Chapter 6 Solutions

https://rpubs.com/Thousandslayers/677948

6E1. List three mechanisms by which multiple regression can produce false inferences about causal effects

The three mechanisms discussed in the chapter are multi-collinearity, post-treatment bias, and collider bias. Multi-collinearity manifests via the “multi-leg” problem – if you use both legs to predict height, the coefficient on one of the legs will be almost 0. Post-treatment bias is similar to including future observations into the training set – you are including information that comes after the test is run, which the model wouldn’t have access to at prediction time. Collider bias occurs when a variable is included in a model –called a collider, which creates the illusion of a causal effect when there really is not one.

6E2. Provide an example

Multi-collinearity: You want to predict sales price of a car. If you use the age of the car and its mileage, the effect of each will be smaller when they are both included compared to if you just used one of them to predict sale price. Essentially, if you want to know the causal effect of one variable on another, then you should make sure they aren’t collinear. But if you just fit an outcome to one variable, then you can be tripped up by masked relationships

```r
library(data.table)
age <- rep(c(2,5,6,10),1000)
mileage <- rep(c(40000, 75000, 180000, 60000),1000)
price <- rep(c(20000, 14000, 11000, 8000),1000)
df = data.table(age = age, price = price, mileage = mileage)

# -3.36e-02, mileage
# -1488, age
# -2.2e02*mileage + -1425*age
lm(price ~ mileage, data = df)
```

```
##
## Call:
## lm(formula = price ~ mileage, data = df)
##
## Coefficients:
## (Intercept)      mileage
##  16232.0000      -0.0336
```

Post Treatment Bias: Suppose you randomly split a group of people into two groups: one to eat vegetarian for a month and one to eat paleo for a month. You went to test the effect of a treatment (no meat) on blood pressure. You measure the baseline blood pressure of both groups and the baseline weight of both groups before you start the dietary regimen. After a month is over, you re-measure blood pressure and re-weight the participants.

If you now want to build a model to predict the change in height, you can’t use the change in weight as a variable because it is a post-treatment effect. Weightloss is mostly a consequence of diet – once we already know if there was weightloss, does the diet matter? If the diet already has its effects on blood pressure by reducing weight. Correct inference for the reason that blood pressure dropped is from the diet, the weight loss wouldn't have occurred without the diet. (Weight loss doesn't cause diet).

Collider Bias: Lung cancer, smoking, and coffee. If you want to know the causal effect of drinking coffee on cancer, "smoking" would be an example of a collider. Coffee and smoking are both addictive substances, so people who drink coffee may be more likely to smoke. So if you condition on smoking, then it may appear that coffee has a positive effect on lung cancer, even though in reality it has none.

6E3. List the four elemental confounds

Fork: X <- Z -> Y. X and Y are independent, conditional on Z Pipe: X -> Z -> Y. X and Y are independent, conditional on Z Collider: X -> Z <- Y. No association between X and Y, unless condition on Z. Descendant: Condition on a descendent of Z in the pipe will weakly close the pipe

6E4. How is a biased sample like conditioning on a collider? Think of the example at the opening of the chapter

The biased sample was the newsworthy/trustworthy thing. Newsworthy -> Acceptance <- Trustworthy. Selection bias is like collider bias?

6M1. Modify the DAG on page 190 to include the variable V, an unobserved cause of C and Y: C ← V → Y. Reanalyze the DAG. How many paths connect X to Y? Which must be closed? Which variables should you condition on now?

