

1 Setup for One Worked Example

```
> library(tidyverse)
> library(mvtnorm)
> library(optmatch)
> library(randomForest)
> library(dRCT)
> source("simulationFunctions.r")
> source("displaySim.r")
> ## parameters
> n <- 400 # total sample size
> p <- 600 # number of covariates
> nt <- 50 # number of treated subjects
> decay <- 0.05
> ## make the dataset
> set.seed(613)
> X <- makeX(ev=exp(-decay*c(1:p)))
> bg <- coefs(0.5,n,p)[[1]]
> dat <- makeDataCurved(X,bg,nt)
> ## propensity score matching
> pmod <- propmodel(dat,X) # propensity score model
> pscores <- pmod$linear
> match <- PSmatch(dat,pscores)
> ## prognostic scores, aka  $\hat{y}^{\text{rem}}$ 
> prog <- rfRemMod(X,dat,match)
> ## matched data (discard remnant)
> matched <- cbind(dat,prog=prog,match=match)[!is.na(match),]
> matched <- within(matched,progC <- prog-mean(prog))
> head(matched)
```

	y	z		prog	match	progC
1	3.6010551	0	-1.508331	1.40	1.2295133	
5	3.2556765	0	-1.249350	1.27	1.4884941	
6	-4.4930637	0	-4.263429	1.3	-1.5255846	
7	-10.4392862	1	-1.625200	1.1	1.1126438	
8	-11.5573065	1	-2.299800	1.2	0.4380446	
13	0.7641727	0	-2.296196	1.18	0.4416483	

```
> ## data with one row per match:
> paired <- matched%>%
+   group_by(match)%>%
+   summarise(
+     deltaY=y[z==1]-y[z==0],
+     meanX=mean(prog),
+     meanXC=mean(progC),
+     deltaX=prog[z==1]-prog[z==0])
```

```

+      )
> head(paired)

# A tibble: 6 × 5
  match  deltaY meanX meanXC deltaX
  <fct>   <dbl> <dbl>  <dbl>  <dbl>
1 1.1    -3.17  -2.15  0.590  1.05
2 1.10   -7.39  -3.28 -0.544  2.80
3 1.11  -12.0   -2.28  0.458  0.534
4 1.12   -0.516 -2.53  0.209  2.45
5 1.13  -13.7   -3.01 -0.273  0.466
6 1.14    5.08  -2.48  0.260 -0.515

Basic estimators:
 $\hat{\tau}_{SD} = \bar{W} = \overline{\Delta Y}$ :

> print(tauhatSD <- mean(paired$deltaY))

[1] -5.539966

> ## compare:
> print(matchEst(dat$y, dat$z, match))

      z
-5.539966

 $\hat{\tau}_{rebar} = \overline{\Delta Y} - \overline{\Delta X}$ 

> print(tauhatRebar <- mean(paired$deltaY) - mean(paired$deltaX))

[1] -6.556388

```

2 Simple forms for the p_loop estimator

To keep things simple, I'm:

- dropping the i subscript and considering an individual pair.
- assuming there's no treatment effect, so $t_1 = c_1 = Y_1$ and $t_2 = c_2 = Y_2$
- assuming $T = 1$, so the subject with subscript 1 is treated

From the paper, we have:

$$\hat{\tau}_{SD} = W = T(t_1 - c_2) + (1 - T)(t_2 - c_1) \equiv Y_1 - Y_2$$

$$\hat{\tau}_{rebar} = Y_1 - Y_2 - (X_1 - X_2)$$

and

$$\hat{\tau} = W - \hat{d} = Y_1 - Y_2 - \hat{d}$$

2.1 Imputing Potential Outcomes

If we estimate \hat{d} by imputing potential outcomes,

$$\hat{d} = \frac{\hat{t}_1 + \hat{c}_1 - \hat{t}_2 - \hat{c}_2}{2}$$

To impute the potential outcomes, we fit a model:

$$Y_i = \beta_0 + \beta_1 Z_i + \beta_2 X_i + \beta_3 Z_i X_i + \epsilon_i \quad (1)$$

where:

- Y is the outcome
- Z is treatment assignment
- X is the prediction from the remnant, i.e. \hat{y}^{rem} , aka prognostic score

Then:

$$\begin{aligned}\hat{c}_i &= \hat{\beta}_0 + \hat{\beta}_2 X_i \\ \hat{t}_i &= \hat{\beta}_0 + \hat{\beta}_1 + (\hat{\beta}_2 + \hat{\beta}_3) X_i\end{aligned}$$

and

$$\begin{aligned}\hat{d} &= \frac{1}{2} (\hat{t}_1 - \hat{t}_2 + \hat{c}_1 - \hat{c}_2) \\ &= \frac{1}{2} [(\hat{\beta}_2 + \hat{\beta}_3)(X_1 - X_2) + \hat{\beta}_2(X_1 - X_2)] \\ &= (\hat{\beta}_2 + \hat{\beta}_3/2)(X_1 - X_2) \\ &= \tilde{\beta} \Delta X\end{aligned}$$

where $\tilde{\beta} = (\hat{\beta}_2 + \hat{\beta}_3/2)$ and $\Delta X = X_1 - X_2$.

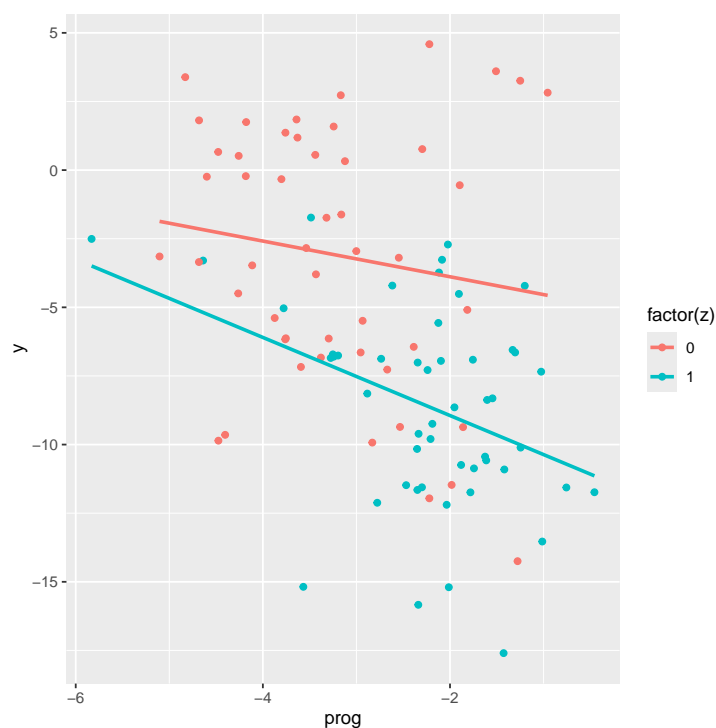
2.1.1 An Example Computation

```
> print(matched%>%
+   ggplot(aes(prog,y,color=factor(z)))+
+   geom_point()+
+   geom_smooth(method="lm",se=FALSE)
+ )
> ## estimating OLS models leaving out each pair
> ### estimated coefficients:
> betas <- matrix(nrow=50,ncol=4,dimnames=list(levels(matched$match),names(coef(lm(y~z*prog,
> ### dhat=(beta_2+beta_3/2)delta X and W=delta Y:
> dhatW <- matrix(nrow=50,ncol=2,dimnames=list(levels(matched$match),c("d","W"))))
> for(m in levels(matched$match)){
+   mod <- lm(y~z*progC,data=matched,subset=match!=m)
```

```

+   betas[m,] <- cc <- unname(coef(mod))
+   dhatW[m,] <- with(subset(paired,match==m),
+                     c(d=(cc[3]+cc[4]/2)*deltaX,W=deltaY))
+ }

