Simulation codes for misperceptions 10

Misperception 10: Derive the cut-off point to obtain decorrelated RVs.

I considered the same set up.

$$Y = \beta_x X + \beta_z Z + \epsilon,$$
$$X = \lambda_z Z + \eta,$$

where $\epsilon \sim \mathrm{N}(0, \sigma_\epsilon)$, $\eta \sim \mathrm{N}(0, \sigma_\eta)$, and $Z \sim \mathrm{N}(0, 1)$. Parameters are chosen to acheieve $\mathrm{Var}(X) = 1$ and $\mathrm{Var}(Y) = 1$. We want to identify a cut-off point, a, so that $\mathrm{Cov}(X, Z|Y > a) = 0$ (or equivalently, $Cor(X, Z|Y > a) = 0$.

Since

$$\mathrm{Cov}(X, Z|Y > a) = \mathrm{Cov}(\lambda_z Z + \eta, Z|Y > a) = \lambda_z \mathrm{Var}(Z|Y > a) + \mathrm{Cov}(\eta, Z|Y > a),$$

I started with the multivariate normal for

$$\begin{pmatrix} Z \\ \eta \\ Y \end{pmatrix} \sim N_3 \left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & \beta_z + \beta_x\lambda_z \\ 0 & 1 & \beta_x \sigma_\eta^2 \\ \beta_z + \beta_x\lambda_z & \beta_x\sigma_\eta^2 & 1 \end{pmatrix} \right]$$

An other reason that I started with the joint distribution for $(Z, \eta, Y)^\top$ instead of $(Z, X, Y)^\top$ is because of the 0's in the covariance-variance matrix. I then plugged those variables into the tri-variate normal density, e.g., https://mathworld.wolfram.com/TrivariateNormalDistribution.html and derived the conditional joint density $(Z, \eta|Y > a)$. Let $\rho_1 = \beta_z + \beta_x\lambda_z$, $\phi(\cdot)$ be the density function for $\mathrm{N}(0, 1)$, and $\Phi(\cdot)$ be the cumulative distribution function for $\mathrm{N}(0, 1)$. After cumbersome algebra, I have the following components that are required to calculate $\mathrm{Cov}(X, Z|Y > a)$.

$$E(Z|Y > a) = \rho_1 \frac{\phi(-a)}{\Phi(-a)}$$

$$E(Z^2|Y > a) = 1 + \rho_1^2 a \frac{\phi(-a)}{\Phi(-a)}$$

$$Cov(\eta, Z|Y > a) = \rho_1 \beta_x \sigma_\eta^2 \frac{\phi(-a)}{\Phi(-a)} \left[a - \frac{\phi(-a)}{\Phi(-a)} \right] = \rho_1 \beta_x (1 - \lambda_z^2) \frac{\phi(-a)}{\Phi(-a)} \left[a - \frac{\phi(-a)}{\Phi(-a)} \right]$$

This implies

$$\mathrm{Var}(Z|Y > a) = 1 + \rho_1^2 \frac{\phi(-a)}{\Phi(-a)} \left[a - \frac{\phi(-a)}{\Phi(-a)} \right],$$

and

$$\begin{aligned}
\mathrm{Cov}(X, Z|Y > a) &= \lambda_z \left\{ 1 + \rho_1^2 \frac{\phi(-a)}{\Phi(-a)} \left[a - \frac{\phi(-a)}{\Phi(-a)} \right] \right\} + \rho_1 \beta_x (1 - \lambda_z^2) \frac{\phi(-a)}{\Phi(-a)} \left[a - \frac{\phi(-a)}{\Phi(-a)} \right] \\
&= \lambda_z + \frac{\phi(-a)}{\Phi(-a)} \left(a - \frac{\phi(-a)}{\Phi(-a)} \right) \rho_1 (\rho_1 \lambda_z + \beta_x - \beta_x \lambda_z) \\
&= \lambda_z + \frac{\phi(-a)}{\Phi(-a)} \left(a - \frac{\phi(-a)}{\Phi(-a)} \right) \rho_1 (\beta_z \lambda_z + \beta_x) \equiv \lambda_z + \frac{\phi(-a)}{\Phi(-a)} \left(a - \frac{\phi(-a)}{\Phi(-a)} \right) \rho_1 \rho_2,
\end{aligned}$$

