

# MATH 598/782 – PROJECT 3 SOLUTIONS

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.

Based on the data set from Project 2

<https://www.math.mcgill.ca/dstephens/598-Causal-2024/Exercises/Project2/P2data.csv>

which comprises four predictor variables  $X_1, X_2, X_3, X_4$ , a binary treatment variable  $Z$ , and outcome variable  $Y$ , compute an estimate of the asymptotic variance of the parameter of interest  $\psi$  based on the sandwich estimation procedure and knitr sheets for the following models and estimating procedures.

(a) An outcome regression model

$$Y_i = \beta_0 + \beta_1 X_{1i} + \psi Z_i + \epsilon_i$$

estimated using ordinary least squares.

**Solution:** In this case, we can use the standard results for OLS. There are no additional parameters to estimate, and we have that

$$\mathbf{U}(\theta) = \mathbf{x}^\top (Y - \mathbf{x}\theta) \quad \dot{\mathbf{U}}(\theta_0) = \mathbf{x}^\top \mathbf{x}$$

where  $\theta = (\beta_0, \beta_1, \psi)^\top$  and  $\mathbf{x} = (1, X_1, Z)$ . Therefore the asymptotic variance is

$$V = \{\mathbb{E}[\mathbf{x}^\top \mathbf{x}]\}^{-1} \mathbb{E}[(Y - \mathbf{x}\theta_0)^2 \mathbf{x}^\top \mathbf{x}] \{\mathbb{E}[\mathbf{x}^\top \mathbf{x}]\}^{-1}$$

that we can estimate using the quantities

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}_i \hat{\theta}_n)^2 \mathbf{x}_i^\top \mathbf{x}_i \quad \hat{J}_n = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i$$

before extracting the component for  $\psi$ .

```
fn<-"https://www.math.mcgill.ca/dstephens/598-Causal-2024/Exercises/Project2/P2data.csv"
P3.data<-read.csv(fn)
n<-nrow(P3.data)
Xm<-cbind(1,P3.data$X1,P3.data$Z)
Y<-P3.data$Y
Z<-P3.data$Z
X1<-P3.data$X1
X2<-P3.data$X2

theta.hat<-solve(t(Xm) %*% Xm) %*% (t(Xm) %*% Y)
theta.hat
```

```

+           [,1]
+ [1,] -3.6864408
+ [2,] -0.2867274
+ [3,]  4.8976581

All.ests<-All.se<-rep(0,6)
All.ests[1]<-theta.hat[3]
eps.y<-Y - Xm %*% theta.hat

I.n<-crossprod(Xm*as.vector(eps.y))/n; round(I.n,5)

+           [,1]      [,2]      [,3]
+ [1,] 15.05223 20.22899  6.05350
+ [2,] 20.22899 38.73610 10.17311
+ [3,]  6.05350 10.17311  6.05350

J.n<-crossprod(Xm)/n; round(J.n,5)

+           [,1]      [,2]      [,3]
+ [1,] 1.00000 0.99122 0.32667
+ [2,] 0.99122 1.90635 0.48323
+ [3,] 0.32667 0.48323 0.32667

#Asymptotic variance
V.n<-solve(J.n) %*% (I.n %*% solve(J.n));round(V.n,5)

+           [,1]      [,2]      [,3]
+ [1,] 21.10129 -7.66351 -13.24632
+ [2,] -7.66351 18.11392 -14.51732
+ [3,] -13.24632 -14.51732 88.10460

#Approximate estimate variance
round(V.n/n,5)

+           [,1]      [,2]      [,3]
+ [1,] 0.07034 -0.02555 -0.04415
+ [2,] -0.02555 0.06038 -0.04839
+ [3,] -0.04415 -0.04839 0.29368

se.vec<-sqrt(diag(V.n)/n); round(se.vec,5)

+ [1] 0.26521 0.24572 0.54192

All.se[1]<-se.vec[3]

```

Therefore the approximate standard errors for the three estimates are the square roots of the diagonal of this matrix divided by  $n$ , that is

|                 | Est.    | s.e.   |
|-----------------|---------|--------|
| $\hat{\beta}_0$ | -3.6864 | 0.2652 |
| $\hat{\beta}_1$ | -0.2867 | 0.2457 |
| $\hat{\psi}$    | 4.8977  | 0.5419 |

4 Marks

For this problem, we can compare with the ordinary `lm` results (that assume correct specification, and estimates the homoscedastic error variance using the standard approach)

```
fit.a<-lm(Y~X1+Z,data=P3.data)
round(summary(fit.a)$coef,4)

+           Estimate Std. Error  t value Pr(>|t|)
+ (Intercept) -3.6864      0.3331 -11.0669  0.0000
+ X1          -0.2867      0.2504  -1.1451  0.2531
+ Z           4.8977      0.5132   9.5437  0.0000

se.standard<-summary(fit.a)$coef[,2]
```

These standard errors are similar to, but slightly different from, the sandwich versions.

|                 | Sandwich | lm     |
|-----------------|----------|--------|
| $\hat{\beta}_0$ | 0.2652   | 0.3331 |
| $\hat{\beta}_1$ | 0.2457   | 0.2504 |
| $\hat{\psi}$    | 0.5419   | 0.5132 |

We can also use the `sandwich` package to obtain the sandwich variance estimates.

```
library(sandwich)
sandwich(fit.a)

+           (Intercept)           X1           Z
+ (Intercept)  0.07033765 -0.02554503 -0.04415439
+ X1          -0.02554503  0.06037974 -0.04839107
+ Z           -0.04415439 -0.04839107  0.29368200
```

This matrix is the  $\hat{V}_n/n$  matrix, and matches the above calculation.

