P8124 Assignment 3

Christine Lucille Kuryla (clk2162)

2023-10-23

Contents

Problem 1

$X_i \perp\!\!\!\perp X_j | X_S$ if and only if $[\Sigma^{-1}_{\{ijS\}\{ijS\}}]_{i,j}$

Let $X \sim N(\mu, \Sigma)$ where $X = (X_1, ..., X_p)$. Let $S = X_S \subset X \setminus X_i, X_j$ for some arbitrary subset of variables. For ease of notation, let $A := \{X_i, X_j\}$. Our goal is to explain and understand why $X_i \perp\!\!\!\perp X_j | X_S$ iff $[\Sigma^{-1}_{\{AS\}\{AS\}}]_{i,j}$. Note that $X_{A|S} \sim N(\mu_{A|S}, \Sigma_{A|S})$.

To begin, we suppose that $X_i \perp\!\!\!\perp X_j | X_S$. This conditional independence means that the conditional covariances of X_i and X_j given S are zero, so $cov(X_i, X_j|S) = cov(X_j, X_i|S) = 0$. This means that the off-diagonal elements of the conditional covariance matrix $\Sigma_{A|S}$ are zero.

$$\Sigma_{A|S} = \begin{bmatrix} var(X_i|S) & cov(X_i, X_j|S) \\ cov(X_j, X_i|S) & var(X_j|S) \end{bmatrix} = \begin{bmatrix} c1 & 0 \\ 0 & c2 \end{bmatrix}$$

where c1 and c2 are constants. The inverse of $\Sigma_{A|S}$ thus also has zeros in its off-diagonal elements, hence $\Sigma^{-1}_{A|S}$ has zeros in its off-diagonal elements. By Schur, we know that $\Sigma^{-1}_{A|S} = K_{AA}$, so we know the precision matrix K_{AA} also has zeroes in it's off-diagonal elements. Let us represent this as

$$\Sigma_{A|S}^{-1} = K_{AA} = \begin{bmatrix} c3 & 0 \\ 0 & c4 \end{bmatrix}$$

where c3 and c4 are constants.

Recall, our goal was to show that the i,j-th elements of the inverse of the covariance matrix $[\Sigma_{\{AS\}\{AS\}}^{-1}]_{i,j}$ must be zero. By definition, $K_{AA} := \Sigma_{AA}^{-1}$, so we have $\Sigma_{\{AS\}\{AS\}}^{-1} = K_{\{AS\}\{AS\}}$. Let us rewrite this in terms of precision matrices $K_{AA}, K_{AS}, K_{SA}, K_{SS}$ as follows:

$$\Sigma_{\{AS\}\{AS\}}^{-1} = K_{\{AS\}\{AS\}} = \begin{bmatrix} K_{AA} & K_{AS} \\ K_{SA} & K_{SS} \end{bmatrix}$$

Rewriting this with K_{AA} as above, we have:

$$\Sigma_{\{AS\}\{AS\}}^{-1} = K_{\{AS\}\{AS\}} = \begin{bmatrix} K_{AA} & K_{AS} \\ K_{SA} & K_{SS} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} c1 & 0 \\ 0 & c2 \end{bmatrix} & K_{A,S} \\ K_{S,A} & K_{S,S} \end{bmatrix}$$

Recall that $A := \{X_i, X_j\}$. This means that the i-th, j-th elements of this matrix must be zeroes, hence $[\Sigma_{\{AS\}\{AS\}}^{-1}]_{i,j} = 0$, as desired.

One may also begin by assuming that $[\Sigma_{\{AS\}\{AS\}}^{-1}]_{i,j} = 0$, and following similar logic and relationships, show that $X_i \perp\!\!\!\perp X_j | X_S$.

Problem 2

Simulate data from given MRF independence model

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
# simulate data from a given MRF independence model

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
```

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

# (Note: in R, the inverse of a matrix M is obtained by solve(M).)
```

Conditional Independencies

What are the conditional independencies that are representing in this precision matrix? Conditional independencies correspond to the zeros in the precision matrix of the elements given everything else. Hence, for K, the conditional independencies are:

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

and

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

Corresponding Graph

What is the corresponding graph? The corresponding MRF has vertices X_1, X_2, X_3, X_4 and edges:

- $X_1 - X_2$.
- $X_2 - X_4$.
- $X_4 - X_3$.
- $X_3 - X_1$.