which is consistent with Roger's derivation from the extended multivariate skew-normal. Here is the implementation for $\mathrm{Cov}(X, Z|Y > a)$ based on my derivation.

```r
> get_CovXZ <- function(bx, bz, lam, a) {
+   rr <- dnorm(-a) / pnorm(-a)
+   r1 <- bz + bx * lam
+   r2 <- bx + bz * lam
+   lam + rr * (a - rr) * r1 * r2
+ }
```

Following the simulation settings in Roger's implementation, I created a function to compare the different cut-points. First, we need a function to generate data. Then based on the generated data, we can find a empirically by doing an exhaustive search for $a \in [0, 5]$.

```r
> emp_a <- function(n, bx = .45, bz = .5, lam = .4) {
+   Z <- rnorm(n)
+   X <- lam * Z + rnorm(n, sd = sqrt(1 - lam^2))
+   Y <- bx * X + bz * Z +
+     rnorm(n, sd = sqrt(1 - bx^2 * (1 - lam^2) - (bx * lam + bz)^2))
+   a0 <- seq(0, 4, 1e-4)
+   cors <- sapply(a0, function(a) cor(X[Y > a], Z[Y > a]))
+   a0[which.min(abs(cors))]
+ }
```

The default parameters are based on the example in Roger's code. The following gives the average of 100 "empirical a's" when the sample sizes are 5000, 10000, and 50000. Note that I am using running the 100 replicates on multi-cores (parallel computing). The un-parallelized equivalence is included in comment.

```r
> library(parallel)
> cl <- makePSOCKcluster(detectCores())
> setDefaultCluster(cl)
> invisible(clusterExport(NULL, "emp_a"))
> summary(emp1 <- parSapply(NULL, 1:100, function(z) emp_a(5000)))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.7918  1.5211  1.8666  1.8814  2.2088  2.9247 
> summary(emp2 <- parSapply(NULL, 1:100, function(z) emp_a(10000)))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.092   1.585   1.902   1.927   2.155   3.364 
> summary(emp3 <- parSapply(NULL, 1:100, function(z) emp_a(50000)))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.366   1.876   2.103   2.232   2.541   3.499 
> stopCluster(cl)
> ## summary(emp <- replicate(100, emp_a(5000)))
```

With the same regression parameters, here are the estimates of a under Roger's and my derivation

```r
> uniroot(f = function(x) get_CovXZ(.45, .5, .4, x), c(0, 7))$root
[1] 2.36493
```

Although the empirical cut-off points seem to approach the derived cut point as the sample sizes increase, I think we need larger sample sizes to verify the derived cut-off point numerically.

For this sub-misperception, I think it is interesting to discuss the existence of such a a that makes $\mathrm{Cov}(X, Z|Y > a) = 0$. Recall

$$\mathrm{Cov}(X, Z|Y > a) = \lambda_z + \frac{\phi(-a)}{\Phi(-a)} \left(a - \frac{\phi(-a)}{\Phi(-a)} \right) \rho_1 \rho_2.$$

Because

$$\Phi(-a) = 1 - \Phi(a) = \int_a^\infty \phi(x)dx = -\frac{\phi(x)}{x}\bigg|_a^\infty - \int_a^\infty \frac{\phi(x)}{x^2}du = \frac{\phi(a)}{a} - \int_a^\infty \frac{\phi(x)}{x^2}du < \frac{\phi(a)}{a},$$

