CAUSAL ADJUSTMENT METHODS USING THE PROPENSITY SCORE

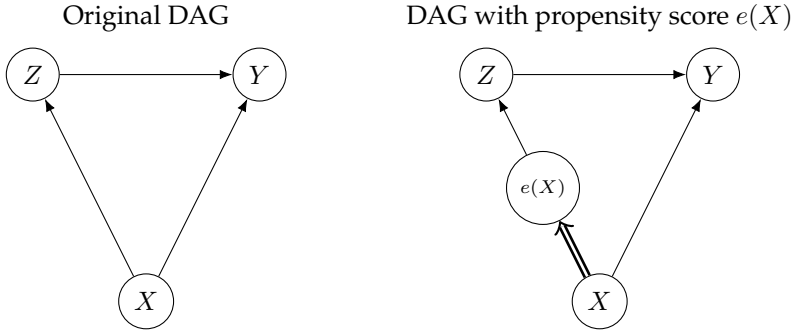
The most common causal inference problem is to estimate the causal effect of treatment Z on outcome Y in the presence of confounders X. The causal effect is defined via the expectation under the experimental measure ε.

$$\mu(z) = \mathbb{E}^{\varepsilon}_{Y|Z}[Y|Z = z] \qquad z \in \mathcal{Z}.$$

The propensity score for binary exposure, denoted $e(X)$, is defined via the observational conditional model for Z given confounders X, as

$$e(x) = f^{\mathcal{O}}_{Z|X}(1|x) = \Pr^{\mathcal{O}}_{Z|X}[Z = 1|X = x]$$

with $e(X)$ being the corresponding random variable. The propensity score is a *balancing score*, that is, we have that $Z \perp\!\!\!\perp X \mid e(X)$.



Therefore, an analysis that conditions on $e(X)$ will block the biasing (backdoor, confounding) path between Z and Y that goes via X.

Several different methods based on the propensity score can be used for causal adjustment:

- **Stratification :** The propensity score space is partitioned into J regions or *strata*, and the 'local' causal effect is estimated in each, and then combined in an average to estimate $\mu(z)$
- **Matching :** The outcomes for individuals with different Z values but equal $e(X)$ values (that is, the individuals are *matched* on their propensity score values) are compared.
- **Regression :** The propensity score is conditioned upon using a *regression model*.
- **Inverse Weighting :** Using an 'importance sampling' *reweighting* via the propensity score of the Y values observed under $\mathcal{O}$, a pseudo-sample is created that can be treated as a sample collected under $\mathcal{E}$ in which exposure is balanced.

Consider the following example where we generate according to the parametric model

- $X \sim \text{Normal}_3(\mu, \Sigma)$ with

$$\mu = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 0.25 & 0.00 & 0.00 \\ 0.00 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.75 \end{bmatrix} \begin{bmatrix} 1.0 & 0.9 & -0.1 \\ 0.9 & 1.0 & -0.2 \\ -0.1 & -0.2 & 1.0 \end{bmatrix} \begin{bmatrix} 0.25 & 0.00 & 0.00 \\ 0.00 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.75 \end{bmatrix}$$

- $Z|X = x \sim Bernoulli(e(x; \alpha))$, with

$$e(x; \alpha) = \frac{\exp\{\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_1 x_2\}}{1 + \exp\{\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_1 x_2\}}$$

  with $\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)^\top = (6.0, -0.2, 0.7, 2)^\top$.

- $Y|X = x, Z = z \sim \text{Normal}(\mu(x, z; \beta, \psi), \sigma^2)$, with

$$\mu(x, z) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + z(\psi_0 + \psi_1 x_1 + \psi_2 x_2 + \psi_3 x_1 x_2) = \mu_0(x; \beta) + z\mu_1(x; \psi)$$

  say, with $\sigma^2 = 0.1^2$, and

$$\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^\top = (10, -2.0, 1.2, 0.6)^\top \qquad \psi = (\psi_0, \psi_1, \psi_2, \psi_3)^\top = (1, 1, 1, 1)^\top$$

  In this model, X_1 and X_2 are confounders, and X_3 is a spurious variable unrelated to Y or Z.

```r
#Data simulation
library(MASS);library(xtable)
set.seed(23987)
n<-1000
D<-diag(c(0.25,0.5,0.75))
Mu<-c(1,-2,-1)
Sigma<-D %*% (matrix(c(1,0.9,-0.1,0.9,1,-0.2,-0.1,-0.2,1),3,3) %*% D)
X<-mvrnorm(n,mu=Mu,Sigma)
al<-c(6.0,-0.2,0.7,2)
Xal<-cbind(1,X[,1],X[,2],X[,1]*X[,2])
expit<-function(x){return(1/(1+exp(-x)))}
ps.true<-expit(Xal %*% al)
Z<-rbinom(n,1,ps.true)
be<-c(10,-2.0,1.2,0.6)
Xb<-cbind(1,X[,1],X[,2],X[,1]*X[,2])
mu0<- Xb %*% be
psi<-c(1,1,1,1)
Xp<-cbind(1,X[,1],X[,2],X[,1]*X[,2])
mu1<- (Xp %*% psi)
eta<-mu0 + Z * mu1 ; sig<-0.1
Y<-rnorm(n,eta,sig)
X1<-X[,1];X2<-X[,2];X3<-X[,3]
true.vals<-c(be,psi)[c(1,2,3,5,4,6,7,8)]
eX<-as.vector(Z-ps.true)
```

In this linear model, we have that the average causal effect can be computed directly in terms of the model parameters:

$$\mu(z) = \mathbb{E}^{\varepsilon}_{Y|Z}[Y|Z=z] \equiv \mathbb{E}^{\mathcal{O}}_{X}\left[\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X,Z=z]\right] = \mathbb{E}^{\mathcal{O}}_{X}\left[\mu(X,z)\right]$$

so in this case

$$\begin{aligned}\mu(z) &= \mathbb{E}^{\mathcal{O}}_{X}\left[\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + z(\psi_0 + \psi_1 X_1 + \psi_2 X_2 + \psi_3 X_1 X_2)\right] \\ &= \beta_0 + \beta_1 \mathbb{E}[X_1] + \beta_2 \mathbb{E}[X_2] + \beta_3 \mathbb{E}[X_1 X_2] + z(\psi_0 + \psi_1 \mathbb{E}[X_1] + \psi_2 \mathbb{E}[X_2] + \psi_3 \mathbb{E}[X_1 X_2])\end{aligned}$$

Here we have that

$$\mathbb{E}[XX^\top] = \begin{bmatrix} \mathbb{E}[X_1^2] & \mathbb{E}[X_1X_2] & \mathbb{E}[X_1X_3] \\ \mathbb{E}[X_1X_2] & \mathbb{E}[X_2^2] & \mathbb{E}[X_2X_3] \\ \mathbb{E}[X_1X_3] & \mathbb{E}[X_2X_2] & \mathbb{E}[X_3^2] \end{bmatrix} = \Sigma + \mu\mu^\top = \begin{bmatrix} 0.0625 & 0.1125 & -0.01875 \\ 0.1125 & 0.25 & -0.075 \\ -0.01875 & -0.075 & 0.5625 \end{bmatrix} + \begin{bmatrix} 1 & -2 & -1 \\ -2 & 4 & 2 \\ -1 & 2 & 1 \end{bmatrix}$$

which yields

```r
(X.M<-Sigma + Mu %*% t(Mu))
```

```
##          [,1]     [,2]     [,3]
## [1,]  1.06250 -1.8875 -1.01875
## [2,] -1.88750  4.2500  1.92500
## [3,] -1.01875  1.9250  1.56250
```

so therefore $\mu(0)$ and $\mu(1)$ are

```r
(mu0<-sum(be*c(1,Mu[1],Mu[2],X.M[1,2])))
```

```
## [1] 4.4675
```

```r
(mu1<-mu0 + sum(psi*c(1,Mu[1],Mu[2],X.M[1,2])))
```

```
## [1] 2.58
```

