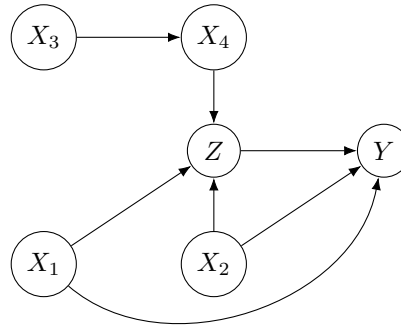


MATH 598/782 – PROJECT 2 SOLUTIONS

The data set

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

comprises four predictor variables X_1, X_2, X_3, X_4 , a binary treatment variable Z , and outcome variable Y . The following DAG is understood to capture the structural relationships between the variables.



1. Carry out an analysis based on the following causal adjustment procedures to estimate the average treatment effect of Z on Y .
 - (a) Propensity score regression based on a linear outcome model, where the effect of treatment is captured by the additive parametric component $\psi_0 Z$. 4 Marks
 - (b) Inverse probability weighting (IPW). 4 Marks
 - (c) Augmented inverse probability weighting (AIPW). 4 Marks

In each case, use a **bootstrap** procedure to estimate the variance of the estimator as follows: for $N = 1000$ resampled replications, resample a new data set of the same sample size, equally weighted and with replacement, from the original data set, and repeat the estimation procedure for each resampled data set. Then report the variance of the 1000 sampled estimates.

Solution: For this DAG,

- there are two open undirected paths between Z and Y ; these are confounding paths, which must be blocked by conditioning;
- X_1 and X_2 are confounders, and we can construct the propensity score by regressing Z on X_1 and X_2 ;
- we do not know precisely the form of $e(x_1, x_2) = \Pr[Z = 1 | X_1 = x_1, X_2 = x_2]$, so we should perhaps try different functional forms, such as including terms in X_1^2, X_2^2 . The structural DAG does not indicate that such terms are present, however, so to simplify the analysis we omit them here.
- no interactions are present anywhere in the structural DAG; therefore there is no need to include interactions in the propensity score model;
- X_4 is a predictor of Z , but is not a confounder; it is therefore an instrument, and we do not need to include it in the analysis;
- X_3 is an indirect cause of Z ;
- for the outcome model, we might compare models that include (i) Z only, (ii) Z and X_1 , (iii) Z and X_2 , (iv) Z and (X_1, X_2) as well as the propensity score. We do not know the functional form, but these models should give us a reasonable impression of the causal effect.

- (a) Propensity score regression:

```
fn<-"https://www.math.mcgill.ca/dstephens/598-Causal-2024/Exercises/Project2/P2data.csv"
P2.data<-read.csv(fn);n<-nrow(P2.data);table(P2.data$Z)
```

```

+
+   0   1
+ 202  98

ps1<-fitted(glm(Z~X1+X2,family=binomial,data=P2.data))

fit00<-lm(Y~Z,data=P2.data)
fit01<-lm(Y~Z+X1,data=P2.data)
fit02<-lm(Y~Z+X2,data=P2.data)
fit03<-lm(Y~Z+X1+X2,data=P2.data)
fit10<-lm(Y~Z+ps1,data=P2.data)
fit11<-lm(Y~Z+X1+ps1,data=P2.data)
fit12<-lm(Y~Z+X2+ps1,data=P2.data)
fit13<-lm(Y~Z+X1+X2+ps1,data=P2.data)
enames<-c('Un','OR1','OR2','OR1','PSR1','PSR2','PSR3','PSR4')

est.vec<-c(coef(fit00)[2],coef(fit01)[2],coef(fit02)[2],coef(fit03)[2],
  coef(fit10)[2],coef(fit11)[2],coef(fit12)[2],coef(fit13)[2])
names(est.vec)<-enames
est.vec

+      Un      OR1      OR2      OR1      PSR1      PSR2      PSR3      PSR4
+ 4.689834 4.897658 2.363256 2.399775 2.120745 1.994893 2.205379 2.118157

#Bootstrap
set.seed(234)
N<-1000
ests.mat.PSR<-matrix(0,nrow=N,ncol=8)
for(ib in 1:N){

  ind<-sample(1:n,size=n,rep=T)
  P2.boot<-P2.data[ind,]

  ps1<-fitted(glm(Z~X1+X2,family=binomial,data=P2.boot))

  fit00<-lm(Y~Z,data=P2.boot)
  fit01<-lm(Y~Z+X1,data=P2.boot)
  fit02<-lm(Y~Z+X2,data=P2.boot)
  fit03<-lm(Y~Z+X1+X2,data=P2.boot)
  fit10<-lm(Y~Z+ps1,data=P2.boot)
  fit11<-lm(Y~Z+X1+ps1,data=P2.boot)
  fit12<-lm(Y~Z+X2+ps1,data=P2.boot)
  fit13<-lm(Y~Z+X1+X2+ps1,data=P2.boot)

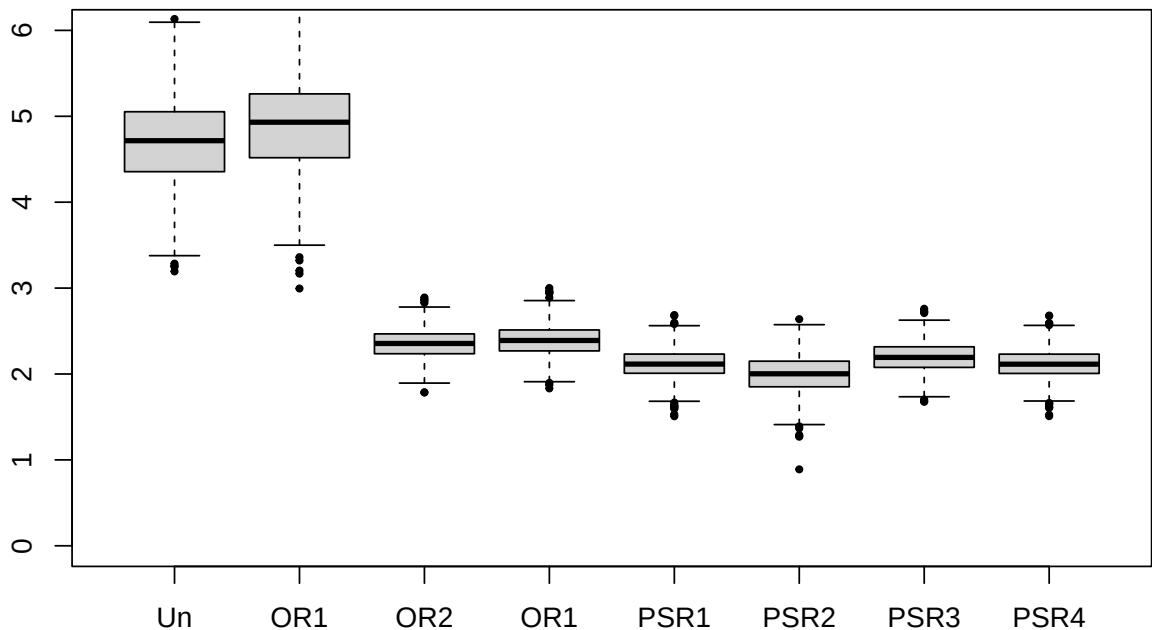
  est.vec<-c(coef(fit00)[2],coef(fit01)[2],coef(fit02)[2],coef(fit03)[2],
    coef(fit10)[2],coef(fit11)[2],coef(fit12)[2],coef(fit13)[2])
  ests.mat.PSR[ib,]<-est.vec
}

esum<-rbind(apply(ests.mat.PSR,2,mean),n*apply(ests.mat.PSR,2,var))
colnames(esum)<-enames
rownames(esum)<-c('Est','n.Var')
round(esum,5)

+      Un      OR1      OR2      OR1      PSR1      PSR2      PSR3      PSR4
+ Est    4.69536 4.90647 2.35324 2.38952 2.11658 1.99343 2.19536 2.11635
+ n.Var 74.85601 88.06200 8.68715 10.27465 8.86353 14.30714 9.44535 8.95670

par(mar=c(4,3,2,0))
boxplot(ests.mat.PSR,ylim=range(0,6),pch=19,cex=0.5,names=enames)

