NHANES Example

NHANES example

We study the publicly available (U.S.) National Health and Nutrition Examination Survey (NHANES).

- `BPSysAve` (systolic blood pressure, *outcome*),
- `SmokeNow` (smoking status: 0–No, 1–Yes, *exposure*),
- `Gender`,
- `Age`,
- `Race3`,
- `Education`,
- `MaritalStatus`, and
- `Poverty`

We restrict our attention to 1377 adults ($>$ 17 years old) in the second wave of the survey.

NHANES example

First the R packages are loaded

```
#Libraries and dependencies muse be pre-loaded in R
suppressPackageStartupMessages({
    library(NHANES)       #Contains the NHANES data
    library(tableone)     #For comparing the treated and untreated
    library(Matching)     #For comparison based on matching
    library(survey)       #For handling reweighted samples
    library(splines)      #For fitting a flexible PS model
    library(Hmisc)        #Some plotting functions
    library(htmlwidgets) #Some plotting functions
})
```

NHANES example

```
#Selecting the subset of data for analysis
small.nhanes <- na.omit(NHANES[NHANES$SurveyYr=="2011_12"
                      & NHANES$Age > 17,c(3,4,8:11,13,25,61)])
small.nhanes$SmokeNow <- as.numeric(small.nhanes$SmokeNow)-1

#dim(small.nhanes) ## 1377
vars <- c("Gender", "Age", "Race3", "Education", "MaritalStatus",
               "Poverty")
```

NHANES example

Regression:

```
fit0<-lm(BPSysAve~SmokeNow,data=small.nhanes)
round(coef(summary(fit0)),5)

+              Estimate Std. Error   t value Pr(>|t|)
+ (Intercept) 125.61381    0.63365 198.23993  0.00000
+ SmokeNow     -3.67936    0.96395  -3.81696  0.00014

fmod<-formula(BPSysAve~SmokeNow+Gender+Age+Race3+
                       Education+MaritalStatus+HHIncome+Poverty)
fit1<-lm(fmod,data=small.nhanes)
round(coef(summary(fit1))[1:2,],5)

+              Estimate Std. Error  t value Pr(>|t|)
+ (Intercept) 107.25006    4.75287 22.56534  0.00000
+ SmokeNow     -1.09777    0.93042 -1.17986  0.23826
```

Checking confounder balance

Typical ways of assessing balance include inspection of

- *Standardized mean difference* (SMD) or proportion:

$$\frac{\bar{x}^{1,w} - \bar{x}^{0,w}}{\sqrt{(v^{1,w} + v^{0,w})/2}} \qquad \text{where,}$$

  where, if w_i is a case weight, the quantities

$$\bar{x}^{z,w} \quad \text{and} \quad v^{z,w}$$

  are the weighted sample mean and sample variance of variable X among those with treatment value z; SMD < 0.1 indicates reasonable balance.

- Empirical CDFs in the treated and untreated groups.

Checking confounder balance

There is clear Age imbalance between the two groups.

```
tabUnmatched<-CreateTableOne(vars=vars[2],strata="SmokeNow",
                     data=small.nhanes,test=FALSE)
print(tabUnmatched, smd = TRUE)

+                   Stratified by SmokeNow
+                    0             1             SMD
+   n                  782           595
+   Age (mean (SD)) 54.33 (16.52) 44.96 (15.11)  0.592
```

`CreateTableOne` can be used for all predictors

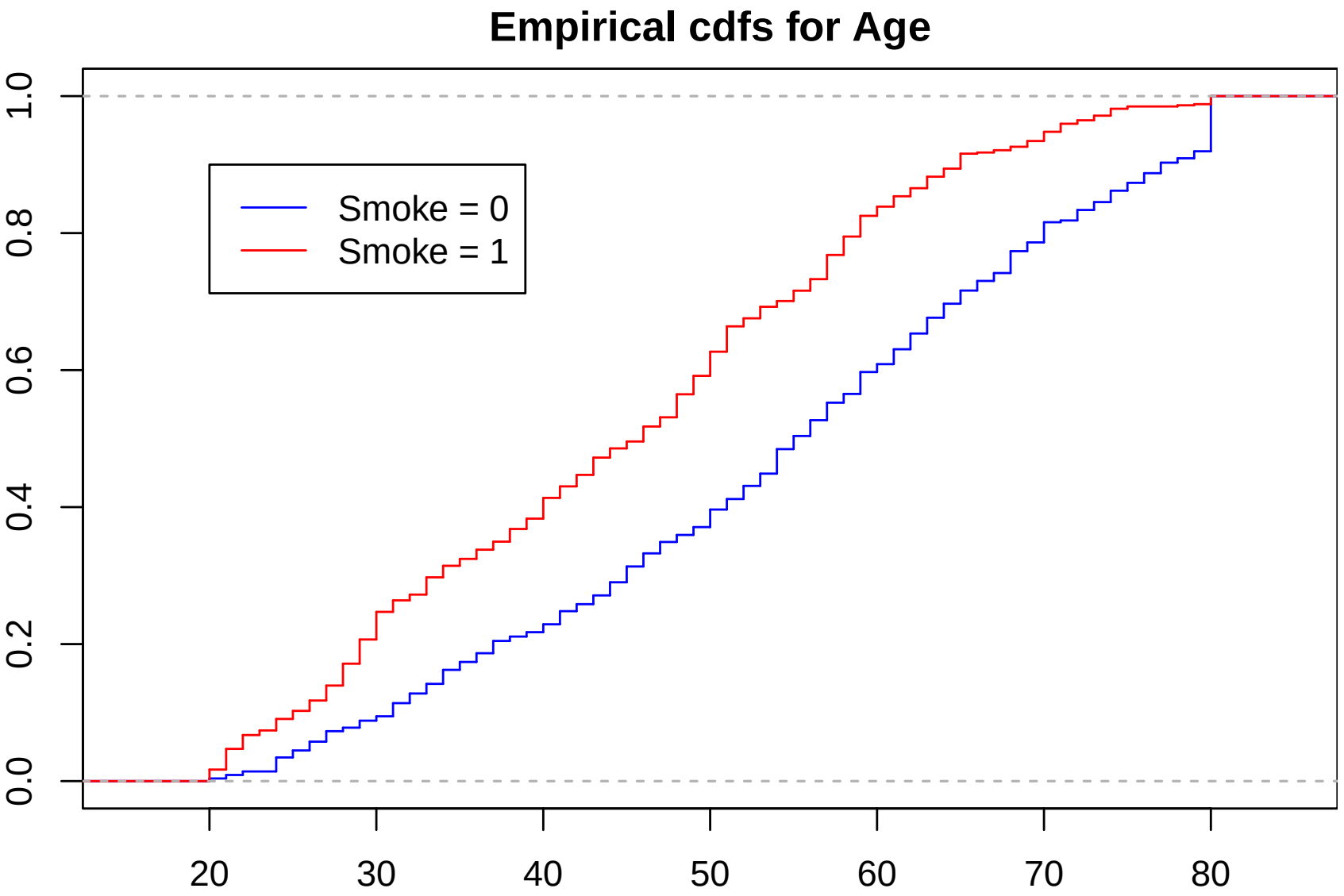
```
tabUnmatched<-CreateTableOne(vars=vars,strata="SmokeNow",
                     data=small.nhanes,test=FALSE)
```