To add v, you include it in the path C -> Y as C <- V -> Y. Now there are (5) paths from X to Y:

(D) X -> Y

(1) X <- U <- A -> C <- V -> Y (C is a collider)
(2) X <- U <- A -> C <- Y (C is a collider)
(3) X <- U -> B <- C <- V -> Y (B is a collider)
(4) X <- U -> B <- C <- Y (B is a collider)

C and B are both colliders, so we can only consider variables A and V. The reason we choose to shut down variable A is…

#6M2. What matters is conditional association – how is variable X associated with Y, conditional on Z? Consider a DAG X -> Z -> Y. Simulate data from this DAG so cor(X, Z) is large. Then include both in a model predicting Y. Do you observe multi-collinearity? What is the difference from the legs example?

```
x <- sort(rnorm(100, 5, 1))
z <- sort(rnorm(100, 5, 1))
cor(x, z)
```

```
## [1] 0.984735
```

```
library(rethinking)
```

```
## Loading required package: rstan

## Loading required package: StanHeaders

## Loading required package: ggplot2
## rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
## Loading required package: parallel
```

```
## rethinking (Version 2.13)
```

```
##
## Attaching package: 'rethinking'
```

```
## The following object is masked from 'package:stats':
##
##     rstudent
```

```
y <- rnorm(100, 10, 2)
d <- data.table(x = x, y = y, z= z)

flist <- alist(y ~ dnorm(mu, sigma),
      mu <- alpha + b1*x + b2*z,
      alpha ~ dnorm(0,1),
      b1 ~ dnorm(0,1),
      b2 ~ dnorm(0,1),
      sigma ~ dunif(0,10))

mod <- quap(flist, d)
precis(mod)
```

```
##             mean        sd        5.5%    94.5%
## alpha 3.6954484 0.8808353  2.28770349 5.103193
## b1    0.2575202 0.6260961 -0.74310232 1.258143
## b2    1.0197666 0.6516556 -0.02170495 2.061238
## sigma 2.4830070 0.1920194  2.17612287 2.789891
```

We do have multi-collinarity because the model thinks x is important but z is not, even thought they are basically the same information.

```
flist <- alist(y ~ dnorm(mu, sigma),
      mu <- alpha + b2*z,
      alpha ~ dnorm(0,1),
      b2 ~ dnorm(0,1),
      sigma ~ dunif(0,10))

mod <- quap(flist, d)
precis(mod)
```

```
##            mean        sd      5.5%    94.5%
## alpha 3.704659 0.8778119 2.3017456 5.107572
## b2    1.277714 0.1784045 0.9925897 1.562839
## sigma 2.477730 0.1906441 2.1730439 2.782416
```

This seems very similar to the legs example. The key difference is that in the legs example, leg length was highly correlated with height. In this example, x and z are not necessarily correlated with y.

6M3.

a) Three paths from X to Y, you should condition on Z X -> Y X <- Z <- A -> Y (pipe) X <- Z -> Y (fork)

b) Three paths from X to Y, Z is a collider. You should condition on A.

X -> Y X -> Z <- A -> Y (descendant) X <- Z -> Y (fork)

c) No open backdoor path from X to Y (the path through A goes through the collider)

X -> Y X -> Z <- Y X <- A -> Z <- Y

d) A is a collider. You should condition on Z.

X to Y X -> Z -> Y X <-A -> Z -> Y

6H1: Use the Waffle House data, data(WaffleDivorce), to find the total causal influence of number of Waffle Houses on divorce rate. Justify your model or models with a causal graph.

```r
library(rethinking)
data(WaffleDivorce)
d <- WaffleDivorce
head(d)
```

```
##     Location Loc Population MedianAgeMarriage Marriage Marriage.SE Divorce
## 1    Alabama  AL       4.78              25.3     20.2        1.27    12.7
## 2     Alaska  AK       0.71              25.2     26.0        2.93    12.5
## 3    Arizona  AZ       6.33              25.8     20.3        0.98    10.8
## 4   Arkansas  AR       2.92              24.3     26.4        1.70    13.5
## 5 California  CA      37.25              26.8     19.1        0.39     8.0
## 6   Colorado  CO       5.03              25.7     23.5        1.24    11.6
##   Divorce.SE WaffleHouses South Slaves1860 Population1860 PropSlaves1860
## 1       0.79          128     1     435080         964201           0.45
## 2       2.05            0     0          0              0           0.00
## 3       0.74           18     0          0              0           0.00
## 4       1.22           41     1     111115         435450           0.26
## 5       0.24            0     0          0         379994           0.00
## 6       0.94           11     0          0          34277           0.00
```