```



(b) IPW estimation: we use IPW methods for the two estimators $\tilde{\mu}_{IPW}$ and $\hat{\mu}_{IPW}$.

```
ps1<-fitted(glm(Z~X1+X2,family=binomial,data=P2.data))
Zv<-P2.data$Z
Yv<-P2.data$Y
```

```
w1<-(1-Zv)/(1-ps1)+Zv/ps1
```

```
fit31<-lm(Y~Z,weights=w1,data=P2.data)
```

```
ate1<-mean(Zv*Yv/ps1)-mean((1-Zv)*Yv/(1-ps1))
```

```
est.vec<-c(ate1,coef(fit31)[2])
names(est.vec)<-c('Mu.tilde','Mu.hat')
est.vec
```

```
+ Mu.tilde Mu.hat
+ 1.946609 2.000880
```

```
#Bootstrap
set.seed(234)
ests.mat.IPW<-matrix(0,nrow=N,ncol=2)
for(ib in 1:N){

  ind<-sample(1:n,size=n,rep=T)
  P2.boot<-P2.data[ind,]

  ps1<-fitted(glm(Z~X1+X2,family=binomial,data=P2.boot))

  Zv<-P2.boot$Z
  Yv<-P2.boot$Y
  w1<-(1-Zv)/(1-ps1)+Zv/ps1
  fit31<-lm(Y~Z,weights=w1,data=P2.boot)
  ate1<-mean(Zv*Yv/ps1)-mean((1-Zv)*Yv/(1-ps1))
```

```

    est.vec<-c(ate1,coef(fit31)[2])
    ests.mat.IPW[ib,]<-est.vec
  }

