Journal Club - Meet 4

Texto motivado pelas notas de aula da disciplina de Causalidade ministrada pelo Prof David Stephens (McGill University)

Widemberg Nobre

22/10/2021

Métodos de Ajuste Causal Utilizando o Propensity Score

Considere o cenário onde o objetivo é estimar o efeito causal de Z em Y na presença de variáveis de confundimento. Lembremos que o efeito causal deve ser calculado com base no ambiente manipulável. Logo, considere que $\mathcal{E}$ faça referência ao ambiente experimental sob manipulação do pesquisador, e que o valor esperado de uma variável aleatória sob a medida experimental seja definido como

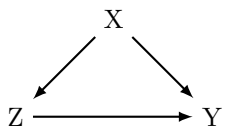
$$\mu(\mathrm{z}) = E^{\mathcal{E}}_{Y|Z}(Y|Z = \mathrm{z}).$$

O propensity score para tratamento binário, denotado por $e(\mathrm{X})$, é definido através do modelo observacional (ou seja, não é derivado do ambiente manipulável) de Z dado X, como

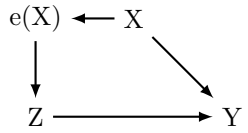
$$e(X) = f^{\mathcal{O}}_{Z|X}(1|X) = Pr^{\mathcal{O}}_{Z|X}(Z = 1|X = x).$$

O propensity score é um balancing score, isto é, $Z \perp X|e(X)$ (lê-se Z é independente de X condicional a $e(X)$. Obs: isso não significa que Z e X são MARGINALMENTE independentes!).

DAG sem o propensity score



DAG com o propensity score



Note que uma análise condionada no propensity score bloqueia a path que induz viés (backdoor path ou confounding path) entre Z e Y.

Existem quatro métodos (gerais) baseados no propensity score que podem ser usados em ajustes causais:

- Stratification: O espaço do propensity score (intervalo de 0 à 1) é particionado em J estratos, e o efeito causal "local" é estimado para cada estrato. Esses então são combinados como uma média (ponderada) visando estimar $\mu(\mathrm{z})$;
- Matching: Outcomes com diferentes valores de Z, mas "iguais" valores do propensity score são comparados;
- Regressão: O propensity score é usado como covariável no modelo de regressão;
- Inverse Weighting: Usando uma abordagem ponderada de "importance sampling", em função do propensity score estimado, cria-se uma pseudo-amostra onde a atribuição do tratamento é balanceada, ou seja, não há confudimento.

Consideremos o seguinte exemplo de geração de dados:

- $X \sim \text{Normal}_3(\mu, \Sigma)$ com

$$\mu = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 0.25 & 0.00 & 0.00 \\ 0.00 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.75 \end{bmatrix} \begin{bmatrix} 1.0 & 0.9 & -0.1 \\ 0.9 & 1.0 & -0.2 \\ -0.1 & -0.2 & 1.0 \end{bmatrix} \begin{bmatrix} 0.25 & 0.00 & 0.00 \\ 0.00 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.75 \end{bmatrix}$$

- $Z \mid X = x \sim \text{Bernoulli}(e(x; \alpha))$, com

$$e(x; \alpha) = \frac{\exp\{\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_1 x_2\}}{1 + \exp\{\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_1 x_2\}}$$

sendo $\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)^\top = (6.0, -0.2, 0.7, 2)^\top$.

- $Y \mid X = x, Z = z \sim \text{Normal}\left(\mu(x, z; \beta, \psi), \sigma^2\right)$, onde

$$\mu(x, z) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + z\left(\psi_0 + \psi_1 x_1 + \psi_2 x_2 + \psi_3 x_1 x_2\right) = \mu_0(x; \beta) + z\mu_1(x; \psi).$$

Completando a especificação do modelo gerador, suponhamos que $\sigma^2 = 0.1^2$, e

$$\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^\top = (10, -2.0, 1.2, 0.6)^\top \quad \psi = (\psi_0, \psi_1, \psi_2, \psi_3)^\top = (1, 1, 1, 1)^\top$$

Nesse modelo X_1 e X_2 são confundidores (afetam a geração de Z e Y, simultaneamente), e X_3 é uma variável espúria, pois não afeta a geração de ambos, Y e Z.

```
#Data simulation
library(MASS)
set.seed(23987)
n<-1000
D<-diag(c(0.25,0.5,0.75))
Mu<-c(1,-2,-1)
Sigma<-D %*% (matrix(c(1,0.9,-0.1,0.9,1,-0.2,-0.1,-0.2,1),3,3) %*% D)
X <- mvrnorm(n,mu=Mu,Sigma)
al <- c(6.0,-0.2,0.7,2)
Xal <- cbind(1,X[,1],X[,2],X[,1]*X[,2])
expit <- function(x){return(1/(1+exp(-x)))}
ps.true <- expit(Xal %*% al)
Z<-rbinom(n,1,ps.true)
be<-c(10,-2.0,1.2,0.6)
Xb<-cbind(1,X[,1],X[,2],X[,1]*X[,2])
mu0<- Xb %*% be
psi<-c(1,1,1,1)
Xp<-cbind(1,X[,1],X[,2],X[,1]*X[,2])
mu1<- (Xp %*% psi)
eta<-mu0 + Z * mu1 ; sig<-0.1
Y<-rnorm(n,eta,sig)
X1<-X[,1];X2<-X[,2];X3<-X[,3]
true.vals<-c(be,psi)[c(1,2,3,5,4,6,7,8)]
eX<-as.vector(Z-ps.true)
```