(b) A propensity score regression model

$$Y_i = \beta_0 + \psi Z_i + \phi e(X_{1i}, X_{2i}; \hat{\alpha}) + \epsilon_i$$

where  $\alpha$  is estimated using maximum likelihood in a logistic regression of  $Z$  on  $(X_1, X_2)$ , including an intercept, that is  $Z|X_1 = x_1, X_2 = x_2 \sim \text{Bernoulli}(e(x_1, x_2; \alpha))$  with  $\alpha = (\alpha_0, \alpha_1, \alpha_2)^\top$ , and

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

**Solution:** For this propensity score regression model with estimated propensity score, we have

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

where  $\mathbf{x} = (1, x_1, x_2)$ . We have therefore that

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

This follows in the third line as by the chain rule

$$\begin{aligned}\frac{\partial}{\partial \alpha} \{e(X; \alpha)(Y - \beta_0 - \psi Z - \phi e(X; \alpha))\} &= -\phi e(X; \alpha) \frac{\partial e(X; \alpha)}{\partial \alpha} + \frac{\partial e(X; \alpha)}{\partial \alpha} (Y - \beta_0 - \psi Z - \phi e(X; \alpha)) \\ &= \frac{\partial e(X; \alpha)}{\partial \alpha} (-\phi e(X; \alpha) + \varepsilon_Y) \\ &= e(X; \alpha)(1 - e(X; \alpha)) \mathbf{x}^\top (\varepsilon_Y - \phi e(X; \alpha)).\end{aligned}$$

Note that

$$\frac{\partial e(X; \alpha)}{\partial \alpha} = e(X; \alpha)(1 - e(X; \alpha)) \mathbf{x}^\top$$

as under the logistic regression propensity model

$$\frac{\partial e(x)}{\partial x} = \frac{\partial}{\partial x} \left\{ \frac{e^x}{1 + e^x} \right\} = \frac{e^x}{(1 + e^x)^2} = \frac{e^x}{1 + e^x} \frac{1}{1 + e^x} = \frac{e^x}{1 + e^x} \left( 1 - \frac{e^x}{1 + e^x} \right) = e(x)(1 - e(x))$$

```
eX<-fitted(glm(Z~X1+X2,family=binomial,data=P3.data))
vx<-eX*(1-eX)
Ptmp<-P3.data
Ptmp$eX<-eX

fit.psr<-lm(Y~Z+eX,data=Ptmp)
psr.ests<-coef(fit.psr)

All.ests[2]<-psr.ests[2]

Xa<-cbind(1,X1,X2)

eps.y<-residuals(fit.psr)
eps.z<-Z-eX

I.n<-crossprod(cbind(eps.y,Z*eps.y,eX*eps.y,Xa*eps.z))/n; round(I.n,5);I.n.psr<-I.n

+          eps.y          X1          X2
+ eps.y 10.03324  3.46494  3.61134  0.00000  0.02143 -0.06402
+       3.46494  3.46494  1.72244 -0.01045 -0.30421  0.36789
+       3.61134  1.72244  1.82563 -0.00142  0.02458 -0.01863
+       0.00000 -0.01045 -0.00142  0.17115  0.20092 -0.31567
+ X1      0.02143 -0.30421  0.02458  0.20092  0.38387 -0.37840
+ X2     -0.06402  0.36789 -0.01863 -0.31567 -0.37840  0.71854

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

Xp<-cbind(1,Z,eX)
J.n[1:3,1:3]<-crossprod(Xp)/n
J.n[4:6,4:6]<-crossprod(Xa*sqrt(vx))/n
J.n[1,4]<-mean(psr.ests[3]*vx)
J.n[1,5]<-mean(psr.ests[3]*X1*vx)
J.n[1,6]<-mean(psr.ests[3]*X2*vx)
J.n[2,4]<-mean(psr.ests[3]*Z*vx)
J.n[2,5]<-mean(psr.ests[3]*Z*X1*vx)
J.n[2,6]<-mean(psr.ests[3]*Z*X2*vx)
J.n[3,4]<-mean((psr.ests[3]*eX-eps.y)*vx)
J.n[3,5]<-mean((psr.ests[3]*eX-eps.y)*X1*vx)
J.n[3,6]<-mean((psr.ests[3]*eX-eps.y)*X2*vx)
J.n.psr<-J.n
```

```

round(J.n,5)

+      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
+ [1,] 1.00000 0.32667 0.32667 2.06994 2.47607 -3.83245
+ [2,] 0.32667 0.32667 0.15417 0.77945 1.20078 -1.23486
+ [3,] 0.32667 0.15417 0.15282 0.78810 1.46592 -1.64081
+ [4,] 0.00000 0.00000 0.00000 0.17384 0.20795 -0.32187
+ [5,] 0.00000 0.00000 0.00000 0.20795 0.36619 -0.40219
+ [6,] 0.00000 0.00000 0.00000 -0.32187 -0.40219 0.74049

#Asymptotic variance
V.n<-solve(J.n) %*% (I.n %*% t(solve(J.n))); round(V.n,5); V.n.psr<-V.n

+      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
+ [1,] 183.42166 -0.66000 -484.32210 -58.19956 38.05719 -2.58582
+ [2,] -0.66000 8.47447 -4.00667 1.28510 -0.56701 0.46675
+ [3,] -484.32210 -4.00667 1571.35795 136.42847 -114.41011 5.96867
+ [4,] -58.19956 1.28510 136.42847 37.36367 -10.11968 10.43912
+ [5,] 38.05719 -0.56701 -114.41011 -10.11968 11.12457 1.82030
+ [6,] -2.58582 0.46675 5.96867 10.43912 1.82030 6.80000

#Approximate estimate variance
round(V.n/n,5)

+      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
+ [1,] 0.61141 -0.00220 -1.61441 -0.19400 0.12686 -0.00862
+ [2,] -0.00220 0.02825 -0.01336 0.00428 -0.00189 0.00156
+ [3,] -1.61441 -0.01336 5.23786 0.45476 -0.38137 0.01990
+ [4,] -0.19400 0.00428 0.45476 0.12455 -0.03373 0.03480
+ [5,] 0.12686 -0.00189 -0.38137 -0.03373 0.03708 0.00607
+ [6,] -0.00862 0.00156 0.01990 0.03480 0.00607 0.02267

se.vec<-sqrt(diag(V.n)/n); se.vec

+ [1] 0.7819243 0.1680721 2.2886371 0.3529101 0.1925666 0.1505545

All.se[2]<-se.vec[2]

```

Thus the  $\psi$  parameter has estimate 2.12075 with standard error 0.16807. We can use the sandwich package to compute the sandwich variance in the outcome model that assumes that the plug-in propensity model is not in fact estimated.

```

library(sandwich)
sandwich::sandwich(fit.psr)

+      (Intercept)      Z      eX
+ (Intercept) 0.107646403 0.008777921 -0.2601478
+ Z          0.008777921 0.210553896 -0.2467374
+ eX         -0.260147802 -0.246737447 1.1265638

```

Therefore this estimated standard error is  $\sqrt{0.21055}$ , that is 0.45886. However, this variance does not accurately reflect the sampling variability.

6 Marks

(c) An inverse probability weighting (IPW) procedure using the propensity score model from (b).

