Answers to Odd-Numbered Exercises for Chapter 26 Fox, *Applied Regression Analysis and Generalized Linear Models, Third Edition* (Sage, 2016)

John Fox

Answers last modified: 2023-03-05

This document provides worked-out answers to the odd-numbered exercises in bonus Chapter 26 on causal directed acyclic graphs (DAGs) (excluding the data-analysis exercises on the website for the text). The answers to both odd- and even-numbered questions, are available from the author for instructors using the text in a college or university class.

Although the text is written to be software-neutral, some of the exercises require the use of statistical software. I employed the R statistical computing environment (R Core Team, 2022) for these exercises, particularly the **dagitty** package (Textor et al., 2017),[1] but you should feel free to substitute other appropriate statistical software for R. See Fox and Wesberg (2019) for an introduction to R in the context of regression analysis that largely overlaps with the text.

[1] I used the **dagitty** package in preparing the chapter, but with a cusomized `plot()` method to draw DAGs for `"dagitty"` objects. The plots of DAGs in these exercise answers are prepared with the `plot()` method from the **daggity** package.

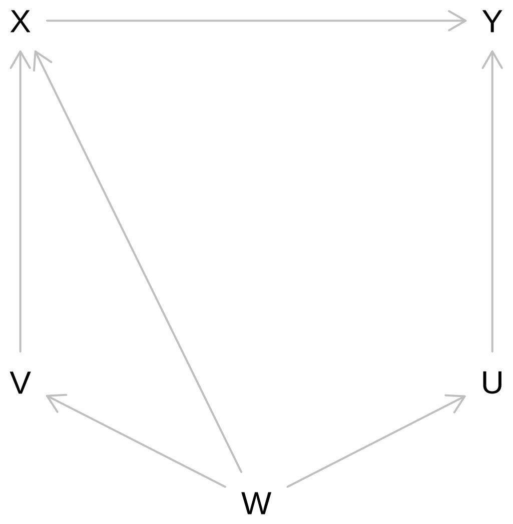
Exercises for Chapter 26

Exercise 26.1

There are two back-door paths: $X \longleftarrow V \longleftarrow W \longrightarrow U \longrightarrow Y$ and $X \longleftarrow W \longrightarrow U \longrightarrow Y$. Both can be blocked by controlling either for W or for U. The latter is farther from X and closer to Y, so it should provide the most efficient estimator. There are many other possibilities that control for more than one variable, for example, controlling for both W and U or for U and V, but if the DAG is correct, these would provide relatively inefficient, if still unbiased, estimators of the effect of X on Y.

Here's what **dagitty** says:

```
> library("dagitty")
> g <- dagitty('dag{
+   X -> Y
+   X <- V
+   U -> Y
+   W -> X
+   V <- W -> U
+   X [exposure,pos="-0.5, -0.5"]
+   Y [outcome,pos="0.5, -0.5"]
+   V [pos="-0.5, 0.25"]
+   U [pos="0.5, 0.25"]
+   W [pos="0, 0.5"]
+ }')
> plot(g)
```



```
> adjustmentSets(g, type="minimal")
{ U }
{ W }
> adjustmentSets(g, type="all")
{ U }
{ U, V }
{ W }
{ U, W }
{ V, W }
{ U, V, W }
```

Exercise 26.3

(a) Here are my computations, with the model regressing Y on X and W added:

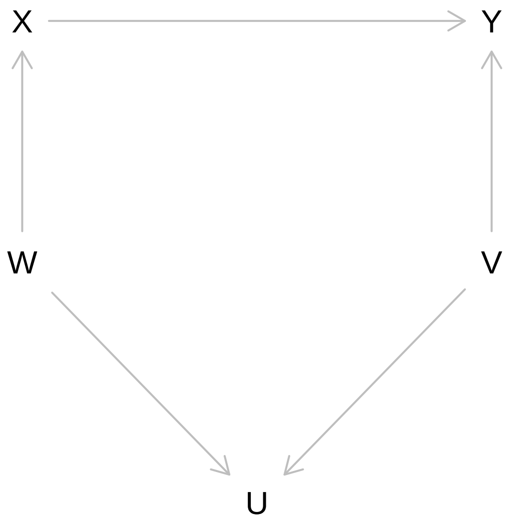
```
> library("car") # for compareCoefs()

Loading required package:  carData

> # a convenience function to compare residual standard errors:
> compareSigmas <- function(..., digits=4){
+   models <- list(...)
+   sigmas <- sapply(models, function(m) summary(m)$sigma)
+   print(sigmas, digits=digits)
+   return(invisible(sigmas))
+ }
> set.seed(3708531) # for reproducibility
> n <- 1000
> w <- rnorm(n)
> v <- rnorm(n)
> x <- w + rnorm(n, sd=0.25)
> y <- x + v + rnorm(n, sd=0.25)
> u <- w + v + rnorm(n, sd=0.25)
> m1 <- lm(y ~ x)
> m2 <- lm(y ~ x + u)
> m3 <- lm(y ~ x + v)
> m4 <- lm(y ~ x + u + v)
> m5 <- lm(y ~ x + u + w)
> m6 <- lm(y ~ x + u + v + w)
> m7 <- lm(y ~ x + w)
> compareCoefs(m1, m2, m3, m4, m5, m6, m7)
Calls:
1: lm(formula = y ~ x)
2: lm(formula = y ~ x + u)
3: lm(formula = y ~ x + v)
4: lm(formula = y ~ x + u + v)
5: lm(formula = y ~ x + u + w)
6: lm(formula = y ~ x + u + v + w)
7: lm(formula = y ~ x + w)

             Model 1  Model 2  Model 3  Model 4  Model 5  Model 6  Model 7
(Intercept) -0.02700  0.00377  0.00594  0.00600  0.00714  0.00590 -0.02735
SE           0.03273  0.01288  0.00809  0.00808  0.01096  0.00808  0.03269

x            1.03263  0.16610  0.99763  0.95616  0.93035  0.93883  0.82083
SE           0.03220  0.01727  0.00796  0.02288  0.04177  0.03082  0.12465

u                      0.8882            0.0441   0.9458   0.0243         
SE                     0.0120            0.0228   0.0106   0.0328         

v                              0.99280  0.94867           0.96812         
SE                             0.00802  0.02420           0.03350         

w                                                -0.8766   0.0385   0.2263
SE                                                0.0449   0.0458   0.1287

> compareSigmas(m1, m2, m3, m4, m5, m6, m7)
```

```
[1] 1.0349 0.4071 0.2558 0.2554 0.3463 0.2555 1.0338
```

