I introduced the notion of *sensitivity to unmeasured confounding* in the context of an observational data analysis. I argued that an estimate of an association between an observed exposure \(D\) and outcome \(Y\) is sensitive to unmeasured confounding if we can conceive of a reasonable alternative data generating process (DGP) that includes some unmeasured confounder that will generate the same observed distribution the observed data. I further argued that reasonableness can be quantified or parameterized by the two correlation coefficients \(\rho\_{UD}\) and \(\rho\_{UY}\), which measure the strength of the relationship of the unmeasured confounder \(U\) with each of the observed measures. Alternative DGPs that are characterized by high correlation coefficients can be viewed as less realistic, and the observed data could be considered less sensitive to unmeasured confounding. On the other hand, DGPs characterized by lower correlation coefficients would be considered more sensitive.

In fact, created an R package called treatSens to facilitate sensitivity analysis. I believe the discussion in these posts here is consistent with the PRIISM methodology, except treatSens is far more flexible (e.g. it can handle binary exposures) and provides more informative output than what I am describing. I am hoping that the examples and derivation of an equivalent DGP that I show here provide some additional insight into what sensitivity means.

OK – now we are ready to walk through the derivation of alternative DGPs for an observed data set.

**Two DGPs, same data**

Recall from Part 1 that we have an observed data model

\[ Y = k\_0 + k\_1D + \epsilon\_Y\]  
where \(\epsilon\_Y \sim N\left(0, \sigma^2\_Y\right)\). We are wondering if there is another DGP that could have generated the data that we have actually observed:

\[  
\begin{aligned}  
D &= \alpha\_0 + \alpha\_1 U + \epsilon\_D \\  
Y &= \beta\_0 + \beta\_1 D + \beta\_2 U + \epsilon\_{Y^\*},  
\end{aligned}  
\]

where \(U\) is some unmeasured confounder, and \(\epsilon\_D \sim N\left(0, \sigma^2\_D\right)\) and \(\epsilon\_{Y^\*} \sim N\left(0, \sigma^2\_{Y^\*}\right)\). Can we go even further and find an alternative DGP where \(D\) has no direct effect on \(Y\) at all?

\[  
\begin{aligned}  
D &= \alpha\_0 + \alpha\_1 U + \epsilon\_D \\  
Y &= \beta\_0 + \beta\_2 U + \epsilon\_{Y^\*},  
\end{aligned}  
\]

**\(\alpha\_1\) (and \(\sigma\_{\epsilon\_D}^2\)) derived from \(\rho\_{UD}\)**

In a simple linear regression model with a single predictor, the coefficient \(\alpha\_1\) can be specified directly in terms \(\rho\_{UD}\), the correlation between \(U\) and \(D\):

\[ \alpha\_1 = \rho\_{UD} \frac{\sigma\_D}{\sigma\_U}\]  
We can estimate \(\sigma\_D\) from the observed data set, and we can reasonably assume that \(\sigma\_U = 1\) (since we could always normalize the original measurement of \(U\)). Finally, we can specify a range of \(\rho\_{UD}\) (I am keeping everything positive here for simplicity), such that \(0 < \rho\_{UD} < 0.90\) (where I assume a correlation of \(0.90\) is at or beyond the realm of reasonableness). By plugging these three parameters into the formula, we can generate a range of \(\alpha\_1\)’s.

Furthermore, we can derive an estimate of the variance for \(\epsilon\_D\) ( \(\sigma\_{\epsilon\_D}^2\)) at each level of \(\rho\_{UD}\):

\[  
\begin{aligned}  
Var(D) &= Var(\alpha\_0 + \alpha\_1 U + \epsilon\_D) \\  
\\  
\sigma\_D^2 &= \alpha\_1^2 \sigma\_U^2 + \sigma\_{\epsilon\_D}^2 \\  
\\  
\sigma\_{\epsilon\_D}^2 &= \sigma\_D^2 – \alpha\_1^2 \; \text{(since } \sigma\_U^2=1)  
\end{aligned}  
\]

So, for each value of \(\rho\_{UD}\) that we generated, there is a corresponding pair \((\alpha\_1, \; \sigma\_{\epsilon\_D}^2)\).

