Problem 3

```
set.seed(123)
( K <- cbind(c(10,7,7,0),c(7,20,0,7),c(7,0,30,7),c(0,7,7,40)) )
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   10    7    7    0
## [2,]    7   20    0    7
## [3,]    7    0   30    7
## [4,]    0    7    7   40
```

```
data <- as.data.frame(mvrnorm(n=10000,mu=c(0,0,0,0),Sigma=solve(K)))

colnames(data) <- c("X1","X2","X3","X4")
```

Conditional independency

It represents following independencies:

$X_1 \perp\!\!\!\perp X_4|X_2, X_3$ and $X_2 \perp\!\!\!\perp X_3|X_1, X_4$

The corresponding graph

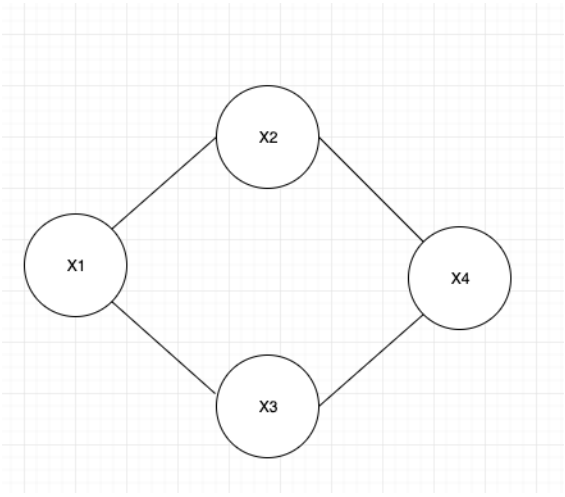


Figure 1: fig 3-1

Fit with OLS

```
lmodel = lm(X1 ~ X4 + X2 + X3, data=data)
summary(lmodel)
```

```
##
## Call:
## lm(formula = X1 ~ X4 + X2 + X3, data = data)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.36729 -0.21127  0.00304  0.21389  1.20994
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.001934   0.003141   0.616    0.538
## X4           0.007927   0.020037   0.396    0.692
## X2          -0.682729   0.012203 -55.950   <2e-16 ***
## X3          -0.695282   0.015540 -44.741   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3141 on 9996 degrees of freedom
## Multiple R-squared:  0.4564, Adjusted R-squared:  0.4563
## F-statistic:  2798 on 3 and 9996 DF,  p-value: < 2.2e-16
```

X4 is not significant while X2 and X3 are. This means X4 and X1 is independent given X2 and X3.

```
lmodel = lm(X2 ~ X3 + X1 + X4, data=data)
summary(lmodel)
```

```
##
## Call:
## lm(formula = X2 ~ X3 + X1 + X4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90282 -0.15318  0.00188  0.15342  0.85952
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.001141   0.002247   0.508    0.612
## X3           0.012316   0.012177   1.011    0.312
## X1          -0.349303   0.006243 -55.950   <2e-16 ***
## X4          -0.352810   0.013891 -25.398   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2246 on 9996 degrees of freedom
## Multiple R-squared:  0.3841, Adjusted R-squared:  0.3839
## F-statistic:  2078 on 3 and 9996 DF,  p-value: < 2.2e-16
```

X3 is not significant while X1 and X4 are. This means X2 and X3 is independent given X1 and X4.

Fit with gRim

cannot install package, **remember to do it later**

```
glist <- list( 'X1', 'X2', 'X3', 'X4' )
ddd <- cov.wt(data, method="ML")
fit <- ggmfit(ddd$cov, ddd$n.obs, glist) # Estimate parameters using IPF
fit$K
```

```
##          X1       X2       X3       X4
```

```
## X1 5.513255  0.00000  0.00000  0.00000
## X2 0.000000 12.21077  0.00000  0.00000
## X3 0.000000  0.00000 20.54787  0.00000
## X4 0.000000  0.00000  0.00000 33.73434
```

It did not work. K has more elements equal to zero than the original one.

Problem 4

```
set.seed(123)
( Sig <- cbind(c(3,-1.4,0,0),c(-1.4,3,1.4,1.4),c(0,1.4,3,0),c(0,1.4,0,3)) )
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  3.0 -1.4  0.0  0.0
## [2,] -1.4  3.0  1.4  1.4
## [3,]  0.0  1.4  3.0  0.0
## [4,]  0.0  1.4  0.0  3.0
```

```
data <- as.data.frame(mvrnorm(n=10000,mu=c(0,0,0,0),Sigma=Sig))
colnames(data) <- c("X1","X2","X3","X4")
```

a)

Correlation represented by graph

$X_1 \perp\!\!\!\perp X_3$ $X_1 \perp\!\!\!\perp X_4$ $X_2 \perp\!\!\!\perp X_4$ and they are not independent given X_2

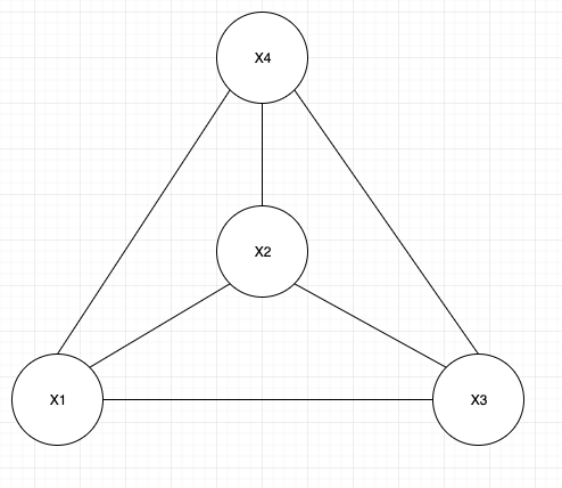
Correlation Matrix

```
solve(Sig)
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,]  0.5427350  0.4487179 -0.2094017 -0.2094017
## [2,]  0.4487179  0.9615385 -0.4487179 -0.4487179
## [3,] -0.2094017 -0.4487179  0.5427350  0.2094017
## [4,] -0.2094017 -0.4487179  0.2094017  0.5427350
```

b)

The moralized graph looks like



Every element of the precision matrix is not equal to 0 because every vertex is adjacent to another one.

