

PROPENSITY SCORE ADJUSTMENTS

CHECKING ASYMPTOTIC VARIANCE CALCULATIONS

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

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

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

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

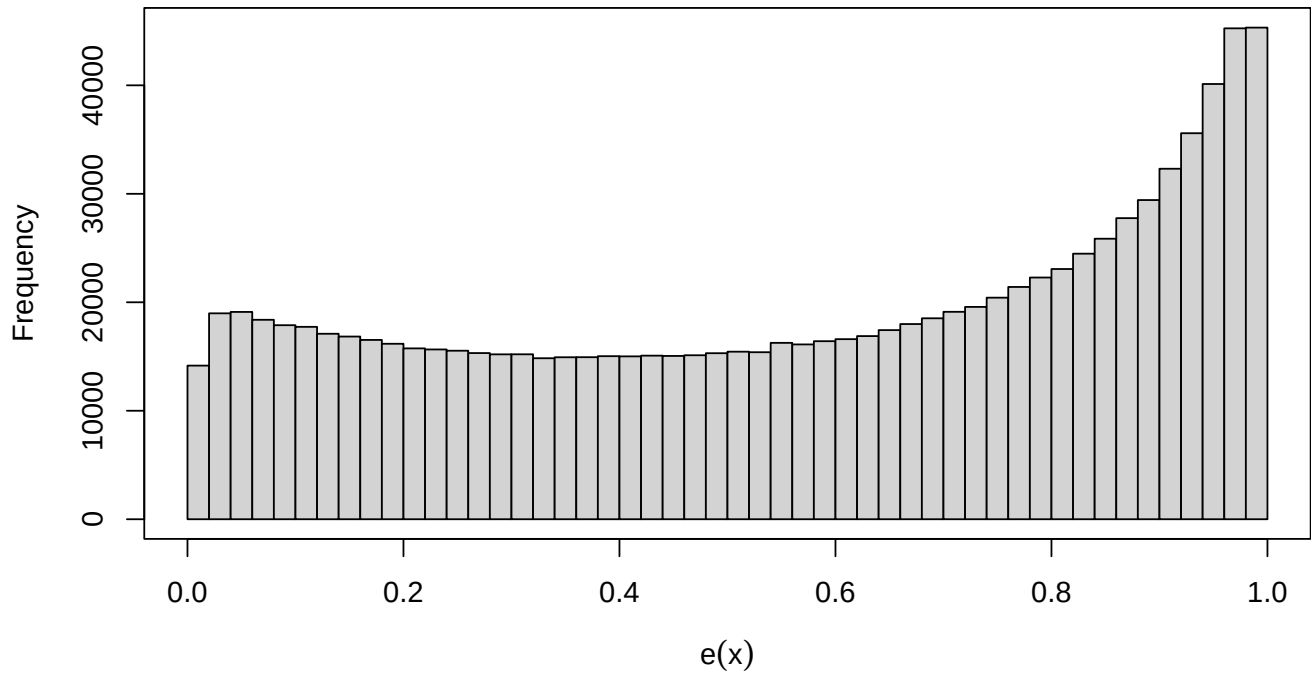
In the analysis a parametric model with parameter α estimated via the logistic regression model is used; and the estimating (score) equation is