Aqui, temos que o average causal effect (ACE) pode ser analiticamente calculado, uma vez que:

$$\mu(z) = \mathbb{E}^{\mathcal{E}}_{Y|Z}[Y \mid Z = z] \equiv \mathbb{E}^{\mathcal{O}}_{X}\left[\mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y \mid X, Z = z]\right] = \mathbb{E}^{\mathcal{O}}_{X}[\mu(X, z)].$$

Logo,

$$\begin{aligned}\mu(z) &= \mathbb{E}_X^{\mathcal{O}}\left[\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + z\left(\psi_0 + \psi_1 X_1 + \psi_2 X_2 + \psi_3 X_1 X_2\right)\right] \\ &= \beta_0 + \beta_1 \mathbb{E}\left[X_1\right] + \beta_2 \mathbb{E}\left[X_2\right] + \beta_3 \mathbb{E}\left[X_1 X_2\right] + z\left(\psi_0 + \psi_1 \mathbb{E}\left[X_1\right] + \psi_2 \mathbb{E}\left[X_2\right] + \psi_3 \mathbb{E}\left[X_1 X_2\right]\right),\end{aligned}$$

onde

$$\mathbb{E}\left[XX^\top\right] = \begin{bmatrix} \mathbb{E}\left[X_1^2\right] & \mathbb{E}\left[X_1 X_2\right] & \mathbb{E}\left[X_1 X_3\right] \\ \mathbb{E}\left[X_1 X_2\right] & \mathbb{E}\left[X_2^2\right] & \mathbb{E}\left[X_2 X_3\right] \\ \mathbb{E}\left[X_1 X_3\right] & \mathbb{E}\left[X_2 X_2\right] & \mathbb{E}\left[X_3^2\right] \end{bmatrix} = \Sigma + \mu\mu^\top = \begin{bmatrix} 0.0625 & 0.1125 & -0.01875 \\ 0.1125 & 0.25 & -0.075 \\ -0.01875 & -0.075 & 0.5625 \end{bmatrix} + \begin{bmatrix} 1 & -2 & -1 \\ -2 & 4 & 2 \\ -1 & 2 & 1 \end{bmatrix}$$

o que nos leva a:

```
(X.M<-Sigma + Mu %*% t(Mu))
```

```
##          [,1]    [,2]     [,3]
## [1,]  1.06250 -1.8875 -1.01875
## [2,] -1.88750  4.2500  1.92500
## [3,] -1.01875  1.9250  1.56250
```

e portanto $\mu(0)$ e $\mu(1)$ são

```
(mu0<-sum(be*c(1,Mu[1],Mu[2],X.M[1,2])))
```

```
## [1] 4.4675
```

```
(mu1<-mu0 + sum(psi*c(1,Mu[1],Mu[2],X.M[1,2])))
```

```
## [1] 2.58
```

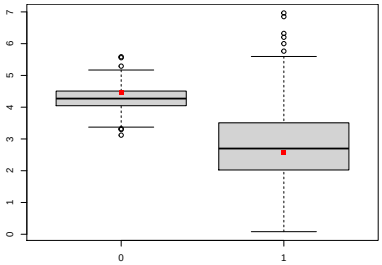
e então o valor verdadeiro do ACE é dado por

$$\mu(1) - \mu(0) = 2.58 - 4.4675 = -1.8875.$$

Análise sem ajuste para confundimento

Quando ajustamos somente Y como função de Z, inferência sobre o ACE será viesada.

```
par(mar=c(2,2,0,0))
boxplot(Y~Z)
points(1:2,c(mu0,mu1),col='red',pch=15)
```



```
c(mean(Y[Z==0]),mean(Y[Z==1]),mean(Y[Z==1])-mean(Y[Z==0]))
```

```
## [1]  4.288448  2.770900 -1.517548
```

Neste cenário, o efeito causal estimado é dado por -1.5175483.

Outcome Regression

Correta especificação

Uma correta especificação do modelo resposta fornece uma correta estimação do ACE.

```
fit0<-lm(Y~X1+X2+X1:X2+Z+Z:X1+Z:X2+Z:X1:X2)
round(cbind(coef(summary(fit0)),true.vals),6)
```

```
##              Estimate Std. Error    t value Pr(>|t|) true.vals
## (Intercept) 10.059622   0.144308  69.709475        0      10.0
## X1          -2.064657   0.115836 -17.823956        0      -2.0
## X2           1.236939   0.054910  22.526689        0       1.2
## Z            0.989901   0.168050   5.890520        0       1.0
## X1:X2        0.559755   0.050561  11.070900        0       0.6
## X1:Z         1.024435   0.130322   7.860769        0       1.0
## X2:Z         0.984275   0.062645  15.711978        0       1.0
## X1:X2:Z      1.023784   0.056071  18.258715        0       1.0
```

Com base nessa estrutura, a estimação do efeito causal pode ser feita através da predição média

$$\widehat{\mu}(z) = \frac{1}{n}\sum_{i=1}^{n} \mu\left(x_i, z; \widehat{\beta}, \widehat{\psi}\right)$$

```
mu0.0<-mean(predict(fit0,newdata=data.frame(X1,X2,Z=0)))
mu1.0<-mean(predict(fit0,newdata=data.frame(X1,X2,Z=1)))
c(mu0.0,mu1.0,mu1.0-mu0.0)
```

```
## [1]  4.454317  2.539963 -1.914354
```