**Solution:** For this, we can simply deploy the code from knitr sheet 12, and either of the IPW estimators. In this case, we have to estimate the parameters using the logistic regression model.

- **IPW version 1:** For the elementary application of IPW, there is a single unknown parameter of interest, and the estimator is

$$\hat{\psi} = \frac{1}{n} \sum_{i=1}^n \left( \frac{Z_i}{e(X)} - \frac{1 - Z_i}{1 - e(X_i)} \right) Y_i$$

For the unknown propensity score case, the score is

$$\mathbf{U}(\theta) = \begin{pmatrix} \left( \frac{Z}{e(X; \alpha)} - \frac{1 - Z}{1 - e(X; \alpha)} \right) Y - \psi \\ \mathbf{x}^\top (Z - e(X; \alpha)) \end{pmatrix} = \begin{pmatrix} \zeta_Y \\ \mathbf{x}^\top \epsilon_Z \end{pmatrix}.$$

say. Therefore

$$\mathcal{I} = \mathbb{E} \begin{bmatrix} \zeta_Y^2 & \mathbf{x}^\top \epsilon_Z \zeta_Y \\ \mathbf{x}^\top \epsilon_Z \zeta_Y & \mathbf{x}^\top \mathbf{x} \epsilon_Z^2 \end{bmatrix} \quad \mathcal{J} = \mathbb{E} \begin{bmatrix} 1 & \left( \frac{Z}{e^2} + \frac{1 - Z}{(1 - e)^2} \right) Y \mathbf{x} e(1 - e) \\ \mathbf{0} & \mathbf{x}^\top \mathbf{x} e(1 - e) \end{bmatrix}$$

as

$$\frac{\partial}{\partial \alpha} \left\{ \left( \frac{Z}{e(X; \alpha)} - \frac{1 - Z}{1 - e(X; \alpha)} \right) Y - \psi \right\} = \left( -\frac{Z}{e(X; \alpha)^2} - \frac{1 - Z}{(1 - e(X; \alpha))^2} \right) Y \frac{\partial e(X; \alpha)}{\partial \alpha}$$

with the final derivative as given above.

```
eX<-fitted(glm(Z~X1+X2,family=binomial,data=P3.data))
wv<-Z/eX-(1-Z)/(1-eX)
psi.hat.ipw1<-mean(wv*Y)
psi.hat.ipw1

+ [1] 1.946609

All.ests[3]<-psi.hat.ipw1

Xa<-cbind(1,X1,X2)
eps.y<-wv*Y-psi.hat.ipw1
eps.z<-Z-eX

I.n<-crossprod(cbind(eps.y,Xa*eps.z))/n; round(I.n,5)

+           eps.y           X1           X2
+ eps.y 165.62082 -1.72138 -0.89623  6.91117
+       -1.72138  0.17115  0.20092 -0.31567
+ X1      -0.89623  0.20092  0.38387 -0.37840
+ X2       6.91117 -0.31567 -0.37840  0.71854

vx<-eX*(1-eX)
J.n<-matrix(0,4,4)
J.n[1,1]<-1
J.n[1,2]<-mean(vx*Y*(Z/eX^2+(1-Z)/(1-eX)^2))
J.n[1,3]<-mean(X1*vx*Y*(Z/eX^2+(1-Z)/(1-eX)^2))
J.n[1,4]<-mean(X2*vx*Y*(Z/eX^2+(1-Z)/(1-eX)^2))
J.n[2:4,2:4]<-crossprod(Xa*sqrt(vx))/n
round(J.n,5)

+           [,1]      [,2]      [,3]      [,4]
+ [1,]      1 -1.72138 -0.89623  6.91117
+ [2,]      0  0.17384  0.20795 -0.32187
+ [3,]      0  0.20795  0.36619 -0.40219
+ [4,]      0 -0.32187 -0.40219  0.74049
```

```

#Asymptotic variance
V.n<-solve(J.n) %*% (I.n %*% t(solve(J.n))); round(V.n,5)

+           [,1]      [,2]      [,3]      [,4]
+ [1,] 37.29145    7.31701   -6.12121   0.01876
+ [2,]  7.31701   37.36367  -10.11968  10.43912
+ [3,] -6.12121  -10.11968   11.12457   1.82030
+ [4,]  0.01876   10.43912   1.82030   6.80000

#Approximate estimate variance
round(V.n/n,5)

+           [,1]      [,2]      [,3]      [,4]
+ [1,]  0.12430    0.02439  -0.02040   0.00006
+ [2,]  0.02439    0.12455  -0.03373   0.03480
+ [3,] -0.02040  -0.03373   0.03708   0.00607
+ [4,]  0.00006    0.03480   0.00607   0.02267

se.vec<-sqrt(diag(V.n)/n); se.vec

+ [1] 0.3525689 0.3529101 0.1925666 0.1505545

All.se[3]<-se.vec[1]

```

Therefore the estimate is 1.94661 with standard error 0.35257.

- **IPW version 2:** For the second version of the IPW estimator, we consider weighted least squares and the objective function

$$\sum_{i=1}^n R_i (Y_i - \beta_0 - Z_i \psi)^2$$

where

$$R_i = \left( \frac{1 - Z_i}{1 - e(X_i)} + \frac{Z_i}{e(X_i)} \right).$$

This yields the estimator

$$\hat{\psi} = \frac{\sum_{i=1}^n \frac{Z_i Y_i}{e(X_i)}}{\sum_{i=1}^n \frac{Z_i}{e(X_i)}} - \frac{\sum_{i=1}^n \frac{(1 - Z_i) Y_i}{1 - e(X_i)}}{\sum_{i=1}^n \frac{(1 - Z_i)}{1 - e(X_i)}}.$$

