

PROPENSITY SCORE ADJUSTMENT WITH PARAMETER ESTIMATION

Consider the data generating (structural) conditional mean model for binary treatment

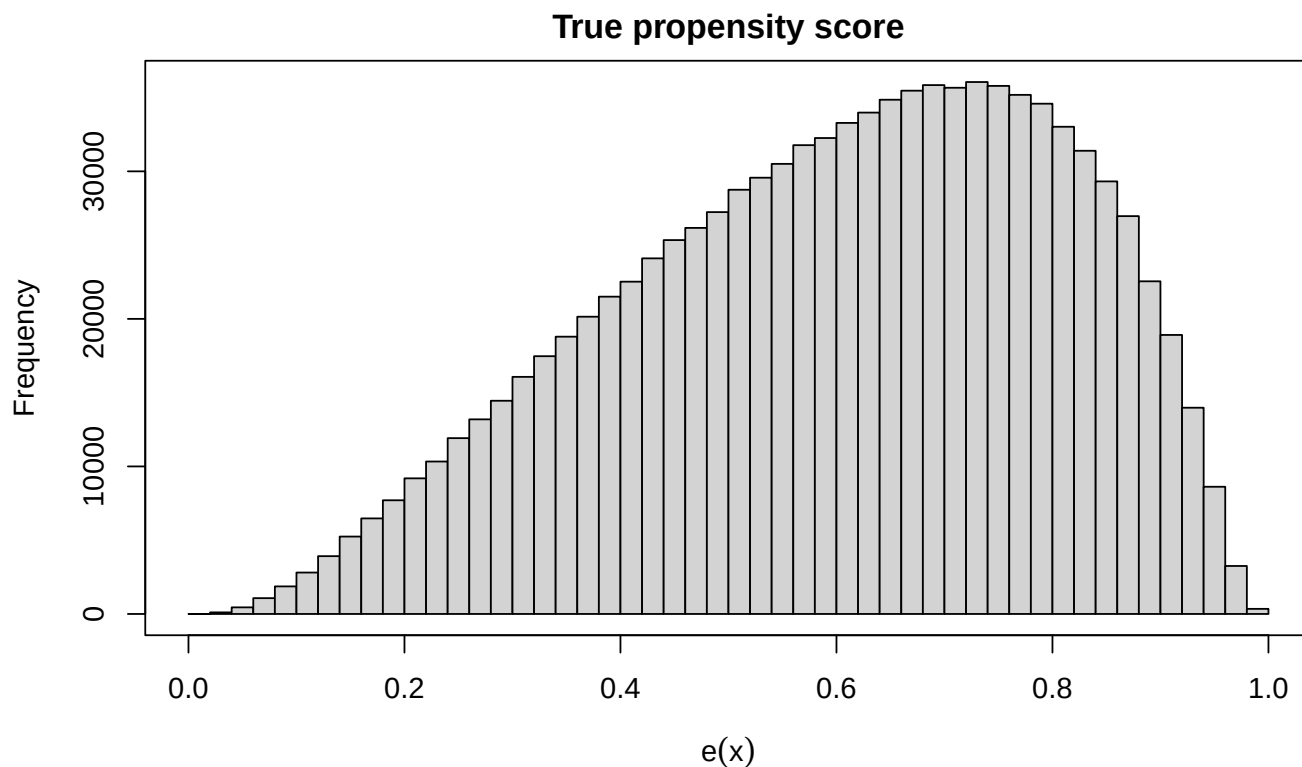
$$\mathbb{E}_{Y|X,Z}^{\circ}[Y|X = x, Z = z] = 2 + 4x + x^2 + 3z$$

and propensity score adjusted estimation of the ATE, which in this case is equal to $\psi_0 = 3$. $X \sim Normal(\mu_X, \sigma_X^2)$ is a confounder, and the conditional model for $Z|X = x$ is $Bernoulli(e(x))$ with

$$e(x) = \frac{\exp\{-3.5 + 2x\}}{1 + \exp\{-3.5 + 2x\}}.$$

In the analysis a parametric model with parameter α estimated via the logistic regression model is used.

```
#Calculation for large N
library(mvnfast)
set.seed(22087)
N<-1000000
muX<-2;sigX<-0.5
X1<-rnorm(N,muX,sigX)
al<-c(-3.5,2)
expit<-function(x){return(1/(1+exp(-x)))}
Xm<-cbind(1,X1)
be<-c(2,3,4,1)
sigY<-3
ps.true<-expit(Xm %*% al)
Z<-rbinom(N,1,ps.true)
Xb<-cbind(1,Z,X1,X1^2)
Y<-Xb %*% be + rnorm(N)*sigY
par(mar=c(4,4,2,0))
hist(ps.true,breaks=seq(0,1,by=0.02),main='True propensity score',xlab=expression(e(x)))
box()
```



It is easy to see that a correctly specified regression model will estimate the average treatment effect (ATE) correctly, and a mis-specified model will not.

```
#Correct model
coef(summary(lm(Y~Z+X1+I(X1^2))))

+             Estimate Std. Error  t value Pr(>|t|)
+ (Intercept) 2.047024  0.034235781   59.79193      0
+ Z           2.985374  0.006709775  444.92902      0
+ X1          3.956903  0.034716756  113.97674      0
+ I(X1^2)      1.010107  0.008490697  118.96631      0

#Incorrect model
coef(summary(lm(Y~Z)))

+             Estimate Std. Error  t value Pr(>|t|)
+ (Intercept) 12.274821  0.007531726 1629.7489      0
+ Z           6.281032  0.009698097  647.6561      0
```

Propensity score regression: Consider OLS estimation using the (mis-specified) mean model:

$$\mathbb{E}_{Y|X,Z}^o[Y|X = x, Z = z] = \beta_0 + z\psi_0 + e(x; \hat{\alpha})\phi_0$$

```
#PS regression: Correct PS model
eX<-fitted(glm(Z~X1,family=binomial))
coef(summary(lm(Y~Z+eX)))

+             Estimate Std. Error  t value Pr(>|t|)
+ (Intercept) 2.551823  0.009875973  258.387      0
+ Z           2.987009  0.007020955  425.442      0
+ eX          19.414736  0.017042626 1139.187      0
```

