PROJECT 3: SOLUTIONS

The data generating conditional mean model for binary treatment is

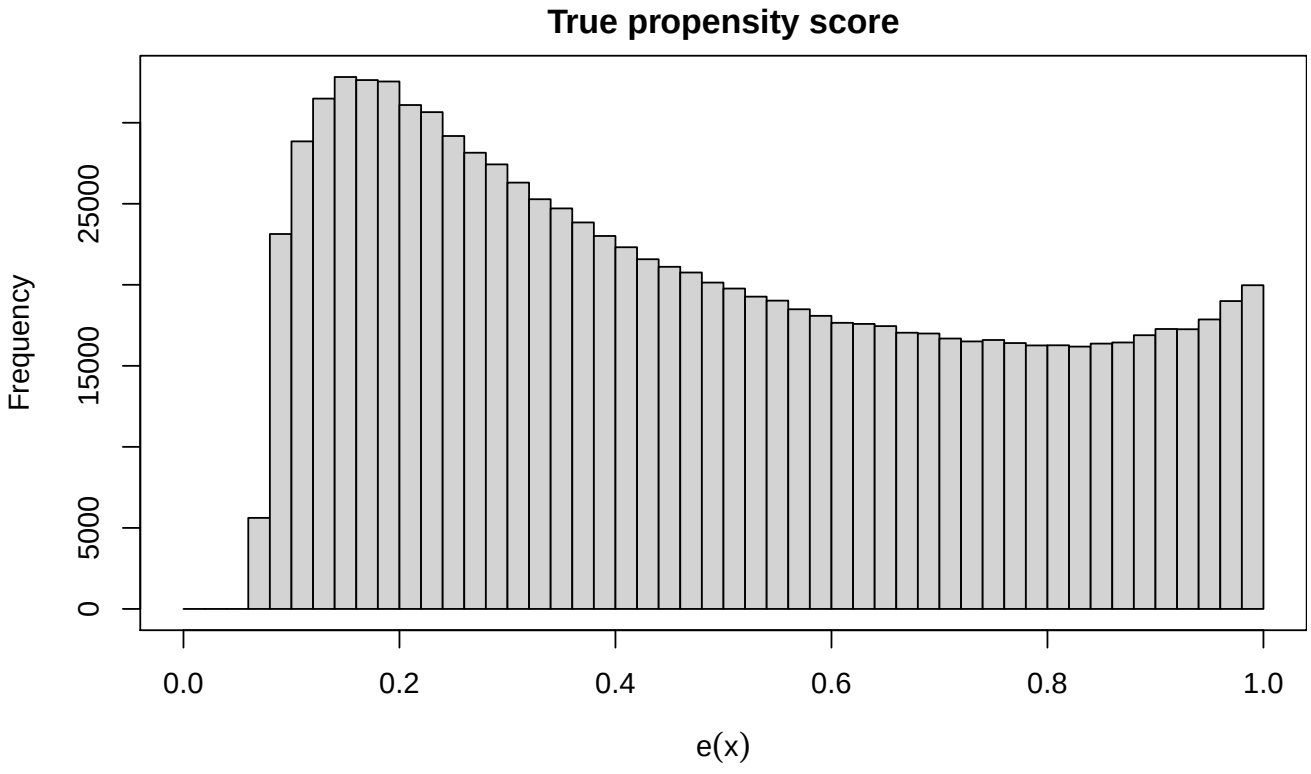
$$\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y|X=x,Z=z] = 1 + x + z + xz$$

with confounder $X \sim Normal(0.5, 0.5^2)$, and the conditional model for $Z|X=x$ is $Bernoulli(e(x))$ with

$$e(x) = \frac{\exp\{-1.5 + 2x + x^2\}}{1 + \exp\{-1.5 + 2x + x^2\}}.$$

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<-0.5;sigX<-0.5
X1<-rnorm(N,muX,sigX)
al<-c(-1.5,2,1)
expit<-function(x){return(1/(1+exp(-x)))}
Xa<-cbind(1,X1,X1^2)
be<-c(1,1,1,1)
sigY<-1
ps.true<-expit(Xa %*% al)
Z<-rbinom(N,1,ps.true)
Xb<-cbind(1,X1,Z,Z*X1)
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()
```



```
#Correct model
coef(summary(lm(Y~X1+Z+X1:Z)))

+              Estimate  Std. Error  t value Pr(>|t|)
+ (Intercept) 0.9998721 0.001619782 617.2882        0
+ X1          0.9941935 0.003375309 294.5489        0
+ Z           0.9997376 0.003324628 300.7066        0
+ X1:Z        1.0039912 0.004690596 214.0434        0

