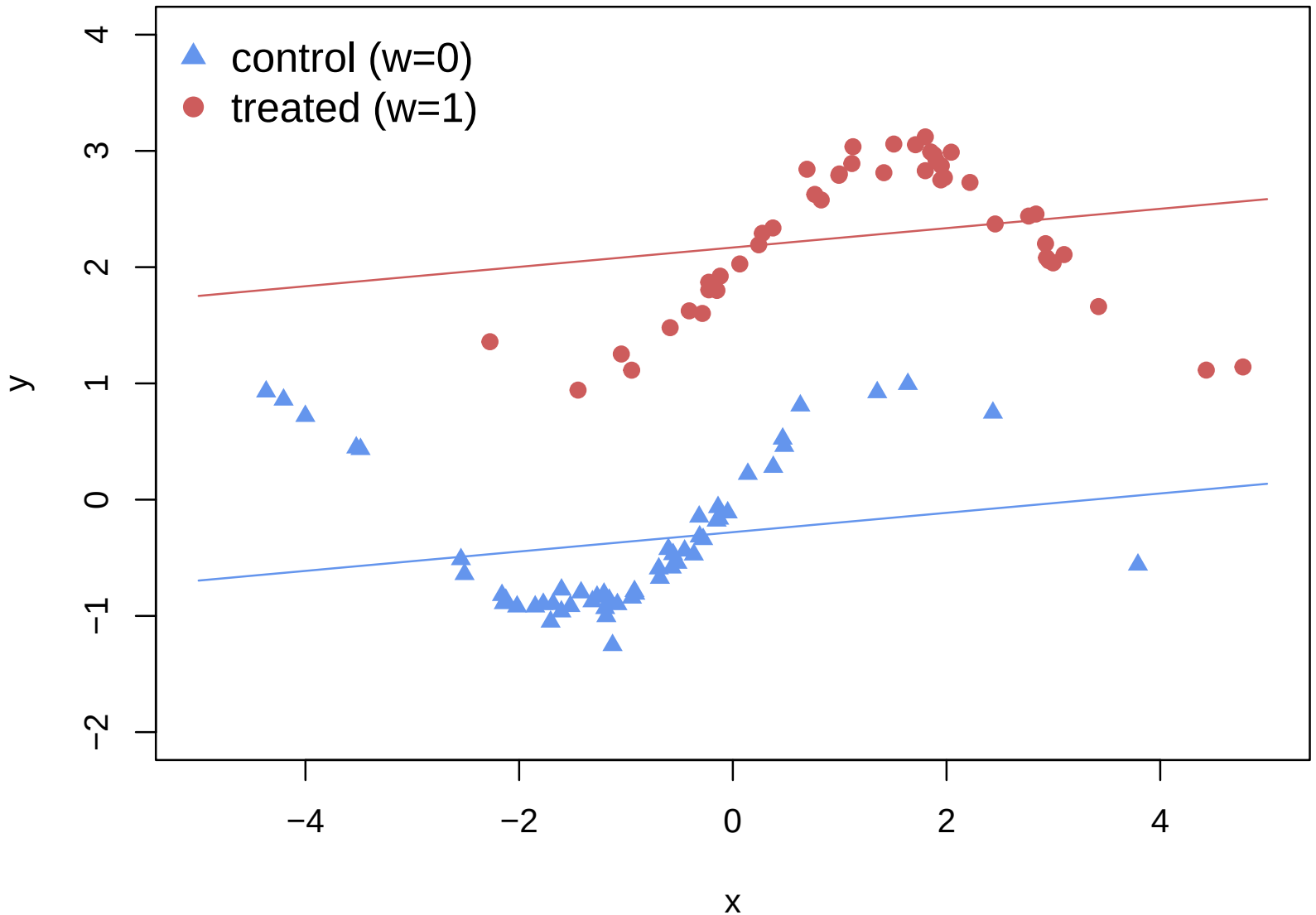
Simulation 4

Linear regression:

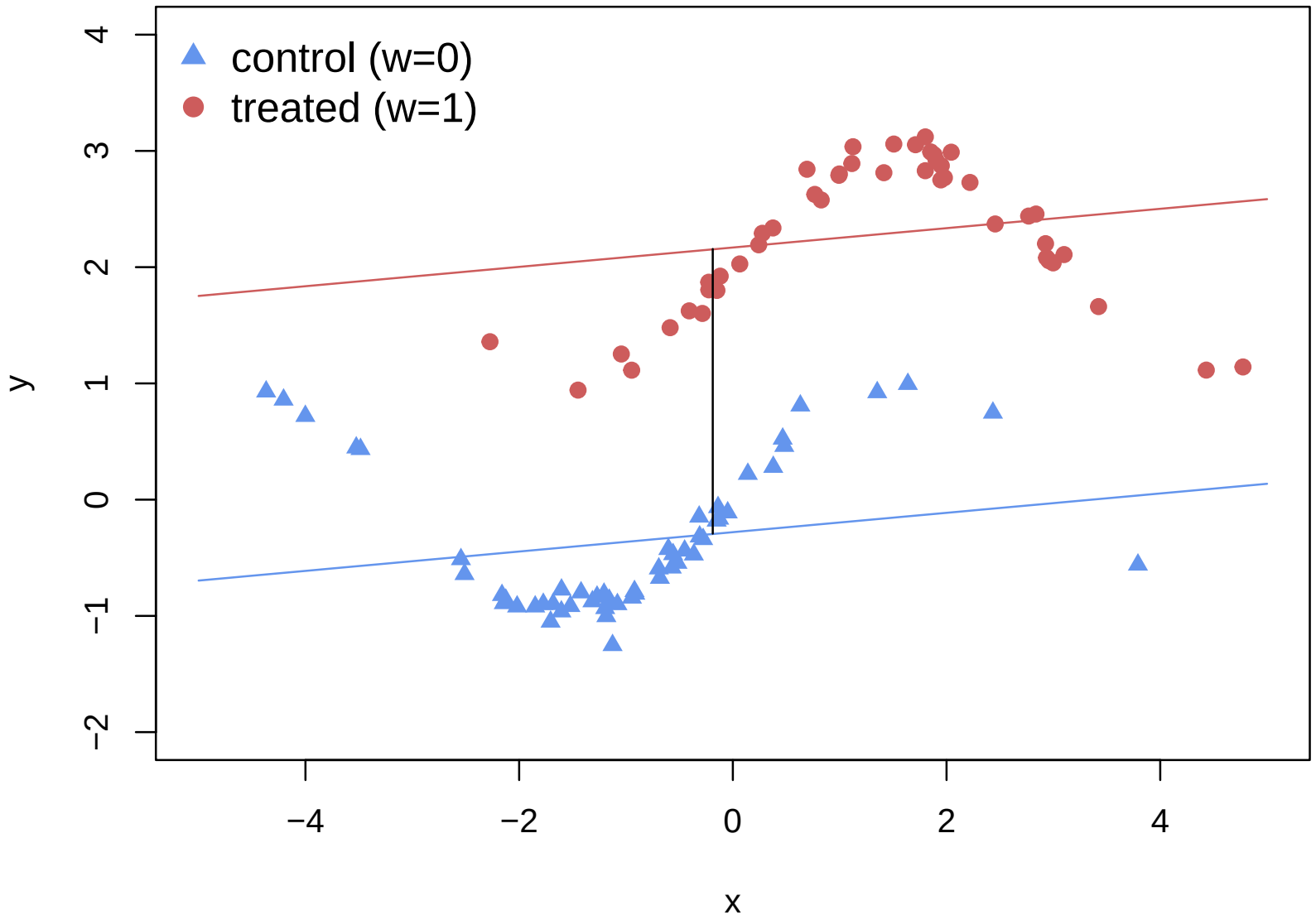
$$y = \beta_0 + \beta_1 w + \beta_2 x + e$$

```
require(AER)
data = data.frame(y=y,w=w,x=x)
fit = lm(y~.,data)
coeftest(fit)

##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -0.279858   0.094458 -2.9628  0.003835 **
## w            2.448736   0.157244 15.5729 < 2.2e-16 ***
## x            0.083252   0.041662  1.9983  0.048487 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simulation 4



Simulation 4



Simulation 4

Semi-parametric generalized additive model:

$$y = \beta_0 + \beta_1 w + g(x) + e$$

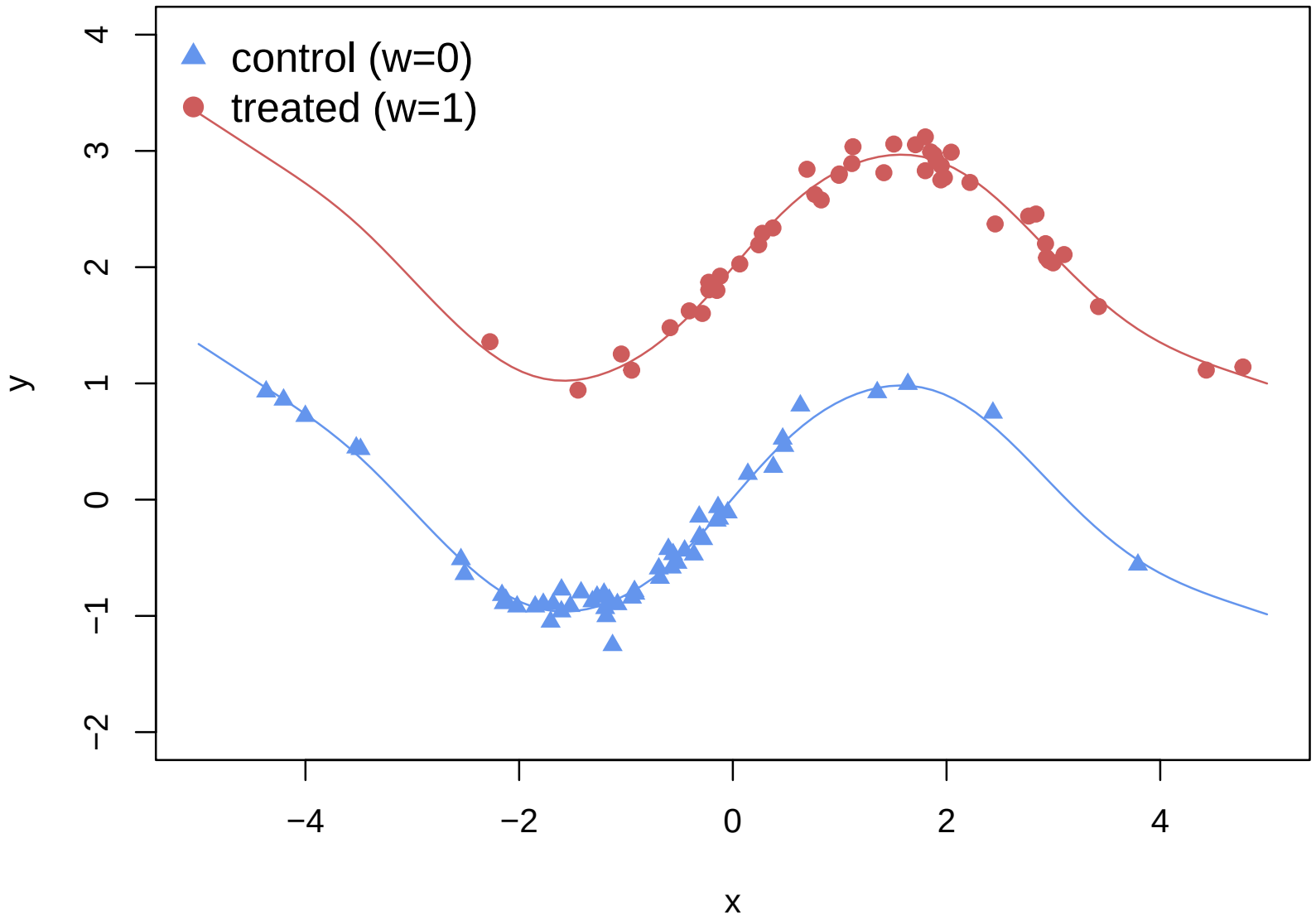
, where $g(x)$ is a smoothing spline.

```
require(mgcv)
fit = gam(y ~ w + s(x),data,family=gaussian)
```

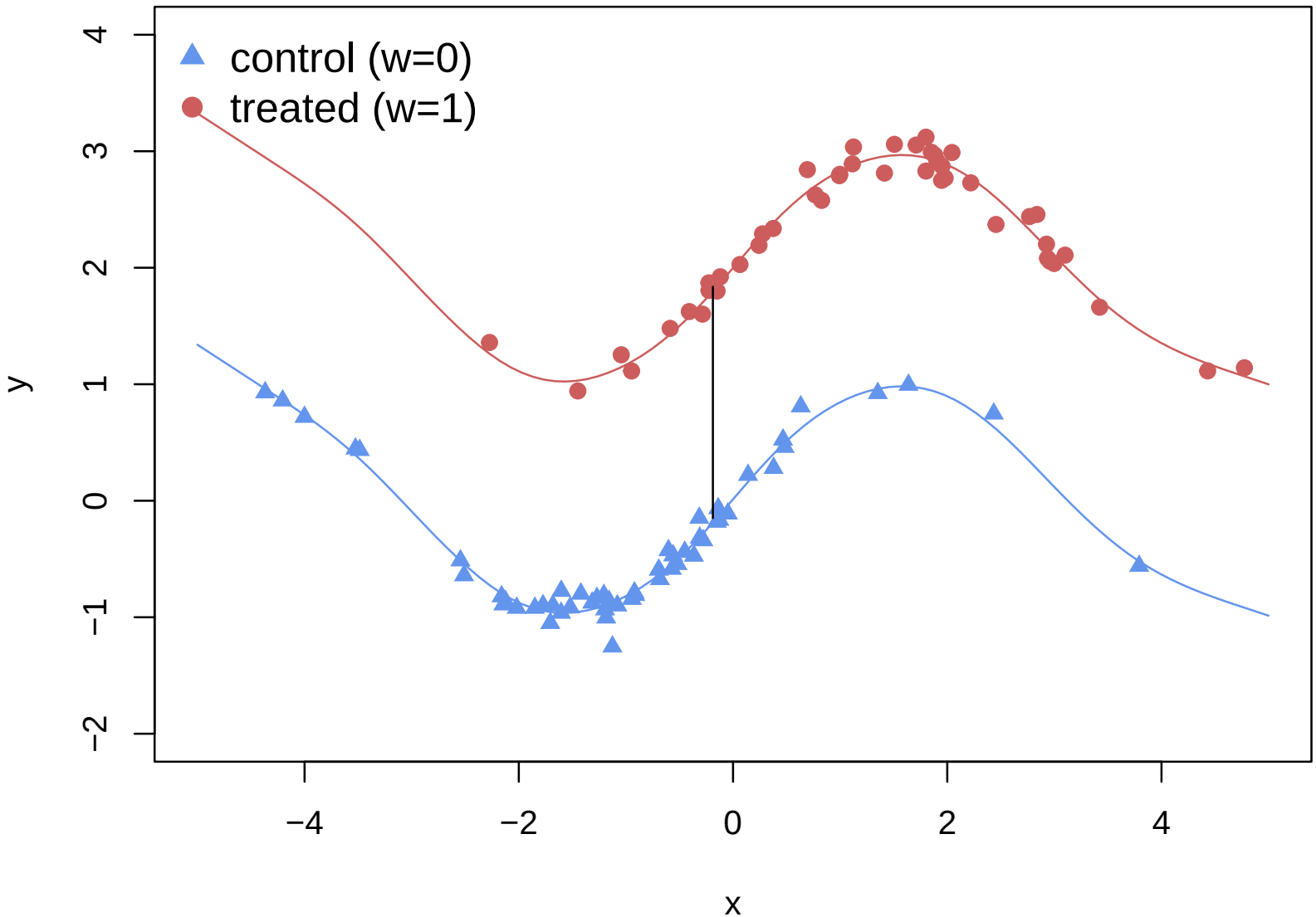
Simulation 4

```
summary(fit)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ w + s(x)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06169    0.01589  -3.883 0.000197 ***
## w            1.98592    0.02665  74.509  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##        edf Ref.df     F p-value
## s(x) 8.455  8.914 422.5  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Simulation 4



Simulation 4

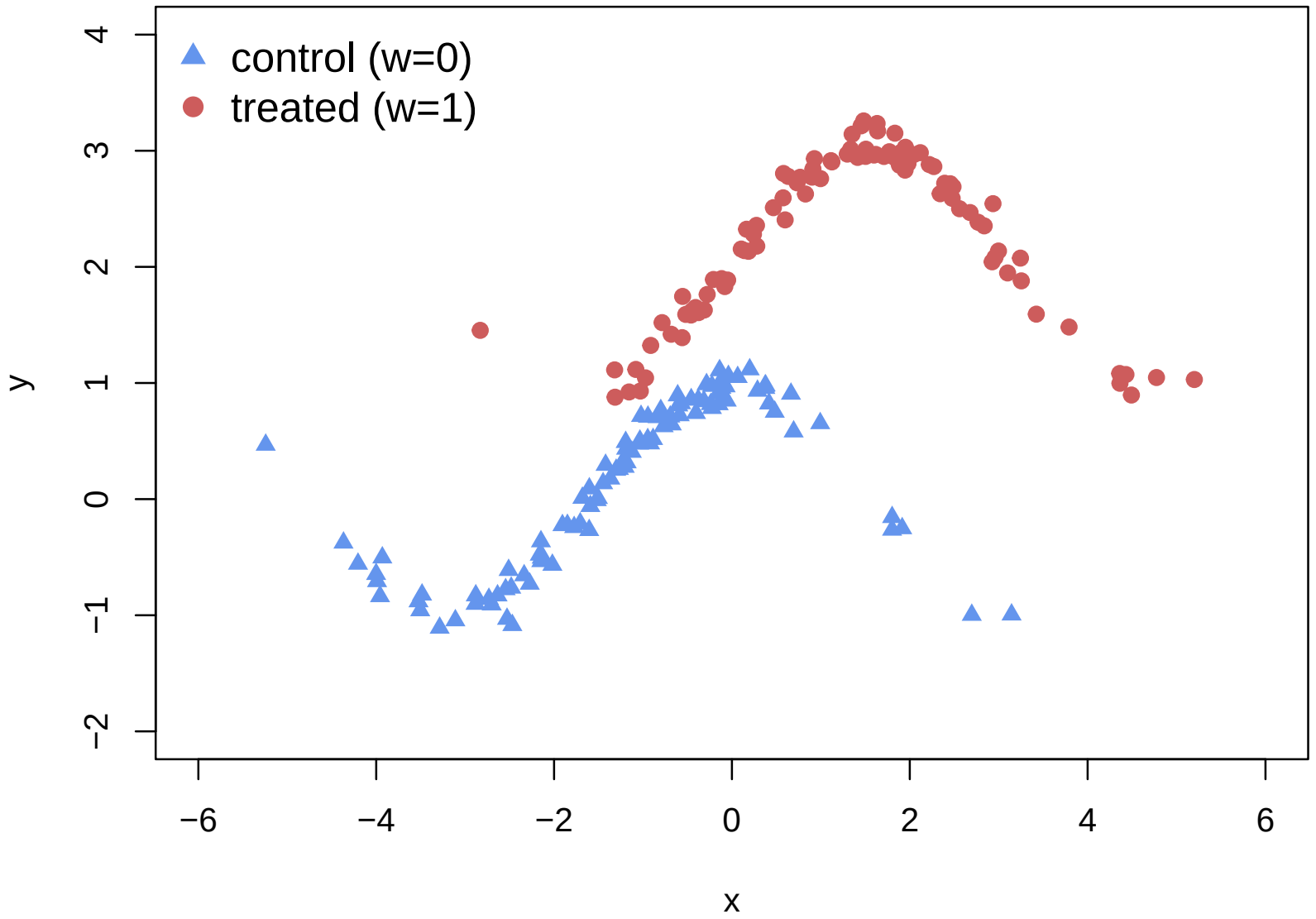


Simulation 5

$$
\begin{aligned}
x &\sim \mathcal{N}(0, 4) \\
w &\leftarrow \text{Bernoulli}\left((1+\exp(-x))^{-1}\right) \\
\mathcal{Y}^0 &\leftarrow \cos(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.01) \\
\mathcal{Y}^1 &\leftarrow 2 + \sin(x) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 0.01) \\
y &= w\mathcal{Y}^1 + (1-w)\mathcal{Y}^0
\end{aligned}
$$

```r
# Simulation
require(sigmoid)
n = 200
x = 2*rnorm(n)
w = rbinom(n,1,sigmoid(x))
y0 = cos(x) + 0.1*rnorm(n)
y1 = 2 + sin(x) + 0.1*rnorm(n)
y = y0*(w==0) + y1*(w==1)
```

Simulation 5



Simulation 5

Here the treatment effect is heterogeneous and varies with x:

$$\begin{aligned}
\tau_{\text{ATE}} &= \mathbb{E}\left[\mathcal{Y}^1 - \mathcal{Y}^0\right] \\
&= \int \left(\mathbb{E}\left[y|w=1,x\right] - \mathbb{E}\left[y|w=0,x\right]\right) p\left(x\right) dx \\
&\approx \frac{1}{N}\sum_i \left(\mathbb{E}\left[y_i|\, w_i=1,x_i\right] - E\left[y_i|\, w_i=0,x_i\right]\right)
\end{aligned}$$

```
# ATE
ate = mean(y1 - y0)
ate
```

```
## [1] 1.869096
```

Simulation 5

To model the interaction between w and x, we can run the following linear regression:

$$y = \beta_0 + \beta_1 w + \beta_2 x + \beta_3 w \cdot x + e \tag{11}$$

Or more flexibly, we can run the following set of regressions[8]:

$$\begin{cases} y = \beta_0 + \beta_1 x + \epsilon & \text{if } w = 0 \\ y = \beta_2 + \beta_3 x + \varepsilon & \text{if } w = 1 \end{cases} \tag{12}$$

Estimating (12) $\Rightarrow \widehat{\mathbb{E}}[y|w = 0, x]$ and $\widehat{\mathbb{E}}[y|w = 1, x]$, from which we obtain:

$$\widehat{\tau}_{\text{ATE}} \approx \frac{1}{N} \sum_i \left(\widehat{\mathbb{E}}[y_i|\, w_i = 1, x_i] - \widehat{\mathbb{E}}[y_i|\, w_i = 0, x_i] \right)$$

[8] (11) and (12) are equivalent when w is binary.

