# Assignment 2 - Methods 4

Laurits Lyngbaek

2025-03-18

## Contents

```r
pacman::p_load(rethinking, dplyr, tidyverse)
```

## Second assignment

The second assignment uses chapter 3, 5 and 6. The focus of the assignment is getting an understanding of causality.

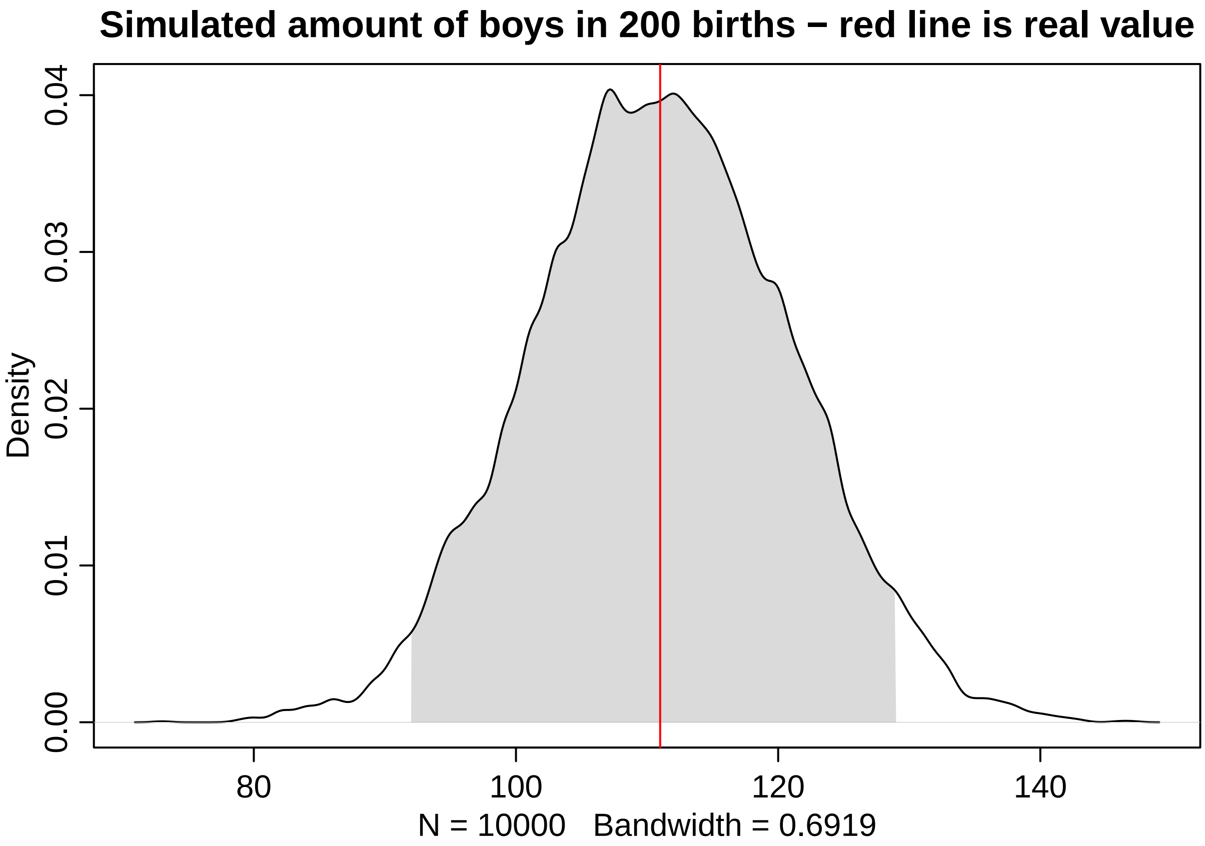
### Chapter 3: Causal Confussion

**Reminder: We are tying to estimate the probability of giving birth to a boy** I have pasted a working solution to questions 6.1-6.3 so you can continue from here:)

**3H3** Use rbinom to simulate 10,000 replicates of 200 births. You should end up with 10,000 numbers, each one a count of boys out of 200 births. Compare the distribution of predicted numbers of boys to the actual count in the data (111 boys out of 200 births).

```r
# 3H1
# Find the posterior probability of giving birth to a boy:
pacman::p_load(rethinking, dplyr)
data(homeworkch3)

set.seed(1)

W <- sum(birth1) + sum(birth2) ## count boys
N <- length(birth1) + length(birth2) ## count children

## defining grid and prior
p_grid <-seq(from =0, to = 1, len =1000)
```