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

+             Estimate  Std. Error  t value Pr(>|t|)
+ (Intercept) 1.247148 0.001675588 744.3045        0
+ Z           2.298907 0.002417376 950.9927        0
```

G-estimation The **G-estimating** form is

$$\sum_{i=1}^{n} \begin{pmatrix} 1 \\ (z_i - e(x_i;\widehat{\alpha})) \\ (z_i - e(x_i;\widehat{\alpha}))x_i \end{pmatrix} (y_i - \beta_0 - z_i\psi_0 - z_i x_i \psi_1) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

that is, using the estimating (score) function

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

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

$$\mathbf{Z}_1 = \begin{pmatrix} 1 & z_1 - e(x_1;\widehat{\alpha}) & (z_1 - e(x_1;\widehat{\alpha}))x_1 \\ 1 & z_2 - e(x_2;\widehat{\alpha}) & (z_2 - e(x_2;\widehat{\alpha}))x_2 \\ \vdots & \vdots & \vdots \\ 1 & z_n - e(x_n;\widehat{\alpha}) & (z_n - e(x_n;\widehat{\alpha}))x_n \end{pmatrix} \qquad \mathbf{Z}_2 = \begin{pmatrix} 1 & z_1 & z_1x_1 \\ 1 & z_2 & z_2x_2 \\ \vdots & \vdots & \vdots \\ 1 & z_n & z_nx_n \end{pmatrix}$$

The solution is therefore

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

```
eX<-fitted(glm(Z~X1+I(X1^2),family=binomial))
Zm1<-cbind(1,Z-eX,(Z-eX)*X1)
Zm2<-cbind(1,Z,Z*X1)
g.ests<-solve(t(Zm1) %*% Zm2) %*% t(Zm1) %*% Y
g.ests