Checking confounder balance

```
Smoke<-small.nhanes$SmokeNow
age0<-small.nhanes$Age[Smoke==0]
age1<-small.nhanes$Age[Smoke==1]
ecdf0 <- ecdf(age0); ecdf1 <- ecdf(age1)
```

```
par(mar=c(2,2,2,0))
plot(ecdf0, verticals=TRUE, do.points=FALSE,
        main='Empirical cdfs for Age',col='blue')
plot(ecdf1, verticals=TRUE, do.points=FALSE, add=TRUE, col='red')
legend(20,0.9,c('Smoke = 0', 'Smoke = 1'),col=c('blue','red'),lty=1)
```

Checking confounder balance



Building the propensity score

We attempt to build and assess a propensity score model using the available covariates

```
#Fit the PS using logistic regression
fmods<-formula(SmokeNow~Gender+Age+Race3+Education+
                          MaritalStatus+HHIncome+Poverty)
ps.mod <- glm(fmods,data=small.nhanes,family="binomial")
ps.lr <- predict(ps.mod,type="response")
```

Building the propensity score

Investigating the PS quintiles:

```
ps0<-ps.lr[Smoke==0];ps1<-ps.lr[Smoke==1]
quints <- c(0,quantile(ps.lr,seq(.2,1,.2)))
ps.tab<-rbind(table(cut(ps.lr[Smoke==0],quints)),
              table(cut(ps.lr[Smoke==1],quints)))
rownames(ps.tab)<-c('Smoke=0','Smoke=1')
ps.tab
```

```
+          (0,0.222] (0.222,0.34] (0.34,0.481] (0.481,0.639] (0.639,0.941]
+ Smoke=0        231           194           167           121            69
+ Smoke=1         47            82           105           157           204
```

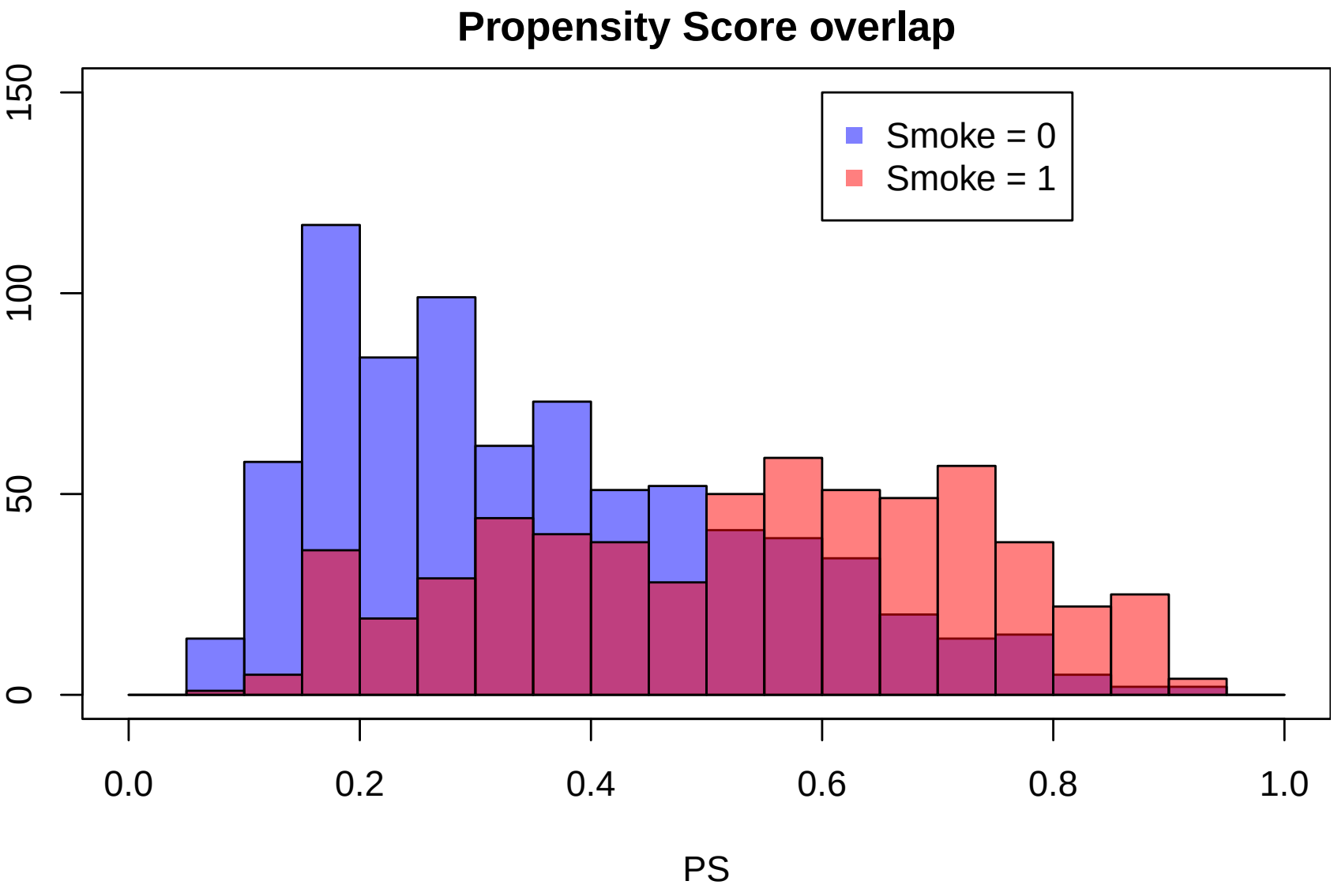
The table indicates that there is good representation of the exposed and unexposed groups in the five quintile-based strata.