```



ReLOOP esti-

mator:

```

> print(tauhat <- mean(dhatW[,"W"]-dhatW[,"d"]))
[1] -4.483226

```

Compare:

```

> reloopOLSpo <- p_loop(Y=dat$y[!is.na(match)],Tr=dat$z[!is.na(match)],
+                       Z=cbind(prog[!is.na(match)]),
+                       P=as.numeric(match[!is.na(match)]),
+                       pred=p_ols_po)
> print(reloopOLSpo)

```

```

      Estimate Std. Error  t value Pr(>|t|)
ATE: -4.4832263   0.8215749  -5.45687 4.846e-08 ***
---

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

2.1.2 Poor Man's ReLOOP (impute potential outcomes version)

Instead of doing leave-one-out, just fit one big model (1).

Center the prognostic scores so that $\bar{X} = 0$.

Then:

$$\begin{aligned}\bar{Y}_1 &= \hat{\beta}_0 + \hat{\beta}_1 + (\hat{\beta}_2 + \hat{\beta}_3)\bar{X}_{Z=1} \\ \bar{Y}_2 &= \hat{\beta}_0 + \hat{\beta}_2\bar{X}_{Z=0} \\ \bar{d} &= (\hat{\beta}_2 + \hat{\beta}_3/2)(\bar{X}_{Z=1} - \bar{X}_{Z=0})\end{aligned}$$

So:

$$\begin{aligned}\hat{\tau}_{\text{poor}} &= \bar{Y}_1 - \bar{Y}_2 - \bar{d} \\ &= \hat{\beta}_1 + \hat{\beta}_2(\bar{X}_{Z=1} - \bar{X}_{Z=0}) + \hat{\beta}_3\bar{X}_{Z=1} \\ &\quad - \hat{\beta}_2(\bar{X}_{Z=1} - \bar{X}_{Z=0}) - \hat{\beta}_3/2(\bar{X}_{Z=1} - \bar{X}_{Z=0}) \\ &= \hat{\beta}_1 + \hat{\beta}_3/2(\bar{X}_{Z=1} + \bar{X}_{Z=0}) \\ &= \hat{\beta}_1 + \hat{\beta}_3\bar{X} \\ &= \hat{\beta}_1\end{aligned}$$

To wit:

```
> poormod <- lm(y~z*progC,data=matched)
```

Explicit calculation:

```
> poorCoef <- coef(poormod)
> dhat <- (poorCoef[3]+poorCoef[4]/2)*paired$deltaX
> #Estimate:
> mean(paired$deltaY)-mean(dhat)

[1] -4.486437
```

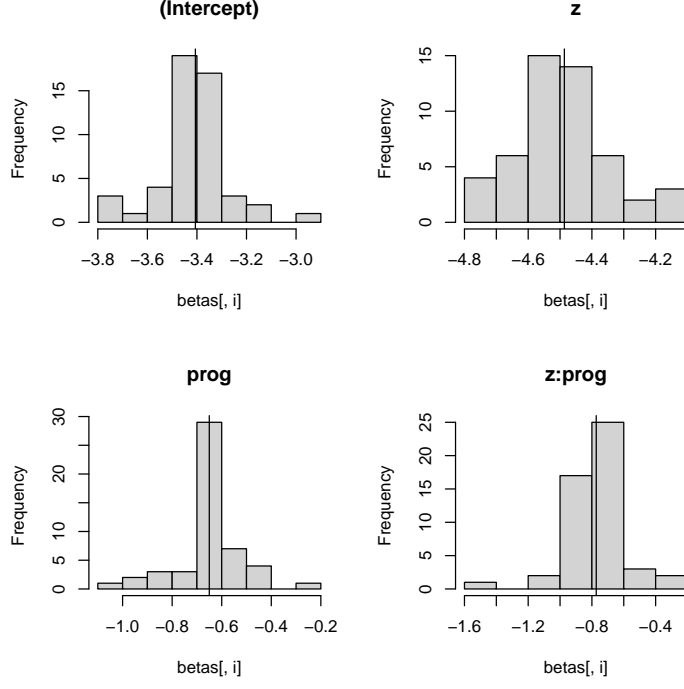
Or just:

```
> poorCoef[2]

      z
-4.486437
```

We can also compare the estimated coefficients from the global OLS (vertical lines) to those from the leave-one-out models (histograms):

```
> par(mfrow=c(2,2))
> for(i in 1:4){
+   hist(betas[,i],main=colnames(betas)[i])
+   abline(v=poorCoef[i])
+ }
> par(mfrow=c(1,1))
```



2.2 Imputing a and b

The other approach to imputing \hat{d} is to impute $a = t_1 - c_2$ and $b = t_2 - c_1$ directly. Then

$$\hat{d} = \frac{\hat{a} - \hat{b}}{2}$$

Let

$$\Delta Y = Y_1 - Y_2$$

Then we fit the OLS model:

$$\Delta Y_i = \gamma_0 + \gamma_1 \Delta X_i + \gamma_2 \frac{X_{i1} + X_{i2}}{2} + \epsilon_i \quad (2)$$

Given estimated coefficients from (2),

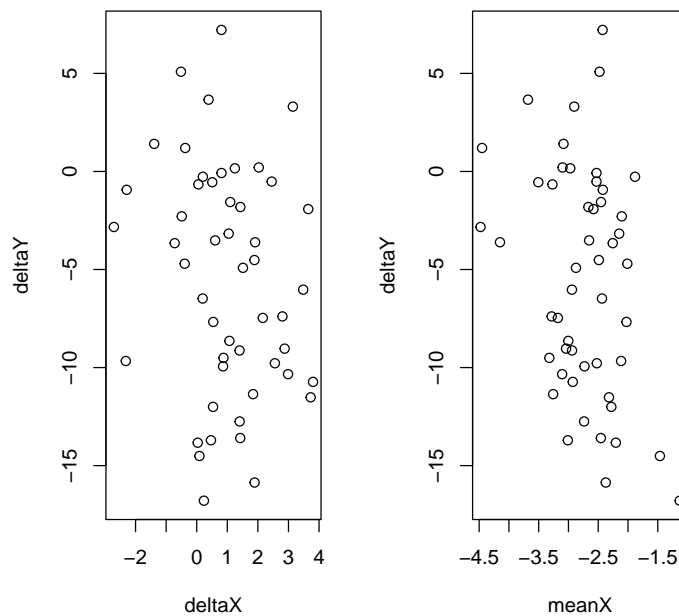
$$\begin{aligned} \hat{a} &= \hat{\gamma}_0 + \hat{\gamma}_1 \Delta X + \hat{\gamma}_2 \frac{X_1 + X_2}{2} \\ \hat{b} &= \hat{\gamma}_0 - \hat{\gamma}_1 \Delta X + \hat{\gamma}_2 \frac{X_1 + X_2}{2} \end{aligned}$$

since with a the first one is treated (as in the assumed setup) and with b the 2nd one is treated so X_1 and X_2 are switched—hence the negative sign before ΔX . Then

$$\hat{d} = \frac{\hat{a} - \hat{b}}{2} = \hat{\gamma}_2 \Delta X$$

2.2.1 An Example Computation

```
> par(mfrow=c(1,2))
> with(paired,plot(deltaX,deltaY))
> with(paired,plot(meanX,deltaY))
```



By hand:

```
> gammas <- t(sapply(1:nrow(paired),\ (i) coef(lm(deltaY~meanX+deltaX,data=paired[-i,]))))
> dhat <- gammas[, "deltaX"]*paired$deltaX
> tauhats <- paired$deltaY-dhat
> mean(tauhats)
```

```
[1] -4.729661
```

Compare:

```
> p_loop(Y=dat$y[!is.na(match)],Tr=dat$z[!is.na(match)],
+       Z=cbind(prog[!is.na(match)]),
+       P=as.numeric(match[!is.na(match)]),
+       pred=p_ols_v12)
```

```
      Estimate Std. Error  t value   Pr(>|t|)
ATE: -4.7296613   0.8087295 -5.84826 4.9674e-09 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.2.2 Poor Man's Version

Assume, as before, that $\bar{X} = 0$.