+        [,1]
+   1.4984902
+ Z 1.0023016
+   0.9994042
```

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 $(p \times p)$ matrices, and

$$\dot{\mathbf{U}}(\theta_0) = -\frac{\partial \mathbf{U}(\theta)}{\partial \theta^\top}\bigg|_{\theta=\theta_0}$$

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.

(a) ***Known propensity score:*** If

$$\epsilon_Y = (Y - \beta_0 - Z\psi_0 - ZX\psi_1) \qquad \epsilon_Z = (Z - e(X;\alpha))$$

then

$$\mathbf{U}(\theta) = \begin{pmatrix} (Y - \beta_0 - Z\psi_0 - ZX\psi_1) \\ (Z - e(X;\alpha))(Y - \beta_0 - Z\psi_0 - ZX\psi_1) \\ (Z - e(X;\alpha))X(Y - \beta_0 - Z\psi_0 - ZX\psi_1) \end{pmatrix} = \begin{pmatrix} \epsilon_Y \\ \epsilon_Z\epsilon_Y \\ \epsilon_Z X\epsilon_Y \end{pmatrix}$$

and the derivative is

$$\dot{\mathbf{U}}(\theta) = \begin{bmatrix} 1 & Z & ZX \\ \epsilon_Z & Z\epsilon_Z & ZX\epsilon_Z \\ X\epsilon_Z & ZX\epsilon_Z & ZX^2\epsilon_Z \end{bmatrix}$$

and the $\mathcal{I}$ and $\mathcal{J}$ matrices are 3×3.

(b) ***Unknown propensity score:*** If $\mathbf{X} = (1, X, X^2)$, the score vector is

$$\mathbf{U}(\theta) = \begin{pmatrix} (Y - \beta_0 - Z\psi_0 - ZX\psi_1) \\ (Z - e(X;\alpha))(Y - \beta_0 - Z\psi_0 - ZX\psi_1) \\ (Z - e(X;\alpha))X(Y - \beta_0 - Z\psi_0 - ZX\psi_1) \\ \mathbf{X}^\top(Z - e(X;\alpha)) \end{pmatrix} = \begin{pmatrix} \epsilon_Y \\ \epsilon_Z\epsilon_Y \\ \epsilon_Z X\epsilon_Y \\ \mathbf{X}^\top\epsilon_Z \end{pmatrix}$$

and writing $e \equiv e(X;\alpha)$, the derivative is

$$\dot{\mathbf{U}}(\theta) = \begin{bmatrix} 1 & Z & ZX & \mathbf{0} \\ \epsilon_Z & Z\epsilon_Z & ZX\epsilon_Z & e(1-e)\mathbf{X}\epsilon_Y \\ X\epsilon_Z & ZX\epsilon_Z & ZX^2\epsilon_Z & e(1-e)X\mathbf{X}\epsilon_Y \\ 0 & 0 & 0 & e(1-e)\mathbf{X}^\top\mathbf{X} \end{bmatrix}$$

and the $\mathcal{I}$ and $\mathcal{J}$ matrices are 6×6.

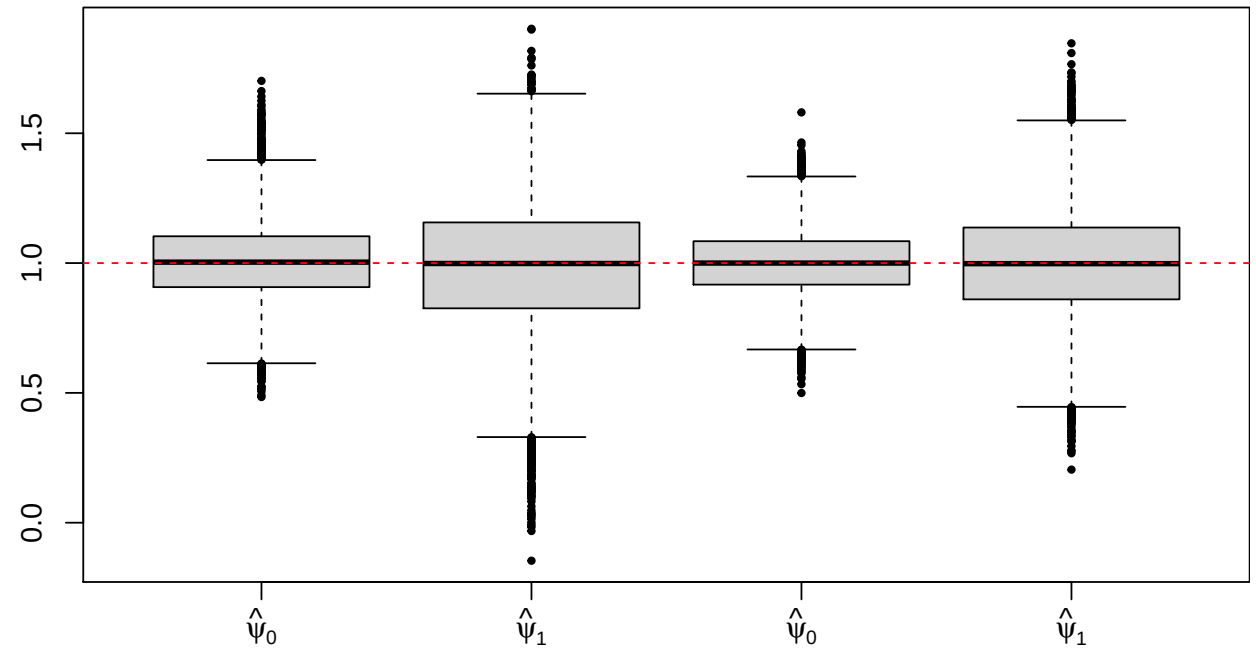
```r
#Large sample calculation

    eX<-ps.true
    Zm1<-cbind(1,Z-eX,X1*(Z-eX))
    Zm2<-cbind(1,Z,Z*X1)
    g.ests0<-solve(t(Zm1) %*% Zm2) %*% t(Zm1) %*% Y

    #G-estimation variance estimate
    res1<-Y-g.ests0[1]-g.ests0[2]*Z - g.ests0[3]*Z*X1
    res2<-Z-eX

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

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

    V1<-solve(J.n) %*% (I.n %*% t(solve(J.n)))

#####################################################################

    fit.p<-glm(Z~X1+I(X1^2),family=binomial)
    eX<-fitted(fit.p)

    Zm1<-cbind(1,Z-eX,X1*(Z-eX))
    Zm2<-cbind(1,Z,Z*X1)
    g.ests1<-solve(t(Zm1) %*% Zm2) %*% t(Zm1) %*% Y

    #G-estimation variance estimate
    res1<-Y-g.ests1[1]-g.ests1[2]*Z - g.ests1[3]*Z*X1
    res2<-Z-eX

