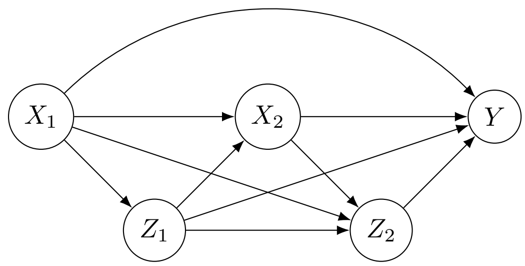
Longitudinal exposure and the Marginal Structural Model

Suppose we wish to study the effect of a longitudinal exposure pattern, over K intervals, on an outcome. We have for $i = 1, \ldots, n$

- Exposures $Z_{1i}, \ldots, Z_{Ki}$;
- Confounders $X_{1i}, \ldots, X_{Ki}$;
- Outcome Y_i.

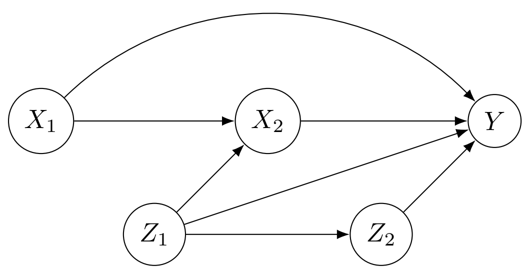
For $K = 2$, we might consider the following DAG structure:



The full probability model for this DAG for observational distribution $\mathcal{O}$ factorizes as

$$f^{\mathcal{O}}_{X_1}(x_1) f^{\mathcal{O}}_{Z_1|X_1}(z_1|x_1) f^{\mathcal{O}}_{X_2|X_1,Z_1}(x_2|x_1,z_1) f^{\mathcal{O}}_{Z_2|X_1,X_2,Z_1}(z_2|x_1,x_2,z_1) f^{\mathcal{O}}_{Y|X_1,X_2,Z_1,Z_2}(y|x_1,x_2,z_1,z_2).$$

To consider the causal effect of an exposure pattern $Z = z$ where $Z = (Z_1, Z_2)$ and $z = (z_1, z_2)$, we first consider the following DAG:



The full probability model for this DAG for experimental distribution $\mathcal{E}$ factorizes as

$$f^{\mathcal{E}}_{X_1}(x_1) f^{\mathcal{E}}_{Z_1}(z_1) f^{\mathcal{E}}_{X_2|X_1,Z_1}(x_2|x_1,z_1) f^{\mathcal{E}}_{Z_2|Z_1}(z_2|z_1) f^{\mathcal{E}}_{Y|X_1,X_2,Z_1,Z_2}(y|x_1,x_2,z_1,z_2).$$

Therefore, the ratio of $f^{\mathcal{E}}$ to $f^{\mathcal{O}}$ becomes

$$\frac{f^{\mathcal{E}}_{X_1}(x_1) f^{\mathcal{E}}_{Z_1}(z_1) f^{\mathcal{E}}_{X_2|X_1,Z_1}(x_2|x_1,z_1) f^{\mathcal{E}}_{Z_2|Z_1}(z_2|z_1) f^{\mathcal{E}}_{Y|X_1,X_2,Z_1,Z_2}(y|x_1,x_2,z_1,z_2)}{f^{\mathcal{O}}_{X_1}(x_1) f^{\mathcal{O}}_{Z_1|X_1}(z_1|x_1) f^{\mathcal{O}}_{X_2|X_1,Z_1}(x_2|x_1,z_1) f^{\mathcal{O}}_{Z_2|X_1,X_2,Z_1}(z_2|x_1,x_2,z_1) f^{\mathcal{O}}_{Y|X_1,X_2,Z_1,Z_2}(y|x_1,x_2,z_1,z_2)}$$

which, under the usual assumptions and cancelations becomes

$$\frac{f^{\mathcal{E}}_{Z_1}(z_1) f^{\mathcal{E}}_{Z_2|Z_1}(z_2|z_1)}{f^{\mathcal{O}}_{Z_1|X_1}(z_1|x_1) f^{\mathcal{O}}_{Z_2|X_1,X_2,Z_1}(z_2|x_1,x_2,z_1)} = \frac{f^{\mathcal{E}}_{Z_1,Z_2}(z_1,z_2)}{f^{\mathcal{O}}_{Z_1,Z_2|X_1,X_2}(z_1,z_2|x_1,x_2)} \equiv \frac{f^{\mathcal{E}}_{Z}(z)}{f^{\mathcal{O}}_{Z|X}(z|x)}$$

say.

Using the inverse weighting (importance sampling) argument, we can write in the usual way

$$\mu(z) \equiv \mathbb{E}^{\mathcal{E}}_{Y|Z}[Y|Z=z] = \frac{\displaystyle\iiint \mathbb{1}_z(t)\; y\; f^{\mathcal{E}}_{Y|X,Z}(y|x,t) f^{\mathcal{E}}_{X,Z}(x,t)\; dy\; dx\; dt}{\displaystyle\iint \mathbb{1}_z(t) f^{\mathcal{E}}_{X,Z}(x,t)\; dx\; dt}$$

$$= \frac{\displaystyle\iiint \mathbb{1}_z(t)\; y\; \frac{f^{\mathcal{E}}_{Z}(t)}{f^{\mathcal{O}}_{Z|X}(t|x)} f^{\mathcal{O}}_{X,Z}(x,t)\; dy\; dx\; dt}{\displaystyle\iint \mathbb{1}_z(t) \frac{f^{\mathcal{E}}_{Z}(t)}{f^{\mathcal{O}}_{Z|X}(t|x)} f^{\mathcal{O}}_{X,Z}(x,t)\; dx\; dt}$$

and as the term $f^{\mathcal{E}}_{Z}(z)$ is a fixed constant that can be extracted from the two integrals, we have that

$$\mu(z) = \frac{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\dfrac{\mathbb{1}_z(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)} Y\right]}{\mathbb{E}^{\mathcal{O}}_{X,Z}\left[\dfrac{\mathbb{1}_z(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)}\right]} \tag{1}$$

where $\mu(z) = \mu(z_1, z_2)$, $\mathbb{1}_z(Z) = \prod_{k=1}^{2} \mathbb{1}_{z_k}(Z_k)$, and

$$f^{\mathcal{O}}_{Z|X}(z|x) = f^{\mathcal{O}}_{Z_1|X_1}(z_1|x_1) f^{\mathcal{O}}_{Z_2|X_1,X_2,Z_1}(z_2|x_1, x_2, z_1).$$

As usual, we have that

$$\mathbb{E}^{\mathcal{O}}_{X,Z}\left[\frac{\mathbb{1}_z(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)}\right] = \mathbb{E}^{\mathcal{O}}_{X}\left[\mathbb{E}^{\mathcal{O}}_{Z|X}\left[\frac{\mathbb{1}_z(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)}\middle| X\right]\right] = \mathbb{E}^{\mathcal{O}}_{X}\left[1\right] = 1$$

as

$$\mathbb{E}^{\mathcal{O}}_{Z|X}\left[\mathbb{1}_z(Z)|X\right] = f^{\mathcal{O}}_{Z|X}(z|X),$$

so we also may write

$$\mu(z) = \mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{1}_z(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)} Y\right]. \tag{2}$$

As in the single time point case, the moment-based estimators corresponding to (1) and (2) are

$$\widehat{\mu}(z) = \frac{\sum_{i=1}^{n} W_{zi} Y_i}{\sum_{i=1}^{n} W_{zi}} \qquad \widetilde{\mu}(z) = \frac{1}{n}\sum_{i=1}^{n} W_{zi} Y_i \qquad \text{where} \qquad W_{zi} = \frac{\mathbb{1}_z(Z_i)}{f^{\mathcal{O}}_{Z|X}(Z_i|X_i)}.$$

Simulated Data Set: Consider the following data generation mechanism:

- $X_1 \sim \text{Normal}(1, 1)$;
- $Z_1|X_1 = x_1 \sim Bernoulli(g(\mathbf{x}_1\alpha_1))$, with $\mathbf{x}_1 = (1, x_1)$ and $\alpha_1 = (1, -0.5)^\top$;
- $X_2|X_1 = x_1, Z_1 = z_1 \sim Normal(x_1 + 2z_1, 1)$;
- $Z_2|X_1 = x_1, Z_1 = z_1, X_2 = x_2 \sim Bernoulli(g(\mathbf{x}_2\alpha_2))$, with $\mathbf{x}_2 = (1, x_1, z_1, x_2)$ and $\alpha_2 = (1.0, 0.1, 0.5, -0.5)^\top$;
- $Y|X_1 = x_1, Z_1 = z_1, X_2 = x_2, Z_2 = z_2 \sim \text{Normal}(x_1 + z_1 + x_2 + z_2, 3^2)$.