For the unknown propensity score case, the score vector is

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

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

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

```

w0<-(1-Z)/(1-eX)
w1<-Z/eX
w<-w0+w1
R<-w

fit.ipw<-lm(Y~Z,weights=w)
summary(fit.ipw)$coef

+           Estimate Std. Error   t value    Pr(>|t|)
+ (Intercept) -2.989732  0.3331726 -8.973523 3.293313e-17
+ Z           2.000880  0.4629684  4.321850 2.111298e-05

ipw.ests<-coef(fit.ipw)
All.ests[4]<-ipw.ests[2]

eps.y<-Y-ipw.ests[1]-ipw.ests[2]*Z
eps.z<-Z-eX

I.n<-crossprod(cbind(R*eps.y,R*Z*eps.y,Xa*eps.z))/n; round(I.n,5);I.n.ipw2<-I.n

+           X1      X2
+ 136.14705 80.86551 -1.19491 -1.40032 2.81950
+ 80.86551 80.86551 -0.58013 -0.19082 3.23438
+ -1.19491 -0.58013 0.17115 0.20092 -0.31567
+ X1 -1.40032 -0.19082 0.20092 0.38387 -0.37840
+ X2 2.81950 3.23438 -0.31567 -0.37840 0.71854

Rdot1<-((1-Z)/(1-eX))*eX-(Z/eX)*(1-eX)
Rdot2<-X1*Rdot1
Rdot3<-X2*Rdot1

J.n<-matrix(0,5,5)
J.n[1,1]<-mean(R)
J.n[1,2]<-mean(R*Z)
J.n[1,3]<-mean(-Rdot1*eps.y)
J.n[1,4]<-mean(-Rdot2*eps.y)
J.n[1,5]<-mean(-Rdot3*eps.y)
J.n[2,1]<-mean(R*Z)
J.n[2,2]<-mean(R*Z^2)
J.n[2,3]<-mean(-Rdot1*Z*eps.y)
J.n[2,4]<-mean(-Rdot2*Z*eps.y)
J.n[2,5]<-mean(-Rdot3*Z*eps.y)
J.n[3:5,3:5]<-crossprod(Xa*sqrt(vx))/n; round(J.n,5);J.n.ipw2<-J.n

+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,] 2.09477 1.08486 -1.19491 -1.40032 2.81950
+ [2,] 1.08486 1.08486 -0.58013 -0.19082 3.23438
+ [3,] 0.00000 0.00000 0.17384 0.20795 -0.32187
+ [4,] 0.00000 0.00000 0.20795 0.36619 -0.40219
+ [5,] 0.00000 0.00000 -0.32187 -0.40219 0.74049

#Asymptotic variance
V.n<-solve(J.n) %*% (I.n %*% t(solve(J.n))); round(V.n,5); V.n.ipw2<-V.n

+           [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,] 33.05424 -10.92430 -2.76831 2.26022 0.01873
+ [2,] -10.92430 25.22783 6.34984 -5.07426 0.10614
+ [3,] -2.76831 6.34984 37.36367 -10.11968 10.43912
+ [4,] 2.26022 -5.07426 -10.11968 11.12457 1.82030
+ [5,] 0.01873 0.10614 10.43912 1.82030 6.80000

```

```
#Approximate estimate variance
round(V.n/n,5)
```

```
+      [,1]      [,2]      [,3]      [,4]      [,5]
+ [1,]  0.11018 -0.03641 -0.00923  0.00753  0.00006
+ [2,] -0.03641  0.08409  0.02117 -0.01691  0.00035
+ [3,] -0.00923  0.02117  0.12455 -0.03373  0.03480
+ [4,]  0.00753 -0.01691 -0.03373  0.03708  0.00607
+ [5,]  0.00006  0.00035  0.03480  0.00607  0.02267
```

```
se.vec<-sqrt(diag(V.n)/n); se.vec
```

```
+ [1]  0.3319349  0.2899875  0.3529101  0.1925666  0.1505545
```

```
All.se[4]<-se.vec[2]
```

Therefore the estimate is 2.00088 with standard error 0.28999.

4 Marks

(d) An augmented outcome regression model

$$Y_i = \beta_0 + \psi Z_i + \phi_0 \frac{1 - Z_i}{1 - e(X_{i1}, X_{i2}; \hat{\alpha})} + \phi_1 \frac{Z_i}{e(X_{i1}, X_{i2}; \hat{\alpha})}$$

with  $\alpha$  estimated using the propensity score model from (b).

**Solution:** For augmented outcome regression model with estimated propensity score, if

$$R_0 = \frac{1 - Z}{1 - e(X; \alpha)} \quad R_1 = \frac{Z}{e(X; \alpha)}$$

we have

$$\mathbf{U}(\theta) = \begin{pmatrix} (Y - \beta_0 - \psi Z - \phi_0 R_0 - \phi_1 R_1) \\ Z(Y - \beta_0 - \psi Z - \phi_0 R_0 - \phi_1 R_1) \\ R_0(Y - \beta_0 - \psi Z - \phi_0 R_0 - \phi_1 R_1) \\ R_1(Y - \beta_0 - \psi Z - \phi_0 R_0 - \phi_1 R_1) \\ \mathbf{x}^\top (Z - e(X; \alpha)) \end{pmatrix} = \begin{pmatrix} \varepsilon_Y \\ Z\varepsilon_Y \\ R_0\varepsilon_Y \\ R_1\varepsilon_Y \\ \mathbf{x}^\top \varepsilon_Z \end{pmatrix}.$$

where  $\mathbf{x} = (1, x_1, x_2)$ . We have therefore that

$$\mathcal{I} = \mathbb{E} \begin{bmatrix} \varepsilon_Y^2 & Z\varepsilon_Y^2 & R_0\varepsilon_Y^2 & R_1\varepsilon_Y^2 & \mathbf{x}\varepsilon_Z\varepsilon_Y \\ Z\varepsilon_Y^2 & Z^2\varepsilon_Y^2 & ZR_0\varepsilon_Y^2 & ZR_1\varepsilon_Y^2 & \mathbf{x}Z\varepsilon_Z\varepsilon_Y \\ R_0\varepsilon_Y^2 & ZR_0\varepsilon_Y^2 & R_0^2\varepsilon_Y^2 & R_0R_1\varepsilon_Y^2 & \mathbf{x}R_0\varepsilon_Z\varepsilon_Y \\ R_1\varepsilon_Y^2 & ZR_1\varepsilon_Y^2 & R_0R_1\varepsilon_Y^2 & R_1^2\varepsilon_Y^2 & \mathbf{x}R_1\varepsilon_Z\varepsilon_Y \\ \mathbf{x}^\top \varepsilon_Z\varepsilon_Y & \mathbf{x}^\top Z\varepsilon_Z\varepsilon_Y & \mathbf{x}^\top R_0\varepsilon_Z\varepsilon_Y & \mathbf{x}^\top R_1\varepsilon_Z\varepsilon_Y & \mathbf{x}^\top \mathbf{x}\varepsilon_Z^2 \end{bmatrix}$$