One could approximate the LOOP estimator by fitting one global model:

$$\Delta Y = \gamma_0 + \gamma_1 \Delta X + \gamma_2 \frac{X_1 + X_2}{2} + \epsilon \quad (3)$$

Then

$$\hat{\tau} \approx \overline{\Delta Y} - \hat{\gamma}_1 \overline{\Delta X} = \hat{\gamma}_0$$

since $\overline{X_1 + X_2} = 0$.

It turns out (by a FWL argument left as an exercise for the reader) that this is equivalent to fitting the model

$$Y_i = \alpha_{0m[i]} + \alpha_1 Z_i + \alpha_2 X_i + \alpha_3 Z_i X_i + \epsilon_i$$

Where $\alpha_{0m[i]}$ are fixed effects for the pair, with $\tau_{poor} = \hat{\alpha}_1$.

Just look:

```
> ## model with deltas:
> coef(lm(deltaY~deltaX+meanXC,data=paired))[1]

(Intercept)
-4.720327

> ## model with fixed effects
> coef(lm(y~match+z*progC,data=matched))['z']

z
-4.720327
```

3 When is $\hat{\tau}$ less biased than $\hat{\tau}_{SD}$?

To make things simpler, rather than talk about bias we'll just talk about error with a fixed dataset. The bias argument follows in a straightforward way.

Since there is no treatment effect, $\hat{\tau} = error(\hat{\tau})$. So essentially, the question is, under what conditions does this hold:

$$|\hat{\tau}_{SD}| > |\hat{\tau}| \quad (4)$$

Recall: $\hat{d} = \tilde{\beta} \Delta X$ or $\hat{\gamma}_2 \Delta X$ depending on whether we're imputing potential outcomes or a and b . To be generic, let's just say $\hat{d} = \delta \Delta X$. Then:

$$\begin{aligned} \hat{\tau}_{SD} &= \overline{\Delta Y} \\ \hat{\tau} &= \overline{\Delta Y} - \overline{\delta \Delta X} \end{aligned}$$

So (??) becomes

$$|\overline{\Delta Y}| > |\overline{\Delta Y} - \overline{\delta \Delta X}| \quad (5)$$

Now some high-school algebra:
If $\overline{\Delta Y} > 0$, (5) is equivalent to

$$0 < \overline{\delta \Delta X} < 2\overline{\Delta Y}$$

If $\overline{\Delta Y} < 0$, (5) is equivalent to

$$0 > \overline{\delta \Delta X} > 2\overline{\Delta Y}$$

Alternatively,

$$0 < \frac{\overline{\delta \Delta X}}{\overline{\Delta Y}} < 2 \quad (6)$$

(assuming $\overline{\Delta Y} \neq 0$; otherwise (??) is impossible.)

As $n \rightarrow \infty$, leave-one-out becomes less important (under suitable regularity conditions) so for intuition we can imagine that there's just one δ (i.e. the "poor man's" version) and write (6) as

$$0 < \delta \frac{\overline{\Delta X}}{\overline{\Delta Y}} < 2 \quad (7)$$

4 Breaking ReLOOP

The first inequality of (7) implies that the sign of $\frac{\overline{\Delta X}}{\overline{\Delta Y}}$ matches the sign of δ . So, for instance, if ΔX tends to be positive and ΔY tends to be negative (i.e. prognostic scores tend to be higher treatment than control but outcomes tend to be lower) or vice-versa then δ must be negative. So to find a case where ReLOOP hurts things, we need to construct a dataset where that's not the case.