Note, entretanto, que não estamos estimando o ACE com base nos valores observados de z.

$$\frac{\sum_{i=1}^{n} z_i \mu\left(x_i, 1; \widehat{\beta}, \widehat{\psi}\right)}{\sum_{i=1}^{n} z_i} \quad \text{e} \quad \frac{\sum_{i=1}^{n} (1-z_i)\, \mu\left(x_i, 0; \widehat{\beta}, \widehat{\psi}\right)}{\sum_{i=1}^{n} (1-z_i)}$$

Tal procedimento de estimação nos levaria a uma estimativa viesada.

```
m0<-sum((1-Z)*predict(fit0))/sum(1-Z)
m1<-sum(Z*predict(fit0))/sum(Z)
c(m0,m1,m1-m0)
```

```
## [1]  4.288448  2.770900 -1.517548
```

Isso ocorre devido ao confundimento presente na estrutura de geração do modelo conjunto.

Especificação Incorreta

A especificação incorreta do modelo deve nos levar a uma estimação incorreta do efeito causal; embora, eventualmente, possa ocorrer de a estimação do ACE ser recuperada. Como exemplo, se os termos $X1 : X2$ e $X1 : X2 : Z$ forem omitidos, o ACE ainda será estimado razoavelmente bem.

```
fit1<-lm(Y~X1+X2+Z+Z:X1+Z:X2)
round(coef(summary(fit1)),6)
```

```
##              Estimate Std. Error    t value Pr(>|t|)
## (Intercept) 11.206903   0.233694  47.955492        0
## X1          -3.216690   0.118417 -27.164058        0
```

```
## X2           1.775659   0.059201  29.993931        0
## Z            3.109948   0.285001  10.912047        0
## X1:Z        -0.927392   0.145289  -6.383066        0
## X2:Z         2.018438   0.072729  27.752731        0
```

```
mu0.1<-mean(predict(fit1,newdata=data.frame(X1,X2,Z=0)))
mu1.1<-mean(predict(fit1,newdata=data.frame(X1,X2,Z=1)))
c(mu0.1,mu1.1,mu1.1-mu0.1)
```

```
## [1]  4.435704  2.565313 -1.870391
```

A omissão de X1 : X2 e Z : X1 : X2 não nos leva a estimativas ruins pois a magnitude do efeito dessas iterações é razoavelmente pequena. No cenário onde X_1 ou X_2 ou ambos é omitido, a estimativa do ACE é ruim.

```
fit2<-lm(Y~X1+Z+Z:X1)
mu0.2<-mean(predict(fit2,newdata=data.frame(X1,Z=0)))
mu1.2<-mean(predict(fit2,newdata=data.frame(X1,Z=1)))
c(mu0.2,mu1.2,mu1.2-mu0.2)
```

```
## [1]  4.286084  2.744744 -1.541341
```

```
fit3<-lm(Y~X2+Z+Z:X2)
mu0.3<-mean(predict(fit3,newdata=data.frame(X1,Z=0)))
mu1.3<-mean(predict(fit3,newdata=data.frame(X1,Z=1)))
c(mu0.3,mu1.3,mu1.3-mu0.3)
```

```
## [1]  4.331889  2.647514 -1.684375
```

```
fit4<-lm(Y~Z)
mu0.4<-mean(predict(fit4,newdata=data.frame(X1,Z=0)))
mu1.4<-mean(predict(fit4,newdata=data.frame(X1,Z=1)))
c(mu0.4,mu1.4,mu1.4-mu0.4)
```

```
## [1]  4.288448  2.770900 -1.517548
```

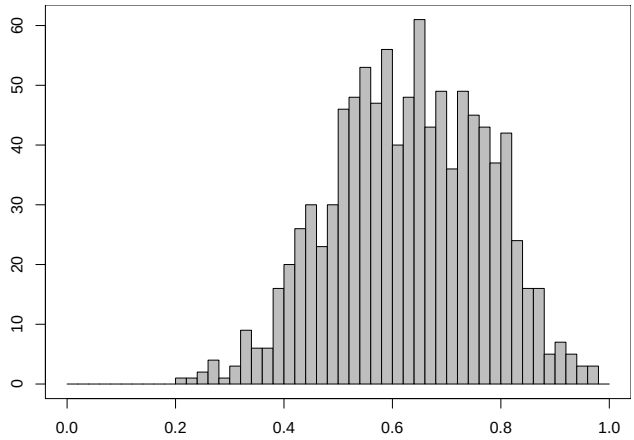
Propensity score stratification

No contexto de stratification, busca-se primeiro construir estratos dos valores do propensity score, $\mathcal{X}_j, j = 1, \ldots, J$. Então estima-se a quantidade causal de interesse dentro de cada estrato, para em seguida calcular o valor médio dos estratos.