Verify with linear regression

Verify the conditional independence constraints by using linear regression.

```
# X1 independent of X4 given X2, X3
summary(glm(data = data, formula = X1 ~ X4 + X2 + X3))
```

```
##
## Call:
## glm(formula = X1 ~ X4 + X2 + X3, data = data)
##
## 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
##
## (Dispersion parameter for gaussian family taken to be 0.09863182)
##
##     Null deviance: 1813.81  on 9999  degrees of freedom
## Residual deviance:  985.92  on 9996  degrees of freedom
## AIC: 5221.2
##
## Number of Fisher Scoring iterations: 2
```

```
# X2 independent of X3 given X1, X4
summary(glm(data = data, formula = X2 ~ X3 + X1 + X4))
```

```
##
## Call:
## glm(formula = X2 ~ X3 + X1 + X4, data = data)
##
## 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
##
## (Dispersion parameter for gaussian family taken to be 0.05046277)
##
##     Null deviance: 818.95  on 9999  degrees of freedom
## Residual deviance: 504.43  on 9996  degrees of freedom
## AIC: -1480.4
##
## Number of Fisher Scoring iterations: 2
```

As demonstrated in the first linear regression, $X_1 \perp X_4 | X_2, X_3$ because we can see that when regressing X_1 on X_4, X_2, X_3 gives a large p-value for X_4 because they are conditionally independent since X_2 and X_3 are given (note that their small p values demonstrate that they are dependent). The same logic applies to the second regression for $X_2 \perp X_3 | X_1, X_4$ by regressing X_2 on the rest of the variables and observing a large p-value for X_3, showing independence, because X_1 and X_4 are conditioned on by putting them in the regression.

Explanation

The zeroes in the precision matrix K correspond to the conditional independencies described above. The MRF is the UG with X_1, X_2, X_3, X_4 that has the edge between X_1 and X_4 removed because of the conditional independence $X_1 \perp\!\!\!\perp X_4 | X_2, X_3$ and the edge between X_2 and X_3 removed because of the conditional independence $X_2 \perp\!\!\!\perp X_3 | X_1, X_4$ that were demonstrated by the zeroes in the precision matrix. The linear regression demonstrates that the conditional independencies are true because when one variable is regressed on the rest, the p-value for the variable that it is conditionally independent of is large (because they are independent), and the p-values of the variables in the conditioning set are small (because they are dependent).

Estimate precision matrix subject to graph constraints

```r
# Use the gRim package to fit the model, i.e., estimate the precision matrix subject to the graph const

library(gRim)
```

```
## Loading required package: gRbase
```

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

```
##           X1        X2        X3        X4
## X1 10.182411  6.988142  7.140856  0.000000
## X2  6.988142 19.832337  0.000000  7.076402
## X3  7.140856  0.000000 29.394792  6.852069
## X4  0.000000  7.076402  6.852069 40.745105
```

```r
# Did it work? How do you know?

# Precision matrix (K)
kable(K)
```

10	7	7	0
7	20	0	7
7	0	30	7
0	7	7	40

```r
# Estimated precision matrix
kable(fit$K)
```

	X1	X2	X3	X4
X1	10.182411	6.988142	7.140856	0.000000
X2	6.988142	19.832337	0.000000	7.076402
X3	7.140856	0.000000	29.394792	6.852069
X4	0.000000	7.076402	6.852069	40.745105

Yes, it worked. We know this because the estimated precision matrix has the expected zeroes that correspond to the conditional independencies, and in general, the values are quite close to K so a good estimation of the actual precision matrix.

Problem 3

Consider the Gaussian Bayesian Network model with the following covariance matrix: and the DAG G with edges X1 → X2 ← X3 and X4 → X2.

a) Correlation constraints and matrix

- a) What correlation constraints does this model represent? Estimate the correlation matrix. * This model represents three marginal independencies (six correlations shown by the 0s).
- $X_4 \perp X_3$ (X_4 is marginally independent of X_3, so the correlation between X_4 and $X_3 = 0$). Correlations are symmetric, so $corr(X_3, X_4) = corr(X_4, X_3) = 0$.
- $X_1 \perp X_3$ (X_1 is marginally independent of X_3, so the correlation between X_1 and $X_3 = 0$). Correlations are symmetric, so $corr(X_3, X_1) = corr(X_1, X_3) = 0$.
- $X_1 \perp X_4$ (X_1 is marginally independent of X_4, so the correlation between X_1 and $X_4 = 0$). Correlations are symmetric, so $corr(X_4, X_1) = corr(X_1, X_4) = 0$.

```r
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
```