```

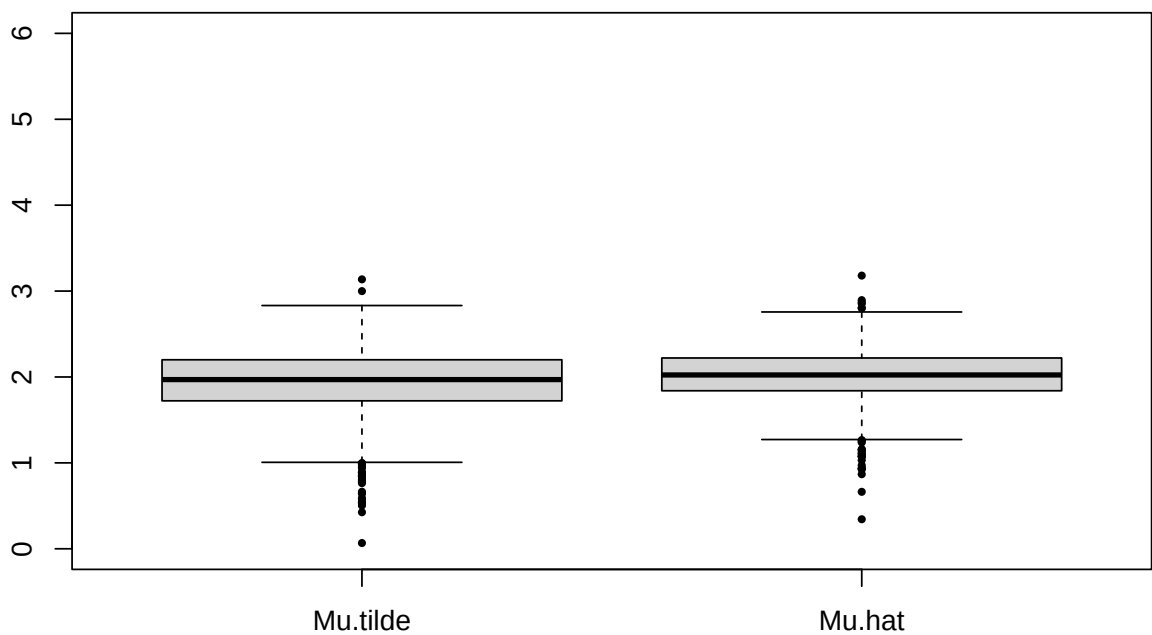
```

esum<-rbind(apply(ests.mat.IPW,2,mean),n*apply(ests.mat.IPW,2,var))
colnames(esum)<-c('Mu.tilde','Mu.hat')
rownames(esum)<-c('Est','n.Var')
round(esum,5)

+      Mu.tilde  Mu.hat
+ Est      1.9341  2.01252
+ n.Var    50.5028 30.31731

par(mar=c(4,3,2,0))
boxplot(ests.mat.IPW,ylim=range(0,6),pch=19,cex=0.5,names=c('Mu.tilde','Mu.hat'))

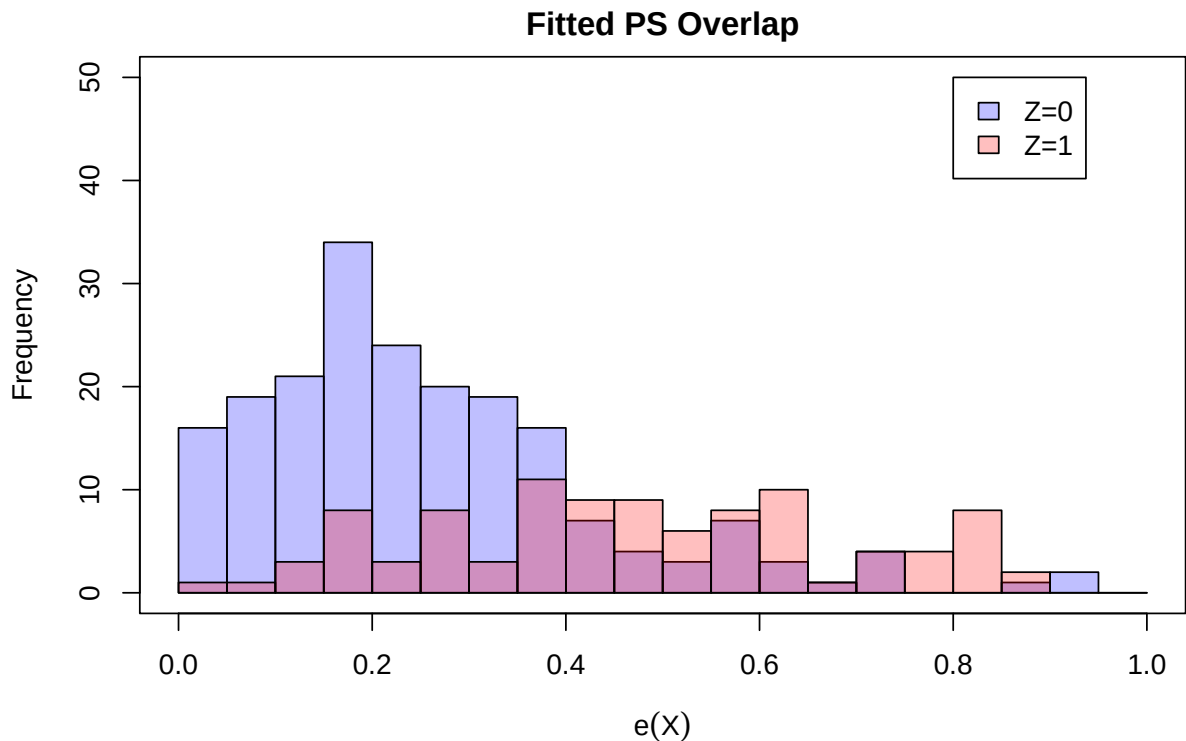
```



```

par(mar=c(4,4,2,0))
tmt.col<-c(rgb(0,0,1,1/4),rgb(1,0,0,1/4))
p1 <- hist(ps1[Zv==0],breaks=seq(0,1,by=0.05),plot=FALSE)
p2 <- hist(ps1[Zv==1],breaks=seq(0,1,by=0.05),plot=FALSE)
par(mar=c(4,4,2,0))
plot( p1, col=tmt.col[1], xlim=c(0,1), xlab=expression(e(X)),
      main='Fitted PS Overlap',ylim=range(0,50));box()
plot( p2, col=tmt.col[2], xlim=c(0,1),add=T)
legend(0.8,50,c('Z=0','Z=1'),fill=tmt.col)

