Propensity score adjustment with parameter estimation

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

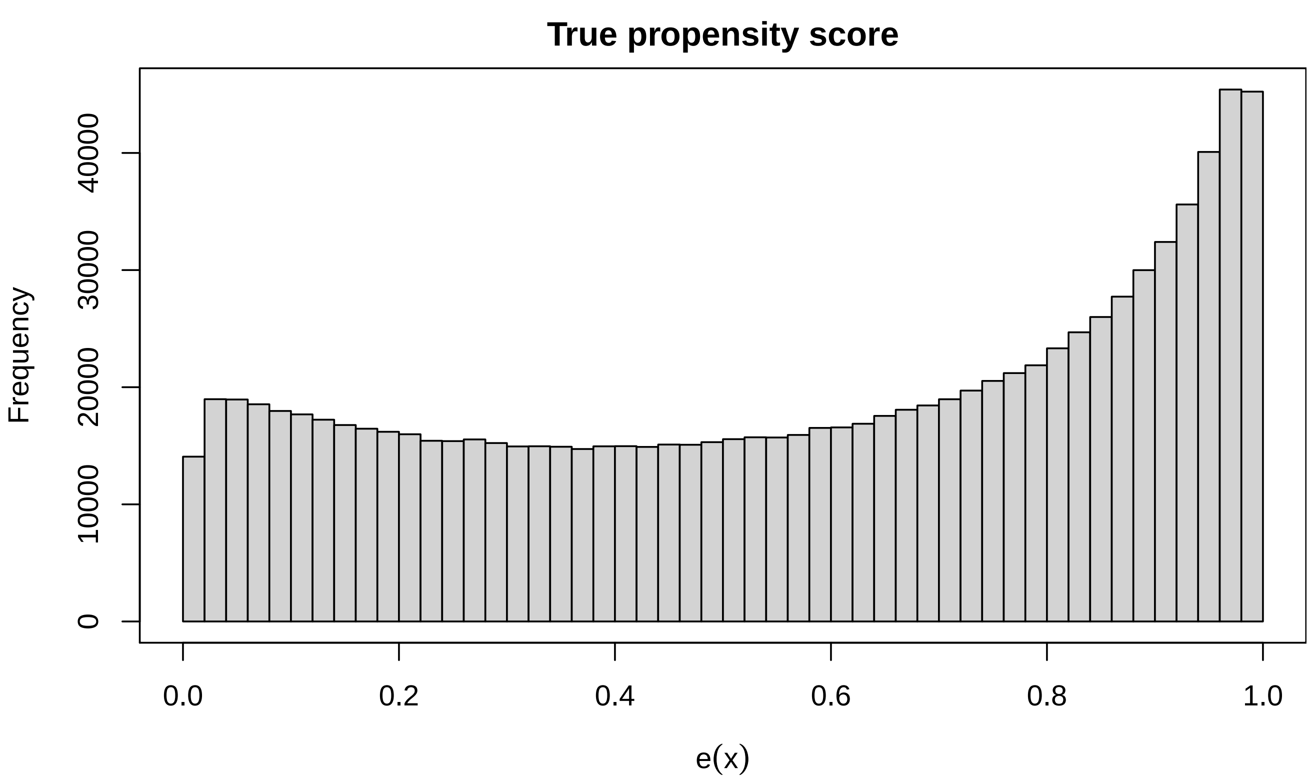
$$\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X=x, Z=z] = 2 + 4x + x^2 + 3z$$

and propensity score adjusted estimation of the ATE , which in this case is equal to $\psi_0 = 3$. $X \sim Normal(\mu_X, \sigma^2_X)$ 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<-1
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) 2.010937 0.009261717 217.1235        0
+ Z           2.991149 0.007611697 392.9673        0
+ X1          3.988195 0.009465141 421.3562        0
+ I(X1^2)     1.002547 0.002127180 471.3034        0

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

+               Estimate Std. Error  t value Pr(>|t|)
+ (Intercept)  9.510201 0.01115860 852.2758        0
+ Z           12.548040 0.01469831 853.7061        0
```

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

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