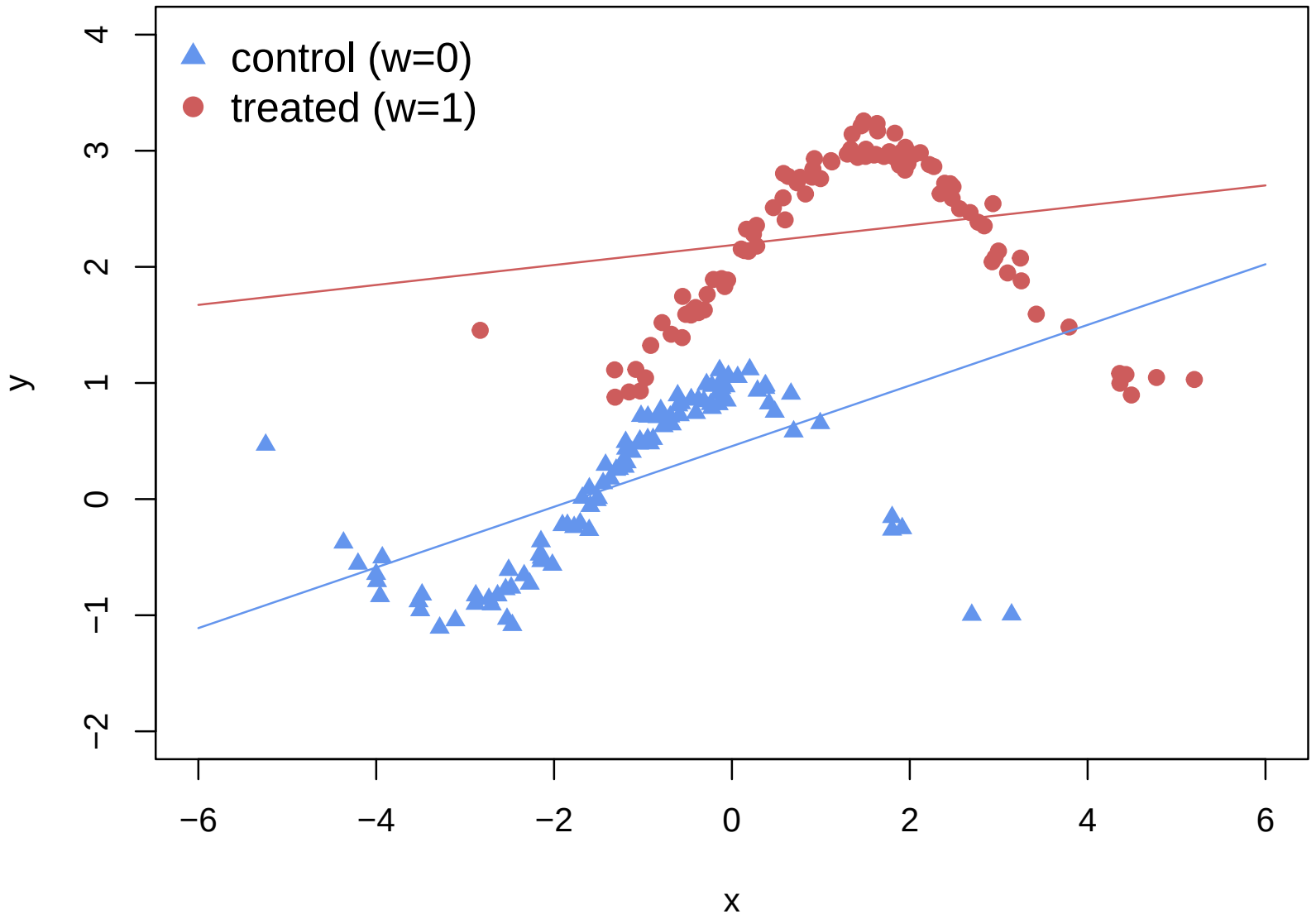
Simulation 5

```r
# Linear Regression
data = data.frame(y=y,w=w,x=x)
fit0 = lm(y ~ x,data,subset=(w==0))
fit1 = lm(y ~ x,data,subset=(w==1))

# Estimated ATE
y0hat = predict(fit0,data)
y1hat = predict(fit1,data)
atehat = mean(y1hat - y0hat)
atehat

## [1] 1.722792
```

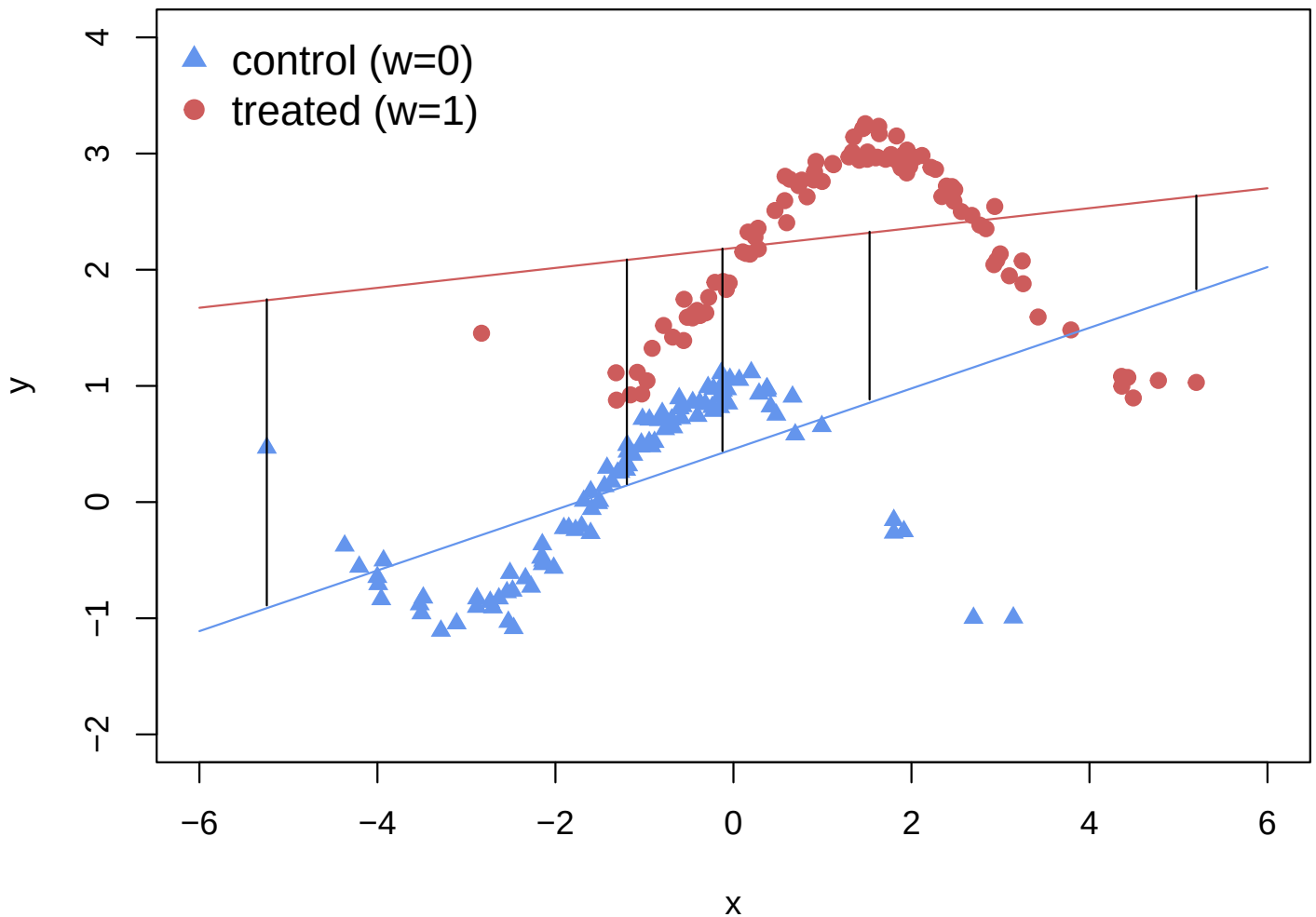
Simulation 5

```
require(AER)
coeftest(fit0)

##
## t test of coefficients:
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 0.455760   0.074658  6.1047 2.123e-08 ***
## x           0.261113   0.038104  6.8527 6.698e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coeftest(fit1)

##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.186996   0.090796  24.087  < 2e-16 ***
## x           0.085725   0.045190   1.897  0.06074 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simulation 5



Simulation 5



Vertical lines represent estimated ATEs conditional on x at 1%, 25%, 50%, 75%, and 99% percentiles.

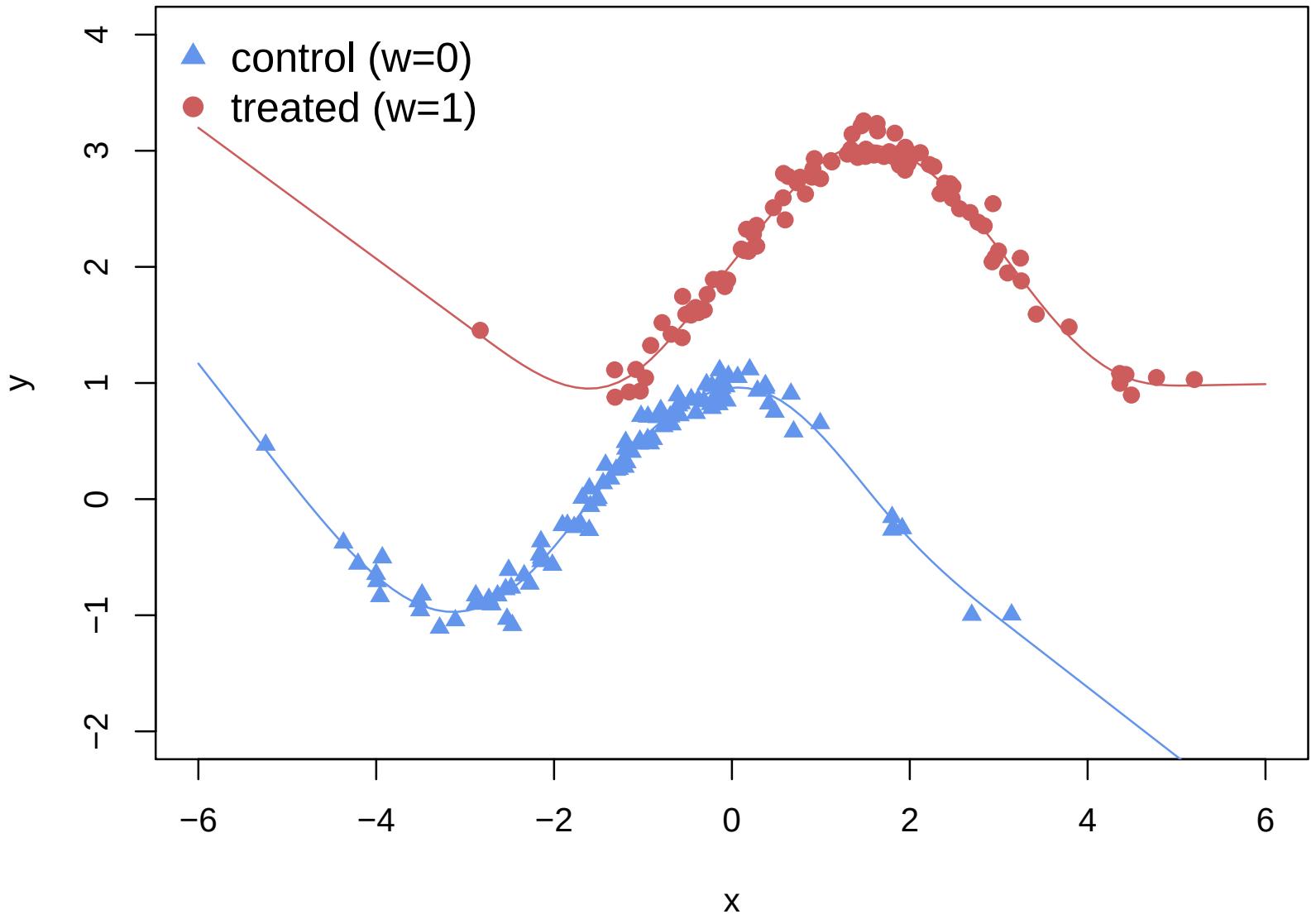
Simulation 5

Smoothing splines:

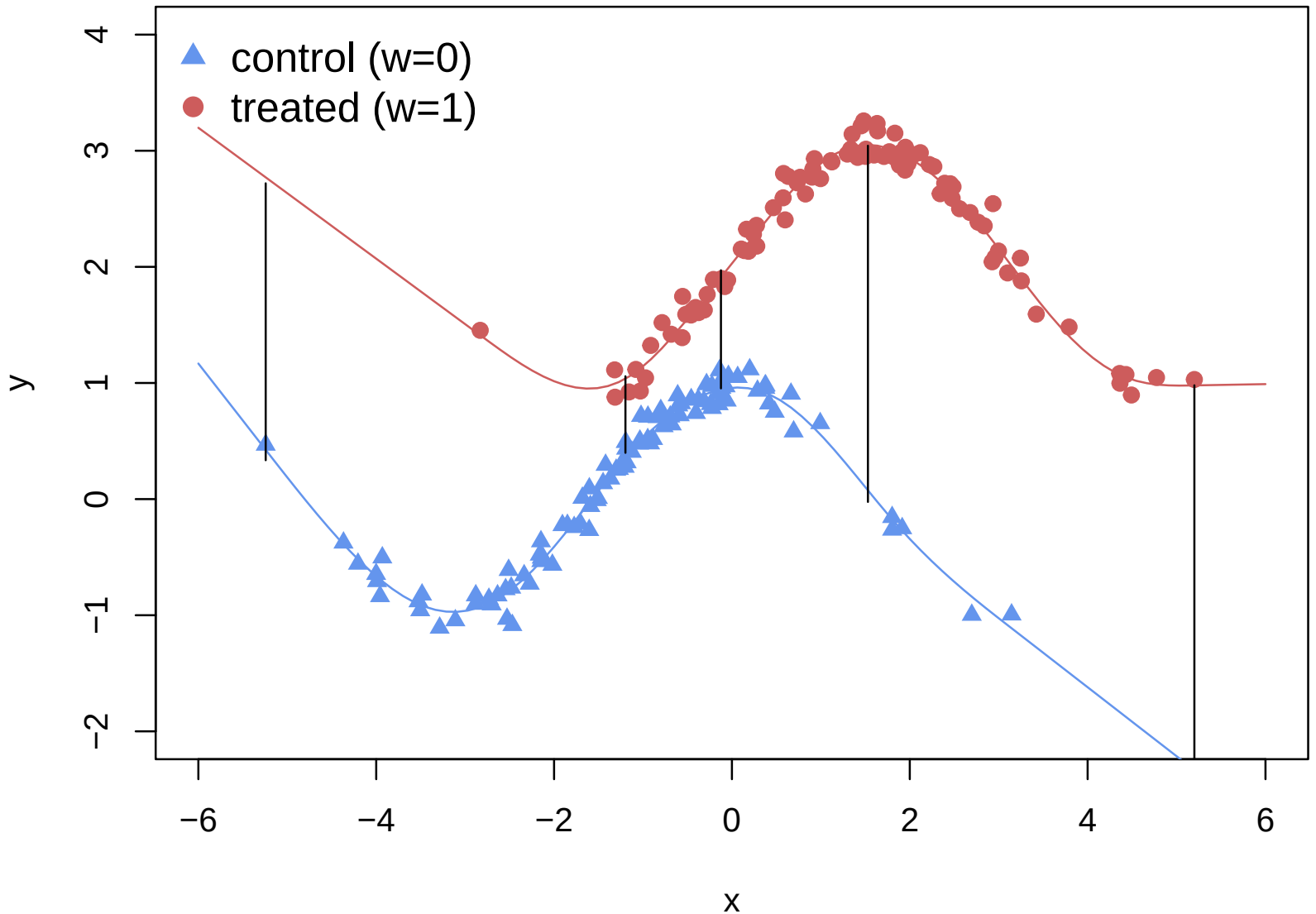
$$\begin{cases} y = g_1(x) + \epsilon & \text{if } w = 0 \\ y = g_0(x) + \varepsilon & \text{if } w = 1 \end{cases}$$

```
# Smoothing Splines
require(mgcv)
fit0 = gam(y ~ s(x),data,subset=(w==0),family=gaussian)
fit1 = gam(y ~ s(x),data,subset=(w==1),family=gaussian)

# Estimated ATE
y0hat = predict(fit0,data)
y1hat = predict(fit1,data)
atehat = mean(y1hat - y0hat)
atehat

## [1] 1.877149
```

Simulation 5



Simulation 5



Simulation 6

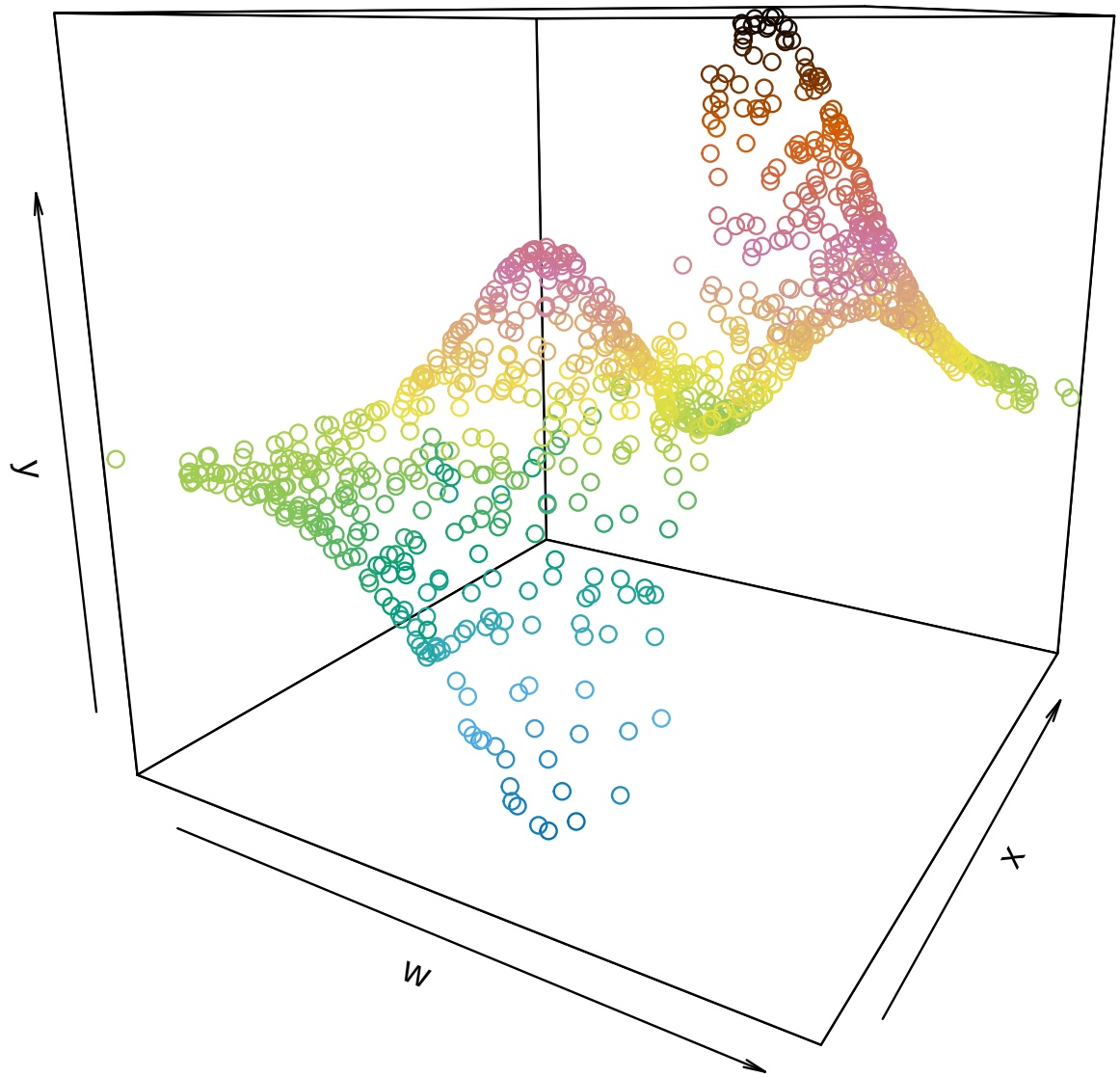
$$x \sim U\left(-\frac{5}{2}, \frac{5}{2}\right)$$

$$w \leftarrow \frac{1}{2}x + \frac{1}{2}\mathcal{N}(0,1)$$

$$y \leftarrow 3(1-w)^2 \exp\left(-w^2 - (1+x)^2\right) - 10\left(\frac{1}{5}w - w^3 - x^5\right)$$
$$\times \exp\left(-w^2 - x^2\right) - \frac{1}{3}\exp\left(-(1+w)^2 - x^2\right)$$