Propensity Score verdadeiro

Inicialmente, assumamos os valores do propensity score como conhecidos (nós o conhecemos pois geramos os valores do propensity score ao simular os dados).

```
par(mar=c(2,2,0,0))
hist(ps.true,main="",col="gray",breaks=seq(0,1,by=0.02));box()
```



Deve-se definir os estratos tomando percentis do propensity score. Como exemplo, os quintis.

```
nstrata<-5
ps.quint.true<-quantile(ps.true,probs=seq(0,1,by=1/nstrata))
ps.quint.true
```

```
##        0%       20%       40%       60%       80%      100%
## 0.2126362 0.5120989 0.5914952 0.6715642 0.7601679 0.9762403
```

```
ps.strat.true<-as.numeric(cut(ps.true,ps.quint.true,include.lowest=TRUE))
table(Z,ps.strat.true)
```

```
##    ps.strat.true
## Z     1   2   3   4   5
##   0 120  95  73  46  29
##   1  80 105 127 154 171
```

Stratification pelos quintis do propensity score deve permitir um número razoável de observações com $Z = 0$ e $Z = 1$ em cada estrato (caso tal condição não seja satisfeita, devemos ter problemas de overlap nos dados).

```
mu.strat.true<-matrix(0,ncol=nstrata,nrow=3)
for(j in 1:nstrata){
mu.strat.true[1,j]<-mean(Y[Z==0 & ps.strat.true == j])
mu.strat.true[2,j]<-mean(Y[Z==1 & ps.strat.true == j])
mu.strat.true[3,j]<-mu.strat.true[2,j]-mu.strat.true[1,j]
}
mu.strat.true
```

```
##           [,1]      [,2]      [,3]      [,4]       [,5]
## [1,]  3.918955  4.234434  4.446315  4.682338  4.9721481
## [2,]  1.293308  1.929582  2.512729  2.960797  3.9994921
## [3,] -2.625647 -2.304852 -1.933586 -1.721540 -0.9726559
```

Podemos então estimar o ACE através da média dentro das linhas dessa matriz.

$$\widehat{\mu}(z) = \sum_{j=1}^{J} \widehat{\mu}^{\mathcal{X}_j}(z) \Pr\left[e(X) \in \mathcal{X}_j\right]$$

```
apply(mu.strat.true,1,mean)
```

```
## [1]  4.450838  2.539182 -1.911656
```

Ajuste do Propensity Score

Na prática, o o propensity score é descinhecido e, portanto, deve ser estimado através dos dados observados. Assumindo uma correta especificação do modelo do tratamento (a correta especificação não sugere que conheçamos os valores verdadeiros dos parâmetros envolvidos), o propensity score pode ser estimado através de uma regressão logística.

```
ps.fit.c<-fitted(glm(Z~X1+X2+X1:X2,family=binomial))
```

Uma vez estimado o propensity score, uma análise idêntica a anterior pode ser feita ao substituirmos o propensity score verdadeiro pelo ajustado.

```
ps.quint.fit.c<-quantile(ps.fit.c,probs=seq(0,1,by=1/nstrata))
ps.quint.fit.c
```

```
##        0%       20%       40%       60%       80%      100%
## 0.1718558 0.4995522 0.5948874 0.6877045 0.7888924 0.9928991
```

```
ps.strat.fit.c<-as.numeric(cut(ps.fit.c,ps.quint.fit.c,include.lowest=TRUE))
table(Z,ps.strat.fit.c)
```

```
##    ps.strat.fit.c
## Z     1   2   3   4   5
##   0 120  96  71  49  27
##   1  80 104 129 151 173
```

```
mu.strat.fit.c<-matrix(0,ncol=nstrata,nrow=3)
for(j in 1:nstrata){
mu.strat.fit.c[1,j]<-mean(Y[Z==0 & ps.strat.fit.c == j])
mu.strat.fit.c[2,j]<-mean(Y[Z==1 & ps.strat.fit.c == j])
mu.strat.fit.c[3,j]<-mu.strat.fit.c[2,j]-mu.strat.fit.c[1,j]
}
mu.strat.fit.c
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,]  3.924982  4.239986  4.438529  4.685589  4.960768
## [2,]  1.328789  1.929324  2.533836  2.970253  3.946459
## [3,] -2.596193 -2.310662 -1.904693 -1.715336 -1.014310
```

```
apply(mu.strat.fit.c,1,mean)
```

```
## [1]  4.449971  2.541732 -1.908239
```

Esses resultados são próximos daqueles observados quando consideramos o valor verdadeiro do propensity score. Contudo, se o modelo do tratamento é mal especificado, o mesmo não acontece. Como exemplo, omitindo X_2, obtemos uma estimação incorreta do ACE.

```
ps.fit.i<-fitted(glm(Z~X1,family=binomial))
ps.quint.fit.i<-quantile(ps.fit.i,probs=seq(0,1,by=1/nstrata))
ps.quint.fit.i
```

```
##        0%       20%       40%       60%       80%      100%
## 0.5576018 0.6141654 0.6313289 0.6432823 0.6594991 0.7163560
```

```
ps.strat.fit.i<-as.numeric(cut(ps.fit.i,ps.quint.fit.i,include.lowest=TRUE))
table(Z,ps.strat.fit.i)
```

```
##    ps.strat.fit.i
## Z     1   2   3   4   5
##   0  70  83  85  75  50
```

```
##   1 130 117 115 125 150
```

```
mu.strat.fit.i<-matrix(0,ncol=nstrata,nrow=3)
for(j in 1:nstrata){
mu.strat.fit.i[1,j]<-mean(Y[Z==0 & ps.strat.fit.i == j])
mu.strat.fit.i[2,j]<-mean(Y[Z==1 & ps.strat.fit.i == j])
mu.strat.fit.i[3,j]<-mu.strat.fit.i[2,j]-mu.strat.fit.i[1,j]
}
mu.strat.fit.i
```

```
##           [,1]      [,2]      [,3]     [,4]       [,5]
## [1,]  4.337441  4.285596  4.278122  4.27935  4.2557960
## [2,]  1.986344  2.266358  2.478530  3.02608  3.8558912
## [3,] -2.351097 -2.019238 -1.799591 -1.25327 -0.3999048
```

```
apply(mu.strat.fit.i,1,mean)
```

```
## [1]  4.287261  2.722641 -1.564620
```