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

# Estimate correlation matrix
sigma_est <- cor(data)

kable(sigma_est)
```

	X1	X2	X3	X4
X1	1.0000000	-0.4661188	0.0121879	-0.0115046
X2	-0.4661188	1.0000000	0.4630349	0.4733142
X3	0.0121879	0.4630349	1.0000000	0.0063924
X4	-0.0115046	0.4733142	0.0063924	1.0000000

The estimated correlation matrix above shows very small values close to zero corresponding to the elements of the actual matrix that are zero, namely, that the correlation between X_3 and X_1 is 0.0121879, between X_4 and X_1 is -0.0115046, and between X_3 and X_4 is 0.0063924.

b) The moralized graph

- b) Consider also the moralized graph Gm and what the corresponding precision matrix K would look like. What are the partial correlation constraints represented in K? How does this make sense with respect to sigma above? *
- The moralized Graph Gm would be the complete graph formed from the skeleton of G. It is the graph formed by making the edges in G undirected and adding edges $X_1 - X_4$, $X_4 - X_3$, and $X_3 - X_1$ because X_2 is an unshielded collider so it's parents are married during the moralization process.
- There are no partial correlation constraints represented in K because there are no missing edges in Gm or conditional independencies.
- This makes sense wrt the correlation matrix Sigma above because there are marginal independencies but no conditional independencies.

c) Estimate K, take inverse, and compare to true Sigma

- c) Following steps similar to the previous problem, estimate the corresponding precision matrix K from this data (using ggmfit). Take the inverse and compare to the true covariance matrix. *

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

```
##           X1         X2         X3         X4
## X1 0.4270365  0.2001141  0.0000000  0.0000000
## X2 0.2001141  0.6213527 -0.1989045 -0.2021127
## X3 0.0000000 -0.1989045  0.4290929  0.0000000
## X4 0.0000000 -0.2021127  0.0000000  0.4188167
```

```r
solve(fit$K) # inverse of K (covariance matrix)
```

```
##            X1        X2         X3         X4
## X1  2.9917221 -1.387081 -0.6429763 -0.6693778
## X2 -1.3870808  2.959982  1.3720889  1.4284287
## X3 -0.6429763  1.372089  2.9665248  0.6621430
## X4 -0.6693778  1.428429  0.6621430  3.0770111
```

```r
# True covariance matrix
Sig
```

```
##      [,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
```

The estimates for the non-zero entries in the covariance matrix are similar numbers to the true covariance matrix (close to -1.4, 1.4, -3, 3, but not exact because it's an estimation from simulated data). However, there are very non-zero values in place of the true zeroes, because during the moralization process, three edges were added because of the unshielded colliders. This is a demonstration of why going from a moralized, undirected graph Gm to a DAG is not a reliable way to completely determine the structure of the original DAG G.

Problem 4

Simulate graph

```r
library(dagitty)
```

```
##
## Attaching package: 'dagitty'

## The following object is masked from 'package:gRim':
##
##     ciTest

## The following objects are masked from 'package:gRbase':
##
##     ancestors, children, edges, moralize, parents
```

```r
#Use dagitty to simulate 10000 observations from this graph:
g <- dagitty( "dag{ x <- u1; u1 -> m <- u2 ; u2 -> y }" )

sim_sem <- simulateSEM(g,
            b.lower = 0.4,
            b.upper = 0.7,
            N = 10000)

# Here U1,U2 represent unmeasured variables.
```

Estimate effects of X on Y

Estimate the effect of X on Y adjusting for M in a linear regression, obtaining a 95% confidence interval for the effect.

```r
# Estimate the effect of X on Y adjusting for M in a linear regression, obtaining a 95% confidence inte

lm_m = lm(data = sim_sem, formula = y ~ x + m )
summary(lm_m)
```

```
##
## Call:
## lm(formula = y ~ x + m, data = sim_sem)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7658 -0.6043  0.0072  0.6044  3.4050
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.008944   0.008972  -0.997    0.319
## x           -0.183864   0.009566 -19.221   <2e-16 ***
## m            0.494160   0.009698  50.956   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8972 on 9997 degrees of freedom
## Multiple R-squared:  0.2062, Adjusted R-squared:  0.2061
## F-statistic:  1299 on 2 and 9997 DF,  p-value: < 2.2e-16
```

```
library(broom)
tidy_ci_m <- tidy(lm_m, conf.int=TRUE)
tidy_ci_m
```

```
## # A tibble: 3 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept) -0.00894   0.00897    -0.997 3.19e- 1  -0.0265   0.00864
## 2 x           -0.184     0.00957   -19.2   7.06e-81  -0.203   -0.165
## 3 m            0.494     0.00970    51.0   0           0.475    0.513
```

The estimated effect of X on Y, adjusting for M, is -0.1838637, with a 95% confidence interval of (-0.2026147 , -0.1651126).

Then estimate the same effect (and confidence interval) using the correct sufficient adjustment set that you can obtain from dagitty.