where $g(t) = \exp\{t\}/(1 + \exp\{t\})$. In this model, we have that

$$\mathbb{E}^{\mathcal{E}}_{Y|X_1,Z_1,X_2,Z_2}[Y|X_1 = x_1, Z_1 = z_1, X_2 = x_2, Z_2 = z_2] = x_1 + z_1 + x_2 + z_2$$

so that the causal quantity of interest is based on

$$\mu(z_1, z_2) = \mathbb{E}^{\mathcal{E}}_{X_1,X_2|Z}[X_1 + X_2|Z = z] + z_1 + z_2 = \mathbb{E}^{\mathcal{E}}_{X_1}[X_1] + \mathbb{E}^{\mathcal{E}}_{X_1}[\mathbb{E}^{\mathcal{E}}_{X_2|X_1,Z}[X_2|X_1, Z = z]] + z_1 + z_2$$

$$= 1 + (1 + 2z_1) + z_1 + z_2 = 2 + 3z_1 + z_2$$

Hence

$$\mu(0, 0) = 2 \qquad \mu(1, 0) = 5 \qquad \mu(0, 1) = 3 \qquad \mu(1, 1) = 6.$$

```
set.seed(34);n<-10000
expit<-function(x){return(1/(1+exp(-x)))}
X1<-rnorm(n,1,1)
al1<-c(1.0,-0.5); al2<-c(1.0,0.1,0.5,-0.5)
eta1<-al1[1]+al1[2]*X1
ps.1<-expit(eta1)
Z1<-rbinom(n,1,ps.1)
X2<-rnorm(n,X1+2*Z1,1)
eta2<-al2[1]+al2[2]*X1+al2[3]*Z1+al2[4]*X2
ps.2<-expit(eta2)
Z2<-rbinom(n,1,ps.2)
eps<-rnorm(n,0,3)
Y<-X1+X2+Z1+Z2+eps
mu.00<-mean(X1)+mean(X2)
mu.10<-mean(X1)+mean(X2)+1
mu.01<-mean(X1)+mean(X2)+1
mu.11<-mean(X1)+mean(X2)+2
c(mu.00,mu.10,mu.01,mu.11)     #Means of the four subsets

+ [1] 3.237393 4.237393 4.237393 5.237393
```

The sample means in the four exposure subgroups are therefore quite different from the causal quantities, as expected due to confounding. In the single timepoint case, we can recover the quantities of interest via a regression of Y on (X, Z). However, here regression does not give the correct result.

```
fit1<-lm(Y~X1+X2+Z1+Z2);round(coef(summary(fit1)),6)

+              Estimate Std. Error   t value Pr(>|t|)
+ (Intercept) 0.213573   0.078020  2.737426 0.006203
+ X1          0.910646   0.043347 21.008092 0.000000
+ X2          1.037746   0.030916 33.566295 0.000000
+ Z1          0.777611   0.087386  8.898532 0.000000
+ Z2          0.978058   0.062915 15.545821 0.000000

mu.00.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=0)))
mu.10.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=0)))
mu.01.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=1)))
mu.11.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=1)))
c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)

+ [1] 3.446219 4.223831 4.424277 5.201888
```

The usual regression approach does not yield the correct result here due to the mediation of the effect of the exposure at the first time interval (that is, the path $Z_1 \to X_2 \to Y$); regressing on X_2, which would be necessary for a correctly specified regression model, has the effect of blocking the mediating path. Note that whereas the coefficient attached to Z_2 is apparently correctly estimated (true value 1, estimated value 0.978058), the coefficient attached to Z_1 is apparently incorrectly estimated (true value 1, estimated value 0.777611)

Inverse weighting: Inverse weighting proceeds using the weights based on

$$W_z = \frac{\mathbb{1}_{z_1}(Z_1)}{e(X_1)^{z_1}(1 - e(X_1))^{1-z_1}} \times \frac{\mathbb{1}_{z_2}(Z_2)}{e(X_1, Z_1, X_2)^{z_2}(1 - e(X_1, Z_1, X_2))^{1-z_2}}$$

and obtains the correct results. For version given by (2) we have

```
W<-(ps.1^Z1*(1-ps.1)^(1-Z1))*(ps.2^Z2*(1-ps.2)^(1-Z2))
W00<-(1-Z1)*(1-Z2)/W
W10<-(Z1)*(1-Z2)/W
W01<-(1-Z1)*(Z2)/W
W11<-(Z1)*(Z2)/W
mu.00.tilde<-mean(W00*Y)
mu.10.tilde<-mean(W10*Y)
mu.01.tilde<-mean(W01*Y)
mu.11.tilde<-mean(W11*Y)
c(mu.00.tilde,mu.10.tilde,mu.01.tilde,mu.11.tilde)

+ [1] 2.115634 4.925918 3.323802 5.960125
```

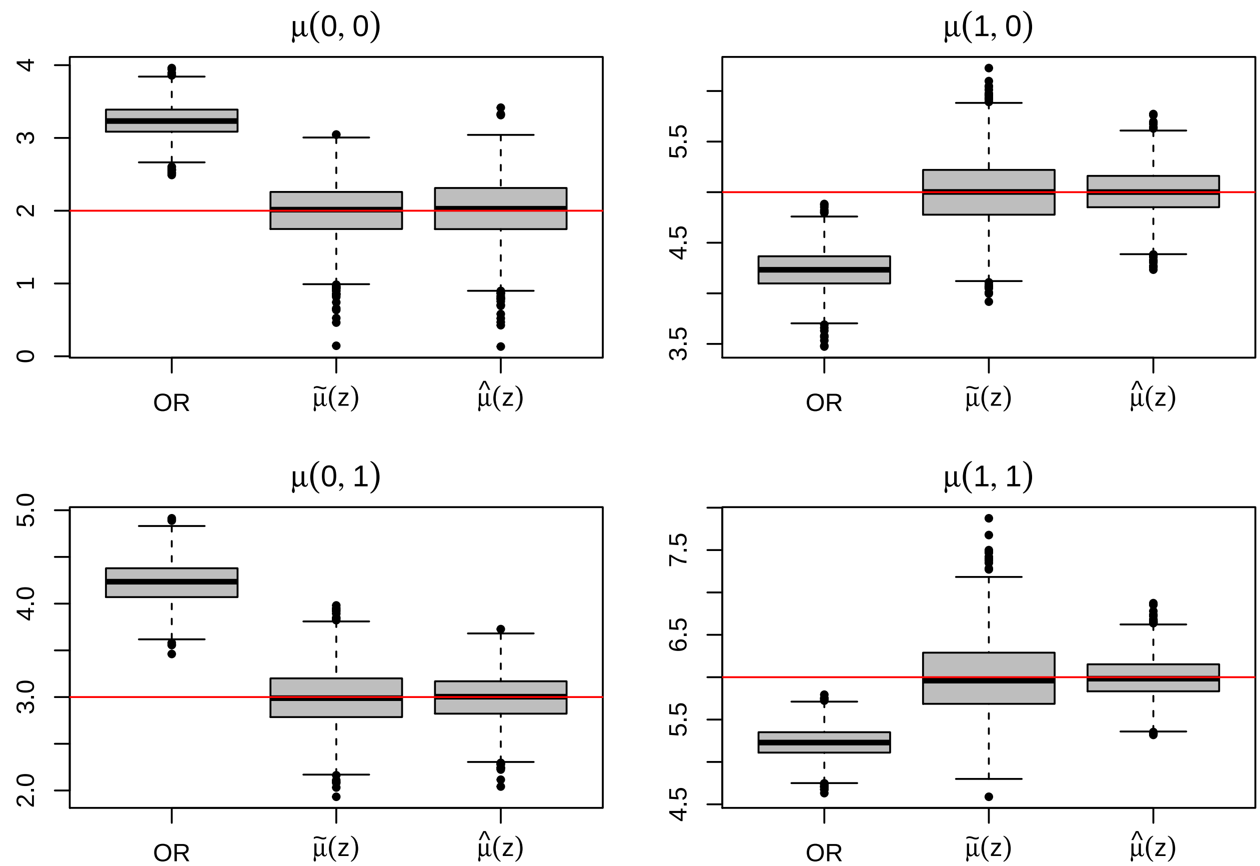
whereas for the version given by (1) we have

```
W00.star<-W00/sum(W00)
W10.star<-W10/sum(W10)
W01.star<-W01/sum(W01)
W11.star<-W11/sum(W11)
mu.00.hat<-sum(W00.star*Y)
mu.10.hat<-sum(W10.star*Y)
mu.01.hat<-sum(W01.star*Y)
mu.11.hat<-sum(W11.star*Y)
c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)

+ [1] 2.206665 4.971965 3.271587 5.942558
```