Com uma má especificação do modelo do tratamento, Z and X não são independentes dentro de cada estrato do propensity score, e portanto a biasing path de Z para Y via X está desbloqueada. Esse entendimento é usualmente referido como falta de balanceamento entre as unidades amostrais.

A falta de balanceamento pode ser diagnosticada numericamente (comparando as médias dos confundidores para $Z = 0$ e $Z = 1$ dentro de cada estrato do propensity score) oru graficamente (comparando os boxplots dentro de cada strata).

Para o modelo correto, nós temos

```
t1<-aggregate(X1,by=list(Z,ps.strat.fit.c),FUN=mean)
t2<-aggregate(X2,by=list(Z,ps.strat.fit.c),FUN=mean)
t3<-aggregate(X3,by=list(Z,ps.strat.fit.c),FUN=mean)
dnames<-list(c("Z=0", "Z=1"),c("Q1", "Q2", "Q3","Q4","Q5"))
matrix(t1$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames ) #Summary for X1: mean
```

```
##            Q1        Q2       Q3        Q4       Q5
## Z=0 0.9978433 0.9316459 0.979086 0.9546425 1.101412
## Z=1 0.9675377 0.9354829 1.008106 0.9665335 1.095633
```

```
matrix(t2$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames ) #Summary for X2: mean
```

```
##            Q1        Q2        Q3        Q4        Q5
## Z=0 -2.282388 -2.231281 -2.049605 -1.931326 -1.578578
## Z=1 -2.335530 -2.220630 -1.985303 -1.929186 -1.583760
```

```
matrix(t3$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames ) #Summary for X3: mean
```

```
##             Q1         Q2         Q3        Q4        Q5
## Z=0 -0.8727500 -1.1261686 -0.9283224 -1.002825 -1.422133
## Z=1 -0.7485271 -0.7775068 -0.9098424 -1.093546 -1.086302
```

Como os valores médios, das observações referentes a $Z = 0$ e $Z = 1$, dentro de cada estrato são próximos para todas as variáveis de confundimento, temos uma indicação de que o propensity score estimado induz balanceamento.

Para o modelo incorreto, nós temos:

```
t1<-aggregate(X1,by=list(Z,ps.strat.fit.i),FUN=mean)
t2<-aggregate(X2,by=list(Z,ps.strat.fit.i),FUN=mean)
t3<-aggregate(X3,by=list(Z,ps.strat.fit.i),FUN=mean)
```

```
dnames<-list(c("Z=0", "Z=1"),c("Q1", "Q2", "Q3","Q4","Q5"))
matrix(t1$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames ) #Summary for X1: mean
```

```
##            Q1        Q2        Q3       Q4       Q5
## Z=0 0.6981254 0.8698683 0.9981973 1.114039 1.314847
## Z=1 0.6476249 0.8667821 0.9940135 1.120220 1.335241
```

```
matrix(t2$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames ) #Summary for X2: mean
```

```
##            Q1        Q2        Q3        Q4        Q5
## Z=0 -2.627302 -2.318033 -2.066489 -1.872057 -1.570087
## Z=1 -2.594987 -2.193335 -1.988481 -1.717672 -1.345579
```

```
matrix(t3$x,nrow=2,ncol=nstrata,byrow=FALSE,dimnames = dnames ) #Summary for X3: mean
```

```
##             Q1         Q2         Q3         Q4         Q5
## Z=0 -0.8686154 -0.9637137 -0.9579892 -1.2352245 -1.0285372
## Z=1 -0.9155792 -1.0579546 -0.8949457 -0.9421799 -0.9844728
```

e agora somente X_1 parece estar balanceado. Algo que sugere que o propensity score só pode induzir balanceamento para aquelas variáveis que aparecem no modelo. Pontuamos que variáveis que permanecem desbalanceadas podem ainda induzir viés devido a confundimento. A análise de balanceamento nos sugere que a estrutura funcional do propensity score está correta para aquela variável incluída.

Matching

De acordo com a teoria envolvendo a construção do propensity score, unidades com o mesmo valor do propensity score, mas diferentes valores do tratamento, são independentes. Neste contexto, a abordagem de Matching sugere usar o propensity score de modo a ‘’marcar” indivíduos comparáveis nos subconjuntos $Z = 0$ e $Z = 1$. Esses individuos ‘’marcados” podem então ser usados para estimar a diferença $\mu(1) - \mu(0)$. O pacote Matching fornece uma alternativa bem simples de se desenvolver uma análise estatística para esse contexto.