**Determine \(\beta\_2\)**

In the addendum I go through a bit of an elaborate derivation of \(\beta\_2\), the coefficient of \(U\) in the alternative outcome model. Here is the bottom line:

\[  
\beta\_2 = \frac{\alpha\_1}{1-\frac{\sigma\_{\epsilon\_D}^2}{\sigma\_D^2}}\left( k\_1 – \beta\_1\right)  
\]

In the equation, we have \(\sigma^2\_D\) and \(k\_1\), which are both estimated from the observed data and the pair of derived parameters \(\alpha\_1\) and \(\sigma\_{\epsilon\_D}^2\) based on \(\rho\_{UD}\). \(\beta\_1\), the coefficient of \(D\) in the outcome model is a free parameter, set externally. That is, we can choose to evaluate all \(\beta\_2\)’s the are generated when \(\beta\_1 = 0\). More generally, we can set \(\beta\_1 = pk\_1\), where \(0 \le p \le 1\). (We could go negative if we want, though I won’t do that here.) If \(p=1\) , \(\beta\_1 = k\_1\) and \(\beta\_2 = 0\); we end up with the original model with no confounding.

So, once we specify \(\rho\_{UD}\) and \(p\), we get the corresponding triplet \((\alpha\_1, \; \sigma\_{\epsilon\_D}^2, \; \beta\_2)\).

**Determine \(\rho\_{UY}\)**

In this last step, we can identify the correlation of \(U\) and \(Y\), \(\rho\_{UY}\), that is associated with all the observed, specified, and derived parameters up until this point. We start by writing the alternative outcome model, and then replace \(D\) with the alternative exposure model, and do some algebraic manipulation to end up with a re-parameterized alternative outcome model that has a single predictor:

\[  
\begin{aligned}  
Y &= \beta\_0 + \beta\_1 D + \beta\_2 U + \epsilon\_Y^\* \\  
&= \beta\_0 + \beta\_1 \left( \alpha\_0 + \alpha\_1 U + \epsilon\_D \right) + \beta\_2 U + \epsilon\_Y^\* \\  
&=\beta\_0 + \beta\_1 \alpha\_0 + \beta\_1 \alpha\_1 U + \beta\_1 \epsilon\_D + \beta\_2 U +  
\epsilon\_Y^\* \\  
&=\beta\_0^\* + \left( \beta\_1 \alpha\_1 + \beta\_2 \right)U + \epsilon\_Y^+ \\  
&=\beta\_0^\* + \beta\_1^\*U + \epsilon\_Y^+ , \\  
\end{aligned}  
\]

where \(\beta\_0^\* = \beta\_0 + \beta\_1 \alpha\_0\), \(\beta\_1^\* = \left( \beta\_1 \alpha\_1 + \beta\_2 \right)\), and \(\epsilon\_Y^+ = \beta\_1 \epsilon\_D + \epsilon\_Y\*\).

As before, the coefficient in a simple linear regression model with a single predictor is related to the correlation of the two variables as follows:

\[  
\beta\_1^\* = \rho\_{UY} \frac{\sigma\_Y}{\sigma\_U}  
\]

Since \(\beta\_1^\* = \left( \beta\_1 \alpha\_1 + \beta\_2 \right)\),

\[  
\begin{aligned}  
\beta\_1 \alpha\_1 + \beta\_2 &= \rho\_{UY} \frac{\sigma\_Y}{\sigma\_U} \\  
\\  
\rho\_{UY} &= \frac{\sigma\_U}{\sigma\_Y} \left( \beta\_1 \alpha\_1 + \beta\_2 \right) \\  
\\  
&= \frac{\left( \beta\_1 \alpha\_1 + \beta\_2 \right)}{\sigma\_Y}  
\end{aligned}  
\]

**Determine \(\sigma^2\_{Y\*}\)**

In order to simulate data from the alternative DGPs, we need to derive the variation for the noise of the alternative model. That is, we need an estimate of \(\sigma^2\_{Y\*}\).

\[  
\begin{aligned}  
Var(Y) &= Var(\beta\_0 + \beta\_1 D + \beta\_2 U + \epsilon\_{Y^\*}) \\  
\\  
&= \beta\_1^2 Var(D) + \beta\_2^2 Var(U) + 2\beta\_1\beta\_2Cov(D, U) + Var(\epsilon\_{y\*}) \\  
\\  
&= \beta\_1^2 \sigma^2\_D + \beta\_2^2 + 2\beta\_1\beta\_2\rho\_{UD}\sigma\_D + \sigma^2\_{Y\*} \\  
\end{aligned}  
\]

So,

\[  
\sigma^2\_{Y\*} = Var(Y) – (\beta\_1^2 \sigma^2\_D + \beta\_2^2 + 2\beta\_1\beta\_2\rho\_{UD}\sigma\_D),  
\]

where \(Var(Y)\) is the variation of \(Y\) from the observed data. Now we are ready to implement this in R.