```
# Then estimate the same effect (and confidence interval) using the correct sufficient adjustment set t

# Sufficient adjustment set
adjustmentSets(g, "y", "x", type = "minimal")
```

```
##  {}
```

```
# This results in an empty set.

lm_sufficient = lm(data = sim_sem, formula = y ~ x )
summary(lm_sufficient)
```

```
##
## Call:
## lm(formula = y ~ x, data = sim_sem)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7294 -0.6828  0.0052  0.7014  3.6712
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.013927   0.010069  -1.383    0.167
## x           -0.006228   0.009998  -0.623    0.533
##
## Residual standard error: 1.007 on 9998 degrees of freedom
## Multiple R-squared:  3.881e-05,  Adjusted R-squared:  -6.121e-05
## F-statistic: 0.388 on 1 and 9998 DF,  p-value: 0.5334
```

```
tidy_ci_sufficient <- tidy(lm_sufficient, conf.int=TRUE)
tidy_ci_sufficient
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept) -0.0139     0.0101    -1.38    0.167  -0.0337   0.00581
## 2 x           -0.00623    0.0100    -0.623   0.533  -0.0258   0.0134
```

The estimate of the effect of X on Y adjusting for M in a linear regression is -0.1838637, with a confidence interval (-0.2026147, -0.1651126). This does not cross zero and implies an association.

The estimate of the effect of X on Y adjusting for nothing (the sufficient set from daggity was the empty set) in a linear regression is -0.0062277, with a confidence interval (-0.0258251, 0.0133698). This crosses zero and does not imply an association.

From this example, we can conclude that adjusting for M induced an association that was not actually there. This is an example of the M bias, or the butterfly bias, where a d-connection is induced by conditioning on the wrong thing, so it results in a bias/association that isn't actually there. We can see this from the underlying data that we simulated from a known DAG.

Problem 5

Construct DAG in dagitty and simulate 10000 observations

```
# Construct the DAG in Figure 1 as a daggity object.

dag_q5 <- dagitty('dag {
    D [pos="0,1"]
    E [pos="1,1"]
    A [pos="2,1"]
    B [pos="3,1"]
    F [pos="4,1"]
    C [pos="1,0"]
    G [pos="4,0"]
    H [pos="5,0"]

    D -> E -> A <- B <- G -> H
    E -> F -> H
    E <- C -> H
    C -> B
    C -> F -> G

}')

plot(dag_q5)
```

$$\begin{tikzcd}
 & C \arrow[rrr] \arrow[d] \arrow[drr] \arrow[drrr] & & & G \arrow[r] \arrow[dl] & H \\
D \arrow[r] & E \arrow[r] & A & B \arrow[l] \arrow[r] & F \arrow[u] \arrow[ur] &
\end{tikzcd}$$

```r
# Simulate 10000 observations from this graph using simulateSEM() as you did on the first homework.

sim_sem <- simulateSEM(dag_q5,
            b.lower = -0.7,
            b.upper = 0.7,
            N = 10000)
```

Estimate the effect of E on F with linear regression and different adjustments

Note that the result from dagitty for the minimal adjustment set is just one: { C }. In order to estimate the effect of E on F, we will regress F on E and adjust for C, as well as regressing it on everything, and compare the results.

```r
# Estimate the effect of E on F and the effect of B on A using backdoor adjustment and linear regressio

# Effect of E on F

adjustmentSets(dag_q5, "E", "F", type = "minimal")
```

```
## { C }
```

```r
# Result: { C }

# Linear regression of F on E, adjusting for C
lm_ef_adj_c = lm(data = sim_sem, formula = F ~ E + C )
summary(lm_ef_adj_c)
```

```
##
## Call:
## lm(formula = F ~ E + C, data = sim_sem)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.73640 -0.49614  0.00195  0.49180  2.56520 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.002099   0.007361  -0.285    0.776    
## E            0.377905   0.007596  49.748   <2e-16 ***
## C            0.656373   0.007587  86.512   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.736 on 9997 degrees of freedom
## Multiple R-squared:  0.4535, Adjusted R-squared:  0.4534 
## F-statistic:  4149 on 2 and 9997 DF,  p-value: < 2.2e-16
```

```
tidy_ef_adj_c <- tidy(lm_ef_adj_c, conf.int=TRUE)
tidy_ef_adj_c
```

```
## # A tibble: 3 x 7
##   term        estimate std.error statistic p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept) -0.00210   0.00736    -0.285   0.776  -0.0165    0.0123
## 2 E            0.378     0.00760    49.7     0       0.363     0.393 
## 3 C            0.656     0.00759    86.5     0       0.642     0.671
```

```
# Linear regression of F on everything
lm_f = lm(data = sim_sem, formula = F ~ E + C + A + B + D + G + H)
summary(lm_f)
```