Na primeira análise, a marcação ocorre de forma 1-para-1 quando relacionado indivíduos do subgrupo referente a $Z = 0$ e $Z = 1$. Existem 637 casos com $Z = 1$ nos dados simulados, contudo o pacote reutiliza os dados de forma a produziruma amostra de tamanho idêntido ao tamanho amostral. Na função Match, a quantidade Tr denota o ‘tratamento’ Z, e X é a variável (ou variáveis) usada para o matching.

```
library(Matching)
```

```
## ##
## ##  Matching (Version 4.9-10, Build Date: 2021-09-08)
## ##  See http://sekhon.berkeley.edu/matching for additional documentation.
## ##  Please cite software as:
## ##   Jasjeet S. Sekhon. 2011. ``Multivariate and Propensity Score Matching
## ##   Software with Automated Balance Optimization: The Matching package for R.''
## ##   Journal of Statistical Software, 42(7): 1-52.
## ##
```

```
fit.match1 <- Match(Y=Y, Tr=Z, X=ps.true, M=1, estimand="ATE")
summary(fit.match1)
```

```
##
## Estimate...  -1.9235
## AI SE......  0.044062
## T-stat.....  -43.653
## p.val......  < 2.22e-16
##
```

```
## Original number of observations..............  1000
## Original number of treated obs...............  637
## Matched number of observations...............  1000
## Matched number of observations  (unweighted).  1288
```

Neste caso, a estimativa do ACE é -1.9234667, e o erro padrão estimado é de 0.0440625.

É possível também fazer uma marcação do tipo 1-para-M. Nesse tipo de análise, todo caso com $Z = 1$ é associado à M unidades no subgrupo $Z = 0$.

```
fit.match2 <- Match(Y=Y, Tr=Z, X=ps.true, M=5, estimand="ATE")
summary(fit.match2)
```

```
##
## Estimate...  -1.9258
## AI SE......  0.042978
## T-stat.....  -44.809
## p.val......  < 2.22e-16
##
## Original number of observations..............  1000
## Original number of treated obs...............  637
## Matched number of observations...............  1000
## Matched number of observations  (unweighted).  5091
```

```
fit.match3 <- Match(Y=Y, Tr=Z, X=ps.true, M=10, estimand="ATE")
summary(fit.match3)
```

```
##
## Estimate...  -1.9216
## AI SE......  0.042942
## T-stat.....  -44.749
## p.val......  < 2.22e-16
##
## Original number of observations..............  1000
## Original number of treated obs...............  637
## Matched number of observations...............  1000
## Matched number of observations  (unweighted).  10047
```

Existem muitas outras possiblidades no pacote Matching.

Regressão com o Propensity Score

Em regressão com o propensity score, um modelo é proposto considerando o propensity score como covariável. Por exemplo, nós podemos ajustar

$$\mu(z, x) = \mathbb{E}^{\mathcal{O}}_{Y|X,Z}[Y \mid X = x, Z = z] = \beta_0 + \psi_0 z + \gamma e(x)$$

```
fit.psr.1<-lm(Y~Z+ps.fit.c)
summary(fit.psr.1)
```

```
##
## Call:
## lm(formula = Y ~ Z + ps.fit.c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.59541 -0.26342  0.07312  0.31082  2.69878
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.58112    0.07007   22.57   <2e-16 ***
## Z           -2.05087    0.03772  -54.37   <2e-16 ***
## ps.fit.c     4.78344    0.11320   42.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5405 on 997 degrees of freedom
## Multiple R-squared:  0.7835, Adjusted R-squared:  0.7831
## F-statistic:  1804 on 2 and 997 DF,  p-value: < 2.2e-16
```

A estimativa derivada desse procedimento é o coeficiente da regressão de Z no modelo gerador dos dados. Nesse particular modelo, podemos ler a quantidade referente a $\mu(1) - \mu(0)$ como ψ_0, que é estimado como -2.0508704. Essa regressão sofre do problema de má especificação do modelo $\mu(\mathrm{x}, \mathrm{z})$. Em suma, se $\mu(x, z)$ é mal especificado, então uma incorreta inferência será fornecida.

O modelo pode ser extendido para incluir outros termos. Suponhamos o seguinte modelo

$$\mu(x, z) = \beta_0 + z\left(\psi_0 + \psi_1 x_1 + \psi_2 x_2 + \psi_3 x_1 x_2\right) + \gamma_0 e(x)$$

```
fit.psr.2<-lm(Y~Z+Z:X1+Z:X2+Z:X1:X2+ps.fit.c)
round(coef(summary(fit.psr.2)),6)
```

```
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept)  3.085174   0.027244 113.24310        0
## Z            3.964332   0.135035  29.35781        0
## ps.fit.c     2.126006   0.046459  45.76071        0
## Z:X1        -0.874477   0.083430 -10.48155        0
## Z:X2         1.894645   0.042690  44.38134        0
## Z:X1:X2      0.627663   0.039762  15.78543        0
```

```
mu0.psr<-mean(predict(fit.psr.2,newdata=data.frame(X1,X2,Z=0,ps.fit.c)))
mu1.psr<-mean(predict(fit.psr.2,newdata=data.frame(X1,X2,Z=1,ps.fit.c)))
c(mu0.psr,mu1.psr,mu1.psr-mu0.psr)
```

```
## [1]  4.439440  2.532145 -1.907295
```

Assim, uma correta estimatimação de $\mu(0)$ e $\mu(1)$ é obtida, embora ψ não seja estimado como no modelo gerador.