**Implementing in R**

If we have an observed data set with observed \(D\) and \(Y\), and some target \(\beta\_1\) determined by \(p\), we can calculate/generate all the quantities that we just derived.

Before getting to the function, I want to make a brief point about what we do if we have other *measured* confounders. We can essentially eliminate measured confounders by regressing the exposure \(D\) on the confounders and conducting the entire sensitivity analysis with the residual exposure measurements derived from this initial regression model. I won’t be doing this here, but if anyone wants to see an example of this, let me know, and I can do a short post.

OK – here is the function, which essentially follows the path of the derivation above:

altDGP <- function(dd, p) {

# Create values of rhoUD

dp <- data.table(p = p, rhoUD = seq(0.0, 0.9, length = 1000))

# Parameters estimated from data

dp[, `:=`(sdD = sd(dd$D), s2D = var(dd$D), sdY = sd(dd$Y))]

dp[, k1:= coef(lm(Y ~ D, data = dd))[2]]

# Generate b1 based on p

dp[, b1 := p \* k1]

# Determine a1

dp[, a1 := rhoUD \* sdD ]

# Determine s2ed

dp[, s2ed := s2D - (a1^2)]

# Determine b2

dp[, g:= s2ed/s2D]

dp <- dp[g != 1]

dp[, b2 := (a1 / (1 - g) ) \* ( k1 - b1 )]

# Determine rhoUY

dp[, rhoUY := ( (b1 \* a1) + b2 ) / sdY ]

# Eliminate impossible correlations

dp <- dp[rhoUY > 0 & rhoUY <= .9]

# Determine s2eyx

dp[, s2eyx := sdY^2 - (b1^2 \* s2D + b2^2 + 2 \* b1 \* b2 \* rhoUD \* sdD)]

dp <- dp[s2eyx > 0]

# Determine standard deviations

dp[, sdeyx := sqrt(s2eyx)]

dp[, sdedx := sqrt(s2ed)]

# Finished

dp[]

}

**Assessing sensitivity**

If we generate the same data set we started out with last post, we can use the function to assess the sensitivity of this association.

defO <- defData(varname = "D", formula = 0, variance = 1)

defO <- defData(defO, varname = "Y", formula = "1.5 \* D", variance = 25)

set.seed(20181201)

dtO <- genData(1200, defO)

In this first example, I am looking for the DGP with \(\beta\_1 = 0\), which is implemented as \(p = 0\) in the call to function altDGP. Each row of output represents an alternative set of parameters that will result in a DGP with \(\beta\_1 = 0\).

dp <- altDGP(dtO, p = 0)

dp[, .(rhoUD, rhoUY, k1, b1, a1, s2ed, b2, s2eyx)]

## rhoUD rhoUY k1 b1 a1 s2ed b2 s2eyx

## 1: 0.295 0.898 1.41 0 0.294 0.904 4.74 5.36

## 2: 0.296 0.896 1.41 0 0.295 0.903 4.72 5.50

## 3: 0.297 0.893 1.41 0 0.296 0.903 4.71 5.63

## 4: 0.298 0.890 1.41 0 0.297 0.902 4.69 5.76

## 5: 0.299 0.888 1.41 0 0.298 0.902 4.68 5.90

## ---

## 668: 0.896 0.296 1.41 0 0.892 0.195 1.56 25.35

## 669: 0.897 0.296 1.41 0 0.893 0.193 1.56 25.35

## 670: 0.898 0.296 1.41 0 0.894 0.191 1.56 25.36

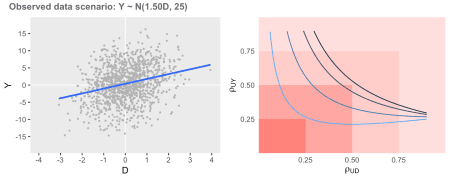
## 671: 0.899 0.295 1.41 0 0.895 0.190 1.56 25.36