Adding the model regressing Y on X and W (`m7`) nicely makes the point that unnecessarily controlling for a cause of X, as here, serves to inflate the standard error of the coefficient of X by increasing collinearity without reducing the error variance. We therefore don't bias the estimator of this coefficient, but we make the etimator much less precise. Contrast this with model 3, which regresses Y on X and V. The latter is a direct cause of Y; including it as a covariate reduces the error variance and thus increases the precision of the estimator of the coefficient of X .

Here is **daggity**'s analysis of the DAG for this exercise:

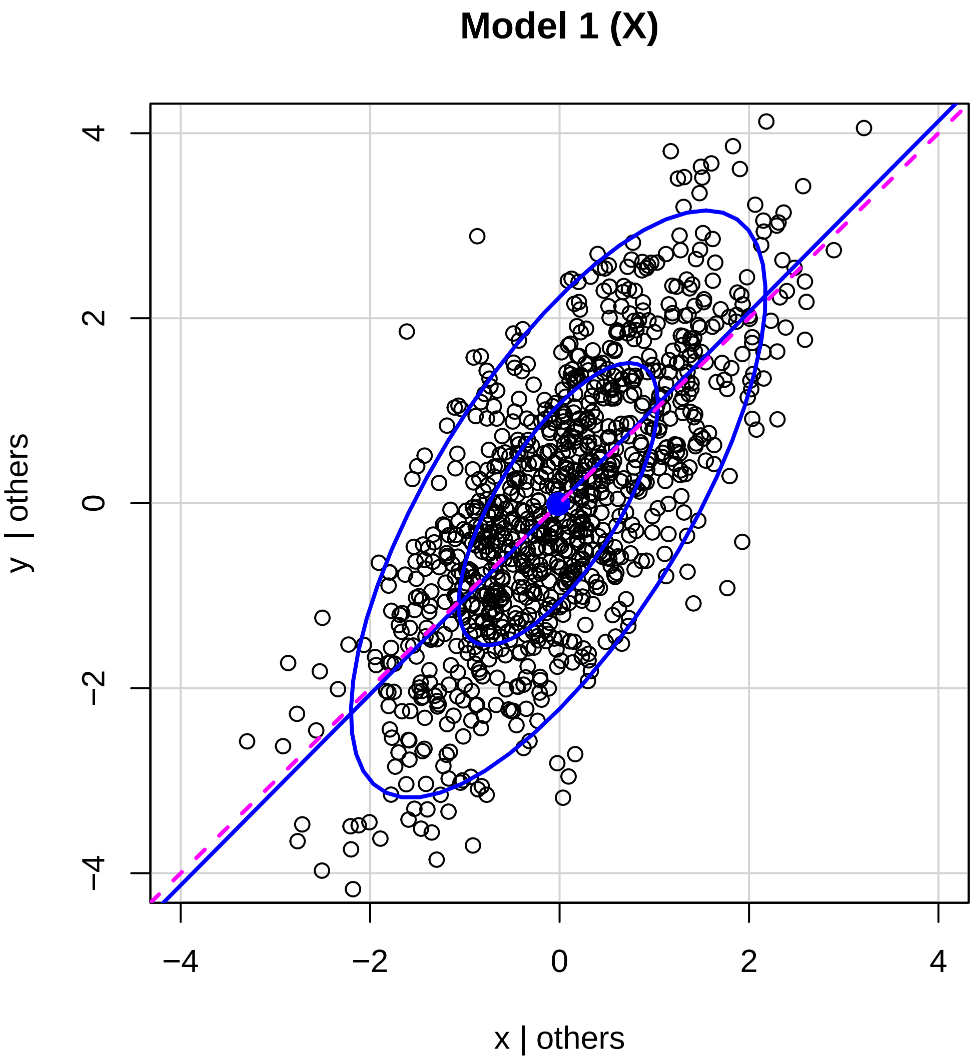
```
> library("dagitty")
> g <- dagitty('dag{
+   W -> X -> Y <- V
+   W -> U <- V
+   X [exposure,pos="-0.5, -0.5"]
+   Y [outcome,pos="0.5, -0.5"]
+   W [pos="-0.5, 0"]
+   V [pos="0.5, 0"]
+   U [pos="0, 0.5"]
+ }')
> plot(g)
```