Building the propensity score

We look for overlap of the propensity score histograms.

```
par(mar=c(4,2,2,0))
colvec<-c(rgb(0,0,1,0.5),rgb(1,0,0,0.5))
hist(ps0, col=rgb(0,0,1,0.5), breaks=seq(0,1,by=0.05), ylim=c(0,150),
        main="Propensity Score overlap", xlab="PS")
hist(ps1, col=rgb(1,0,0,0.5), breaks=seq(0,1,by=0.05), add=T);box()
legend(0.6,150,c('Smoke = 0', 'Smoke = 1'),col=colvec,pch=15)
```

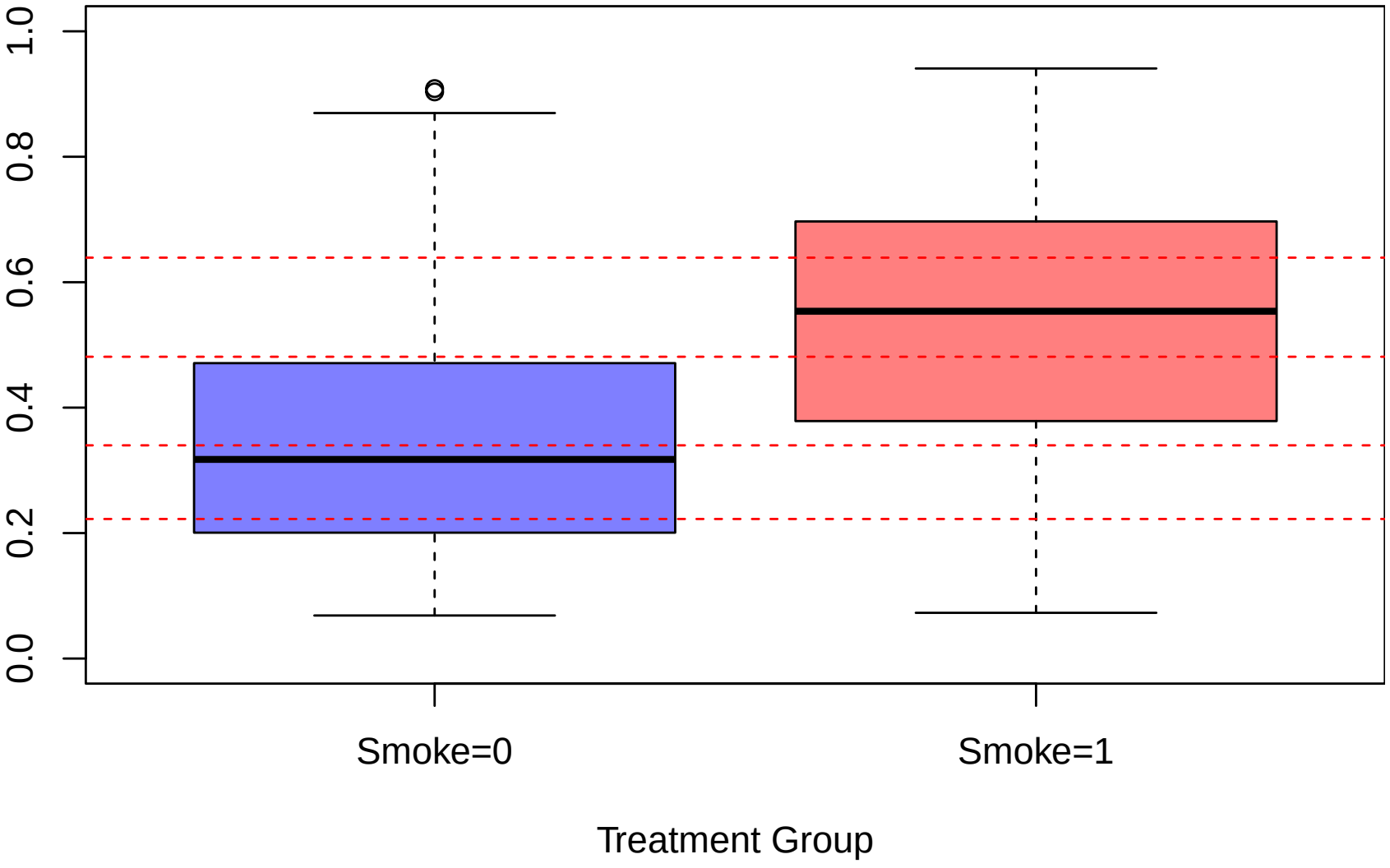
Building the propensity score



Building the propensity score

```
boxplot(ps0,ps1,ylab="PS",xlab="Treatment Group",ylim=range(0,1),
        names=c('Smoke=0','Smoke=1'),col=colvec);
abline(h=quints[2:5],col="red",lty=2)
```

Building the propensity score



Building the propensity score

There seems to be good overlap.

We can therefore proceed to check for balance knowing we have sufficient numbers of smokers and non-smokers in each quintile to ensure the stratum-specific estimates are not too unstable.

Building the propensity score

We can proceed to check for balance within PS quintiles

```
Pcat<-as.numeric(cut(ps.lr,quints,include.lowest=T))
smd.mat<-ExtractSmd(tabUnmatched)
for(k in 1:5){
    nhanesQ<-small.nhanes[Pcat == k,]
    tabQs <- CreateTableOne(vars = vars, strata = "SmokeNow",
                data = nhanesQ, test = FALSE)
    smd.mat<-cbind(smd.mat,ExtractSmd(tabQs))
}
colnames(smd.mat)<-c('All','Q1','Q2','Q3','Q4','Q5')
round(smd.mat,4)
```

```
+                  All     Q1     Q2     Q3     Q4     Q5
+ Gender         0.1379 0.1017 0.1043 0.0286 0.2003 0.0308
+ Age            0.5918 0.2574 0.1715 0.0993 0.3106 0.1642
+ Race3          0.3148 0.3169 0.1117 0.3444 0.4147 0.2869
+ Education      0.5119 0.5378 0.4167 0.2800 0.2378 0.3018
+ MaritalStatus 0.4877 0.4320 0.2386 0.2725 0.2332 0.2608
+ Poverty        0.4530 0.0865 0.1256 0.1136 0.0041 0.1455
```