## 672: 0.900 0.295 1.41 0 0.896 0.188 1.55 25.37

Now, I am creating a data set that will be based on four levels of \(\beta\_1\). I do this by creating a vector \(p = \; <0.0, \; 0.2, \; 0.5, \; 0.8>\). The idea is to create a plot that shows the curve for each value of \(p\). The most extreme curve (in this case, the curve all the way to the right, since we are dealing with positive associations only) represents the scenario where \(p = 0\) (i.e. \(\beta\_1 = 0\)). The curves moving to the left reflect increasing sensitivity as \(p\) increases.

dsenO <- rbindlist(lapply(c(0.0, 0.2, 0.5, 0.8),

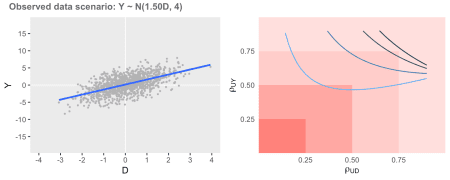
function(x) altDGP(dtO, x)))



I would say that in this first case the observed association is moderately sensitive to unmeasured confounding, as correlations as low as 0.5 would enough to erase the association.

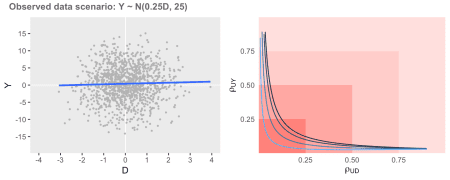
In the next case, if the association remains unchanged but the variation of \(Y\) is considerably reduced, the observed association is much less sensitive. However, it is still quite possible that the observed overestimation is at least partially overstated, as relatively low levels of correlation could reduce the estimated association.

defA1 <- updateDef(defO, changevar = "Y", newvariance = 4)



In this last scenario, variance is the same as the initial scenario, but the association is considerably weaker. Here, we see that the estimate of the association is extremely sensitive to unmeasured confounding, as low levels of correlation are required to entirely erase the association.

defA2 <- updateDef(defO, changevar = "Y", newformula = "0.25 \* D")



**treatSens package**

I want to show output generated by the treatSens package I referenced earlier. treatSens requires a formula that includes an outcome vector \(Y\), an exposure vector \(Z\), and at least one vector of measured of confounders \(X\). In my examples, I have included no measured confounders, so I generate a vector of independent noise that is not related to the outcome.

library(treatSens)

X <- rnorm(1200)

Y <- dtO$Y

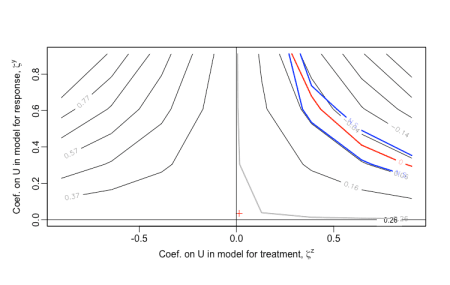
Z <- dtO$D

testsens <- treatSens(Y ~ Z + X, nsim = 5)

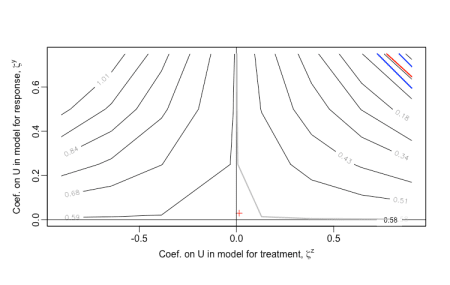
sensPlot(testsens)

Once treatSens has been executed, it is possible to generate a sensitivity plot, which looks substantively similar to the ones I have created. The package uses sensitivity parameters \(\zeta^Z\) and \(\zeta^Y\), which represent the coefficients of \(U\), the unmeasured confounder. Since treatSens normalizes the data (in the default setting), these coefficients are actually equivalent to the correlations \(\rho\_{UD}\) and \(\rho\_{UY}\) that are the basis of my sensitivity analysis. A important difference in the output is that treatSens provides uncertainty bands, and extends into regions of negative correlation. (And of course, a more significant difference is that treatSens is flexible enough to handle binary exposures, whereas I have not yet extended my analytic approach in that direction, and I suspect it is no possible for me to do so due to non-collapsibility of logistic regression estimands – I hope to revisit this in the future.)