Simulation Study: We now can replicate this study over 2000 replications, for a smaller sample size $n = 1000$.

```
set.seed(34);n<-1000;nreps<-2000
al1<-c(1.0,-0.5);al2<-c(1.0,0.1,0.5,-0.5)
ests.mat<-array(0,c(nreps,3,4))
for(irep in 1:nreps){
    X1<-rnorm(n,1,1)
    eta1<-al1[1]+al1[2]*X1
    ps.1<-expit(eta1)
    Z1<-rbinom(n,1,ps.1)
    X2<-rnorm(n,X1+2*Z1,1)
    eta2<-al2[1]+al2[2]*X1+al2[3]*Z1+al2[4]*X2
    ps.2<-expit(eta2)
    Z2<-rbinom(n,1,ps.2)
    eps<-rnorm(n,0,3)
    Y<-X1+X2+Z1+Z2+eps
    fit1<-lm(Y~X1+X2+Z1+Z2)
    mu.00.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=0)))
    mu.10.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=0)))
    mu.01.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=1)))
    mu.11.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=1)))
    ests.mat[irep,1,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    W<-(ps.1^Z1*(1-ps.1)^(1-Z1))*(ps.2^Z2*(1-ps.2)^(1-Z2))
    W00<-(1-Z1)*(1-Z2)/W
    W10<-(Z1)*(1-Z2)/W
    W01<-(1-Z1)*(Z2)/W
    W11<-(Z1)*(Z2)/W
    mu.00.tilde<-mean(W00*Y)
    mu.10.tilde<-mean(W10*Y)
    mu.01.tilde<-mean(W01*Y)
    mu.11.tilde<-mean(W11*Y)
    ests.mat[irep,2,]<-c(mu.00.tilde,mu.10.tilde,mu.01.tilde,mu.11.tilde)
    W00.star<-W00/sum(W00)
```

```r
    W10.star<-W10/sum(W10)
    W01.star<-W01/sum(W01)
    W11.star<-W11/sum(W11)
    mu.00.hat<-sum(W00.star*Y)
    mu.10.hat<-sum(W10.star*Y)
    mu.01.hat<-sum(W01.star*Y)
    mu.11.hat<-sum(W11.star*Y)
    ests.mat[irep,3,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
}
true.mu<-c(2,5,3,6);par(mar=c(3,2,2,2),mfrow=c(2,2))
nvec<-c("OR",expression(tilde(mu)(z)),expression(hat(mu)(z)))
boxplot(ests.mat[,,1],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,0)))
abline(h=true.mu[1],col='red')
boxplot(ests.mat[,,2],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,0)))
abline(h=true.mu[2],col='red')
boxplot(ests.mat[,,3],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,1)))
abline(h=true.mu[3],col='red')
boxplot(ests.mat[,,4],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,1)))
abline(h=true.mu[4],col='red')
```



The outcome regression estimator (OR) is evidently biased whereas the two inverse weighting estimators are unbiased. Consider now a second specification, identical to the first except for the component that specifies that

$$\mathbb{E}^{\mathcal{E}}_{Y|X_1,Z_1,X_2,Z_2}[Y|X_1 = x_1, Z_1 = z_1, X_2 = x_2, Z_2 = z_2] = x_1 + z_1 + x_2 + z_2 + x_1.z_1$$

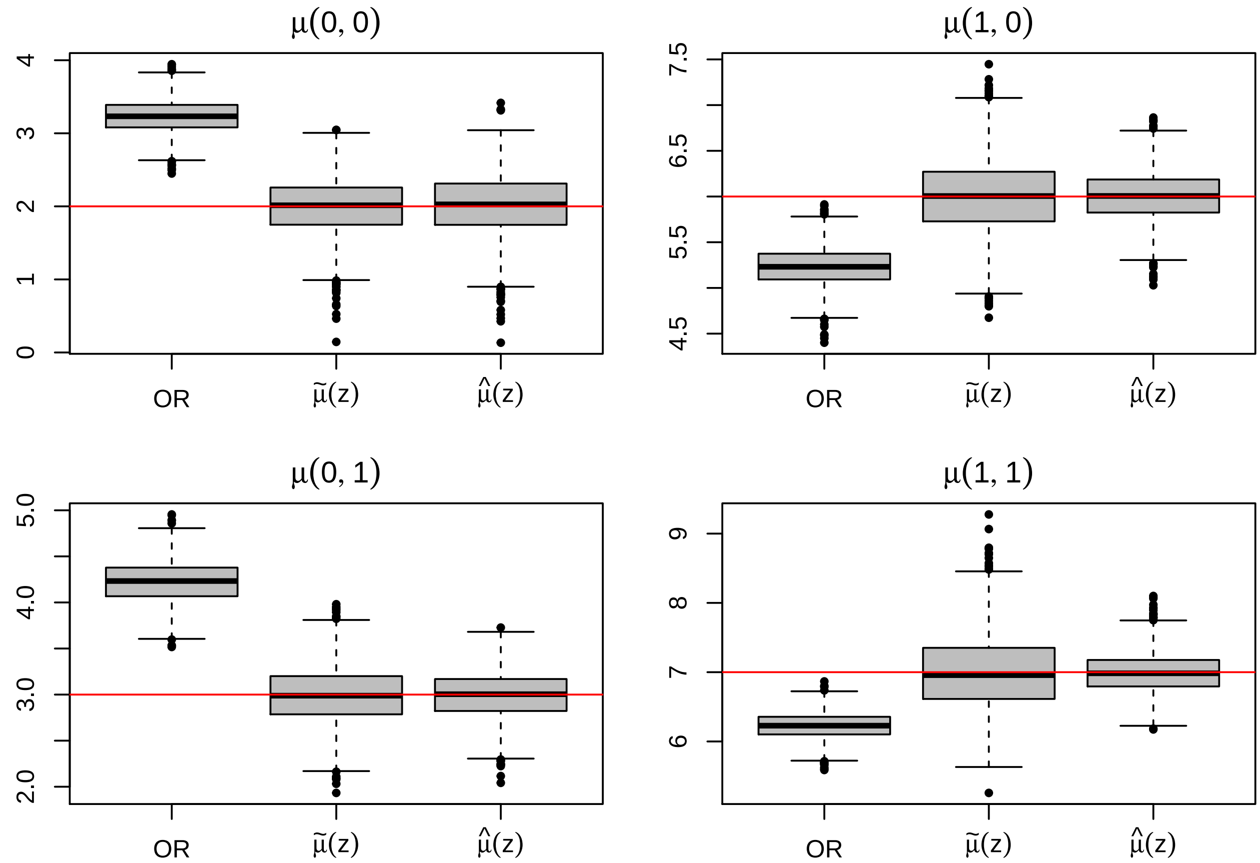
that is, with an interaction between Z_1 and X_1. The causal quantity of interest is now given by

$$\mu(z_1, z_2) = \mathbb{E}^{\mathcal{E}}_{X_1,X_2|Z}[X_1 + X_2 + X_1 z_1|Z = z] + z_1 + z_2 = \mathbb{E}^{\mathcal{E}}_{X_1}[X_1](1 + z_1) + \mathbb{E}^{\mathcal{E}}_{X_1}[\mathbb{E}^{\mathcal{E}}_{X_2|X_1,Z}[X_2|X_1, Z = z]] + z_1 + z_2$$
$$= 1(1 + z_1) + (1 + 2z_1) + z_1 + z_2 = 2 + 4z_1 + z_2$$