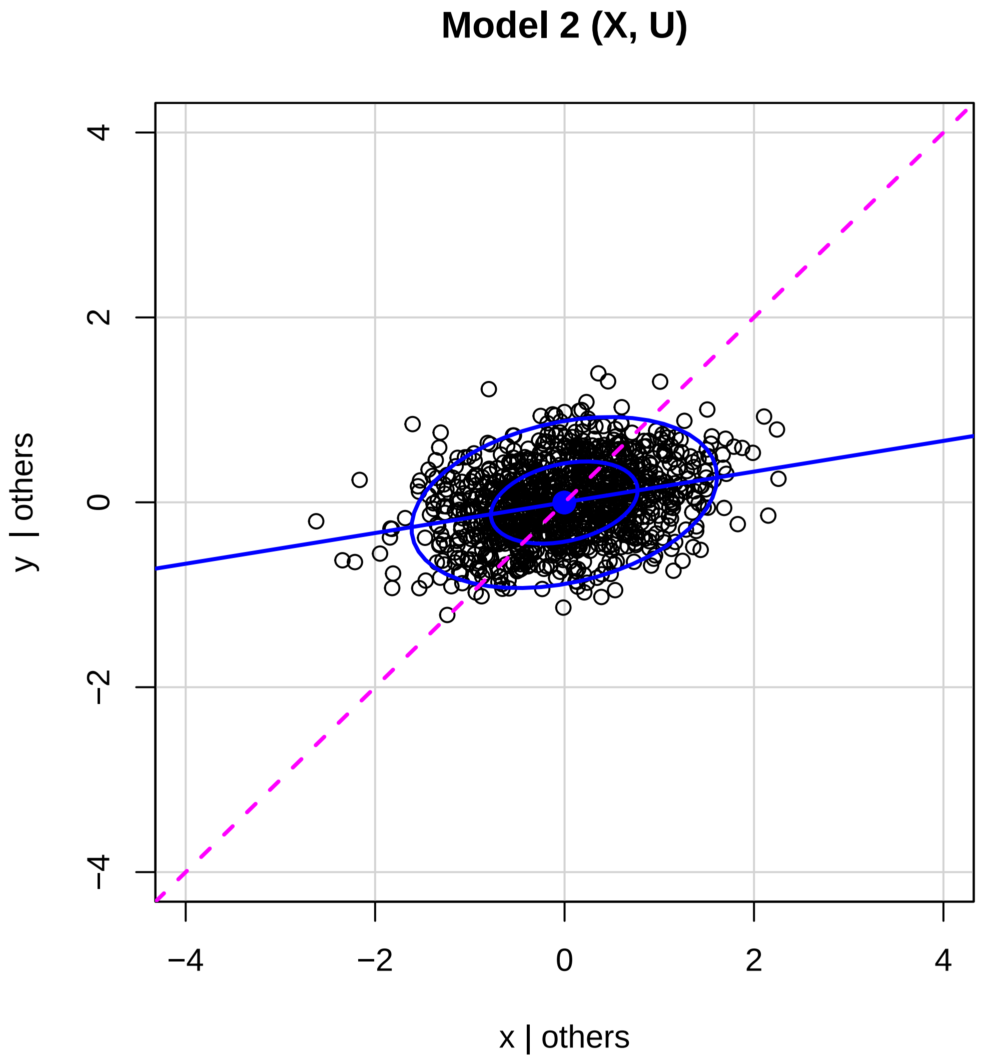
```
> adjustmentSets(g, type="minimal")
 {}
> adjustmentSets(g, type="all")
 {}
{ V }
{ U, V }
{ W }
{ U, W }
{ V, W }
{ U, V, W }
> isCollider(g, "W", "U", "V")
[1] TRUE
```

(b) Here are the added-variable plots for X in models 1 (with X alone), 2 (with X and the collider U), 3 (X and the covariate V), and 5 (X, U, and W):

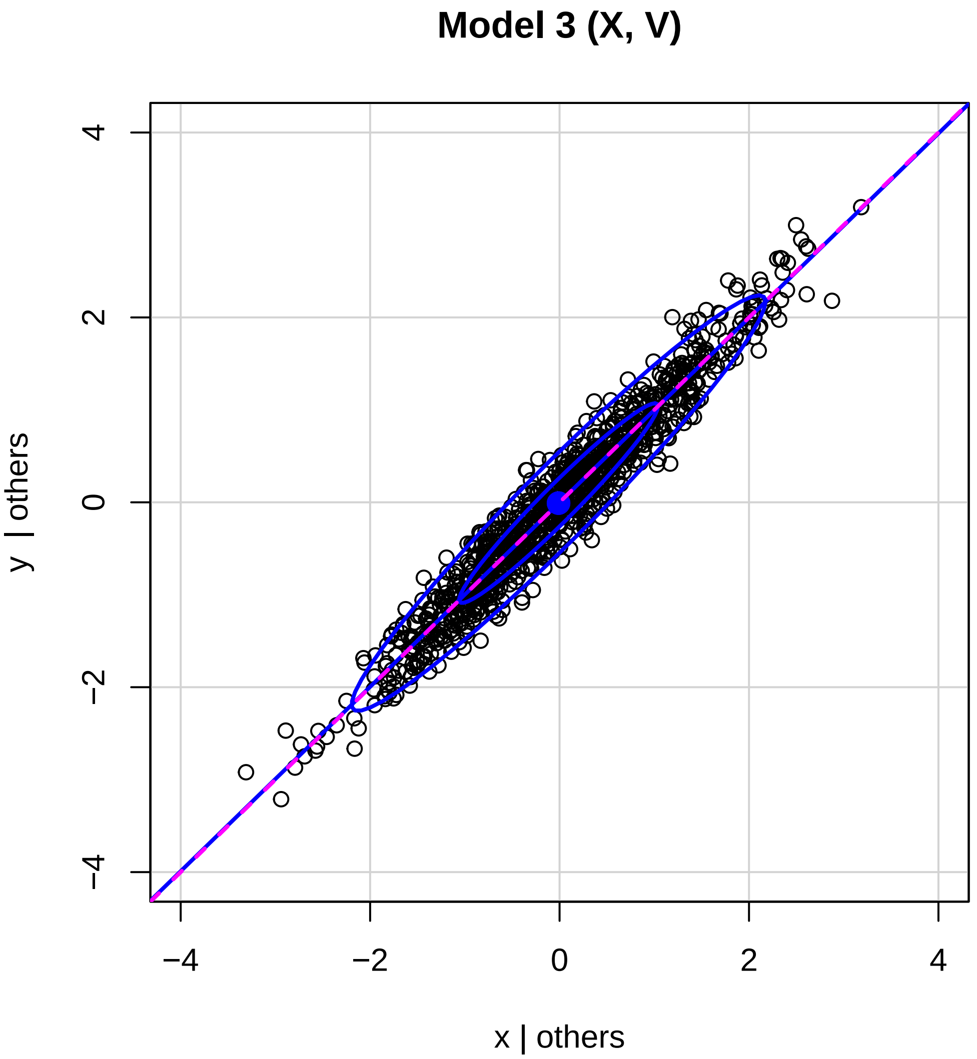
```
> avPlot(m1, "x", ellipse=TRUE, id=FALSE, main="Model 1 (X)",
+        xlim=c(-4, 4), ylim=c(-4, 4))
> abline(0, 1, col="magenta", lty=2, lwd=2)
```