Using the dag from earlier in the chapter, we learned that conditioning on "South" should break the path from wafflehouses to divorce.

```r
d2 = d[, c('Population', 'MedianAgeMarriage', 'Marriage', 'WaffleHouses', 'South', 'Slaves1860', 'Popula
pc = prcomp(d2, scale = TRUE)
pc$rotation
```

```
##                           PC1         PC2         PC3         PC4         PC5
## Population         0.13330139  0.36873554 -0.71412128 -0.54721911  0.11123224
## MedianAgeMarriage -0.09559427  0.55884675  0.39903466 -0.16238703 -0.20642693
## Marriage           0.02297226 -0.57549778 -0.37298268  0.11519777 -0.23560823
## WaffleHouses       0.46039253  0.01122304 -0.00055666 -0.11379843 -0.80876613
## South              0.49008156 -0.06576871  0.02642693  0.02773498  0.43839180
## Slaves1860         0.50265197 -0.02243303  0.17849145  0.12230256  0.08006740
## Population1860     0.13189314  0.46417234 -0.35915451  0.78233696 -0.04206322
## PropSlaves1860     0.50036860 -0.01077215  0.17359784 -0.14206629  0.18717355
##                          PC6         PC7         PC8
## Population        -0.07255193  0.07225228 -0.11827974
## MedianAgeMarriage -0.60245382 -0.29430796 -0.02782056
## Marriage          -0.65850074 -0.15965459  0.03555178
## WaffleHouses       0.31334678 -0.14439231  0.04271265
## South              0.08601990 -0.74376497 -0.03540903
## Slaves1860        -0.20619165  0.40078168 -0.70033485
## Population1860    -0.04391044  0.02537909  0.14752501
## PropSlaves1860    -0.21939380  0.38388451  0.68458841
```

```
summary(pc)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5    PC6     PC7
## Standard deviation     1.8672 1.4761 0.9529 0.75584 0.58372 0.5254 0.39075
## Proportion of Variance 0.4358 0.2724 0.1135 0.07141 0.04259 0.0345 0.01909
## Cumulative Proportion  0.4358 0.7082 0.8217 0.89310 0.93569 0.9702 0.98928
##                            PC8
## Standard deviation     0.29288
## Proportion of Variance 0.01072
## Cumulative Proportion  1.00000
```

PC1 explains 43%, PC2 explains 27%. PC 1 is mainly South + WaffleHouses + Slaves1860 + PropSlaves1860, PC2 is mainly marriage and age at marriage. PC 3 is population.

```
lm(Divorce ~ South, data = d)$coef
```

```
## (Intercept)       South
##    9.300000    1.385714
```

```
lm(Divorce ~ South + WaffleHouses, data = d)$coef
```

```
##  (Intercept)        South WaffleHouses
## 9.2958758398 1.2939325104 0.0009221725
```

```
lm(Divorce ~ WaffleHouses, data = d)$coef
```

```
##  (Intercept) WaffleHouses
##  9.460231241  0.007042942
```

```
flist <- alist(Divorce ~ dnorm(mu, sigma),
               mu <- alpha + b1*South + b2*WaffleHouses,
               alpha ~ dnorm(0),
               b1 ~ dnorm(10, 5),
               b2 ~ dnorm(0, 1),
               sigma ~ dnorm(2,1))

model <- quap(flist, data = d)
precis(model)
```

```
##              mean          sd         5.5%       94.5%
## alpha 8.472814853 0.339761222  7.929810799 9.015818907
## b1    2.171353272 0.802360113  0.889026844 3.453679701
## b2    0.001124007 0.005445115 -0.007578339 0.009826353
## sigma 1.835415344 0.210467701  1.499047308 2.171783381
```

The true causal effect of waffle houses on divorce rate is 0. The causal graph is something like W <- S -> D.