Hence

$$\mu(0,0) = 2 \qquad \mu(1,0) = 6 \qquad \mu(0,1) = 3 \qquad \mu(1,1) = 7.$$

```r
set.seed(34)
n<-1000;nreps<-2000
al1<-c(1.0,-0.5);al2<-c(1.0,0.1,0.5,-0.5)
ests.mat<-array(0,c(nreps,3,4))
for(irep in 1:nreps){
    X1<-rnorm(n,1,1)
    eta1<-al1[1]+al1[2]*X1
    ps.1<-expit(eta1)
    Z1<-rbinom(n,1,ps.1)
    X2<-rnorm(n,X1+2*Z1,1)
    eta2<-al2[1]+al2[2]*X1+al2[3]*Z1+al2[4]*X2
    ps.2<-expit(eta2)
    Z2<-rbinom(n,1,ps.2)
    eps<-rnorm(n,0,3)
    Y<-X1+X2+Z1+Z2+X1*Z1+eps
    fit1<-lm(Y~X1+X2+Z1+Z2+X1:Z1)
    mu.00.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=0)))
    mu.10.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=0)))
    mu.01.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=1)))
    mu.11.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=1)))
    ests.mat[irep,1,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    W<-(ps.1^Z1*(1-ps.1)^(1-Z1))*(ps.2^Z2*(1-ps.2)^(1-Z2))
    W00<-(1-Z1)*(1-Z2)/W
    W10<-(Z1)*(1-Z2)/W
    W01<-(1-Z1)*(Z2)/W
    W11<-(Z1)*(Z2)/W
    mu.00.tilde<-mean(W00*Y)
    mu.10.tilde<-mean(W10*Y)
    mu.01.tilde<-mean(W01*Y)
    mu.11.tilde<-mean(W11*Y)
    ests.mat[irep,2,]<-c(mu.00.tilde,mu.10.tilde,mu.01.tilde,mu.11.tilde)
    W00.star<-W00/sum(W00)
    W10.star<-W10/sum(W10)
    W01.star<-W01/sum(W01)
    W11.star<-W11/sum(W11)
    mu.00.hat<-sum(W00.star*Y)
    mu.10.hat<-sum(W10.star*Y)
    mu.01.hat<-sum(W01.star*Y)
    mu.11.hat<-sum(W11.star*Y)
    ests.mat[irep,3,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
}
true.mu<-c(2,6,3,7)
par(mar=c(3,2,2,2),mfrow=c(2,2))
nvec<-c("OR",expression(tilde(mu)(z)),expression(hat(mu)(z)))
boxplot(ests.mat[,,1],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,0)))
abline(h=true.mu[1],col='red')
boxplot(ests.mat[,,2],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,0)))
abline(h=true.mu[2],col='red')
boxplot(ests.mat[,,3],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,1)))
abline(h=true.mu[3],col='red')
boxplot(ests.mat[,,4],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,1)))
abline(h=true.mu[4],col='red')
```



Again, the outcome regression estimator (OR) is evidently biased whereas the two inverse weighting estimators are unbiased. Now a third specification, identical to the first except for the component that specifies that

$$\mathbb{E}^{\mathcal{E}}_{Y|X_1,Z_1,X_2,Z_2}[Y|X_1=x_1,Z_1=z_1,X_2=x_2,Z_2=z_2]=x_1+z_1+x_2+z_2+x_2.z_2$$

with an interaction between Z_2 and X_2. The causal quantity of interest is now given by

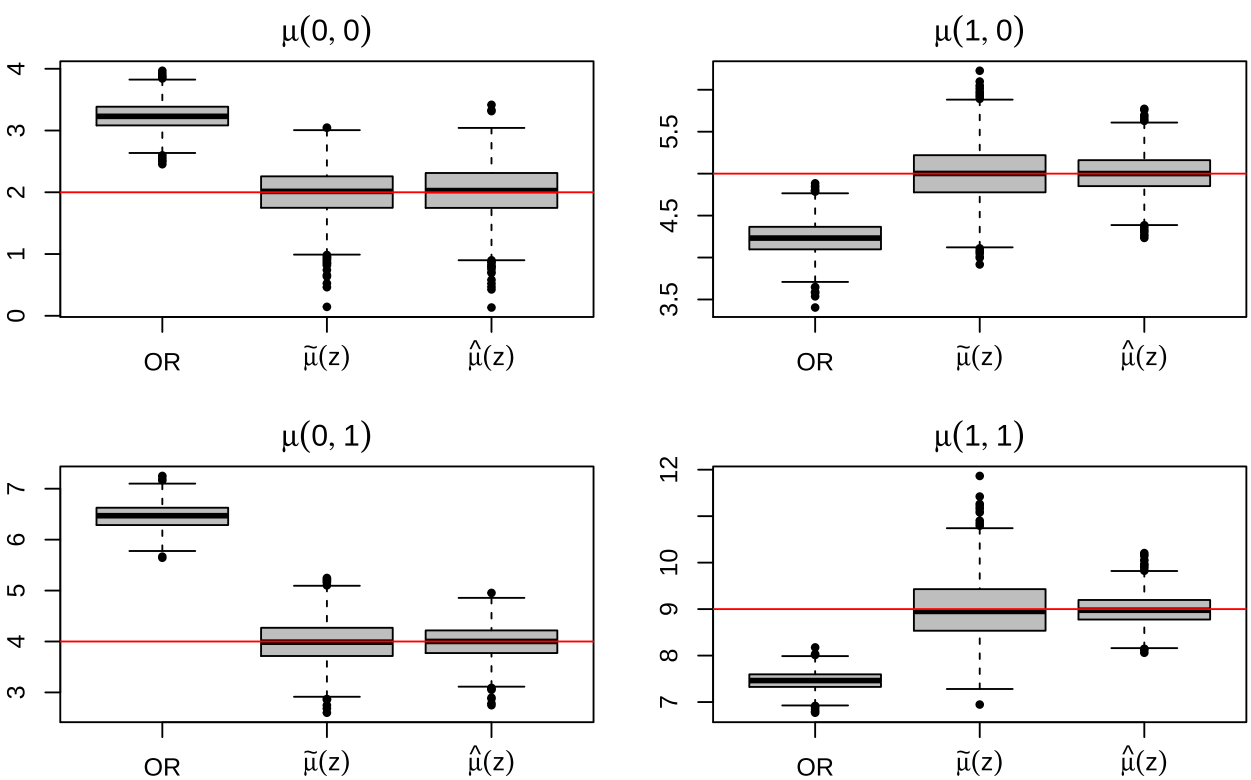
$$\mu(z_1,z_2)=\mathbb{E}^{\mathcal{E}}_{X_1,X_2|Z}[X_1+X_2+X_2z_2|Z=z]+z_1+z_2=\mathbb{E}^{\mathcal{E}}_{X_1}[X_1]+\mathbb{E}^{\mathcal{E}}_{X_1}[\mathbb{E}^{\mathcal{E}}_{X_2|X_1,Z}[X_2(1+z_2)|X_1,Z=z]]+z_1+z_2$$
$$=1+(1+2z_1)(1+z_2)+z_1+z_2=2+3z_1+2z_2+2z_1z_2$$

Hence

$$\mu(0,0)=2 \qquad \mu(1,0)=5 \qquad \mu(0,1)=4 \qquad \mu(1,1)=9.$$

```
set.seed(34);n<-1000;nreps<-2000
al1<-c(1.0,-0.5);al2<-c(1.0,0.1,0.5,-0.5)
ests.mat<-array(0,c(nreps,3,4))
for(irep in 1:nreps){
    X1<-rnorm(n,1,1)
    eta1<-al1[1]+al1[2]*X1
    ps.1<-expit(eta1)
    Z1<-rbinom(n,1,ps.1)
    X2<-rnorm(n,X1+2*Z1,1)
    eta2<-al2[1]+al2[2]*X1+al2[3]*Z1+al2[4]*X2
    ps.2<-expit(eta2)
    Z2<-rbinom(n,1,ps.2)
    eps<-rnorm(n,0,3)
    Y<-X1+X2+Z1+Z2+X2*Z2+eps
    fit1<-lm(Y~X1+X2+Z1+Z2+X2:Z2)
```

```
    mu.00.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=0)))
    mu.10.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=0)))
    mu.01.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=1)))
    mu.11.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=1)))
    ests.mat[irep,1,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    W<-(ps.1^Z1*(1-ps.1)^(1-Z1))*(ps.2^Z2*(1-ps.2)^(1-Z2))
    W00<-(1-Z1)*(1-Z2)/W
    W10<-(Z1)*(1-Z2)/W
    W01<-(1-Z1)*(Z2)/W
    W11<-(Z1)*(Z2)/W
    mu.00.tilde<-mean(W00*Y); mu.10.tilde<-mean(W10*Y)
    mu.01.tilde<-mean(W01*Y); mu.11.tilde<-mean(W11*Y)
    ests.mat[irep,2,]<-c(mu.00.tilde,mu.10.tilde,mu.01.tilde,mu.11.tilde)
    W00.star<-W00/sum(W00)
    W10.star<-W10/sum(W10)
    W01.star<-W01/sum(W01)
    W11.star<-W11/sum(W11)
    mu.00.hat<-sum(W00.star*Y); mu.10.hat<-sum(W10.star*Y)
    mu.01.hat<-sum(W01.star*Y); mu.11.hat<-sum(W11.star*Y)
    ests.mat[irep,3,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
}
true.mu<-c(2,5,4,9);par(mar=c(3,2,2,2),mfrow=c(2,2))
boxplot(ests.mat[,,1],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,0)))
abline(h=true.mu[1],col='red')
boxplot(ests.mat[,,2],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,0)))
abline(h=true.mu[2],col='red')
boxplot(ests.mat[,,3],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,1)))
abline(h=true.mu[3],col='red')
boxplot(ests.mat[,,4],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,1)))
abline(h=true.mu[4],col='red')
```



Again, the outcome regression estimator (OR) is evidently biased whereas the two inverse weighting estimators are unbiased.