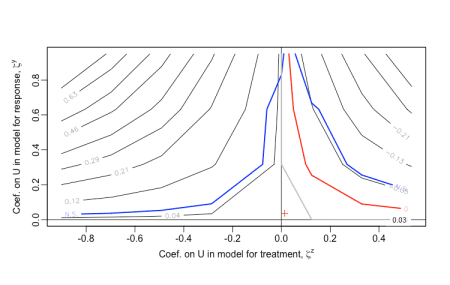
**Observed data scenario 1: \(\small{Y \sim N(1.50Z, \; 25)}\)**



**Observed data scenario 2: \(\small{Y \sim N(1.50Z, \; 4)}\)**



**Observed data scenario 3: \(\small{Y \sim N(0.25Z, \; 25)}\)**



**Addendum: Derivation of \(\beta\_2\)**

In case you want more detail on how we derive \(\beta\_2\) from the observed data model and assumed correlation parameters, here it is. We start by specifying the simple observed outcome model:

\[ Y = k\_0 + k\_1D + \epsilon\_Y\]

We can estimate the parameters \(k\_0\) and \(k\_1\) using this standard matrix solution:

\[ \; = (W^TW)^{-1}W^TY,\]

where \(W\) is the \(n \times 2\) design matrix:

\[ W = [\mathbf{1}, D]\_{n \times 2}.\]

We can replace \(Y\) with the alternative outcome model:

\[  
\begin{aligned}  
\; &= (W^TW)^{-1}W^T(\beta\_0 + \beta\_1 D + \beta\_2 U + \epsilon\_Y^\*) \\  
&= \;<\beta\_0, 0> + <0, \beta\_1> +\; (W^TW)^{-1}W^T(\beta\_2U) + \mathbf{0} \\  
&= \;<\beta\_0, \beta\_1> +\; (W^TW)^{-1}W^T(\beta\_2U)  
\end{aligned}  
\]

Note that

\[  
\begin{aligned}  
(W^TW)^{-1}W^T(\beta\_0) &= \; <\beta\_0,\; 0> \; \; and\\  
\\  
(W^TW)^{-1}W^T(\beta\_1D) &= \; <0,\; \beta\_1>.  
\end{aligned}  
\]

Now, we need to figure out what \((W^TW)^{-1}W^T(\beta\_2U)\) is. First, we rearrange the alternate exposure model:  
\[  
\begin{aligned}  
D &= \alpha\_0 + \alpha\_1 U + \epsilon\_D \\  
\alpha\_1 U &= D – \alpha\_0 – \epsilon\_D \\  
U &= \frac{1}{\alpha\_1} \left( D – \alpha\_0 – \epsilon\_D \right) \\  
\beta\_2 U &= \frac{\beta\_2}{\alpha\_1} \left( D – \alpha\_0 – \epsilon\_D \right)  
\end{aligned}  
\]

We can replace \(\beta\_2 U\):

\[  
\begin{aligned}  
(W^TW)^{-1}W^T(\beta\_2U) &= (W^TW)^{-1}W^T \left[ \frac{\beta\_2}{\alpha\_1} \left( D – \alpha\_0 – \epsilon\_D \right) \right] \\  
&= <-\frac{\beta\_2}{\alpha\_1}\alpha\_0, 0> + <0,\frac{\beta\_2}{\alpha\_1}>-\;\frac{\beta\_2}{\alpha\_1}(W^TW)^{-1}W^T \epsilon\_D \\  
&= <-\frac{\beta\_2}{\alpha\_1}\alpha\_0, \frac{\beta\_2}{\alpha\_1}>-\;\frac{\beta\_2}{\alpha\_1}(W^TW)^{-1}W^T \epsilon\_D \\  
\end{aligned}  
\]