and

$$\mathcal{J} = \mathbb{E} \begin{bmatrix} 1 & Z & R_0 & R_1 & \phi_0 \dot{\mathbf{R}}_0 + \phi_1 \dot{\mathbf{R}}_1 \\ Z & Z^2 & ZR_0 & ZR_1 & Z(\phi_0 \dot{\mathbf{R}}_0 + \phi_1 \dot{\mathbf{R}}_1) \\ R_0 & ZR_0 & R_0^2 & R_0R_1 & R_0(\phi_0 \dot{\mathbf{R}}_0 + \phi_1 \dot{\mathbf{R}}_1) - \dot{\mathbf{R}}_0\varepsilon_Y \\ R_1 & ZR_1 & R_0R_1 & R_1^2 & R_1(\phi_0 \dot{\mathbf{R}}_0 + \phi_1 \dot{\mathbf{R}}_1) - \dot{\mathbf{R}}_1\varepsilon_Y \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{x}^\top \mathbf{x}e(1 - e) \end{bmatrix}.$$

where

$$\dot{\mathbf{R}}_0 = \frac{\partial}{\partial \alpha} \left\{ \frac{1-Z}{1-e(X; \alpha)} \right\} = \frac{1-Z}{1-e(X; \alpha)} e(X; \alpha) \mathbf{x}^\top = R_0 e(X; \alpha) \mathbf{x}^\top$$

$$\dot{\mathbf{R}}_1 = \frac{\partial}{\partial \alpha} \left\{ \frac{Z}{e(X; \alpha)} \right\} = -\frac{Z}{e(X; \alpha)} (1-e(X; \alpha)) \mathbf{x}^\top = -R_1 (1-e(X; \alpha)) \mathbf{x}^\top$$

```
eX<-fitted(glm(Z~X1+X2,family=binomial,data=P3.data))
vx<-eX*(1-eX)
Ptmp<-P3.data
Ptmp$eX<-eX

R0<-(1-Z)/(1-eX)
R1<-Z/eX

Ptmp$R0<-R0
Ptmp$R1<-R1
Ptmp$R<-R0+R1

fit.aor<-lm(Y~Z+R0+R1,data=Ptmp)
aor.ests<-coef(fit.aor)
All.ests[5]<-aor.ests[2]

Xa<-cbind(1,X1,X2)
eps.y<-residuals(fit.aor)
eps.z<-Z-eX

I.n<-crossprod(cbind(eps.y,Z*eps.y,R0*eps.y,R1*eps.y,Xa*eps.z))/n; I.n.aor<-I.n
round(I.n,5)

+          eps.y          X1          X2
+ eps.y 12.81094  5.36826 12.97726 18.61146 -0.22512 -0.15826  0.42238
+       5.36826  5.36826  0.00000 18.61146 -0.14200 -0.36077  0.72279
+       12.97726  0.00000 35.52626  0.00000  0.00000  0.95614 -0.66488
+       18.61146 18.61146  0.00000 218.23535  0.00000 -2.49236  0.83269
+       -0.22512 -0.14200  0.00000  0.00000  0.17115  0.20092 -0.31567
+ X1      -0.15826 -0.36077  0.95614 -2.49236  0.20092  0.38387 -0.37840
+ X2       0.42238  0.72279 -0.66488  0.83269 -0.31567 -0.37840  0.71854

J.n<-matrix(0,7,7)
Xp<-cbind(1,Z,R0,R1)

dR0<-R0*eX
dR1<-R1*(1-eX)
dterm<-(aor.ests[3]*dR0+aor.ests[4]*dR1)

J.n[1:4,1:4]<-crossprod(Xp)/n
J.n[5:7,5:7]<-crossprod(Xa*sqrt(vx))/n
J.n[1,5]<-mean(dterm)
J.n[1,6]<-mean(dterm*X1)
J.n[1,7]<-mean(dterm*X2)
J.n[2,5]<-mean(Z*dterm)
J.n[2,6]<-mean(Z*dterm*X1)
J.n[2,7]<-mean(Z*dterm*X2)
J.n[3,5]<-mean((R0*dterm-dR0*eps.y))
J.n[3,6]<-mean((R0*dterm-dR0*eps.y)*X1)
J.n[3,7]<-mean((R0*dterm-dR0*eps.y)*X2)
J.n[4,5]<-mean((R1*dterm-dR1*eps.y))
J.n[4,6]<-mean((R1*dterm-dR1*eps.y)*X1)
J.n[4,7]<-mean((R1*dterm-dR1*eps.y)*X2)
round(J.n,5);J.n.aor<-J.n
```

```

+      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
+ [1,] 1.00000 0.32667 1.00991 1.08486 0.85966 1.08309 -1.51096
+ [2,] 0.32667 0.32667 0.00000 1.08486 0.23860 0.10513 -0.53468
+ [3,] 1.00991 0.00000 2.01446 0.00000 1.85359 5.07249 -2.86226
+ [4,] 1.08486 1.08486 0.00000 9.72486 2.71902 -3.85146 -6.36196
+ [5,] 0.00000 0.00000 0.00000 0.00000 0.17384 0.20795 -0.32187
+ [6,] 0.00000 0.00000 0.00000 0.00000 0.20795 0.36619 -0.40219
+ [7,] 0.00000 0.00000 0.00000 0.00000 -0.32187 -0.40219 0.74049

#Asymptotic variance
V.n<-solve(J.n) %*% (I.n %*% t(solve(J.n))); round(V.n,5); V.n.aor<-V.n

+      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
+ [1,] 564.80096 -735.27520 -380.83757 55.35497 -56.96572 70.61393 11.45389
+ [2,] -735.27520 1059.33508 495.35973 -88.41583 99.66298 -98.76986 -5.69088
+ [3,] -380.83757 495.35973 267.67734 -36.56264 26.58153 -47.74176 -12.10463
+ [4,] 55.35497 -88.41583 -36.56264 11.69059 -9.20931 9.43420 1.94692
+ [5,] -56.96572 99.66298 26.58153 -9.20931 37.36367 -10.11968 10.43912
+ [6,] 70.61393 -98.76986 -47.74176 9.43420 -10.11968 11.12457 1.82030
+ [7,] 11.45389 -5.69088 -12.10463 1.94692 10.43912 1.82030 6.80000

#Approximate estimate variance
round(V.n/n,5)

+      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
+ [1,] 1.88267 -2.45092 -1.26946 0.18452 -0.18989 0.23538 0.03818
+ [2,] -2.45092 3.53112 1.65120 -0.29472 0.33221 -0.32923 -0.01897
+ [3,] -1.26946 1.65120 0.89226 -0.12188 0.08861 -0.15914 -0.04035
+ [4,] 0.18452 -0.29472 -0.12188 0.03897 -0.03070 0.03145 0.00649
+ [5,] -0.18989 0.33221 0.08861 -0.03070 0.12455 -0.03373 0.03480
+ [6,] 0.23538 -0.32923 -0.15914 0.03145 -0.03373 0.03708 0.00607
+ [7,] 0.03818 -0.01897 -0.04035 0.00649 0.03480 0.00607 0.02267

se.vec<-sqrt(diag(V.n)/n); se.vec

+ [1] 1.3721042 1.8791266 0.9445940 0.1974048 0.3529101 0.1925666 0.1505545

All.se[5]<-se.vec[2]

```

Therefore the estimate is 8.50251 with standard error 1.87913.