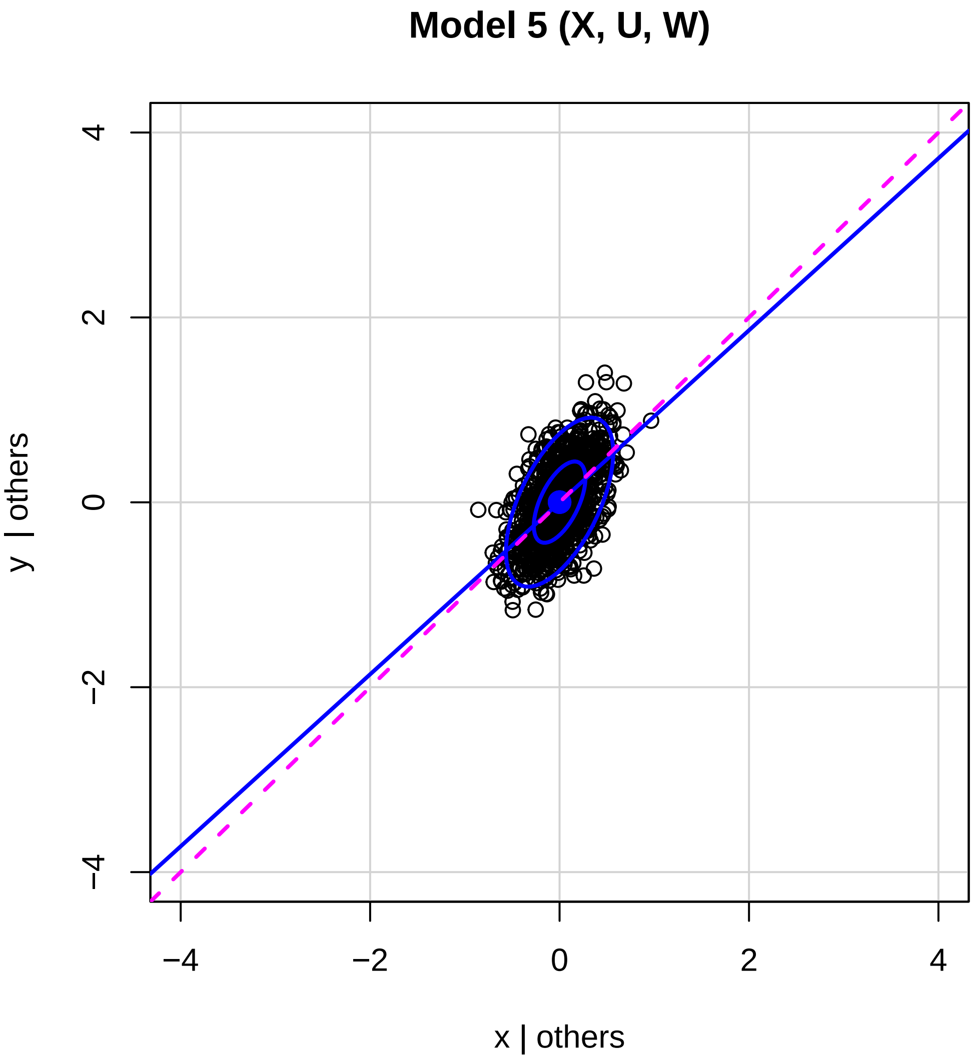
```
> avPlot(m2, "x", ellipse=TRUE, id=FALSE, main="Model 2 (X, U)",
+        xlim=c(-4, 4), ylim=c(-4, 4))
> abline(0, 1, col="magenta", lty=2, lwd=2)
```



```
> avPlot(m3, "x", ellipse=TRUE, id=FALSE, main="Model 3 (X, V)",
+        xlim=c(-4, 4), ylim=c(-4, 4))
> abline(0, 1, col="magenta", lty=2, lwd=2)
```



```
> avPlot(m5, "x", ellipse=TRUE, id=FALSE, main="Model 5 (X, U, W)",
+        xlim=c(-4, 4), ylim=c(-4, 4))
> abline(0, 1, col="magenta", lty=2, lwd=2)
```



I added the population regression line with slope $\beta_X = 1$ (the broken magenta line) to each plot, along with data ellipses scaled to enclose 50 and 95% of the data for bivariately normally distributed variables. I also used the same axis scales for the four graphs, to facilitate comparisons among them.

The following points are noteworthy:

- The AV plot for X in model 1 is just the scatterplot for Y versus X, centering the two variables to 0 means.
- The bias of B_1 in model 2, which controls for the collider U, is apparent. The slope estimates B_X for the other models are very close to $\beta_X = 1$.
- The increased precision of B_X in model 3, relative to model 1 as a consequence of including the covariate V is apparent in the tighter spread of points around the regression line in the AV plot, representing smaller residual variation.
- In contrast, the decreased precision of B_X in model 5 relative to model 1 as a consequence of including U and W in the model is apparent in the decreased conditional variation in X (i.e., horizontal spread) in the AV plot.