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

1. Simulation settings

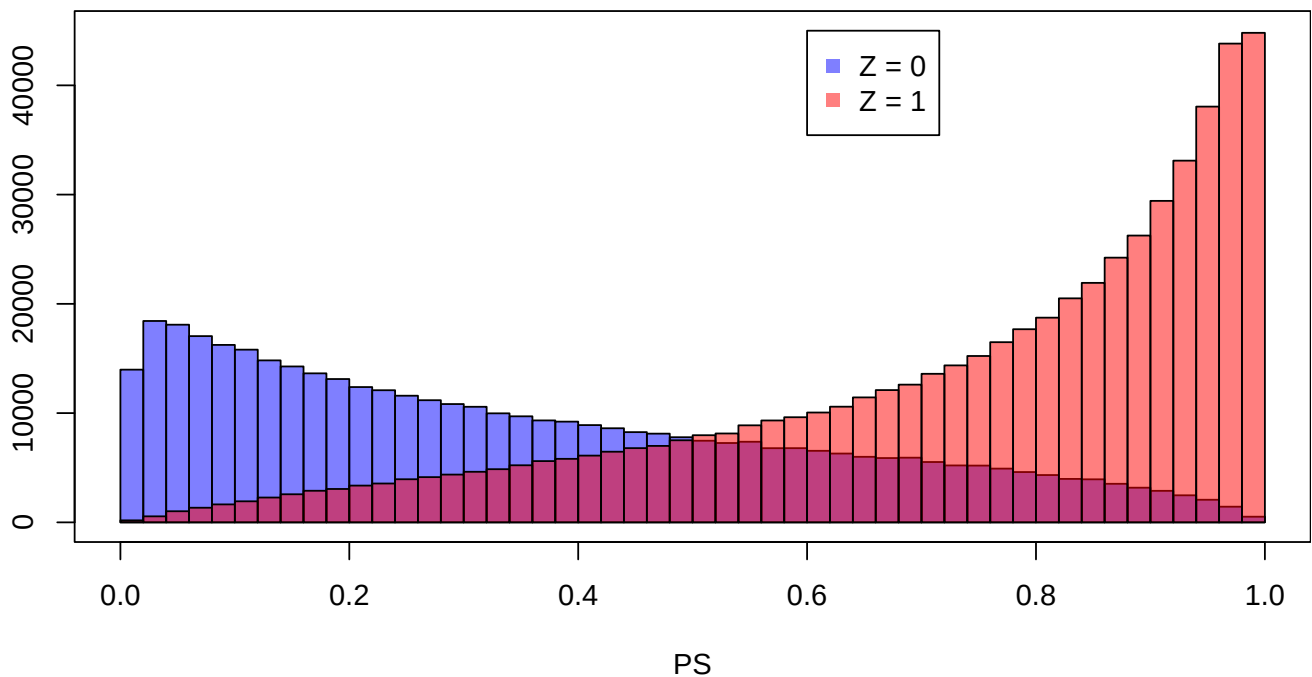
```
library(mvnfast)
set.seed(2207); N<-1000000
muX<-2;sigX<-1
X<-rnorm(N,muX,sigX)
al<-c(-3.5,2); expit<-function(x){return(1/(1+exp(-x)))}
Xm<-cbind(1,X); be<-c(2,3,4,1); sigY<-3
ps.true<-expit(Xm %*% al)
Z<-rbinom(N,1,ps.true)
Xb<-cbind(1,Z,X,X^2)
Y<-Xb %*% be + rnorm(N)*sigY
par(mar=c(4,4,2,0))
hist(ps.true,breaks=seq(0,1,by=0.02),main='True propensity score',xlab=expression(e(x)))
box()
```

True propensity score



```
par(mar=c(4,2,2,0))
colvec<-c(rgb(0,0,1,0.5),rgb(1,0,0,0.5))
hist(ps.true[Z==0], col=rgb(0,0,1,0.5), breaks=seq(0,1,by=0.02), ylim=c(0,45000),
     main="Propensity Score overlap", xlab="PS")
hist(ps.true[Z==1], col=rgb(1,0,0,0.5), breaks=seq(0,1,by=0.02), add=T);box()
legend(0.6,45000,c('Z = 0', 'Z = 1'),col=colvec,pch=15)
```

Propensity Score overlap



1.1 Estimation

It is easy to see that a correctly specified regression model will estimate the average treatment effect (ATE) correctly,

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

+             Estimate Std. Error  t value Pr(>|t|)
+ (Intercept)  1.9845627  0.009265639  214.1852      0
+ Z            2.9910348  0.007625811  392.2252      0
+ X            4.0202732  0.009461585  424.9048      0
+ I(X^2)       0.9959736  0.002125705  468.5379      0

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

+             Estimate Std. Error  t value Pr(>|t|)
+ (Intercept)   9.510127  0.01116999  851.3998      0
+ Z            12.552638  0.01472004  852.7584      0
```

In the first (correctly specified) analysis, the ATE estimate is 2.9910348; in the second (incorrectly specified) analysis, the ATE estimate is 12.5526381.

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

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

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

+             Estimate Std. Error  t value Pr(>|t|)
+ (Intercept)   0.9426632  0.009019143  104.5180      0
+ Z            2.9876279  0.011064236  270.0257      0
+ eX           24.4437032  0.017692035 1381.6219      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]
+ 0.9426632
+ Z    2.9876279
+ eX   24.4437032
```

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

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

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]
+ 15.01305
+ Z    2.99599
```

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

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

2. Asymptotic variance calculations

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

$$\mathbf{V} = \mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-\top}$$

with

$$\mathbf{B} = \mathbb{E}[\mathbf{U}(\theta_0) \mathbf{U}(\theta_0)^\top] \quad \mathbf{A} = \mathbb{E}[\dot{\mathbf{U}}(\theta_0)]$$

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, $\hat{\mathbf{B}}_n$ and $\hat{\mathbf{A}}_n$, of the two matrices based on the observed data, and then computing

$$\hat{\mathbf{V}} = \hat{\mathbf{A}}_n^{-1} \hat{\mathbf{B}}_n \hat{\mathbf{A}}_n^{-\top}$$

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 $\mathbf{B}$ and $\mathbf{A}$ 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

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