```
## 
## Call:
## lm(formula = F ~ E + C + A + B + D + G + H, data = sim_sem)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.64971 -0.45393  0.00391  0.45342  2.32159 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.0034266  0.0067545  -0.507   0.6120    
## E            0.3223556  0.0097042  33.218   <2e-16 ***
## C            0.5544605  0.0130265  42.564   <2e-16 ***
## A           -0.0002678  0.0077110  -0.035   0.9723    
## B            0.0086331  0.0193922   0.445   0.6562    
## D            0.0063511  0.0085031   0.747   0.4551    
## G           -0.3250400  0.0141664 -22.944   <2e-16 ***
## H           -0.0125339  0.0071464  -1.754   0.0795 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.6752 on 9992 degrees of freedom
## Multiple R-squared:  0.5404, Adjusted R-squared:  0.5401 
## F-statistic:  1678 on 7 and 9992 DF,  p-value: < 2.2e-16
```

```r
tidy_f <- tidy(lm_f, conf.int=TRUE)
tidy_f
```

```
## # A tibble: 8 x 7
##   term         estimate std.error statistic   p.value conf.low conf.high
##   <chr>           <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept) -0.00343    0.00675   -0.507  6.12e-  1  -0.0167   0.00981
## 2 E            0.322      0.00970   33.2    1.32e-229   0.303    0.341
## 3 C            0.554      0.0130    42.6    0           0.529    0.580
## 4 A           -0.000268   0.00771   -0.0347 9.72e-  1  -0.0154   0.0148
## 5 B            0.00863    0.0194     0.445  6.56e-  1  -0.0294   0.0466
## 6 D            0.00635    0.00850    0.747  4.55e-  1  -0.0103   0.0230
## 7 G           -0.325      0.0142   -22.9    1.40e-113  -0.353   -0.297
## 8 H           -0.0125     0.00715   -1.75   7.95e-  2  -0.0265   0.00147
```

```r
# Linear regression of F on E, with no adjustments
lm_ef_adj_0 = lm(data = sim_sem, formula = F ~ E)
summary(lm_ef_adj_0)
```

```
##
## Call:
## lm(formula = F ~ E, data = sim_sem)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5526 -0.6646 -0.0021  0.6601  3.6746
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.006836   0.009732   0.702    0.482
## E           0.209345   0.009709  21.562   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9732 on 9998 degrees of freedom
## Multiple R-squared:  0.04444,    Adjusted R-squared:  0.04434
## F-statistic: 464.9 on 1 and 9998 DF,  p-value: < 2.2e-16
```

```r
tidy_ef_adj_0 <- tidy(lm_ef_adj_0, conf.int=TRUE)
tidy_ef_adj_0
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)  0.00684   0.00973     0.702 4.82e-  1  -0.0122    0.0259
## 2 E            0.209     0.00971    21.6   7.83e-101   0.190     0.228
```

```r
# Table of results

ef_adj_c <- c (tidy_ef_adj_c$term[2],
             tidy_ef_adj_c$estimate[2],
             tidy_ef_adj_c$std.error[2],
             tidy_ef_adj_c$conf.low[2],
             tidy_ef_adj_c$conf.high[2],
             (tidy_ef_adj_c$conf.high[2] - tidy_ef_adj_c$conf.low[2])
                )
```

```r
ef_adj_all <- c (tidy_f$term[2],
             tidy_f$estimate[2],
             tidy_f$std.error[2],
             tidy_f$conf.low[2],
             tidy_f$conf.high[2],
             (tidy_f$conf.high[2] - tidy_f$conf.low[2])
                )

ef_adj_0 <- c (tidy_ef_adj_0$term[2],
             tidy_ef_adj_0$estimate[2],
             tidy_ef_adj_0$std.error[2],
             tidy_ef_adj_0$conf.low[2],
             tidy_ef_adj_0$conf.high[2],
             (tidy_ef_adj_0$conf.high[2] - tidy_ef_adj_0$conf.low[2])
                )

table_ef <- rbind(ef_adj_c, ef_adj_all, ef_adj_0)

colnames(table_ef) <- c("Var", "Estimate", "SE", "CI low", "CI high", "CI length")

kable(table_ef)
```

	Var	Estimate	SE	CI low	CI high	CI length
ef_adj_c	E	0.377905443851971	0.00759644322956106	0.363014885873711	0.392796001830231	0.02978111595652
ef_adj_all	E	0.322355599401436	0.00970417668831716	0.303333458381291	0.341377740421582	0.038044282040291
ef_adj_0	E	0.20934502526251	0.00970875778721680	0.190313905749372	0.228376144775649	0.0380622390262762

The estimate of the effect of E on F adjusting for C in a linear regression is 0.3779054, with a confidence interval (0.3630149, 0.392796). The standard error is 0.0075964. The CI length is 0.0297811.

The estimate of the effect of E on F adjusting for all of the variables in a linear regression is 0.3223556, with a confidence interval (0.3033335, 0.3413777). The standard error is 0.0097042. The CI length is 0.0380443.

The estimate of the effect of E on F adjusting for none of the variables in a linear regression is 0.3223556, with a confidence interval (0.3033335, 0.3413777). The standard error is 0.0097042. The CI length is 0.0380443.

Estimate the effect of B on A with linear regression and different adjustments

Note that the result from dagitty for the minimal adjustment sets are { E }, { C, F }, { C, G }. In order to estimate the effect of B on A, we will regress A on B and adjust for the sufficient sets, as well as regressing it on everything, and compare the results.