    I.n<-matrix(0,6,6)
    I.n[1,1]<-mean(res1^2)
    I.n[1,2]<-I.n[2,1]<-mean(res1^2*res2)
    I.n[1,3]<-I.n[3,1]<-mean(res1^2*res2*X1)
    I.n[1,4]<-I.n[4,1]<-mean(res1*res2)
    I.n[1,5]<-I.n[5,1]<-mean(res1*res2*X1)
    I.n[1,6]<-I.n[6,1]<-mean(res1*res2*X1^2)
    I.n[2,2]<-mean(res1^2*res2^2)
    I.n[2,3]<-I.n[3,2]<-mean(res1^2*res2^2*X1)
    I.n[2,4]<-I.n[4,2]<-mean(res1*res2^2)
    I.n[2,5]<-I.n[5,2]<-mean(res1*res2^2*X1)
    I.n[2,6]<-I.n[6,2]<-mean(res1*res2^2*X1^2)

    I.n[3,3]<-mean(res1^2*res2^2*X1^2)
    I.n[3,4]<-I.n[4,3]<-mean(res1*res2^2*X1)
    I.n[3,5]<-I.n[5,3]<-mean(res1*res2^2*X1^2)
```

```
    I.n[3,6]<-I.n[6,3]<-mean(res1*res2^2*X1^3)

    I.n[4,4]<-mean(res2^2)
    I.n[4,5]<-I.n[5,4]<-mean(X1*res2^2)
    I.n[4,6]<-I.n[6,4]<-mean(X1^2*res2^2)
    I.n[5,5]<-mean(X1^2*res2^2)
    I.n[5,6]<-I.n[6,5]<-mean(X1^3*res2^2)
    I.n[6,6]<-I.n[6,6]<-mean(X1^4*res2^2)

    J.n<-matrix(0,6,6)
    vX<-eX*(1-eX)
    J.n[1,1]<-1
    J.n[1,2]<-mean(Z)
    J.n[1,3]<-mean(Z*X1)
    J.n[2,1]<-mean(res2)
    J.n[2,2]<-mean(Z*res2)
    J.n[2,3]<-mean(Z*X1*res2)
    J.n[2,4]<-mean(vX*res1)
    J.n[2,5]<-mean(vX*res1*X1)
    J.n[2,6]<-mean(vX*res1*X1^2)
    J.n[3,1]<-mean(X1*res2)
    J.n[3,2]<-mean(Z*X1*res2)
    J.n[3,3]<-mean(Z*X1^2*res2)
    J.n[3,4]<-mean(vX*X1*res1)
    J.n[3,5]<-mean(vX*res1*X1^2)
    J.n[3,6]<-mean(vX*res1*X1^3)
    J.n[4,4]<-mean(vX)
    J.n[4,5]<-J.n[5,4]<-mean(X1*vX)
    J.n[4,6]<-J.n[6,4]<-mean(X1^2*vX)
    J.n[5,5]<-mean(X1^2*vX)
    J.n[5,6]<-J.n[6,5]<-mean(X1^3*vX)
    J.n[6,6]<-J.n[6,6]<-mean(X1^4*vX)

    V2<-solve(J.n) %*% (I.n %*% t(solve(J.n)))
```

```
#Monte Carlo study
nreps<-20000
n<-1000
psr.ests<-matrix(0,nrow=nreps,ncol=3)
g.ests0<-g.ests1<-al.ests<-matrix(0,nrow=nreps,ncol=3)
V1.ests<-array(0,c(nreps,3,3))
V2.ests<-array(0,c(nreps,6,6))
for(irep in 1:nreps){

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

    eX<-ps.true
    Zm1<-cbind(1,Z-eX,X1*(Z-eX))
    Zm2<-cbind(1,Z,Z*X1)
    g.ests0[irep,]<-solve(t(Zm1) %*% Zm2) %*% t(Zm1) %*% Y

    #G-estimation variance estimate
    res1<-Y-g.ests0[irep,1]-g.ests0[irep,2]*Z - g.ests0[irep,3]*Z*X1
    res2<-Z-eX