The estimation is equivalent to OLS estimation that solves an OLS linear system

$$\sum_{i=1}^n \begin{pmatrix} 1 \\ z_i \\ e(x_i; \hat{\alpha}) \end{pmatrix} (y_i - \beta_0 - z_i\psi_0 - e(x_i; \hat{\alpha})\phi_0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

or equivalently

$$\sum_{i=1}^n \begin{pmatrix} 1 \\ (z_i - e(x_i; \hat{\alpha})) \\ e(x_i; \hat{\alpha}) \end{pmatrix} (y_i - \beta_0 - z_i\psi_0 - e(x_i; \hat{\alpha})\phi_0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

to yield

$$\begin{pmatrix} \hat{\beta}_0 \\ \hat{\psi}_0 \\ \hat{\phi}_0 \end{pmatrix} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \quad \mathbf{X} = \begin{pmatrix} 1 & z_1 & e(x_1; \hat{\alpha}) \\ 1 & z_2 & e(x_2; \hat{\alpha}) \\ \vdots & \vdots & \vdots \\ 1 & z_n & e(x_n; \hat{\alpha}) \end{pmatrix}.$$

```
Xm<-cbind(1,Z,eX)
solve(t(Xm) %*% Xm) %*% t(Xm) %*% Y

+             [,1]
+ 2.551823
+ Z 2.987009
+ eX 19.414736
```


However, an alternative **G-estimating** form is

$$\sum_{i=1}^n \begin{pmatrix} 1 \\ (z_i - e(x_i; \hat{\alpha})) \end{pmatrix} (y_i - \beta_0 - z_i \psi_0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

that is, using the estimating (score) function

$$\mathbf{Z}_1^\top (\mathbf{y} - \mathbf{Z}_2 \theta) = \mathbf{0}$$

say, where $\theta = (\beta_0, \psi_0)$.

$$\mathbf{Z}_1 = \begin{pmatrix} 1 & z_1 - e(x_1; \hat{\alpha}) \\ 1 & z_2 - e(x_2; \hat{\alpha}) \\ \vdots & \vdots \\ 1 & z_n - e(x_n; \hat{\alpha}) \end{pmatrix} \quad \mathbf{Z}_2 = \begin{pmatrix} 1 & z_1 \\ 1 & z_2 \\ \vdots & \vdots \\ 1 & z_n \end{pmatrix}$$

The solution is therefore

$$\begin{pmatrix} \hat{\beta}_0 \\ \hat{\psi}_0 \end{pmatrix} = (\mathbf{Z}_1^\top \mathbf{Z}_2)^{-1} \mathbf{Z}_1^\top \mathbf{y}$$

```
Zm1<-cbind(1,Z-eX)
Zm2<-cbind(1,Z)
g.ests<-solve(t(Zm1) %*% Zm2) %*% t(Zm1) %*% Y
g.ests
+      [,1]
+ 14.262253
+ Z    2.985874
```

We can incorporate the estimation of the propensity score parameters into the same estimating equation formulation by adding in the estimating equation for α , namely, for the logistic regression model, the equations

$$\sum_{i=1}^n \mathbf{x}_i^\top (z_i - e(x_i; \alpha)) = \mathbf{0}.$$

We can compute an estimate of the variance-covariance matrix using the theory of estimating equations. For the $p \times 1$ system of estimating equations

$$\sum_{i=1}^n \mathbf{U}_i(\theta) = \mathbf{0}$$

with $\mathbb{E}[\mathbf{U}(\theta_0)] = \mathbf{0}$, we have that

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \xrightarrow{d} \text{Normal}_p(\mathbf{0}, \mathbf{V})$$

where

$$\mathbf{V} = \mathcal{J}^{-1} \mathcal{I} \mathcal{J}^{-\top}$$

with

$$\mathcal{I} = \mathbb{E}[\mathbf{U}(\theta_0) \mathbf{U}(\theta_0)^\top] \quad \mathcal{J} = \mathbb{E}[\dot{\mathbf{U}}(\theta_0)]$$

both $(p \times p)$ matrices, and

$$\dot{\mathbf{U}}(\theta_0) = \left. \frac{\partial \mathbf{U}(\theta)}{\partial \theta^\top} \right|_{\theta=\theta_0}$$

We proceed by computing estimates, $\hat{I}_n$ and $\hat{J}_n$, of the two matrices based on the observed data, and then computing

$$\hat{\mathbf{V}} = \hat{J}_n^{-1} \hat{I}_n \hat{J}_n^{-\top}$$

to estimate the asymptotic variance; this approach is known as *sandwich* or *robust* variance estimation.

- For **propensity score regression** we stack the estimating equations

$$\sum_{i=1}^n \begin{pmatrix} (y_i - \beta_0 - z_i\psi_0 - e(x_i; \alpha)\phi_0) \\ z_i(y_i - \beta_0 - z_i\psi_0 - e(x_i; \alpha)\phi_0) \\ e(x_i; \alpha)(y_i - \beta_0 - z_i\psi_0 - e(x_i; \alpha)\phi_0) \\ \mathbf{x}_i^\top (z_i - e(x_i; \alpha)) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

and then compute the $\mathcal{I}$ and $\mathcal{J}$ and their estimated values in this extended system. The score vector is

$$\mathbf{U}(\theta) = \begin{pmatrix} (Y - \beta_0 - Z\psi_0 - e(X; \alpha)\phi_0) \\ Z(Y - \beta_0 - Z\psi_0 - e(X; \alpha)\phi_0) \\ e(X; \alpha)(Y - \beta_0 - Z\psi_0 - e(X; \alpha)\phi_0) \\ \mathbf{X}^\top (Z - e(X; \alpha)) \end{pmatrix} = \begin{pmatrix} \epsilon_Y \\ Z\epsilon_Y \\ e\epsilon_Y \\ \mathbf{X}^\top \epsilon_Z \end{pmatrix}$$

say, denoting $e \equiv e(X; \alpha)$, where $\mathbf{X} = (1, X)$, and

$$\epsilon_Y = (Y - \beta_0 - Z\psi_0 - e(X; \alpha)\phi_0) \quad \epsilon_Z = (Z - e(X; \alpha)).$$