Weighted regression: It is also possible to compute the expectation required for $\mu(z)$ using the iterated form

$$\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{1}_z(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)}Y\right] = \mathbb{E}^{\mathcal{O}}_{X,Z}\left[\frac{\mathbb{1}_z(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)}\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X,Z]\right] \tag{3}$$

after proposing a model for $\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X,Z]$. Consider the first simulation with

$$\mathbb{E}^{\mathcal{O}}_{Y|X_1,Z_1,X_2,Z_2}[Y|X_1 = x_1, Z_1 = z_1, X_2 = x_2, Z_2 = z_2] = x_1 + z_1 + x_2 + z_2. \tag{4}$$

By the earlier calculation, we demonstrated that conditioning on X_2 in this model does not yield unbiased estimation as the indirect path from Z_1 to Y is blocked. Instead, consider fitting the model

$$\mathbb{E}^{\mathcal{O}}_{Y|X_1,Z_1,X_2,Z_2}[Y|X_1 = x_1, Z_1 = z_1, X_2 = x_2, Z_2 = z_2] = \beta_0 + \psi_1 z_1 + \psi_2 z_2 = m(z_1, z_2; \beta, \psi) \tag{5}$$

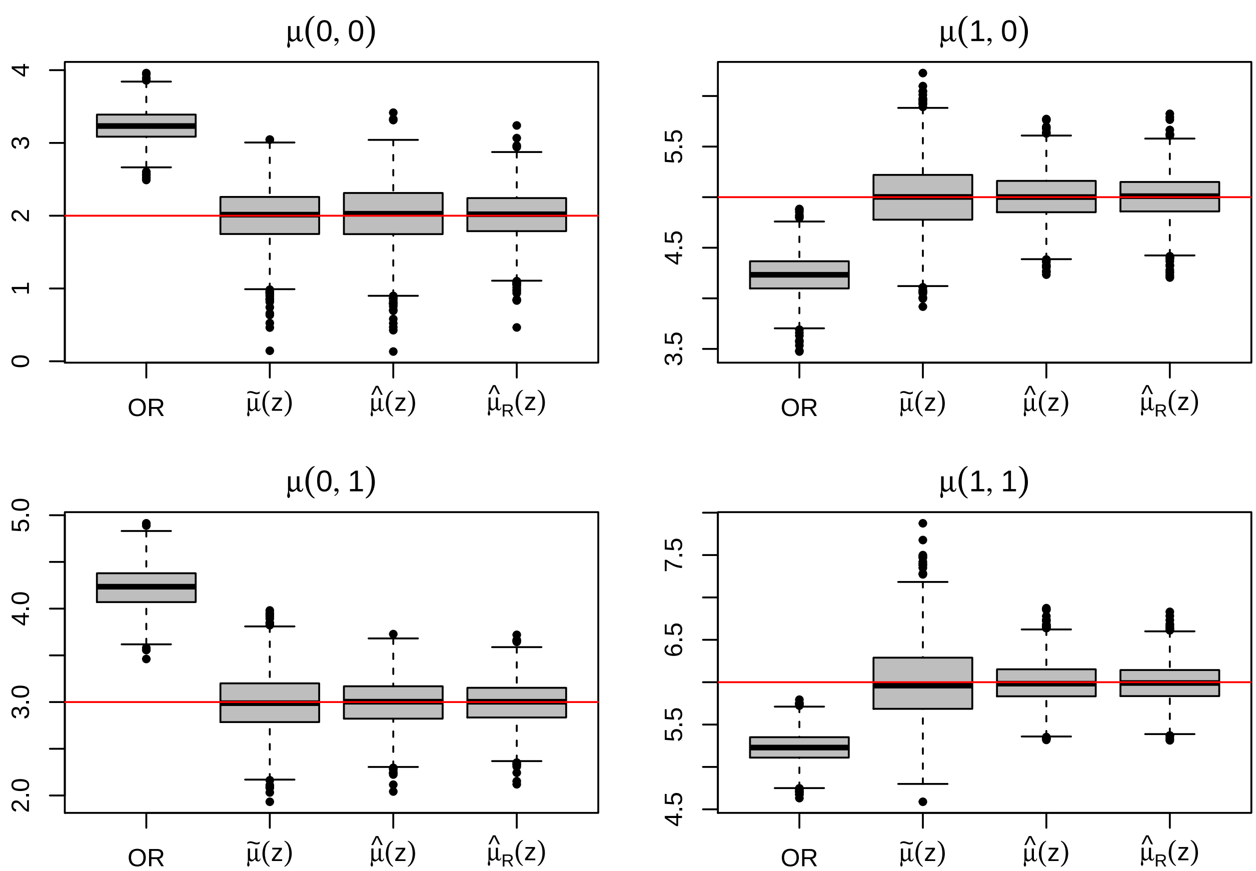
say to the observed data via weighted least squares with weights given by W_z. In the reweighted pseudo-sample, the confounding between X and Z is removed by construction. However, as there is no confounding, the parameter estimators for coefficients ψ_1 and ψ_2 in the model given by (5), are unbiased estimators of the true coefficients in the marginalized version of (4); after marginalization, we saw previously that

$$\mathbb{E}^{\mathcal{O}}_{Y|Z_1,Z_2}[Y|Z_1 = z_1, Z_2 = z_2] = 2 + 3z_1 + z_2 \tag{6}$$