```
# Simulation
n = 1000
x = 5*runif(n) - 2.5
w = 0.5*x + 0.5*rnorm(n)
g = expression((3*(1-w)^2)*exp(-(w^2)-(x+1)^2)-
               10*(w/5-w^3-x^5)*exp(-w^2-x^2)-1/3*exp(-(w+1)^2-x^2))
y = eval(g)
```

Simulation 6



Simulation 6

Here the treatment effect is both heterogeneous and varies with x, and non-constant in w:

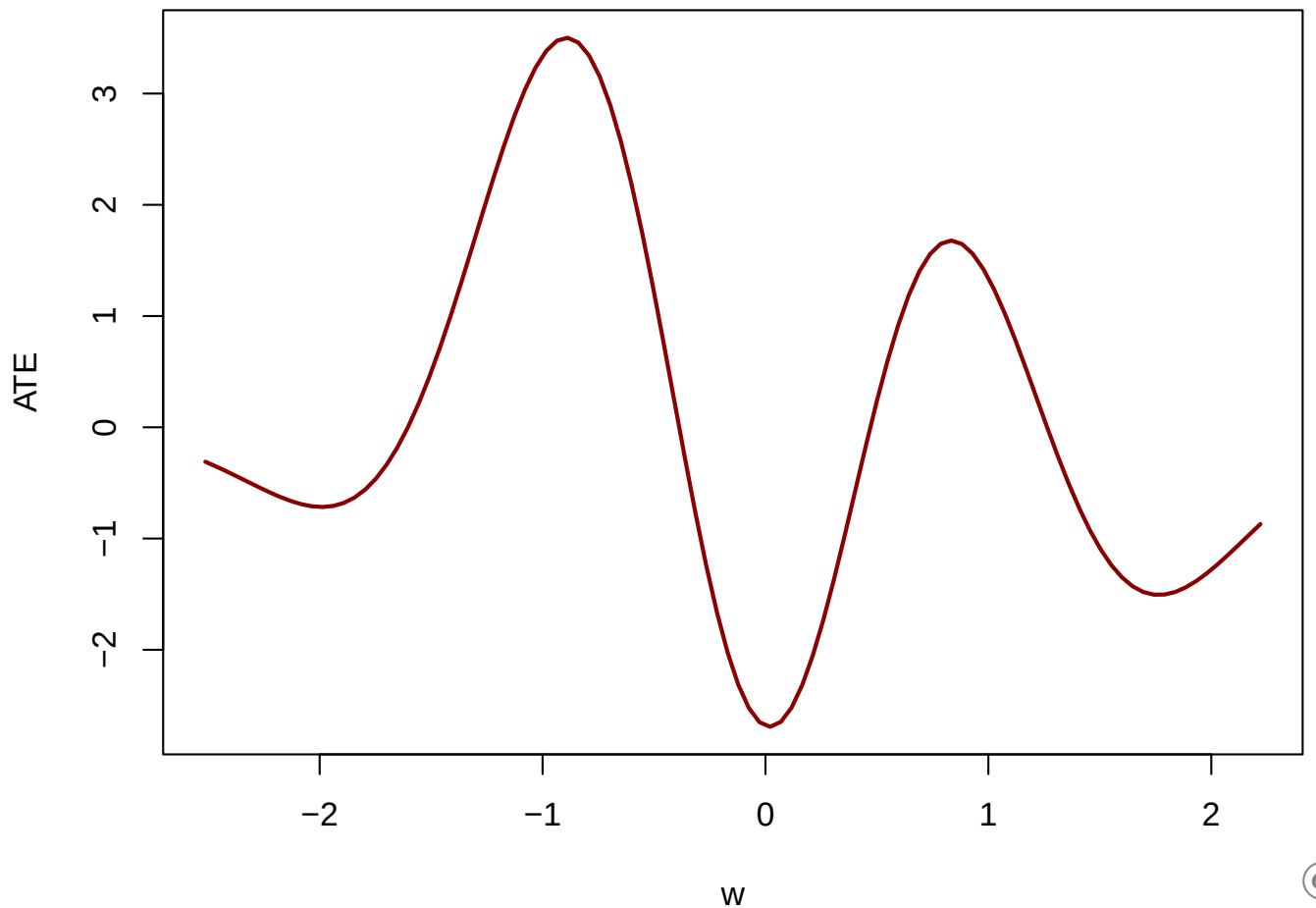
$$\begin{aligned}\tau_{\text{ATE}}(w) &= \int \frac{\partial \mathbb{E}[y|w,x]}{\partial w} p(x)\, dx \\ &\approx \frac{1}{N}\sum_i \frac{\partial \mathbb{E}[y_i|\, w, x_i]}{\partial w}\end{aligned}$$

Alternatively,

$$\begin{aligned}\tau_{\text{ATE}}(w) &= \frac{d}{dw}\int \mathbb{E}[y|w,x]\, p(x)\, dx \\ &\approx \frac{d}{dw}\left(\frac{1}{N}\sum_i \mathbb{E}[y_i|\, w, x_i]\right)\end{aligned}$$

Simulation 6

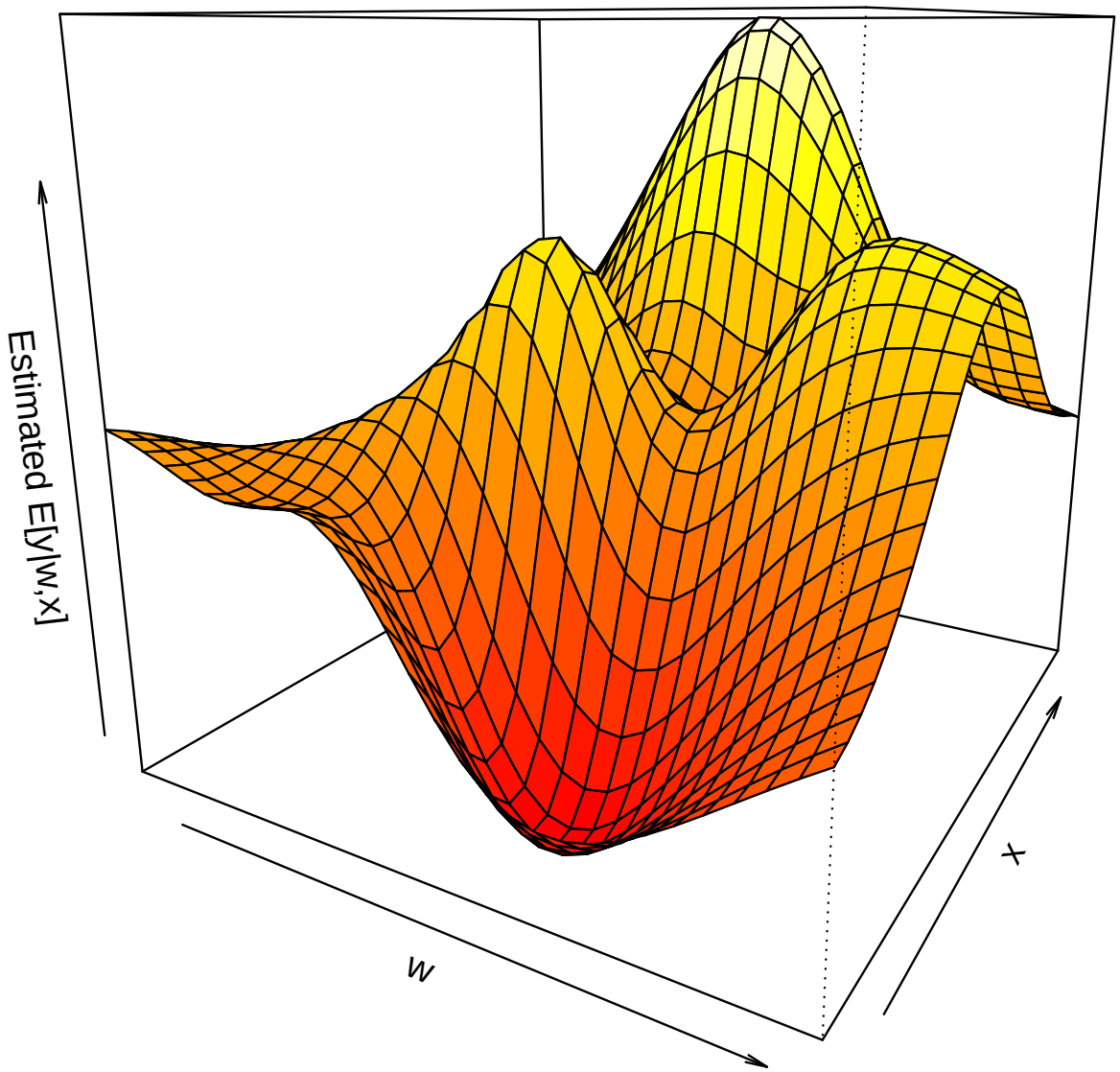
```
# ATE
dw = D(g,"w") # partial derivative w.r.t. w
wgrid = seq(min(w),max(w),length.out=100)
ate = sapply(wgrid, function(a) mean(eval(dw,envir=list(w=a,x=x))))
```



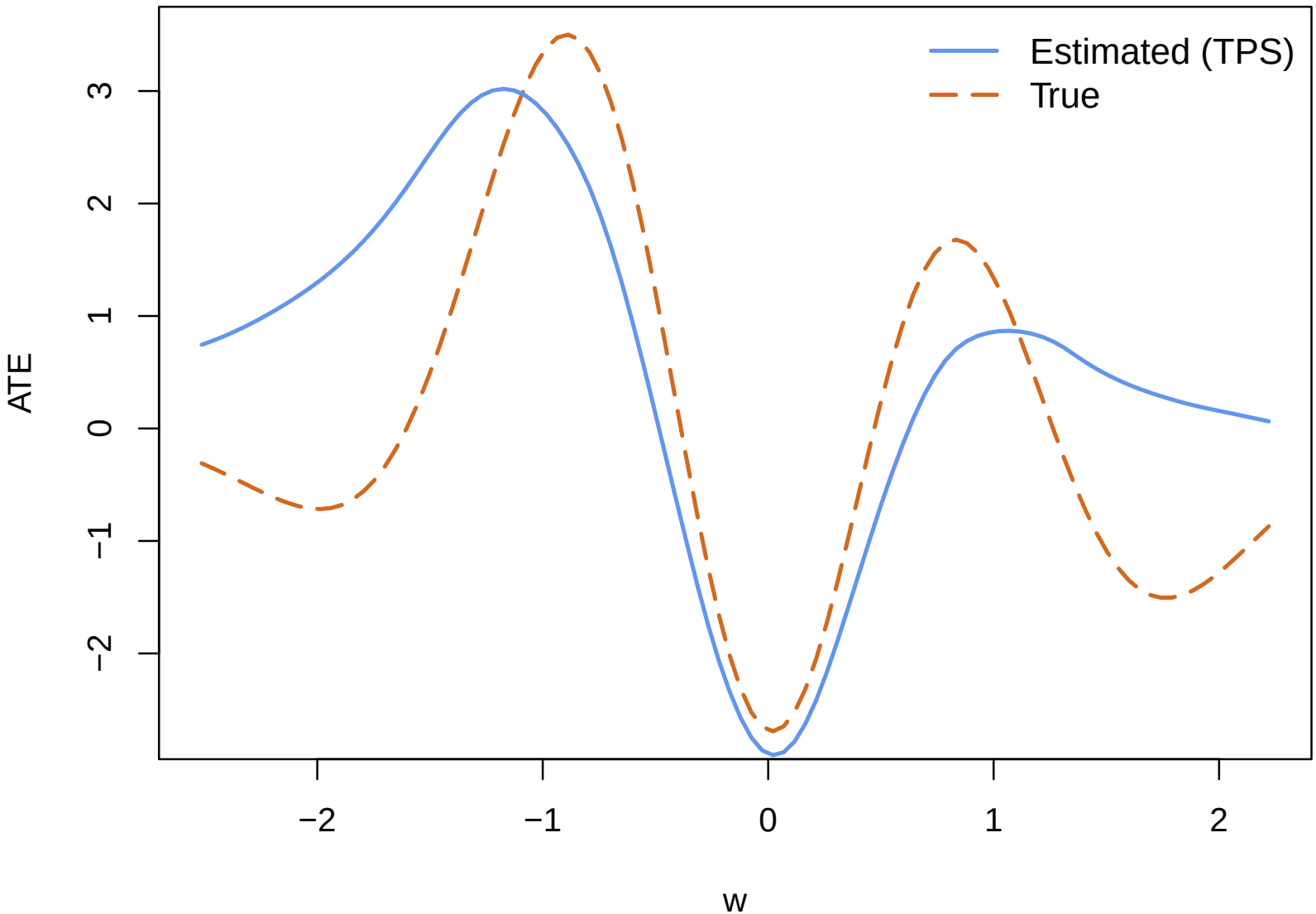
Simulation 6

```
######################
# Thin-plate Spline #
######################
require(mgcv)
fit = gam(y ~ s(w,x),family=gaussian)

# Estimated ATE
# here we first compute E[y|do(w)] = Integrate(E[y|w,x]) over x
# then we take the derivative of E[y|do(w)] to obtain ATE
require(numDeriv)
cef = function(a) mean(predict(fit,data.frame(w=a,x=x)))
atehat = sapply(wgrid,function(a) grad(cef,a))
```

Simulation 6



Simulation 6



Simulation 6

To choose the best statistical model for $\mathbb{E}[y|w,x]$, we can split our data set into training and test sets and perform model selection.

```
# Create Training and Test Sets
require(caret)
train = createDataPartition(y,p=0.5,list=F)
data = data.frame(w=w,x=x,y=y)
data_train = data[train,]
data_test = data[-train,]
```

Simulation 6

```
######################
# Thin-plate Spline #
######################
require(mgcv)
fit.tps = gam(y ~ s(w,x),data_train,family=gaussian)

# test err
yhat = predict(fit.tps,data_test)
mean((data_test$y - yhat)^2)

## [1] 0.05288992
```

Simulation 6

```
############################
# Random Forest (Bagging) #
############################
require(randomForest)
fit.rf = randomForest(y~.,data_train,mtry=2)

# test err
yhat = predict(fit.rf,data_test)
mean((data_test$y - yhat)^2)

## [1] 0.1065462
```

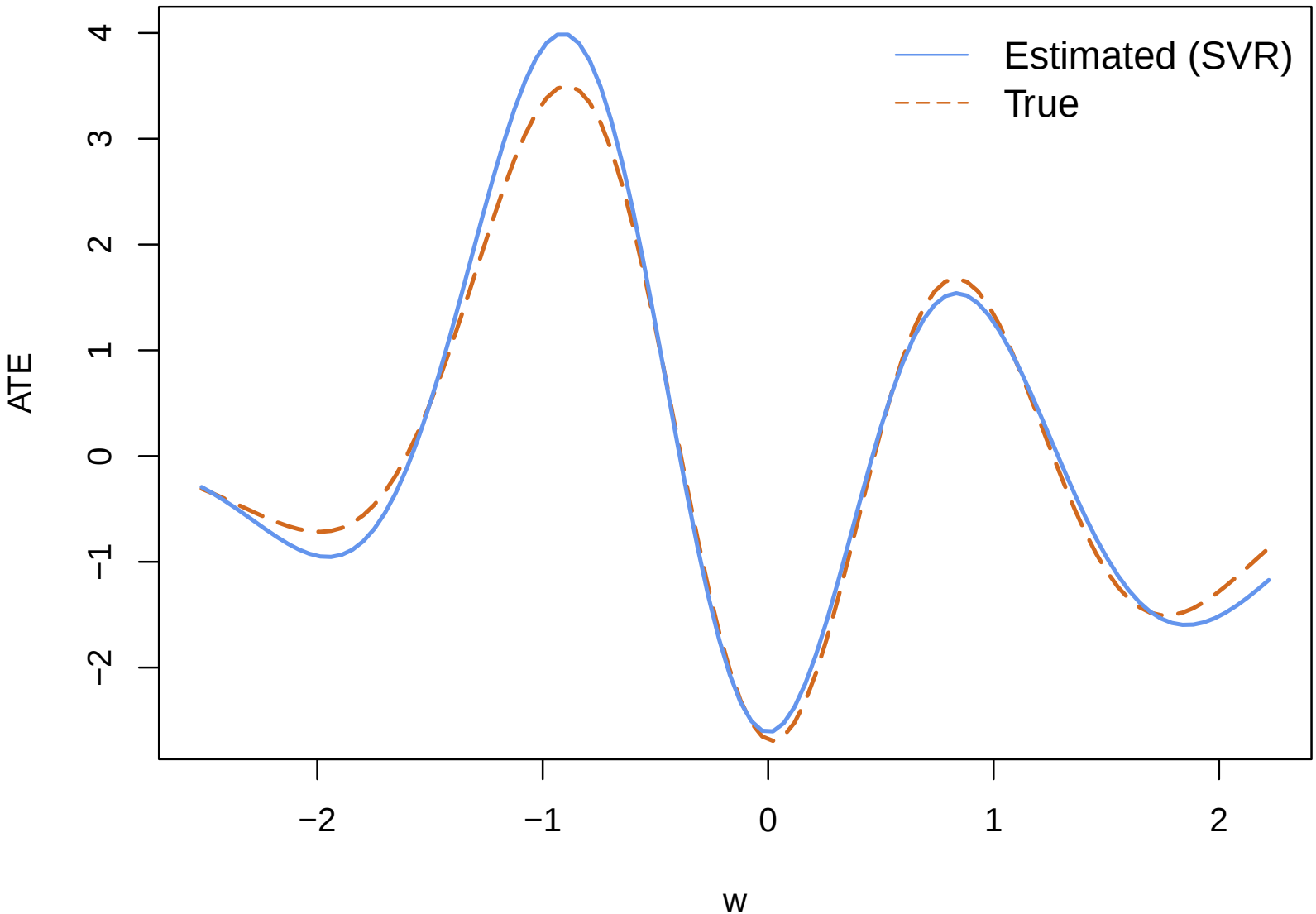
Simulation 6

```
##############################
# Support Vector Regression #
##############################
require(e1071)
fit.svm = svm(y~.,data_train,kernel="radial",
              cost=1e3,gamma=1) # tuning parameters chosen by cv

# test err
yhat = predict(fit.svm,data_test)
mean((data_test$y - yhat)^2)

## [1] 0.02468993
```

Simulation 6

```
###############
# Best Model #
###############
# Fit the best model (here: svr) on the entire data set
# note: the "test set" here is technically still a validation set,
#       since we are using it to select the best model.
#       we can then re-fit the selected model on the combined data
fit = update(fit.svm,data=data)