Then

$$\mathcal{I} = \mathbb{E} \begin{bmatrix} \epsilon_Y^2 & Z\epsilon_Y^2 & e\epsilon_Y^2 & \mathbf{X}\epsilon_Z\epsilon_Y \\ Z\epsilon_Y^2 & Z^2\epsilon_Y^2 & Ze\epsilon_Y^2 & \mathbf{X}Z\epsilon_Z\epsilon_Y \\ e\epsilon_Y^2 & Ze\epsilon_Y^2 & e^2\epsilon_Y^2 & e\epsilon_Z\epsilon_Y \\ \mathbf{X}^\top \epsilon_Z\epsilon_Y & \mathbf{X}^\top Z\epsilon_Z\epsilon_Y & \mathbf{X}^\top e\epsilon_Z\epsilon_Y & \mathbf{X}^\top \mathbf{X}\epsilon_Z^2 \end{bmatrix} \quad \mathcal{J} = \mathbb{E} \begin{bmatrix} 1 & Z & e & \mathbf{X}e(1-e)\phi_0 \\ Z & Z^2 & Ze & Z\mathbf{X}e(1-e)\phi_0 \\ e & Ze & e^2 & \mathbf{X}e(1-e)(e\phi_0 - \epsilon_Y) \\ 0 & 0 & 0 & \mathbf{X}^\top \mathbf{X}e(1-e) \end{bmatrix}.$$

```
#Propensity score regression
psr.ests<-coef(lm(Y~Z+eX))
res1<-Y-psr.ests[1]-psr.ests[2]*Z - psr.ests[3]*eX
res2<-Z-eX
eterm<-eX*(1-eX)
dterm<-eX*psr.ests[3]-res1

Xm1<-cbind(1,Z,eX) * matrix(res1,nrow=N,ncol=3)
Xm2<-cbind(1,X1) * matrix(res2,nrow=N,ncol=2)
Xm<-cbind(Xm1,Xm2)
I.n<-matrix((t(Xm) %*% Xm)/N,5,5)

round(I.n,6)
+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  9.797738  6.088400  6.084857  0.000000 -0.000003
+ [2,]  6.088400  6.088400  4.219443 -0.029306 -0.068972
+ [3,]  6.084857  4.219443  4.217272 -0.000067  0.000067
+ [4,]  0.000000 -0.029306 -0.000067  0.198763  0.382644
+ [5,] -0.000003 -0.068972  0.000067  0.382644  0.771574
J.n<-matrix(0,5,5)
J.n[1:3,1:3]<-crossprod(cbind(1,Z,eX))/N
J.n[1,4:5]<-c(mean(eterm*psr.ests[3]),mean(X1*eterm*psr.ests[3]))
J.n[2,4:5]<-c(mean(Z*eterm*psr.ests[3]),mean(Z*X1*eterm*psr.ests[3]))
J.n[3,4:5]<-c(mean(eterm*dterm),mean(X1*eterm*dterm))
J.n[4:5,4:5]<-t(cbind(1,X1) * matrix(eterm,nrow=N,ncol=2)) %*% cbind(1,X1)/N

round(J.n,6)
+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  1.000000  0.603137  0.603137  3.858473  7.428266
+ [2,]  0.603137  0.603137  0.404386  2.222304  4.565475
+ [3,]  0.603137  0.404386  0.404398  2.251628  4.634533
+ [4,]  0.000000  0.000000  0.000000  0.198739  0.382610
+ [5,]  0.000000  0.000000  0.000000  0.382610  0.771603
```

```
V1<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V1,6)
+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]      790.4649 -0.185000 -1138.13570 -269.115851 126.768023
+ [2,]      -0.1850  47.025920 -46.70646  0.074073 -0.037077
+ [3,] -1138.1357 -46.706456 1880.98509  413.907572 -210.134421
+ [4,] -269.1159  0.074073  413.90757  110.680150 -54.875248
+ [5,] 126.7680 -0.037077 -210.13442 -54.875248  28.503117
```

Thus the asymptotic variance of $\hat{\psi}_0$ (that is, the marginal asymptotic variance of $\sqrt{n}(\hat{\psi}_0 - \psi_0^{\text{TRUE}})$) under propensity score estimation is approximately 47.02592

- For **G-estimation** we stack the estimating equations

$$\sum_{i=1}^n \begin{pmatrix} (y_i - \beta_0 - z_i \psi_0) \\ (z_i - e(x_i; \alpha))(y_i - \beta_0 - z_i \psi_0) \\ \mathbf{x}_i^\top (z_i - e(x_i; \alpha)) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

The score vector is

$$\mathbf{U}(\theta) = \begin{pmatrix} (Y - \beta_0 - Z\psi_0) \\ (Z - e(X; \alpha))(Y - \beta_0 - Z\psi_0) \\ \mathbf{X}^\top (Z - e(X; \alpha)) \end{pmatrix} = \begin{pmatrix} \varepsilon_Y \\ \epsilon_Z \varepsilon_Y \\ \mathbf{X}^\top \epsilon_Z \end{pmatrix}$$

where

$$\varepsilon_Y = (Y - \beta_0 - Z\psi_0)$$

Therefore

$$\mathcal{I} = \mathbb{E} \begin{bmatrix} \varepsilon_Y^2 & \epsilon_Z \varepsilon_Y^2 & \mathbf{X} \epsilon_Z \varepsilon_Y \\ \epsilon_Z \varepsilon_Y^2 & \epsilon_Z^2 \varepsilon_Y^2 & \mathbf{X} \epsilon_Z^2 \varepsilon_Y \\ \mathbf{X}^\top \epsilon_Z \varepsilon_Y & \mathbf{X}^\top \epsilon_Z^2 \varepsilon_Y & \mathbf{X}^\top \mathbf{X} \epsilon_Z^2 \end{bmatrix} \quad \mathcal{J} = \mathbb{E} \begin{bmatrix} 1 & Z & \mathbf{0} \\ \epsilon_Z & Z \epsilon_Z & \mathbf{X} e(1-e) \varepsilon_Y \\ \mathbf{0} & \mathbf{0} & \mathbf{X}^\top \mathbf{X} e(1-e) \end{bmatrix}.$$

Now, the first component of the score vector, ε_Y , is required to have expectation zero; also, we know by assumption that this term does not depend on Z . Therefore,

$$\mathbb{E}_{Y|X,Z}[\varepsilon_Y|X,Z] = h(X) \quad \text{with} \quad \mathbb{E}_X[h(X)] = 0$$

and

$$\mathbb{E}_{Y|X,Z}[\varepsilon_Y^2|X,Z] = v(X).$$

Hence by iterated expectation we must have that

$$\mathbb{E}_{X,Y,Z}[\varepsilon_Y] = 0.$$

Similarly we require by the estimating equation for α that

$$\mathbb{E}_{Z|X}[\epsilon_Z|X] = 0$$

and hence

$$\mathbb{E}_{X,Y,Z}[\epsilon_Z] = \mathbb{E}_{X,Y,Z}[\epsilon_Z \varepsilon_Y^2] = 0 \quad \mathbb{E}_{X,Y,Z}[\mathbf{X} \epsilon_Z \varepsilon_Y] = \mathbf{0}.$$