```r
# Effect of B on A
adjustmentSets(dag_q5, "B", "A", type = "minimal")
```

```
## { E }
## { C, F }
## { C, G }
```

```r
# Result: { E }, { C, F }, { C, G }

# If there is more than one sufficient adjustment set, try each of the ones identified by dagitty and c
# Linear regression of A on B, adjusting for E
lm_ab_adj_e = lm(data = sim_sem, formula = A ~ B + E )
summary(lm_ab_adj_e)
```

```
##
## Call:
## lm(formula = A ~ B + E, data = sim_sem)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3303 -0.5920 -0.0005  0.5861  3.1899
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.010431   0.008761  -1.191    0.234
## B           -0.006529   0.008881  -0.735    0.462
## E            0.498024   0.008761  56.846   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.876 on 9997 degrees of freedom
## Multiple R-squared:  0.2449, Adjusted R-squared:  0.2448
## F-statistic:  1621 on 2 and 9997 DF,  p-value: < 2.2e-16
```

```r
tidy_ab_adj_e <- tidy(lm_ab_adj_e, conf.int=TRUE)
tidy_ab_adj_e
```

```
## # A tibble: 3 x 7
##   term        estimate std.error statistic p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept) -0.0104    0.00876    -1.19    0.234  -0.0276    0.00674
## 2 B           -0.00653   0.00888    -0.735   0.462  -0.0239    0.0109
## 3 E            0.498     0.00876    56.8     0        0.481    0.515
```

```r
# Linear regression of A on B, adjusting for C and F
lm_ab_adj_cf = lm(data = sim_sem, formula = A ~ B + C + F )
summary(lm_ab_adj_cf)
```

```
##
## Call:
## lm(formula = A ~ B + C + F, data = sim_sem)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7920 -0.6439 -0.0018  0.6595  3.7793
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.010407   0.009777  -1.065    0.287
## B           -0.009079   0.015483  -0.586    0.558
## C           -0.272138   0.014465 -18.814   <2e-16 ***
## F            0.257898   0.012827  20.106   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.9775 on 9996 degrees of freedom
## Multiple R-squared:  0.05984,    Adjusted R-squared:  0.05956
## F-statistic: 212.1 on 3 and 9996 DF,  p-value: < 2.2e-16
```

```
tidy_ab_adj_cf <- tidy(lm_ab_adj_cf, conf.int=TRUE)
tidy_ab_adj_cf
```

```
## # A tibble: 4 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept) -0.0104    0.00978    -1.06  2.87e- 1  -0.0296   0.00876
## 2 B           -0.00908   0.0155     -0.586 5.58e- 1  -0.0394   0.0213
## 3 C           -0.272     0.0145    -18.8   1.26e-77  -0.300   -0.244
## 4 F            0.258     0.0128     20.1   3.55e-88   0.233    0.283
```

```
# Linear regression of A on B, adjusting for C and G
lm_ab_adj_cg = lm(data = sim_sem, formula = A ~ B + C + G )
summary(lm_ab_adj_cg)
```

```
##
## Call:
## lm(formula = A ~ B + C + G, data = sim_sem)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8898 -0.6638  0.0098  0.6702  4.0909
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.011555   0.009967  -1.159 0.246325
## B           -0.039502   0.028615  -1.380 0.167471
## C           -0.169897   0.018612  -9.128  < 2e-16 ***
## G           -0.072294   0.020617  -3.507 0.000456 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9964 on 9996 degrees of freedom
## Multiple R-squared:  0.02302,    Adjusted R-squared:  0.02272
## F-statistic:  78.5 on 3 and 9996 DF,  p-value: < 2.2e-16
```

```
tidy_ab_adj_cg <- tidy(lm_ab_adj_cg, conf.int=TRUE)
tidy_ab_adj_cg
```

```
## # A tibble: 4 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)  -0.0116   0.00997     -1.16 2.46e- 1  -0.0311   0.00798
## 2 B            -0.0395   0.0286      -1.38 1.67e- 1  -0.0956   0.0166
## 3 C            -0.170    0.0186      -9.13 8.30e-20  -0.206   -0.133
## 4 G            -0.0723   0.0206      -3.51 4.56e- 4  -0.113   -0.0319
```