```
#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,X) * 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,] 18.193963 13.344788 13.337365 0.000000 -0.000337
+ [2,] 13.344788 13.344788 11.480577 -0.127067 -0.316668
+ [3,] 13.337365 11.480577 11.476643 -0.000057 -0.000560
+ [4,] 0.000000 -0.127067 -0.000057 0.148623 0.274828
+ [5,] -0.000337 -0.316668 -0.000560 0.274828 0.568606
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(X*eterm*psr.ests[3]))
J.n[2,4:5]<-c(mean(Z*eterm*psr.ests[3]),mean(Z*X*eterm*psr.ests[3]))
J.n[3,4:5]<-c(mean(eterm*dterm),mean(X*eterm*dterm))
J.n[4:5,4:5]<-t(cbind(1,X) * matrix(eterm,nrow=N,ncol=2)) %*% cbind(1,X)/N

round(J.n,6)
+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,] 1.000000 0.575821 0.575821 3.635381 6.726156
+ [2,] 0.575821 0.575821 0.427147 1.954524 4.162668
+ [3,] 0.575821 0.427147 0.427096 2.082627 4.480298
+ [4,] 0.000000 0.000000 0.000000 0.148725 0.275169
+ [5,] 0.000000 0.000000 0.000000 0.275169 0.569089
V1<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V1,6)
+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,] 394.310136 0.122029 -525.94291 -142.51695 63.557801
+ [2,] 0.122029 74.357365 -74.73247 -0.02234 0.031491
+ [3,] -525.942910 -74.732468 1034.91339 204.92158 -110.310945
+ [4,] -142.516949 -0.022340 204.92158 64.26997 -31.088160
+ [5,] 63.557801 0.031491 -110.31095 -31.08816 16.793415

```

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 74.3573652

- 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

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

where

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

Therefore

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

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

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

and

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

On multiplying out, and noting that $\mathbf{B}_{22} = \mathbf{A}_{22}$, we can conclude that $\mathbf{V} = \mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-\top}$ 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(X*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(X*res2^2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X*res2^2)
I.n[4,4]<-mean(X^2*res2^2)

round(I.n,6)
+      [,1]      [,2]      [,3]      [,4]
+ [1,] 75.231488  0.028432  0.000000  0.001529
+ [2,]  0.028432  5.475285 -0.266466 -0.026163
+ [3,]  0.000000 -0.266466  0.148623  0.274828
+ [4,]  0.001529 -0.026163  0.274828  0.568606
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(X*eX*(1-eX)*res1)
J.n[3,3]<-mean(eX*(1-eX))
J.n[3,4]<-J.n[4,3]<-mean(X*eX*(1-eX))
J.n[4,4]<-mean(X^2*eX*(1-eX))
round(J.n,6)
+      [,1]      [,2]      [,3]      [,4]
+ [1,]  1 0.575821  0.000000  0.000000
+ [2,]  0 0.148674 -0.264902 -0.026644
+ [3,]  0 0.000000  0.148725  0.275169
+ [4,]  0 0.000000  0.275169  0.569089
V2<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V2,6)
+      [,1]      [,2]      [,3]      [,4]
+ [1,] 96.005556 -36.189077 -0.057880  0.039247
+ [2,] -36.189077  63.041903  0.018613 -0.023890
+ [3,] -0.057880  0.018613  64.269972 -31.088160
+ [4,]  0.039247 -0.023890 -31.088160  16.793415

```

- **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

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

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~X,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] 2.867621
#IPW via Standardized weights
sum(W1*Y)-sum(W0*Y)
+ [1] 2.894396
#IPW via regression
fit.w<-lm(Y~Z,weights=w)
coef(summary(fit.w))
+           Estimate Std. Error  t value Pr(>|t|)
+ (Intercept) 15.085537 0.01230896 1225.574      0
+ Z           2.894396 0.01741671  166.185      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<-X*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(X*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(X*R*Z*res2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X*res2^2)
I.n[4,4]<-mean(X^2*res2^2)

round(I.n,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,] 6047.282756 711.948592 -4.715952 -10.876144
+ [2,] 711.948592 711.948592 -2.350981 -1.323286
+ [3,] -4.715952 -2.350981 0.148623 0.274828
+ [4,] -10.876144 -1.323286 0.274828 0.568606
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(X*res2^2)
J.n[4,4]<-mean(X^2*res2^2)
round(J.n,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,] 2.005725 1.001802 -4.715952 -10.876144
+ [2,] 1.001802 1.001802 -2.350981 -1.323286
+ [3,] 0.000000 0.000000 0.148623 0.274828
+ [4,] 0.000000 0.000000 0.274828 0.568606
V3<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V3,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,] 4815.677 -4595.031 0.00000 0.00000
+ [2,] -4595.031 4895.861 0.00000 0.00000
+ [3,] 0.000 0.000 63.33930 -30.61427
+ [4,] 0.000 0.000 -30.61427 16.55572

```

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**

3. Monte Carlo study