```
#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)  0.9579624 0.009005296  106.3777        0
+ Z            2.9952251 0.011047092  271.1325        0
+ eX          24.3915121 0.017650859 1381.8881        0
```

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

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

or equivalently

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

to yield

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

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

+          [,1]
+     0.9579624
+ Z   2.9952251
+ eX 24.3915121
```

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

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

that is, using the estimating (score) function

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

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

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

The solution is therefore

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

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

+         [,1]
+   15.017430
+ Z  2.992635
```

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

$$\sqrt{n}(\widehat{\theta}_n - \theta_0) \xrightarrow{d} Normal_p(\mathbf{0}, \mathbf{V})$$

where

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

with

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

both $(ptimesp)$ 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, $\widehat{I}_n$ and $\widehat{J}_n$, of the two matrices based on the observed data, and then computing

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

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

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

Then

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

```
#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,] 18.129763 13.304180 13.301418  0.000000  0.000096
+ [2,] 13.304180 13.304180 11.456160 -0.127319 -0.316910
+ [3,] 13.301418 11.456160 11.450928 -0.000097 -0.000036
+ [4,]  0.000000 -0.127319 -0.000097  0.148558  0.274969
+ [5,]  0.000096 -0.316910 -0.000036  0.274969  0.568771
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 0.576347 0.576347 3.622797 6.704319
+ [2,] 0.576347 0.576347 0.427804 1.950247 4.153025
+ [3,] 0.576347 0.427804 0.427820 2.077296 4.469776
+ [4,] 0.000000 0.000000 0.000000 0.148527 0.274863
+ [5,] 0.000000 0.000000 0.000000 0.274863 0.568601
```

```
V1<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V1,6)
+              [,1]       [,2]       [,3]        [,4]        [,5]
+ [1,]  389.401835  -0.145438 -517.92116 -140.907667   62.737407
+ [2,]   -0.145438  73.544089  -73.23028    0.001664   -0.002807
+ [3,] -517.921164 -73.230277 1019.05521  202.210076 -108.881812
+ [4,] -140.907667   0.001664  202.21008   63.737267  -30.807988
+ [5,]   62.737407  -0.002807 -108.88181  -30.807988   16.650549
```

Thus the asymptotic variance of $\widehat{\psi}_0$ (that is, the marginal asymptotic variance of $\sqrt{n}(\widehat{\psi}_0 - \psi_0^{\text{TRUE}})$) under propensity score estimation is approximately 73.5440887

- 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

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

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,

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

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

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

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 $(\widehat{\beta}_0, \widehat{\psi}_0) \perp\!\!\!\perp \widehat{\alpha}$, 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,] 75.044963 -0.010879  0.000000 -0.000489
+ [2,] -0.010879  5.424493 -0.264698 -0.027910
+ [3,]  0.000000 -0.264698  0.148558  0.274969
+ [4,] -0.000489 -0.027910  0.274969  0.568771
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 0.576347  0.000000  0.000000
+ [2,]    0 0.148543 -0.265539 -0.028291
+ [3,]    0 0.000000  0.148527  0.274863
+ [4,]    0 0.000000  0.274863  0.568601
V2<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V2,6)
+             [,1]      [,2]       [,3]       [,4]
+ [1,]  95.946811 -36.21828  -0.021759   0.004331
+ [2,] -36.218280  62.75816   0.063950  -0.021670
+ [3,]  -0.021759   0.06395  63.737267 -30.807988
+ [4,]   0.004331  -0.02167 -30.807988  16.650549
```