The model in (5) is termed a *marginal structural model*.

```
set.seed(34);n<-1000;nreps<-2000
al1<-c(1.0,-0.5);al2<-c(1.0,0.1,0.5,-0.5)
ests.mat<-array(0,c(nreps,4,4))
psi.ests<-matrix(0,nrow=nreps,ncol=3)
for(irep in 1:nreps){
    X1<-rnorm(n,1,1)
    eta1<-al1[1]+al1[2]*X1
    ps.1<-expit(eta1)
    Z1<-rbinom(n,1,ps.1)
    X2<-rnorm(n,X1+2*Z1,1)
    eta2<-al2[1]+al2[2]*X1+al2[3]*Z1+al2[4]*X2
    ps.2<-expit(eta2)
    Z2<-rbinom(n,1,ps.2)
    eps<-rnorm(n,0,3)
    Y<-X1+X2+Z1+Z2+eps
    fit1<-lm(Y~X1+X2+Z1+Z2)
    mu.00.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=0)))
    mu.10.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=0)))
    mu.01.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=1)))
    mu.11.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=1)))
    ests.mat[irep,1,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    W<-(ps.1^Z1*(1-ps.1)^(1-Z1))*(ps.2^Z2*(1-ps.2)^(1-Z2))
    W00<-(1-Z1)*(1-Z2)/W
    W10<-(Z1)*(1-Z2)/W
    W01<-(1-Z1)*(Z2)/W
    W11<-(Z1)*(Z2)/W
    mu.00.tilde<-mean(W00*Y)
    mu.10.tilde<-mean(W10*Y)
    mu.01.tilde<-mean(W01*Y)
    mu.11.tilde<-mean(W11*Y)
    ests.mat[irep,2,]<-c(mu.00.tilde,mu.10.tilde,mu.01.tilde,mu.11.tilde)
    W00.star<-W00/sum(W00)
    W10.star<-W10/sum(W10)
    W01.star<-W01/sum(W01)
    W11.star<-W11/sum(W11)
    mu.00.hat<-sum(W00.star*Y);    mu.10.hat<-sum(W10.star*Y)
    mu.01.hat<-sum(W01.star*Y);    mu.11.hat<-sum(W11.star*Y)
```

```
    ests.mat[irep,3,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    Wvec<-W00+W01+W10+W11
    fit2<-lm(Y~Z1+Z2, weights=Wvec)
    psi.ests[irep,]<-coef(fit2)
    mu.00.reg<-predict(fit2,newdata=data.frame(Z1=0,Z2=0))
    mu.10.reg<-predict(fit2,newdata=data.frame(Z1=1,Z2=0))
    mu.01.reg<-predict(fit2,newdata=data.frame(Z1=0,Z2=1))
    mu.11.reg<-predict(fit2,newdata=data.frame(Z1=1,Z2=1))
    mu.00.rhat<-mean(mu.00.reg);    mu.10.rhat<-mean(mu.10.reg)
    mu.01.rhat<-mean(mu.01.reg);    mu.11.rhat<-mean(mu.11.reg)
    ests.mat[irep,4,]<-c(mu.00.rhat,mu.10.rhat,mu.01.rhat,mu.11.rhat)
}
true.mu<-c(2,5,3,6)
nvec<-c("OR",expression(tilde(mu)(z)),expression(hat(mu)(z)),expression(hat(mu)[R](z)))
par(mar=c(3,2,2,2),mfrow=c(2,2))
boxplot(ests.mat[,,1],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,0)))
abline(h=true.mu[1],col='red')
boxplot(ests.mat[,,2],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,0)))
abline(h=true.mu[2],col='red')
boxplot(ests.mat[,,3],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,1)))
abline(h=true.mu[3],col='red')
boxplot(ests.mat[,,4],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,1)))
abline(h=true.mu[4],col='red')
```




```
apply(psi.ests,2,mean)

+ [1] 2.0069251 2.9972314 0.9880967
```

This approach can be used if there is an interaction between Z and X in the data generating model. Specifically, consider the simulation with

$$\mathbb{E}^{\mathcal{O}}_{Y|X_1,Z_1,X_2,Z_2}[Y|X_1 = x_1, Z_1 = z_1, X_2 = x_2, Z_2 = z_2] = x_1 + z_1 + x_2 + z_2 + x_2.z_2. \tag{7}$$

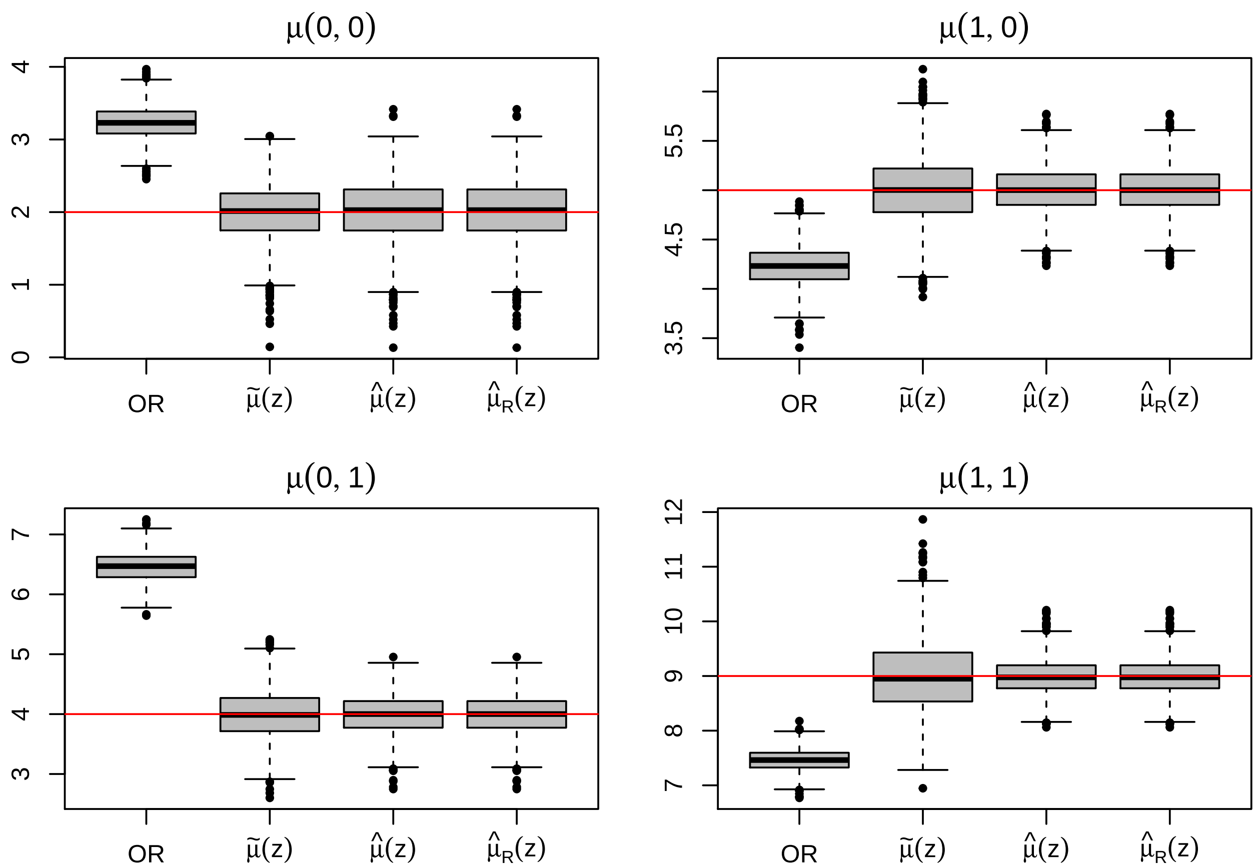
After marginalization, we saw previously that

$$\mathbb{E}^{\mathcal{O}}_{Y|Z_1,Z_2}[Y|Z_1 = z_1, Z_2 = z_2] = 2 + 3z_1 + 2z_2 + 2z_1z_2 \tag{8}$$

and fitting this 'marginal' model yields the correct answer.

```
set.seed(34)
n<-1000;nreps<-2000
al1<-c(1.0,-0.5);al2<-c(1.0,0.1,0.5,-0.5)
ests.mat<-array(0,c(nreps,4,4))
psi.ests<-matrix(0,nrow=nreps,ncol=4)
for(irep in 1:nreps){
    X1<-rnorm(n,1,1)
    eta1<-al1[1]+al1[2]*X1
    ps.1<-expit(eta1)
    Z1<-rbinom(n,1,ps.1)
    X2<-rnorm(n,X1+2*Z1,1)
    eta2<-al2[1]+al2[2]*X1+al2[3]*Z1+al2[4]*X2
    ps.2<-expit(eta2)
    Z2<-rbinom(n,1,ps.2)
    eps<-rnorm(n,0,3)
    Y<-X1+X2+Z1+Z2+Z2*X2+eps
    fit1<-lm(Y~X1+X2+Z1+Z2+Z2:X2)
    mu.00.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=0)))
    mu.10.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=0)))
    mu.01.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=1)))
    mu.11.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=1)))
    ests.mat[irep,1,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    W<-(ps.1^Z1*(1-ps.1)^(1-Z1))*(ps.2^Z2*(1-ps.2)^(1-Z2))
    W00<-(1-Z1)*(1-Z2)/W
    W10<-(Z1)*(1-Z2)/W
    W01<-(1-Z1)*(Z2)/W
    W11<-(Z1)*(Z2)/W
    mu.00.tilde<-mean(W00*Y);    mu.10.tilde<-mean(W10*Y)
    mu.01.tilde<-mean(W01*Y);    mu.11.tilde<-mean(W11*Y)
    ests.mat[irep,2,]<-c(mu.00.tilde,mu.10.tilde,mu.01.tilde,mu.11.tilde)
    W00.star<-W00/sum(W00)
    W10.star<-W10/sum(W10)
    W01.star<-W01/sum(W01)
    W11.star<-W11/sum(W11)
    mu.00.hat<-sum(W00.star*Y);    mu.10.hat<-sum(W10.star*Y)
    mu.01.hat<-sum(W01.star*Y);    mu.11.hat<-sum(W11.star*Y)
    ests.mat[irep,3,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    Wvec<-W00+W01+W10+W11
    fit2<-lm(Y~Z1+Z2+Z1:Z2, weights=Wvec)
    psi.ests[irep,]<-coef(fit2)
    mu.00.reg<-predict(fit2,newdata=data.frame(Z1=0,Z2=0))
    mu.10.reg<-predict(fit2,newdata=data.frame(Z1=1,Z2=0))
    mu.01.reg<-predict(fit2,newdata=data.frame(Z1=0,Z2=1))
    mu.11.reg<-predict(fit2,newdata=data.frame(Z1=1,Z2=1))
    mu.00.rhat<-mean(mu.00.reg);    mu.10.rhat<-mean(mu.10.reg)
    mu.01.rhat<-mean(mu.01.reg);    mu.11.rhat<-mean(mu.11.reg)
    ests.mat[irep,4,]<-c(mu.00.rhat,mu.10.rhat,mu.01.rhat,mu.11.rhat)
}
true.mu<-c(2,5,4,9)
nvec<-c("OR",expression(tilde(mu)(z)),expression(hat(mu)(z)),expression(hat(mu)[R](z)))
par(mar=c(3,2,2,2),mfrow=c(2,2))
```