we have

$$a - \frac{\phi(-a)}{\Phi(-a)} < 0.$$

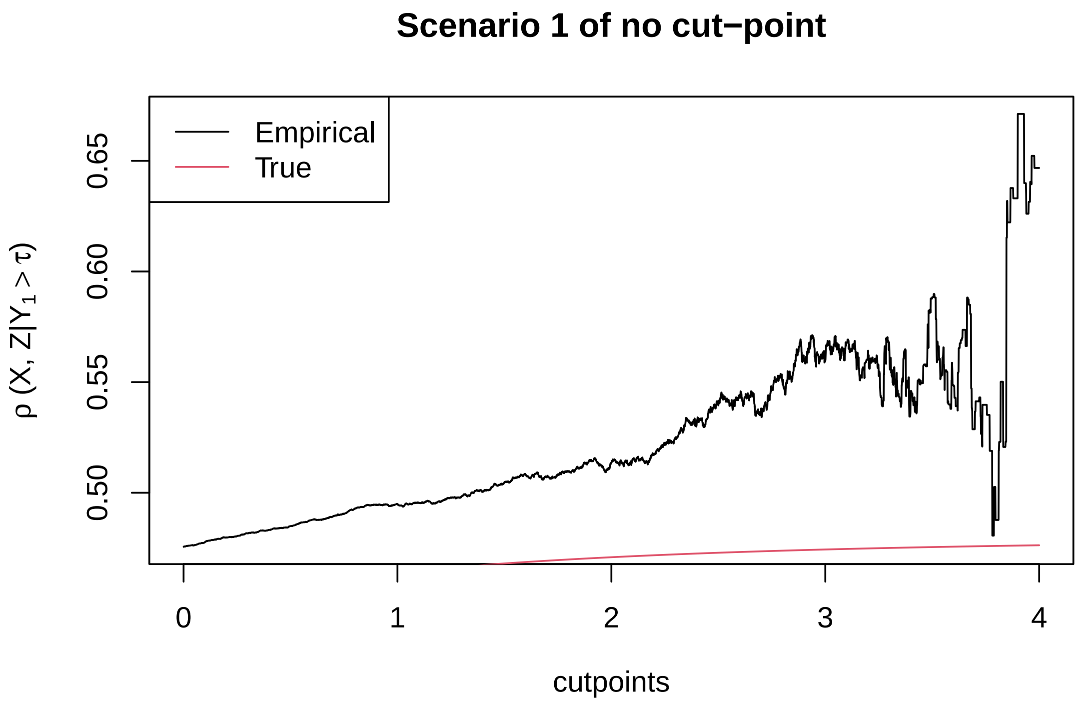
Without loss of generality, let's assume λ_z is positive. Since $\phi(-a)/\Phi(-a)$ is also positive, it is possible that $\mathrm{Cov}(X, Z|Y > a)$ will never decrease to 0 if

- one ρ_1 and ρ_2 is negative and the other one is positive.
- one ρ_1 and ρ_2 is zero.

Those scenarios are possible to occur when β_x and β_z have opposite signs. For example, when $\beta_x = -0.45$, $\beta_z = 0.5$, and $\lambda_z = 0.4$, we have $\rho_1 = \beta_z + \beta_x\lambda_z = 0.5 - 0.45 \cdot 0.4 = 0.32 > 0$ and $\rho_1 = \beta_x + \beta_z\lambda_z = -0.45 + 0.5 \cdot 0.4 = -0.25 < 0$. The following codes $\mathrm{Cov}(X, Z|Y > a)$ against a when $\beta_x = -0.45$, $\beta_z = 0.5$, and $\lambda_z = 0.4$.

```
> simDat <- function(n, bx = -.45, bz = .5, lam = .4) {
+   Z <- rnorm(n)
+   X <- lam * Z + rnorm(n, sd = sqrt(1 - lam^2))
+   Y <- bx * X + bz * Z +
+     rnorm(n, sd = sqrt(1 - bx^2 * (1 - lam^2) - (bx * lam + bz)^2))
+   data.frame(Y = Y, X = X, Z = Z)
+ }
> set.seed(1); dat <- simDat(500000)
```