    I.n<-matrix(0,3,3)
```

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

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

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

################################################################

    fit.p<-glm(Z~X1+I(X1^2),family=binomial)
    eX<-fitted(fit.p)
    al.ests[irep,]<-coef(fit.p)

    Zm1<-cbind(1,Z-eX,X1*(Z-eX))
    Zm2<-cbind(1,Z,Z*X1)
    g.ests1[irep,]<-solve(t(Zm1) %*% Zm2) %*% t(Zm1) %*% Y

    #G-estimation variance estimate
    res1<-Y-g.ests1[irep,1]-g.ests1[irep,2]*Z - g.ests1[irep,3]*Z*X1
    res2<-Z-eX

    I.n<-matrix(0,6,6)
    I.n[1,1]<-mean(res1^2)
    I.n[1,2]<-I.n[2,1]<-mean(res1^2*res2)
    I.n[1,3]<-I.n[3,1]<-mean(res1^2*res2*X1)
    I.n[1,4]<-I.n[4,1]<-mean(res1*res2)
    I.n[1,5]<-I.n[5,1]<-mean(res1*res2*X1)
    I.n[1,6]<-I.n[6,1]<-mean(res1*res2*X1^2)
    I.n[2,2]<-mean(res1^2*res2^2)
    I.n[2,3]<-I.n[3,2]<-mean(res1^2*res2^2*X1)
    I.n[2,4]<-I.n[4,2]<-mean(res1*res2^2)
    I.n[2,5]<-I.n[5,2]<-mean(res1*res2^2*X1)
    I.n[2,6]<-I.n[6,2]<-mean(res1*res2^2*X1^2)

    I.n[3,3]<-mean(res1^2*res2^2*X1^2)
    I.n[3,4]<-I.n[4,3]<-mean(res1*res2^2*X1)
    I.n[3,5]<-I.n[5,3]<-mean(res1*res2^2*X1^2)
    I.n[3,6]<-I.n[6,3]<-mean(res1*res2^2*X1^3)

    I.n[4,4]<-mean(res2^2)
    I.n[4,5]<-I.n[5,4]<-mean(X1*res2^2)
    I.n[4,6]<-I.n[6,4]<-mean(X1^2*res2^2)
    I.n[5,5]<-mean(X1^2*res2^2)
    I.n[5,6]<-I.n[6,5]<-mean(X1^3*res2^2)
    I.n[6,6]<-I.n[6,6]<-mean(X1^4*res2^2)

    J.n<-matrix(0,6,6)
```

```
    vX<-eX*(1-eX)
    J.n[1,1]<-1
    J.n[1,2]<-mean(Z)
    J.n[1,3]<-mean(Z*X1)
    J.n[2,1]<-mean(res2)
    J.n[2,2]<-mean(Z*res2)
    J.n[2,3]<-mean(Z*X1*res2)
    J.n[2,4]<-mean(vX*res1)
    J.n[2,5]<-mean(vX*res1*X1)
    J.n[2,6]<-mean(vX*res1*X1^2)
    J.n[3,1]<-mean(X1*res2)
    J.n[3,2]<-mean(Z*X1*res2)
    J.n[3,3]<-mean(Z*X1^2*res2)
    J.n[3,4]<-mean(vX*X1*res1)
    J.n[3,5]<-mean(vX*res1*X1^2)
    J.n[3,6]<-mean(vX*res1*X1^3)
    J.n[4,4]<-mean(vX)
    J.n[4,5]<-J.n[5,4]<-mean(X1*vX)
    J.n[4,6]<-J.n[6,4]<-mean(X1^2*vX)
    J.n[5,5]<-mean(X1^2*vX)
    J.n[5,6]<-J.n[6,5]<-mean(X1^3*vX)
    J.n[6,6]<-J.n[6,6]<-mean(X1^4*vX)

    V<-solve(J.n) %*% (I.n %*% t(solve(J.n)))/n
    V2.ests[irep,,]<-V

}
```




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

+     Psi0-0     Psi1-0     Psi0-1     Psi1-1
+ 0.02190342 0.06098783 0.01514903 0.04031111
```

```
round(apply(V1.ests,2:3,mean),6)[2:3,2:3]        #Monte Carlo sandwich estimate

+           [,1]      [,2]
+ [1,]  0.021604 -0.030333
+ [2,] -0.030333  0.060778

round(cov(cbind(g.ests0,al.ests)),6)[2:3,2:3]    #Empirical estimate

+           [,1]      [,2]
+ [1,]  0.021903 -0.030520
+ [2,] -0.030520  0.060988

round(V1/n,6)[2:3,2:3]                           #Monte Carlo sandwich estimate (large sample)

+           [,1]      [,2]
+ [1,]  0.021101 -0.029573
+ [2,] -0.029573  0.059475

round(apply(V2.ests,2:3,mean),6)[2:3,2:3]        #Monte Carlo sandwich estimate

+           [,1]      [,2]
+ [1,]  0.014886 -0.019239
+ [2,] -0.019239  0.040096

round(cov(cbind(g.ests1,al.ests)),6)[2:3,2:3]    #Empirical estimate

+           [,1]      [,2]
+ [1,]  0.015149 -0.019396
+ [2,] -0.019396  0.040311

round(V2/n,6)[2:3,2:3]                           #Monte Carlo sandwich estimate (large sample)

+           [,1]      [,2]
+ [1,]  0.014858 -0.019256
+ [2,] -0.019256  0.040204
```