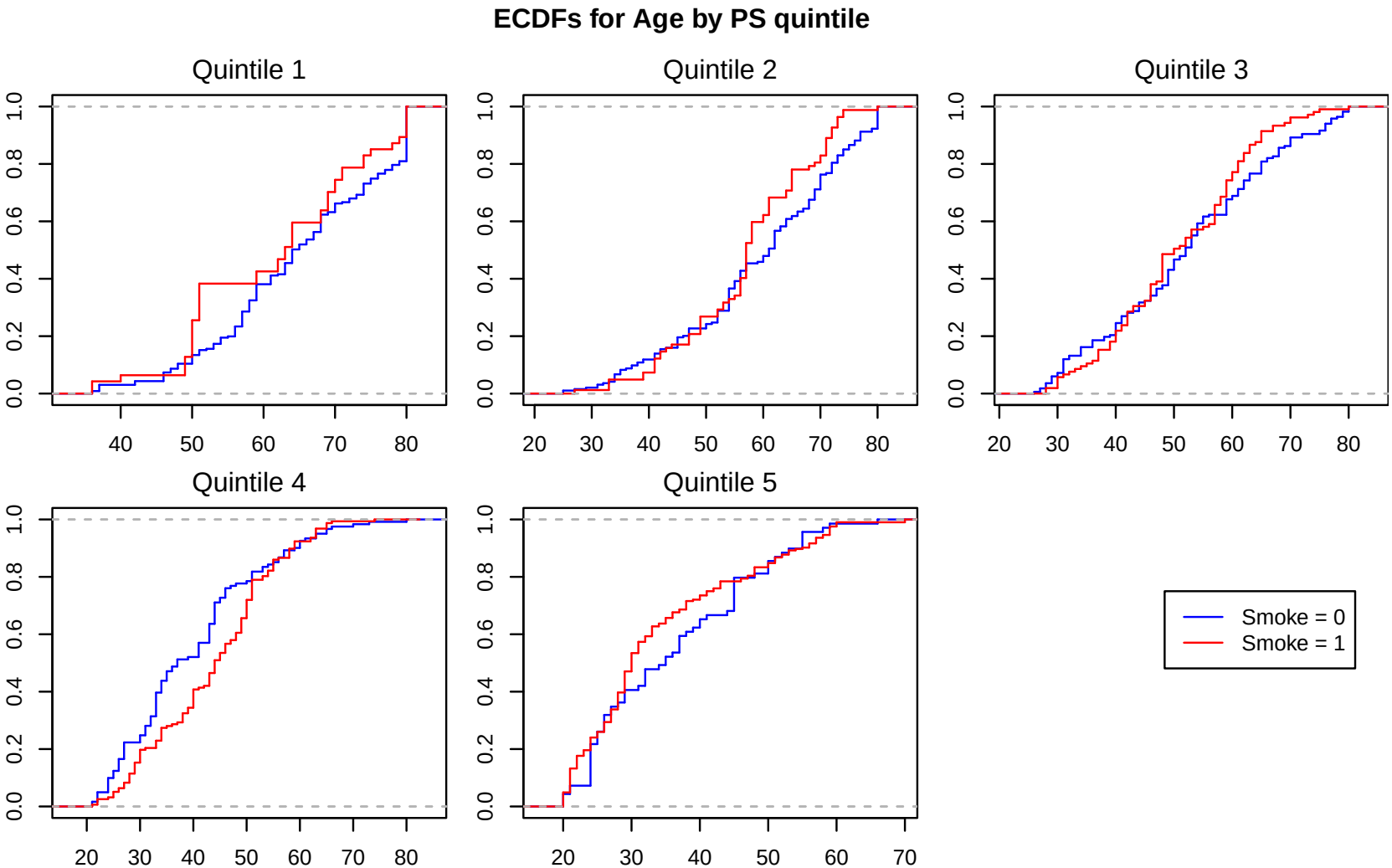
Building the propensity score

Balance does not appear to have been achieved: we have SMDs > 0.1 for at least three quintiles for all variables, and the empirical CDFs of age do not overlap in several quintiles.

Building the propensity score

```
par(mar=c(2,3,2,0),mfrow=c(2,3),oma=c(0,0,2,0))
for(k in 1:5){
    age0 <- small.nhanes$Age[Smoke==0 & Pcat==k]
    age1 <- small.nhanes$Age[Smoke==1 & Pcat==k]
    ecdf0 <- ecdf(age0)
    ecdf1 <- ecdf(age1)
    plot(ecdf0, verticals=TRUE, do.points=FALSE,
        main=substitute(paste('Quintile ',k),list(k=k)),col='blue')
    plot(ecdf1, verticals=TRUE, do.points=FALSE, add=TRUE, col='red')
}
plot(age0,type='n',ylim=range(0,1),axes=FALSE)
title("ECDFs for Age by PS quintile",outer = TRUE)
legend(30,0.75,c('Smoke = 0', 'Smoke = 1'),col=c('blue','red'),lty=1)
```

Building the propensity score



Matching using the propensity score

The function `MatchBalance` from the `Matching` library provides many more details than `CreateTableOne`, including:

- mean, median, and maximum difference in empirical CDF plots,
- mean, median, and maximum difference in empirical QQ plots,
- Kolmogorov-Smirnov statistics,
- ratio of variances,
- p-value for t-test.

Matching using the propensity score

Create the PS-matched sample:

```
small.nhanes$ps.lr<-ps.lr
ps.lr.match <- Match(Tr=small.nhanes$SmokeNow,X=small.nhanes$ps.lr,
                         estimand="ATE",ties=FALSE)
length(ps.lr.match$index.dropped)

+ [1] 0
```

The matched sample can be checked for balance:

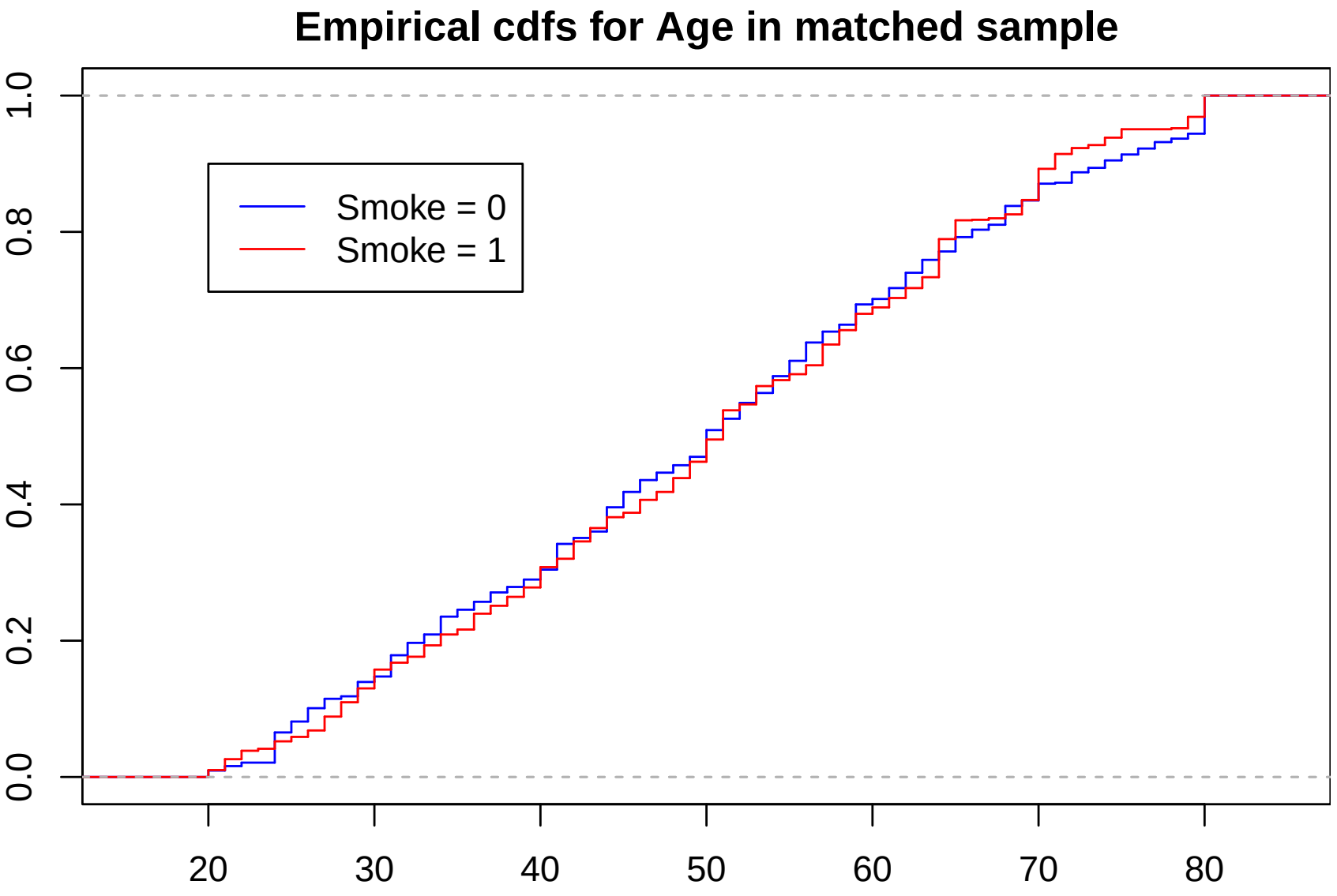
```
MatchBalance(SmokeNow~Gender+Age+Education+MaritalStatus+Poverty,
                    data=small.nhanes,match.out=ps.lr.match)
```

(results not shown).

Matching using the propensity score

```
matched.samp <- small.nhanes[c(ps.lr.match$index.control,
                               ps.lr.match$index.treated),]
tabMatched<-CreateTableOne(vars = vars, strata = "SmokeNow",
                data = matched.samp, test = FALSE)
```

Matching using the propensity score



Matching using the propensity score

Matching seems to have matched the age profiles for the smoking and non-smoking groups if we compare the SMDs:

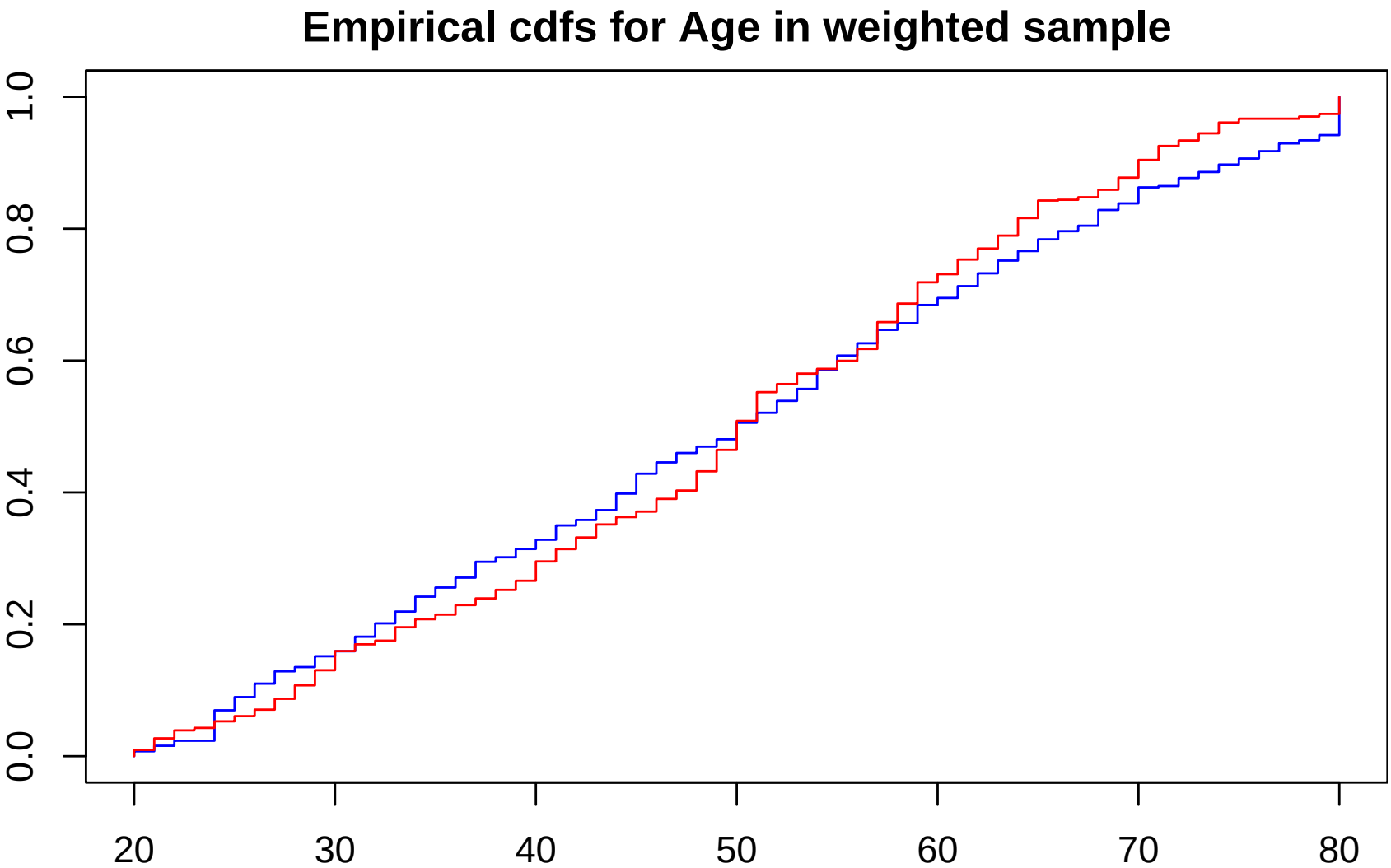
```
smd.mat<-cbind(smd.mat,ExtractSmd(tabMatched))
colnames(smd.mat)<-c('All','Q1','Q2','Q3','Q4','Q5',"Match")
round(smd.mat,4)

+                   All     Q1     Q2     Q3     Q4     Q5  Match
+ Gender          0.1379 0.1017 0.1043 0.0286 0.2003 0.0308 0.0131
+ Age             0.5918 0.2574 0.1715 0.0993 0.3106 0.1642 0.0114
+ Race3           0.3148 0.3169 0.1117 0.3444 0.4147 0.2869 0.1172
+ Education       0.5119 0.5378 0.4167 0.2800 0.2378 0.3018 0.1382
+ MaritalStatus  0.4877 0.4320 0.2386 0.2725 0.2332 0.2608 0.1078
+ Poverty         0.4530 0.0865 0.1256 0.1136 0.0041 0.1455 0.0617
```