6 Marks

**Note:** In this model, the parameter  $\psi$  does not quantify the causal effect of  $Z$  on  $Y$  unless the outcome mean model component is correctly specified. If this were true, we would expect to see the estimates of  $\phi_0$  and  $\phi_1$  close to zero:

```

round(summary(fit.aor)$coef,5)

+      Estimate Std. Error  t value Pr(>|t|)
+ (Intercept) -6.67032    0.50903 -13.10392 0.00000
+ Z           8.50251    0.68526  12.40781 0.00000
+ R0          1.84520    0.29429   6.26991 0.00000
+ R1         -0.31470    0.08408  -3.74285 0.00022

```

The coefficients of  $R_0$  and  $R_1$  are seemingly different from zero on this basis of the OLS analysis; using the sandwich form of the asymptotic variance, we have the following equivalent calculation:

```

aor.summary<-matrix(0,4,4)
aor.summary[,1]<-aor.ests

```

```

aor.summary[,2]<-se.vec[1:4]
aor.summary[,3]<-aor.ests/se.vec[1:4]
aor.summary[,4]<-2*pnorm(-abs(aor.ests/se.vec[1:4]))
rownames(aor.summary)<-c('(Intercept)', 'Z', 'R0', 'R1')
colnames(aor.summary)<-c('Estimate', 'Std. Error', 't value', 'Pr(>|t|)')
round(aor.summary,5)

```

```

+           Estimate Std. Error  t value Pr(>|t|)
+ (Intercept) -6.67032    1.37210 -4.86138  0.00000
+ Z           8.50251    1.87913  4.52472  0.00001
+ R0          1.84520    0.94459  1.95343  0.05077
+ R1         -0.31470    0.19740 -1.59419  0.11089

```

```

newd0<-Ptmp;newd0$Z<-0;newd0$R0<-1/(1-eX);newd0$R1<-0
newd1<-Ptmp;newd1$Z<-1;newd1$R1<-1/eX;newd1$R0<-0

```

```

mu0.est<-mean(predict(fit.aor,newdata=newd0))
mu1.est<-mean(predict(fit.aor,newdata=newd1))
c(mu0.est,mu1.est,mu1.est-mu0.est)

```

```

+ [1] -3.33685093 -0.08029446  3.25655646

```

```

All.ests[6]<-mu1.est-mu0.est

```

Therefore, using the prediction method based on the fitted AOR, we have that the estimated average treatment effect is 3.25656.

```

library(sandwich)
round(sandwich(fit.aor),5)

```

```

+           (Intercept)           Z           R0           R1
+ (Intercept)    0.40142 -0.40142 -0.24912  0.00000
+ Z             -0.40142  0.71132  0.24912 -0.04413
+ R0            -0.24912  0.24912  0.17808  0.00000
+ R1             0.00000 -0.04413  0.00000  0.01368

```

**Additional Results:** We can check the sense of these results using the bootstrap

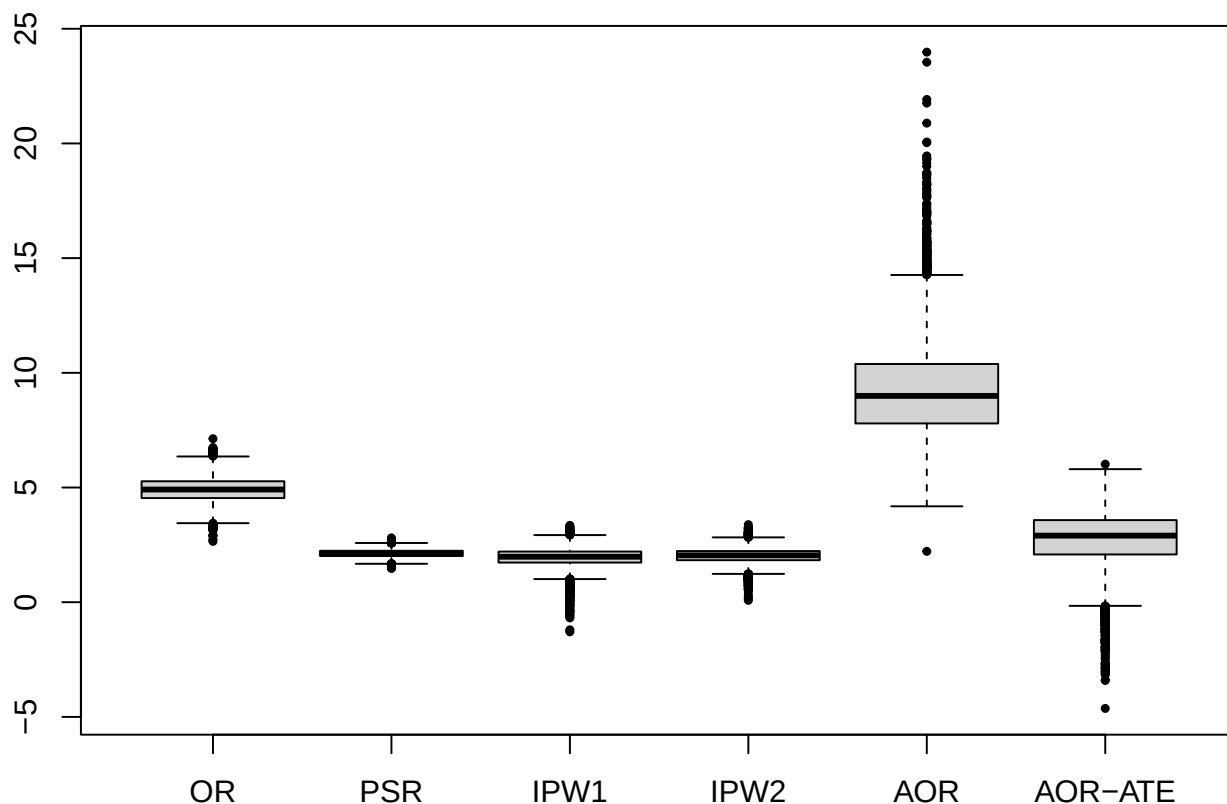
```
nboot<-10000; ests.mat<-matrix(0,nrow=nboot,ncol=6)
for(ib in 1:nboot){
  ind<-sample(1:n,size=n,rep=T)
  Pboot<-P3.data[ind,]
  ps1<-fitted(glm(Z~X1+X2,family=binomial,data=Pboot))

  Pboot$eX<-ps1; Pboot$R0<-(1-Pboot$Z)/(1-ps1); Pboot$R1<-Pboot$Z/ps1
  Pboot$R<-Pboot$R0+Pboot$R1
  wv<-Pboot$R1-Pboot$R0

  ests.mat[ib,1]<-coef(lm(Y~X1+Z,data=Pboot))[3]
  ests.mat[ib,2]<-coef(lm(Y~Z+eX,data=Pboot))[2]
  ests.mat[ib,3]<-mean(wv*Pboot$Y)
  ests.mat[ib,4]<-coef(lm(Y~Z,weights=R,data=Pboot))[2]
  fit.aor<-lm(Y~Z+R0+R1,data=Pboot)
  ests.mat[ib,5]<-coef(fit.aor)[2]
  newd0<-Pboot;newd0$Z<-0;newd0$R0<-1/(1-eX);newd0$R1<-0
  newd1<-Pboot;newd1$Z<-1;newd1$R1<-1/eX;newd1$R0<-0

  mu0.est<-mean(predict(fit.aor,newdata=newd0))
  mu1.est<-mean(predict(fit.aor,newdata=newd1))
  ests.mat[ib,6]<-mu1.est-mu0.est
}
```

```
colnames(ests.mat)<-c('OR','PSR','IPW1','IPW2','AOR','AOR-ATE')
par(mar=c(4,3,1,1))
boxplot(ests.mat,pch=19,cex=0.5)
```



```

cmat<-rbind(apply(ests.mat,2,mean),All.ests);rownames(cmat)<-c('Boot','Est.');
```