And now we get back to \(\) :

\[  
\begin{aligned}  
\; &= \;<\beta\_0,\; \beta\_1> +\; (W^TW)^{-1}W^T(\beta\_2U) \\  
&= \;<\beta\_0-\frac{\beta\_2}{\alpha\_1}\alpha\_0, \; \beta\_1 + \frac{\beta\_2}{\alpha\_1}>-\;\frac{\beta\_2}{\alpha\_1}(W^TW)^{-1}W^T \epsilon\_D \\  
&= \;<\beta\_0-\frac{\beta\_2}{\alpha\_1}\alpha\_0, \; \beta\_1 + \frac{\beta\_2}{\alpha\_1}>-\;\frac{\beta\_2}{\alpha\_1}<\gamma\_0, \; \gamma\_1>  
\end{aligned}  
\]

where \(\gamma\_0\) and \(\gamma\_1\) come from regressing \(\epsilon\_D\) on \(D\):

\[ \epsilon\_D = \gamma\_0 + \gamma\_1 D\]  
so,

\[  
\begin{aligned}  
\; &= \;<\beta\_0-\frac{\beta\_2}{\alpha\_1}\alpha\_0 – \frac{\beta\_2}{\alpha\_1}\gamma\_0, \; \beta\_1 + \frac{\beta\_2}{\alpha\_1} – \frac{\beta\_2}{\alpha\_1}\gamma\_1 > \\  
&= \;<\beta\_0-\frac{\beta\_2}{\alpha\_1}\left(\alpha\_0 + \gamma\_0\right), \; \beta\_1 + \frac{\beta\_2}{\alpha\_1}\left(1 – \gamma\_1 \right) >  
\end{aligned}  
\]

Since we can center all the observed data, we can easily assume that \(k\_0 = 0\). All we need to worry about is \(k\_1\):

\[  
\begin{aligned}  
k\_1 &= \beta\_1 + \frac{\beta\_2}{\alpha\_1}\left(1 – \gamma\_1 \right) \\  
\frac{\beta\_2}{\alpha\_1}\left(1 – \gamma\_1 \right) &= k\_1 – \beta\_1 \\  
\beta\_2 &= \frac{\alpha\_1}{1-\gamma\_1}\left( k\_1 – \beta\_1\right)  
\end{aligned}  
\]

We have generated \(\alpha\_1\) based on \(\rho\_{UD}\), \(k\_1\) is a estimated from the data, and \(\beta\_1\) is fixed based on some \(p, \; 0 \le p \le 1\) such that \(\beta\_1 = pk\_1\). All that remains is \(\gamma\_1\):

\[  
\gamma\_1 = \rho\_{\epsilon\_D D} \frac{\sigma\_{\epsilon\_D}}{\sigma\_D}  
\]

Since \(D = \alpha\_0 + \alpha\_1 U + \epsilon\_D\) (and \(\epsilon\_D \perp \! \! \! \perp U\))

\[  
\begin{aligned}  
\rho\_{\epsilon\_D D} &= \frac{Cov(\epsilon\_D, D)}{\sigma\_{\epsilon\_D} \sigma\_D} \\  
\\  
&=\frac{Cov(\epsilon\_D, \;\alpha\_0 + \alpha\_1 U + \epsilon\_D )}{\sigma\_{\epsilon\_D} \sigma\_D} \\  
\\  
&= \frac{\sigma\_{\epsilon\_D}^2}{\sigma\_{\epsilon\_D} \sigma\_D} \\  
\\  
&= \frac{\sigma\_{\epsilon\_D}}{\sigma\_D}  
\end{aligned}  
\]

It follows that

\[  
\begin{aligned}  
\gamma\_1 &= \rho\_{\epsilon\_D D} \frac{\sigma\_{\epsilon\_D}}{\sigma\_D} \\  
\\  
&=\frac{\sigma\_{\epsilon\_D}}{\sigma\_D} \times \frac{\sigma\_{\epsilon\_D}}{\sigma\_D} \\  
\\  
&=\frac{\sigma\_{\epsilon\_D}^2}{\sigma\_D^2}  
\end{aligned}  
\]

So, now, we have all the elements to generate \(\beta\_2\) for a range of \(\alpha\_1\)’s and \(\sigma\_{\epsilon\_D}^2\)’s:

\[  
\beta\_2 = \frac{\alpha\_1}{1-\frac{\sigma\_{\epsilon\_D}^2}{\sigma\_D^2}}\left( k\_1 – \beta\_1\right)  
\]