```
#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){

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

  fit.p<-glm(Z~X,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,X) * 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(X*eterm*psr.fit[3]))
  J.n[2,4:5]<-c(mean(Z*eterm*psr.fit[3]),mean(Z*X*eterm*psr.fit[3]))
  J.n[3,4:5]<-c(mean(eterm*dterm),mean(X*eterm*dterm))
  J.n[4:5,4:5]<-t(cbind(1,X) * matrix(eterm,nrow=n,ncol=2)) %*% cbind(1,X)/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(X*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(X*res2^2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X*res2^2)
I.n[4,4]<-mean(X^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(X*eX*(1-eX)*res1)
J.n[3,3]<-mean(eX*(1-eX))
J.n[3,4]<-J.n[4,3]<-mean(X*eX*(1-eX))
J.n[4,4]<-mean(X^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<-X*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(X*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(X*R*Z*res2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X*res2^2)
I.n[4,4]<-mean(X^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)

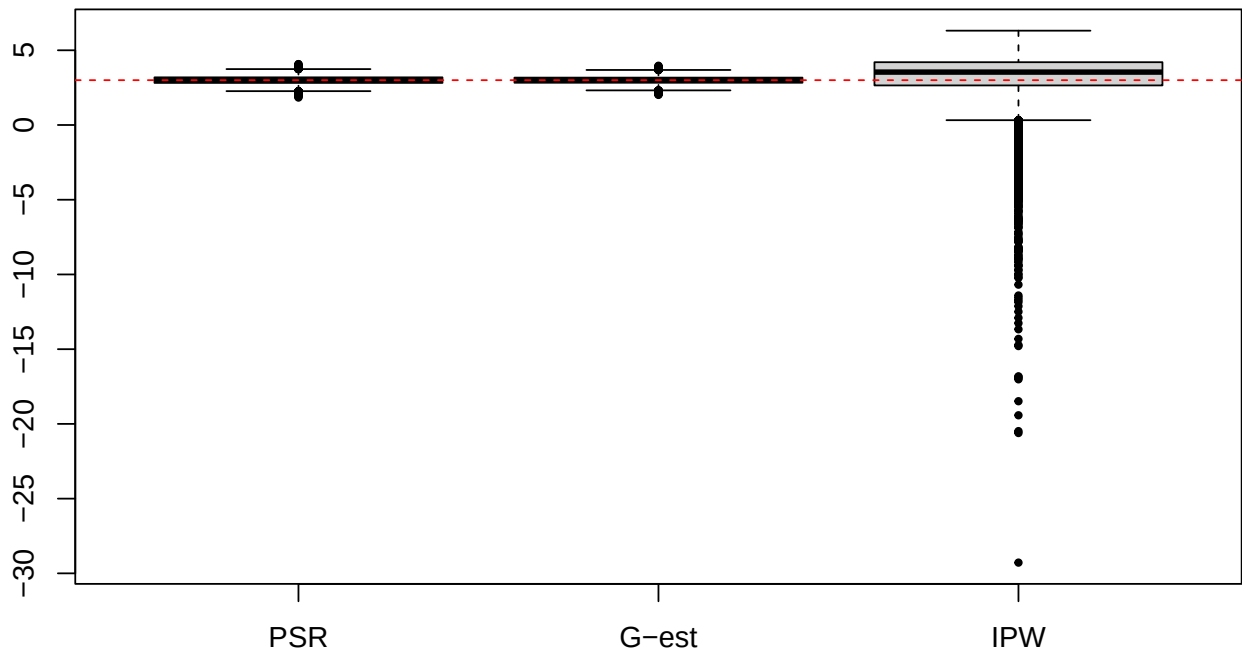
```

```

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(X*res2^2)
J.n[4,4]<-mean(X^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
+ 0.07470036 0.06362183 2.89020565

```

```

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

```

```

+      [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,] 0.392418 -0.000469 -0.522335 -0.141285 0.062975
+ [2,] -0.000469 0.073941 -0.072934 0.000242 -0.000140
+ [3,] -0.522335 -0.072934 1.027028 0.202932 -0.109314
+ [4,] -0.141285 0.000242 0.202932 0.064826 -0.031394
+ [5,] 0.062975 -0.000140 -0.109314 -0.031394 0.016978

```

```

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

```

```

+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.388640 -0.001339 -0.517907 -0.141100  0.063004
+ [2,] -0.001339  0.074700 -0.072702  0.000459 -0.000239
+ [3,] -0.517907 -0.072702  1.021071  0.202801 -0.109186
+ [4,] -0.141100  0.000459  0.202801  0.064826 -0.031439
+ [5,]  0.063004 -0.000239 -0.109186 -0.031439  0.016999

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

+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.394310  0.000122 -0.525943 -0.142517  0.063558
+ [2,]  0.000122  0.074357 -0.074732 -0.000022  0.000031
+ [3,] -0.525943 -0.074732  1.034913  0.204922 -0.110311
+ [4,] -0.142517 -0.000022  0.204922  0.064270 -0.031088
+ [5,]  0.063558  0.000031 -0.110311 -0.031088  0.016793