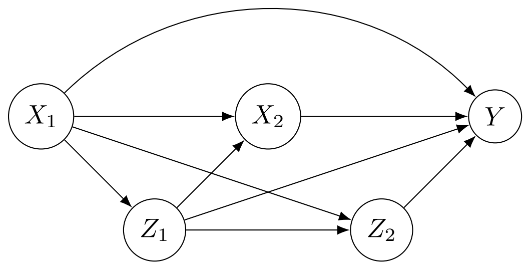
```r
boxplot(ests.mat[,,1],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,0)))
abline(h=true.mu[1],col='red')
boxplot(ests.mat[,,2],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,0)))
abline(h=true.mu[2],col='red')
boxplot(ests.mat[,,3],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,1)))
abline(h=true.mu[3],col='red')
boxplot(ests.mat[,,4],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,1)))
abline(h=true.mu[4],col='red')
```




```r
apply(psi.ests,2,mean)

+ [1] 2.012241 2.990797 1.984063 2.003081

apply(psi.ests,2,var)

+ [1] 0.1772699 0.2316459 0.2710176 0.4286061
```

Note that the marginal model needs to be correctly specified in order for the inferences to be correct.

Stabilized weights: It is possible to utilize so-called *stabilized* weights in the marginal structural model. In this version of the analysis, there is conditioning on baseline confounders X_1; the corresponding DAG makes it clear that conditioning on X_1 does not block any paths other than the confounding path $Z_1 \to X_1 \to Y$.



Because of the universal conditioning on X_1, the corresponding causal quantities to be considered are based on

$$\mu_S(z, x_1) = \mathbb{E}^{\mathcal{E}}_{Y|X_1,Z}[Y|X_1 = x_1, Z = z].$$

We now have by the usual arguments,

$$\mu_S(z, x_1) = \frac{\displaystyle\iiint \mathbb{1}_z(t)\ y\ \frac{f^{\mathcal{E}}_{Z|X_1}(t|x_1)}{f^{\mathcal{O}}_{Z|X_1,X_2}(t|x_1, x_2)} f^{\mathcal{O}}_{X_1,X_2,Z}(x_1, x_2, t)\ dy\ dx_2\ dt}{\displaystyle\iint \mathbb{1}_z(t) \frac{f^{\mathcal{E}}_{Z|X_1}(t|x_1)}{f^{\mathcal{O}}_{Z|X_1,X_2}(t|x_1, x_2)} f^{\mathcal{O}}_{X_1,X_2,Z}(x_1, x_2, t)\ dx_2\ dt}$$

$$\equiv \frac{\displaystyle\iiint \mathbb{1}_z(t)\mathbb{1}_{x_1}(u)\ y\ \frac{f^{\mathcal{E}}_{Z|X_1}(t|u)}{f^{\mathcal{O}}_{Z|X_1,X_2}(t|u, x_2)} f^{\mathcal{O}}_{X_1,X_2,Z}(u, x_2, t)\ dt\ dy\ du\ dx_2\ dt}{\displaystyle\iint \mathbb{1}_z(t)\mathbb{1}_{x_1}(u) \frac{f^{\mathcal{E}}_{Z|X_1}(t|u)}{f^{\mathcal{O}}_{Z|X_1,X_2}(t|u, x_2)} f^{\mathcal{O}}_{X_1,X_2,Z}(u, x_2, t)\ du\ dx_2\ dt}$$

Inference would then proceed in the usual way, based on sample averages: for example,

$$\widehat{\mu}_S(z, x_1) = \frac{\displaystyle\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_z(Z_i)\mathbb{1}_{x_1}(X_i) \frac{f^{\mathcal{E}}_{Z|X_1}(Z_i|X_{i1})}{f^{\mathcal{O}}_{Z|X}(Z_i|X_i)} Y}{\displaystyle\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_z(Z_i)\mathbb{1}_{x_1}(X_{i1}) \frac{f^{\mathcal{E}}_{Z|X_1}(Z_i|X_{i1})}{f^{\mathcal{O}}_{Z|X}(Z_i|X_i)}}$$

However, if X_1 is high-dimensional and/or continuous-valued, the number of terms in the sums may be very small, and so typically regression methods are favoured to introduce the conditioning on X_1; we may again pick a 'working' marginal structural (regression) model and solve using weighted least squares with weights

$$\mathbb{1}_z(Z_i)\mathbb{1}_{x_1}(X_i) \frac{f^{\mathcal{E}}_{Z|X_1}(Z_i|X_{i1})}{f^{\mathcal{O}}_{Z|X}(Z_i|X_i)}. \tag{9}$$

Note that for the ith data point, we might propose a marginal model

$$\mathbb{E}^{\mathcal{O}}_{Y|X_1,Z_1,X_2,Z_2}[Y|X_1 = x_{i1}, Z_1 = z_{i1}, Z_2 = z_{i2}] = \beta_0 + \beta_1 x_{i1} + \psi_1 z_{i1} + \psi_2 z_{i2} \tag{10}$$

and so the indicator function $\mathbb{1}_{x_1}(X_{i1})$ in (9) always takes the value 1.

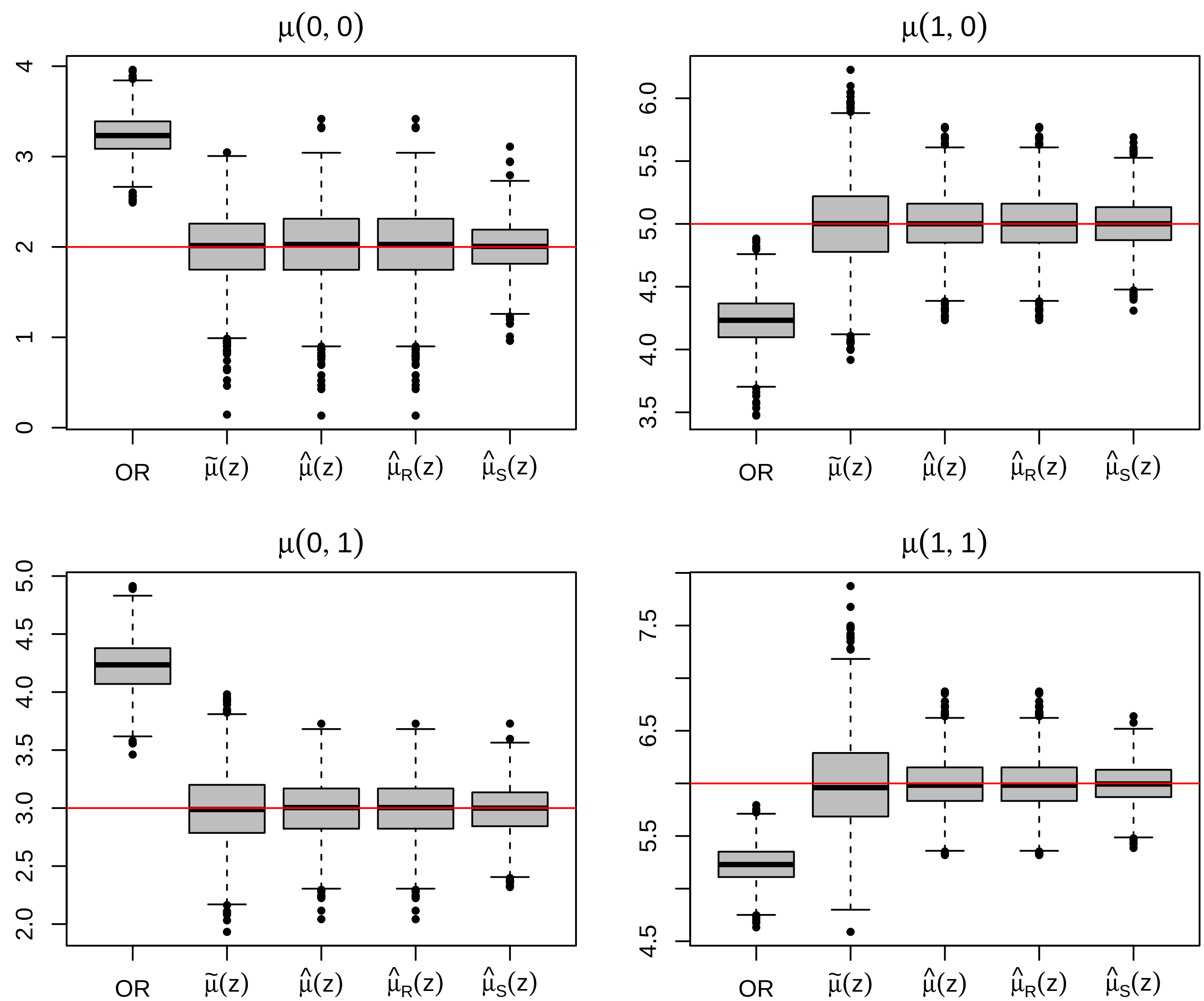
From (9), it is clear that a model for $f^{\mathcal{E}}_{Z|X_1}(Z_i|X_{i1})$ is needed. Also, we have that