Inverse Weighting

We may study an *inverse weighted* sample using the `survey` and `Hmisc` packages.

```
ps.lr.weight <- small.nhanes$SmokeNow/ps.lr +
                    (1-small.nhanes$SmokeNow)/(1-ps.lr)
nhanes.IPW.lr <- svydesign(ids=~0, data=small.nhanes,
                  weights=ps.lr.weight)
tabIPW<-svyCreateTableOne(vars = vars, strata = "SmokeNow",
                                data = nhanes.IPW.lr, test = FALSE)
```

Inverse Weighting



Inverse Weighting

```
+                    All     Q1     Q2     Q3     Q4     Q5  Match    IPW
+ Gender          0.1379 0.1017 0.1043 0.0286 0.2003 0.0308 0.0131 0.0235
+ Age             0.5918 0.2574 0.1715 0.0993 0.3106 0.1642 0.0114 0.0139
+ Race3           0.3148 0.3169 0.1117 0.3444 0.4147 0.2869 0.1172 0.0516
+ Education       0.5119 0.5378 0.4167 0.2800 0.2378 0.3018 0.1382 0.0292
+ MaritalStatus 0.4877 0.4320 0.2386 0.2725 0.2332 0.2608 0.1078 0.0235
+ Poverty         0.4530 0.0865 0.1256 0.1136 0.0041 0.1455 0.0617 0.0001
```

Matching and weighting samples yield good balance.

Summary

- Creating or restoring confounder balance is essential to estimating a causal effect.
- It can be hard to assess overlap or achieve balance in high dimensions.
- The propensity score, a scalar summary of confounding variables, simplifies this task.
- However:
  - fitting a model for treatment does not guarantee balance,
  - fitting a model that predicts treatment with a high degree of precision can be unhelpful.

Estimating the ATE

We proceed now to estimating the average treatment effect (ATE), using:

- outcome regression:
- PS stratification,
- PS matching,
- PS regression,
- IPW & AIPW.

Estimating the ATE

Note that the IPW and AIPW estimators can suffer from the fact that the inverse weights based on the propensity score can get large if the propensity score is near zero or one.

For these estimators, *truncation* can be used, where propensity score values more extreme than a given threshold (0.001 or 0.999 say) are set equal to these thresholds.

Outcome regression

Outcome regression, unadjusted:

```
coef(summary(lm(BPSysAve~SmokeNow,data=small.nhanes)))[2,]

+      Estimate    Std. Error       t value      Pr(>|t|)
+ -3.6793569602  0.9639505436 -3.8169561548  0.0001411118
```

Outcome regression

Outcome regression no interaction: We use the model with other predictors included additively:

```
fmod<-formula(BPSysAve~SmokeNow+Gender+Age+Race3+Education+
                           MaritalStatus+HHIncome+Poverty)
coef(lm(fmod,data=small.nhanes))[2]

+  SmokeNow
+ -1.097768
```

```
nhanes.allsmoke <- small.nhanes
nhanes.allsmoke$SmokeNow <- 1
nhanes.nosmoke <- small.nhanes
nhanes.nosmoke$SmokeNow <- 0
mod1.lm <- lm(fmod,data=small.nhanes)
APO.lm.1 <- mean(predict(mod1.lm,nhanes.allsmoke))
APO.lm.0 <- mean(predict(mod1.lm,nhanes.nosmoke))
APO.lm.1 - APO.lm.0  #Note: same as for previous answer regression

+ [1] -1.097768
```

The naive conditional effect estimate is more than 3 times greater than its confounder-adjusted counterpart.

Outcome regression

Outcome regression with interactions: We now add interaction terms

```
fmodi<-formula(BPSysAve~SmokeNow+Gender+Age+Race3+Education+
        MaritalStatus+HHIncome+Poverty+
        SmokeNow:HHIncome+SmokeNow:Gender+SmokeNow:Age)
mod1.lmX <- lm(fmodi,data=small.nhanes)
```

Outcome regression

```
round(coef(summary(mod1.lmX))[-c(3:30),],3)

+                                Estimate Std. Error t value Pr(>|t|)
+ (Intercept)                     114.998      5.892  19.516    0.000
+ SmokeNow                        -14.281      6.041  -2.364    0.018
+ SmokeNow:HHIncome 5000-9999      20.654      7.813   2.644    0.008
+ SmokeNow:HHIncome10000-14999     17.000      6.498   2.616    0.009
+ SmokeNow:HHIncome15000-19999     12.239      6.354   1.926    0.054
+ SmokeNow:HHIncome20000-24999     15.199      6.334   2.400    0.017
+ SmokeNow:HHIncome25000-34999     20.683      6.123   3.378    0.001
+ SmokeNow:HHIncome35000-44999      7.055      6.216   1.135    0.257
+ SmokeNow:HHIncome45000-54999     23.352      6.351   3.677    0.000
+ SmokeNow:HHIncome55000-64999     16.682      6.449   2.587    0.010
+ SmokeNow:HHIncome65000-74999     16.020      6.566   2.440    0.015
+ SmokeNow:HHIncome75000-99999     11.190      6.248   1.791    0.074
+ SmokeNow:HHIncomemore 99999      15.321      5.885   2.603    0.009
+ SmokeNow:Gendermale               5.591      1.745   3.205    0.001
+ SmokeNow:Age                     -0.109      0.055  -1.970    0.049
```

Outcome regression

Here the coefficients alone *do not* provide the ATE estimate, and we need to use the `predict` method to compute the ATE estimate as

$$\frac{1}{n}\sum_{i=1}^{n} m(x_i, 1; \widehat{\beta}) - \frac{1}{n}\sum_{i=1}^{n} m(x_i, 0; \widehat{\beta})$$