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

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

and in standardized weight form

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

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

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

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

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

Therefore

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

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] 2.969675
#IPW via Standardized weights
sum(W1*Y)-sum(W0*Y)
+ [1] 2.94618
```

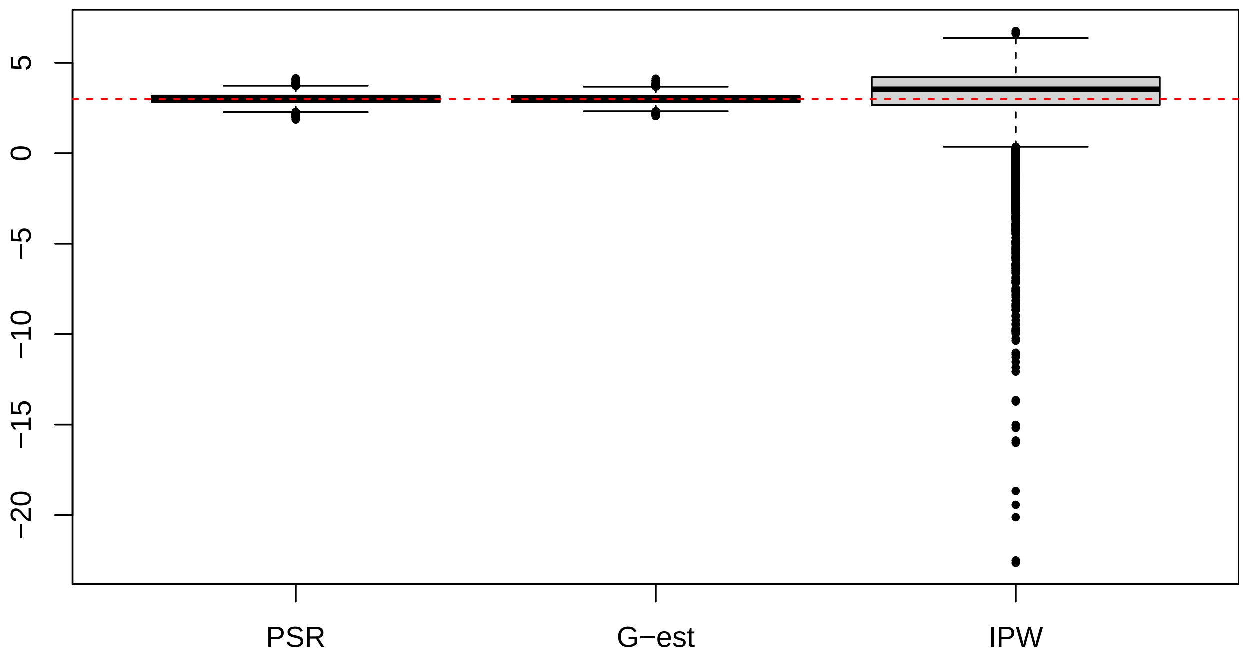
```
#IPW via regression
fit.w<-lm(Y~Z,weights=w)
coef(summary(fit.w))
+               Estimate Std. Error   t value Pr(>|t|)
+ (Intercept) 15.03606 0.01230545 1221.9022        0
+ Z            2.94618 0.01739705  169.3495        0
```

```
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,] 9579.895139 1171.601993 -4.690238 -10.722000
+ [2,] 1171.601993 1171.601993 -2.349192  -1.266542
+ [3,]   -4.690238   -2.349192  0.148558   0.274969
+ [4,]  -10.722000   -1.266542  0.274969   0.568771
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,] 2.001790 1.001527 -4.690238 -10.722000
+ [2,] 1.001527 1.001527 -2.349192  -1.266542
+ [3,] 0.000000 0.000000  0.148558   0.274969
+ [4,] 0.000000 0.000000  0.274969   0.568771
V3<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V3,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,]  7928.643 -7702.218   0.00000   0.00000
+ [2,] -7702.218  8448.542   0.00000   0.00000
+ [3,]     0.000     0.000  63.99521 -30.93813
+ [4,]     0.000     0.000 -30.93813  16.71505
```

Again, we note that the matrix is block-diagonal, and therefore that asymptotically $(\widehat{\beta}_0, \widehat{\psi}_0) \perp\!\!\!\perp \widehat{\alpha}$, 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)
```