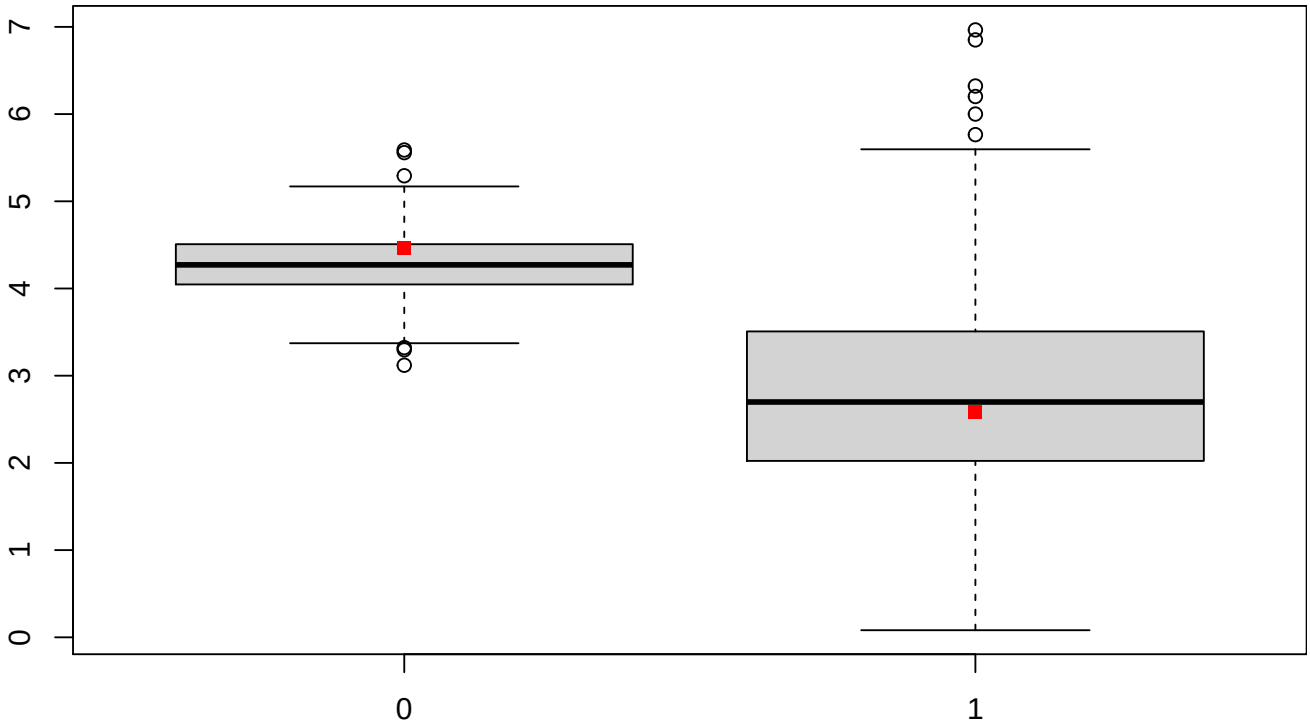
and hence the 'causal effect' is

$$\mu(1) - \mu(0) = 2.58 - 4.4675 = -1.8875.$$

1. Unadjusted analysis

If we try to analyze the data as if they arose from an experimental study, as in an unadjusted analysis, it is evident that incorrect inferences are drawn.

```
par(mar=c(2,2,0,0))
boxplot(Y~Z)
points(1:2,c(mu0,mu1),col='red',pch=15)
```




```
c(mean(Y[Z==0]),mean(Y[Z==1]),mean(Y[Z==1])-mean(Y[Z==0]))

+ [1]  4.288448  2.770900 -1.517548
```

Thus the estimated causal effect is -1.5175483.

2. Outcome Regression

2.1 Correct Specification

A regression analysis using a correctly specified model returns correct parameter estimation.

```
fit0<-lm(Y~X1+X2+X1:X2+Z+Z:X1+Z:X2+Z:X1:X2)
round(cbind(coef(summary(fit0)),true.vals),6)

+              Estimate Std. Error    t value Pr(>|t|) true.vals
+ (Intercept) 10.059622   0.144308  69.709475        0      10.0
+ X1          -2.064657   0.115836 -17.823956        0      -2.0
+ X2           1.236939   0.054910  22.526689        0       1.2
+ Z            0.989901   0.168050   5.890520        0       1.0
+ X1:X2        0.559755   0.050561  11.070900        0       0.6
+ X1:Z         1.024435   0.130322   7.860769        0       1.0
+ X2:Z         0.984275   0.062645  15.711978        0       1.0
+ X1:X2:Z      1.023784   0.056071  18.258715        0       1.0
```

From this fit, it is possible to estimate the causal quantities using an average prediction

$$\widehat{\mu}(z) = \frac{1}{n}\sum_{i=1}^{n}\mu(x_i, z; \widehat{\beta}, \widehat{\psi}) \tag{1}$$

```
mu0.0<-mean(predict(fit0,newdata=data.frame(X1,X2,Z=0)))
mu1.0<-mean(predict(fit0,newdata=data.frame(X1,X2,Z=1)))
c(mu0.0,mu1.0,mu1.0-mu0.0)

+ [1]  4.454317  2.539963 -1.914354
```

Notice, however, that what is being computed is **not the average of the predictions for the observed z values** in the two cases for $z_i = 1$ and $z_i = 0$, which would be respectively

$$\frac{\sum_{i=1}^{n} z_i\mu(x_i, 1; \widehat{\beta}, \widehat{\psi})}{\sum_{i=1}^{n} z_i} \qquad \frac{\sum_{i=1}^{n}(1-z_i)\mu(x_i, 0; \widehat{\beta}, \widehat{\psi})}{\sum_{i=1}^{n}(1-z_i)}.$$

Such a procedure would not estimate the required quantities correctly:

```
m0<-sum((1-Z)*predict(fit0))/sum(1-Z)
m1<-sum(Z*predict(fit0))/sum(Z)
c(m0,m1,m1-m0)

+ [1]  4.288448  2.770900 -1.517548
```

This is due to the confounding present in the structure of the joint distribution,

2.2 Incorrect specification

An incorrectly specified model leads to incorrect estimation, although the estimated causal parameters can still potentially be recovered. For example, if the interaction terms are omitted, although it is not possible to estimate individual coefficients correctly, the causal quantities of interest are estimated reasonably well.

```
fit1<-lm(Y~X1+X2+Z+Z:X1+Z:X2)
round(coef(summary(fit1)),6)

+              Estimate Std. Error    t value Pr(>|t|)
+ (Intercept) 11.206903   0.233694  47.955492        0
+ X1          -3.216690   0.118417 -27.164058        0
+ X2           1.775659   0.059201  29.993931        0
+ Z            3.109948   0.285001  10.912047        0
+ X1:Z        -0.927392   0.145289  -6.383066        0
+ X2:Z         2.018438   0.072729  27.752731        0

mu0.1<-mean(predict(fit1,newdata=data.frame(X1,X2,Z=0)))
mu1.1<-mean(predict(fit1,newdata=data.frame(X1,X2,Z=1)))
c(mu0.1,mu1.1,mu1.1-mu0.1)

+ [1]  4.435704  2.565313 -1.870391
```

The omission of the `X1:X2` and `Z:X1:X2` interactions does not lead to poor results in this case because the magnitude of the effect of these terms is relatively small.

If either X_1 or X_2 or both is omitted, there is no correct recovery.

```
fit2<-lm(Y~X1+Z+Z:X1)
mu0.2<-mean(predict(fit2,newdata=data.frame(X1,Z=0)))
mu1.2<-mean(predict(fit2,newdata=data.frame(X1,Z=1)))
c(mu0.2,mu1.2,mu1.2-mu0.2)

+ [1]  4.286084  2.744744 -1.541341

fit3<-lm(Y~X2+Z+Z:X2)
mu0.3<-mean(predict(fit3,newdata=data.frame(X1,Z=0)))
mu1.3<-mean(predict(fit3,newdata=data.frame(X1,Z=1)))
c(mu0.3,mu1.3,mu1.3-mu0.3)

+ [1]  4.331889  2.647514 -1.684375

fit4<-lm(Y~Z)
mu0.4<-mean(predict(fit4,newdata=data.frame(X1,Z=0)))
mu1.4<-mean(predict(fit4,newdata=data.frame(X1,Z=1)))
c(mu0.4,mu1.4,mu1.4-mu0.4)

+ [1]  4.288448  2.770900 -1.517548
```