|        | OR       | PSR      | IPW1     | IPW2     | AOR      | AOR-ATE  |
|--------|----------|----------|----------|----------|----------|----------|
| + Boot | 4.906535 | 2.128953 | 1.937601 | 2.022149 | 9.242577 | 2.738781 |
| + Est. | 4.897658 | 2.120745 | 1.946609 | 2.000880 | 8.502515 | 3.256556 |

```

smat<-rbind(apply(ests.mat,2,sd),All.se);rownames(smat)<-c('Boot','Est.');
```

|        | OR        | PSR       | IPW1      | IPW2      | AOR      | AOR-ATE  |
|--------|-----------|-----------|-----------|-----------|----------|----------|
| + Boot | 0.5431530 | 0.1682682 | 0.4306815 | 0.3320229 | 2.070406 | 1.217192 |
| + Est. | 0.5419244 | 0.1680721 | 0.3525689 | 0.2899875 | 1.879127 | 0.000000 |

**Computing the sandwich estimate numerically:** We can use the `nleqslv` package to return estimates that solve the estimating equations, and also to compute the  $\hat{U}$  estimated values in the jacobian argument.

### Outcome regression:

```

library(nleqslv)
or.eqn<-function(xp,dfx){
  be<-xp[1:3]
  Xp<-cbind(1,dfx$X1,dfx$Z)

  r1<-t(Xp) %*% (dfx$Y - Xp %*% be)
  return(r1)
}
or.hess<-function(xp,dfx){
  be<-xp[1:3]
  Xp<-cbind(1,dfx$X1,dfx$Z)
  epsy<-(dfx$Y - Xp %*% be)
  hmat<-crossprod(cbind(epsy,dfx$X1*epsy,dfx$Z*epsy))
  return(hmat)
}
xs<-rep(0,3)
fit.nl<-nleqslv(xs,fn=or.eqn,dfx=Ptmp,
               jacobian=TRUE,method='Newton',control=list(ftol=1e-10,maxit=500,allowSingular=TRUE))
est<-fit.nl$x
J.hat<-fit.nl$jac/n
I.hat<-or.hess(est,Ptmp)/n
V.hat.nl.or<-solve(J.hat) %*% I.hat %*% t(solve(J.hat))/n
```

### Propensity score regression:

```

library(nleqslv)
expit<-function(x){return(1/(1+exp(-x)))}
psr.eqn<-function(xp,dfx){
  be<-xp[1:3]
  al<-xp[4:6]
  Xa<-cbind(1,dfx$X1,dfx$X2)
  psv<- expit(Xa %*% al)
  Xp<-cbind(1,dfx$Z,psv)

  r1<-t(Xp) %*% (dfx$Y - Xp %*% be)
  r2<-t(Xa) %*% (dfx$Z - psv)
  return(c(r1,r2))
}
psr.hess<-function(xp,dfx){
  be<-xp[1:3]
  al<-xp[4:6]
  Xa<-cbind(1,dfx$X1,dfx$X2)
  psv<- expit(Xa %*% al)
```

```

Xp<-cbind(1,dfx$Z,psv)

epsy<-(dfx$Y - Xp %*% be)
epsz<-as.vector(dfx$Z - psv)

hmat<-crossprod(cbind(epsy,dfx$Z*epsy,psv*epsy,Xa*epsz))
return(hmat)
}
xs<-rep(0,6)
fit.nl<-nleqslv(xs,fn=psr.eqn,dfx=Ptmp,
               jacobian=TRUE,method='Newton',control=list(ftol=1e-10,maxit=500,allowSingular=TRUE))
est<-fit.nl$x
J.hat<-fit.nl$jac/n
I.hat<-psr.hess(est,Ptmp)/n
V.hat.nl.psr<-solve(J.hat) %*% I.hat %*% t(solve(J.hat))/n