```



- (c) AIPW estimation: we retain PS1, use the two estimators $\tilde{\mu}_{\text{AIPW}}$ and $\hat{\mu}_{\text{AIPW}}$ augmenting by using three different mean models $\mu(x, z)$

```
ps1<-fitted(glm(Z~X1+X2,family=binomial,data=P2.data))
fit.out<-lm(Y~Z*X1,data=P2.data)

new0<-P2.data;new0$Z<-0
new1<-P2.data;new1$Z<-1
mufit0<-predict(fit.out,new0)
mufit1<-predict(fit.out,new1)

Zv<-P2.data$Z
Yv<-P2.data$Y

w01<-(1-Zv)/(1-ps1);W01<-w01/sum(w01)
w11<-Zv/ps1;W11<-w11/sum(w11)

ate1.tilde<-(mean(w11*(Yv-mufit1))+mean(mufit1))-(mean(w01*(Yv-mufit0))+mean(mufit0))
ate1.hat<-(sum(W11*(Yv-mufit1))+mean(mufit1))-(sum(W01*(Yv-mufit0))+mean(mufit0))

est.vec<-c(ate1.tilde,ate1.hat)
est.vec
+ [1] 1.930824 2.068854
```

```
#Bootstrap
set.seed(234)
ests.mat.AIPW<-matrix(0,nrow=N,ncol=6)
for(ib in 1:N){

  ind<-sample(1:n,size=n,rep=T)
  P2.boot<-P2.data[ind,]
```

```

ps1<-fitted(glm(Z~X1+X2,family=binomial,data=P2.boot))

fit.out1<-lm(Y~Z*X1,data=P2.boot)
fit.out2<-lm(Y~Z*X2,data=P2.boot)
fit.out3<-lm(Y~Z*(X1+X2),data=P2.boot)

new0<-P2.boot;new0$Z<-0
new1<-P2.boot;new1$Z<-1
mufit01<-predict(fit.out1,new0);m01<-mean(mufit01)
mufit11<-predict(fit.out1,new1);m11<-mean(mufit11)
mufit02<-predict(fit.out2,new0);m02<-mean(mufit02)
mufit12<-predict(fit.out2,new1);m12<-mean(mufit12)
mufit03<-predict(fit.out3,new0);m03<-mean(mufit03)
mufit13<-predict(fit.out3,new1);m13<-mean(mufit13)

Zv<-P2.boot$Z
Yv<-P2.boot$Y

w01<-(1-Zv)/(1-ps1);W01<-w01/sum(w01)
w11<-Zv/ps1;W11<-w11/sum(w11)

ate1.tilde<-(mean(w11*(Yv-mufit11))+m11)-(mean(w01*(Yv-mufit01))+m01)
ate1.hat<-(sum(W11*(Yv-mufit11))+m11)-(sum(W01*(Yv-mufit01))+m01)
ate2.tilde<-(mean(w11*(Yv-mufit12))+m12)-(mean(w01*(Yv-mufit02))+m02)
ate2.hat<-(sum(W11*(Yv-mufit12))+m12)-(sum(W01*(Yv-mufit02))+m02)
ate3.tilde<-(mean(w11*(Yv-mufit13))+m13)-(mean(w01*(Yv-mufit03))+m03)
ate3.hat<-(sum(W11*(Yv-mufit13))+m13)-(sum(W01*(Yv-mufit03))+m03)

est.vec<-c(ate1.tilde,ate1.hat,ate2.tilde,ate2.hat,ate3.tilde,ate3.hat)

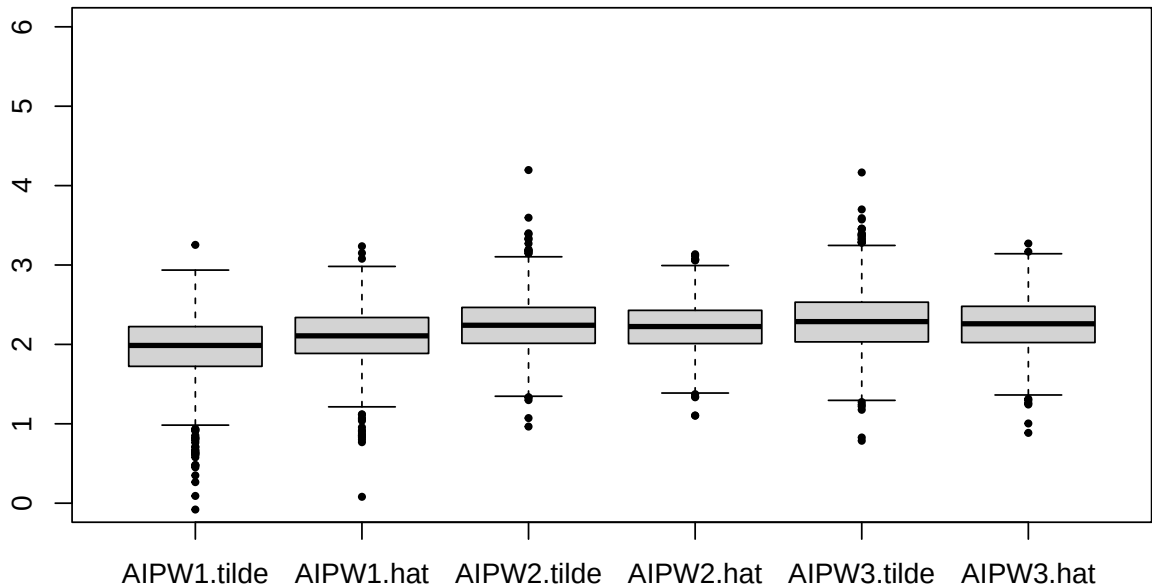
ests.mat.AIPW[ib,]<-est.vec
}

esum<-rbind(apply(ests.mat.AIPW,2,mean),n*apply(ests.mat.AIPW,2,var))
enames<-c('AIPW1.tilde','AIPW1.hat','AIPW2.tilde','AIPW2.hat',
          'AIPW3.tilde','AIPW3.hat')
colnames(esum)<-enames
rownames(esum)<-c('Est','n.Var')
round(esum,5)

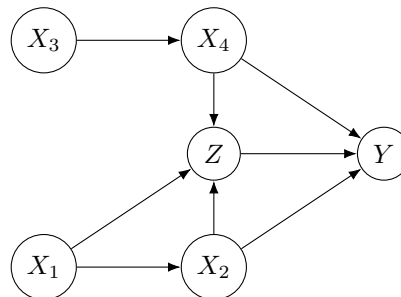
+      AIPW1.tilde AIPW1.hat AIPW2.tilde AIPW2.hat AIPW3.tilde AIPW3.hat
+ Est      1.93293  2.09688   2.24571   2.21364   2.28793   2.24414
+ n.Var    69.33227  38.47620   39.44261  29.86681   48.81184   34.99719

par(mar=c(4,3,2,0))
boxplot(ests.mat.AIPW,ylim=range(0,6),pch=19,cex=0.5,names=enames)

```



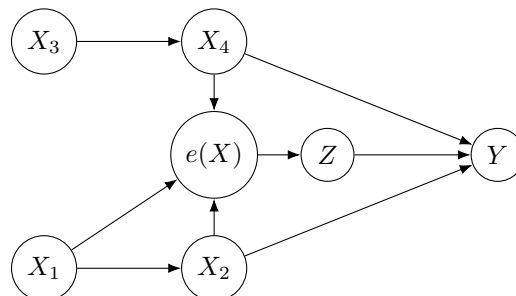
2. Suppose now that the structural DAG is believed to take the following form:



Explain how (if at all) the analysis should change from Q1; if the analysis changes, perform the new analysis using one of the methods from Q1; if it does not need to change, explain why not. 4 Marks

Solution: For this DAG,

- now X_4 is also a confounder as it lies on an open, undirected path between Z and Y ;
- X_1 has an indirect influence on Y mediated through X_2 ; it is still a confounder; if we omit X_1 from the propensity score, there is still an open confounding path from Z to Y via X_1 .
- this indicates that the propensity score should be based on (X_1, X_2, X_4) .