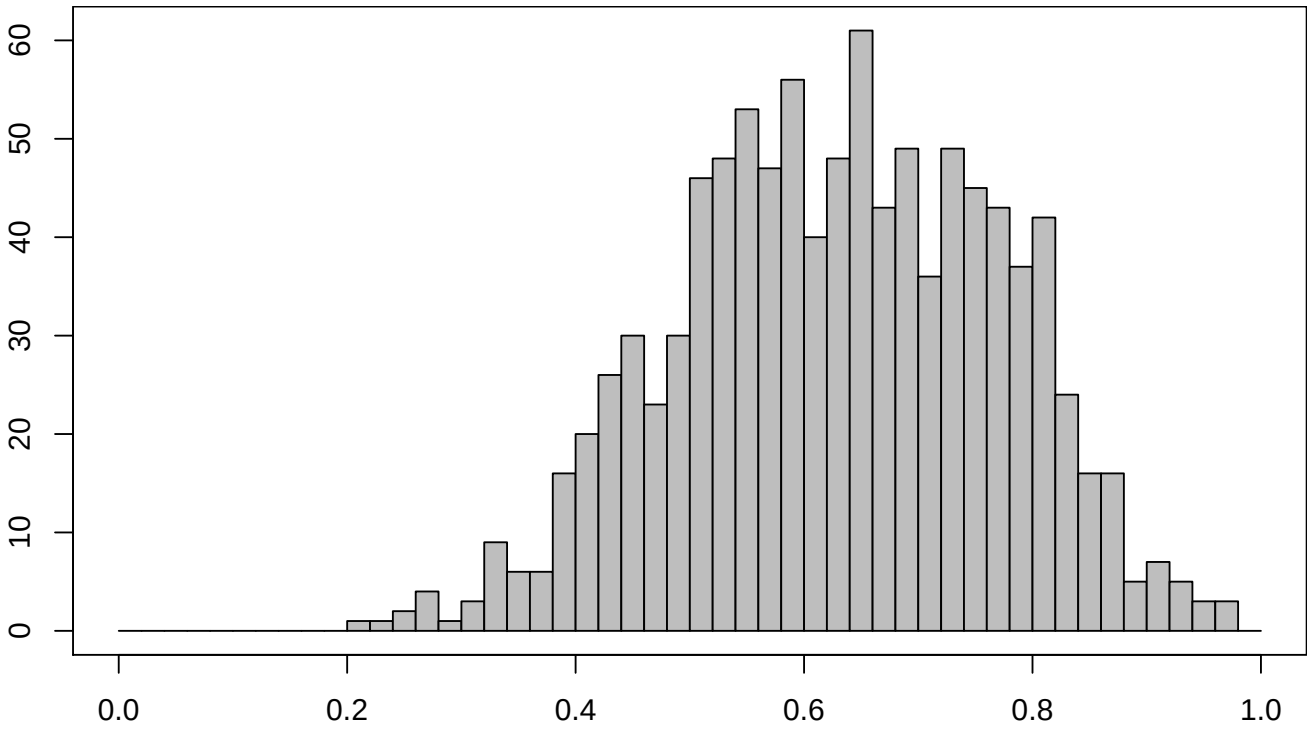
3. Propensity score stratification

For propensity score stratification, we seek to first construct strata of X, $\mathcal{X}_j, j = 1, \ldots, J$ defined by propensity score values, then estimate the causal quantities within each stratum, and then combine across strata.

3.1 True propensity score

First, we assume the true propensity score values are known and are given as in the simulation.

```
par(mar=c(2,2,0,0))
hist(ps.true,main='',col='gray',breaks=seq(0,1,by=0.02));box()
```



We may define the strata by taking percentiles of the propensity score, for example, quintiles.

```
nstrata<-5
ps.quint.true<-quantile(ps.true,probs=seq(0,1,by=1/nstrata))
ps.quint.true

+        0%       20%       40%       60%       80%      100%
+ 0.2126362 0.5120989 0.5914952 0.6715642 0.7601679 0.9762403

ps.strat.true<-as.numeric(cut(ps.true,ps.quint.true,include.lowest=TRUE))
table(Z,ps.strat.true)

+    ps.strat.true
+ Z     1   2   3   4   5
+   0 120  95  73  46  29
+   1  80 105 127 154 171
```

Stratification by the quintiles of the propensity score allows a reasonable number of both $Z = 0$ and $Z = 1$ cases in each of the strata.

```
mu.strat.true<-matrix(0,ncol=nstrata,nrow=3)
for(j in 1:nstrata){
    mu.strat.true[1,j]<-mean(Y[Z==0 & ps.strat.true == j])
    mu.strat.true[2,j]<-mean(Y[Z==1 & ps.strat.true == j])
    mu.strat.true[3,j]<-mu.strat.true[2,j]-mu.strat.true[1,j]
}
mu.strat.true

+           [,1]      [,2]      [,3]      [,4]       [,5]
+ [1,]  3.918955  4.234434  4.446315  4.682338  4.9721481
+ [2,]  1.293308  1.929582  2.512729  2.960797  3.9994921
+ [3,] -2.625647 -2.304852 -1.933586 -1.721540 -0.9726559
```

We can then estimate the causal quantities by averaging within the rows of this matrix to compute

$$\widehat{\mu}(z) = \sum_{j=1}^{J} \widehat{\mu}^{\mathcal{X}_j}(z) \Pr[X \in \mathcal{X}_j]$$

```
apply(mu.strat.true,1,mean)

+ [1]  4.450838  2.539182 -1.911656
```

3.2 Fitted propensity score

In practice, we do not know the true values of the propensity score, and so typically we would estimate them from the observed data. If we assume the correct form for the model, but not the values of the parameters, we can estimate them from the observed data using a logistic regression model and the `glm` function.

```
ps.fit.c<-fitted(glm(Z~X1+X2+X1:X2,family=binomial))
```

We can then proceed with an identical analysis, replacing the true PS values with the fitted ones.

```
ps.quint.fit.c<-quantile(ps.fit.c,probs=seq(0,1,by=1/nstrata))
ps.quint.fit.c

+        0%       20%       40%       60%       80%      100%
+ 0.1718558 0.4995522 0.5948874 0.6877045 0.7888924 0.9928991
```

```
ps.strat.fit.c<-as.numeric(cut(ps.fit.c,ps.quint.fit.c,include.lowest=TRUE))
table(Z,ps.strat.fit.c)

+    ps.strat.fit.c
+ Z     1   2   3   4   5
+   0 120  96  71  49  27
+   1  80 104 129 151 173

mu.strat.fit.c<-matrix(0,ncol=nstrata,nrow=3)
for(j in 1:nstrata){
    mu.strat.fit.c[1,j]<-mean(Y[Z==0 & ps.strat.fit.c == j])
    mu.strat.fit.c[2,j]<-mean(Y[Z==1 & ps.strat.fit.c == j])
    mu.strat.fit.c[3,j]<-mu.strat.fit.c[2,j]-mu.strat.fit.c[1,j]
}
mu.strat.fit.c

+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  3.924982  4.239986  4.438529  4.685589  4.960768
+ [2,]  1.328789  1.929324  2.533836  2.970253  3.946459
+ [3,] -2.596193 -2.310662 -1.904693 -1.715336 -1.014310

apply(mu.strat.fit.c,1,mean)

+ [1]  4.449971  2.541732 -1.908239
```

The results are broadly similar to when the true PS values are used. However, if the PS model is misspecified, this does not hold; for example if we omit X_2 from the PS model, incorrect results are obtained.

```
ps.fit.i<-fitted(glm(Z~X1,family=binomial))
ps.quint.fit.i<-quantile(ps.fit.i,probs=seq(0,1,by=1/nstrata))
ps.quint.fit.i

+        0%       20%       40%       60%       80%      100%
+ 0.5576018 0.6141654 0.6313289 0.6432823 0.6594991 0.7163560

ps.strat.fit.i<-as.numeric(cut(ps.fit.i,ps.quint.fit.i,include.lowest=TRUE))
table(Z,ps.strat.fit.i)

+    ps.strat.fit.i
+ Z     1   2   3   4   5
+   0  70  83  85  75  50
+   1 130 117 115 125 150

mu.strat.fit.i<-matrix(0,ncol=nstrata,nrow=3)
for(j in 1:nstrata){
    mu.strat.fit.i[1,j]<-mean(Y[Z==0 & ps.strat.fit.i == j])
    mu.strat.fit.i[2,j]<-mean(Y[Z==1 & ps.strat.fit.i == j])
    mu.strat.fit.i[3,j]<-mu.strat.fit.i[2,j]-mu.strat.fit.i[1,j]
}
mu.strat.fit.i

+           [,1]      [,2]      [,3]     [,4]       [,5]
+ [1,]  4.337441  4.285596  4.278122  4.27935  4.2557960
+ [2,]  1.986344  2.266358  2.478530  3.02608  3.8558912
+ [3,] -2.351097 -2.019238 -1.799591 -1.25327 -0.3999048

apply(mu.strat.fit.i,1,mean)

+ [1]  4.287261  2.722641 -1.564620
```

With a mis-specified PS model, we no longer have a balancing score for the known confounding variables, that is, Z and X are no longer independent within strata of the propensity score. Lack of balance can be diagnosed numerically (by comparing means of the confounders for $Z = 0$ and $Z = 1$ within strata of the propensity score) or graphically (by comparing boxplots within strata). For the correct model, we have

```
t1<-aggregate(X1,by=list(Z,ps.strat.fit.c),FUN=mean)
t2<-aggregate(X2,by=list(Z,ps.strat.fit.c),FUN=mean)
t3<-aggregate(X3,by=list(Z,ps.strat.fit.c),FUN=mean)
dnames<-list(c("Z=0", "Z=1"),c("Q1", "Q2", "Q3","Q4","Q5"))
matrix(t1$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames )   #Summary for X1: mean

+            Q1        Q2       Q3        Q4       Q5
+ Z=0 0.9978433 0.9316459 0.979086 0.9546425 1.101412
+ Z=1 0.9675377 0.9354829 1.008106 0.9665335 1.095633

matrix(t2$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames )   #Summary for X2: mean

+            Q1        Q2        Q3        Q4        Q5
+ Z=0 -2.282388 -2.231281 -2.049605 -1.931326 -1.578578
+ Z=1 -2.335530 -2.220630 -1.985303 -1.929186 -1.583760

matrix(t3$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames )   #Summary for X3: mean

+             Q1         Q2         Q3        Q4        Q5
+ Z=0 -0.8727500 -1.1261686 -0.9283224 -1.002825 -1.422133
+ Z=1 -0.7485271 -0.7775068 -0.9098424 -1.093546 -1.086302
```