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.095764 -0.036201  0.000018 -0.000008
+ [2,] -0.036201  0.062962 -0.000001 -0.000002
+ [3,]  0.000018 -0.000001  0.064826 -0.031394
+ [4,] -0.000008 -0.000002 -0.031394  0.016978

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.097339 -0.037018 -0.001182  0.000625
+ [2,] -0.037018  0.063622  0.000319 -0.000144
+ [3,] -0.001182  0.000319  0.064826 -0.031439
+ [4,]  0.000625 -0.000144 -0.031439  0.016999

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.096006 -0.036189 -0.000058  0.000039
+ [2,] -0.036189  0.063042  0.000019 -0.000024
+ [3,] -0.000058  0.000019  0.064270 -0.031088
+ [4,]  0.000039 -0.000024 -0.031088  0.016793

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  1.672038 -1.498654  0.000000  0.000000
+ [2,] -1.498654  1.715373  0.000000  0.000000
+ [3,]  0.000000  0.000000  0.065552 -0.031765
+ [4,]  0.000000  0.000000 -0.031765  0.017170

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

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  2.809452 -2.594206  0.005832 -0.002761
+ [2,] -2.594206  2.890206 -0.010222  0.004824
+ [3,]  0.005832 -0.010222  0.064826 -0.031439
+ [4,] -0.002761  0.004824 -0.031439  0.016999

```

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

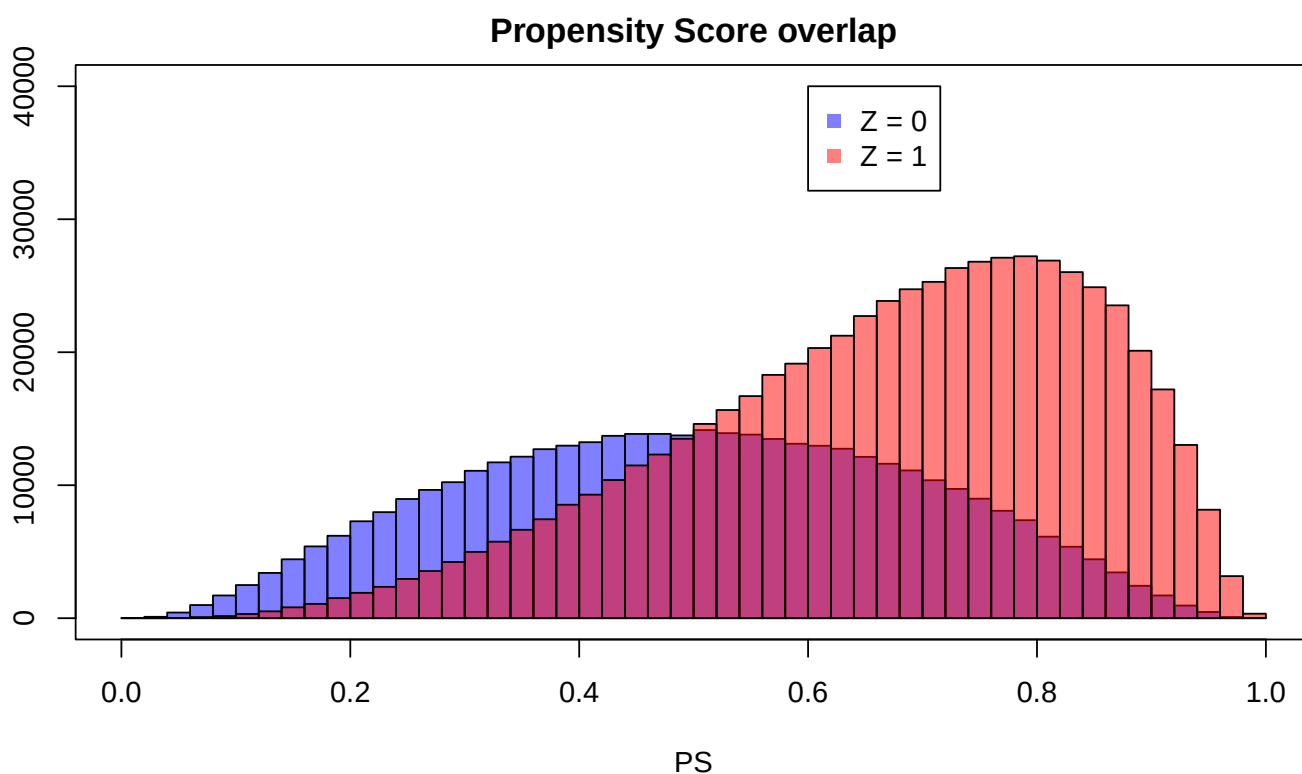
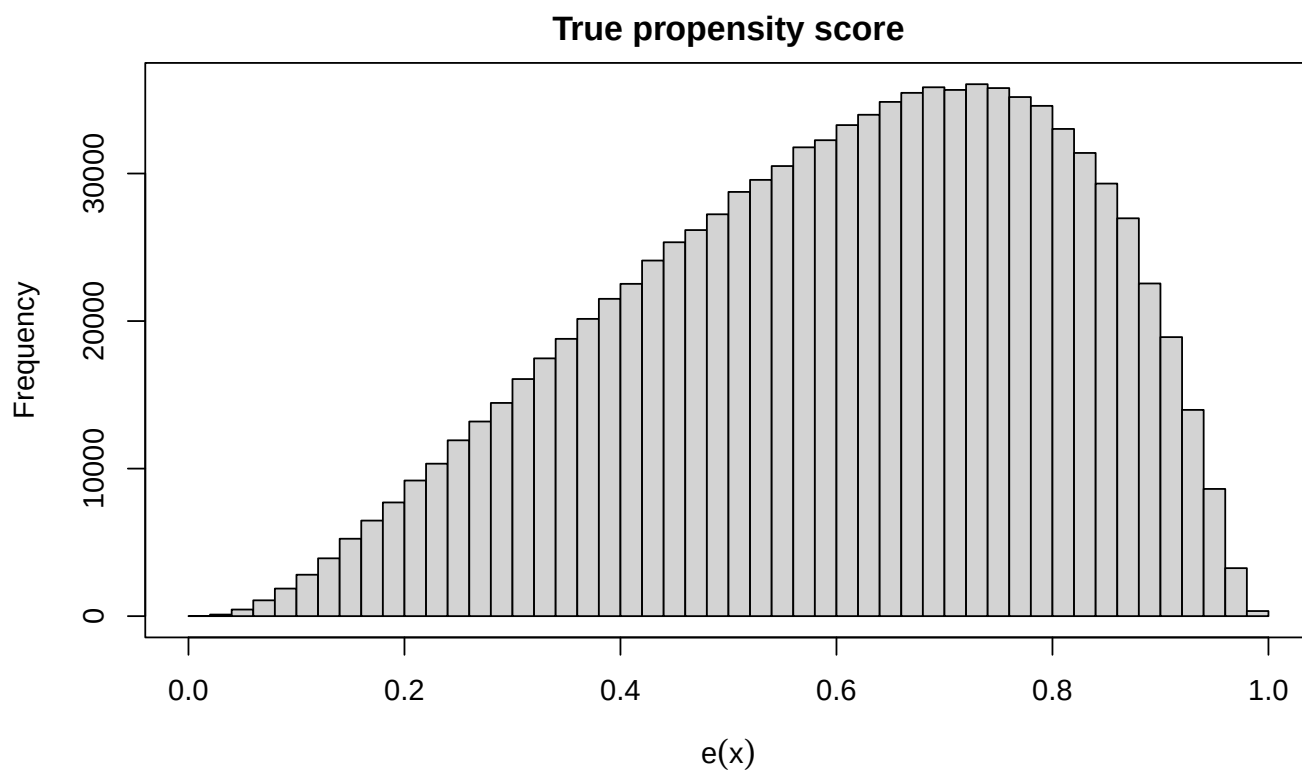