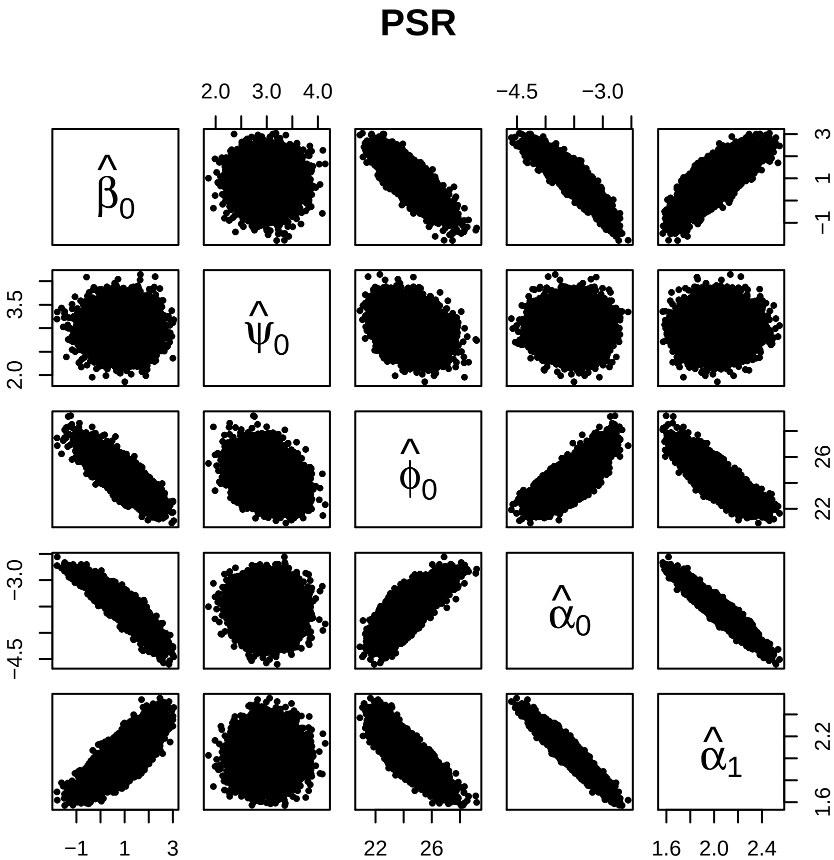
```
    I.n[1,4]<-I.n[4,1]<-mean(X1*res2*res1)
    I.n[2,2]<-mean(res2^2*res1^2)
    I.n[2,3]<-I.n[3,2]<-mean(res2^2*res1)
    I.n[2,4]<-I.n[4,2]<-mean(X1*res2^2*res1)
    I.n[3,3]<-mean(res2^2)
    I.n[3,4]<-I.n[4,3]<-mean(X1*res2^2)
    I.n[4,4]<-mean(X1^2*res2^2)

    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
}
```



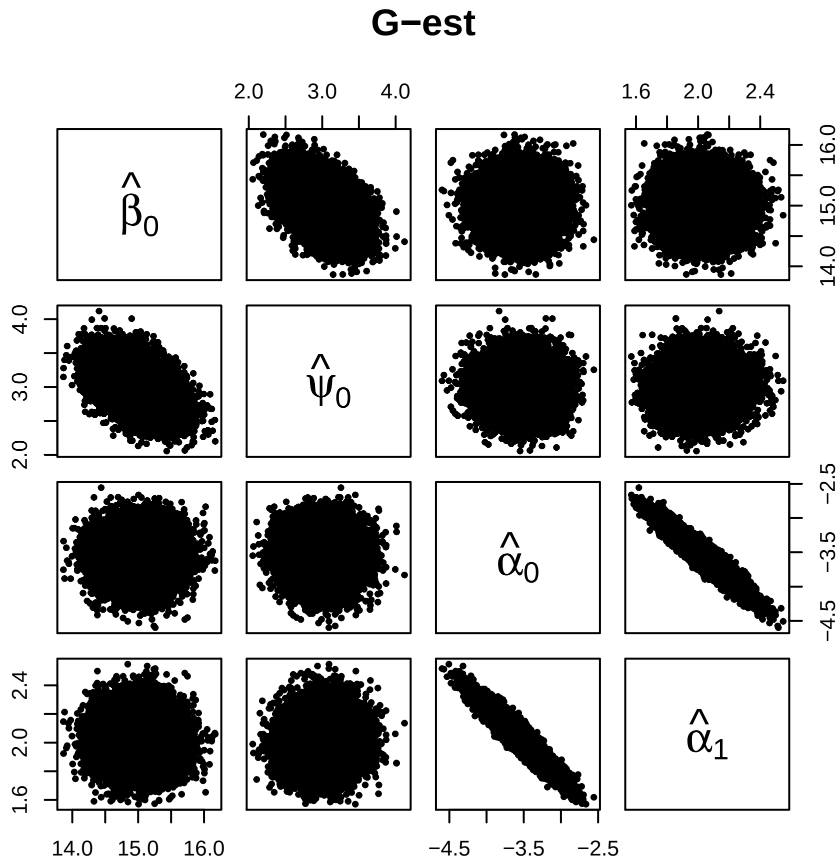

```
#Variances
apply(psi0.ests,2,var)

+        PSR      G-est        IPW
+ 0.07475169 0.06373594 2.78898295
```