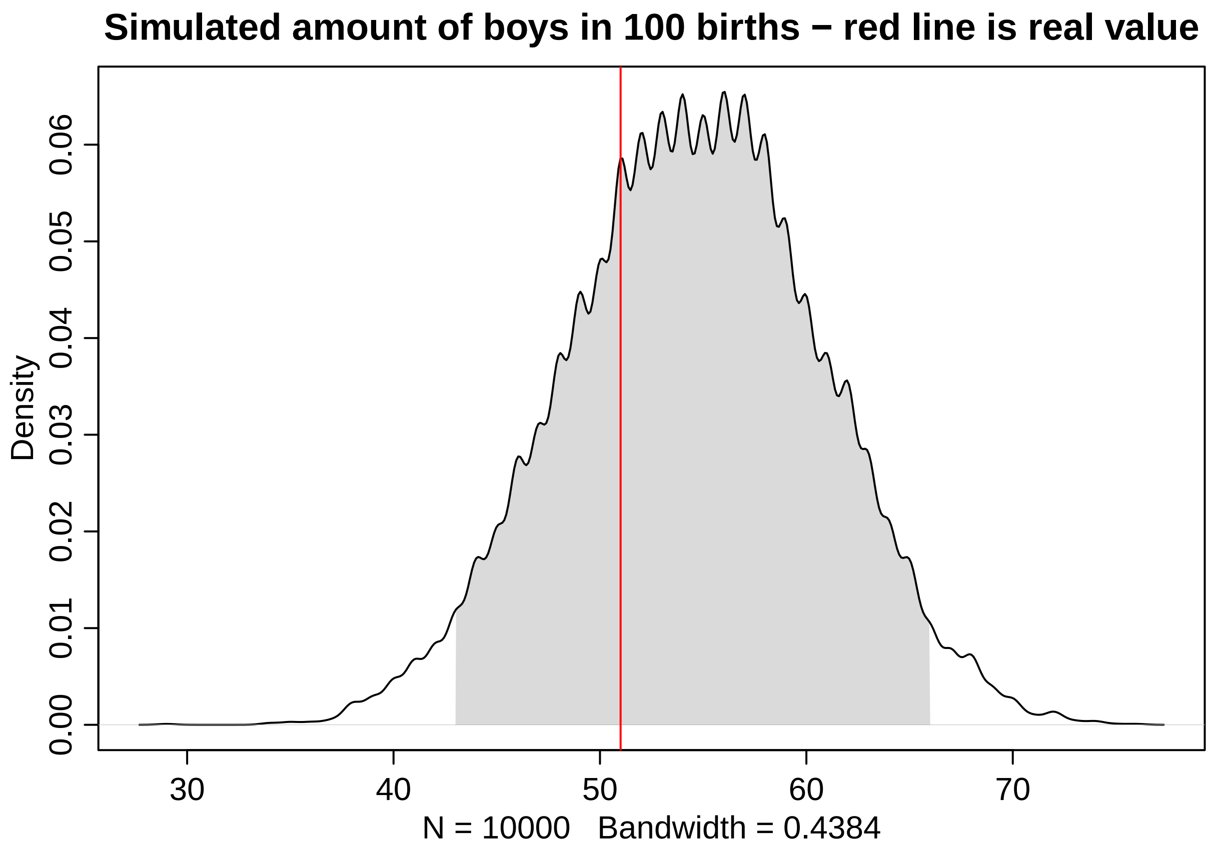
```
prob_p <- rep(1,1000)

prob_data <- dbinom(W,N,prob=p_grid) ## likelihood
unst.posterior <-prob_data * prob_p ## non-standardised posterior
posterior <- unst.posterior / sum(unst.posterior) ## standardised posterior



# 3H2
# Sample probabilities (i.e. parameter = boy birth prob.) from posterior distribution:
samples <- sample(p_grid, prob = posterior, size =1e4, replace =TRUE)


# 3H3
# Simulate births using sampled probabilities as simulation input, and check if they allign with real v
simulated_births <- rbinom(n = 1e4, size = N, prob = samples)
rethinking::dens(simulated_births,show.HPDI = 0.95)
abline(v=W, col="red")
title("Simulated amount of boys in 200 births - red line is real value")
```