and the variables fitted within the PS are balanced. For the incorrect model we have

```
t1<-aggregate(X1,by=list(Z,ps.strat.fit.i),FUN=mean)
t2<-aggregate(X2,by=list(Z,ps.strat.fit.i),FUN=mean)
t3<-aggregate(X3,by=list(Z,ps.strat.fit.i),FUN=mean)
dnames<-list(c("Z=0", "Z=1"),c("Q1", "Q2", "Q3","Q4","Q5"))
matrix(t1$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames )   #Summary for X1: mean

+            Q1        Q2        Q3       Q4       Q5
+ Z=0 0.6981254 0.8698683 0.9981973 1.114039 1.314847
+ Z=1 0.6476249 0.8667821 0.9940135 1.120220 1.335241

matrix(t2$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames )   #Summary for X2: mean

+            Q1        Q2        Q3        Q4        Q5
+ Z=0 -2.627302 -2.318033 -2.066489 -1.872057 -1.570087
+ Z=1 -2.594987 -2.193335 -1.988481 -1.717672 -1.345579

matrix(t3$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames )   #Summary for X3: mean

+             Q1         Q2         Q3         Q4         Q5
+ Z=0 -0.8686154 -0.9637137 -0.9579892 -1.2352245 -1.0285372
+ Z=1 -0.9155792 -1.0579546 -0.8949457 -0.9421799 -0.9844728
```

and now only X_1 appears balanced. Therefore, the PS mechanism is producing balance amongst those variables that appear in the fitted PS model. However, those variables that remain unbalanced may still have some confounding influence. This analysis informs us that the functional form of the propensity score is correct for the included variables.

4. Matching

Propensity score matching uses the propensity score to find matched (and therefore comparable) individuals in the $Z = 0$ and $Z = 1$ subsets. These matched individuals can then be used to estimate the difference $\mu(1) - \mu(0)$. A statistical analysis using the `Matching` package is straightforward. In the first analysis, we match 1–1 for observations with $Z = 0$ and $Z = 1$. There are 637 cases with $Z = 1$ in the simulated data, but the package re-uses the data to produce a sample of size n of matched pairs. In the function `Match`, the quantity `Tr` denotes the 'treatment' Z, and X is the variable (or variables) used for matching.

```
library(Matching)
fit.match1  <- Match(Y=Y, Tr=Z, X=ps.true, M=1, estimand='ATE')
summary(fit.match1)

+
+ Estimate...  -1.9235
+ AI SE......  0.044062
+ T-stat.....  -43.653
+ p.val......  < 0.000000000000000222
+
+ Original number of observations..............  1000
+ Original number of treated obs...............  637
+ Matched number of observations...............  1000
+ Matched number of observations  (unweighted).  1288
```

In this case the estimate is -1.9234667, and the estimated standard error is 0.0440625.

It is possible also to carry out 1–M matching for $M = 5$ and $M = 10$, say. In this analysis, every case with $Z = 1$ is matched to M individuals with $Z = 0$.

```
fit.match2  <- Match(Y=Y, Tr=Z, X=ps.true, M=5, estimand='ATE')
summary(fit.match2)

+
+ Estimate...  -1.9258
+ AI SE......  0.042978
+ T-stat.....  -44.809
+ p.val......  < 0.000000000000000222
+
+ Original number of observations..............  1000
+ Original number of treated obs...............  637
+ Matched number of observations...............  1000
+ Matched number of observations  (unweighted).  5091

fit.match3  <- Match(Y=Y, Tr=Z, X=ps.true, M=10, estimand='ATE')
summary(fit.match3)

+
+ Estimate...  -1.9216
+ AI SE......  0.042942
+ T-stat.....  -44.749
+ p.val......  < 0.000000000000000222
+
+ Original number of observations..............  1000
+ Original number of treated obs...............  637
+ Matched number of observations...............  1000
+ Matched number of observations  (unweighted).  10047
```

There are many other options in the `Matching` package.

5. Propensity score regression

In propensity score regression, a regression model is built using the propensity score as a predictor. For example, we might fit

$$\mu(z,x) = \mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X=x,Z=z] = \beta_0 + \psi_0 z + \phi e(x)$$

with the idea that conditioning on the propensity score is achieved by entering the term into the regression model.

```
fit.psr.1<-lm(Y~Z+ps.fit.c)
summary(fit.psr.1)

+
+ Call:
+ lm(formula = Y ~ Z + ps.fit.c)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -2.59541 -0.26342  0.07312  0.31082  2.69878
+
+ Coefficients:
+             Estimate Std. Error t value            Pr(>|t|)
+ (Intercept)  1.58112    0.07007   22.57 <0.0000000000000002 ***
+ Z           -2.05087    0.03772  -54.37 <0.0000000000000002 ***
+ ps.fit.c     4.78344    0.11320   42.26 <0.0000000000000002 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 0.5405 on 997 degrees of freedom
+ Multiple R-squared:  0.7835,Adjusted R-squared:  0.7831
+ F-statistic:  1804 on 2 and 997 DF,  p-value: < 0.00000000000000022
```

The estimator that follows from this procedure is a regression estimator as in equation (1). In a this model, we can read off the estimator of $\mu(1) - \mu(0)$ as $\widehat{\psi}_0$, which here is equal to -2.0508704. As a regression model, it suffers the same problems as the regression model from Section 2., which relate to mis-specification: if $\mu(x,z)$ is mis-specified, incorrect inference will follow.

The regression model may be extended to include other terms. Suppose we specify

$$\mu(x,z) = \beta_0 + z(\psi_0 + \psi_1 x_1 + \psi_2 x_2 + \psi_3 x_1 x_2) + \phi_0 e(x)$$

```
fit.psr.2<-lm(Y~Z+Z:X1+Z:X2+Z:X1:X2+ps.fit.c)
round(coef(summary(fit.psr.2)),6)

+              Estimate Std. Error   t value Pr(>|t|)
+ (Intercept)  3.085174   0.027244 113.24310        0
+ Z            3.964332   0.135035  29.35781        0
+ ps.fit.c     2.126006   0.046459  45.76071        0
+ Z:X1        -0.874477   0.083430 -10.48155        0
+ Z:X2         1.894645   0.042690  44.38134        0
+ Z:X1:X2      0.627663   0.039762  15.78543        0

mu0.psr<-mean(predict(fit.psr.2,newdata=data.frame(X1,X2,Z=0,ps.fit.c)))
mu1.psr<-mean(predict(fit.psr.2,newdata=data.frame(X1,X2,Z=1,ps.fit.c)))
c(mu0.psr,mu1.psr,mu1.psr-mu0.psr)

+ [1]  4.439440  2.532145 -1.907295
```

then an approximately correct conclusion is obtained for estimating $\mu(0)$ and $\mu(1)$ – this is despite the fact that the ψ parameters are not estimated correctly, that is, the estimates do not match the ψ parameters in the data generating model. Compare this with the incorrectly specified outcome regression

$$\mu(x,z) = \beta_0 + z(\psi_0 + \psi_1 x_1 + \psi_2 x_2 + \psi_3 x_1 x_2) = m_0(\beta_0) + z m_1(x;\psi)$$

say. Then

```
fit.or2<-lm(Y~Z+Z:X1+Z:X2+Z:X1:X2)
round(coef(summary(fit.or2)),6)

+              Estimate Std. Error    t value Pr(>|t|)
+ (Intercept)  4.288448   0.012557 341.526494        0
+ Z            6.761074   0.212130  31.872303        0
+ Z:X1        -1.040222   0.146840  -7.084072        0
+ Z:X2         2.221214   0.074149  29.956267        0
+ Z:X1:X2      1.583540   0.059604  26.567677        0

mu0.or2<-mean(predict(fit.or2,newdata=data.frame(X1,X2,Z=0)))
mu1.or2<-mean(predict(fit.or2,newdata=data.frame(X1,X2,Z=1)))
c(mu0.or2,mu1.or2,mu1.or2-mu0.or2)

+ [1]  4.288448  2.539963 -1.748485
```

which does not yield the correct result. We have in this model that the *interaction* model $m_1(x;\psi) \equiv \mu_1(x;\psi)$ is correctly specified in that it coincides with the true interaction model. However we have that $m_0(\beta_0)$ does **not** match the true *outcome nuisance model* $\mu_0(x;\beta) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$, and as the parameter β_0 in $m_0(.)$ is estimated, we do not obtain correct inference.