We begin with an AIPW analysis, and compare the two PS model

$$\text{PS 1} \quad X_1 + X_2 + X_4$$

$$\text{PS 2} \quad X_2 + X_4$$

and using the augmentation model that simply includes Z .

```
ps1<-fitted(glm(Z~X1+X2+X4,family=binomial,data=P2.data))
ps2<-fitted(glm(Z~X2+X4,family=binomial,data=P2.data))

fit.out<-lm(Y~Z,data=P2.data)

new0<-P2.data;new0$Z<-0
new1<-P2.data;new1$Z<-1
mufit0<-predict(fit.out,new0)
mufit1<-predict(fit.out,new1)

Zv<-P2.data$Z
Yv<-P2.data$Y

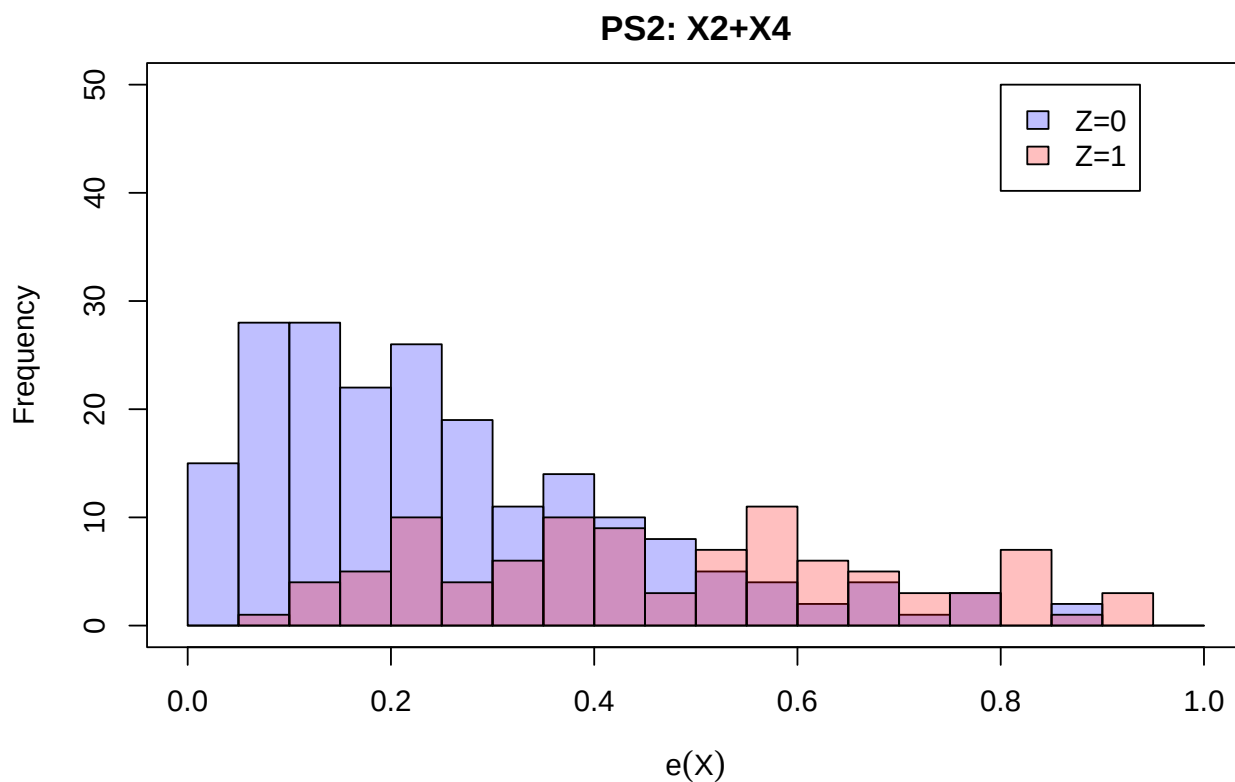
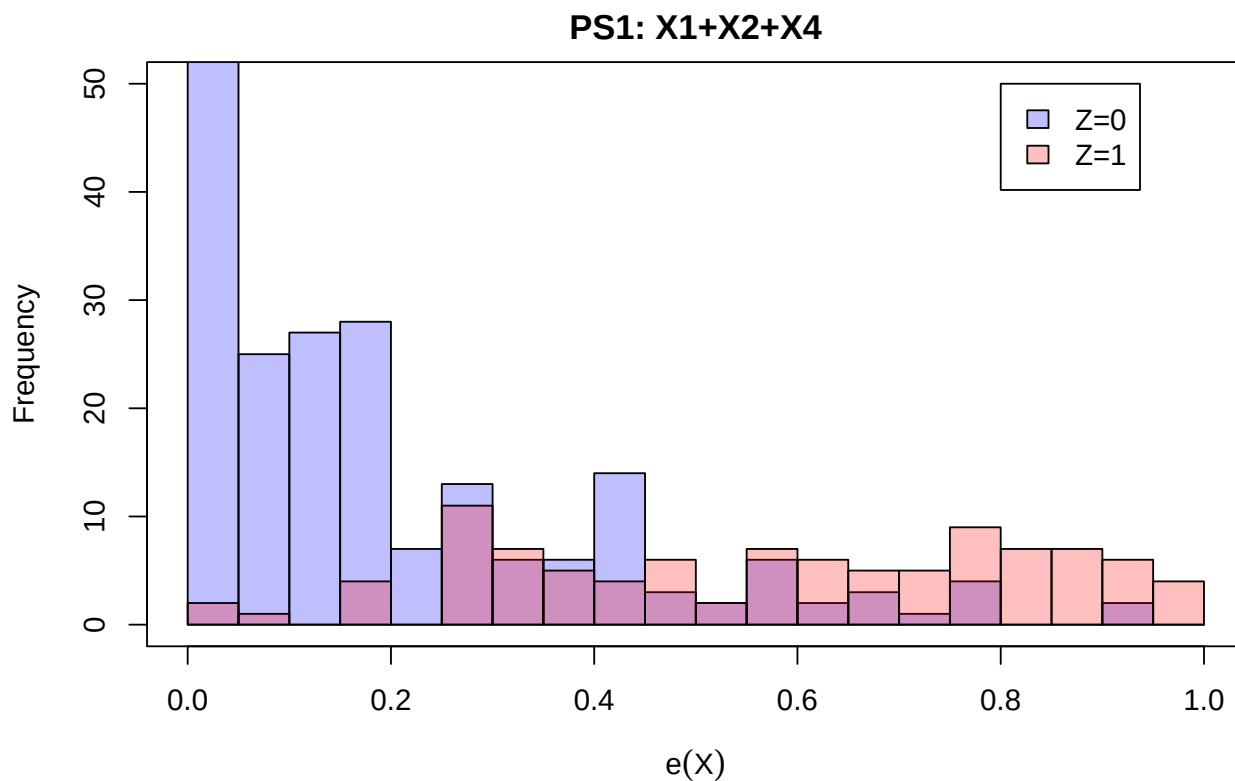
w01<-(1-Zv)/(1-ps1);W01<-w01/sum(w01)
w11<-Zv/ps1;W11<-w11/sum(w11)
w02<-(1-Zv)/(1-ps2);W02<-w02/sum(w02)
w12<-Zv/ps2;W12<-w12/sum(w12)

ate1.tilde<-(mean(w11*(Yv-mufit1))+mean(mufit1))-(mean(w01*(Yv-mufit0))+mean(mufit0))
ate1.hat<-(sum(W11*(Yv-mufit1))+mean(mufit1))-(sum(W01*(Yv-mufit0))+mean(mufit0))
ate2.tilde<-(mean(w12*(Yv-mufit1))+mean(mufit1))-(mean(w02*(Yv-mufit0))+mean(mufit0))
ate2.hat<-(sum(W12*(Yv-mufit1))+mean(mufit1))-(sum(W02*(Yv-mufit0))+mean(mufit0))

est.vec<-c(ate1.tilde,ate1.hat,ate2.tilde,ate2.hat)
est.vec

+ [1] -3.0514670  0.8682885  2.6169097  2.5533992
```