Outcome regression

```
APO.lmX.1 <- mean(predict(mod1.lmX,nhanes.allsmoke))
APO.lmX.0 <- mean(predict(mod1.lmX,nhanes.nosmoke))
APO.lmX.1 - APO.lmX.0

+ [1] -1.402538
```

PS stratification

We first fit the propensity score model based on logistic regression and then use *stratification* based on PS quintiles.

```
fmods<-formula(SmokeNow~Gender+Age+Race3+Education+
                            MaritalStatus+HHIncome+Poverty)
ps.mod <- glm(fmods,data=small.nhanes,family="binomial")
ps.lr <- predict(ps.mod,type="response")
Y<-small.nhanes$BPSysAve
ps.lr.quints <- cut(ps.lr,quints,labels=1:5)
p.strat <- table(ps.lr.quints)/length(ps.lr.quints)
ATE.strat <- rep(NA,5)
for(j in 1:5) {
    ATE.strat[j]<-mean(Y[Smoke==1 & ps.lr.quints==j])-
                  mean(Y[Smoke==0 & ps.lr.quints==j])
}
sum(ATE.strat*p.strat)

+ [1] -1.816879
```

PS matching

```
ps.lr.match <- Match(Tr=small.nhanes$SmokeNow,
      X=small.nhanes$ps.lr,estimand="ATE",ties=FALSE)
matched.samp <- small.nhanes[c(ps.lr.match$index.control,
                               ps.lr.match$index.treated),]
mean(matched.samp$BPSysAve[matched.samp$SmokeNow == 1]) -
    mean(matched.samp$BPSysAve[matched.samp$SmokeNow == 0])

+ [1] -0.6129267
```

PS regression

We now attempt three propensity score regression analyses based on increasingly complex specifications

$$
\begin{aligned}
\text{PSR 1} \quad Y &= \beta_0 + \beta_1 e(X;\widehat{\alpha}) + \psi_1 Z + \epsilon \\
\text{PSR 2} \quad Y &= \beta_0 + \beta_1 e(X;\widehat{\alpha}) + \beta_2 \{e(X;\widehat{\alpha})\}^2 + \\
&\qquad \psi_1 Z + \psi_2 Z e(X;\widehat{\alpha}) + \psi_3 Z \{e(X;\widehat{\alpha})\} + \epsilon \\
\text{PSR 3} \quad Y &= s(e(X;\widehat{\alpha})) + \psi_1 Z + \psi_2 Z s(e(X;\widehat{\alpha})) + \epsilon
\end{aligned}
$$

where $s(.)$ is a flexible spline function.

PS regression

PSR 1:

```
mod1.PSlm1 <- lm(BPSysAve~SmokeNow+ps.lr,data=small.nhanes)
APO.PSlm1.1 <- mean(predict(mod1.PSlm1,nhanes.allsmoke))
APO.PSlm1.0 <- mean(predict(mod1.PSlm1,nhanes.nosmoke))
APO.PSlm1.1 - APO.PSlm1.0

+ [1] -1.10791
```

PS regression

PSR 2:

```
psmod2<-formula(BPSysAve~(1+SmokeNow)*(ps.lr+I(ps.lr^2)))
mod1.PSlm2 <- lm(psmod2,data=small.nhanes)
APO.PSlm2.1 <- mean(predict(mod1.PSlm2,nhanes.allsmoke))
APO.PSlm2.0 <- mean(predict(mod1.PSlm2,nhanes.nosmoke))
APO.PSlm2.1 - APO.PSlm2.0

+ [1] -1.681372
```

PS regression

PSR 3:

```
psmod3<-formula(BPSysAve~
            (1+SmokeNow)*bs(ps.lr,df=4,Boundary.knots = range(0,1)))
mod1.PSlm3 <- lm(psmod3,data=small.nhanes)
APO.PSlm3.1 <- mean(predict(mod1.PSlm3,nhanes.allsmoke))
APO.PSlm3.0 <- mean(predict(mod1.PSlm3,nhanes.nosmoke))
APO.PSlm3.1 - APO.PSlm3.0

+ [1] -1.971314
```

Inverse Weighting

The IPW estimator can be computed using a direct calculation based on the weight

$$w_i = \frac{Z_i}{e(X_i; \widehat{\alpha})} + \frac{(1 - Z_i)}{1 - e(X_i; \widehat{\alpha})}$$

computed after the PS model has been estimated.

Inverse Weighting

IPW0: Ordinary weight form

$$\frac{1}{n}\sum_{i=1}^{n} w_i Z_i Y_i - \frac{1}{n}\sum_{i=1}^{n} w_i (1 - Z_i) Y_i$$

IPW1: Standardized weight form

$$\frac{1}{\sum_{i=1}^{n} w_i}\left(\sum_{i=1}^{n} w_i Z_i Y_i - \sum_{i=1}^{n} w_i (1 - Z_i) Y_i\right)$$

obtained via weighted least squares.

Inverse Weighting

```
ps.lr.weight <- Smoke/ps.lr + (1-Smoke)/(1-ps.lr)
mean(Smoke*Y*ps.lr.weight) -mean((1-Smoke)*Y*ps.lr.weight)   #IPW0

+ [1] -1.928655

coef(lm(Y ~ Smoke, weights = ps.lr.weight))[2]               #IPW1

+     Smoke
+ -1.991233
```

Inverse Weighting

The augmented IPW (AIPW) estimator adds a proposed mean model for the APOs, that is

$$\widehat{\mu}(1) = \frac{1}{n}\sum_{i=1}^{n} \frac{Z_i(Y_i - m(X_i, 1; \widehat{\beta}))}{e(X_i; \widehat{\alpha})} + \frac{1}{n}\sum_{i=1}^{n} m(X_i, 1; \widehat{\beta})$$

$$\widehat{\mu}(0) = \frac{1}{n}\sum_{i=1}^{n} \frac{(1 - Z_i)(Y_i - m(X_i, 0; \widehat{\beta}))}{1 - e(X_i; \widehat{\alpha})} + \frac{1}{n}\sum_{i=1}^{n} m(X_i, 0; \widehat{\beta})$$

or computed via an augmented outcome regression with the two additional predictors

$$\frac{Z_i}{e(X_i; \widehat{\alpha})} \qquad \frac{(1 - Z_i)}{1 - e(X_i; \widehat{\alpha})}.$$

Inverse Weighting

Here we use the outcome model with interactions fitted previously:

```
fmodi<-formula(BPSysAve~SmokeNow+Gender+Age+Race3+Education+
        MaritalStatus+HHIncome+Poverty+
        SmokeNow:HHIncome+SmokeNow:Gender+SmokeNow:Age)
mod1 <- lm(fmodi,data=small.nhanes)
m1<-predict(mod1,nhanes.allsmoke)
m0<-predict(mod1,nhanes.nosmoke)
ps.trunc<-pmin(pmax(0.001,ps.lr),0.999)
APO.1<-mean(Smoke*(Y-m1)/ps.trunc)+mean(m1)
APO.0<-mean((1-Smoke)*(Y-m0)/(1-ps.trunc))+mean(m0)
APO.1-APO.0

+ [1] -1.854776
```