**3H4.** Now compare 10,000 counts of boys from 100 simulated first borns only to the number of boys in the first births, birth1. How does the model look in this light?

```
### the posterior (of boy birth prob.) is kept, assuming that sex of first and second births are indepe
```

```
# Sample probabilities (i.e. parameter = boy birth prob.) from posterior distribution:
samples <- sample(p_grid, prob = posterior, size = length(birth1), replace =TRUE)


# Simulate births using sampled probabilities as simulation input, and check if they allign with real v
simulated_births <- rbinom(n = 10000, size = length(birth1), prob = samples)

## plotting
rethinking::dens(simulated_births,show.HPDI = 0.95)
abline(v=sum(birth1), col="red")
title("Simulated amount of boys in 100 births - red line is real value")
```

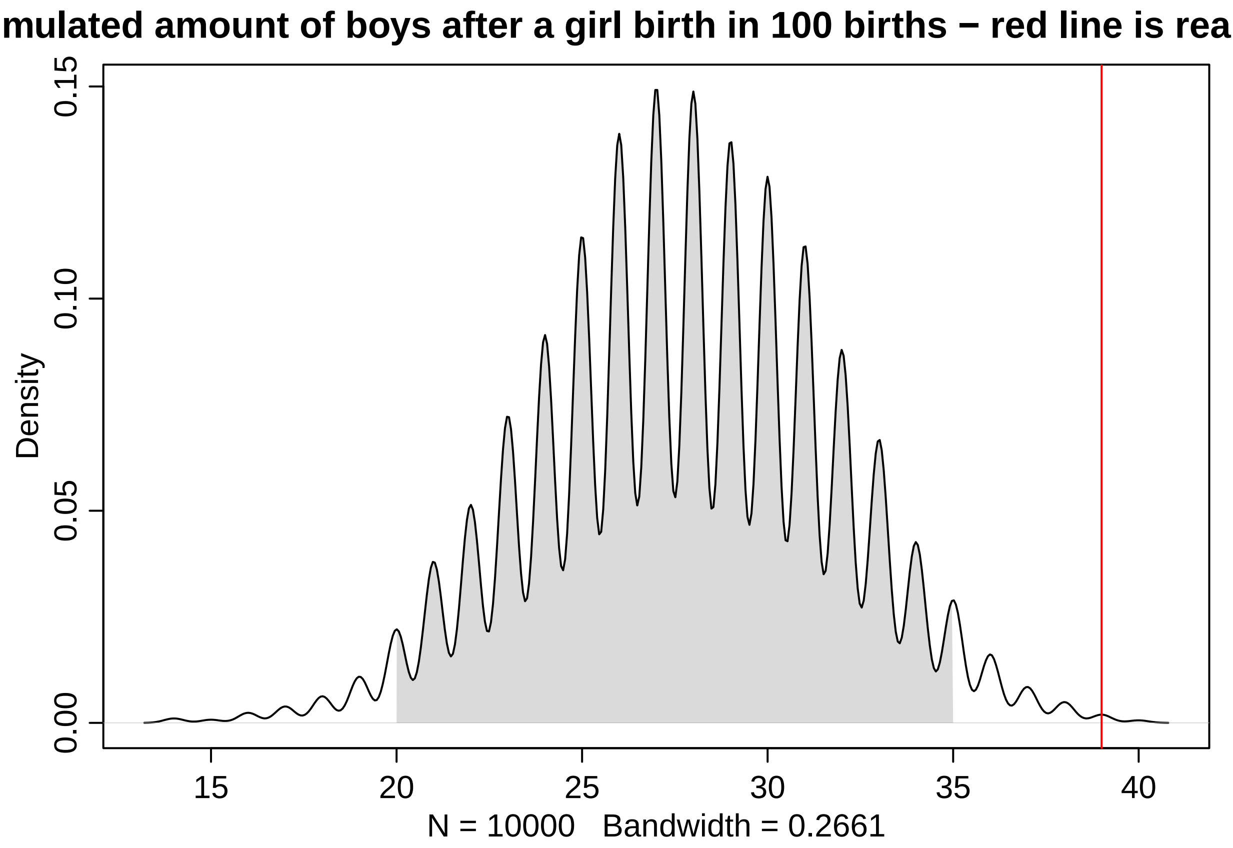


*Answer* - The model seems to perform less well. This is seen in how the actual amount of boy births (the red line) is further away from the peak / bulk of the posterior-sampled counts of boy births out of 100, than is the case for 3H3 with 200 births (first and second borns). Additionally the distribution is much less smooth.