We also know that

$$\mathbb{E}_{Z|X}[\epsilon_Z^2|X] = \mathbb{E}_{Z|X}[(Z - e(X))^2|X] = e(X)(1 - e(X)).$$

Thus

$$\mathcal{I} = \mathbb{E} \begin{bmatrix} v(X) & 0 & 0 & 0 \\ 0 & e(1-e)v(X) & e(1-e)h(X) & e(1-e)Xh(X) \\ 0 & e(1-e)h(X) & e(1-e) & e(1-e)X \\ 0 & e(1-e)Xh(X) & e(1-e)X & X^2e(1-e) \end{bmatrix} = \begin{bmatrix} \mathcal{I}_{11} & \mathcal{I}_{12} \\ \mathcal{I}_{21} & \mathcal{I}_{22} \end{bmatrix}$$

and

$$\mathcal{J} = \mathbb{E} \begin{bmatrix} 1 & e & 0 & 0 \\ 0 & e(1-e) & e(1-e)h(X) & e(1-e)Xh(X) \\ 0 & 0 & e(1-e) & e(1-e)X \\ 0 & 0 & e(1-e)X & e(1-e)X^2 \end{bmatrix} = \begin{bmatrix} \mathcal{J}_{11} & \mathcal{J}_{12} \\ \mathbf{0} & \mathcal{J}_{22} \end{bmatrix}.$$

On multiplying out, and noting that $\mathcal{I}_{22} = \mathcal{J}_{22}$, we can conclude that $\mathbf{V} = \mathcal{J}^{-1}\mathcal{I}\mathcal{J}^{-\top}$ is **block diagonal**, that is

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22} \end{bmatrix}$$

from which we can conclude that asymptotically $(\hat{\beta}_0, \hat{\psi}_0) \perp\!\!\!\perp \hat{\alpha}$, that is, the estimators are **independent**.

```
Zm1<-cbind(1,Z-eX)
Zm2<-cbind(1,Z)
g.ests<-solve(t(Zm1) %*% Zm2) %*% t(Zm1) %*% Y
res1<-Y-g.ests[1]-g.ests[2]*Z
res2<-Z-eX

I.n<-matrix(0,4,4)
I.n[1,1]<-mean(res1^2)
I.n[1,2]<-I.n[2,1]<-mean(res2*res1^2)
I.n[1,3]<-I.n[3,1]<-mean(res2*res1)
I.n[1,4]<-I.n[4,1]<-mean(X1*res2*res1)
I.n[2,2]<-mean(res2^2*res1^2)
I.n[2,3]<-I.n[3,2]<-mean(res2^2*res1)
I.n[2,4]<-I.n[4,2]<-mean(X1*res2^2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X1*res2^2)
I.n[4,4]<-mean(X1^2*res2^2)

round(I.n,6)
+      [,1]      [,2]      [,3]      [,4]
+ [1,] 25.111776  0.000572  0.000000  0.000100
+ [2,]  0.000572  4.037645 -0.134160  0.015854
+ [3,]  0.000000 -0.134160  0.198763  0.382644
+ [4,]  0.000100  0.015854  0.382644  0.771574
J.n<-matrix(0,4,4)
J.n[1,1]<-1
J.n[1,2]<-mean(Z)
J.n[2,1]<-mean(res2)
J.n[2,2]<-mean(Z*res2)
J.n[2,3]<-mean(eX*(1-eX)*res1)
J.n[2,4]<-mean(X1*eX*(1-eX)*res1)
J.n[3,3]<-mean(eX*(1-eX))
J.n[3,4]<-J.n[4,3]<-mean(X1*eX*(1-eX))
J.n[4,4]<-mean(X1^2*eX*(1-eX))
round(J.n,6)
+      [,1]      [,2]      [,3]      [,4]
+ [1,]  1 0.603137  0.000000  0.000000
+ [2,]  0 0.198751 -0.134313  0.016472
+ [3,]  0 0.000000  0.198739  0.382610
+ [4,]  0 0.000000  0.382610  0.771603
V2<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V2,6)
+      [,1]      [,2]      [,3]      [,4]
+ [1,] 41.652635 -27.423627 -0.065051  0.031792
+ [2,] -27.423627 45.466520  0.098717 -0.047965
+ [3,] -0.065051  0.098717 110.680150 -54.875248
+ [4,]  0.031792 -0.047965 -54.875248 28.503117
```

- **Inverse Probability Weighting:** The original IPW estimators are

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

and in standardized weight form

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

The standardized weight estimators arise from the fit of a weighted least squares problem; if we denote $\beta_0 = \mu(0)$ and $\psi_0 = \mu(1) - \mu(0)$, with $\theta = (\beta_0, \psi_0)$, then

$$(\hat{\beta}_0, \hat{\psi}_0) = \arg \min_{(\beta, \psi)} \sum_{i=1}^n R_i (Y_i - \beta_0 - Z_i \psi_0)^2$$

where for each i

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

say.

We have for the estimating equations

$$\sum_{i=1}^n \begin{pmatrix} r_i(y_i - \beta_0 - z_i \psi_0) \\ r_i z_i (y_i - \beta_0 - z_i \psi_0) \\ \mathbf{x}_i^\top (z_i - e(x_i; \alpha)) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

The score vector is

$$\mathbf{U}(\theta) = \begin{pmatrix} R(Y - \beta_0 - Z\psi_0) \\ RZ(Y - \beta_0 - Z\psi_0) \\ \mathbf{X}^\top (Z - e(X; \alpha)) \end{pmatrix} = \begin{pmatrix} R\epsilon_Y \\ RZ\epsilon_Y \\ \mathbf{X}^\top \epsilon_Z \end{pmatrix}$$