Suppose finally we fit the model that includes interactions between $e(x)$ and x_1 and x_2.

$$\mu(x,z) = \beta_0 + z(\psi_0 + \psi_1 x_1 + \psi_2 x_2 + \psi_3 x_1 x_2) + e(x)(\phi_0 + \phi_1 x_1 + \phi_2 x_2 + \phi_3 x_1 x_2)$$

```
fit.psr.3<-lm(Y~Z+Z:X1+Z:X2+Z:X1:X2+ps.fit.c+ps.fit.c:X1+ps.fit.c:X2 + ps.fit.c:X1:X2)
round(coef(summary(fit.psr.3)),6)

+                 Estimate Std. Error    t value Pr(>|t|)
+ (Intercept)     3.477510   0.028840 120.580161 0.000000
+ Z               0.989859   0.202076   4.898455 0.000001
+ ps.fit.c        6.918620   0.237115  29.178343 0.000000
+ Z:X1            1.022784   0.157344   6.500303 0.000000
+ Z:X2            0.982109   0.076041  12.915514 0.000000
+ X1:ps.fit.c    -2.761434   0.175974 -15.692298 0.000000
+ X2:ps.fit.c     1.441217   0.087193  16.528950 0.000000
+ Z:X1:X2         1.027346   0.068479  15.002324 0.000000
+ X1:X2:ps.fit.c -0.107657   0.077801  -1.383748 0.166747

mu0.psr3<-mean(predict(fit.psr.3,newdata=data.frame(X1,X2,Z=0,ps.fit.c)))
mu1.psr3<-mean(predict(fit.psr.3,newdata=data.frame(X1,X2,Z=1,ps.fit.c)))
c(mu0.psr3,mu1.psr3,mu1.psr3-mu0.psr3)

+ [1]  4.456889  2.538435 -1.918454
```

The resulting estimation of the causal quantities does not change appreciably. However,

```
round(cbind(coef(summary(fit.psr.3))[c(2,4,5,8),],psi),6)

+         Estimate Std. Error   t value Pr(>|t|) psi
+ Z       0.989859   0.202076  4.898455 0.000001   1
+ Z:X1    1.022784   0.157344  6.500303 0.000000   1
+ Z:X2    0.982109   0.076041 12.915514 0.000000   1
+ Z:X1:X2 1.027346   0.068479 15.002324 0.000000   1
```

we observe that now the ψ values are now also correctly estimated.

6. Inverse Weighting

We now study the inverse weighting (or inverse probability weighting, IPW) approach that is based on an importance sampling (or change of measure) procedure. We have that

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

which becomes in the binary treatment case

$$\mu(0) = \frac{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{(1-Z)Y}{1-e(X)}\right]}{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{(1-Z)}{1-e(X)}\right]} \qquad \mu(1) = \frac{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{ZY}{e(X)}\right]}{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{Z}{e(X)}\right]}$$

6.1 IPW

We deduce that suitable estimators following from the importance sampling results are

$$\widehat{\mu}_{IPW}(0) = \frac{\frac{1}{n}\sum_{i=1}^{n}\frac{(1-Z_i)Y_i}{1-e(X_i)}}{\frac{1}{n}\sum_{i=1}^{n}\frac{(1-Z_i)}{1-e(X_i)}} \qquad \widehat{\mu}_{IPW}(1) = \frac{\frac{1}{n}\sum_{i=1}^{n}\frac{Z_iY_i}{e(X_i)}}{\frac{1}{n}\sum_{i=1}^{n}\frac{Z_i}{e(X_i)}}$$

which we may write

$$\widehat{\mu}_{IPW}(0) = \frac{\sum_{i=1}^{n} w_{0i}Y_i}{\sum_{i=1}^{n} w_{0i}} = \sum_{i=1}^{n} W_{0i}Y_i \qquad \widehat{\mu}_{IPW}(1) = \frac{\sum_{i=1}^{n} w_{1i}Y_i}{\sum_{i=1}^{n} w_{1i}} = \sum_{i=1}^{n} W_{1i}Y_i$$

with

$$w_{0i} = \frac{(1-Z_i)}{1-e(X_i)} \qquad w_{1i} = \frac{Z_i}{e(X_i)} \qquad W_{zi} = \frac{w_{zi}}{\sum_{j=1}^{n} w_{zj}} \quad z = 0, 1.$$

Note that by construction,

$$\mathbb{E}^{\mathcal{O}}_{X,Y,Z}[\widehat{\mu}_{IPW}(z)] = \mu(z) \qquad z = 0, 1$$

so the estimators are unbiased, and also that

$$\begin{aligned}\mathbb{E}^{\mathcal{O}}_{X,Y,Z}[W_{zi}] = \mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{1}_{\{z\}}(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)}\right] &= \mathbb{E}^{\mathcal{O}}_{X}\left[\mathbb{E}^{\mathcal{O}}_{Z|X}\left[\mathbb{1}_{\{z\}}(Z)\frac{1}{f^{\mathcal{O}}_{Z|X}(Z|X)}\middle|X\right]\right] \\ &= \mathbb{E}^{\mathcal{O}}_{X}\left[\mathbb{E}^{\mathcal{O}}_{Z|X}\left[f^{\mathcal{O}}_{Z|X}(Z|X)\frac{1}{f^{\mathcal{O}}_{Z|X}(Z|X)}\middle|X\right]\right] = 1.\end{aligned}$$

This latter result suggests the alternative IPW estimators where we replace the denominator by its expectation, n, that is,

$$\widetilde{\mu}_{IPW}(0) = \frac{1}{n}\sum_{i=1}^{n}\frac{(1-Z_i)Y_i}{1-e(X_i)} \qquad \widetilde{\mu}_{IPW}(1) = \frac{1}{n}\sum_{i=1}^{n}\frac{Z_iY_i}{e(X_i)}.$$

Using fitted PS, we can compute as follows: for $\widehat{\mu}(z)$, we have

```
w0<-(1-Z)/(1-ps.fit.c)
W0<-w0/sum(w0)
mu.hat0<-sum(W0*Y)
w1<-Z/ps.fit.c
W1<-w1/sum(w1)
mu.hat1<-sum(W1*Y)
c(mu.hat0,mu.hat1,mu.hat1-mu.hat0)

+ [1]  4.488312  2.522956 -1.965355
```

and for $\widetilde{\mu}(z)$, we have

```
mu.tilde0<-mean(w0*Y)
mu.tilde1<-mean(w1*Y)
c(mu.tilde0,mu.tilde1,mu.tilde1-mu.tilde0)

+ [1]  4.578971  2.535878 -2.043093
```

Note: We may recover the $\widehat{\mu}(z)$ quantities using a weighted least squares (WLS) regression of Y on Z

$$\mathbb{E}^{o}_{Y|Z}[Y|Z=z]=\beta_0+\psi_0 z$$

with weights

$$w_{0i}+w_{1i}=\frac{(1-Z_i)}{1-e(X_i)}+\frac{Z_i}{e(X_i)}=R_i$$

say.

```
w<-w0+w1
fit.wls<-lm(Y~Z,weights=w)
round(coef(summary(fit.wls)),6)

+             Estimate Std. Error  t value Pr(>|t|)
+ (Intercept)  4.488312   0.037154 120.8032        0
+ Z           -1.965355   0.052740 -37.2648        0
```

The estimated coefficients are precisely $\widehat{\mu}(0)$ and $\widehat{\mu}(1)-\widehat{\mu}(1)$. This result gives us further insight into how the inverse probability weighting succeeds in correcting the confounding; it creates a pseudo-sample from the distribution from the experimental measure ε: that is

- in the observed sample, collected under o, each data point carries a weight $1/n$.
- for the pseudo sample, we wish to have that each data point carries a weight $1/n$ under ε, but in fact datum i carries a weight that is dependent on X_i, that is

$$f^{o}_{Z|X}(z|x_i)=\{e(x_i)\}^z\{1-e(x_i)\}^{1-z}\qquad z=0,1.$$

6.2 Augmented IPW

In augmented IPW (AIPW) estimation, we utilize the fact that if we denote

$$\mu(x,z)=\mathbb{E}^{\varepsilon}_{Y|X,Z}[Y|X=z,Z=z]=\mathbb{E}^{o}_{Y|X,Z}[Y|X=z,Z=z]$$

we may write

$$\begin{aligned}\mathbb{E}^{o}_{X,Y,Z}\left[\frac{\mathbb{1}_{\{z\}}(Z)Y}{f^{o}_{Z|X}(Z|X)}\right]&=\mathbb{E}^{o}_{X,Y,Z}\left[\frac{\mathbb{1}_{\{z\}}(Z)(Y-\mu(X,Z)+\mu(X,Z))}{f^{o}_{Z|X}(Z|X)}\right]\\&=\mathbb{E}^{o}_{X,Y,Z}\left[\frac{\mathbb{1}_{\{z\}}(Z)(Y-\mu(X,Z))}{f^{o}_{Z|X}(Z|X)}\right]+\mathbb{E}^{o}_{X,Y,Z}\left[\frac{\mathbb{1}_{\{z\}}(Z)\mu(X,Z)}{f^{o}_{Z|X}(Z|X)}\right]\\&=\mathbb{E}^{o}_{X,Y,Z}\left[\frac{\mathbb{1}_{\{z\}}(Z)(Y-\mu(X,Z))}{f^{o}_{Z|X}(Z|X)}\right]+\mathbb{E}^{o}_{X}\left[\mu(X,z)\right]\end{aligned}$$

with the last simplification following by iterated expectation. This result suggests the alternative estimator

$$\widehat{\mu}_{AIPW}(z) = \sum_{i=1}^{n} W_{zi}(Y_i - \mu(X_i, z)) + \frac{1}{n}\sum_{i=1}^{n}\mu(X_i, z) \qquad z = 0, 1.$$

By the above construction, $\widehat{\mu}_{AIPW}(z)$ is also an unbiased estimator of $\mu(z)$. We may also utilize the estimator

$$\widetilde{\mu}_{AIPW}(z) = \frac{1}{n}\sum_{i=1}^{n} w_{zi}(Y_i - \mu(X_i, z)) + \frac{1}{n}\sum_{i=1}^{n}\mu(X_i, z) \qquad z = 0, 1$$

which is also unbiased for $\mu(z)$.