**3H5.** The model assumes that sex of first and second births are independent. To check this assumption, focus now on second births that followed female first borns. Compare 10,000 simulated counts of boys to only those second births that followed girls. To do this correctly, you need to count the number of first borns who were girls and simulate that many births, 10,000 times. Compare the counts of boys in your simulations to the actual observed count of boys following girls. How does the model look in this light? Any guesses what is going on in these data?

```
# reminder: male=1, female=0

### the actual number of first born girls
```

```r
countFirstBornGirl <- 100 - sum(birth1)


### simulating births for countFirstBornGirl
samples <- sample(p_grid, prob = posterior, size = countFirstBornGirl, replace =TRUE) ## sampling proba

# Simulate births using sampled probabilities as simulation input
simulated_births <- rbinom(n = 10000, size = countFirstBornGirl, prob = samples)


# - - -


### the actual number of second births that followed female first borns
firstBirth <- as.logical(birth1)

births <- data.frame(first_birth = firstBirth, second_birth = birth2)


boyFollowingGirl <- filter(births, first_birth == FALSE)

CountBoyFollowingGirl <- sum(filter(births, first_birth == FALSE)) ## number of boy births following gi


## plotting
rethinking::dens(simulated_births, show.HPDI = 0.95)
abline(v=CountBoyFollowingGirl, col="red")
title("Simulated amount of boys after a girl birth in 100 births - red line is real value")
```



*Answer* - The model does is relatively poor job when compared to the actual observed count of boys following girls as it is within one of the distribution's tails. Additionally, the distribution is not smooth. This might be *REASON*

## Chapter 5: Spurrious Correlations

Start of by checking out all the spurrious correlations that exists in the world. Some of these can be seen on this wonderfull website: https://www.tylervigen.com/spurious/random All the medium questions are only asking you to explain a solution with words, but feel free to simulate the data and prove the concepts.

**5M1**. Invent your own example of a spurious correlation. An outcome variable should be correlated with both predictor variables. But when both predictors are entered in the same model, the correlation between the outcome and one of the predictors should mostly vanish (or at least be greatly reduced).

*Answer* - Such an example could be the effect of clouds in the sky and rain on people wearing raincoats. Whether people wear raincoats is likely highly correlated with it raining as well as (all though to a lesser degree) correlated with there being clouds in the sky. Since clouds must be present for it to rain the two predictor variables are likewise correlated. Since multiple regression answers "What is the value of knowing each predictor, after already knowing all of the other predictors", our question becomes "what is the value of knowing it is cloudy / rainy, after already knowing if it is rainy / cloudy. Since it cannot rain without clouds, the estimated parameter for the cloud-predictor would be greatly reduced or vanish since it adds little new information - depending on how likely people are to wear raincoats without it raining. The spurious correlation is the relationship between clouds and raincoat-wearing, which is mostly because of their shared association with rain, thus adding the cloud predictor in addition to rain is redundant. This is of course a more complex situation as other aspects also affect what jacket people wear, such as what jackets are available to them.

**5M2**. Invent your own example of a masked relationship. An outcome variable should be correlated with both predictor variables, but in opposite directions. And the two predictor variables should be correlated with one another.

*Answer* - Predicting disposable income from age and tax rate is one example, as disposable income does depend on both (income tends to increase with age and income decreases with higher tax rates) and the two predictor variables are likewise correlated, since people in the working population (often defined by age from childhood to pension age) are taxed more. This constitutes a masked relationship in that including both predictor variables likely makes the effect of both stronger than they would be individually. Again, this situation is more complicated in that higher tax rates are more common for people with higher wages as well, but other things being equal a higher tax rate would have a negative effect of the disposable income.