$$\frac{f^{\mathcal{E}}_{Z|X_1}(Z|X_1)}{f^{\mathcal{O}}_{Z|X}(Z|X)} = \frac{f^{\mathcal{E}}_{Z_1|X_1}(Z_1|X_1) \times f^{\mathcal{E}}_{Z_2|X_1,Z_1}(Z_2|X_1, Z_1)}{f^{\mathcal{O}}_{Z_1|X_1}(Z_1|X_1) \times f^{\mathcal{O}}_{Z_2|X_1,Z_1,X_2}(Z_2|X_1, Z_1, X_2)} \equiv \frac{f^{\mathcal{E}}_{Z_2|X_1,Z_1}(Z_2|X_1, Z_1)}{f^{\mathcal{O}}_{Z_2|X_1,Z_1,X_2}(Z_2|X_1, Z_1, X_2)} \tag{11}$$

as it can be seen from the DAG that the terms $f^{\mathcal{E}}_{Z_1|X_1}(Z_1|X_1)$ and $f^{\mathcal{O}}_{Z_1|X_1}(Z_1|X_1)$ cancel.

```
set.seed(34)
n<-1000;nreps<-2000
al1<-c(1.0,-0.5);al2<-c(1.0,0.1,0.5,-0.5)
ests.mat<-array(0,c(nreps,5,4))
psi.ests<-matrix(0,nrow=nreps,ncol=4)
psi.ests.s<-matrix(0,nrow=nreps,ncol=5)
for(irep in 1:nreps){
    X1<-rnorm(n,1,1)
    eta1<-al1[1]+al1[2]*X1
    ps.1<-expit(eta1)
    Z1<-rbinom(n,1,ps.1)
    X2<-rnorm(n,X1+2*Z1,1)
    eta2<-al2[1]+al2[2]*X1+al2[3]*Z1+al2[4]*X2
    ps.2<-expit(eta2)
    Z2<-rbinom(n,1,ps.2)
    eps<-rnorm(n,0,3)
    Y<-X1+X2+Z1+Z2+eps
    fit1<-lm(Y~X1+X2+Z1+Z2)
    mu.00.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=0)))
    mu.10.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=0)))
    mu.01.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=0,Z2=1)))
    mu.11.hat<-mean(predict(fit1,newdata=data.frame(X1,X2,Z1=1,Z2=1)))
    ests.mat[irep,1,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    W<-(ps.1^Z1*(1-ps.1)^(1-Z1))*(ps.2^Z2*(1-ps.2)^(1-Z2))
    W00<-(1-Z1)*(1-Z2)/W
    W10<-(Z1)*(1-Z2)/W
    W01<-(1-Z1)*(Z2)/W
    W11<-(Z1)*(Z2)/W
    mu.00.tilde<-mean(W00*Y)
    mu.10.tilde<-mean(W10*Y)
    mu.01.tilde<-mean(W01*Y)
    mu.11.tilde<-mean(W11*Y)
    ests.mat[irep,2,]<-c(mu.00.tilde,mu.10.tilde,mu.01.tilde,mu.11.tilde)
    W00.star<-W00/sum(W00)
    W10.star<-W10/sum(W10)
    W01.star<-W01/sum(W01)
    W11.star<-W11/sum(W11)
    mu.00.hat<-sum(W00.star*Y)
    mu.10.hat<-sum(W10.star*Y)
    mu.01.hat<-sum(W01.star*Y)
    mu.11.hat<-sum(W11.star*Y)
    ests.mat[irep,3,]<-c(mu.00.hat,mu.10.hat,mu.01.hat,mu.11.hat)
    Wvec<-W00+W01+W10+W11
    fit2<-lm(Y~Z1+Z2+Z1:Z2, weights=Wvec)
    psi.ests[irep,]<-coef(fit2)
    mu.00.reg<-predict(fit2,newdata=data.frame(Z1=0,Z2=0))
    mu.10.reg<-predict(fit2,newdata=data.frame(Z1=1,Z2=0))
    mu.01.reg<-predict(fit2,newdata=data.frame(Z1=0,Z2=1))
    mu.11.reg<-predict(fit2,newdata=data.frame(Z1=1,Z2=1))
    mu.00.rhat<-mean(mu.00.reg);    mu.10.rhat<-mean(mu.10.reg)
    mu.01.rhat<-mean(mu.01.reg);    mu.11.rhat<-mean(mu.11.reg)
    ests.mat[irep,4,]<-c(mu.00.rhat,mu.10.rhat,mu.01.rhat,mu.11.rhat)
    ps.num<-fitted(glm(Z2~X1+Z1,family=binomial))
    ps.den<-fitted(glm(Z2~X1+X2+Z1,family=binomial))
    W<-(ps.num^Z2*(1-ps.num)^(1-Z2))/(ps.den^Z2*(1-ps.den)^(1-Z2))
    fit3<-lm(Y~X1+Z1+Z2+Z1:Z2, weights=W)
    psi.ests.s[irep,]<-coef(fit3)
    mu.00.reg<-predict(fit3,newdata=data.frame(X1,Z1=0,Z2=0))
    mu.10.reg<-predict(fit3,newdata=data.frame(X1,Z1=1,Z2=0))
    mu.01.reg<-predict(fit3,newdata=data.frame(X1,Z1=0,Z2=1))
    mu.11.reg<-predict(fit3,newdata=data.frame(X1,Z1=1,Z2=1))
```

```r
    mu.00.rhat<-mean(mu.00.reg);     mu.10.rhat<-mean(mu.10.reg)
    mu.01.rhat<-mean(mu.01.reg);     mu.11.rhat<-mean(mu.11.reg)
    ests.mat[irep,5,]<-c(mu.00.rhat,mu.10.rhat,mu.01.rhat,mu.11.rhat)
}
true.mu<-c(2,5,3,6)
nvec<-c("OR",expression(tilde(mu)(z)),expression(hat(mu)(z)),expression(hat(mu)[R](z)),
             expression(hat(mu)[S](z)))
par(mar=c(3,2,2,2),mfrow=c(2,2))
boxplot(ests.mat[,,1],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,0)))
abline(h=true.mu[1],col='red')
boxplot(ests.mat[,,2],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,0)))
abline(h=true.mu[2],col='red')
boxplot(ests.mat[,,3],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(0,1)))
abline(h=true.mu[3],col='red')
boxplot(ests.mat[,,4],names=nvec,pch=19,cex=0.6,col='gray',main=expression(mu(1,1)))
abline(h=true.mu[4],col='red')
```




```r
apply(psi.ests,2,mean)

+ [1] 2.012241008 2.990796929 0.981646229 0.008283561

apply(psi.ests.s,2,mean)

+ [1] -0.001705085  2.000457627  3.002125472  0.992038458  0.003833490
```