These augmented (AIPW) estimators are preferred to the original (IPW) estimators as they can have lower variance. To see this, note that we have that

$$\widetilde{\mu}_{AIPW}(z) = \frac{1}{n}\sum_{i=1}^{n}(w_{zi}Y_i + (1 - w_{zi})\mu(X_i, z)).$$

From this form, we see that $\widetilde{\mu}_{AIPW}(z)$ is an unbiased estimator, as by iterated expectation

$$\mathbb{E}[w_{zi}Y_i + (1 - w_{zi})\mu(X_i, z)] = \mathbb{E}[w_{zi}\mu(X_i, z) + (1 - w_{zi})\mu(X_i, z)] = \mathbb{E}[\mu(X_i, z)] = \mu(z).$$

The variance of the estimator is the variance of the individual terms in the summation divided by n as the terms are independent. In the following calculation, all expectations and variances are taken with respect to the joint observational distribution o. We have from first principles that

$$\text{Var}[W_z Y + (1 - W_z)\mu(X, z)] = \text{Var}[W_z Y] + \text{Var}[(1 - W_z)\mu(X, z)] + 2\text{Cov}[W_z Y, (1 - W_z)\mu(X, z)].$$

Now

$$\text{Var}[(1 - W_z)\mu(X, z)] = \mathbb{E}[(1 - W_z)^2\{\mu(X, z)\}^2] - \{\mathbb{E}[(1 - W)\mu(X, z)]\}^2 = \mathbb{E}[(1 - W)^2\{\mu(X, z)\}^2]$$

as $\mathbb{E}[W_z] = 1$ by the earlier result, so $\mathbb{E}[(1 - W)\mu(X, z)] = 0$ by iterated expectation. Secondly

$$\text{Cov}[W_z Y, (1 - W_z)\mu(X, z)] = \mathbb{E}[W_z(1 - W_z)Y\mu(X, z)] - \mathbb{E}[W_z Y]\mathbb{E}[(1 - W_z)\mu(X, z)] = \mathbb{E}[W_z(1 - W_z)\{\mu(X, z)\}^2]$$

again by iterated expectation, again as $\mathbb{E}[(1 - W)\mu(X, z)] = 0$. Therefore

$$\begin{aligned}
\text{Var}[W_z Y + (1 - W_z)\mu(X, z)] &= \text{Var}[W_z Y] + \mathbb{E}[(1 - W_z)^2\{\mu(X, z)\}^2] + 2\mathbb{E}[W_z(1 - W_z)\{\mu(X, z)\}^2] \\
&= \text{Var}[W_z Y] + \mathbb{E}[(1 - W_z^2)\{\mu(X, z)\}^2] \\
&< \text{Var}[W_z Y]
\end{aligned}$$

as, in the second term, on computing by iterated expectation, we have that

$$\mathbb{E}^{o}_{Z|X}[(1 - W_z^2)] = \mathbb{E}^{o}_{Z|X}\left[1 - \left(\frac{\mathbb{1}_{\{z\}}(Z)}{f^{o}_{Z|X}(Z|X)}\right)^2\right] = \mathbb{E}^{o}_{Z|X}\left[1 - \frac{\mathbb{1}_{\{z\}}(Z)}{\left(f^{o}_{Z|X}(Z|X)\right)^2}\right] = 1 - \frac{1}{f^{o}_{Z|X}(z|X)}$$

that is,

$$\mathbb{E}^{o}_{Z|X}[(1 - W_z^2)|X] = 1 - \frac{1}{\{e(X)\}^z\{1 - e(X)\}^{1-z}} < 0$$

almost surely (that is, this quantity is a random variable defined in terms of X that is negative with probability 1), and so

$$\mathbb{E}[(1 - W_z^2)\{\mu(X, z)\}^2] < 0.$$

This result follows also for the estimator $\widetilde{\mu}(z)$.

For the calculation with known, correctly specified $\mu(x,z)$ we have

```
mu.x0<- Xb %*% be
mu.x1<- Xb %*% be + Xp %*% psi
mu.hat.a0<-sum(W0*(Y-mu.x0))+mean(mu.x0)
mu.hat.a1<-sum(W1*(Y-mu.x1))+mean(mu.x1)
c(mu.hat.a0,mu.hat.a1,mu.hat.a1-mu.hat.a0)

+ [1]  4.455653  2.539759 -1.915894

mu.tilde.a0<-mean(w0*(Y-mu.x0))+mean(mu.x0)
mu.tilde.a1<-mean(w1*(Y-mu.x1))+mean(mu.x1)
c(mu.tilde.a0,mu.tilde.a1,mu.tilde.a1-mu.tilde.a0)

+ [1]  4.455630  2.539749 -1.915881
```

In practice we need to estimate the parameters in the specification $\mu(x,z;\beta,\psi)$: under correct specification

```
fit0<-lm(Y~X1+X2+X1:X2+Z+Z:X1+Z:X2+Z:X1:X2)
mu.x0<-predict(fit0,newdata=data.frame(X1,X2,Z=0))
mu.x1<-predict(fit0,newdata=data.frame(X1,X2,Z=1))
mu.hat.a0<-sum(W0*(Y-mu.x0))+mean(mu.x0)
mu.hat.a1<-sum(W1*(Y-mu.x1))+mean(mu.x1)
c(mu.hat.a0,mu.hat.a1,mu.hat.a1-mu.hat.a0)

+ [1]  4.456039  2.539762 -1.916277

mu.tilde.a0<-mean(w0*(Y-mu.x0))+mean(mu.x0)
mu.tilde.a1<-mean(w1*(Y-mu.x1))+mean(mu.x1)
c(mu.tilde.a0,mu.tilde.a1,mu.tilde.a1-mu.tilde.a0)

+ [1]  4.456073  2.539761 -1.916313
```

This realization highlights the fact that under an assumption of correct specification, we could merely perform outcome regression, which is known to optimal, so that in fact the IPW version is redundant. Specifically, under correct specification of $\mu(X,z;\beta,\psi)$, we have by iterated expectation that

$$\mathbb{E}[W_z(Y-\mu(X,z;\beta,\psi))] = \mathbb{E}[W_z(Y-\mu(x,z;\widehat{\beta},\widehat{\psi}))] = 0.$$

However, suppose that we mis-specify the conditional mean model as $m(x,z;\beta,\psi)$, say. Then provided the propensity score model is correctly specified, we still have that

$$\text{Var}[W_zY + (1-W_z)m(X,z;\widehat{\beta},\widehat{\psi})] < \text{Var}[W_zY]$$

but also that

$$\begin{aligned}\mathbb{E}[W_zY + (1-W_z)\mu(X,z;\widehat{\beta},\widehat{\psi})] &= \mathbb{E}[W_zY + (1-W_z)(\mu(X,z;\widehat{\beta},\widehat{\psi}) - \mu(X,z)) + (1-W_z)\mu(X,z)] \\ &= \mathbb{E}[W_z\mu(X,z) + (1-W_z)(\mu(X,z;\widehat{\beta},\widehat{\psi}) - \mu(X,z)) + (1-W_z)\mu(X,z)] \\ &= \mathbb{E}[\mu(X,z)] = \mu(z)\end{aligned}$$

by iterated expectation, as $\mathbb{E}^{\mathcal{O}}_{Z|X}[(1-W_z)|X] = 0$. Therefore, we obtain an unbiased estimator even if the conditional mean model is mis-specified. For example, if we specify

$$m(x,z) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \psi_0 z$$

then the result holds.

```
fit0<-lm(Y~X1+X2+Z)
mu.x0<-predict(fit0,newdata=data.frame(X1,X2,Z=0))
mu.x1<-predict(fit0,newdata=data.frame(X1,X2,Z=1))
mu.hat.a0<-sum(W0*(Y-mu.x0))+mean(mu.x0)
mu.hat.a1<-sum(W1*(Y-mu.x1))+mean(mu.x1)
c(mu.hat.a0,mu.hat.a1,mu.hat.a1-mu.hat.a0)
```

```
+ [1]  4.408073  2.535561 -1.872511

mu.tilde.a0<-mean(w0*(Y-mu.x0))+mean(mu.x0)
mu.tilde.a1<-mean(w1*(Y-mu.x1))+mean(mu.x1)
c(mu.tilde.a0,mu.tilde.a1,mu.tilde.a1-mu.tilde.a0)

+ [1]  4.404497  2.535222 -1.869275
```