+           [,1]      [,2]      [,3]      [,4]
+ [1,]  4.815677 -4.595031  0.000000  0.000000
+ [2,] -4.595031  4.895861  0.000000  0.000000
+ [3,]  0.000000  0.000000  0.063339 -0.030614
+ [4,]  0.000000  0.000000 -0.030614  0.016556
```

```
+ [1] "Correlation matrix"
+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  1.000000000 -0.007857679 -0.8221476 -0.888949865  0.775134309
+ [2,] -0.007857679  1.000000000 -0.2632441  0.006592302 -0.006708655
+ [3,] -0.822147562 -0.263244147  1.0000000  0.788256335 -0.828744769
+ [4,] -0.888949865  0.006592302  0.7882563  1.000000000 -0.947049962
+ [5,]  0.775134309 -0.006708655 -0.8287448 -0.947049962  1.000000000
+ [1] "Correlation matrix"
+           [,1]      [,2]      [,3]      [,4]
+ [1,]  1.00000000 -0.470400015 -0.014883984  0.015367909
+ [2,] -0.47040002  1.000000000  0.004972201 -0.004385456
+ [3,] -0.01488398  0.004972201  1.000000000 -0.947049962
+ [4,]  0.01536791 -0.004385456 -0.947049962  1.000000000
+ [1] "Correlation matrix"
+           [,1]      [,2]      [,3]      [,4]
+ [1,]  1.00000000 -0.91039371  0.01366500 -0.01263616
+ [2,] -0.91039371  1.00000000 -0.02361449  0.02176495
+ [3,]  0.01366500 -0.02361449  1.00000000 -0.94704996
+ [4,] -0.01263616  0.02176495 -0.94704996  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 weight.

4. Second parameter setting