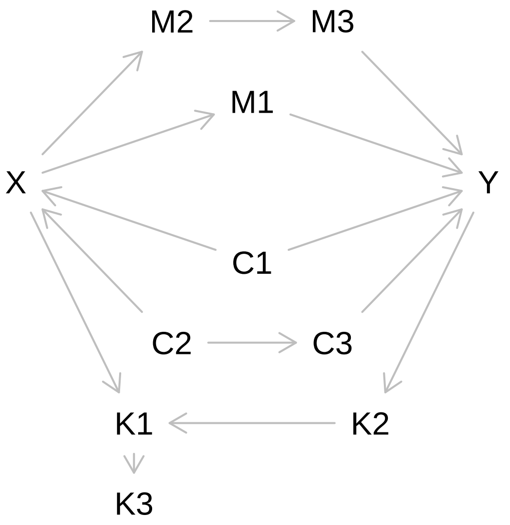
Exercise 26.5

(a) It's apparent from the DAG that C_1 and C_3 are also d-separated and hence unconditionally independent.

(b) Yes, there are $\{C_1, C_2, M_1, M_3\}$, $\{C_1, C_3, M_1, M_2\}$, and $\{C_1, C_3, M_1, M_3\}$, and thus $(Y \perp\!\!\!\perp X)|(C_1, C_2, M_1, M_3)$, $(Y \perp\!\!\!\perp X)|(C_1, C_3, M_1, M_2)$, and $(Y \perp\!\!\!\perp X)|(C_1, C_3, M_1, M_3)$.

(c) Here's **dagitty**'s analysis, showing many, but not all conditional independencies (and unconditional independencies)[2]:

```
> library("dagitty")
> g <- dagitty('dag{
+   X -> M2 -> M3 -> Y
+   X -> M1 -> Y
+   X <- C1 -> Y
+   X <- C2 -> C3 -> Y
+   X -> K1 <- K2 <- Y
+   K1 -> K3
+   X [exposure,pos="0, 0"]
+   Y [outcome,pos="1, 0"]
+   M1 [pos="0.5, -0.25"]
+   M2 [pos="0.33, -0.5"]
+   M3 [pos="0.67, -0.5"]
+   C1 [pos="0.5, 0.25"]
+   C2 [pos="0.33, 0.5"]
+   C3 [pos="0.67, 0.5"]
+   K1 [pos="0.25, 0.75"]
+   K2 [pos="0.75, 0.75"]
+   K3 [pos="0.25, 1"]
+ }')
>
> plot(g)
```



[2] Asking **dagitty** to report all pair-wise independencies reported 6802 of them and took abot half a minute on a reasonably fast laptop.

```
> adjustmentSets(g, type="all")
{ C1, C2 }
{ C1, C3 }
{ C1, C2, C3 }
> impliedConditionalIndependencies(g, type="missing.edge")
C1 _||_ C2
C1 _||_ C3
C1 _||_ K1 | K2, X
C1 _||_ K1 | X, Y
C1 _||_ K2 | Y
C1 _||_ K3 | K1
C1 _||_ K3 | K2, X
C1 _||_ K3 | X, Y
C1 _||_ M1 | X
C1 _||_ M2 | X
C1 _||_ M3 | M2
C1 _||_ M3 | X
C2 _||_ K1 | K2, X
C2 _||_ K1 | X, Y
C2 _||_ K1 | C1, C3, X
C2 _||_ K2 | Y
C2 _||_ K2 | C1, C3, M1, M3
C2 _||_ K2 | C1, C3, M1, M2
C2 _||_ K2 | C1, C3, X
C2 _||_ K3 | K1
C2 _||_ K3 | K2, X
C2 _||_ K3 | X, Y
C2 _||_ K3 | C1, C3, X
C2 _||_ M1 | X
C2 _||_ M2 | X
C2 _||_ M3 | M2
C2 _||_ M3 | X
C2 _||_ Y | C1, C3, M1, M3
C2 _||_ Y | C1, C3, M1, M2
C2 _||_ Y | C1, C3, X
C3 _||_ K1 | K2, X
C3 _||_ K1 | X, Y
C3 _||_ K1 | C1, C2, K2, M1, M2
C3 _||_ K1 | C1, C2, M1, M2, Y
C3 _||_ K1 | C1, C2, K2, M1, M3
C3 _||_ K1 | C1, C2, M1, M3, Y
C3 _||_ K2 | Y
C3 _||_ K3 | K1
C3 _||_ K3 | K2, X
C3 _||_ K3 | X, Y
C3 _||_ K3 | C1, C2, K2, M1, M2
C3 _||_ K3 | C1, C2, M1, M2, Y
C3 _||_ K3 | C1, C2, K2, M1, M3
C3 _||_ K3 | C1, C2, M1, M3, Y
C3 _||_ M1 | X
C3 _||_ M1 | C2
C3 _||_ M2 | X
C3 _||_ M2 | C2
```

```
C3 _||_ M3 | M2
C3 _||_ M3 | X
C3 _||_ M3 | C2
C3 _||_ X | C2
K1 _||_ M1 | X, Y
K1 _||_ M1 | K2, X
K1 _||_ M2 | M3, X
K1 _||_ M2 | X, Y
K1 _||_ M2 | K2, X
K1 _||_ M3 | C1, C3, M1, M2, Y
K1 _||_ M3 | C1, C2, M1, M2, Y
K1 _||_ M3 | X, Y
K1 _||_ M3 | C1, C3, K2, M1, M2
K1 _||_ M3 | C1, C2, K2, M1, M2
K1 _||_ M3 | K2, X
K1 _||_ Y | C1, C3, K2, M1, M3
K1 _||_ Y | C1, C2, K2, M1, M3
K1 _||_ Y | C1, C3, K2, M1, M2
K1 _||_ Y | C1, C2, K2, M1, M2
K1 _||_ Y | K2, X
K2 _||_ K3 | K1
K2 _||_ M1 | Y
K2 _||_ M2 | M3, X
K2 _||_ M2 | C1, C2, M1, M3
K2 _||_ M2 | C1, C3, M1, M3
K2 _||_ M2 | Y
K2 _||_ M3 | Y
K2 _||_ X | C1, C2, M1, M2
K2 _||_ X | C1, C2, M1, M3
K2 _||_ X | C1, C3, M1, M2
K2 _||_ X | C1, C3, M1, M3
K2 _||_ X | Y
K3 _||_ M1 | X, Y
K3 _||_ M1 | K2, X
K3 _||_ M1 | K1
K3 _||_ M2 | M3, X
K3 _||_ M2 | X, Y
K3 _||_ M2 | K2, X
K3 _||_ M2 | K1
K3 _||_ M3 | C1, C3, M1, M2, Y
K3 _||_ M3 | C1, C2, M1, M2, Y
K3 _||_ M3 | X, Y
K3 _||_ M3 | C1, C3, K2, M1, M2
K3 _||_ M3 | C1, C2, K2, M1, M2
K3 _||_ M3 | K2, X
K3 _||_ M3 | K1
K3 _||_ X | K1
K3 _||_ Y | C1, C3, K2, M1, M3
K3 _||_ Y | C1, C2, K2, M1, M3
K3 _||_ Y | C1, C3, K2, M1, M2
K3 _||_ Y | C1, C2, K2, M1, M2
K3 _||_ Y | K2, X
K3 _||_ Y | K1
```

```
M1 _||_ M2 | X
M1 _||_ M3 | M2
M1 _||_ M3 | X
M2 _||_ Y | C1, C3, M1, M3
M2 _||_ Y | C1, C2, M1, M3
M2 _||_ Y | M3, X
M3 _||_ X | M2
X _||_ Y | C1, C3, M1, M3
X _||_ Y | C1, C2, M1, M3
X _||_ Y | C1, C3, M1, M2
X _||_ Y | C1, C2, M1, M2
>     # note: try setting type="all.pairs", but be prepared to wait awhile
```