# Estimated ATE
cef = function(a) mean(predict(fit,data.frame(w=a,x=data$x)))
atehat = sapply(wgrid,function(a) grad(cef,a))
```

Simulation 6

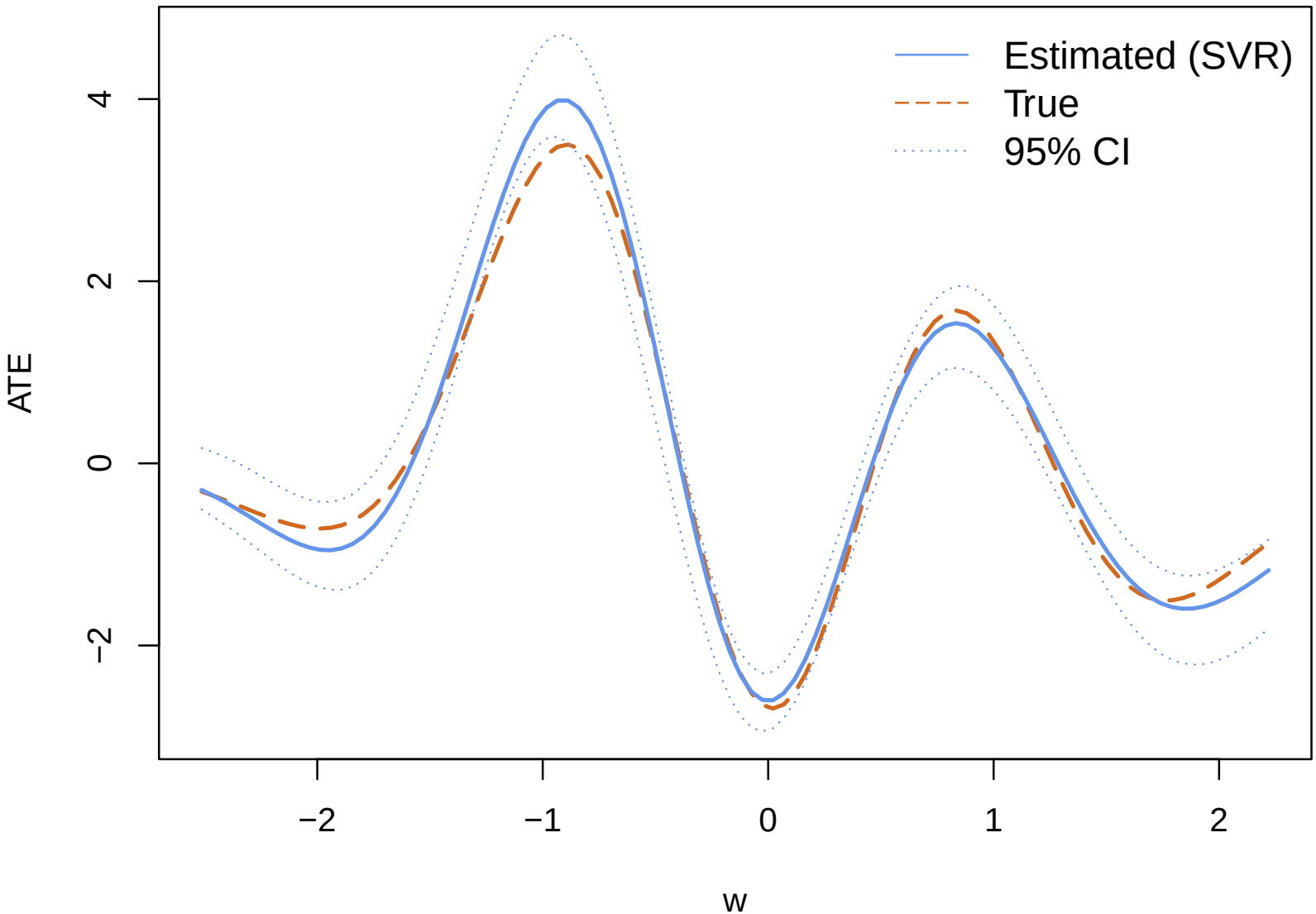


Simulation 6

We can use bootstrap to obtain standard errors and confidence intervals for estimated ATEs.

```
# Function to calculate ATE
calcATE = function(data,index){
  data_i = data[index,]
  fit_i = update(fit,data=data_i)
  cef_i = function(a) mean(predict(fit_i,data.frame(w=a,x=data_i$x)))
  atehat_i = sapply(wgrid,function(a) grad(cef_i,a))
  return(atehat_i)
}

# Bootstrap
require(boot)
B = 1000 # number of bootstrap samples
bootATE = boot(data,calcATE,R=B,parallel="multicore") # use parallel

# Confidence Intervals
ateci = sapply(1:length(wgrid),
        function(i) boot.ci(bootATE,type="norm",index=i)$normal[2:3])
```

Simulation 6



Matching

- Matching is another method for controlling confounders. The goal of matching is to construct a new sample in which the confounding variables have the same distribution conditional on each value of the treatment variable.

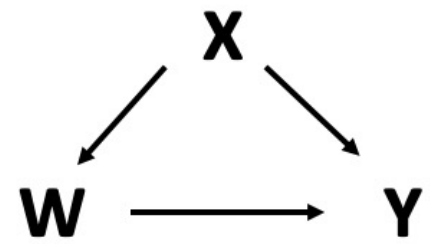
Matching

- We are interested in the causal effect of w on y.
- In observational settings, w and y share a common cause x.
- In a randomized experiment, by randomly assigning values to w, x is no longer a cause of w. x and w becomes *independent*.

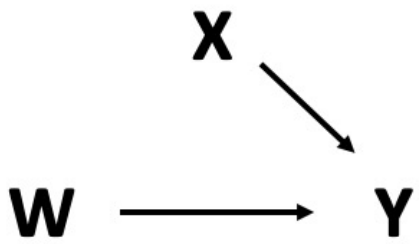
$$p(x|w) = p(x)$$

In this case, we say x is **balanced** across treatment groups.

Matching



Observational Study



Randomized Trial

Matching

- In randomized trials, **covariate balance** – the balance of x across values of w – is achieved at the design phase.
- **Matching** is a method that attempts to achieve covariate balance in observational studies, thereby making them resemble randomized trials.

Matching

	Age (x)	Treatment (w)	Outcome (y)
Zeus	old	1	10
Poseidon	old	1	12
Hades	old	1	11
Athena	young	1	5
Apollo	young	1	4
Hera	old	0	5
Ares	young	0	-2
Aphrodite	young	0	0
Hermes	young	0	1
Odin	old	0	7
Frigg	old	0	6
Thor	young	0	-1
Loki	young	0	2
Balder	young	0	-3

Matching

- Older people are more likely to get the treatment. Equivalently, the treatment group contains more old people than the control group.
- If we believe that age affects outcome, then age is a confounder that needs to be controlled.
- By matching treated individuals with untreated individuals *of the same age*, we can create a new sample in which the treatment and the control groups have the same age distribution, thereby eliminating the confounding effect of age.
- The constructed **matched sample** is a subsample of the original and can be treated *as if* generated by a randomized trial.

Matching

	Age (x)	Treatment (w)	Outcome (y)
Zeus	old	1	10
Poseidon	old	1	12
Hades	old	1	11
Athena	young	1	5
Apollo	young	1	4
Hera	old	0	5
Odin	old	0	7
Frigg	old	0	6
Aphrodite	young	0	0
Thor	young	0	-1

Matching

Based on the matched sample, let $\mathcal{N}_1$ denote the set of treated individuals and $\mathcal{N}_0$ denote the set of untreated individuals. Let N be the size of the matched sample. Then

$$\begin{aligned}\tau_{\text{ATT}} &= \frac{1}{N}\left(\sum_{i\in\mathcal{N}_1} y_i - \sum_{i\in\mathcal{N}_0} y_i\right) \\ &= \frac{1}{5}\times[(10+12+11+5+4)-(5+7+6-1)] = 5\end{aligned}$$

τ_{ATT} is an estimate of the *average treatment effect on the treated* (*ATT*)[9,10].

[9] τ_{ATT} is the sample ATE in the matched sample and the sample ATT in the original sample. It is an estimate of the population ATT in the study population – the population from which the original sample is drawn.

[10] Compare this to the naive calculation of taking the average of the treated minus the average of the untreated in the original sample, which gives 6.73.

Matching

- The matched sample can be considered as drawn from a counterfactual population – one in which the treatment and the control groups have the same distribution of covariates.
- Recall that causal effects are *population-specific*: here the causal effect we compute is ATE in the counterfactual population and the ATT in the original study population.
- Often one chooses the group with fewer individuals and uses the other group to find their matches. The chosen group defines the target population on which the causal effect is being computed.

Matching

- When matching cannot be exact, we match based on smallest distance – this is called **nearest neighbor matching**[11].
- When computing the ATT, for each individual i in the treatment group, we find the j in the control group whose x_j is closest to x_i.

[11] In this lecture, we focus on nearest neighbor matching. For other types of matching methods, see footnote 19.

Matching

	Age (millions of years)	Treatment	Outcome
Zeus	98	1	10
Poseidon	93	1	12
Hades	85	1	11
Athena	35	1	5
Apollo	45	1	4
Hera	87	0	5
Ares	40	0	-2
Aphrodite	28	0	0
Hermes	33	0	1
Odin	100	0	7
Frigg	90	0	6
Thor	44	0	-1
Loki	30	0	2
Balder	25	0	-3

Matching

	Age (millions of years)	Treatment	Outcome
Zeus	98	1	10
Poseidon	93	1	12
Hades	85	1	11
Athena	35	1	5
Apollo	45	1	4
Odin	100	0	7
Frigg	90	0	6
Hera	87	0	5
Hermes	33	0	1
Thor	44	0	-1

$$\tau_{\text{ATT}} = \frac{1}{5} \times [(10 + 12 + 11 + 5 + 4) - (7 + 6 + 5 + 1 - 1)] = 4.8$$

Matching

In general, let $x = (x_1, \ldots, x_p)$. We can measure the distance between two points x and x' using the **Mahalanobis distance**:

$$\|x - x'\|_{\mathcal{M}} = (x - x')' \Sigma_x^{-1} (x - x')$$

, where Σ_x is the the sample covariance matrix[12].

[12] When x and x' are independent and standardized to have standard deviation one, the Mahalanobis distance is equivalent to the Euclidean distance.

Matching

- The matching estimator can be viewed as imputing the missing potential outcomes using the outcomes of nearest neighbors of the opposite treatment group.
- For example, suppose we observe $(w_i = 1, x_i, y_i)$. Then $y_i = \mathcal{Y}_i(1)$. To impute $\mathcal{Y}_i(0)$, we find individuals in the control group whose x_j is closest to x_i.
- Let $\Omega_K(i)$ denote the set of K nearest neighbors of i with *opposite* treatment. A **K:1 matching estimator** imputes $\mathcal{Y}_i(0)$ as

$$\widehat{\mathcal{Y}}_i(0) = \frac{1}{K} \sum_{j \in \Omega_K(i)} y_j$$

Matching

Given binary treatment $w \in \{0, 1\}$, let $\mathcal{N}_1 \doteq \{i : w_i = 1\}$ denote the set of treated individuals and let N_1 be the size of $\mathcal{N}_1$. Let $\mathcal{N}_0 \doteq \{i : w_i = 0\}$ denote the set of untreated individuals and let N_0 be the size of $\mathcal{N}_0$. $N = N_0 + N_1$ is the size of the observed sample.

Then for $a \in \{0, 1\}$, we have:

$$\widehat{\mathcal{Y}}_i(a) = \begin{cases} y_i & w_i = a \\ \frac{1}{K} \sum_{j \in \Omega_K(i)} y_j & w_i = 1 - a \end{cases} \tag{13}$$

Matching

The matching estimators for the ATE and the ATT are given by

$$\widehat{\tau}_{\mathsf{ATT}} = \frac{1}{N_1} \sum_{i \in \mathcal{N}_1} \Big(y_i - \widehat{\mathcal{Y}}_i(0) \Big) \tag{14}$$

$$\widehat{\tau}_{\mathsf{ATE}} = \frac{1}{N} \sum_{i=1}^{N} \Big(\widehat{\mathcal{Y}}_i(1) - \widehat{\mathcal{Y}}_i(0) \Big) \tag{15}$$

Matching

- To compute the ATE, we would need to construct *two* matched samples.
  - In the first sample, we match control units to treated units to obtain $\left(\widehat{\mathcal{Y}}_i(1), \widehat{\mathcal{Y}}_i(0)\right)$ for $i \in \mathcal{N}_1$[13].
  - In the second sample, we match treated units to control units to obtain $\left(\widehat{\mathcal{Y}}_i(1), \widehat{\mathcal{Y}}_i(0)\right)$ for $i \in \mathcal{N}_0$[14].
  - We then calculate the ATE according to (15)[15].

[13] according to (13).

[14] When matching a group with fewer units to a group with more units, matching *with replacement* is necessary.

[15] Equivalently, we obtain the ATT from the first matched sample and the ATC from the second matched sample. $\text{ATE} = \text{ATT} \times p(w = 1) + \text{ATC} \times p(w = 0)$.

Matching

16

[16] Given multi-valued treatment, $w \in \{1, \ldots, A\}$, $A \geq 3$, for *each* treatment value a, we would match individuals who receive a - the "treated with a" sample ($\mathcal{N}_a$) – to individuals who receive other treatments – the "untreated with a" sample ($\mathcal{N}_{-a}$), to obtain $\widehat{\mathcal{Y}}_i(a)$ for $i \in \mathcal{N}_{-a}$. Combined with $y_i = \mathcal{Y}_i(a)$ for $i \in \mathcal{N}_a$, we obtain $\widehat{\mathcal{Y}}_i(a)$ for the entire sample. Doing this for each treatment value would allow us to obtain $\widehat{\mathcal{Y}}_i(1), \ldots, \widehat{\mathcal{Y}}_i(A)$ $\forall i$.

Matching as Nonparametric Regression

(14) and (15) can be equivalently written as[17]

$$\widehat{\tau}_{\text{ATT}} = \frac{1}{N_1} \sum_{i \in \mathcal{N}_1} \left(y_i - \widehat{\mu}_0 \left(x_i \right) \right) \tag{16}$$

$$\widehat{\tau}_{\text{ATE}} = \frac{1}{N} \sum_{i=1}^{N} \left(\widehat{\mu}_1 \left(x_i \right) - \widehat{\mu}_0 \left(x_i \right) \right) \tag{17}$$

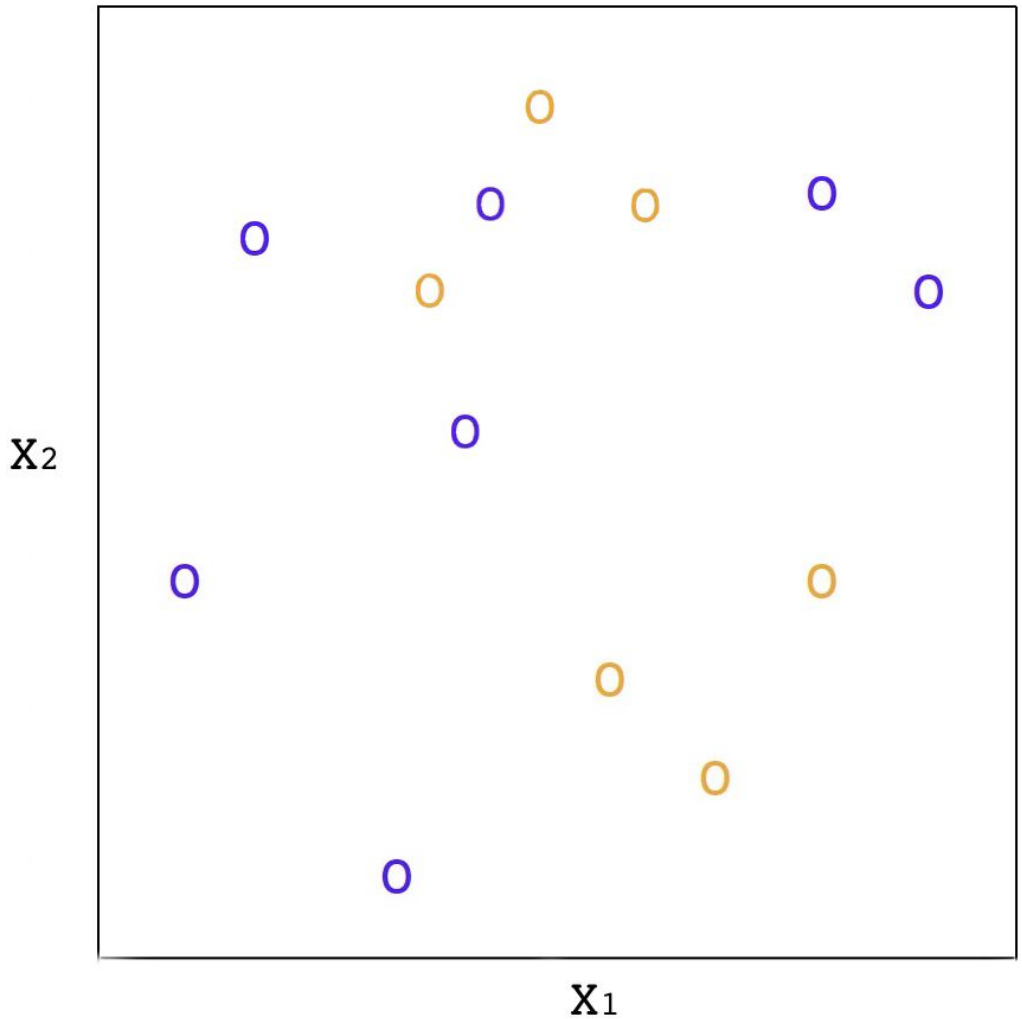
Hence, the matching estimator can alternatively be viewed as nearest neighbor estimation of the conditional regression function $\mu_a \left(x \right) = \mathbb{E} \left[y | w = a, x \right]$ using the sample $\mathcal{N}_a = \{ i : w_i = a \}$[18].

[17] Recall that $\mu_a \left(x_i \right) = \mathbb{E} \left[\mathcal{Y}_i \left(a \right) \right]$. Hence $\widehat{\mu}_a \left(x_i \right) = \widehat{\mathcal{Y}}_i \left(a \right)$.

[18] To be more precise, we can think of an $m : 1$ matching estimator as estimating $\mu_a \left(x_i \right)$ with a KNN model where $K = 1$ if $w_i = a$ and $K = m$ if $w_i = 1 - a$, i.e.,

$$\widehat{\mu}_a \left(x_i \right) = \begin{cases} y_i & w_i = a \\ \frac{1}{m} \sum_{j \in \Omega_m(i)} y_j & w_i = 1 - a \end{cases}$$

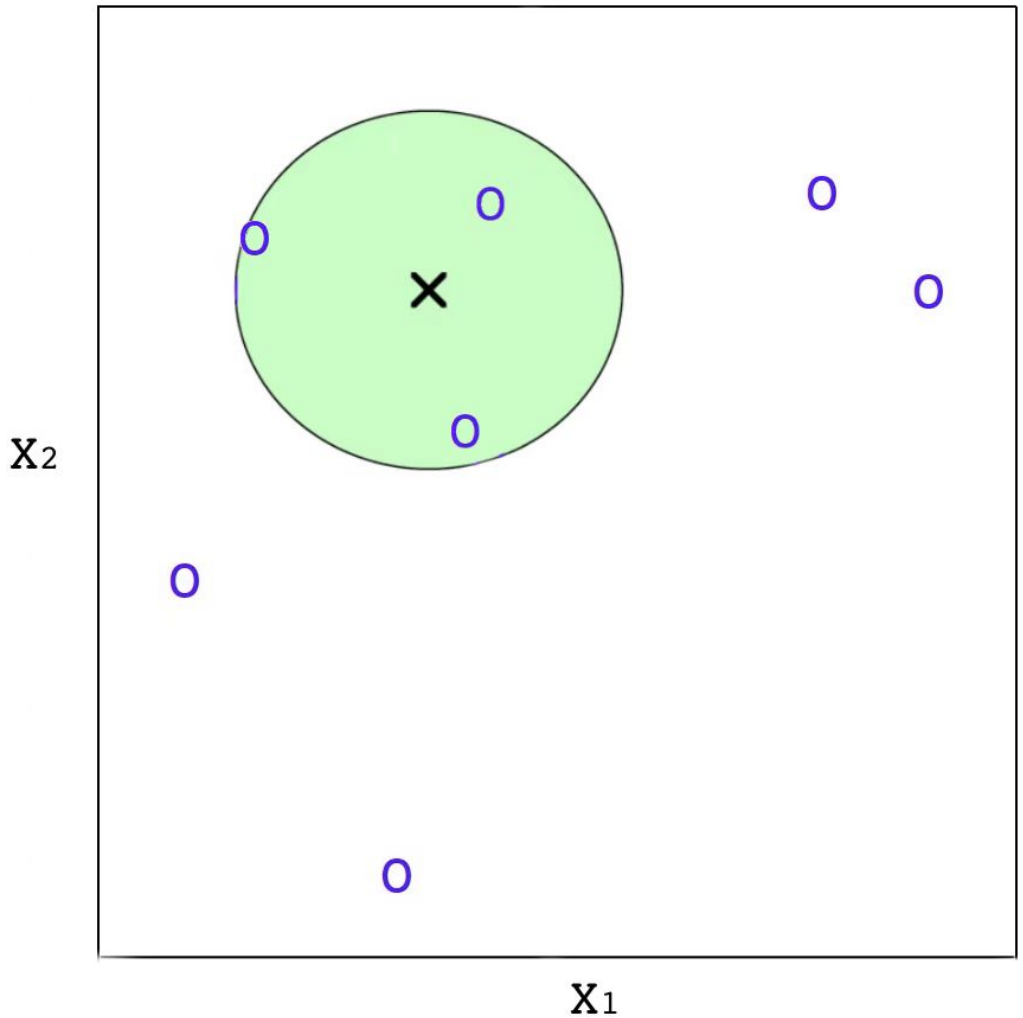
Matching as Nonparametric Regression



Observed Sample

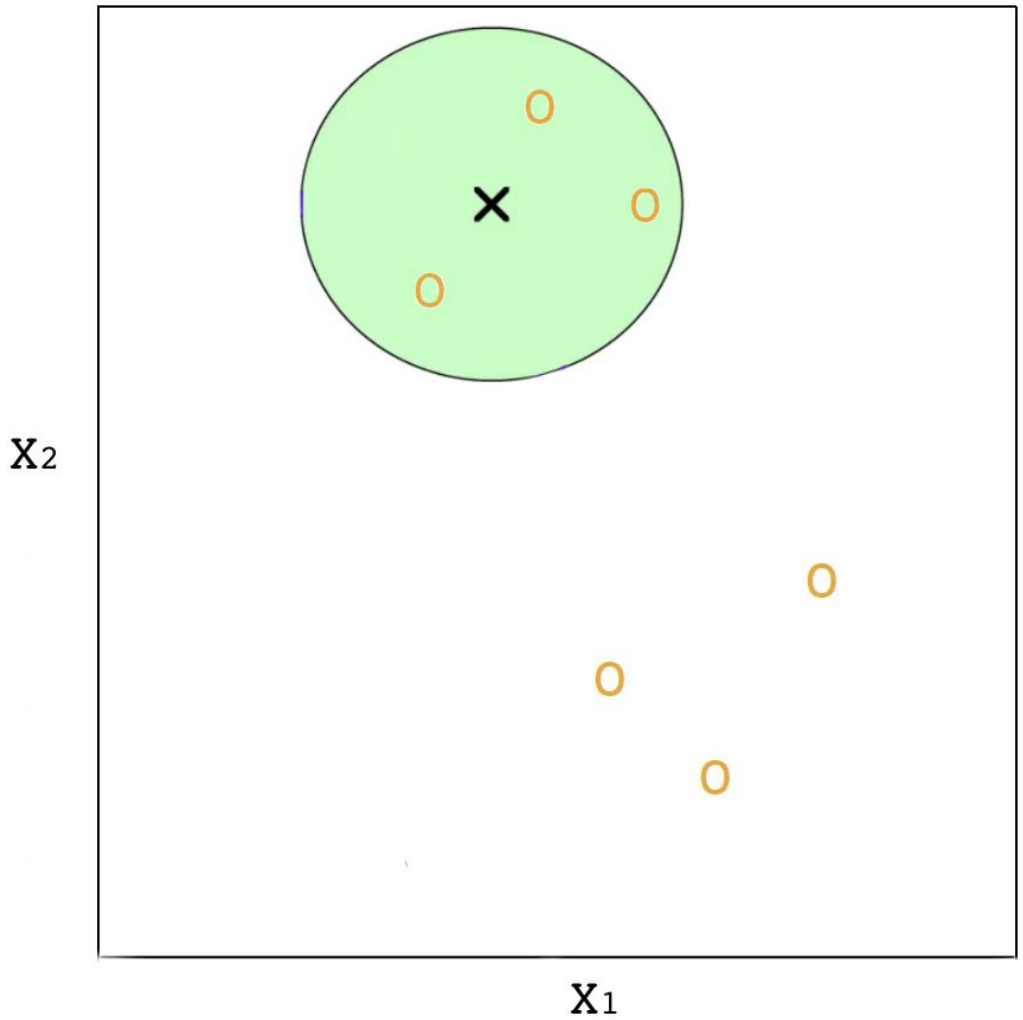
Blue: $w_i = 0$. Orange: $w_i = 1$.

Matching as Nonparametric Regression



$\widehat{\mu}_0(x)$ based on 3 : 1 matching

Matching as Nonparametric Regression



$\widehat{\mu}_1(x)$ based on 3 : 1 matching

Matching as Nonparametric Regression

- Traditionally, the regression approach to causal inference relies on linear models to estimate $\mu_a(x)$. Thus, one difference between regression and matching is parametric vs. nonparametric estimation.

- If, however, we view the regression approach as using any appropriate statistical model – parametric or nonparametric – to fit $\mu_a(x)$, then matching can be considered a type of nearest neighbor regression estimator.

Matching as Covariate Imbalance Minimization

- There is, however, a key difference between the regression approach and the matching approach: the goal of matching is to achieve covariate balance rather than estimating $\mathbb{E}[y|w, x]$ itself.
- In matching, controlling for confounders is achieved at the design phase – *without* looking at the outcome, i.e., y is not used to *supervise* the matching process.
  - Thus, if we adaptively choose the number of nearest neighbors used in the matching process – as we do in KNN – by computing cross-validated errors involving y, then we are technically doing regression rather than matching.
  - The matching approach to causal inference uses the outcome values only in computing treatment effects based on the matched sample.

Matching as Covariate Imbalance Minimization

- Fundamentally, regression and matching have different loss functions: the goal of matching is to minimize the distance between $p(x|w=0)$ and $p(x|w=1)$.
- In this perspective, any statistical procedure that attempts to do so is a matching procedure[19].

[19] In addition to nearest neighbor matching, popular matching methods include **subclassification** and **full matching**. More recent proposals include the use of adversarial networks to minimize the discrepancy between $p(x|w=0)$ and $p(x|w=1)$. See, e.g., Ozery-Flato et al. (2020).

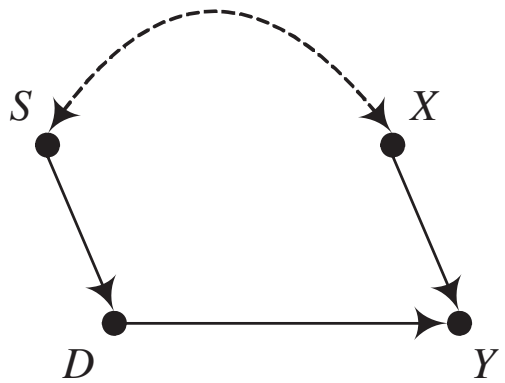
Matching as Covariate Imbalance Minimization

Conditioning to Balance vs. Conditioning to Adjust

- One may also think of the difference between regression and matching as that of *conditioning to adjust* vs. *conditioning to balance*.

- Matching is considered a technique to balance the determinants of the treament variable, while regression is considered a technique to adjust for other causes of the outcome.

Conditioning to Balance vs. Conditioning to Adjust



D: treatment; Y: outcome

Here either S or X satisfies the back-door criterion. Matching estimators often condition on S, while regression estimators often condition on X.

- Conditioning on S is considered a balancing conditioning strategy.
- Conditioning on X is considered an adjustment-for-other-causes conditioning strategy.
  - In this case, the goal is not to balance X but to partial out its effects on Y in order to isolate the net effect of D on Y.

Matching as Pre-processing

- After constructing a matched sample, the causal effect of interest can be computed as a simple difference in means[20].
- However, we can also fit statistical models of $\mathbb{E}[y|w,x]$ *on the matched sample* and compute causal effects based on the estimated conditional regression functions.
  - Doing so may further improve the performance of our estimate, particularly when the matching procedure fails to produce a well-balanced sample.
  - Causal effects computed this way – by combining matching with regression – have a doubly robust property: if either the matching procedure produces a well-balanced sample or the regression model is correct, the causal effect estimate will be consistent.

[20] See (14) and (15).

Matching as Pre-processing

In this perspective, matching can be thought of as a **pre-processing** step to be followed by an **outcome analysis** step[21].

- The pre-processing step aims to eliminate or reduce the relation between x and w[22].
- In the outcome analysis step, we compute difference in means or conduct further regression analysis. The modeling choice at this stage will be less consequential if matching is performed first.

[21] One therefore sometimes distinguishes between *matching* and *the matching estimator*, where matching refers only to the pre-processing step that produces matched samples with improved covariate balance, while the matching estimator estimates causal effects by conducting outcome analysis on matched samples.

[22] The outcome y is not involved in this step!

Implementation of Nearest Neighbor Matching

- For $K:1$ matching, the choice of K involves a bias-variance tradeoff.
  - When K is small (say, $K = 1$), bias is low and variance is high.
  - Larger K increases bias and decreases variance[23].

[23] This is the same tradeoff involved in KNN. We could adaptively choose K using techniques like cross-validation, but in that case our choice of K is guided not by how well we improve covariate balance, but by how well we predict the outcome y – we are doing regression rather than matching.

Implementation of Nearest Neighbor Matching

- Matching can be done *with* or *without* replacement.
  - When matching a group with fewer units to a group with more units, matching with replacement is necessary.
  - When we have enough units to match, whether to match with or without replacement involves a bias-variance tradeoff: matching with replacement results in lower bias but higher variance[24].

[24] Suppose we are matching control units to the treatment group. If a control unit is closer to multiple treated units, then this control unit can be used multiple times when we are matching with replacement. Doing so improves balance but increases variance because the resulting matches all depend on this one unit.

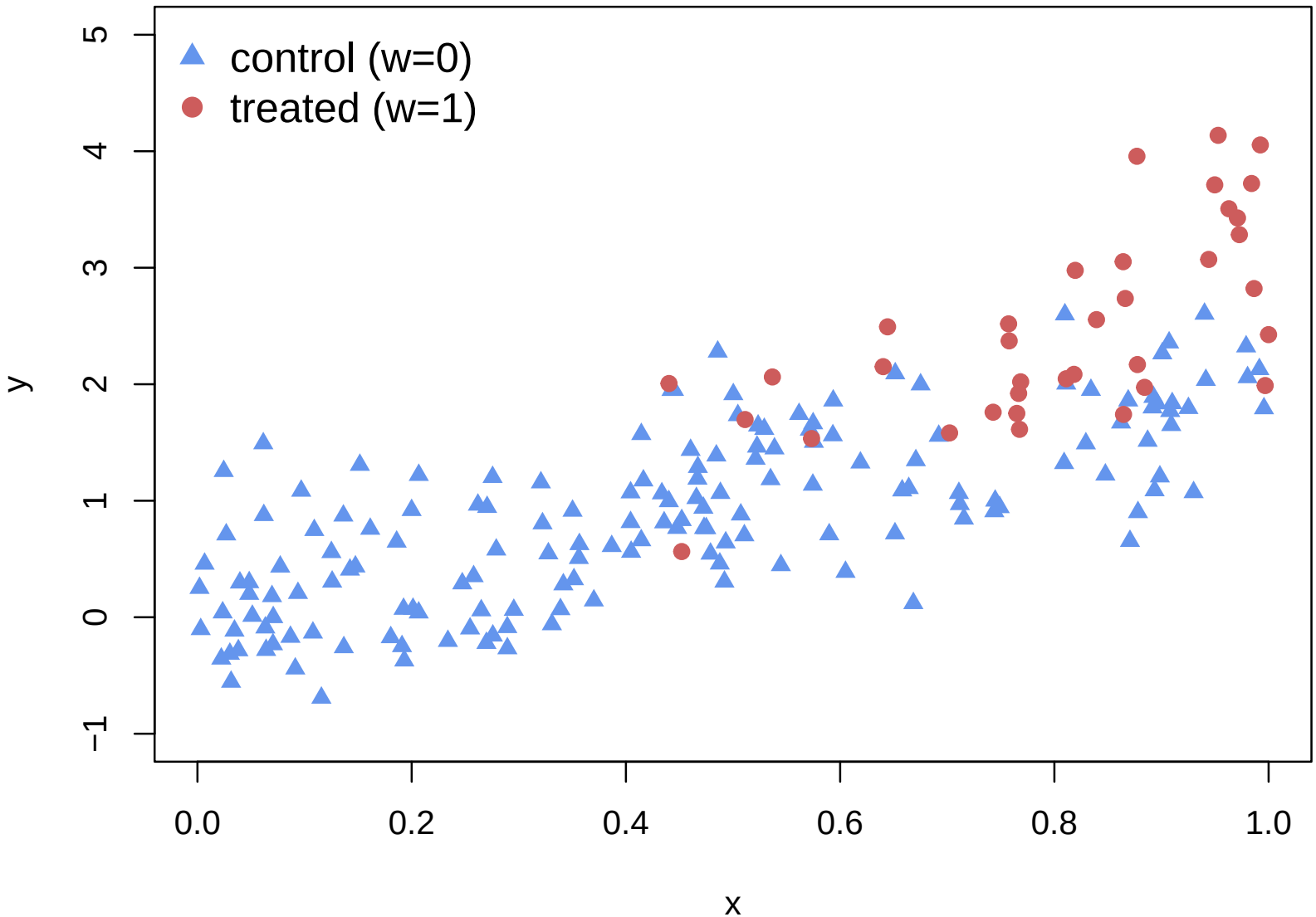
Implementation of Nearest Neighbor Matching

- Since the goal of matching is to improve covariate balance, a crucial part of a matching procedure is to compare the empirical covariate distributions in the two treatment groups after matching – this step is called assessing balance.

Simulation 7