```
#Calculation for large N
library(mvnfast)
set.seed(22087)
N<-1000000
muX<-2;sigX<-0.5
X<-rnorm(N,muX,sigX)
al<-c(-3.5,2)
expit<-function(x){return(1/(1+exp(-x)))}
Xm<-cbind(1,X)
be<-c(2,3,4,1)
sigY<-3
eX<-ps.true<-expit(Xm %*% al)
Z<-rbinom(N,1,ps.true)
Xb<-cbind(1,Z,X,X^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+X+I(X^2))))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)				
Z				
X				
I(X^2)				

```

+ (Intercept) 2.047024 0.034235781 59.79193      0
+ Z           2.985374 0.006709775 444.92902      0
+ X           3.956903 0.034716756 113.97674      0
+ I(X^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:

```

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

+           [,1]
+ 2.569361
+ Z 2.986506
+ 19.408261

```

For the **G-estimating** form:

```

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

+           [,1]
+ 14.263723
+ Z 2.983436

```

For the asymptotic variances:

- For propensity score regression

```

#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,X) * 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,]  9.793841  6.085465  6.075510  0.000000 -0.000048
+ [2,]  6.085465  6.085465  4.213232 -0.029204 -0.068809
+ [3,]  6.075510  4.213232  4.207364 -0.000095  0.000020

```

```

+ [4,] 0.000000 -0.029204 -0.000095 0.198763 0.382645
+ [5,] -0.000048 -0.068809 0.000020 0.382645 0.771575
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(X*eterm*psr.ests[3]))
J.n[2,4:5]<-c(mean(Z*eterm*psr.ests[3]),mean(Z*X*eterm*psr.ests[3]))
J.n[3,4:5]<-c(mean(eterm*dterm),mean(X*eterm*dterm))
J.n[4:5,4:5]<-t(cbind(1,X) * matrix(eterm,nrow=N,ncol=2)) %*% cbind(1,X)/N

round(J.n,6)
+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,] 1.000000 0.603137 0.602450 3.859158 7.431566
+ [2,] 0.603137 0.603137 0.403992 2.223548 4.568998
+ [3,] 0.602450 0.403992 0.403609 2.249965 4.632571
+ [4,] 0.000000 0.000000 0.000000 0.198841 0.382907
+ [5,] 0.000000 0.000000 0.000000 0.382907 0.772386
V1<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V1,6)
+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,] 787.463508 -0.269193 -1134.4264 -268.640342 126.494431
+ [2,] -0.269193 47.015300 -46.5884 0.096372 -0.050509
+ [3,] -1134.426414 -46.588400 1876.7239 413.418267 -209.814651
+ [4,] -268.640342 0.096372 413.4183 110.704843 -54.861459
+ [5,] 126.494431 -0.050509 -209.8147 -54.861459 28.480751

```

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 47.0152999

- For G-estimation

```

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

round(I.n,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,] 25.115622 0.013918 0.000000 0.001051
+ [2,] 0.013918 4.037204 -0.134248 0.015650
+ [3,] 0.000000 -0.134248 0.198763 0.382645
+ [4,] 0.001051 0.015650 0.382645 0.771575
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(X*eX*(1-eX)*res1)
J.n[3,3]<-mean(eX*(1-eX))
J.n[3,4]<-J.n[4,3]<-mean(X*eX*(1-eX))
J.n[4,4]<-mean(X^2*eX*(1-eX))
round(J.n,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,]  1.000000  0.603137  0.000000  0.000000
+ [2,]  0.000687  0.199145 -0.133599  0.017956
+ [3,]  0.000000  0.000000  0.198841  0.382907
+ [4,]  0.000000  0.000000  0.382907  0.772386
V2<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
round(V2,6)
+           [,1]      [,2]      [,3]      [,4]
+ [1,]  41.728998 -27.486608 -0.134033  0.068143
+ [2,] -27.486608  45.476021  0.126386 -0.063212
+ [3,] -0.134033  0.126386 110.704843 -54.861459
+ [4,]  0.068143 -0.063212 -54.861459  28.480751

```

- **Inverse Probability Weighting:**

```

eX<-fitted(glm(Z~X,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<-X*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(X*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(X*R*Z*res2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X*res2^2)
I.n[4,4]<-mean(X^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(X*res2^2)
J.n[4,4]<-mean(X^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

```

```

#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){

  X<-rnorm(n,muX,sigX)
  Xm<-cbind(1,X)
  ps.true<-expit(Xm %*% al)

```

```

Z<-rbinom(n,1,ps.true)
Xb<-cbind(1,Z,X,X^2)
Y<-Xb %*% be + rnorm(n)*sigY

fit.p<-glm(Z~X,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,X) * 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(X*eterm*psr.fit[3]))
J.n[2,4:5]<-c(mean(Z*eterm*psr.fit[3]),mean(Z*X*eterm*psr.fit[3]))
J.n[3,4:5]<-c(mean(eterm*dterm),mean(X*eterm*dterm))
J.n[4:5,4:5]<-t(cbind(1,X) * matrix(eterm,nrow=n,ncol=2)) %*% cbind(1,X)/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(X*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(X*res2^2*res1)
I.n[3,3]<-mean(res2^2)

```