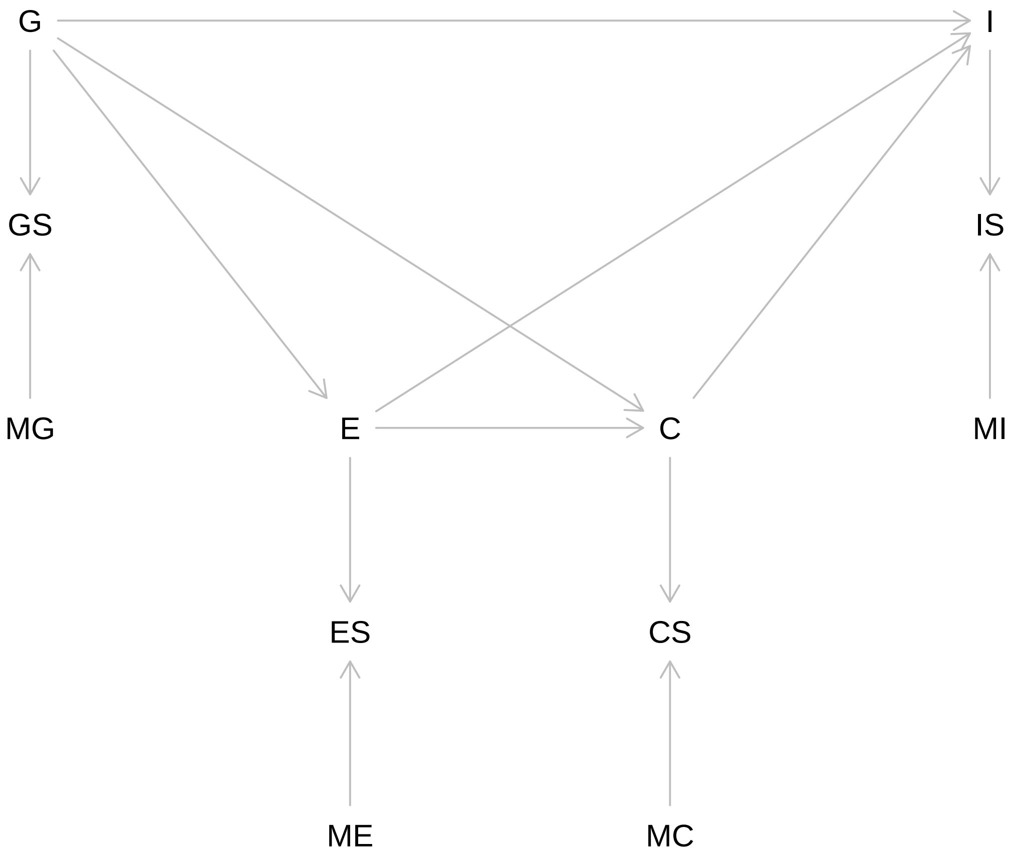
Exercise 26.7

(a) No, I don't find the proposed m-graph plausible because it seems unlikely to me that missingness in the several observed variables is independent—that is, I expect there to be dependencies among M_G, M_E, M_C and M_I. See part (b) for an m-graph that treats the missingness indicators as dependent.