We can see that the estimates are quite different: we can examine the different propensity score distributions to try to explain this.



There appears to be an issue with overlap and positivity for the PS model that includes X_1 . For the equivalent propensity score regression model:

```
fit.psr1<-lm(Y~Z+ps1,data=P2.data)
summary(fit.psr1)$coef[,1:3]
```

```

+           Estimate Std. Error    t value
+ (Intercept) -5.340074  0.3154067 -16.930757
+ Z           2.062980  0.5608956  3.678011
+ ps1         7.026789  0.9288194  7.565291

```

```

fit.psr2<-lm(Y~Z+ps2,data=P2.data)
summary(fit.psr2)$coef[,1:3]

```

```

+           Estimate Std. Error    t value
+ (Intercept) -6.547967  0.3416337 -19.166632
+ Z           2.384850  0.4633545  5.146923
+ ps2         10.402548  0.9787652  10.628236

```

Because the PSR estimator is less sensitive to propensity score values near zero, the issue is somewhat mitigated using propensity score regression. Alternatively can alleviate the positivity issue for AIPW by including X_1 in the outcome model: this also blocks the confounding path that is left open if we do not include X_1 in the PS model. We combine these estimators in the following bootstrap analysis.

```

set.seed(234)
ests.mat.AIPW2<-matrix(0,nrow=N,ncol=8)
for(ib in 1:N){
  ind<-sample(1:n,size=n,rep=T)
  P2.boot<-P2.data[ind,]
  Zv<-P2.boot$Z
  Yv<-P2.boot$Y

  ps1<-fitted(glm(Z~X1+X2+X4,family=binomial,data=P2.boot))
  ps2<-fitted(glm(Z~X2+X4,family=binomial,data=P2.boot))

  fit.out1<-lm(Y~Z,data=P2.boot)
  fit.out2<-lm(Y~Z*X1,data=P2.boot)

  psr1.est<-coef(lm(Yv~Zv+ps1))[2]
  psr2.est<-coef(lm(Yv~Zv+ps2))[2]

  new0<-P2.boot;new0$Z<-0
  new1<-P2.boot;new1$Z<-1
  mufit01<-predict(fit.out1,new0);m01<-mean(mufit01)
  mufit11<-predict(fit.out1,new1);m11<-mean(mufit11)
  mufit02<-predict(fit.out2,new0);m02<-mean(mufit02)
  mufit12<-predict(fit.out2,new1);m12<-mean(mufit12)

  w01<-(1-Zv)/(1-ps1);W01<-w01/sum(w01)
  w11<-Zv/ps1;W11<-w11/sum(w11)
  w02<-(1-Zv)/(1-ps2);W02<-w02/sum(w01)
  w12<-Zv/ps2;W12<-w12/sum(w12)

  ate11.tilde<-(mean(w11*(Yv-mufit11))+m11)-(mean(w01*(Yv-mufit01))+m01)
  ate11.hat<-(sum(W11*(Yv-mufit11))+m11)-(sum(W01*(Yv-mufit01))+m01)
  ate12.tilde<-(mean(w12*(Yv-mufit11))+m11)-(mean(w02*(Yv-mufit01))+m01)
  ate12.hat<-(sum(W12*(Yv-mufit11))+m11)-(sum(W02*(Yv-mufit01))+m01)

  ate22.tilde<-(mean(w12*(Yv-mufit12))+m12)-(mean(w02*(Yv-mufit02))+m02)
  ate22.hat<-(sum(W12*(Yv-mufit12))+m12)-(sum(W02*(Yv-mufit02))+m02)

  est.vec<-c(psr1.est,psr2.est,ate11.tilde,ate11.hat,ate12.tilde,ate12.hat,
             ate22.tilde,ate22.hat)

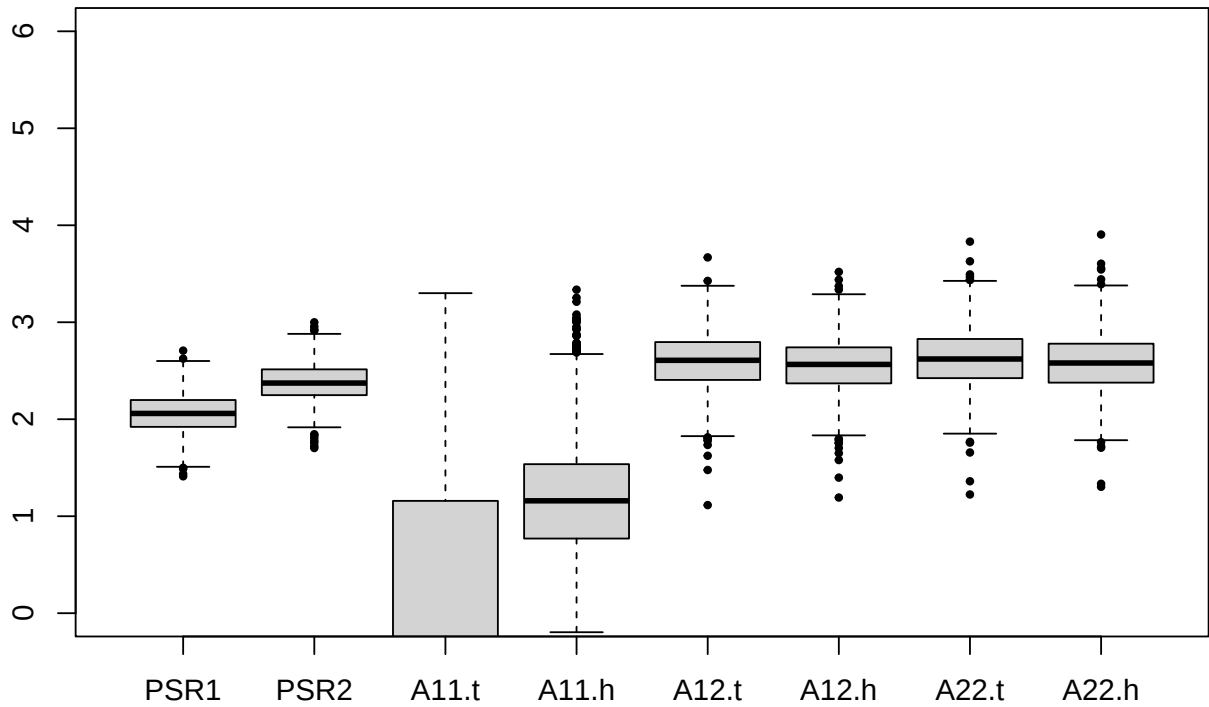
  ests.mat.AIPW2[ib,]<-est.vec
}