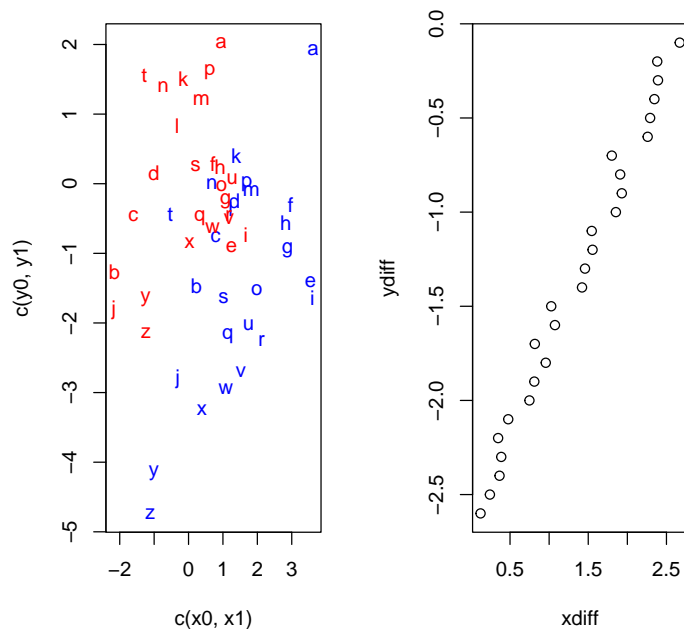
Here's one:

```
> set.seed(613)
> bad <- data.frame(
+   m=letters, # a match
+   y0=rnorm(26),
+   x0=rnorm(26))
> bad <- within(bad,{
+   y1 <- y0-(1:26)/10 #y1<y0, differences grow
+   x1 <- x0+(26:1)/10+rnorm(26,sd=0.1) #x1>x0, differences shrink
+ })
> bad <- within(bad,{
+   ydiff <- y1-y0
+   xdiff <- x1-x0
+   xmean <- (x1+x0)/2
+ })
> ## plot x vs y and delta x vs delta y
```

```

> par(mfrow=c(1,2))
> with(bad,plot(c(x0,x1),c(y0,y1),type="n"))
> text(bad$x1,bad$y1,bad$m,col="blue",xlim=c(.5,11.5),ylim=c(.5,8))
> text(bad$x0,bad$y0,bad$m,col="red")
> with(bad,plot(xdiff,ydiff))
> ## in "long form"
> badLong <- rbind(
+   with(bad, data.frame(y=y0,x=x0,z=0,m=m)),
+   with(bad, data.frame(y=y1,x=x1,z=1,m=m))
+ )

```



Results:

Simple difference:

```
> mean(bad$ydiff)
```

```
[1] -1.35
```

Reloop:

```
> with(badLong,p_loop(Y=y,Tr=z,Z=cbind(x),P=as.numeric(as.factor(m)),pred=p_ols_po))
```

```

      Estimate Std. Error  t value  Pr(>|t|)
ATE: -1.926757   0.110167 -17.48942 < 2.22e-16 ***
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> with(badLong,p_loop(Y=y,Tr=z,Z=cbind(x),P=as.numeric(as.factor(m)),pred=p_ols_v12))
```

```
      Estimate Std. Error  t value  Pr(>|t|)
ATE: -2.64601770  0.02190698 -120.7842 < 2.22e-16 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> with(badLong,p_loop(Y=y,Tr=z,Z=cbind(x),P=as.numeric(as.factor(m)),pred=p_ols_interp))
```

```
      Estimate Std. Error  t value  Pr(>|t|)
ATE: -2.64377740  0.02291407 -115.3779 < 2.22e-16 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Rebar:

```
> mean(bad$ydiff-bad$xdiff)
```

```
[1] -2.703525
```

OK but that was constructed to be bad. What if we constructed a new match based on the prognostic scores? (this is backwards but still)

```
> newmatch <- pairmatch(match_on(z~x,data=badLong),data=badLong)
```

```
> with(badLong,p_loop(Y=y,Tr=z,Z=cbind(x),P=as.numeric(as.factor(newmatch)),pred=p_ols_interp))
```

```
      Estimate Std. Error t value  Pr(>|t|)
ATE: -1.9232884  0.3299515  -5.829 5.576e-09 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Still worse