Let's assume the reasonableness of the m-graph. The relevant sets of variables are $\mathbf{V}_c$ (i.e., the set of completely observed variables), which is empty; $\mathbf{V}_m = \{G, E, C, I\}$ (the completely observed versions of the variables with missing data); $\mathbf{V}^* = \{G^*, E^*, C^*, I^*\}$ (the observed variables with missing data); and $\mathbf{M} = \{M_G, M_E, M_C, M_F\}$ (the missingness indicators). There are no latent variables. From the DAG, it's apparent that each of the variables in $\mathbf{V}_m$ is independent of all of the variables in $\mathbf{M}$ (the starred variables in $\mathbf{V}^*$ are effectively colliders blocking paths between the variables in $\mathbf{V}_m$ and in $\mathbf{M}$), and so the missing data are MCAR. We could do a complete-case analysis or use multiple imputation to estimate the regression.

Here's **daggity**'s analysis of pairwise independencies (though avoiding the time-consuming computation of *all* pairwise independencies):

```
> library("dagitty")
> g <- dagitty('dag{
+   G -> E
+   G -> C
+   G -> I
+   E -> C
+   E -> I
+   C -> I
+   G -> GS <- MG
+   E -> ES <- ME
+   C -> CS <- MC
+   I -> IS <- MI
+   G [pos="c(0, 0)"]
+   E [pos="c(0.2, 0.1)"]
+   C [pos="c(0.4, 0.1)"]
+   I [pos="c(0.6, 0.0)"]
+   GS [pos="c(0, 0.05)"]
+   ES [pos="c(0.2,  0.15)"]
+   CS [pos="c(0.4, 0.15)"]
+   IS [pos="c(0.6, 0.05)"]
+   MG [pos="c(0, 0.1)"]
+   ME [pos="c(0.2,  0.2"]
+   MC [pos="c(0.4, 0.2)"]
+   MI [pos="c(0.6, 0.1)"]
+ }')
> plot(g)
```



```
> impliedConditionalIndependencies(g, type="missing.edge")
C _||_ ES | E
C _||_ GS | G
C _||_ IS | I
C _||_ MC
C _||_ ME
C _||_ MG
C _||_ MI
CS _||_ E | C
CS _||_ ES | E
CS _||_ ES | C
CS _||_ G | C
CS _||_ GS | G
CS _||_ GS | C
CS _||_ I | C
CS _||_ IS | I
CS _||_ IS | C
CS _||_ ME
CS _||_ MG
CS _||_ MI
E _||_ GS | G
```

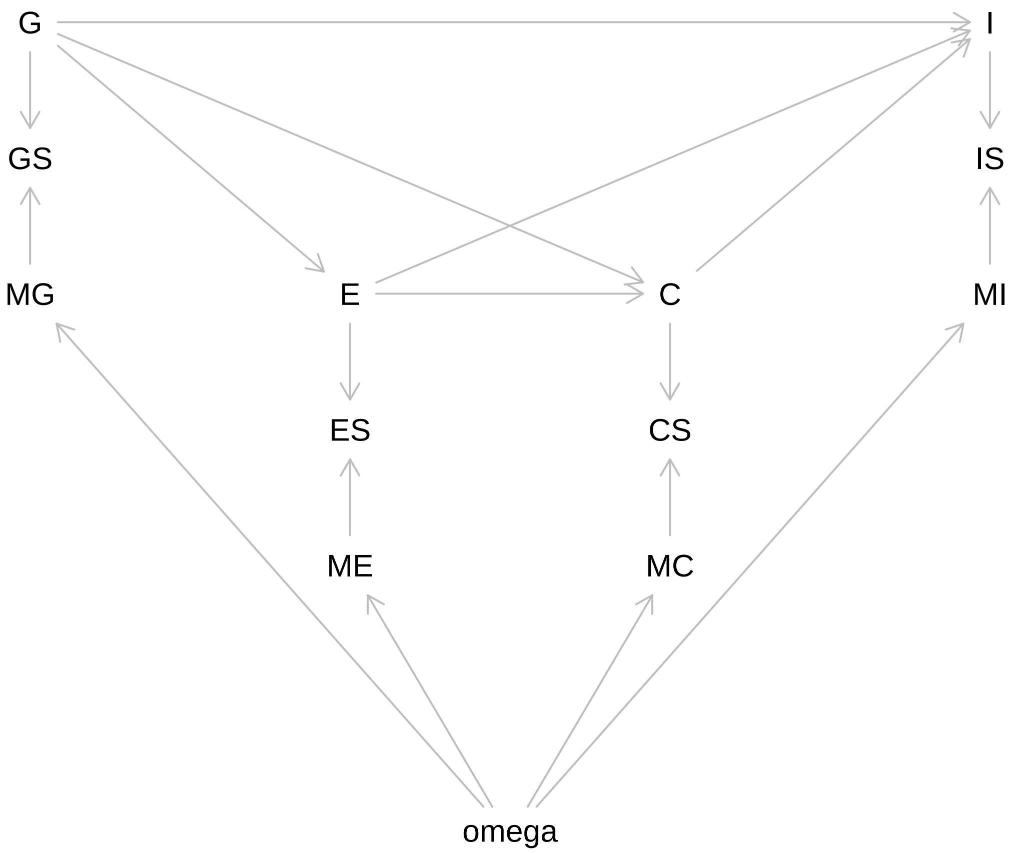
```
E _||_ IS | I
E _||_ MC
E _||_ ME
E _||_ MG
E _||_ MI
ES _||_ G | E
ES _||_ GS | G
ES _||_ GS | E
ES _||_ I | E
ES _||_ IS | I
ES _||_ IS | E
ES _||_ MC
ES _||_ MG
ES _||_ MI
G _||_ IS | I
G _||_ MC
G _||_ ME
G _||_ MG
G _||_ MI
GS _||_ I | G
GS _||_ IS | I
GS _||_ IS | G
GS _||_ MC
GS _||_ ME
GS _||_ MI
I _||_ MC
I _||_ ME
I _||_ MG
I _||_ MI
IS _||_ MC
IS _||_ ME
IS _||_ MG
MC _||_ ME
MC _||_ MG
MC _||_ MI
ME _||_ MG
ME _||_ MI
MG _||_ MI
```

(b) I believe that this structure is more plausible, with a common latent variable particular to country affecting all of the missingness indicators. The problem that this induces, however, is that missing data are now neither MCAR nor MAR.