Comparemos esse resultado com o derivado do seguinte modelo mal especificado

$$\mu(x, z) = \beta_0 + z\left(\psi_0 + \psi_1 x_1 + \psi_2 x_2 + \psi_3 x_1 x_2\right) = m_0\left(\beta_0\right) + z m_1(x; \psi)$$

Assim, temos

```
fit.or2<-lm(Y~Z+Z:X1+Z:X2+Z:X1:X2)
round(coef(summary(fit.or2)),6)
```

```
##              Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  4.288448   0.012557 341.526494        0
## Z            6.761074   0.212130  31.872303        0
## Z:X1        -1.040222   0.146840  -7.084072        0
## Z:X2         2.221214   0.074149  29.956267        0
## Z:X1:X2      1.583540   0.059604  26.567677        0
```

```
mu0.or2<-mean(predict(fit.or2,newdata=data.frame(X1,X2,Z=0)))
mu1.or2<-mean(predict(fit.or2,newdata=data.frame(X1,X2,Z=1)))
c(mu0.or2,mu1.or2,mu1.or2-mu0.or2)
```

```
## [1]  4.288448  2.539963 -1.748485
```

que nos leva a incorreta inferência. Isso acontece porque $m_1(x;\psi) \equiv \mu_1(x;\psi)$ é corretamente especificado com base no modelo gerador. Contudo. $m_0\,(\beta_0)$ é especificado de forma incorreta, uma vez que no modelo gerador, temos $\mu_0(x;\beta) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$. Como β_0 em $m_0(.)$ precisa ser estimado, temos uma inferência incorreta.

Finalmente, suponhamos que haja iterações entre $e(x)$ e x_1, e x_2.

```
fit.psr.3<-lm(Y~Z+Z:X1+Z:X2+Z:X1:X2+ps.fit.c+ps.fit.c:X1+ps.fit.c:X2 + ps.fit.c:X1:X2)
round(coef(summary(fit.psr.3)),6)
```

```
##                 Estimate Std. Error    t value Pr(>|t|)
## (Intercept)     3.477510   0.028840 120.580161 0.000000
## Z               0.989859   0.202076   4.898455 0.000001
## ps.fit.c        6.918620   0.237115  29.178343 0.000000
## Z:X1            1.022784   0.157344   6.500303 0.000000
## Z:X2            0.982109   0.076041  12.915514 0.000000
## X1:ps.fit.c    -2.761434   0.175974 -15.692298 0.000000
## X2:ps.fit.c     1.441217   0.087193  16.528950 0.000000
## Z:X1:X2         1.027346   0.068479  15.002324 0.000000
## X1:X2:ps.fit.c -0.107657   0.077801  -1.383748 0.166747
```

```
mu0.psr3<-mean(predict(fit.psr.3,newdata=data.frame(X1,X2,Z=0,ps.fit.c)))
mu1.psr3<-mean(predict(fit.psr.3,newdata=data.frame(X1,X2,Z=1,ps.fit.c)))
c(mu0.psr3,mu1.psr3,mu1.psr3-mu0.psr3)
```

```
## [1]  4.456889  2.538435 -1.918454
```

Os valores estimados do ACE não mudam consideravelmente. Muito embora,

```
round(cbind(coef(summary(fit.psr.3))[c(2,4,5,8),],psi),6)
```

```
##         Estimate Std. Error   t value Pr(>|t|) psi
## Z       0.989859   0.202076  4.898455    1e-06   1
## Z:X1    1.022784   0.157344  6.500303    0e+00   1
## Z:X2    0.982109   0.076041 12.915514    0e+00   1
## Z:X1:X2 1.027346   0.068479 15.002324    0e+00   1
```

os valores de ψ sejam corretamente estimados.

Inverse Weighting

Agora vamos discutir a abordagem de inverse weighting (ou inverse probability weighting, IPW). Essa abordagem é baseada em um procedimento de importance sampling (ou mudança de medida).

$$\mu(z) = \mathbb{E}^{\mathcal{E}}_{Y|Z}[Y \mid Z = z] = \frac{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)Y}{f^{\mathcal{O}}_{Z|X}(Z|X)}\right]}{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)}{f^{\mathcal{O}}_{Z|X}(Z|X)}\right]}$$

Vamos provar esse resultado. A prova envolve o princípio de consistência do contexto causal. Tal principio sustenta que a resposta observada é definida como $Y = (1-Z)Y(0) + ZY(1)$. Logo, se $Z = 0$, então a

unidade observada é tal que $Y = Y(0)$, e se $Z = 1$, então, $Y = Y(1)$, onde $Y(0)$ e $Y(1)$ fazem referência ao ambiente manipulável.

Daí, temos

$$\begin{aligned}\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)Y}{f^{\mathcal{O}}_{Z|X}(Z \mid X)}\right] &= \mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)Y(z)}{f^{\mathcal{O}}_{Z|X}(Z \mid X)}\right] = E\left[\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)Y(z)}{f^{\mathcal{O}}_{Z|X}(Z \mid X)}\right]|X\right] \\ &= E\left[\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)}{f^{\mathcal{O}}_{Z|X}(Z \mid X)}\right]|X\right] E\left[\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[Y(z)\right]|X\right] \\ &= \mathbb{E}^{\mathcal{E}}_{Y|Z}\left[Y(z)\right]\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)}{f^{\mathcal{O}}_{Z|X}(Z \mid X)}\right] = \mathbb{E}^{\mathcal{E}}_{Y|Z}\left[Y|Z=z\right]\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)}{f^{\mathcal{O}}_{Z|X}(Z \mid X)}\right],\end{aligned}$$

e se isolarmos $\mathbb{E}^{\mathcal{E}}_{Y|Z}\left[Y|Z=z\right]$ teremos a identidade desejada.