```
# Linear regression of A on B, adjusting for all variables
lm_ab_adj_all = lm(data = sim_sem, formula = A ~ B + C + D + E + F + G + H )
summary(lm_ab_adj_all)
```

```
##
## Call:
```

```
## lm(formula = A ~ B + C + D + E + F + G + H, data = sim_sem)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3452 -0.5915  0.0023  0.5840  3.2198
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0105945  0.0087626  -1.209   0.2267
## B           -0.0236763  0.0251580  -0.941   0.3467
## C            0.0004633  0.0183685   0.025   0.9799
## D            0.0106630  0.0110315   0.967   0.3338
## E            0.5066043  0.0122607  41.319   <2e-16 ***
## F           -0.0004507  0.0129789  -0.035   0.9723
## G            0.0214940  0.0188558   1.140   0.2543
## H            0.0164169  0.0092715   1.771   0.0766 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8759 on 9992 degrees of freedom
## Multiple R-squared:  0.2453, Adjusted R-squared:  0.2448
## F-statistic:   464 on 7 and 9992 DF,  p-value: < 2.2e-16
```

```r
tidy_ab_adj_all <- tidy(lm_ab_adj_all, conf.int=TRUE)
tidy_ab_adj_all
```

```
## # A tibble: 8 x 7
##   term         estimate std.error statistic p.value conf.low conf.high
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept) -0.0106     0.00876   -1.21    0.227  -0.0278    0.00658
## 2 B           -0.0237     0.0252    -0.941   0.347  -0.0730    0.0256
## 3 C            0.000463   0.0184     0.0252  0.980  -0.0355    0.0365
## 4 D            0.0107     0.0110     0.967   0.334  -0.0110    0.0323
## 5 E            0.507      0.0123    41.3     0        0.483    0.531
## 6 F           -0.000451   0.0130    -0.0347  0.972  -0.0259    0.0250
## 7 G            0.0215     0.0189     1.14    0.254  -0.0155    0.0585
## 8 H            0.0164     0.00927    1.77    0.0766 -0.00176   0.0346
```

```r
# Linear regression of A on B, adjusting for nothing
lm_ab_adj_0 = lm(data = sim_sem, formula = A ~ B )
summary(lm_ab_adj_0)
```

```
##
## Call:
## lm(formula = A ~ B, data = sim_sem)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8972 -0.6630  0.0031  0.6785  3.8940
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.01228    0.01008  -1.218  0.22318
## B            0.02968    0.01019   2.913  0.00358 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.008 on 9998 degrees of freedom
## Multiple R-squared:  0.0008482,  Adjusted R-squared:  0.0007483
## F-statistic: 8.487 on 1 and 9998 DF,  p-value: 0.003584
```

```
tidy_ab_adj_0 <- tidy(lm_ab_adj_0, conf.int=TRUE)
tidy_ab_adj_0
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)  -0.0123    0.0101     -1.22 0.223   -0.0320    0.00748
## 2 B             0.0297    0.0102      2.91 0.00358  0.00971   0.0497
```

```
# Table of results

ab_adj_e <- c (tidy_ab_adj_e$term[2],
             tidy_ab_adj_e$estimate[2],
             tidy_ab_adj_e$std.error[2],
             tidy_ab_adj_e$conf.low[2],
             tidy_ab_adj_e$conf.high[2],
             (tidy_ab_adj_e$conf.high[2] - tidy_ab_adj_e$conf.low[2])
                )
ab_adj_cf <- c (tidy_ab_adj_cf$term[2],
             tidy_ab_adj_cf$estimate[2],
             tidy_ab_adj_cf$std.error[2],
             tidy_ab_adj_cf$conf.low[2],
             tidy_ab_adj_cf$conf.high[2],
             (tidy_ab_adj_cf$conf.high[2] - tidy_ab_adj_cf$conf.low[2])
                )

ab_adj_cg <- c (tidy_ab_adj_cg$term[2],
             tidy_ab_adj_cg$estimate[2],
             tidy_ab_adj_cg$std.error[2],
             tidy_ab_adj_cg$conf.low[2],
             tidy_ab_adj_cg$conf.high[2],
             (tidy_ab_adj_cg$conf.high[2] - tidy_ab_adj_cg$conf.low[2])
                )

ab_adj_all <- c (tidy_ab_adj_all$term[2],
             tidy_ab_adj_all$estimate[2],
             tidy_ab_adj_all$std.error[2],
             tidy_ab_adj_all$conf.low[2],
             tidy_ab_adj_all$conf.high[2],
             (tidy_ab_adj_all$conf.high[2] - tidy_ab_adj_all$conf.low[2])
                )

ab_adj_0 <- c (tidy_ab_adj_0$term[2],
             tidy_ab_adj_0$estimate[2],
             tidy_ab_adj_0$std.error[2],
             tidy_ab_adj_0$conf.low[2],
             tidy_ab_adj_0$conf.high[2],
             (tidy_ab_adj_0$conf.high[2] - tidy_ab_adj_0$conf.low[2])
                )
```