The sets of variables $\mathbf{V}_c$ (still empty), $\mathbf{V}_m$, $\mathbf{V}^*$, and $\mathbf{M}$ are the same as before, but now the set of latent variables isn't empty, $\boldsymbol{\Omega} = \{\omega\}$. The latent variable ω is obviously not independent of $\mathbf{M}$ (since it directly affects each variable in $\mathbf{M}$), ruling out MCAR. Nor is ω independent of $\mathbf{M}$ given $\mathbf{V}_c$, since the latter is empty. Given the tools of the chapter, that leaves us with no way to estimate the model.

Here's **dagitty**'s analysis:

```
> g <- dagitty('dag{
+   G -> E
+   G -> C
+   G -> I
+   E -> C
+   E -> I
+   C -> I
+   G -> GS <- MG <- omega
+   E -> ES <- ME <- omega
+   C -> CS <- MC <- omega
+   I -> IS <- MI <- omega
+   G [pos="c(0, 0)"]
+   E [pos="c(0.2, 0.1)"]
+   C [pos="c(0.4, 0.1)"]
+   I [pos="c(0.6, 0.0)"]
+   GS [pos="c(0, 0.05)"]
+   ES [pos="c(0.2,  0.15)"]
+   CS [pos="c(0.4, 0.15)"]
+   IS [pos="c(0.6, 0.05)"]
+   MG [pos="c(0, 0.1)"]
+   ME [pos="c(0.2,  0.2"]
+   MC [pos="c(0.4, 0.2)"]
+   MI [pos="c(0.6, 0.1)"]
+   omega [pos="c(0.3, 0.3)"]
+ }')
> plot(g)
```



```
> impliedConditionalIndependencies(g, type="missing.edge")
C _||_ ES | E
C _||_ GS | G
C _||_ IS | I
C _||_ MC
C _||_ ME
C _||_ MG
C _||_ MI
C _||_ omeg
CS _||_ E | C
CS _||_ ES | E, ME
CS _||_ ES | E, omeg
CS _||_ ES | E, MC
CS _||_ ES | C, ME
CS _||_ ES | C, omeg
CS _||_ ES | C, MC
CS _||_ G | C
CS _||_ GS | G, MG
CS _||_ GS | G, omeg
CS _||_ GS | G, MC
CS _||_ GS | C, MG
CS _||_ GS | C, omeg
CS _||_ GS | C, MC
CS _||_ I | C
CS _||_ IS | I, MI
CS _||_ IS | I, omeg
CS _||_ IS | I, MC
CS _||_ IS | C, MI
CS _||_ IS | C, omeg
CS _||_ IS | C, MC
CS _||_ ME | omeg
CS _||_ ME | MC
CS _||_ MG | omeg
CS _||_ MG | MC
CS _||_ MI | omeg
CS _||_ MI | MC
CS _||_ omeg | MC
E _||_ GS | G
E _||_ IS | I
E _||_ MC
E _||_ ME
E _||_ MG
E _||_ MI
E _||_ omeg
ES _||_ G | E
ES _||_ GS | G, MG
ES _||_ GS | G, omeg
ES _||_ GS | G, ME
ES _||_ GS | E, MG
ES _||_ GS | E, omeg
ES _||_ GS | E, ME
ES _||_ I | E
ES _||_ IS | I, MI
ES _||_ IS | I, omeg
ES _||_ IS | I, ME
ES _||_ IS | E, MI
ES _||_ IS | E, omeg
ES _||_ IS | E, ME
ES _||_ MC | omeg
ES _||_ MC | ME
ES _||_ MG | omeg
ES _||_ MG | ME
ES _||_ MI | omeg
ES _||_ MI | ME
ES _||_ omeg | ME
G _||_ IS | I
G _||_ MC
G _||_ ME
G _||_ MG
G _||_ MI
G _||_ omeg
GS _||_ I | G
GS _||_ IS | I, MI
GS _||_ IS | I, omeg
GS _||_ IS | I, MG
GS _||_ IS | G, MI
GS _||_ IS | G, omeg
GS _||_ IS | G, MG
GS _||_ MC | omeg
GS _||_ MC | MG
GS _||_ ME | omeg
GS _||_ ME | MG
GS _||_ MI | omeg
GS _||_ MI | MG
GS _||_ omeg | MG
I _||_ MC
I _||_ ME
I _||_ MG
I _||_ MI
I _||_ omeg
IS _||_ MC | omeg
IS _||_ MC | MI
IS _||_ ME | omeg
IS _||_ ME | MI
IS _||_ MG | omeg
IS _||_ MG | MI
IS _||_ omeg | MI
MC _||_ ME | omeg
MC _||_ MG | omeg
MC _||_ MI | omeg
ME _||_ MG | omeg
ME _||_ MI | omeg
MG _||_ MI | omeg
```

References

J. Fox and S. Wesberg. *An R Companion to Applied Regression.* Sage, Thousand Oaks CA, third edition, 2019.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, 2022. URL https://www.R-project.org/.

J. Textor, B. van der Zander, M. S. Gilthorpe, M. Liśkiewicz, and G. T. Ellison. Robust causal inference using directed acyclic graphs: the R package 'dagitty'. *International Journal of Epidemiology*, 45(6):1887–1894, 2017.