```

### IPW version 1:

```

library(nleqslv)
expit<-function(x){return(1/(1+exp(-x)))}
ipw1.eqn<-function(xp,dfx){
  be<-xp[1]
  al<-xp[2:4]
  Xa<-cbind(1,dfx$X1,dfx$X2)
  psv<- expit(Xa %*% al)
  Rv<-as.vector(dfx$Z/psv-(1-dfx$Z)/(1-psv))

  r1<-rep(1,n)%*%(Rv*dfx$Y-be)
  r2<-t(Xa) %*% (dfx$Z - psv)
  return(c(r1,r2))
}
ipw1.hess<-function(xp,dfx){
  be<-xp[1]
  al<-xp[2:4]
  Xa<-cbind(1,dfx$X1,dfx$X2)
  psv<- expit(Xa %*% al)
  Rv<-dfx$Z/psv-(1-dfx$Z)/(1-psv)

  epsy<-Rv*dfx$Y-be
  epsz<-as.vector(dfx$Z - psv)

  hmat<-crossprod(cbind(epsy,Xa*epsz))
  return(hmat)
}
xs<-rep(0,4)
fit.nl<-nleqslv(xs,fn=ipw1.eqn,dfx=Ptmp,
               jacobian=TRUE,method='Newton',control=list(ftol=1e-10,maxit=500))
est<-fit.nl$x
J.hat<-fit.nl$jac/n
I.hat<-ipw1.hess(est,Ptmp)/n
V.hat.nl.ipw1<-solve(J.hat) %*% I.hat %*% t(solve(J.hat))/n

```

### IPW version 2:

```

library(nleqslv)
expit<-function(x){return(1/(1+exp(-x)))}
ipw2.eqn<-function(xp,dfx){
  be<-xp[1:2]
  al<-xp[3:5]

```

```

Xa<-cbind(1,dfx$X1,dfx$X2)
Xp<-cbind(1,dfx$Z)
psv<- expit(Xa %*% al)
Rv<-(1-dfx$Z)/(1-psv)+dfx$Z/psv

r1<-t(Xp) %*% (Rv*(dfx$Y - Xp %*% be))
r2<-t(Xa) %*% (dfx$Z - psv)
return(c(r1,r2))
}
ipw2.hess<-function(xp,dfx){
  be<-xp[1:2]
  al<-xp[3:5]
  Xa<-cbind(1,dfx$X1,dfx$X2)
  Xp<-cbind(1,dfx$Z)
  psv<- expit(Xa %*% al)
  Rv<-(1-dfx$Z)/(1-psv)+dfx$Z/psv

  epsy<-(dfx$Y - Xp %*% be)
  epsz<-as.vector(dfx$Z - psv)

  hmat<-crossprod(cbind(Rv*epsy,Rv*dfx$Z*epsy,Xa*epsz))
  return(hmat)
}
xs<-rep(0,5)
fit.nl<-nleqslv(xs,fn=ipw2.eqn,dfx=Ptmp,
               jacobian=TRUE,method='Newton',control=list(ftol=1e-10,maxit=500))
est<-fit.nl$x
J.hat<-fit.nl$jac/n
I.hat<-ipw2.hess(est,Ptmp)/n
V.hat.nl.ipw2<-solve(J.hat) %*% I.hat %*% t(solve(J.hat))/n

```

### Augmented Outcome Regression:

```

library(nleqslv)
expit<-function(x){return(1/(1+exp(-x)))}
aor.eqn<-function(xp,dfx){
  be<-xp[1:4]
  al<-xp[5:7]
  Xa<-cbind(1,dfx$X1,dfx$X2)
  psv<- expit(Xa %*% al)
  R0v<-(1-dfx$Z)/(1-psv)
  R1v<-dfx$Z/psv
  Xp<-cbind(1,dfx$Z,R0v,R1v)

  r1<-t(Xp) %*% (dfx$Y - Xp %*% be)
  r2<-t(Xa) %*% (dfx$Z - psv)
  return(c(r1,r2))
}
aor.hess<-function(xp,dfx){
  be<-xp[1:4]
  al<-xp[5:7]
  Xa<-cbind(1,dfx$X1,dfx$X2)
  psv<- expit(Xa %*% al)
  R0v<-(1-dfx$Z)/(1-psv)
  R1v<-dfx$Z/psv
  Xp<-cbind(1,dfx$Z,R0v,R1v)

  epsy<-(dfx$Y - Xp %*% be)
  epsz<-as.vector(dfx$Z - psv)

```

```

    hmat<-crossprod(cbind(epsy,dfx$Z*epsy,R0v*epsy,R1v*epsy,Xa*epsz))
    return(hmat)
  }
  xs<-rep(0,7)
  fit.nl<-nleqslv(xs,fn=aor.eqn,dfx=Ptmp,
    jacobian=TRUE,method='Newton',control=list(ftol=1e-10,maxit=500,allowSingular=TRUE))
  xs<-fit.nl$x
  fit.nl<-nleqslv(xs,fn=aor.eqn,dfx=Ptmp,
    jacobian=TRUE,method='Newton',control=list(ftol=1e-10,maxit=500))
  est<-fit.nl$x
  J.hat<-fit.nl$jac/n
  I.hat<-aor.hess(est,Ptmp)/n
  V.hat.nl.aor<-solve(J.hat) %*% (I.hat %*% t(solve(J.hat)))/n

```

```

smat<-rbind(apply(ests.mat,2,sd),All.se,
  c(sqrt(V.hat.nl.or[3,3]),sqrt(V.hat.nl.psr[2,2]),sqrt(V.hat.nl.ipw1[1,1]),
    sqrt(V.hat.nl.ipw2[2,2]),sqrt(V.hat.nl.aor[2,2]),0))
rownames(smat)<-c('Boot','Est.','NL');smat

+          OR          PSR          IPW1          IPW2          AOR  AOR-ATE
+ Boot 0.5431530 0.1682682 0.4306815 0.3320229 2.070406 1.217192
+ Est. 0.5419244 0.1680721 0.3525689 0.2899875 1.879127 0.000000
+ NL   0.5419244 0.1680721 0.3525689 0.2899875 1.879127 0.000000

```

Note that it is sometimes useful to compute some of the expectations analytically if it is possible, assuming correct specification. For example, we have in this binary treatment case that

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

meaning that there are two possible estimators based on the data, namely

$$\frac{1}{n} \sum_{i=1}^n (Z_i - e(X_i))^2 \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^n e(X_i)(1 - e(X_i)).$$

The second estimator assumes that  $\mathbb{E}[Z|X] = e(X)$ , that is, that we have correct specification of the PS, that is required for IPW to give a consistent result in estimation.