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

+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.392423 -0.000618 -0.522134 -0.141295  0.062976
+ [2,] -0.000618  0.073883 -0.072663  0.000312 -0.000175
+ [3,] -0.522134 -0.072663  1.026348  0.202813 -0.109241
+ [4,] -0.141295  0.000312  0.202813  0.064824 -0.031389
+ [5,]  0.062976 -0.000175 -0.109241 -0.031389  0.016972

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

+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.385977  0.002562 -0.518380 -0.140862  0.062989
+ [2,]  0.002562  0.074752 -0.078825 -0.001000  0.000470
+ [3,] -0.518380 -0.078825  1.028160  0.203539 -0.109868
+ [4,] -0.140862 -0.001000  0.203539  0.064855 -0.031415
+ [5,]  0.062989  0.000470 -0.109868 -0.031415  0.016991

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

+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.389402 -0.000145 -0.517921 -0.140908  0.062737
+ [2,] -0.000145  0.073544 -0.073230  0.000002 -0.000003
+ [3,] -0.517921 -0.073230  1.019055  0.202210 -0.108882
+ [4,] -0.140908  0.000002  0.202210  0.063737 -0.030808
+ [5,]  0.062737 -0.000003 -0.108882 -0.030808  0.016651
```

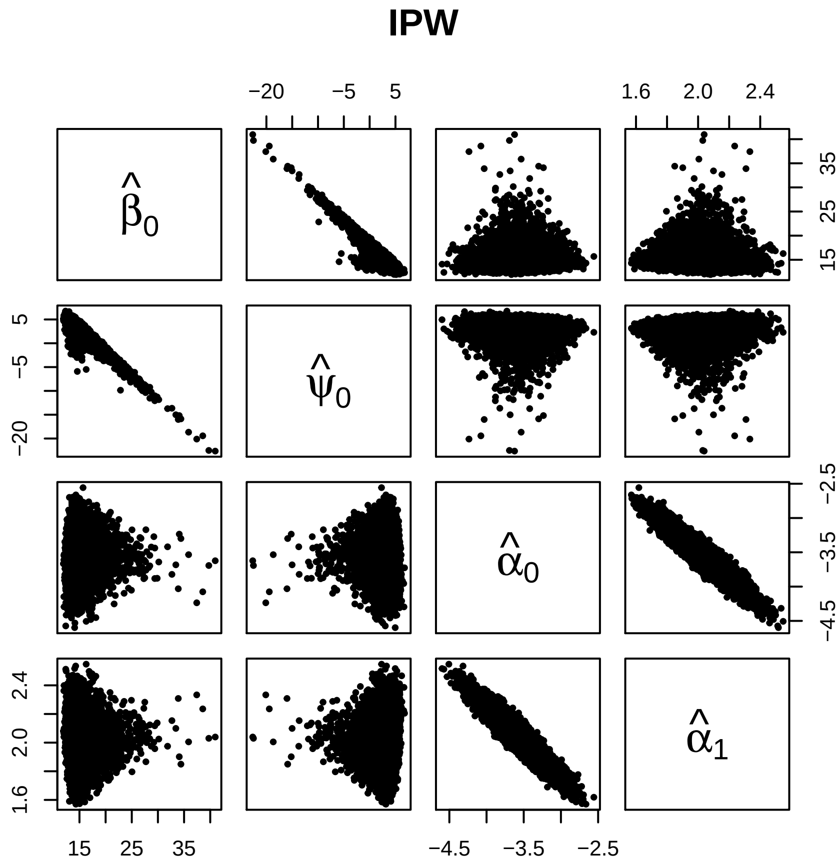
```r
#G-estimation regression variances
round(apply(V2.ests,2:3,mean),6)          #Monte Carlo sandwich estimate

