

HUDM6026 Homework_04

Seng Lei and Chenguang Pan

Feb 05, 2023

PART 01: Homework 03's Solution

Step 1, input the given variance-covariance matrix; Note, since these ten variables follow the standard normal distribution, the values in correlation matrix is same to the value in var-cov matrix.

```
> library(mvtnorm)
> cov = matrix(c(1,0,0,0,0.2,0,0,0,0,0,
+               0,1,0,0,0,0.9,0,0,0,0,
+               0,0,1,0,0,0,0.2,0,0,
+               0,0,0,1,0,0,0,0.9,0,
+               0.2,0,0,0,1,0,0,0,0,0,
+               0,0.9,0,0,0,1,0,0,0,0,
+               0,0,0,0,0,0,1,0,0,0,
+               0,0,0.2,0,0,0,0,1,0,0,
+               0,0,0,0.9,0,0,0,0,1,0,
+               0,0,0,0,0,0,0,0,0,1),10,10)
> # randomly generate one row of data to test this matrix
> # X <- rmvnorm(1,sigma = cov) # looking good
```

Step 2, write a function to generate 500 observations

```
> dat_gen <- function(n, cov){
+   return(rmvnorm(n, mean=rep(0,10), sigma=cov))}
> s1 <- dat_gen(500, cov)
> head(s1) # looking good
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	-1.4495647	2.4304171	-0.5550750	-1.7774593	-2.28458843	2.5885841
[2,]	-1.5535961	-0.9980552	-0.2887975	0.1971154	-0.58816164	-0.3058877
[3,]	0.5264633	-0.7758964	1.0103404	-0.3780678	-0.35108755	-0.8565865
[4,]	2.0188461	-1.4394062	1.1914730	-1.9042487	-0.49660141	-1.2617427
[5,]	0.4331412	-1.0096178	1.9452516	-0.1053070	0.08668391	-1.1676213
[6,]	-0.6843456	-1.5652379	0.2742782	0.5308587	0.63274537	-0.5509560

	[,7]	[,8]	[,9]	[,10]
[1,]	-0.01079033	0.6198942	-1.54020439	1.35558351
[2,]	0.18792892	-1.3272023	-0.06897668	-1.67448711
[3,]	0.88319683	-1.4548352	-0.16851130	0.79565520
[4,]	0.51101058	0.6041980	-1.96296845	-1.20586395
[5,]	0.04052240	-0.3339130	-0.34540695	-0.82768851
[6,]	-0.09529678	-0.5981320	1.13041165	-0.04900414

Step 3, dicotormize the 1st, 3rd, 5th, 6th, 8th, and 9th variable within a for-loop.

```

> for (i in c(1,3,5,6,8,9)) {
+   s1[,i] <- ifelse(s1[,i] > mean(s1[,i]), 1, 0)
+ }
> head(s1)
      [,1]      [,2] [,3]      [,4] [,5] [,6]      [,7] [,8] [,9]
[1,]    0  2.4304171    0 -1.7774593    0    1 -0.01079033    1    0
[2,]    0 -0.9980552    0  0.1971154    0    0  0.18792892    0    0
[3,]    1 -0.7758964    1 -0.3780678    0    0  0.88319683    0    0
[4,]    1 -1.4394062    1 -1.9042487    0    0  0.51101058    1    0
[5,]    1 -1.0096178    1 -0.1053070    1    0  0.04052240    0    0
[6,]    0 -1.5652379    1  0.5308587    1    0 -0.09529678    0    1
      [,10]
[1,]  1.35558351
[2,] -1.67448711
[3,]  0.79565520
[4,] -1.20586395
[5,] -0.82768851
[6,] -0.04900414

```

PART 02: Homework 03 Continued

2.1 Choose the Propensity Score Models

For this homework, I choose the scenario B:

$$Pr[A = 1|W_i] = (1 + e^{-(\beta_0 + \beta_1 W_1 + \beta_2 W_2 + \beta_3 W_3 + \beta_4 W_4 + \beta_5 W_5 + \beta_6 W_6 + \beta_7 W_7 + \beta_2 W_2 W_2)})^{-1}$$

.

2.2 Using BinNor to Generate Observations

In the initial stage for data generating, one problem is if we first generate data based on the correlation matrix and then dichotomize some of the data according to the means, the final correlation will be attenuated comparing to the original one. Based on the Prof.Keller's in-class notes, here I choose the function **BinNor**.

```

> library(BinNor)
> # import the cor matrix
> cmat <- matrix(c(1,0,0,0,0.2,0,0,0,0,0,0,
+                 0,1,0,0,0,0.9,0,0,0,0,0,
+                 0,0,1,0,0,0,0,0.2,0,0,
+                 0,0,0,1,0,0,0,0,0.9,0,
+                 0.2,0,0,0,1,0,0,0,0,0,
+                 0,0.9,0,0,0,1,0,0,0,0,
+                 0,0,0,0,0,0,1,0,0,0,
+                 0,0,0.2,0,0,0,0,1,0,0,
+                 0,0,0,0.9,0,0,0,0,1,0,
+                 0,0,0,0,0,0,0,0,0,1),10,10)
>
> # always report warning: All correlations must be in feasible range!
> sigma.star <- compute.sigma.star(no.bin = 6,
+                                   no.nor = 4,
+                                   prop.vec.bin = c(0.5,0.5,0.5,0.5,0.5,0.5),

```

```

+                               corr.mat = cmat)
>
> mydata <- jointly.generate.binary.normal(no.rows = 500,
+                                         no.bin = 6,
+                                         no.nor = 4,
+                                         prop.vec.bin = c(0.5,0.5,0.5,0.5,0.5,0.5),
+                                         mean.vec.nor = c(0,0,0,0),
+                                         var.nor = c(1,1,1,1),
+                                         sigma.star = sigma.star$sigma_star,
+                                         continue.with.warning = T)

```

By setting the chunk's parameter `eval = FALSE`, the code chunk above won't run. But after several tries, I can't figure out how generate the required data through this `BinNor`. I temporarily skipped this and use the attenuated data to continue this script.

2.3 Augmented the Data Structure

Augment your data generation function so that the 11th column is the probability of exposure (i.e., the propensity score, `ps`) based on the scenario you pick, the 12th column is the dichotomous exposure variable `A`, and the 13th and final column is the outcome variable `Y`.

```

> # calculate the propensity scores
> R <- 1000
> obs <- 500
> data_list <- replicate(n = R,
+                         expr = dat_gen(obs, cov = cov),
+                         simplify = FALSE)

```