```

I.n[3,4]<-I.n[4,3]<-mean(X*res2^2)
I.n[4,4]<-mean(X^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(X*eX*(1-eX)*res1)
J.n[3,3]<-mean(eX*(1-eX))
J.n[3,4]<-J.n[4,3]<-mean(X*eX*(1-eX))
J.n[4,4]<-mean(X^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<-X*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(X*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(X*R*Z*res2*res1)
I.n[3,3]<-mean(res2^2)
I.n[3,4]<-I.n[4,3]<-mean(X*res2^2)
I.n[4,4]<-mean(X^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(X*res2^2)
J.n[4,4]<-mean(X^2*res2^2)

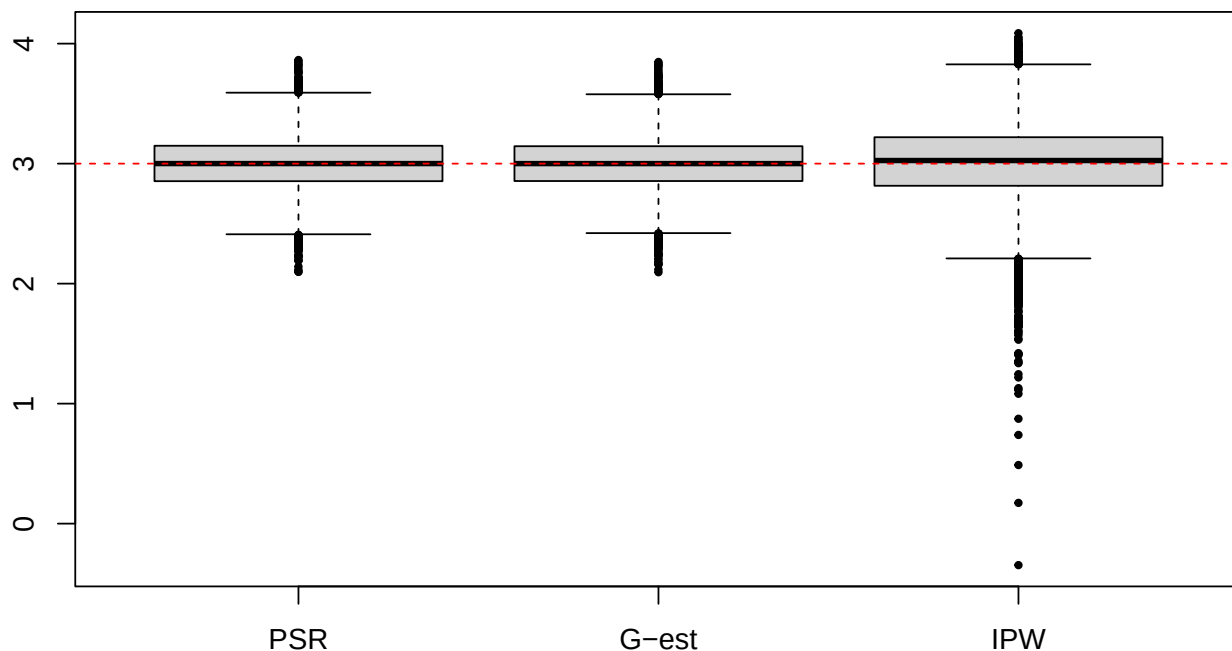
V<-solve(J.n) %*% (I.n %*% t(solve(J.n)))/n
V3.ests[irep,,]<-V

```

```

}

```



```
#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,]  0.787464 -0.000269 -1.134426 -0.268640  0.126494
```

```

+ [2,] -0.000269  0.047015 -0.046588  0.000096 -0.000051
+ [3,] -1.134426 -0.046588  1.876724  0.413418 -0.209815
+ [4,] -0.268640  0.000096  0.413418  0.110705 -0.054861
+ [5,]  0.126494 -0.000051 -0.209815 -0.054861  0.028481

#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,]  0.041729 -0.027487 -0.000134  0.000068
+ [2,] -0.027487  0.045476  0.000126 -0.000063
+ [3,] -0.000134  0.000126  0.110705 -0.054861
+ [4,]  0.000068 -0.000063 -0.054861  0.028481

#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

```

```

+ [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
+ [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
+ [1] "Correlation matrix"
+           [,1]           [,2]           [,3]           [,4]
+ [1,]  1.000000000 -0.79833532  0.01550379 -0.01539324
+ [2,] -0.79833532  1.000000000 -0.01312576  0.01257895
+ [3,]  0.01550379 -0.01312576  1.000000000 -0.97733471
+ [4,] -0.01539324  0.01257895 -0.97733471  1.000000000

```

Here the sandwich estimates match the empirical estimates well.