6H2. Build a series of models to test the implied conditional independencies of the causal graph you used in the previous problem. If any of the tests fail, how do you think the graph needs to be amended? Does the graph need more or fewer arrows? Feel free to nominate variables that aren't in the data.

```
# build a model with only waffles
flist <- alist(Divorce ~ dnorm(mu, sigma),
               mu <- alpha + b2*WaffleHouses,
               alpha ~ dnorm(0),
               b2 ~ dnorm(0, 1),
               sigma ~ dnorm(2,1))

model <- quap(flist, data = d)
precis(model)
```

```
##            mean          sd       5.5%      94.5%
## alpha 8.6964056 0.325410652 8.17633649 9.21647464
## b2    0.0117145 0.004157669 0.00506974 0.01835926
## sigma 1.8769500 0.211971823 1.53817808 2.21572191
```

I would have expected this to show that waffle houses have a stronger effect because the unobserved south should show up in the coefficient for waffles, since there is a path from W to D through S in my DAG. This test failed. I should probably amend the dag to include variables A (age at marriage) and M (marriage rate) somehow.

```
# build a model with only "south", expect this to show a positive effect of b1 of about 2
flist <- alist(Divorce ~ dnorm(mu, sigma),
               mu <- alpha + b1*South,
               alpha ~ dnorm(0),
               b1 ~ dnorm(10, 5),
               sigma ~ dnorm(2,1))

model <- quap(flist, data = d)
precis(model)
```

```
##           mean        sd     5.5%    94.5%
## alpha 8.477261 0.3389431 7.935565 9.018958
## b1    2.282814 0.5976924 1.327586 3.238041
## sigma 1.836065 0.2105462 1.499571 2.172558
```

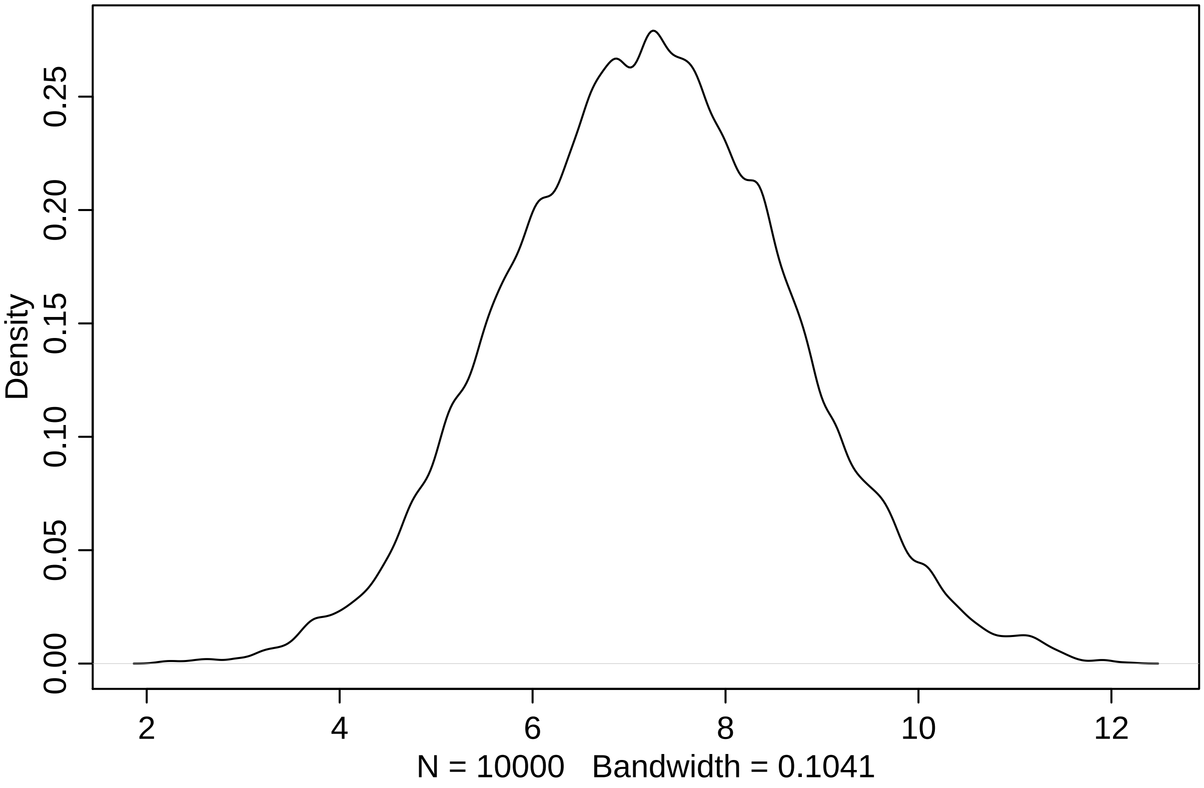
This test passed. South still has about the expected effect without accounting for waffles.