```r
# Simulation
require(sigmoid)
n = 200
x = runif(n)
w = rbinom(n,1,sigmoid(5*x-5))
tau = 1 + 0.2*rnorm(n) # treatment effect
y = tau*w + 2*x + 0.5*rnorm(n)
data = data.frame(x=x,w=w,y=y)
```

Simulation 7



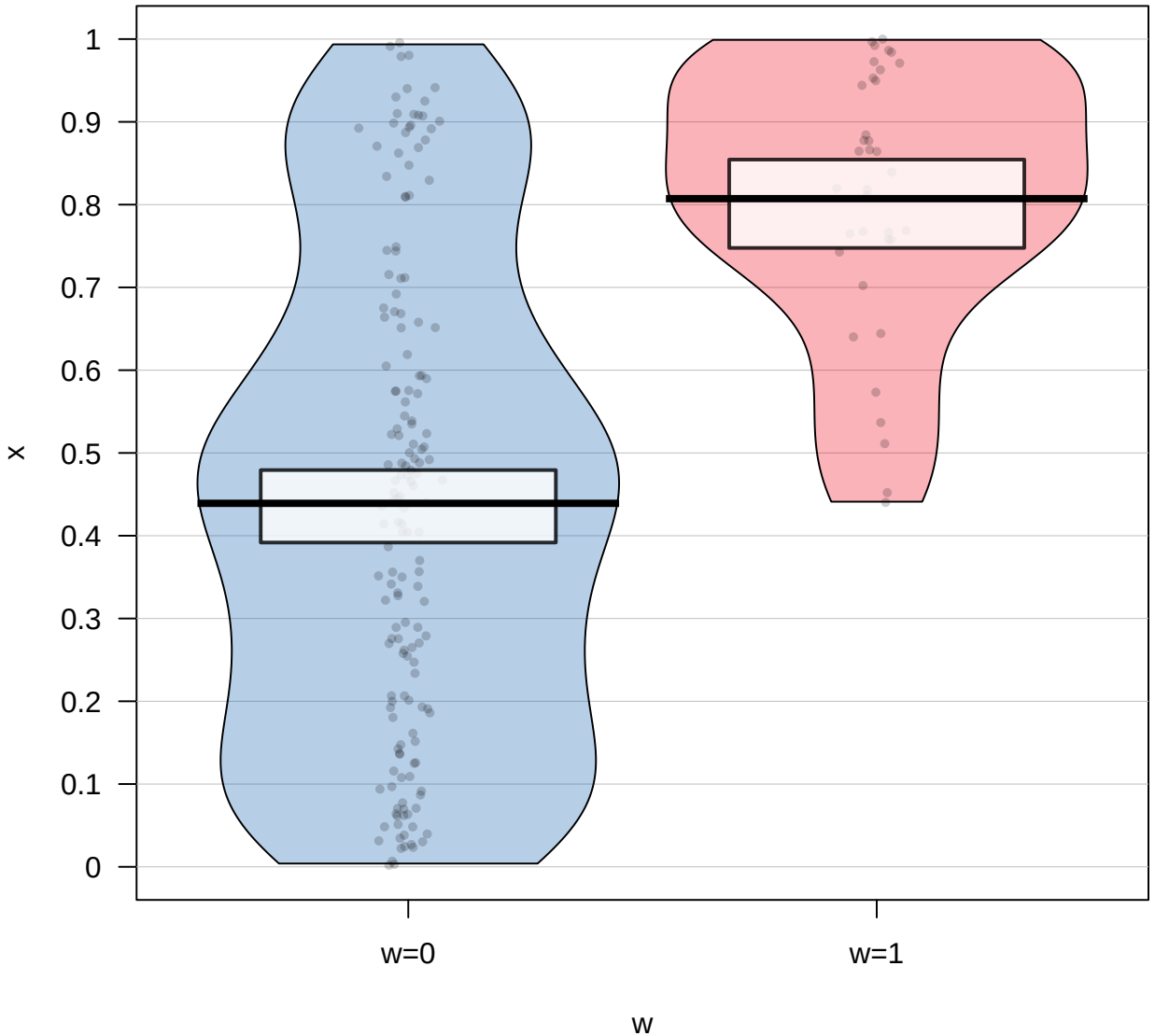
Simulation 7

Here the treatment effect does not vary with x. Hence $\tau_{\text{ATT}} = \tau_{\text{ATE}} = 1$. The distribution of x is unbalanced across treatment groups:

```
require(tableone)
CreateTableOne(data = data, vars = c("x","y"), strata = "w")
```

```
##                  Stratified by w
##                   0            1            p      test
##   n                 164           36
##   x (mean (SD)) 0.44 (0.29) 0.81 (0.16) <0.001
##   y (mean (SD)) 0.87 (0.75) 2.49 (0.84) <0.001
```

Simulation 7



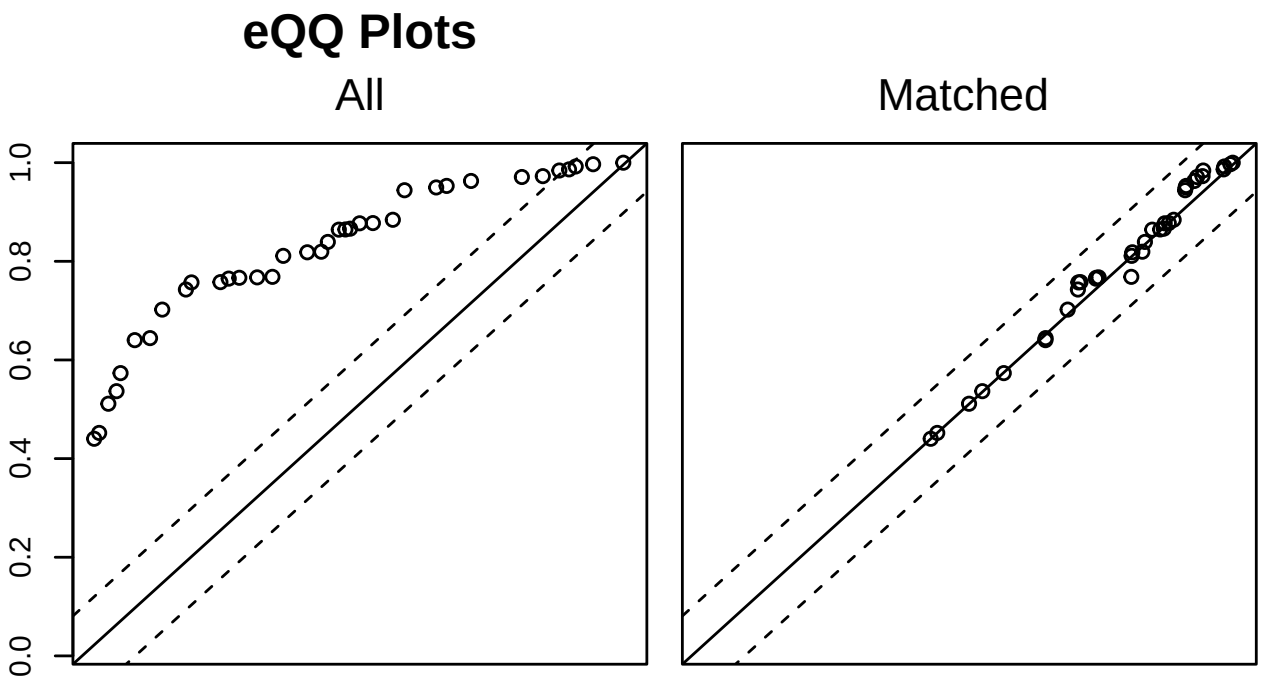
Simulation 7

```
##############################
# Nearest Neighbor Matching #
##############################
require(MatchIt)
m = matchit(w ~ x, data, distance = "mahalanobis") # 1-NN matching
```

Simulation 7

```
summary(m)