```r
table_ab <- rbind(ab_adj_e, ab_adj_cf, ab_adj_cg, ab_adj_all, ab_adj_0)

colnames(table_ab) <- c("Var", "Estimate", "SE", "CI low", "CI high", "CI length")

kable(table_ab)
```

	Var	Estimate	SE	CI low	CI high	CI length
ab_adj_e	B	-0.00652894617052005	0.00888078158990123	-0.0239370658866111	0.010879173545571	0.0348162394321821
ab_adj_cf	B	-0.00907879018857182	0.0154825719074832	-0.0394277483084083	0.0212701679312646	0.0606979162396729
ab_adj_cg	B	-0.0395018514965009	0.0286146270344121	-0.0955922816053627	0.0165885786123609	0.112180860217723
ab_adj_all	B	-0.0236763100548855	0.0251579606631312	-0.0729909805193557	0.0256383604095848	0.0986293409289405
ab_adj_0	B	0.0296835697057907	0.0101889501340964	0.00971117653612038	0.049655962875461	0.0399447863393406

```r
# Are the point estimates similar?
# Do the estimates have similar variance (or confidence interval length)?
# Compare also these estimates against an approach which simply adjusts for all other variables in the 
# How are the results different (if they are) and what is the explanation?
```

Summary:

	Estimate	SE	CI low	CI high
ef_adj_c	0.377905443851971	0.00759644322956105	0.363014885873711	0.392796001830231
ef_adj_all	0.322355599401436	0.00970417668831715	0.303333458381291	0.341377740421582
ef_adj_0	0.20934502526251	0.00970875778721681	0.190313905749372	0.228376144775649

	Estimate	SE	CI length
ef_adj_c	0.377905443851971	0.00759644322956105	0.02978111595652
ef_adj_all	0.322355599401436	0.00970417668831715	0.038044282040291
ef_adj_0	0.20934502526251	0.00970875778721681	0.0380622390262762

When exploring the effect of E on F, there was only one set of sufficient adjustment variables from dagitty. Compared to the model with all of the variables in the equation, the point estimates are not similar and the confidence intervals do not usually overlap. The point estimates also have relatively different variance (and CI length). Compared to the unadjusted model, the point estimates are different but the unadjusted variances/CI lengths are lower.

	Estimate	SE	CI low	CI high
ab_adj_e	-0.00652894617052005	0.00888078158990123	-0.0239370658866111	0.010879173545571
ab_adj_cf	-0.00907879018857182	0.0154825719074832	-0.0394277483084083	0.0212701679312646
ab_adj_cg	-0.0395018514965009	0.0286146270344121	-0.0955922816053627	0.0165885786123609
ab_adj_all	-0.0236763100548855	0.0251579606631312	-0.0729909805193557	0.0256383604095848
ab_adj_0	0.0296835697057907	0.0101889501340964	0.00971117653612038	0.049655962875461

	Estimate	SE	CI length
ab_adj_e	-0.00652894617052005	0.00888078158990123	0.0348162394321821
ab_adj_cf	-0.00907879018857182	0.0154825719074832	0.0606979162396729
ab_adj_cg	-0.0395018514965009	0.0286146270344121	0.112180860217723
ab_adj_all	-0.0236763100548855	0.0251579606631312	0.0986293409289405
ab_adj_0	0.0296835697057907	0.0101889501340964	0.0399447863393406

For the effect of B on A, the point estimates are very similar for all three minimally sufficient sets, as are the variances (or confidence interval lengths). Additionally, they are all similar to the regression where all of the variables were adjusted for in the model. It seems that the sufficient adjustment sets do indeed sufficiently adjust as needed. Compared to the unadjusted model, the point estimates are different when no adjustment is performed, but the variances/CI lengths are similar. Usually, in the adjusted models, the CI for the estimate crosses zero, while it does not for the unadjusted.

When comparing the results of the different adjustments for the effect of E on F versus the effect of B on A, we see that adjusting for all of the variables in the E on F model causes the effect estimate to change much more than for the model looking at the effect of B on A, and the variances/CI lengths are markedly different for E on F as well as compared to those of the B on A. Both models have different point estimates when adjusted. This makes sense, because adjusting for appropriate variables should change the result (make it more accurate). When and when not adjusted, the first model (E on F) shows an effect. When not adjusted, the second model (B on A) shows an effect, but when adjusted, the CI crosses zero, which does not suggest an effect.