**5M3**. It is sometimes observed that the best predictor of fire risk is the presence of firefighters— States and localities with many firefighters also have more fires. Presumably firefighters do not cause fires. Nevertheless, this is not a spurious correlation. Instead fires cause firefighters. Consider the same reversal of causal inference in the context of the divorce and marriage data. How might a high divorce rate cause a higher marriage rate? Can you think of a way to evaluate this relationship, using multiple regression

*Answer* - High divorce rates do not cause higher marriage rates, but divorce rates would be able to predict marriage rates, because the higher the marriage rate the more people can get divorced. To evaluate this relationship with multiple regression we could add another predictor variable thought more predictive of marriage rate than divorce rate. The new predictor could be age at marriage (henceforth referred to as age) since the younger people marry, the more time they have to get divorced as well as more time to repeat that process. By comparing the estimated betas for models with both predictor variables together and separately, we would see a somewhat stable coefficient for divorce rate if it is the strongest driver for marriage rate, whereas the effect of age would be greatly reduced in the model with both predictors if it doesn't add much predictive power when already knowing the divorce rate. We would expect to see the opposite (i.e. the coefficient for age to remain stable and the coefficient for divorce rate to be greatly reduced upon adding age as a predictor, thus indicating that age is more predictive of marriage rates (as seen in the book). For the firefighter example the same could probably have been done by adding a mean of fires per year, as the number of fires usually seen in an area is likely predictive of the number of firefighters and thus of fire risk.

**5M5**. One way to reason through multiple causation hypotheses is to imagine detailed mechanisms through which predictor variables may influence outcomes. For example, it is sometimes argued that the price of gasoline (predictor variable) is positively associated with lower obesity rates (outcome variable). However, there are at least two important mechanisms by which the price of gas could reduce obesity. First, it could lead to less driving and therefore more exercise. Second, it could lead to less driving, which leads to less eating out, which leads to less consumption of huge restaurant meals. Can you outline one or more multiple regressions that address these two mechanisms? Assume you can have any predictor data you need.

*Answer* - The two mechanisms suggest that causally higher gasoline prices results in less time spent driving and thus 1) more time spent exercising (like walking) and 2) less time spent at restaurants (using time as a measure since more time at a restaurant should correlate to eating bigger or more meals). Similarly to 5M3, I would compare the estimated betas for models including the time spent walking & spent at restaurants predictors and gasoline prices together and separately like so:

Full model: $\mu_{obesityRate} = \alpha + \beta_{GP} * GP_i + \beta_{WT} * WT_i + \beta_{RT} * RT_i$ 1. Reduced model: $\mu_{obesityRate} = \alpha + \beta_{WT} * WT_i + \beta_{RT} * RT_i$ 2. Reduced model: $\mu_{obesityRate} = \alpha + \beta_{GP} * GP_i$

Where GP is gasoline prices, WT is time spent walking, Rt is time spent at restaurants and i represents an individual.

If obesity rates is a main driver for the time spent walking and the time spent at restaurants, then the estimated beta-coefficient for obesity rates would be somewhat stable when the sole predictor for obesity rate as well as in the full model, whereas the beta-coefficients for walking and restaurant time would diminish as (some of) their predictive power is overtaken by the gasoline price predictor.