##
## Call:
## matchit(formula = w ~ x, data = data, distance = "mahalanobis")
##
## Summary of Balance for All Data:
##   Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
## x        0.8072        0.4391          2.3034     0.3062    0.3457   0.5989
##
##
## Summary of Balance for Matched Data:
##   Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
## x        0.8072        0.7945          0.0798     1.0886    0.0161   0.1944
##   Std. Pair Dist.
## x          0.1172
##
## Percent Balance Improvement:
##   Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
## x            96.5       92.8      95.3     67.5
##
## Sample Sizes:
##           Control Treated
## All           164      36
## Matched        36      36
```

Simulation 7

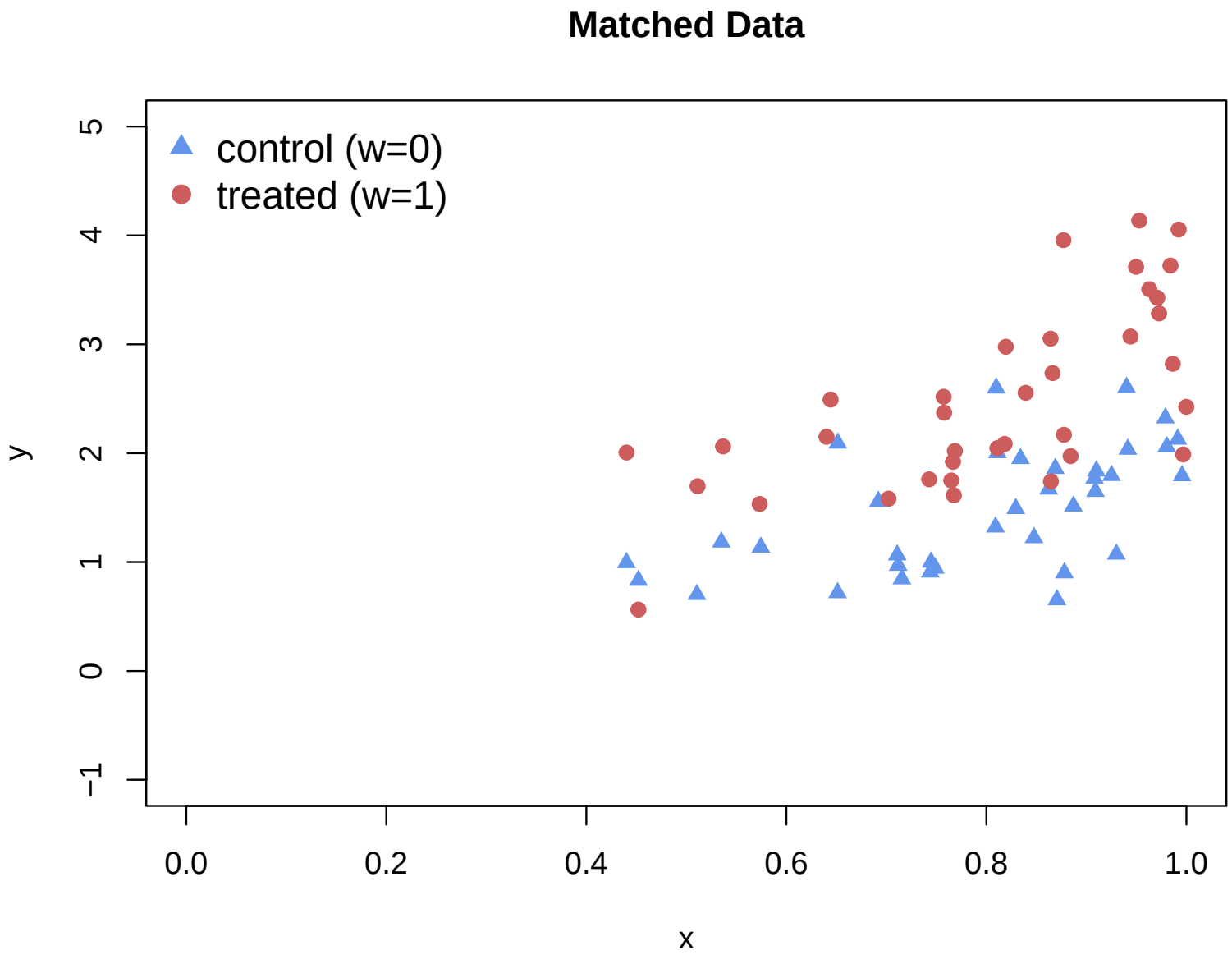


Simulation 7

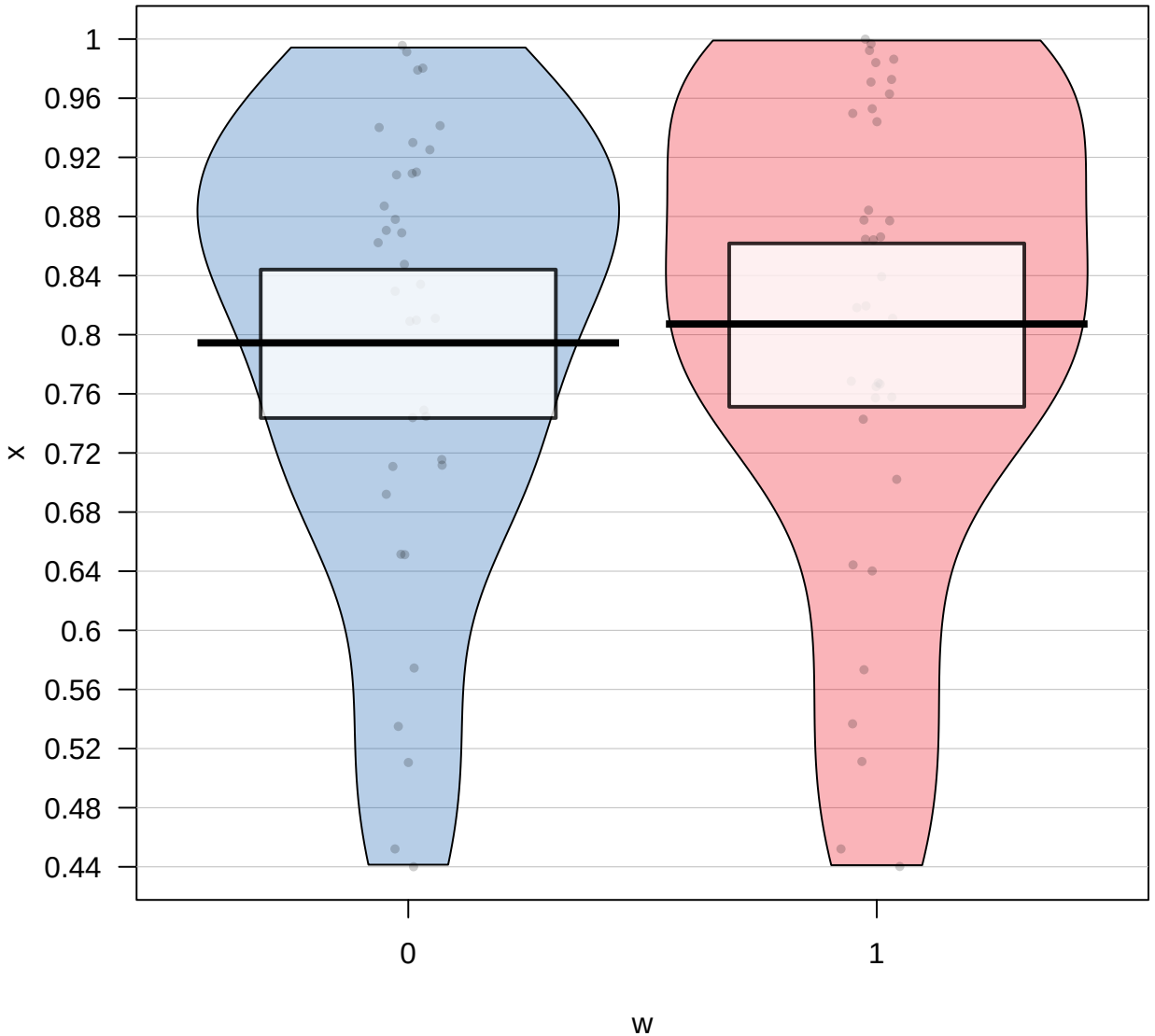
```
# create matched data set
m.data = match.data(m)

# Assessing balance
CreateTableOne(data = m.data, vars = c("x","y"), strata = "w")
```

```
##                  Stratified by w
##                   0            1            p      test
##   n                 36           36
##   x (mean (SD)) 0.79 (0.15) 0.81 (0.16)  0.730
##   y (mean (SD)) 1.48 (0.56) 2.49 (0.84) <0.001
```

Simulation 7



Simulation 7



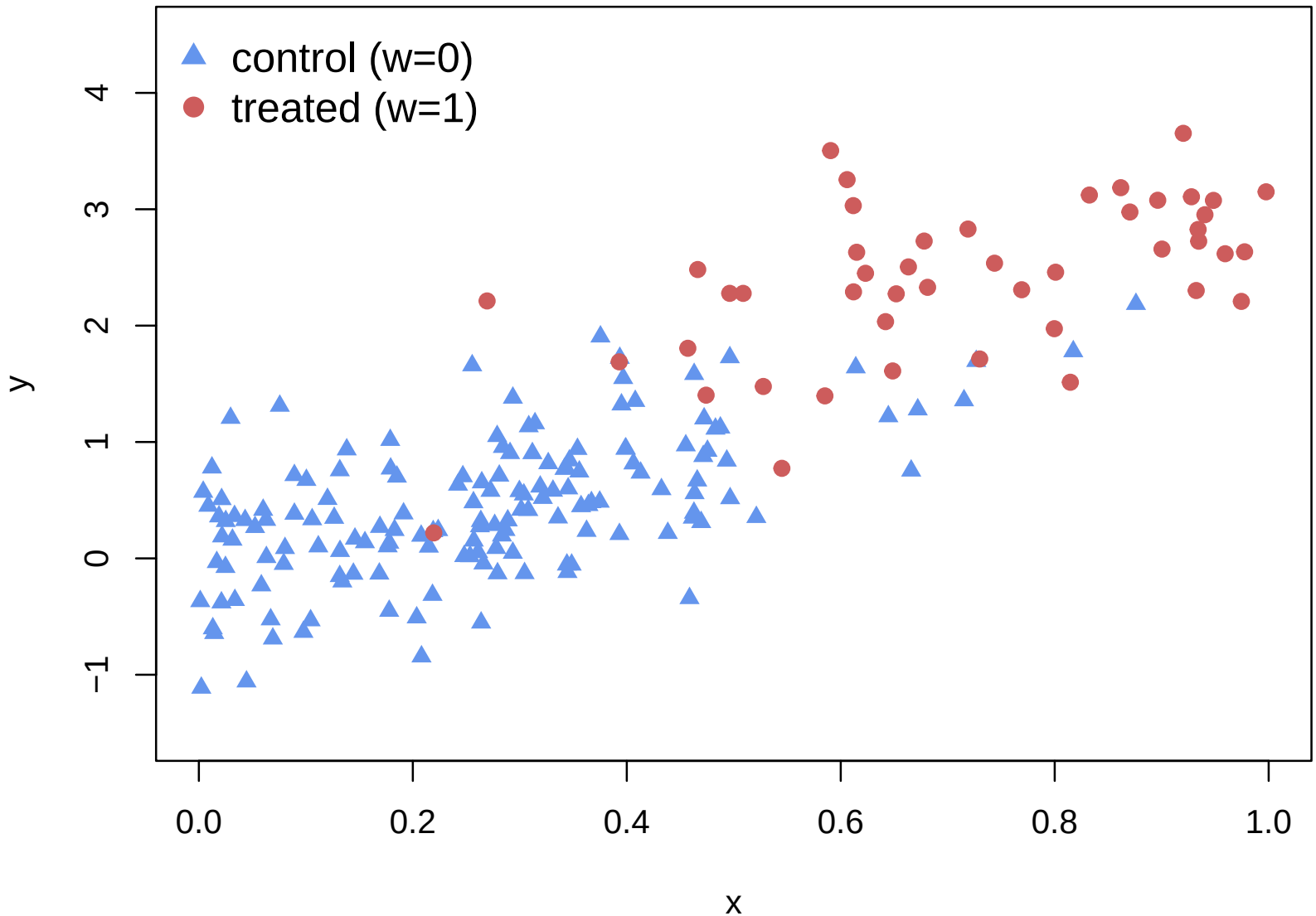
Simulation 7

```r
# ATT estimate based on matched ata
atthat = mean(m.data$y[m.data$w==1]) - mean(m.data$y[m.data$w==0])
atthat

## [1] 1.007687

# Alternatively, run regression of y on w on the matched data
require(AER)
fit = lm(y ~ w, data = m.data)
coeftest(fit)

##
## t test of coefficients:
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  1.47804    0.11832 12.4922 < 2.2e-16 ***
## w            1.00769    0.16733  6.0223 7.155e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

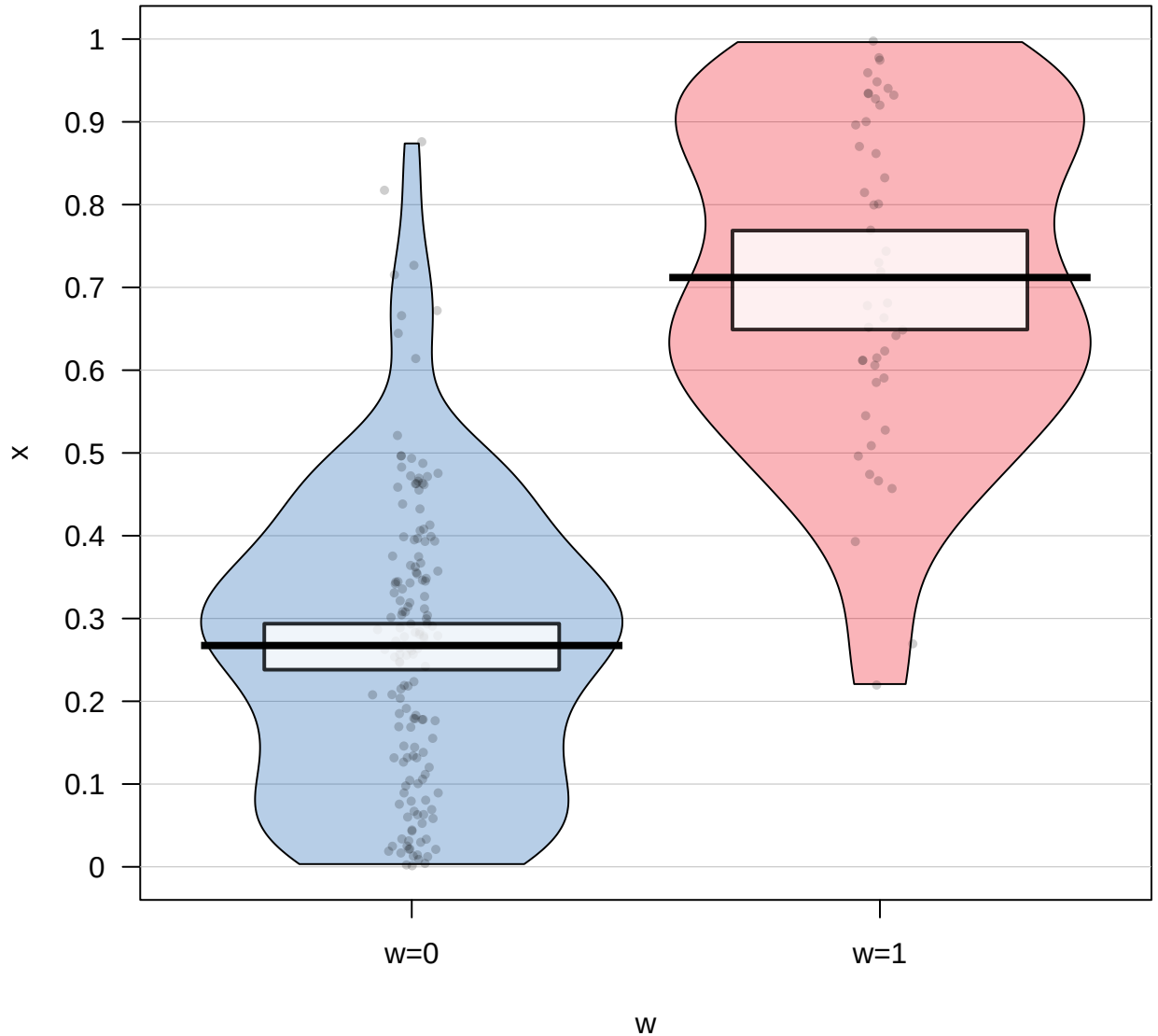
Simulation 8

```r
# Simulation
require(sigmoid)
n = 100
x = runif(n)
w = rbinom(n,1,sigmoid(10*x-5))
x = c(x,runif(n)*0.5)
w = c(w,rep(0,n))
tau = 1 + 0.2*rnorm(n) # treatment effect
y = tau*w + 2*x + 0.5*rnorm(n)
data = data.frame(x=x,w=w,y=y)
```

Simulation 8



Simulation 8



Simulation 8

- Here our data shows a lack of overlap[25]: at large values of x, there are only treated units. At small values of x, there are only control units. This poses challenges for both regression and matching[26].

- For regression, to predict $\mu_1(x_i)$ at small x_i – where there are only control units – and to predict $\mu_0(x_i)$ at large x_i – where there are only treated units, we will have to rely heavily on extrapolation.

- For matching, lack of overlap will result in poor matches and a matched sample that still has significant covariate imbalance[27].

---

[25] Formally defined on page 131.

[26] Lack of overlap is a problem for causal inference in general, as we will have to estimate certain counterfactual outcomes based on samples that do not overlap.

[27] A benefit of the matching procedure is that it helps reveal whether there is any lack of overlap. The regression procedure does not do so.

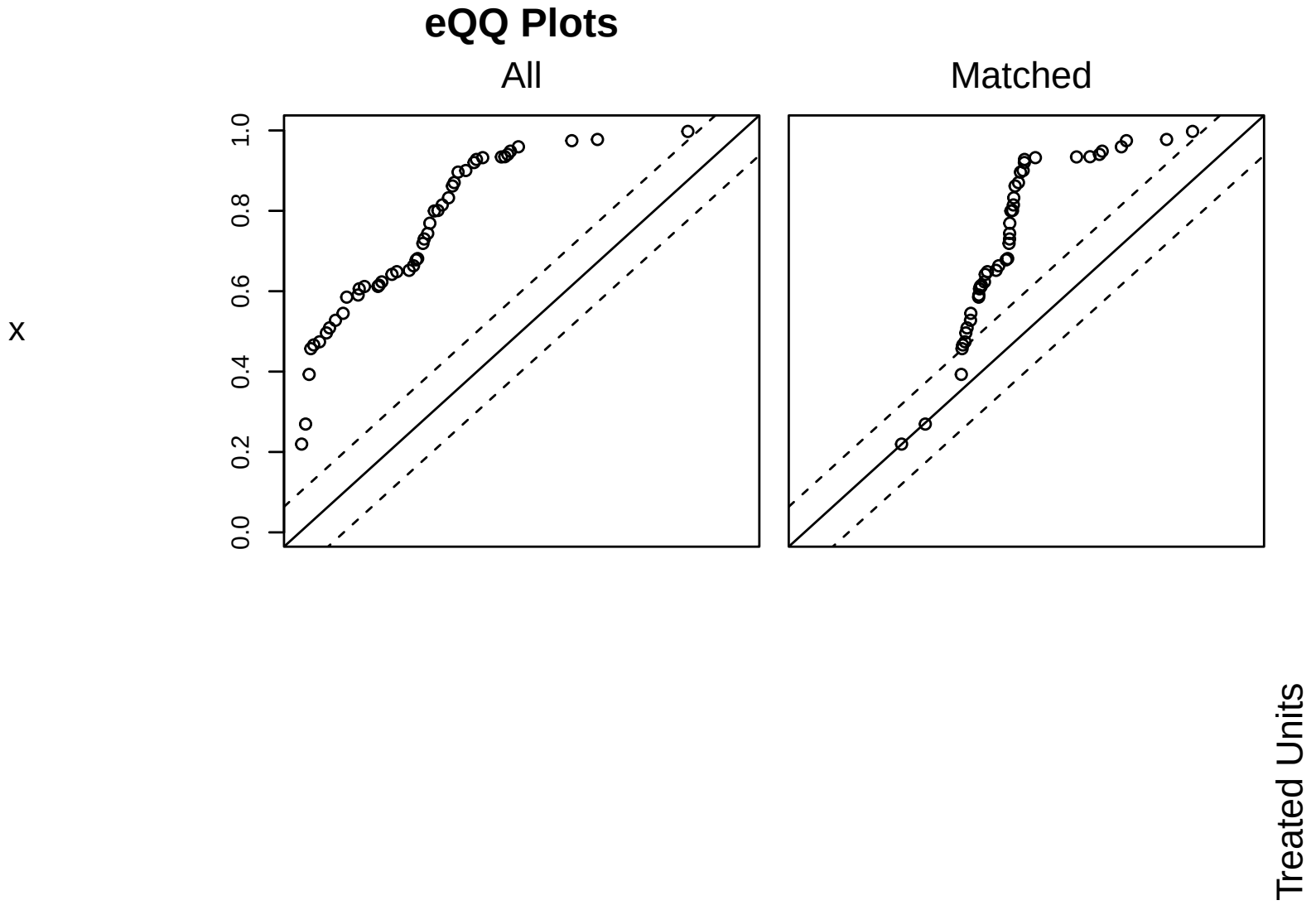
Simulation 8

```
################################
# Nearest Neighbor Matching #
################################
require(MatchIt)
m = matchit(w ~ x, data, distance = "mahalanobis") # 1-NN matching

# create matched data set
m.data = match.data(m)
```