Therefore

$$\mathcal{I} = \mathbb{E} \begin{bmatrix} R^2 \epsilon_Y^2 & R^2 Z \epsilon_Y^2 & \mathbf{X} R \epsilon_Z \epsilon_Y \\ R^2 Z \epsilon_Y^2 & R^2 Z^2 \epsilon_Y^2 & \mathbf{X} R Z \epsilon_Z \epsilon_Y \\ \mathbf{X}^\top R \epsilon_Z \epsilon_Y & \mathbf{X}^\top R Z \epsilon_Z \epsilon_Y & \mathbf{X}^\top \mathbf{X} \epsilon_Z^2 \end{bmatrix} \quad \mathcal{J} = \mathbb{E} \begin{bmatrix} R & RZ & -\dot{\mathbf{R}}^\top \epsilon_Y \\ RZ & RZ^2 & -\dot{\mathbf{R}}^\top Z \epsilon_Y \\ \mathbf{0} & \mathbf{0} & \mathbf{X}^\top \mathbf{X} e(1 - e) \end{bmatrix}.$$

where

$$\dot{\mathbf{R}} = \frac{dR}{d\alpha} = \frac{1 - Z}{1 - e(X)} e(X) \mathbf{X}^\top - \frac{Z}{e(X)} (1 - e(X)) \mathbf{X}^\top.$$

```
eX<-fitted(glm(Z~X1,family=binomial))
w0<-(1-Z)/(1-eX); w0<-w0/sum(w0)
w1<-Z/eX; w1<-w1/sum(w1)
w<-w0+w1

#IPW via original weights
mean(w1*Y)-mean(w0*Y)
+ [1] 2.998142
#IPW via Standardized weights
sum(W1*Y)-sum(W0*Y)
+ [1] 2.993164
```

```

#IPW via regression
fit.w<-lm(Y~Z,weights=w)
coef(summary(fit.w))
+           Estimate Std. Error   t value Pr(>|t|)
+ (Intercept) 14.256245 0.007082108 2012.9946      0
+ Z           2.993164 0.010014720  298.8764      0

ipw.ests<-coef(fit.w)
res1<-Y-ipw.ests[1]-ipw.ests[2]*Z
res2<-Z-eX
R<-w
Rdot1<-((1-Z)/(1-eX))*eX-(Z/eX)*(1-eX)
Rdot2<-X1*Rdot1

I.n<-matrix(0,4,4)
I.n[1,1]<-mean(R^2*res1^2)
I.n[1,2]<-I.n[2,1]<-mean(R^2*Z*res1^2)
I.n[1,3]<-I.n[3,1]<-mean(R*res2*res1)
I.n[1,4]<-I.n[4,1]<-mean(X1*R*res2*res1)
I.n[2,2]<-mean(R^2*Z^2*res1^2)
I.n[2,3]<-I.n[3,2]<-mean(R*Z*res2*res1)
I.n[2,4]<-I.n[4,2]<-mean(X1*R*Z*res2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X1*res2^2)
I.n[4,4]<-mean(X1^2*res2^2)

round(I.n,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,] 214.060261 60.297693 -1.574319 -3.521282
+ [2,]  60.297693 60.297693 -0.787965 -0.765196
+ [3,] -1.574319 -0.787965  0.198763  0.382644
+ [4,] -3.521282 -0.765196  0.382644  0.771574
J.n<-matrix(0,4,4)
J.n[1,1]<-mean(R)
J.n[1,2]<-mean(R*Z)
J.n[1,3]<-mean(-Rdot1*res1)
J.n[1,4]<-mean(-Rdot2*res1)
J.n[2,1]<-mean(R*Z)
J.n[2,2]<-mean(R*Z^2)
J.n[2,3]<-mean(-Rdot1*Z*res1)
J.n[2,4]<-mean(-Rdot2*Z*res1)
J.n[3,3]<-mean(res2^2)
J.n[3,4]<-J.n[4,3]<-mean(X1*res2^2)
J.n[4,4]<-mean(X1^2*res2^2)
round(J.n,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,] 1.999575 0.999966 -1.574319 -3.521282
+ [2,] 0.999966 0.999966 -0.787965 -0.765196
+ [3,] 0.000000 0.000000  0.198763  0.382644
+ [4,] 0.000000 0.000000  0.382644  0.771574
V3<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V3,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,] 106.5614 -82.9376  0.00000  0.00000
+ [2,] -82.9376 100.3144  0.00000  0.00000
+ [3,]  0.0000  0.0000 111.11765 -55.10618
+ [4,]  0.0000  0.0000 -55.10618  28.62466

```

Again, we note that the matrix is block-diagonal, and therefore that asymptotically $(\hat{\beta}_0, \hat{\psi}_0) \perp\!\!\!\perp \hat{\alpha}$, that is, the estimators are **independent**

```

#Monte Carlo study
nreps<-20000
n<-1000
psr.ests<-matrix(0,nrow=nreps,ncol=3)
g.ests<-ipw.ests<-al.ests<-matrix(0,nrow=nreps,ncol=2)
V1.ests<-array(0,c(nreps,5,5))
V2.ests<-V3.ests<-array(0,c(nreps,4,4))
for(irep in 1:nreps){

  X1<-rnorm(n,muX,sigX)
  Xm<-cbind(1,X1)
  ps.true<-expit(Xm %*% al)
  Z<-rbinom(n,1,ps.true)
  Xb<-cbind(1,Z,X1,X1^2)
  Y<-Xb %*% be + rnorm(n)*sigY

  fit.p<-glm(Z~X1,family=binomial)
  eX<-fitted(fit.p)
  w0<-(1-Z)/(1-eX)
  W0<-w0/sum(w0)
  w1<-Z/eX
  W1<-w1/sum(w1)
  w<-w0+w1

  psr.fit<-coef(lm(Y~Z+eX))
  psr.ests[irep,]<-psr.fit
  ipw.ests[irep,]<-coef(summary(lm(Y~Z,weights=w)))[1:2]
  al.ests[irep,]<-coef(fit.p)

  Zm1<-cbind(1,Z-eX)
  Zm2<-cbind(1,Z)
  g.ests[irep,]<-solve(t(Zm1) %*% Zm2) %*% t(Zm1) %*% Y

  #PSR variance estimate
  res1<-Y-psr.fit[1]-psr.fit[2]*Z - psr.fit[3]*eX
  res2<-Z-eX
  eterm<-eX*(1-eX)
  dterm<-eX*psr.fit[3]-res1

  Xm1<-cbind(1,Z,eX) * matrix(res1,nrow=n,ncol=3)
  Xm2<-cbind(1,X1) * matrix(res2,nrow=n,ncol=2)
  Xm<-cbind(Xm1,Xm2)
  I.n<-matrix((t(Xm) %*% Xm)/n,5,5)

  J.n<-matrix(0,5,5)
  J.n[1:3,1:3]<-crossprod(cbind(1,Z,eX))/n
  J.n[1,4:5]<-c(mean(eterm*psr.fit[3]),mean(X1*eterm*psr.fit[3]))
  J.n[2,4:5]<-c(mean(Z*eterm*psr.fit[3]),mean(Z*X1*eterm*psr.fit[3]))
  J.n[3,4:5]<-c(mean(eterm*dterm),mean(X1*eterm*dterm))
  J.n[4:5,4:5]<-t(cbind(1,X1) * matrix(eterm,nrow=n,ncol=2)) %*% cbind(1,X1)/n

  V1.ests[irep,]<-solve(J.n) %*% (I.n %*% t(solve(J.n)))/n

  #G-estimation variance estimate
  res1<-Y-g.ests[irep,1]-g.ests[irep,2]*Z
  res2<-Z-eX

  I.n<-matrix(0,4,4)
  I.n[1,1]<-mean(res1^2)
  I.n[1,2]<-I.n[2,1]<-mean(res2*res1^2)
  I.n[1,3]<-I.n[3,1]<-mean(res2*res1)