Conversely, if the propensity score model is mis-specified, but the conditional mean model is correctly specified, we again obtain an unbiased estimator: suppose that the propensity score model proposed is $f^{\dagger}_{Z|X}(Z|X)$, and denote

$$w^{\dagger}_{zi} = \frac{\mathbb{1}_{\{z\}}(Z_i)}{f^{\dagger}_{Z|X}(Z_i|X_i)}.$$

Then

$$\mathbb{E}[w^{\dagger}_{zi}] = \mathbb{E}^{\mathcal{O}}_{Z|X}\left[\frac{\mathbb{1}_{\{z\}}(Z_i)}{f^{\dagger}_{Z|X}(Z_i|X_i)}\middle| X_i\right] = \frac{f^{\mathcal{O}}_{Z|X}(z|X_i)}{f^{\dagger}_{Z|X}(z|X_i)} = r^{\dagger}(X_i, z)$$

say, and we still have an unbiased estimator as

$$\mathbb{E}[(w^{\dagger}_{zi}Y_i + (1 - w^{\dagger}_{zi})\mu(X_i, z)] = \mathbb{E}[(w^{\dagger}_{zi}\mu(X_i, z) + (1 - w^{\dagger}_{zi})\mu(X_i, z)] = \mu(z).$$

Recall that the IPW estimator is not unbiased if the PS model is mis-specified, as we would have

$$\mathbb{E}^{\mathcal{O}}_{X,Y,Z}[\widetilde{\mu}(z)] = \mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{1}_{\{z\}}(Z_i)Y_i}{f^{\dagger}_{Z|X}(Z_i|X_i)}\right] = \mathbb{E}^{\mathcal{O}}_{X}\left[r^{\dagger}(X_i, z)\mu(X_i, z)\right] \neq \mu(z).$$

Thus the AIPW estimator is unbiased even if precisely one of the propensity or conditional mean models is mis-specified. This property is termed *double robustness*.

Note: The double robustness to mis-specification of the AIPW estimator might imply that it is superior to the singly robust outcome regression or IPW estimators. This is not necessarily true in practice, as a badly mis-specified model for the PS or the conditional mean model may yield an AIPW estimator that is not appreciably better than either singly robust estimator.

If both PS and outcome regression model are mis-specified, then a biased estimator results

```
fit0<-lm(Y~X1+Z)
w0.i<-(1-Z)/(1-ps.fit.i)
W0.i<-w0.i/sum(w0.i)
w1.i<-Z/ps.fit.i
W1.i<-w1.i/sum(w1.i)
mu.hat.i0<-sum(W0.i*Y)
mu.hat.i1<-sum(W1.i*Y)
c(mu.hat.i0,mu.hat.i1,mu.hat.i1-mu.hat.i0)

+ [1]  4.286822  2.741359 -1.545463

mu.x0<-predict(fit0,newdata=data.frame(X1,X2,Z=0))
mu.x1<-predict(fit0,newdata=data.frame(X1,X2,Z=1))
mu.hat.a0<-sum(W0.i*(Y-mu.x0))+mean(mu.x0)
mu.hat.a1<-sum(W1.i*(Y-mu.x1))+mean(mu.x1)
c(mu.hat.a0,mu.hat.a1,mu.hat.a1-mu.hat.a0)

+ [1]  4.295775  2.744420 -1.551355

mu.tilde.a0<-mean(w0.i*(Y-mu.x0))+mean(mu.x0)
mu.tilde.a1<-mean(w1.i*(Y-mu.x1))+mean(mu.x1)
c(mu.tilde.a0,mu.tilde.a1,mu.tilde.a1-mu.tilde.a0)

+ [1]  4.295803  2.744417 -1.551386
```

7. G-estimation

7.1 Construction

G–estimation can be considered a modified form of outcome regression estimation that introduces a robustness to mis-specification via PS adjustment. Consider first the data generating model

$$\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X=x,Z=z] = z\psi_0.$$

Then the OLS estimating equation for the ψ_0 is

$$\sum_{i=1}^{n} z_i(y_i - z_i\psi_0) = 0.$$

Then we have that the resulting estimate for ψ_0 is

$$\widehat{\psi}_0 = \frac{\sum\limits_{i=1}^{n} z_i y_i}{\sum\limits_{i=1}^{n} z_i^2}.$$

Suppose now that the estimating equation is modified to

$$\sum_{i=1}^{n}(z_i - e(x_i))(y_i - z_i\psi_0) = 0.$$

Then we have that the new estimate for ψ_0 is

$$\widehat{\psi}_0 = \frac{\sum\limits_{i=1}^{n}(z_i - e(x_i))y_i}{\sum\limits_{i=1}^{n} z_i(z_i - e(x_i))}$$

Consider now the more general linear model

$$\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X=x,Z=z] = \mathbf{x}_1\beta + z\mathbf{x}_2\psi = \mu_0(x;\beta) + z\mu_1(x;\psi)$$

and the OLS estimating equation

$$\sum_{i=1}^{n}\begin{pmatrix}\mathbf{x}_{1i}^\top \\ z\mathbf{x}_{2i}^\top\end{pmatrix}(y_i - \mathbf{x}_{1i}\beta - z\mathbf{x}_{2i}\psi) = \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}.$$

In modified form, this becomes

$$\sum_{i=1}^{n}\begin{pmatrix}\mathbf{x}_{1i}^\top \\ (z - e(x_i))\mathbf{x}_{2i}^\top\end{pmatrix}(y_i - \mathbf{x}_{1i}\beta - z\mathbf{x}_{2i}\psi) = \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}.$$

These equations are referred to as the *G–estimating equations*. It can be shown that the estimates for ψ obtained by solving these equations are realizations of unbiased estimators of the true values even if the model

$$\mu_0(x;\beta) = \mathbf{x}_1\beta$$

is mis-specified, **provided that the model**

$$\mu_1(x;\psi) = \mathbf{x}_2\psi$$

is correctly specified, **and** provided that the PS model is correctly specified. Thus this procedure is a *doubly robust* procedure if $\mu_1(x;\psi)$ is correctly specified.

To solve the estimating equations, we consider estimating the model

$$\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X=x,Z=z] = \mu_0(x;\beta) + z\mu_1(x;\psi) + e(x)\mu_1(x;\phi)$$

via OLS. For example, suppose we have the linear forms

$$\mu_0(x;\beta) = \beta_0$$
$$\mu_1(x;\psi) = \psi_0 + \psi_1 x_1 + \psi_2 x_2 + \psi_3 x_1 x_2$$
$$\mu_1(x;\phi) = \phi_0 + \phi_1 x_1 + \phi_2 x_2 + \phi_3 x_1 x_2$$

Then the OLS estimating equations take the form

$$\sum_{i=1}^{n} \begin{pmatrix} \mathbf{x}_{1i}^\top \\ z_i \mathbf{x}_{2i}^\top \\ e(x_i)\mathbf{x}_{2i}^\top \end{pmatrix} (y_i - \beta_0 - z\mathbf{x}_{2i}\psi - e(x)\mathbf{x}_{2i}\phi) = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

and subtracting the third block from the second block yields

$$\sum_{i=1}^{n} \begin{pmatrix} \mathbf{x}_{1i}^\top \\ (z_i - e(x_i))\mathbf{x}_{2i}^\top \\ e(x_i)\mathbf{x}_{2i}^\top \end{pmatrix} (y_i - \beta_0 - z\mathbf{x}_{2i}\psi - e(x)\mathbf{x}_{2i}\phi) = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}.$$

which recovers the G–estimating equations. Hence, **G–estimation is a form of propensity score regression**.

```
fit.gee<-lm(Y~Z+Z:X1+Z:X2+Z:X1:X2+ps.fit.c+ps.fit.c:X1+ps.fit.c:X2+ps.fit.c:X1:X2)
round(coef(summary(fit.gee)),6)

+                 Estimate Std. Error    t value Pr(>|t|)
+ (Intercept)     3.477510   0.028840 120.580161 0.000000
+ Z               0.989859   0.202076   4.898455 0.000001
+ ps.fit.c        6.918620   0.237115  29.178343 0.000000
+ Z:X1            1.022784   0.157344   6.500303 0.000000
+ Z:X2            0.982109   0.076041  12.915514 0.000000
+ X1:ps.fit.c    -2.761434   0.175974 -15.692298 0.000000
+ X2:ps.fit.c     1.441217   0.087193  16.528950 0.000000
+ Z:X1:X2         1.027346   0.068479  15.002324 0.000000
+ X1:X2:ps.fit.c -0.107657   0.077801  -1.383748 0.166747

mu0.gee<-mean(predict(fit.gee,newdata=data.frame(X1,X2,Z=0,ps.fit.c)))
mu1.gee<-mean(predict(fit.gee,newdata=data.frame(X1,X2,Z=1,ps.fit.c)))
c(mu0.gee,mu1.gee,mu1.gee-mu0.gee)

+ [1]  4.456889  2.538435 -1.918454
```

7.2 The `drgee` package

The doubly robust version of G–estimation is carried out by the `drgee` package in `R`. This package considers the model specification

$$\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X=x,Z=z] = \mu_0(x;\beta) + z\mu_1(x;\psi)$$

for the outcome regression component, and

$$\mathbb{E}^{\mathcal{O}}_{Z|X}[Z|X=x] = g(x;\alpha)$$

for the PS component, where $g(.)$ is an appropriate link function. In the `drgee` function

- $\mu_0(x;\beta)$ is specified via the `oformula` and `olink` arguments;
- $\mu_1(x;\beta)$ is specified via the `iaformula` and `olink` arguments;
- $g(x;\alpha)$ is specified via the `eformula` and `elink` arguments.