6H3. Prior Predictive simulation of model using area to predict fox weight, and eventual model predictions.

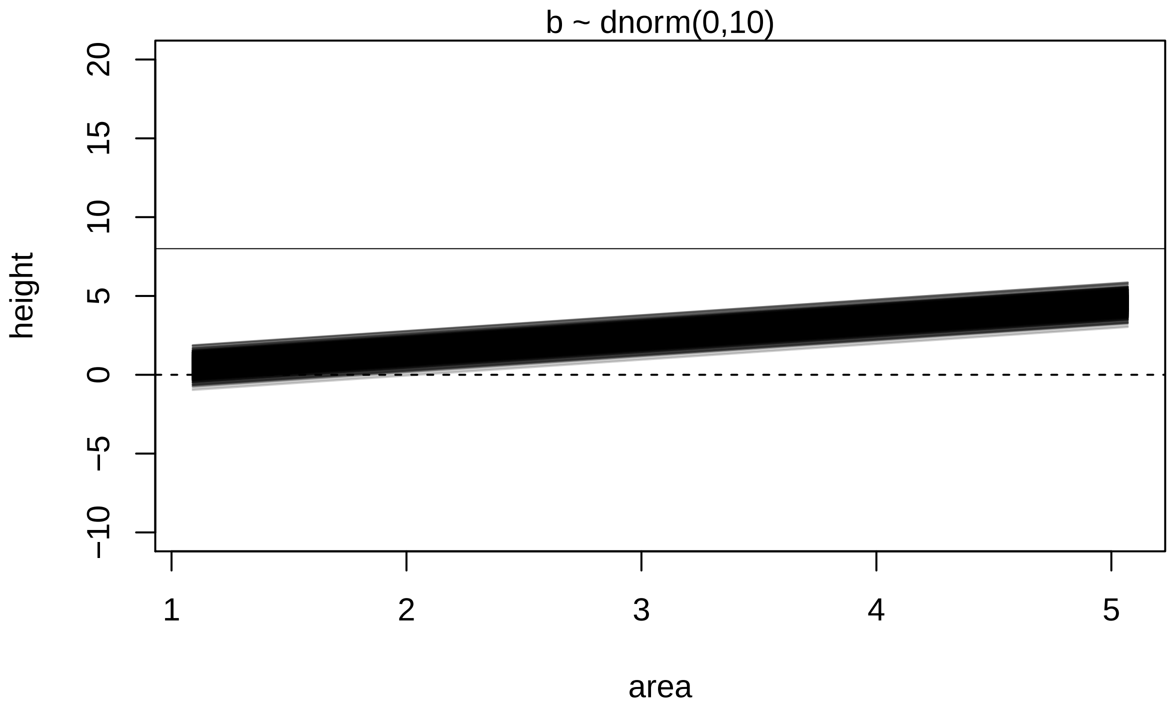
```
library(rethinking)
data(foxes)
d <- foxes
N <- 1000
a <- rnorm(N, 4,.5)
b1 <- rlnorm(N, 0,.01)
```

```
sigma <- runif(N, 1,1.01)
area <- sample(d$area, 100)

prior_h <- rnorm( 1e4 , a + b1*area , sigma )
dens( prior_h )
```



```
plot( NULL , xlim=range(d$area) , ylim=c(-10,20) ,
xlab="area" , ylab="height" )
abline( h=0 , lty=2 )
abline( h=8 , lty=1 , lwd=0.5 )
mtext( "b ~ dnorm(0,10)" )
xbar <- mean(d$weight)
for ( i in 1:N ) curve( a[i] + b1[i]*(x - xbar) ,
from=min(d$area) , to=max(d$area) , add=TRUE ,
col=col.alpha("black",0.2) )
```



```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
flist <- alist(weight ~ dnorm(mu, sigma),
               mu ~ alpha + b1 * area,
               alpha ~ dnorm(0, .2),
               b1 ~ dlnorm(0, .5),
               sigma ~ dexp(1))

foxes_scaled <- foxes %>% mutate(avgfood = (avgfood - mean(avgfood)) / sd(avgfood),
                                 groupsize = (groupsize - mean(groupsize)) / sd(groupsize),
                                 area = (area - mean(area)) / sd(area),
                                 weight = (weight - mean(weight)) / sd(weight)
                                 )

mod <- quap(flist, foxes_scaled)
precis(mod)
```

```
##               mean         sd       5.5%     94.5%
## alpha 1.589470e-08 0.08498906 -0.1358289 0.1358290
## b1    2.214507e-01 0.05894262  0.1272490 0.3156523
## sigma 1.011203e+00 0.06695721  0.9041925 1.1182136
```