```

```

I.n[1,4]<-I.n[4,1]<-mean(X1*res2*res1)
I.n[2,2]<-mean(res2^2*res1^2)
I.n[2,3]<-I.n[3,2]<-mean(res2^2*res1)
I.n[2,4]<-I.n[4,2]<-mean(X1*res2^2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X1*res2^2)
I.n[4,4]<-mean(X1^2*res2^2)

J.n<-matrix(0,4,4)
J.n[1,1]<-1
J.n[1,2]<-mean(Z)
J.n[2,1]<-mean(res2)
J.n[2,2]<-mean(Z*res2)
J.n[2,3]<-mean(eX*(1-eX)*res1)
J.n[2,4]<-mean(X1*eX*(1-eX)*res1)
J.n[3,3]<-mean(eX*(1-eX))
J.n[3,4]<-J.n[4,3]<-mean(X1*eX*(1-eX))
J.n[4,4]<-mean(X1^2*eX*(1-eX))
V<-solve(J.n) %*% (I.n %*% t(solve(J.n)))/n
V2.ests[irep,,]<-V

#IPW Variance estimate
res1<-Y-ipw.ests[irep,1]-ipw.ests[irep,2]*Z
res2<-Z-eX
R<-w
Rdot1<-((1-Z)/(1-eX))*eX-(Z/eX)*(1-eX)
Rdot2<-X1*Rdot1

```

```

I.n<-matrix(0,4,4)
I.n[1,1]<-mean(R^2*res1^2)
I.n[1,2]<-I.n[2,1]<-mean(R^2*Z*res1^2)
I.n[1,3]<-I.n[3,1]<-mean(R*res2*res1)
I.n[1,4]<-I.n[4,1]<-mean(X1*R*res2*res1)
I.n[2,2]<-mean(R^2*Z^2*res1^2)
I.n[2,3]<-I.n[3,2]<-mean(R*Z*res2*res1)
I.n[2,4]<-I.n[4,2]<-mean(X1*R*Z*res2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X1*res2^2)
I.n[4,4]<-mean(X1^2*res2^2)

```

```

J.n<-matrix(0,4,4)
J.n[1,1]<-mean(R)
J.n[1,2]<-mean(R*Z)
J.n[1,3]<-mean(-Rdot1*res1)
J.n[1,4]<-mean(-Rdot2*res1)
J.n[2,1]<-mean(R*Z)
J.n[2,2]<-mean(R*Z^2)
J.n[2,3]<-mean(-Rdot1*Z*res1)
J.n[2,4]<-mean(-Rdot2*Z*res1)
J.n[3,3]<-mean(res2^2)
J.n[3,4]<-J.n[4,3]<-mean(X1*res2^2)
J.n[4,4]<-mean(X1^2*res2^2)

```

```

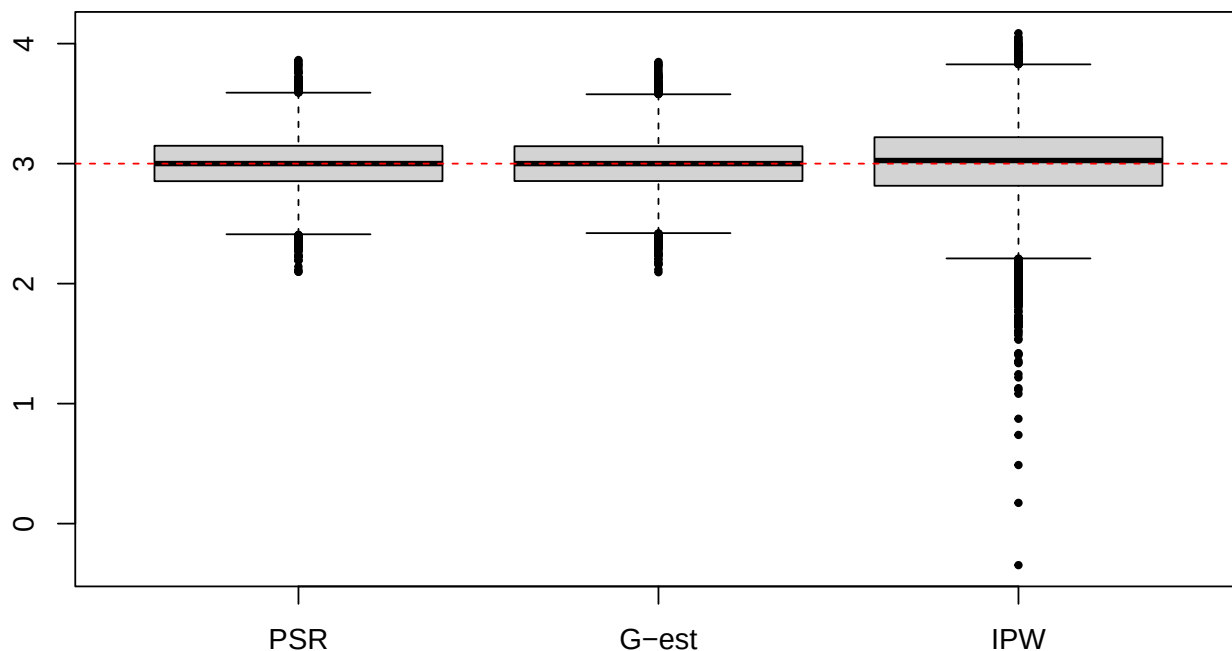
V<-solve(J.n) %*% (I.n %*% t(solve(J.n)))/n
V3.ests[irep,,]<-V