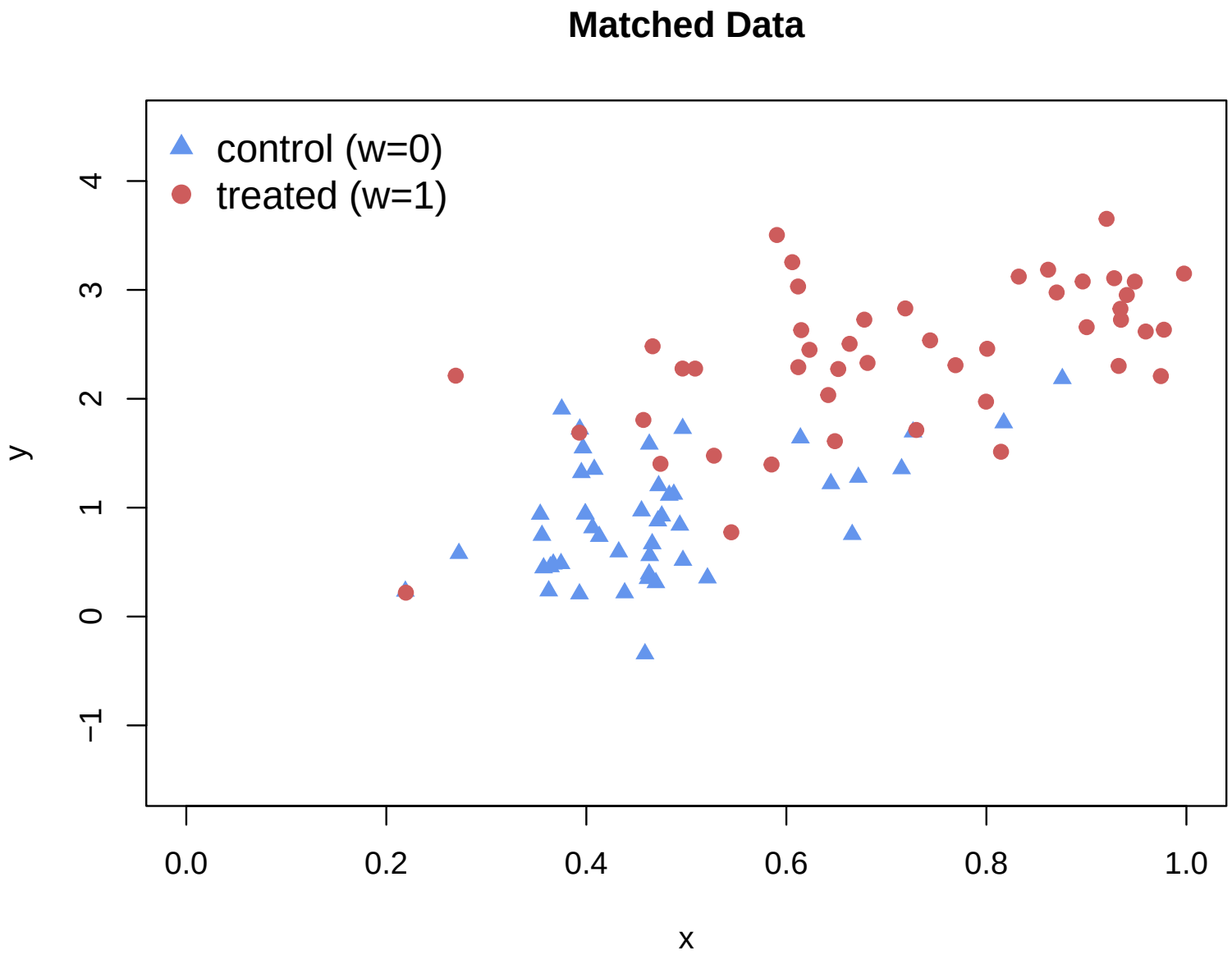
Simulation 8

```
summary(m)

##
## Call:
## matchit(formula = w ~ x, data = data, distance = "mahalanobis")
##
## Summary of Balance for All Data:
##   Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
## x        0.7119        0.2674          2.2494     1.2263    0.4433   0.7981
##
##
## Summary of Balance for Matched Data:
##   Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
## x        0.7119        0.4718          1.2148     2.2355    0.1598   0.6522
##   Std. Pair Dist.
## x          1.2204
##
## Percent Balance Improvement:
##   Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
## x              46     -294.3        64     18.3
##
## Sample Sizes:
##           Control Treated
## All           154      46
## Matched        46      46
```

Simulation 8



Simulation 8



Simulation 8

```
# ATT estimate based on matched data
require(AER)
fit = lm(y ~ w, data = m.data)
coeftest(fit)

##
## t test of coefficients:
##
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 0.912693   0.092987  9.8153 6.912e-16 ***
## w           1.484231   0.131503 11.2867 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simulation 8

To improve estimate, we can fit regression models *on the matched data*. Doing so allows us to adjust for the imbalances that remain in the matched sample.

```
############################
# Post-Matching Regression #
############################
fit = lm(y ~ w + x, data = m.data)
coeftest(fit)

##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -0.067768   0.174504 -0.3883    0.6987
## w            0.985291   0.135782  7.2564 1.416e-10 ***
## x            2.077969   0.331001  6.2778 1.223e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Propensity Score

Given binary treatment $w \in \{0, 1\}$, let

$$e(x) \doteq \mathbb{E}[w|x] = \Pr(w = 1|x)$$

be the **propensity score** – the conditional probability of receiving the treatment.

Propensity Score

Overlap (Positivity)

$$0 < \Pr(w = 1|x) < 1 \quad \forall x \tag{18}$$

Unconfoundedness Given the Propensity Score

Given binary treatment $w \in \{0, 1\}$, if the unconfoundedness condition (5) and the overlap condition (18) are satisfied, then

$$w \perp (\mathcal{Y}(0), \mathcal{Y}(1)) | e(x) \tag{19}$$

Proof

Propensity Score

When a set of variables x satisfies the backdoor criterion, it suffices to condition on the propensity score $e(x)$ in order to make the treatment assignment mechanism ignorable.

(19) $\Rightarrow$ under unconfoundedness, for $a \in \{0, 1\}$,

$$\begin{aligned} \mathbb{E}[y|\text{do}(w = a)] &= \int \mathbb{E}[y|\text{do}(w = a), e(x)] p(x) dx \\ &= \int \mathbb{E}[y|w = a, e(x)] p(x) dx \\ &= \int \mathbb{E}[y|w = a, e] p(e) de \end{aligned} \tag{20}$$

Propensity Score

For $e \in (0,1)$, define $\nu_a(e) \doteq \mathbb{E}[y|\, w = a, e(x) = e]$. Then[28]

$$\tau_{\text{ATE}} = \mathbb{E}_e[\nu_1(e) - \nu_0(e)]$$

$$\widehat{\tau}_{\text{ATE}} = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{\nu}_1\left(\widehat{e}(x_i)\right) - \widehat{\nu}_0\left(\widehat{e}(x_i)\right)\right)$$

, where $\widehat{e}(x)$ is the estimated propensity score.

[28] An equivalent way to derive (20) is to note that by (19),

$$\begin{aligned}\nu_a(e) &= \mathbb{E}[\mathcal{Y}(a)|\, w = a, e(x) = e] \\ &= \mathbb{E}[\mathcal{Y}(a)|\, e(x) = e]\end{aligned}$$

, so that $\mathbb{E}_e[\nu_a(e)] = \mathbb{E}[\mathcal{Y}(a)]$.

Regression on the Propensity Score

- If the propensity score $e(x)$ is known, instead of regressing y on w and x, we can regress y on w and $e(x)$, and calculate our causal effects of interest based on the estimated $\widehat{\nu}_a(e)$[29].

---

[29] Intuitively, suppose

$$y = \beta_0 + \beta_1 w + \beta_2 x + \epsilon$$

, where both x and w are scalars. Then omitting x from the regression leads to a biased estimate of β_1:

$$\beta_1 = \frac{Cov(w, y)}{\mathbb{V}(w)} = \frac{\beta_1 \mathbb{V}(w) + \beta_2 Cov(w, x)}{\mathbb{V}(w)}$$

, where the bias is equivalent to $\beta_2 \gamma$, where $\gamma = \frac{Cov(w,x)}{\mathbb{V}(w)}$ is the coefficient in the following regression:

$$x = \alpha + \gamma \cdot w + e$$

Because $x \perp w | e(x)$, conditioning on the propensity score will remove the correlation between x and w, leading to an unbiased estimate of β_1.

Regression on the Propensity Score

- Since the propensity score is unknown[30], we need to estimate $e(x) = \Pr(w = 1|x)$ first. This can be achieved using *any* appropriate classification and discrete choice models[31].

[30] In RCT studies, the true propensity score would be known, since the investigator controls the treatment assignment mechanism.

[31] Since the regression on the propensity score approach requires first estimating the propensity score and then estimating the conditional expectation of y given w and the estimated propensity score, the approach is generally less efficient than other propensity-score based methods and therefore seldom used.

Matching on the Propensity Score

- (19) implies that when the unconfoundedness condition is satisfied, it is sufficient to adjust solely for differences in the propensity score between treated and control units[32].

[32] In other words, we do not necessarily need the treatment and the control groups to have the same distriubtion of x, as long as they have the same distriubtion of $e(x)$. Note that $p(x|w=0) = p(x|w=1) \Rightarrow p(e(x)|w=0) = p(e(x)|w=1)$ but not vice versa.

Matching on the Propensity Score

- Therefore, instead of matching on x, we can match on the propensity score $e(x)$. The goal is to create a matched sample in which

$$p(e(x)|w=0) = p(e(x)|w=1)$$

  Doing so "reduces" the problem of matching a multi-dimensional variable x to the problem of matching a scalar variable $e(x)$[33].

- Since the propensity score is unknown, we can match based on the estimated score $\widehat{e}(x)$. This method is called **propensity score matching** (**PSM**).

[33] In practice, this means matching based on the distance measure $|e(x) - e(x')|$ rather than $\|x - x'\|_{\mathcal{M}}$.

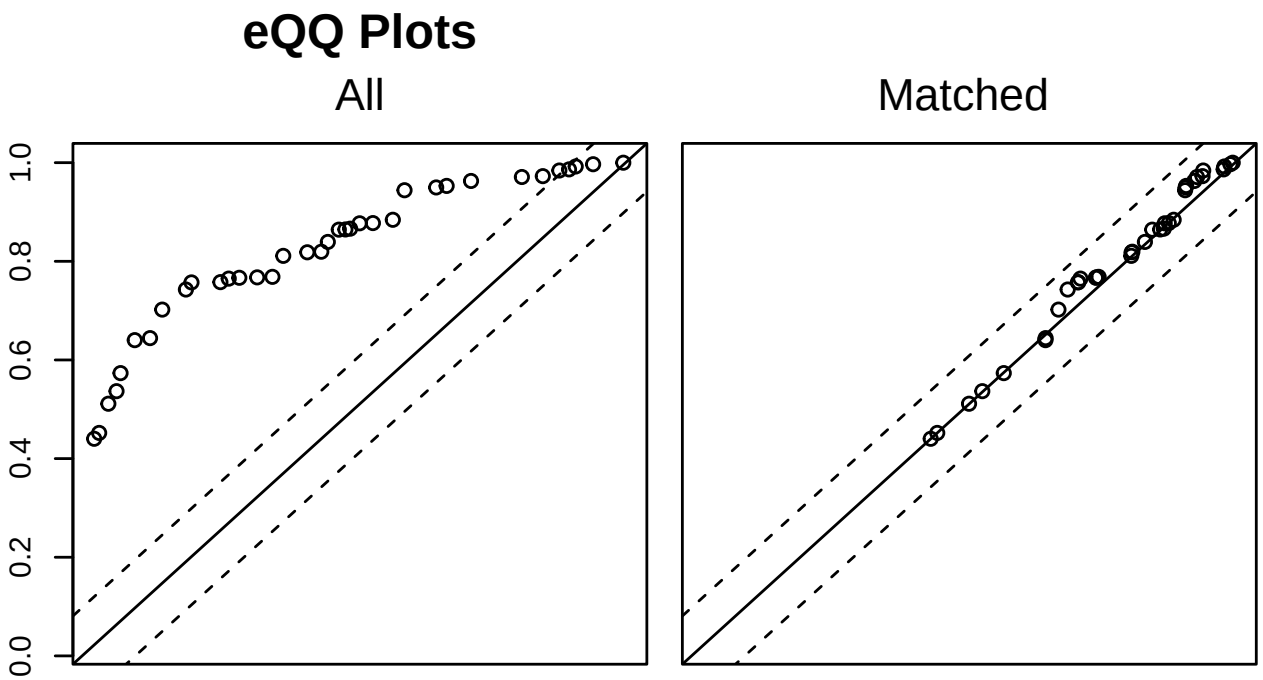
Simulation 7

```r
####################################
# Propensity Score Matching (PSM) #
####################################
# Estimate propensity score (ps) by logistic regression
psfit = glm(w ~ x, data, family=binomial)
data$ps = predict(psfit,data,type="response") # estimated ps

# Matching on estimated ps
psm = matchit(w ~ ps, data, distance = "mahalanobis")

# Equivalently,
psm = matchit(w ~ x, data, distance = "logit")
```

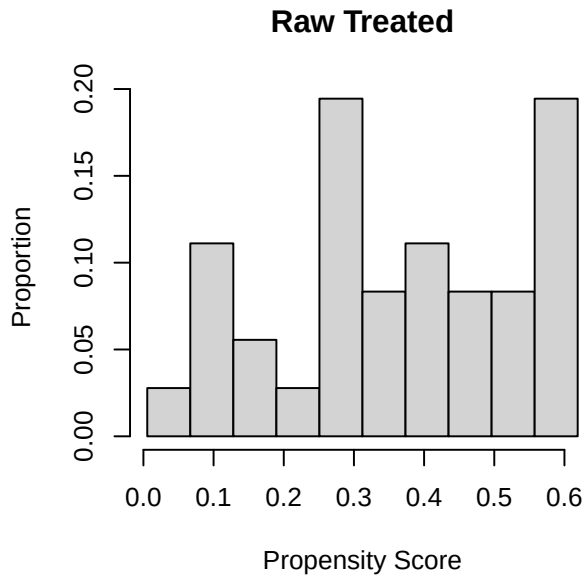
Simulation 7

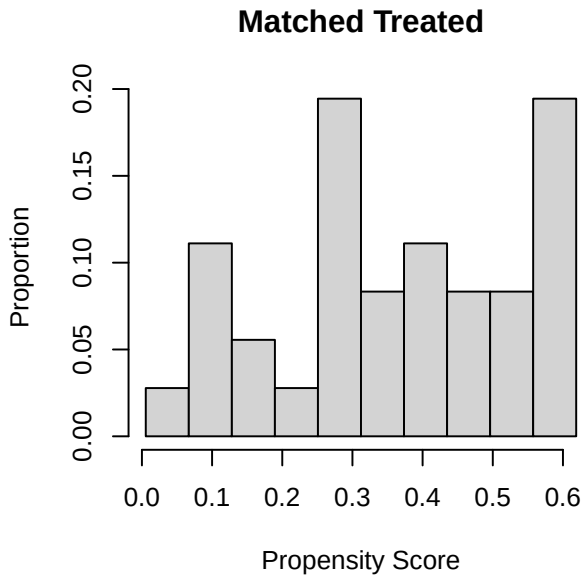
```
summary(psm)

##
## Call:
## matchit(formula = w ~ x, data = data, distance = "logit")
##
## Summary of Balance for All Data:
##          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance        0.3784        0.1365          1.3855     1.0416    0.3457
## x               0.8072        0.4391          2.3034     0.3062    0.3457
##          eCDF Max
## distance   0.5989
## x          0.5989
##
##
## Summary of Balance for Matched Data:
##          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance        0.3784        0.3566          0.1245     1.0976    0.0186
## x               0.8072        0.7902          0.1067     1.0724    0.0186
##          eCDF Max Std. Pair Dist.
## distance   0.1944          0.1323
## x          0.1944          0.1155
##
## Percent Balance Improvement:
##          Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
```