Increasing area should make foxes heavier, according to this coefficient 89% CI of 1. Should show though that there is no causal relationship between area and weight…

6H4. Now infer the causal impact of adding food to a territory. Would this make foxes heavier? Which covariate do you need to adjust for to estimate the total causal influence of food?

Given our DAG, there are two paths from avgfood to weight. However, none of them are a backdoor. Thus, we do not need to adjust for any other variable to identify the causal effect of avgfood on weight.

```
flist <- alist(weight ~ dnorm(mu, sigma),
               mu ~ alpha + b1 * area + b2*avgfood,
               alpha ~ dnorm(5,.5),
               b1 ~ dlnorm(0,.01),
               b2 ~ dlnorm(0, .05),
               sigma ~ dunif(1,2))

#alpha ~ dnorm(0,1),
#b1 ~ dnorm(0,1),
#sigma ~ dunif(0,10))

mod <- quap(flist, d)
precis(mod)
```

```
##            mean          sd      5.5%    94.5%
## alpha 1.0403339 0.157077513 0.7892937 1.291374
## b1    0.9907572 0.009846881 0.9750200 1.006494
## b2    0.9496442 0.046260085 0.8757117 1.023577
## sigma 1.6336403 0.112945278 1.4531319 1.814149
```

Food makes foxes heavier. You need to adjust for groupsize. Should show though that food, on its own has no causal realtionship with weight.

6H5. Now infer the causal impact of group size. Which covariates do you need to adjust for? Looking at the posterior distribution of the resulting model, what do you think explains these data? That is, can you explain the estimates for all three problems? How do they go together?

Given our DAG, there are two paths from groupsize to weight. And of them has a backdoor through which our estimates will be confounded. That is, given that in our bayesian network the information flows freely, if we run an univariate regression, the coefficient for groupsize will pick up the effect of avgfood on weight too. Therefore, we need to control for avgfood to close this backdoor.

- Conditioning on groupsize, the average food available increases the weight of the foxes
- The larger the groupsize, adjusting for avgfood, the lower the weight of the foxes.
- Avgfood and area have two causal channels through which it influences the foxes' weight. It increases the food available to them, which helps them get heavier. But it also increases the groupsize. Thus,

they get thinner. These effects in opposite directions end up cancelling the overall causal effect of area or avgdfood on weight.

*If one were to intervene to increase the foxes' weight, one would need to increase the avgfood available to them while maintaining the groupsize constant.

6H6. Come back to this one...

6H7. Come back to this one...