```

```

}

```

```
#Variances
apply(psi0.ests,2,var)
```

```
+      PSR      G-est      IPW
+ 0.04838084 0.04669432 0.10217064
```

```
#Propensity score regression variances
round(apply(V1.ests,2:3,mean),6)      #Monte Carlo sandwich estimate
```

```
+      [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.828733 -0.000239 -1.198205 -0.273191  0.128825
+ [2,] -0.000239  0.047071 -0.046630  0.000087 -0.000047
+ [3,] -1.198205 -0.046630  1.980095  0.420959 -0.213772
+ [4,] -0.273191  0.000087  0.420959  0.111718 -0.055431
+ [5,]  0.128825 -0.000047 -0.213772 -0.055431  0.028804
```

```
round(cov(cbind(psr.ests,al.ests)),6)      #Empirical estimate
```

```
+      [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.817380  0.001317 -1.185538 -0.273812  0.129391
+ [2,]  0.001317  0.048381 -0.049843 -0.000079 -0.000013
+ [3,] -1.185538 -0.049843  1.963864  0.423242 -0.215148
+ [4,] -0.273812 -0.000079  0.423242  0.112057 -0.055699
+ [5,]  0.129391 -0.000013 -0.215148 -0.055699  0.028985
```

```
round(V1/n,6)      #Monte Carlo sandwich estimate (large sample)
```

```
+      [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.790465 -0.000185 -1.138136 -0.269116  0.126768
+ [2,] -0.000185  0.047026 -0.046706  0.000074 -0.000037
+ [3,] -1.138136 -0.046706  1.880985  0.413908 -0.210134
+ [4,] -0.269116  0.000074  0.413908  0.110680 -0.054875
+ [5,]  0.126768 -0.000037 -0.210134 -0.054875  0.028503
```

```

#G-estimation regression variances
round(apply(V2.ests,2:3,mean),6)           #Monte Carlo sandwich estimate

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.041606 -0.027396  0.000002  0.000000
+ [2,] -0.027396  0.045501 -0.000015  0.000007
+ [3,]  0.000002 -0.000015  0.111718 -0.055431
+ [4,]  0.000000  0.000007 -0.055431  0.028804

round(cov(cbind(g.ests,al.ests)),6)       #Empirical estimate

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.041704 -0.027960  0.000047  0.000031
+ [2,] -0.027960  0.046694 -0.000046 -0.000037
+ [3,]  0.000047 -0.000046  0.112057 -0.055699
+ [4,]  0.000031 -0.000037 -0.055699  0.028985

round(V2/n,6)                             #Monte Carlo sandwich estimate (large sample)

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.041653 -0.027424 -0.000065  0.000032
+ [2,] -0.027424  0.045467  0.000099 -0.000048
+ [3,] -0.000065  0.000099  0.110680 -0.054875
+ [4,]  0.000032 -0.000048 -0.054875  0.028503

#IPW variances
round(apply(V3.ests,2:3,mean),6)           #Monte Carlo sandwich estimate

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.100544 -0.077357  0.000000  0.000000
+ [2,] -0.077357  0.095103  0.000000  0.000000
+ [3,]  0.000000  0.000000  0.112317 -0.055743
+ [4,]  0.000000  0.000000 -0.055743  0.028968

round(cov(cbind(ipw.ests,al.ests)),6)     #Empirical estimate

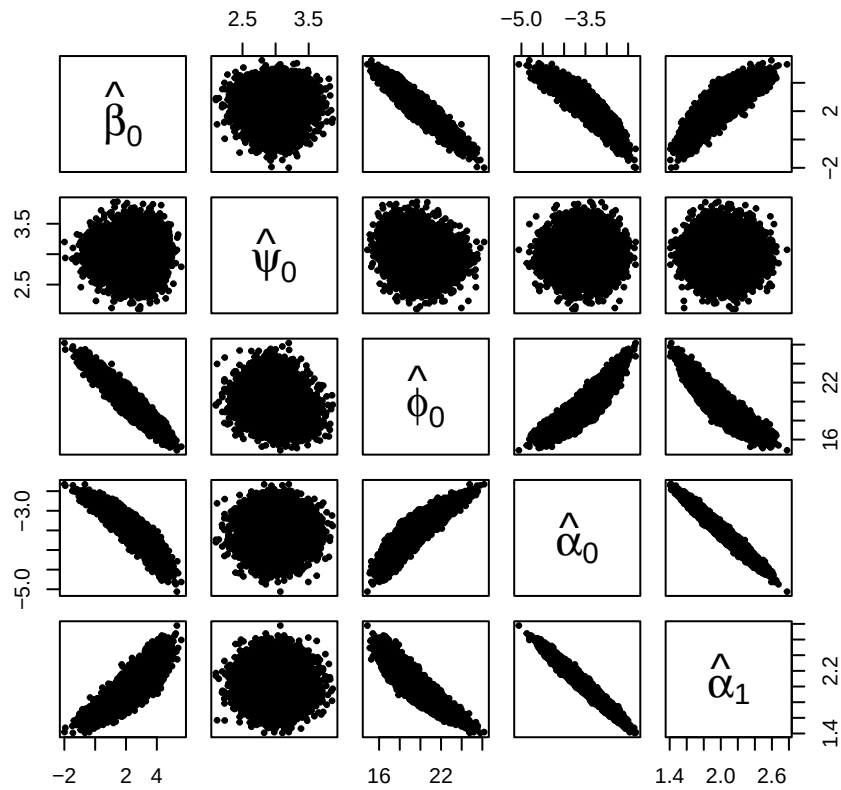
+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.105680 -0.082955  0.001687 -0.000852
+ [2,] -0.082955  0.102171 -0.001404  0.000685
+ [3,]  0.001687 -0.001404  0.112057 -0.055699
+ [4,] -0.000852  0.000685 -0.055699  0.028985

round(V3/n,6)                             #Monte Carlo sandwich estimate (large sample)

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.106561 -0.082938  0.000000  0.000000
+ [2,] -0.082938  0.100314  0.000000  0.000000
+ [3,]  0.000000  0.000000  0.111118 -0.055106
+ [4,]  0.000000  0.000000 -0.055106  0.028625

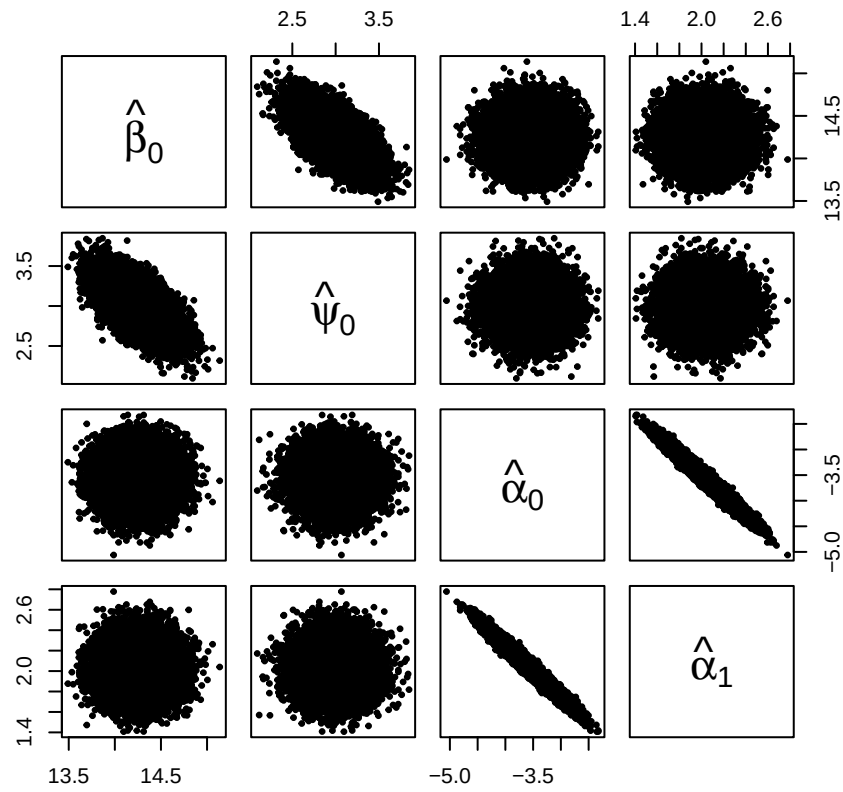
```