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.095774 -0.036216 -0.000033  0.000020
+ [2,] -0.036216  0.062994  0.000016 -0.000012
+ [3,] -0.000033  0.000016  0.064824 -0.031389
+ [4,]  0.000020 -0.000012 -0.031389  0.016972

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.096138 -0.036707  0.000940 -0.000393
+ [2,] -0.036707  0.063736 -0.001208  0.000544
+ [3,]  0.000940 -0.001208  0.064855 -0.031415
+ [4,] -0.000393  0.000544 -0.031415  0.016991

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.095947 -0.036218 -0.000022  0.000004
+ [2,] -0.036218  0.062758  0.000064 -0.000022
+ [3,] -0.000022  0.000064  0.063737 -0.030808
+ [4,]  0.000004 -0.000022 -0.030808  0.016651

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  1.653293 -1.480072  0.000000  0.000000
+ [2,] -1.480072  1.697533  0.000000  0.000000
+ [3,]  0.000000  0.000000  0.065423 -0.031713
+ [4,]  0.000000  0.000000 -0.031713  0.017151

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  2.693965 -2.492213  0.010184 -0.006237
+ [2,] -2.492213  2.788983 -0.013542  0.007901
+ [3,]  0.010184 -0.013542  0.064855 -0.031415
+ [4,] -0.006237  0.007901 -0.031415  0.016991

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  7.928643 -7.702218  0.000000  0.000000
+ [2,] -7.702218  8.448542  0.000000  0.000000
+ [3,]  0.000000  0.000000  0.063995 -0.030938
+ [4,]  0.000000  0.000000 -0.030938  0.016715
```




```
+ [1] "Correlation matrix"
+             [,1]        [,2]       [,3]        [,4]        [,5]
+ [1,]  1.00000000  0.01508329 -0.8228810 -0.89031704  0.77781495
+ [2,]  0.01508329  1.00000000 -0.2843316 -0.01436736  0.01318209
+ [3,] -0.82288096 -0.28433163  1.0000000  0.78822038 -0.83125673
+ [4,] -0.89031704 -0.01436736  0.7882204  1.00000000 -0.94637964
+ [5,]  0.77781495  0.01318209 -0.8312567 -0.94637964  1.00000000
```




```
+ [1] "Correlation matrix"
+             [,1]        [,2]        [,3]        [,4]
+ [1,]  1.00000000 -0.46892909  0.01190604 -0.00971323
+ [2,] -0.46892909  1.00000000 -0.01879017  0.01654256
+ [3,]  0.01190604 -0.01879017  1.00000000 -0.94637964
+ [4,] -0.00971323  0.01654256 -0.94637964  1.00000000
```




```
+ [1] "Correlation matrix"
+             [,1]        [,2]        [,3]        [,4]
+ [1,]  1.00000000 -0.90921409  0.02436316 -0.02915466
+ [2,] -0.90921409  1.00000000 -0.03184033  0.03629398
+ [3,]  0.02436316 -0.03184033  1.00000000 -0.94637964
+ [4,] -0.02915466  0.03629398 -0.94637964  1.00000000
```

Here the sandwich estimates match the empirical estimates well for PSR and G-estimation, but less well for the IPW estimator due to the presence of extreme weights.