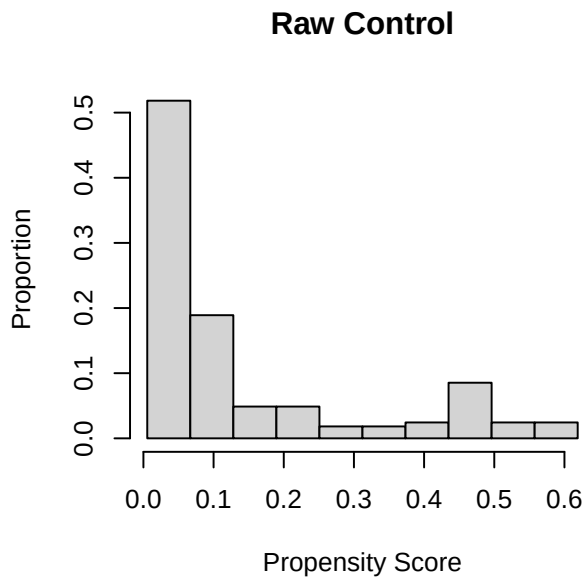
Simulation 7

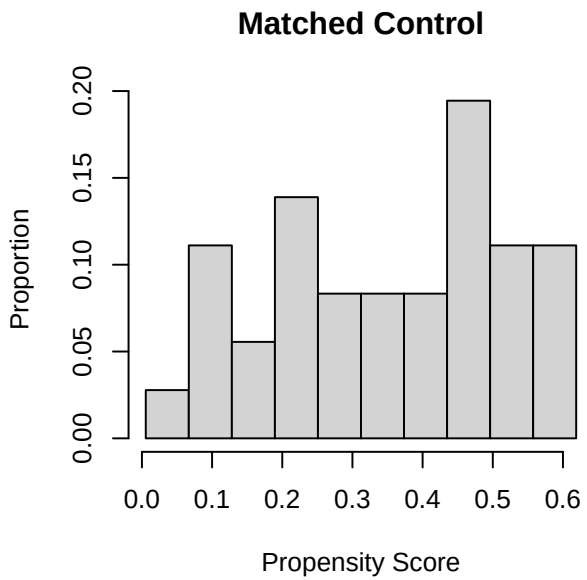


Simulation 7

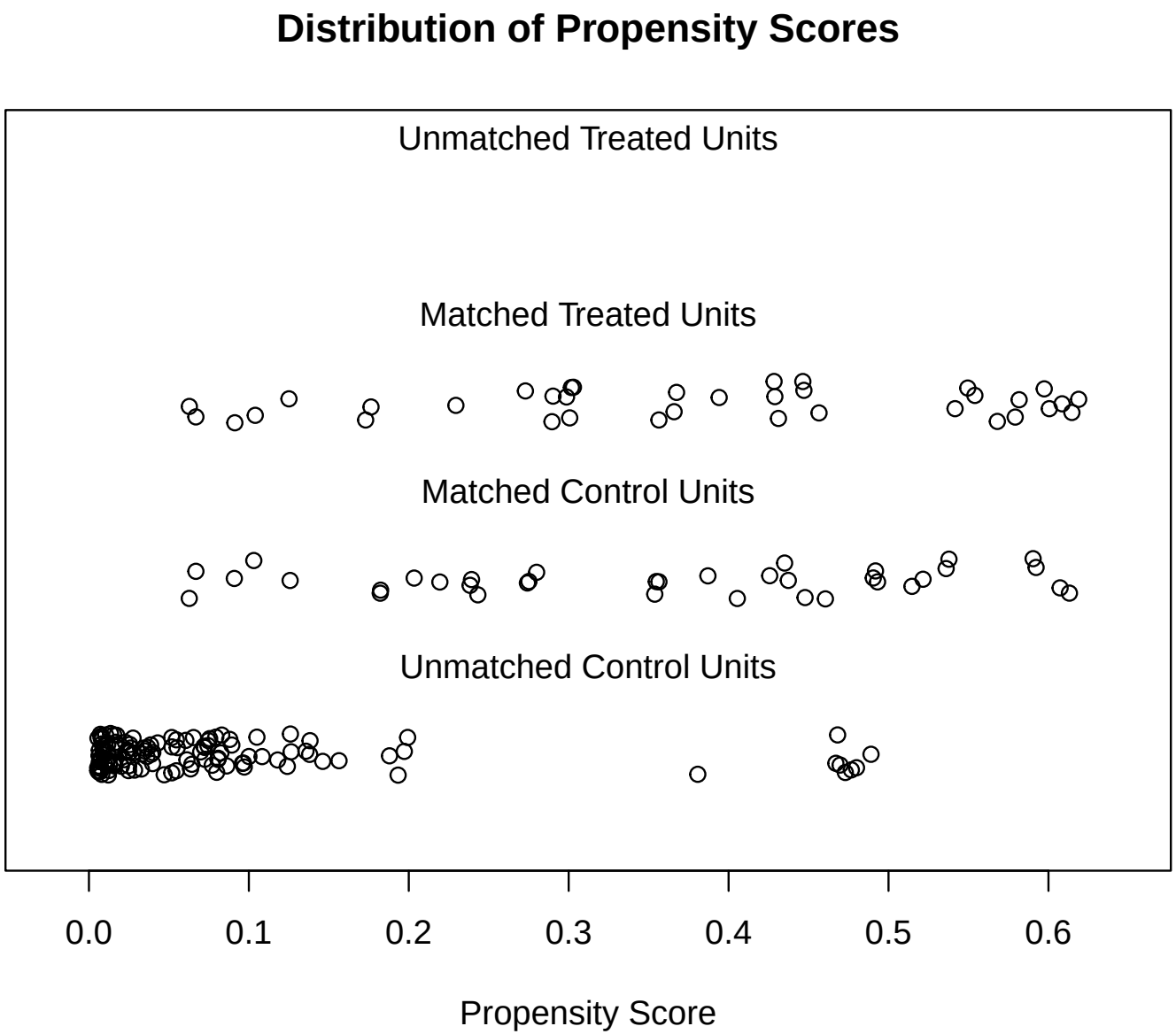








Simulation 7



Simulation 7

```
# create matched data set
psm.data = match.data(psm)

# ATT estimate based on matched data
require(AER)
fit = lm(y ~ w, data = psm.data)
coeftest(fit)

##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.49210    0.11876 12.5643  < 2e-16 ***
## w            0.99363    0.16795  5.9163  1.1e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Inverse Probability Weighting

- Suppose we have a population of N individudals, out of which we randomly draw a sample of n individuals. Let y be a variable of interest (e.g., income). Based on the observed $\{y_i\}_{i=1}^n$, can we estimate the mean and the total of y_i in the underlying population?

- Let $e = n/N$ be the **sampling probability**. Let $Y = \sum_{i=1}^N y_i$ denote the total y in the population. Then

$$\widehat{Y} = \frac{1}{e} \sum_{i=1}^{n} y_i \tag{21}$$

$$\widehat{\mathbb{E}}[y] = \frac{1}{n} \sum_{i=1}^{n} y_i = \frac{1}{N} \left(\frac{1}{e} \sum_{i=1}^{n} y_i \right) \tag{22}$$

, where $\mathbb{E}[.]$ is with respect to the population[34].

[34] Here because we are doing random sampling, our sample mean is an unbiased estimate of the population mean.

Inverse Probability Weighting

Let $w_i \in \{0, 1\}$, $i = 1, \ldots, N$ be an indicator of whether an individual i in the population is selected into the sample. Then $e = \Pr(w_i = 1)$ and we can write (21) and (22) as[35]:

$$\widehat{Y} = \frac{1}{e} \sum_{i=1}^{N} w_i \cdot y_i \tag{23}$$

$$\widehat{\mathbb{E}}[y] = \frac{1}{N} \left(\frac{1}{e} \sum_{i=1}^{N} w_i \cdot y_i \right) \tag{24}$$

[35] (23) and (24) also follow from the following observation:

$$\begin{aligned} \mathbb{E}[w \cdot y] &= \mathbb{E}[w \cdot y \,|\, w = 1] \Pr(w = 1) + \mathbb{E}[w \cdot y \,|\, w = 1] \Pr(w = 1) \\ &= \mathbb{E}[y \,|\, w = 1] \Pr(w = 1) \end{aligned}$$

Hence $\frac{1}{n} \sum_{i=1}^{n} y_i \approx \mathbb{E}[y \,|\, w = 1] = \mathbb{E}[w \cdot y] / \Pr(w = 1) \approx \frac{1}{Ne} \sum_{i=1}^{N} w_i \cdot y_i$.

Inverse Probability Weighting

- Now suppose the population is characterized by $\{x_i, y_i\}_{i=1}^{N}$, where x_i are individual characteristics (e.g., gender) and y_i is the variable of interest (e.g., income).
- Suppose when we generate our sample, individuals of different characteristics are sampled with different probabilites. Specifically, individuals of characteristics x_i are drawn with probability $e(x_i)$[36].
- In this case, based on the observed $\{x_i, y_i\}_{i=1}^{n}$, can we estimate the mean and the total of y_i in the underlying population?

---

[36] This sampling scheme is called **conditional random sampling**. Conditional on x, individuals are sampled randomly from the population, but people of different x have different sampling probabilities. For example, out of a population consisting of 200 men and 100 women, we might create a sample of 10 men, each drawn with 5% probability from the underlying male population, and 10 women, each drawn with 10% probability from the underlying female population.

Inverse Probability Weighting

The **Horvitz–Thompson estimator** for the population total and the population mean are given by:

$$\widehat{Y} = \sum_{i=1}^{n} \frac{y_i}{e(x_i)} \tag{25}$$

$$\widehat{\mathbb{E}}[y] = \frac{1}{N} \sum_{i=1}^{n} \frac{y_i}{e(x_i)} \tag{26}$$

Proof

Inverse Probability Weighting

Let $\{w_i\}_{i=1}^N$ indicate whether an individual in the population is selected into the sample. Then $e(x_i) = \Pr(w_i = 1 | x_i)$ and we can write (25) and (26) as:

$$\widehat{Y} = \sum_{i=1}^{N} \frac{w_i \cdot y_i}{e(x_i)} \tag{27}$$

$$\widehat{\mathbb{E}}[y] = \frac{1}{N} \sum_{i=1}^{N} \frac{w_i \cdot y_i}{e(x_i)} \tag{28}$$

Inverse Probability Weighting

- When the sampling probabilities are known, we can estimate population statistics by weighting each observation with the reciprocal of its sampling probability. This approach is called **inverse probability weighting** (**IPW**)[37].

[37] Intuitively, one can think of the inverse probability $1/e(x_i)$ as "how many people in the underlying population this sample individual represents". Using the example in footnote 36, each man in the sample would represent $1/0.05 = 20$ men in the population and each woman in the sample would represent $1/0.1 = 10$ women in the population.

Inverse Probability Weighting

- Now suppose we observe data $\{w_i, x_i, y_i\}$, where $w_i \in \{0, 1\}$ is the treatment variable, y_i is the outcome, and x_i are measured covariates that satisfy the back-door criterion.

- For treated individuals, we observe $y_i = \mathcal{Y}_i(1)$. Based on these individuals, can we infer the population mean $\mathbb{E}[\mathcal{Y}(1)]$?

Inverse Probability Weighting

Let $e(x_i) = \Pr(w_i = 1 | x_i)$, then

$$\begin{aligned} \widehat{\mathbb{E}}[\mathcal{Y}(1)] &= \frac{1}{N} \sum_{i=1}^{N} \frac{w_i \cdot \mathcal{Y}_i(1)}{e(x_i)} \\ &= \frac{1}{N} \sum_{i=1}^{N} \frac{w_i \cdot y_i}{e(x_i)} \end{aligned} \tag{29}$$

, where N is now the size of the observed sample.

Inverse Probability Weighting

Similarly, we can infer the population mean $\mathbb{E}[\mathcal{Y}(0)]$ by applying inverse probability weighting to the untreated individuals:

$$\begin{aligned}\widehat{\mathbb{E}}[\mathcal{Y}(0)] &= \frac{1}{N}\sum_{i=1}^{N}\frac{(1-w_i)\cdot\mathcal{Y}_i(0)}{1-e(x_i)} \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{(1-w_i)\cdot y_i}{1-e(x_i)}\end{aligned} \tag{30}$$

Based on (29) and (30), we can compute the ATE as

$$\widehat{\tau}_{\mathsf{ATE}} = \widehat{\mathbb{E}}[\mathcal{Y}(1)] - \widehat{\mathbb{E}}[\mathcal{Y}(0)] \tag{31}$$

Inverse Probability Weighting

In general, given $w_i \in \{1, \ldots, A\}$, when the unconfoundedness condition is satisfied[38], we have, for $a \in \{1, \ldots, A\}$,

$$\mathbb{E}\left[\mathcal{Y}\left(a\right)\right] = \mathbb{E}\left[\frac{\mathcal{I}\left(w = a\right) \cdot y}{\Pr\left(w = a | x\right)}\right] \tag{32}$$

Proof

- We can infer the population mean of the potential outcomes under treatment a – the mean outcome if *all* subjects receive a – by weighting the observed outcomes under a with the reciprocal of their treatment assignment probabilities.

[38] The unconfoundedness condition is necessary for (32) to hold. While (32) can be viewed as following directly from the Horvitz–Thompson estimator, see Proof for how the Horvitz–Thompson estimator itself can be viewed as requiring an exchangeability condition.

Inverse Probability Weighting

39

[39] Since $\mathbb{E}\left[\mu_a\left(x\right)\right] = \mathbb{E}\left[\mathcal{Y}\left(a\right)\right]$, we have:

$$\mathbb{E}_x\left[\mu_a\left(x\right)\right] = \mathbb{E}_{x,w,y}\left[\frac{\mathcal{I}\left(w=a\right)\cdot y}{\Pr\left(w=a|\,x\right)}\right] \tag{33}$$

, where we make explicit the variables on which the expectations are the taken. (33)$\Rightarrow$

$$\begin{aligned}\mu_a\left(x\right) &= \mathbb{E}\left[\left.\frac{\mathcal{I}\left(w=a\right)\cdot y}{\Pr\left(w=a|\,x\right)}\right| x\right] \\ &= \frac{\mathbb{E}\left[\mathcal{I}\left(w=a\right)\cdot y|\,x\right]}{\Pr\left(w=a|\,x\right)}\end{aligned}$$

Propensity Score Weighting

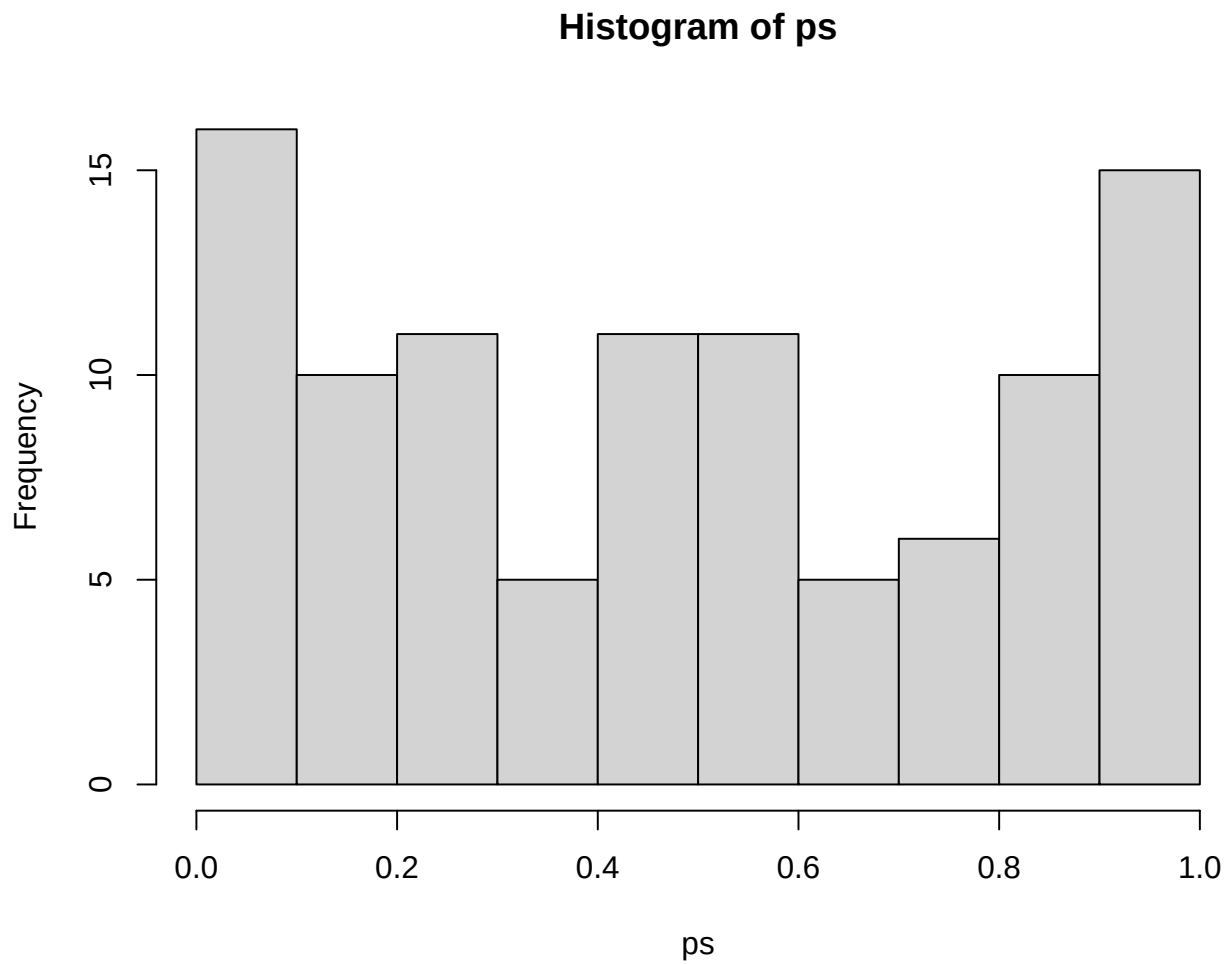
- To estimate the ATE based on (31), we can plug the estimated propensity score $\widehat{e}(x)$ into (29) and (30). The resulting estimator is called the **propensity score weighting** (**PSW**) estimator.

Simulation 1

```
# Estimate propensity score (ps) by logistic regression
psfit = glm(w ~ x, data, family=binomial)
coeftest(psfit)

##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept) -0.09452    0.26422 -0.3577    0.7205
## x            1.14829    0.24171  4.7508 2.026e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simulation 1

```
ps = predict(psfit,data,type="response") # estimated ps
```



Simulation 1

```
#####################################
# Propensity Score Weighting (PSW) #
#####################################
y1hat = mean(w*y/ps)
y0hat = mean((1-w)*y/(1-ps))
atehat = y1hat - y0hat
atehat

## [1] 2.052948
```

Propensity Score Based Methods

- Propensity score based methods avoid adjusting directly for all covariates and instead focus on adjusting for differences in the propensity score.

- In practice, although they avoid potentially high-dimensional regression of conditional expectations, they require potentially equally high-dimensional regression of the treatment indicator on the covariates.

- The relative merits of these estimators will depend on whether the propensity score is more or less smooth than the regression functions, and on whether additional information is available about either the propensity score or the regression functions.

Appendix: Propensity Score

Proof.

$$
\begin{aligned}
&\Pr\left(w=1\left|\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right),e\left(x\right)\right.\right)\\
&=\mathbb{E}\left[w\left|\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right),e\left(x\right)\right.\right]\\
&=\mathbb{E}\left[\mathbb{E}\left[w\left|\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right),e\left(x\right),x\right.\right]\right|\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right),e\left(x\right)\right]\\
&\overset{[1]}{=}\mathbb{E}\left[\mathbb{E}\left[w\left|x\right.\right]\right|\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right),e\left(x\right)\right]\\
&=\mathbb{E}\left[e\left(x\right)\right|\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right),e\left(x\right)\right]\\
&=\mathbb{E}\left[e\left(x\right)\right|e\left(x\right)\right]=e\left(x\right)
\end{aligned}
$$

, where [1] follows from unconfoundedness that $w \perp \left(\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right)\right)\left|x\right.$. □

Appendix: Propensity Score

Proof. (cont.)

Similarly,

$$\begin{aligned}\Pr\left(w=1\left|e\left(x\right)\right.\right) &= \mathbb{E}\left[w\left|e\left(x\right)\right.\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[w\left|e\left(x\right),x\right.\right]\right| e\left(x\right)] \\ &= \mathbb{E}\left[\mathbb{E}\left[w\left|x\right.\right]\right| e\left(x\right)] = e\left(x\right)\end{aligned}$$

Since $\Pr\left(w=1\left|\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right),e\left(x\right)\right.\right)=\Pr\left(w=1\left|e\left(x\right)\right.\right)=e\left(x\right)$,

$$w\perp\left(\mathcal{Y}\left(0\right),\mathcal{Y}\left(1\right)\right)\left|e\left(x\right)\right.$$

□

Appendix: Horvitz-Thompson Estimator

Proof.

Let $w_i \in \{0, 1\}$, $i = 1, \ldots, N$ indicate whether an individual in the population is selected into the sample. Then

$$
\begin{aligned}
\mathbb{E}\left[\frac{w \cdot y}{e(x)}\right] &= \mathbb{E}\left[\mathbb{E}\left[\frac{w \cdot y}{e(x)}\middle| x\right]\right] \\
&= \mathbb{E}\left[\frac{\mathbb{E}[w \cdot y| x]}{e(x)}\right] \\
&\overset{[1]}{=} \mathbb{E}\left[\mathbb{E}[y| w = 1, x]\right] \\
&\overset{[2]}{=} \mathbb{E}\left[\mathbb{E}[y| x]\right] \\
&= \mathbb{E}[y]
\end{aligned}
$$

□

Appendix: Horvitz-Thompson Estimator

Proof. (cont.)

[1] follows since

$$\begin{aligned}\mathbb{E}[w \cdot y|\, x] &= \mathbb{E}[w \cdot y|\, w=1, x]\,\Pr(w=1|x) \\ &\quad + \mathbb{E}[w \cdot y|\, w=0, x]\,\Pr(w=0|x) \\ &= \mathbb{E}[y|\, w=1, x]\,\Pr(w=1|x) \\ &= \mathbb{E}[y|\, w=1, x]\, e(x)\end{aligned}$$

□

Appendix: Horvitz-Thompson Estimator

Proof. (cont.)

[2] follows since in the population, individuals of the same characteristics x have the same probability of sample selection, i.e., sampling is random conditional on x. Therefore, we have

$$w \perp y | x \tag{34}$$

Hence $\mathbb{E}[y|w=1,x] = \mathbb{E}[y|w=0,x] = \mathbb{E}[y|x]$. (34) is an conditional exchangeability condition.

□

Appendix: Weighting Estimator

Proof.

$$
\begin{aligned}
\mathbb{E}\left[\frac{\mathcal{I}(w=a)\cdot y}{\Pr(w=a|x)}\right] &= \mathbb{E}\left[\frac{\mathcal{I}(w=a)\cdot \mathcal{Y}(a)}{\Pr(w=a|x)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{\mathcal{I}(w=a)\cdot \mathcal{Y}(a)}{\Pr(w=a|x)}\middle| x\right]\right] \\
&\overset{[1]}{=} \mathbb{E}\left[\frac{\mathbb{E}\left[\mathcal{I}(w=a)|\, x\right]\cdot \mathbb{E}\left[\mathcal{Y}(a)|\, x\right]}{\Pr(w=a|x)}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\mathcal{Y}(a)|\, x\right]\right] = \mathbb{E}\left[\mathcal{Y}(a)\right]
\end{aligned}
$$

, where [1] follows due to unconfoundedness: $w \perp \mathcal{Y}(a)|\, x$. □

Acknowledgement I

Part of this lecture is adapted from the following sources:

- Blackwell, M. 2015. *Causal Inference*. Lecture at Harvard University, retrieved on 2017.01.01. [link]
- Hernán, M. A. and J. M. Robins. 2018. *Causal Inference*. Boca Raton: Chapman & Hall/CRC, forthcoming.
- Morgan, S. L. and C. Winship. 2007. *Counterfactuals and Causal Inference: Methods and Principles for Social Research*. Cambridge University Press.
- Pearl, J. 2009. *Causality: Models, Reasoning and Inference* (2nd ed.). Cambridge University Press.
- Roy, J. A. *A Crash Course in Causality: Inferring Causal Effects from Observational Data*, Lecture at the University of Pennsylvania, retrieved on 2019.01.01. [link]

Reference

Imbens, G. 2004. "Nonparametric estimation of average treatment effects under exogeneity," *The Review of Economics and Statistics*, 86(1).

Ozery-Flato, M., P. Thodoroff, M. Ninio, M. Rosen-Zvi, and T. El-Hay. 2020. "Adversarial Balancing for Causal Inference," arXiv:2004.12601.