```

```

esum<-rbind(apply(ests.mat.AIPW2,2,mean),n*apply(ests.mat.AIPW2,2,var))
enames<-c('PSR1','PSR2','A11.t','A11.h','A12.t','A12.h','A22.t','A22.h')
colnames(esum)<-enames
rownames(esum)<-c('Est','n.Var')
par(mar=c(4,3,2,0))
boxplot(ests.mat.AIPW2,ylim=range(0,6),pch=19,cex=0.5,names=enames)

```



These eight estimators are as follows

Name	Method	Outcome	PS
PSR1	PSR	$Z + e(X)$	$X_1 + X_2 + X_4$
PSR2	PSR	$Z + e(X)$	$X_2 + X_4$
A11.t	AIPW $\tilde{\mu}$	Z	$X_1 + X_2 + X_4$
A11.h	AIPW $\hat{\mu}$	Z	$X_1 + X_2 + X_4$
A12.t	AIPW $\tilde{\mu}$	Z	$X_2 + X_4$
A12.h	AIPW $\hat{\mu}$	Z	$X_2 + X_4$
A22.t	AIPW $\tilde{\mu}$	$Z + X_1 + Z.X_1$	$X_2 + X_4$
A22.h	AIPW $\hat{\mu}$	$Z + X_1 + Z.X_1$	$X_2 + X_4$

```
round(esum,4)
```

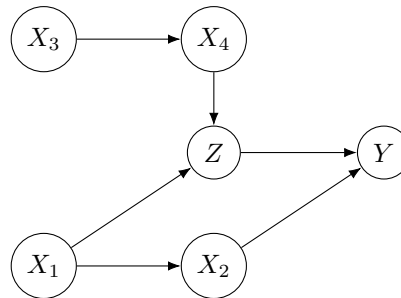
```

+      PSR1  PSR2    A11.t  A11.h  A12.t  A12.h  A22.t  A22.h
+ Est    2.0575  2.3743  -3.1806  1.2095  2.5995  2.5532  2.6244  2.5769
+ n.Var 11.8386 11.2850 18241.1632 127.4280 25.3449 24.6242 29.5453 28.7498

```

It is clear that the issues caused by extreme values in the propensity score can be overcome. The resulting estimated values are comparable.

3. Suppose finally that the structural DAG is believed to take the following form:



Explain how (if at all) the analysis should change from Q1; if the analysis changes, perform the new analysis using one of the methods from Q1; if it does not need to change, explain why not.

4 Marks

Solution: For this DAG,

- there is one open confounding path between Z and Y via X_1 and X_2 ;
- this path can be blocked by either conditioning on X_1 or X_2 or both, or by constructing the propensity score from X_1 alone, and performing propensity score adjustment;
- if we believe this structural DAG, then the simplest model involves regressing Y on X_2 and Z ; this analysis should give an unbiased estimate;
- because the propensity score is constructed only from X_1 , it is not clear whether propensity adjustment is warranted.