The argument `estimation.method` allows for estimation via three different methods:

- `estimation.method = 'dr'` performs doubly robust G–estimation;

- `estimation.method = 'e'` performs singly robust G–estimation, that is, it uses the model with

$$\mu_0(x) \equiv 0;$$

- `estimation.method = 'o'` performs outcome regression using the model

$$\mu_0(x;\beta) + z\mu_1(x;\psi).$$

```
mydata<-data.frame(X1,X2,Z,Y)
library(drgee)

#Correct specification
dr.est <- drgee(oformula = formula(Y~X1*X2), eformula = formula(Z~X1*X2),
iaformula = formula(~X1*X2), olink = "identity", elink = "logit",
data = mydata, estimation.method = "dr")
round(coef(summary(dr.est)),6)

+         Estimate Std. Error   z value Pr(>|z|)
+ Z       0.960527   0.186412  5.152697        0
+ Z:X1    1.042272   0.142903  7.293565        0
+ Z:X2    0.971475   0.070635 13.753405        0
+ Z:X1:X2 1.031380   0.062547 16.489619        0

#Mis-specification of PS model, correct specification of the OR nuisance model
dr.est <- drgee(oformula = formula(Y~X1*X2), eformula = formula(Z~X1),
iaformula = formula(~X1*X2), olink = "identity", elink = "logit",
data = mydata, estimation.method = "dr")
round(coef(summary(dr.est)),6)

+         Estimate Std. Error   z value Pr(>|z|)
+ Z       0.976057   0.184511  5.289964        0
+ Z:X1    1.033960   0.141403  7.312173        0
+ Z:X2    0.979746   0.068582 14.285742        0
+ Z:X1:X2 1.026245   0.059186 17.339432        0

#Correct specification of PS model, mis-specification of the OR nuisance model
dr.est <- drgee(oformula = formula(Y~X1), eformula = formula(Z~X1*X2),
iaformula = formula(~X1*X2), olink = "identity", elink = "logit",
data = mydata, estimation.method = "dr")
round(coef(summary(dr.est)),6)

+         Estimate Std. Error  z value Pr(>|z|)
+ Z       1.137266   0.876159 1.298013 0.194283
+ Z:X1    1.058655   0.728725 1.452750 0.146293
+ Z:X2    0.960641   0.333634 2.879326 0.003985
+ Z:X1:X2 1.141582   0.337615 3.381314 0.000721
```

To match the estimation in the earlier section, we note that `drgee` utilizes an alternative formulation and an additional estimating equation. Using bold symbols to denote vectors and matrices, in the linear case `drgee` proceeds by solving the estimating equations

$$\begin{pmatrix} \mathbf{X}_\beta^\top(\mathbf{y} - \mathbf{X}_\beta\beta - \mathbf{z}\odot\mathbf{X}_\psi\psi') \\ \mathbf{X}_\psi^\top\mathbf{z}\odot(\mathbf{y} - \mathbf{X}_\beta\beta - \mathbf{z}\odot\mathbf{X}_\psi\psi') \\ \mathbf{X}_\psi^\top(\mathbf{z}-\mathbf{e})\odot(\mathbf{y} - \mathbf{X}_\beta\beta - \mathbf{z}\odot\mathbf{X}_\psi\psi) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

where ψ is the parameter of interest, and where the symbol $\odot$ denotes *elementwise* multiplication, that is for two vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^k$,

$$\mathbf{x}_1 \odot \mathbf{x}_2 = [x_{1j}x_{2j}]_{j=1}^k \in \mathbb{R}^k.$$

`drgee` solves the first two equations to get $\widehat{\beta}$ utilizing the additional nuisance parameter ψ', then solves

$$\mathbf{X}_\psi^\top(\mathbf{z}-\mathbf{e})\odot(\mathbf{y} - \mathbf{X}_\beta\widehat{\beta} - \mathbf{z}\odot\mathbf{X}_\psi\psi) = 0$$

that is, yielding the solution

$$\widehat{\psi} = (\mathbf{X}_\psi^\top \mathbf{D} \mathbf{X}_\psi)^{-1} \mathbf{X}_\psi^\top (\mathbf{z} - \mathbf{e}) \odot (\mathbf{y} - \mathbf{X}_\beta \widehat{\beta})$$

where $\mathbf{D}$ is the $n \times n$ diagonal matrix

$$\mathbf{D} = \operatorname{diag}\{(\mathbf{z} - \mathbf{e}) \odot \mathbf{z}\}.$$

In contrast, the ordinary G-estimation method uses the reduced system

$$\begin{pmatrix} \mathbf{X}_\beta^\top (\mathbf{y} - \mathbf{X}_\beta \beta - \mathbf{z} \odot \mathbf{X}_\psi \psi) \\ \mathbf{X}_\psi^\top (\mathbf{z} - \mathbf{e}) \odot (\mathbf{y} - \mathbf{X}_\beta \beta - \mathbf{z} \odot \mathbf{X}_\psi \psi) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

which does not have the extra nuisance parameters ψ'. The estimator $\widehat{\psi}$ takes precisely the same form except that the plug-in form of $\widehat{\beta}$ is different. Under correct specification of the blip (always assumed in our analysis), the two methods are essentially equivalent.

```
dr.est <- drgee(oformula = formula(Y~1), eformula = formula(Z~X1*X2),
iaformula = formula(~X1*X2), olink = "identity", elink = "logit",
data = mydata, estimation.method = "dr")
pars<-dr.est$coefficients.all
psi.hat<-pars[1:4]                          #psi parameters
alpha.hat<-pars[5:8]                        #ps parameters
psi.prime.hat<-pars[9:12]                   #psi additional parameters
beta.hat<-pars[13]                          #beta.parameters
psi.hat

+         Z      Z:X1      Z:X2   Z:X1:X2
+ 1.1247753 1.0759005 0.9522669 1.1527323
```

```
#Computing the ATE
Xm<-cbind(1,X1,X2,X1*X2)
ps.fit.drgee<-1/(1+exp(-Xm %*% alpha.hat))
mu0.drgee<-mean(beta.hat + (0-ps.fit.drgee)*Xm%*%psi.hat+ps.fit.drgee*Xm%*%psi.prime.hat)
mu1.drgee<-mean(beta.hat + (1-ps.fit.drgee)*Xm%*%psi.hat+ps.fit.drgee*Xm%*%psi.prime.hat)
c(mu0.drgee,mu1.drgee,mu1.drgee-mu0.drgee)

+ [1]  4.446842  2.537881 -1.908961
```

```
#Now to check the calculation using the theory above
psi.prime.hat

+         Z      Z:X1      Z:X2   Z:X1:X2
+  6.761074 -1.040222  2.221214  1.583540

beta.hat

+ (Intercept)
+    4.288448

coef(fit0<-lm(Y~Z+Z:X1+Z:X2+Z:X1:X2))

+ (Intercept)           Z        Z:X1        Z:X2     Z:X1:X2
+    4.288448    6.761074   -1.040222    2.221214    1.583540

alpha.hat

+ (Intercept)          X1          X2       X1:X2
+   5.3725997   1.2112540   0.1009883   3.0090512

coef(fit1<-glm(Z~X1*X2,family=binomial))
```

```
+ (Intercept)          X1          X2       X1:X2
+   5.3725997   1.2112540   0.1009883   3.0090512

eX<-fitted(fit1)
Y1<-(Z-eX)*(Y-beta.hat)
Xp<-cbind(1,X1,X2,X1*X2)
D<-diag(Z*(Z-eX))
psi.ests<-solve((t(Xp) %*% D) %*% Xp) %*% t(Xp) %*% Y1
cbind(psi.hat,psi.ests)  #Results match

+           psi.hat
+ Z       1.1247753 1.1247753
+ Z:X1    1.0759005 1.0759005
+ Z:X2    0.9522669 0.9522669
+ Z:X1:X2 1.1527323 1.1527323
```