No caso binário, temos

$$\mu(0) = \frac{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{(1-Z)Y}{1-e(X)}\right]}{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{(1-Z)}{1-e(X)}\right]} \quad \mu(1) = \frac{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{ZY}{e(X)}\right]}{\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{Z}{e(X)}\right]}$$

IPW

Pode-se deduzir que os estimadores adequados a esta abordage são tais como

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

que podem ser escritos como

$$\widehat{\mu}_{IPW}(0) = \frac{\sum_{i=1}^{n}W_{0i}Y_i}{\sum_{i=1}^{n}W_{0i}} = \sum_{i=1}^{n}W^*_{0i}Y_i \quad \widehat{\mu}_{IPW}(1) = \frac{\sum_{i=1}^{n}W_{1i}Y_i}{\sum_{i=1}^{n}W_{1i}} = \sum_{i=1}^{n}W^*_{1i}Y_i$$

onde

$$W_{0i} = \frac{(1-Z_i)}{1-e\left(X_i\right)} \quad W_{1i} = \frac{Z_i}{e\left(X_i\right)} \quad W^*_{zi} = \frac{W_{zi}}{\sum_{j=1}^{n}W_{zj}} \quad z = 0,1$$

Note ainda que por construção,

$$\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\widehat{\mu}_{IPW}(z)\right] = \mu(z) \quad z = 0,1$$

e, portanto, os estimadores são não-viesados. Além disso, temos

$$\begin{aligned}\mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[W_{zi}\right] = \mathbb{E}^{\mathcal{O}}_{X,Y,Z}\left[\frac{\mathbb{I}_{\{z\}}(Z)}{f^{\mathcal{O}}_{Z|X}(Z \mid X)}\right] &= \mathbb{E}^{\mathcal{O}}_{X}\left[\mathbb{E}^{\mathcal{O}}_{Z|X}\left[\mathbb{I}_{\{z\}}(Z)\frac{1}{f^{\mathcal{O}}_{Z|X}(Z \mid X)} \mid X\right]\right] \\ &= \mathbb{E}^{\mathcal{O}}_{X}\left[\mathbb{E}^{\mathcal{O}}_{Z|X}\left[f^{\mathcal{O}}_{Z|X}(Z \mid X)\frac{1}{f^{\mathcal{O}}_{Z|X}(Z \mid X)} \mid X\right]\right] = 1\end{aligned}$$

Esse resultado sugere um estimador IPW alternativo, onde substituimos o denominador por seu valor esperado (que é n), é tal que

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

Usando o propensity score ajustado, podemos fazer o seguinte cálculo: para $\widehat{\mu}(z)$, temos

```
W0<-(1-Z)/(1-ps.fit.c)
W0.star<-W0/sum(W0)
mu.hat0<-sum(W0.star*Y)
W1<-Z/ps.fit.c
W1.star<-W1/sum(W1)
mu.hat1<-sum(W1.star*Y)
c(mu.hat0,mu.hat1,mu.hat1-mu.hat0)
```

```
## [1]  4.488312  2.522956 -1.965355
```

e para $\widetilde{\mu}(z)$

```
mu.tilde0<-mean(W0*Y)
mu.tilde1<-mean(W1*Y)
c(mu.tilde0,mu.tilde1,mu.tilde1-mu.tilde0)
```

```
## [1]  4.578971  2.535878 -2.043093
```

Note que podemos recuperar $\widehat{\mu}(z)$ usando o weighted least squares (WLS) na regressão de Y em Z

$$\mathbb{E}^{\mathcal{O}}_{Y|Z}[Y \mid Z = z] = \beta_0 + \psi_0 z$$

com pesos

$$W_{0i} + W_{1i} = \frac{(1-Z_i)}{1-e\,(X_i)} + \frac{Z_i}{e\,(X_i)}$$

```
W<-W0+W1
fit.wls<-lm(Y~Z,weights=W)
round(coef(summary(fit.wls)),6)
```

```
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  4.488312   0.037154 120.8032        0
## Z           -1.965355   0.052740 -37.2648        0
```

Os coeficientes estimados são precisamente $\widehat{\mu}(0)$ e $\widehat{\mu}(1) - \widehat{\mu}(1)$. Esse resultado nos fornece ainda mais informação sobre como o IPW corrige o confundimento; ele cria uma pseudo-amostra a partir da distribuição da medida experimental $\mathcal{E}$, isto é:

- na amostra observada, obtida sob $\mathcal{O}$, cada ponto carrega um peso $1/n$.
- Para a pseudo amostra, desejamos que cada ponto tenha um peso $1/n$ sob $\mathcal{E}$, mas na verdade, a unidade i carrega um peso que é dependente de X_i, ou seja

$$f^{\mathcal{O}}_{Z|X}\,(z \mid x_i) = \{e\,(x_i)\}^{z}\,\{1-e\,(x_i)\}^{1-z} \quad z = 0, 1$$