To assess this, we can perform five analyses

- OR1: The outcome regression $Z + X_1$.
- OR2: The outcome regression $Z + X_2$
- OR12: The outcome regression $Z + X_1 + X_2$
- PSR1: The propensity score regression $Z + e(X_1)$
- PSR2: The propensity score regression $Z + e(X_2)$

```
ps1<-fitted(glm(Z~X1,family=binomial,data=P2.data))
ps2<-fitted(glm(Z~X2,family=binomial,data=P2.data))
```

```
fit.or1<-lm(Y~Z+X1,data=P2.data)
coef(summary(fit.or1))[,1:3]
```

```
+           Estimate Std. Error  t value
+ (Intercept) -3.6864408  0.3331051 -11.066901
+ Z           4.8976581  0.5131820   9.543706
+ X1          -0.2867274  0.2504059  -1.145051
```

```
fit.or2<-lm(Y~Z+X2,data=P2.data)
coef(summary(fit.or2))[,1:3]
```

```
+           Estimate Std. Error  t value
+ (Intercept) 4.737742  0.20129703 23.53607
+ Z           2.363256  0.16865244 14.01258
+ X2           3.914746  0.08103037 48.31209
```

```
fit.or3<-lm(Y~Z+X1+X2,data=P2.data)
coef(summary(fit.or3))[,1:3]
```

```

+           Estimate Std. Error    t value
+ (Intercept)  4.76808878  0.20845143  22.8738597
+ Z            2.39977455  0.18060952  13.2870881
+ X1          -0.04815892  0.08455819  -0.5695358
+ X2           3.91203406  0.08126236  48.1407857

```

```

fit.psr1<-lm(Y~Z+ps1,data=P2.data)
coef(summary(fit.psr1))[1:3]

```

```

+           Estimate Std. Error    t value
+ (Intercept) -3.500247  0.4894735 -7.1510460
+ Z            4.882610  0.5180116  9.4256761
+ ps1          -1.424965  1.4346738 -0.9932325

```

```

fit.psr2<-lm(Y~Z+ps2,data=P2.data)
coef(summary(fit.psr2))[1:3]

```

```

+           Estimate Std. Error    t value
+ (Intercept) -12.36555  0.2022269 -61.14692
+ Z            2.40173  0.1716923  13.98857
+ ps2          28.19460  0.5962631  47.28550

```

The results obtained from models that do not include X_2 are different from those models that do include it. We verify using a bootstrap analysis.

```

#Bootstrap
set.seed(234)
ests.mat.PSR2<-matrix(0,nrow=N,ncol=5)
for(ib in 1:N){

  ind<-sample(1:n,size=n,rep=T)
  P2.boot<-P2.data[ind,]

  ps1<-fitted(glm(Z~X1,family=binomial,data=P2.boot))
  ps2<-fitted(glm(Z~X2,family=binomial,data=P2.boot))

  fit.or1<-lm(Y~Z+X1,data=P2.boot)
  fit.or2<-lm(Y~Z+X2,data=P2.boot)
  fit.or3<-lm(Y~Z+X1+X2,data=P2.boot)
  fit.psr1<-lm(Y~Z+ps1,data=P2.boot)
  fit.psr2<-lm(Y~Z+ps2,data=P2.boot)

  est.vec<-c(coef(fit.or1)[2],coef(fit.or2)[2],coef(fit.or3)[2],
             coef(fit.psr1)[2],coef(fit.psr2)[2])

  ests.mat.PSR2[ib,]<-est.vec
}

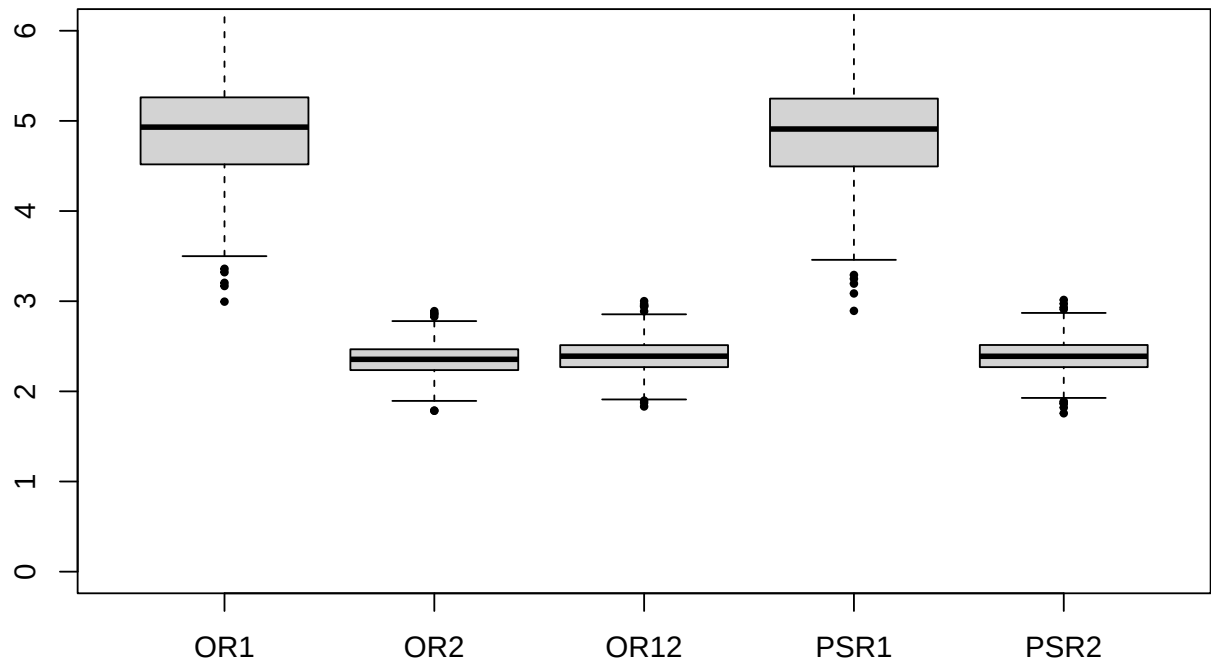
```

```

esum<-rbind(apply(ests.mat.PSR2,2,mean),n*apply(ests.mat.PSR2,2,var))
enames<-c('OR1','OR2','OR12','PSR1','PSR2')
colnames(esum)<-enames
rownames(esum)<-c('Est','n.Var')

par(mar=c(4,3,2,0))
boxplot(ests.mat.PSR2,ylim=range(0,6),pch=19,cex=0.5,names=enames)

```



```
round(esum,4)
```

```
+      OR1   OR2   OR12   PSR1   PSR2
+ Est    4.9065 2.3532 2.3895 4.8887 2.3908
+ n.Var 88.0620 8.6872 10.2747 90.2506 9.8126
```

The differences in the results indicate that the DAG may well not reflect the data generating mechanism, either because the structural DAG is not correct (for example, X_2 may be a direct cause of Z) or because there may be unmeasured confounding.