PSR



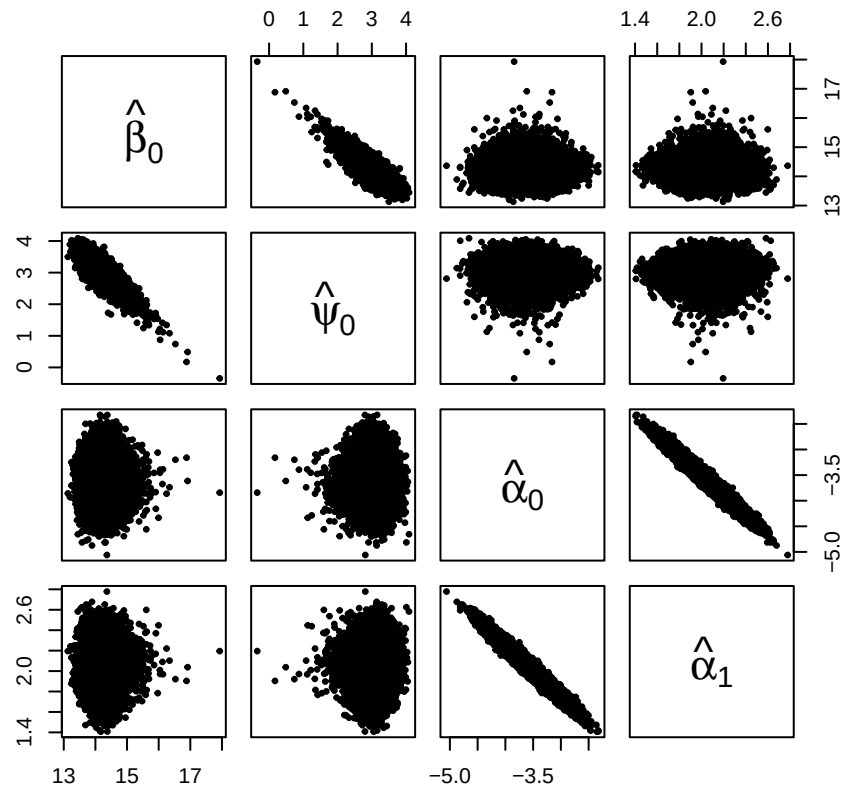
```
+ [1] "Correlation matrix"
+      [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  1.000000000  0.0066234455 -0.9357241 -0.904736328  0.8406336861
+ [2,]  0.006623445  1.0000000000 -0.1617016 -0.001067374 -0.0003475241
+ [3,] -0.935724065 -0.1617015583  1.0000000  0.902224754 -0.9017691976
+ [4,] -0.904736328 -0.0010673742  0.9022248  1.000000000 -0.9773347133
+ [5,]  0.840633686 -0.0003475241 -0.9017692 -0.977334713  1.0000000000
```

G-est



```
+ [1] "Correlation matrix"
+      [,1]      [,2]      [,3]      [,4]
+ [1,] 1.0000000000 -0.6335878461 0.0006905362 0.0008863634
+ [2,] -0.6335878461 1.0000000000 -0.0006340358 -0.0009927897
+ [3,] 0.0006905362 -0.0006340358 1.0000000000 -0.9773347133
+ [4,] 0.0008863634 -0.0009927897 -0.9773347133 1.0000000000
```

IPW



```
+ [1] "Correlation matrix"
+      [,1]      [,2]      [,3]      [,4]
+ [1,]  1.00000000 -0.79833532  0.01550379 -0.01539324
+ [2,] -0.79833532  1.00000000 -0.01312576  0.01257895
+ [3,]  0.01550379 -0.01312576  1.00000000 -0.97733471
+ [4,] -0.01539324  0.01257895 -0.97733471  1.00000000
```

Here the sandwich estimates match the empirical estimates well.