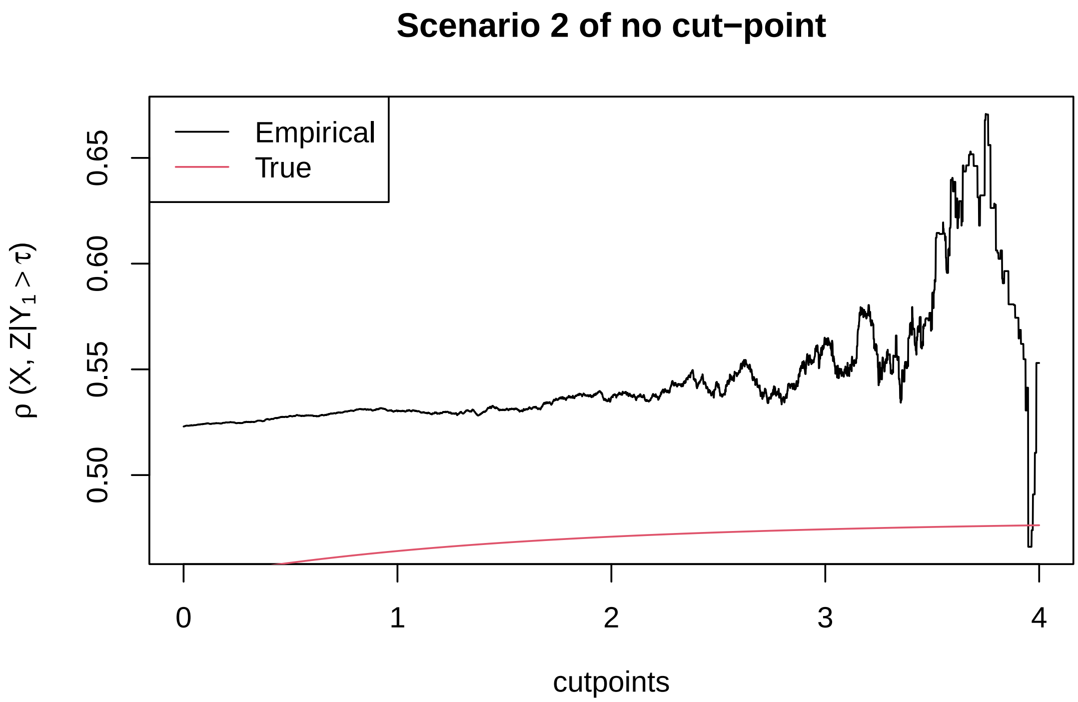
```
> a0 <- seq(0, 4, 1e-3)
> plot(a0, sapply(a0, function(a) with(subset(dat, Y > a), cor(X, Z))), 'l',
+   xlab = expression("cutpoints"), ylab = expression(rho~"(X, Z|"*Y[1]>tau*")"),
+   main = "Scenario 1 of no cut-point")
> lines(a0, sapply(a0, function(x) get_CovXZ(-0.45, 0.5, 0.4, x)), col = 2)
> legend("topleft", legend = c("Empirical", "True"), lty = 1, col = 1:2)
```



The above figure shows one example when there is no cut-point. Another example is to have $\beta_x = -0.5$, $\beta_z = 0.25$, and $\lambda_z = 0.5$, implying $\rho_1 = 0$.

```
> set.seed(2); dat2 <- simDat(500000, bx = -.5, bz = .25, lam = .5)
```

```
> plot(a0, sapply(a0, function(a) with(subset(dat2, Y > a), cor(X, Z))), 'l',
+   xlab = expression("cutpoints"), ylab = expression(rho~"(X, Z|"*Y[1]>tau*")"),
+   main = "Scenario 2 of no cut-point")
> lines(a0, sapply(a0, function(x) get_CovXZ(-0.45, 0.5, 0.4, x)), col = 2)
> legend("topleft", legend = c("Empirical", "True"), lty = 1, col = 1:2)
```



I think it will be interesting to look at scenarios when the cut-off point exists. Since

$$\frac{\phi(-a)}{\Phi(-a)}\left[a - \frac{\phi(-a)}{\Phi(-a)}\right] \to -1 \text{ as } a \to \infty,$$

$Cov(X, Z|Y > a)$ goes to $\lambda_z - \rho_1\rho_2$. By mid value theorem, a cut-off exists if $\lambda_z < \rho_1\rho_2$.