It does not imply the correlation suggested in (a)

c)

```
glist <- list( 'X1', 'X2', 'X3', 'X4' )
ddd <- cov.wt(data, method="ML")
fit <- ggmfit(ddd$cov, ddd$n.obs, glist) # Estimate parameters using IPF
solve(fit$K)
```

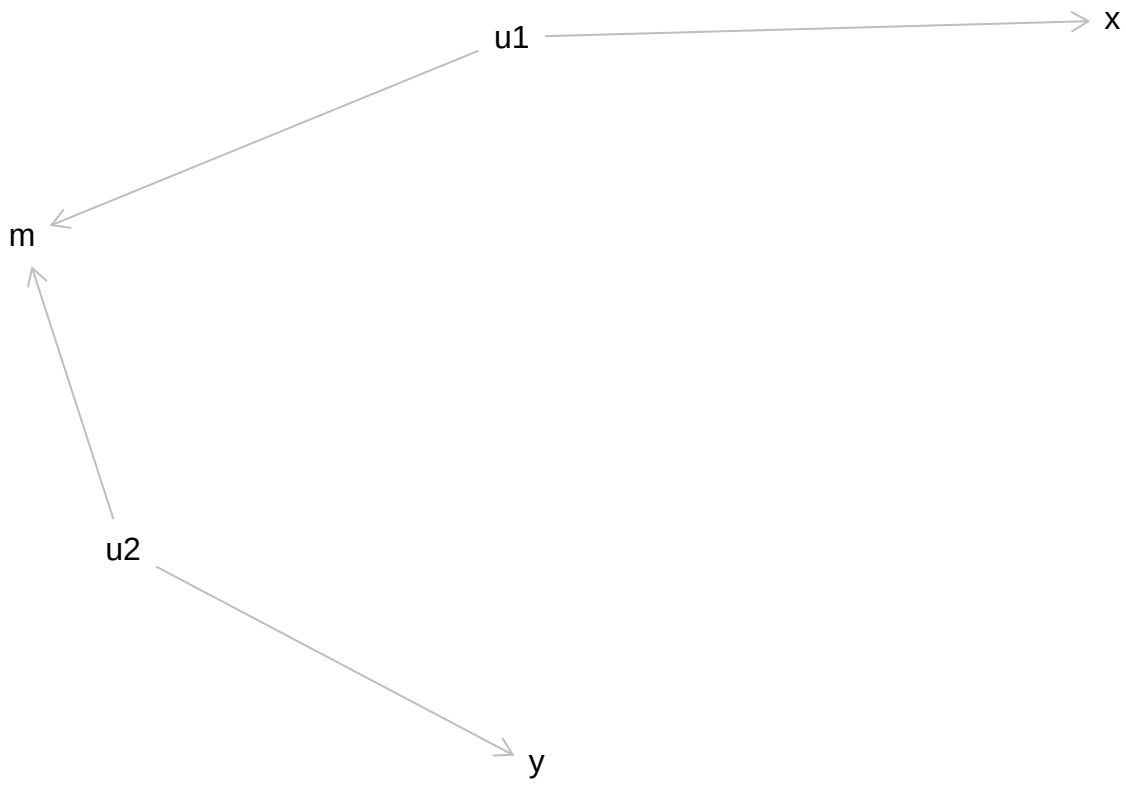
```
##          X1       X2       X3       X4
## X1 2.991722 0.000000 0.000000 0.000000
## X2 0.000000 2.959982 0.000000 0.000000
## X3 0.000000 0.000000 2.966525 0.000000
## X4 0.000000 0.000000 0.000000 3.077011
```

It is different from original covariance matrix as the elements on the diagonal are not the same.

Problem 5

```
g <- dagitty( "dag{ x <- u1; u1 -> m <- u2 ; u2 -> y }" )
df = simulateSEM(g, N = 1000, standardized = TRUE)
plot(g)
```

```
## Plot coordinates for graph not supplied! Generating coordinates, see ?coordinates for how to set you
```




```
reg = lm(y ~ x + m, data = df)
summary(reg)
```

```
##
## Call:
## lm(formula = y ~ x + m, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89042 -0.68302 -0.03076  0.67329  3.15642
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02718    0.03113  -0.873   0.3829
## x           -0.06637    0.03043  -2.181   0.0294 *
## m            0.17789    0.03119   5.703 1.55e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9843 on 997 degrees of freedom
## Multiple R-squared:  0.03459,    Adjusted R-squared:  0.03265
## F-statistic: 17.86 on 2 and 997 DF,  p-value: 2.398e-08
```

```
reg = lm(y ~ u2, data = df)
summary(reg)
```

```
##
## Call:
## lm(formula = y ~ u2, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6370 -0.5855 -0.0396  0.6020  3.1702
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.01729    0.02861  -0.604    0.546
## u2           0.44110    0.02941  14.999   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9045 on 998 degrees of freedom
## Multiple R-squared:  0.184,  Adjusted R-squared:  0.1831
## F-statistic:   225 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("graph")
```

```
## Bioconductor version 3.15 (BiocManager 1.30.18), R 4.2.1 (2022-06-23)
```

```
## Warning: package(s) not installed when version(s) same as current; use `force = TRUE` to
##   re-install: 'graph'
```