Here we use the *augmented outcome regression* (AOR) approach

Inverse Weighting

```
small.nhanes$ps <- ps.lr
fmodia<-formula(BPSysAve~SmokeNow+Gender+Age+Race3+Education+
                         MaritalStatus+HHIncome+Poverty+
                         SmokeNow:HHIncome+SmokeNow:Gender+SmokeNow:Age+
                         I(SmokeNow/ps)+I((1-SmokeNow)/(1-ps)))
mod1.lmX <- lm(fmodia,data=small.nhanes)
nhanes.allsmoke <- nhanes.nosmoke <- small.nhanes;
nhanes.allsmoke$SmokeNow <- 1
nhanes.nosmoke$SmokeNow <- 0
mean(predict(mod1.lmX,nhanes.allsmoke))-
    mean(predict(mod1.lmX,nhanes.nosmoke))

+ [1] -1.818062
```

Additional considerations

- All of the PS approaches considered rely on ‘substitution estimators’.
  - In PS regression, we plug in an estimated PS as a covariate.
  - In IPW, we plug in estimated weights.

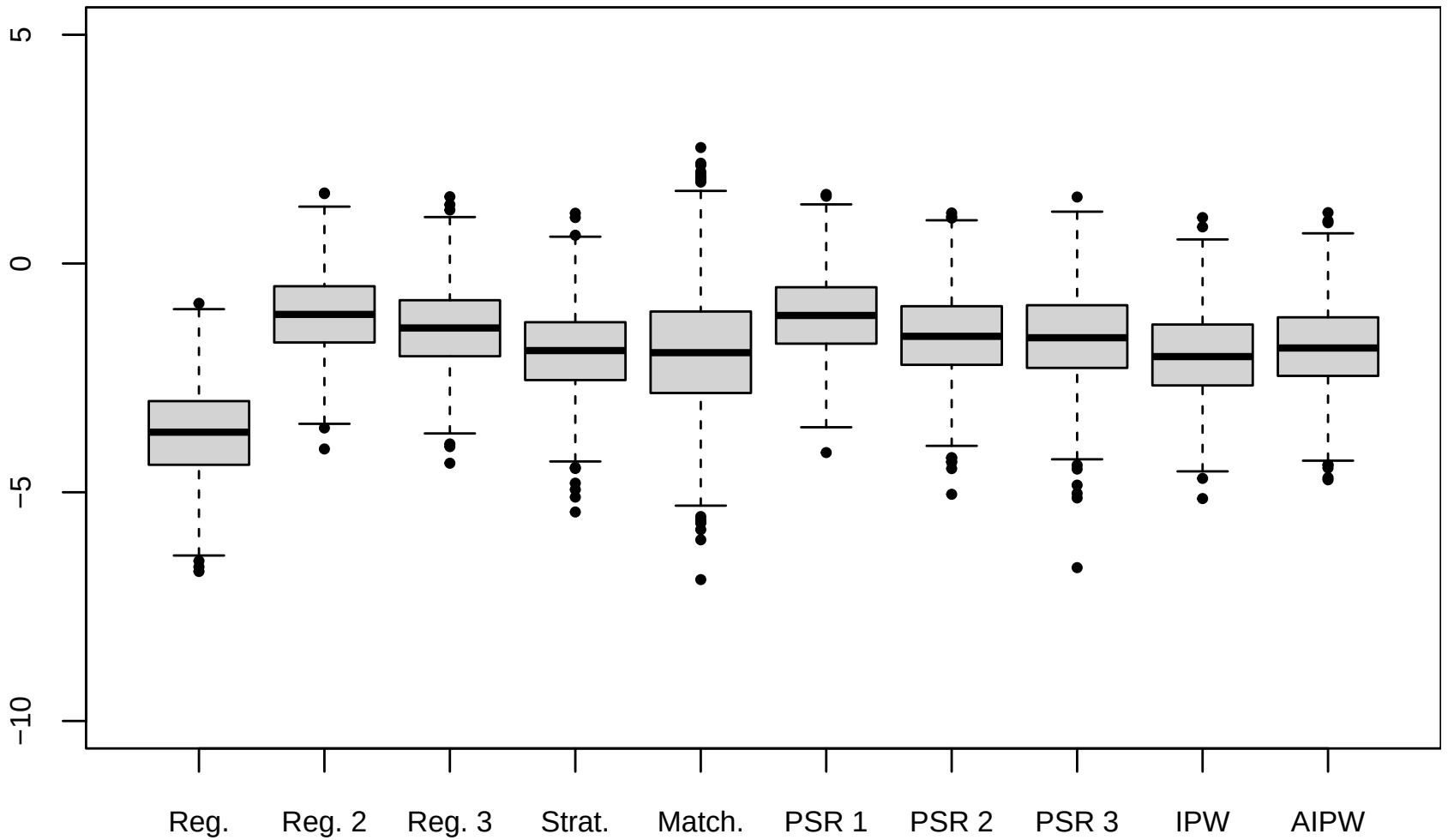
Additional considerations

- We need to account for this when estimating standard errors and/or confidence intervals.
- Analytically derived asymptotic variances can be used, but are not provided in many standard software packages.
- The easiest approach is to bootstrap; however, that the bootstrap is *not* in general valid for matching.

Bootstrap analysis

1000 bootstrap samples are used to approximate the sampling distribution of the various estimators.

Bootstrap analysis



Bootstrap analysis

The boxplot demonstrates largely similar estimates from quite different analysis methods. The estimates and standard errors are as follows.

```
ATE<-apply(ests.mat,2,mean)
ATE.se<-sqrt(apply(ests.mat,2,var))
ATE.res<-cbind(ATE,ATE.se)
rownames(ATE.res)<-lvec
colnames(ATE.res)<-c('Est.','s.e.')
ATE.res
```

```
+              Est.      s.e.
+ Reg.    -3.697501 0.9908478
+ Reg. 2  -1.112514 0.9154300
+ Reg. 3  -1.420021 0.9393691
+ Strat.  -1.921612 0.9964034
+ Match.  -1.944049 1.3799181
+ PSR 1    -1.131901 0.9118278
+ PSR 2    -1.578981 0.9734583
+ PSR 3    -1.633296 1.0567595
+ IPW      -1.999992 1.0182845
+ AIPW     -1.833907 0.9977362
```