## Chapter 5: Foxes and Pack Sizes

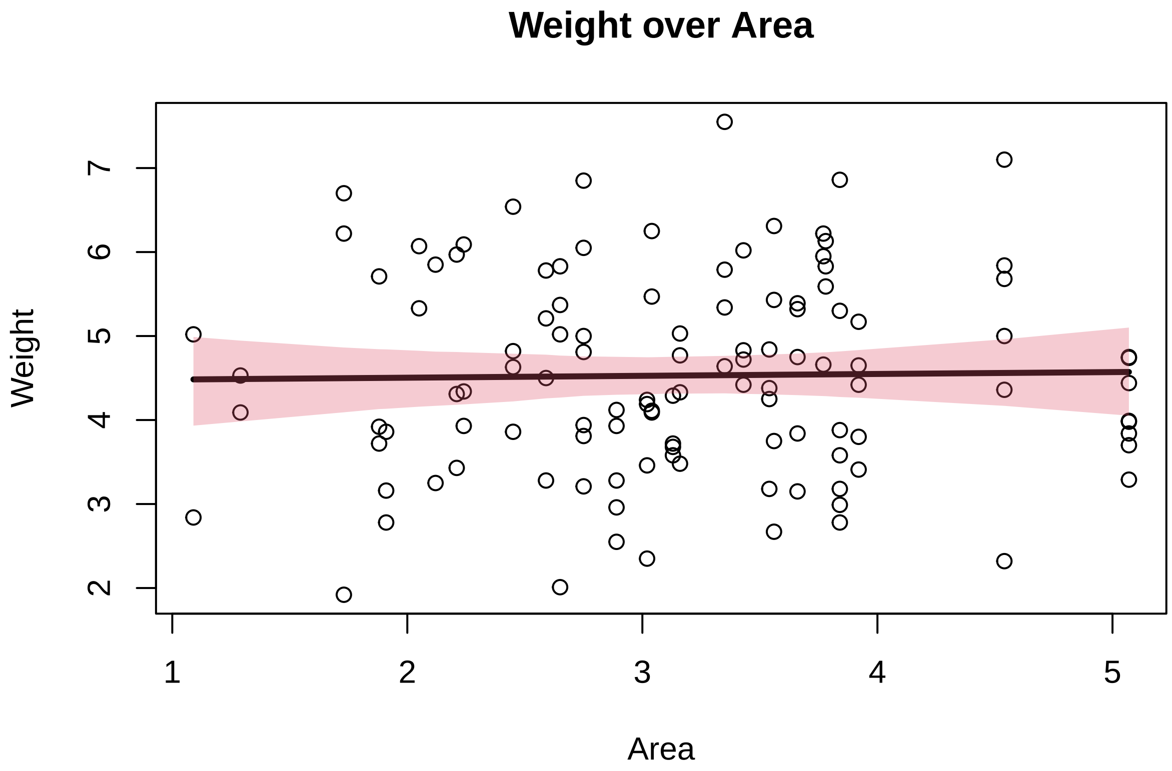
All five exercises below use the same data, data(foxes) (part of rethinking). The urban fox (Vulpes vulpes) is a successful exploiter of human habitat. Since urban foxes move in packs and defend territories, data on habitat quality and population density is also included. The data frame has five columns: (1) group: Number of the social group the individual fox belongs to (2) avgfood: The average amount of food available in the territory (3) groupsize: The number of foxes in the social group (4) area: Size of the territory (5) weight: Body weight of the individual fox

**5H1.** Fit two bivariate Gaussian regressions, using quap: (1) body weight as a linear function of territory size (area), and (2) body weight as a linear function of groupsize. Plot the results of these regressions, displaying the MAP regression line and the 95% interval of the mean. Is either variable important for predicting fox body weight?

```
data(foxes)
foxes_data <- foxes

set.seed(13)

### Regression Model 1

## quap formula   (priors are not well-founded but somewhat arbitrary)
foxes5H1_quapFormula1 <- alist(
    weight ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_a*area,
    alpha ~ dnorm( 5 , 5 ) , # Prior
    beta_a ~ dnorm( 0 , 1 )  , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )

foxes5H1_model1 <- quap(foxes5H1_quapFormula1, data=foxes_data)
precis(foxes5H1_model1)
```

```
##              mean         sd      5.5%     94.5%
## alpha  4.45530919 0.38717837  3.836523 5.0740950
## beta_a 0.02353426 0.11724821 -0.163851 0.2109195
## sigma  1.17867185 0.07738214  1.055000 1.3023435
```

```
## PLOTTING
plot(weight ~ area, data = foxes_data,
     xlab = "Area", ylab = "Weight", main = "Weight over Area")

## sequence of areas used for plotting = all unique areas
xseq <- sort(unique(foxes_data$area))



## superimposing the MAP regression line
mu <- link( foxes5H1_model1, data=list(area=xseq) ) ## retrieving the model's posterior to then generat

lines( xseq, apply(mu, 2, mean), lwd=3 ) ## adding the MAP regression line,    i.e. the mean

## 95% interval showing uncertainty about the regression line,   i.e. the 89% interval for the mean
shade( apply(mu, 2, PI, prob=0.95), xseq, col=col.alpha(2, 0.3) )
```

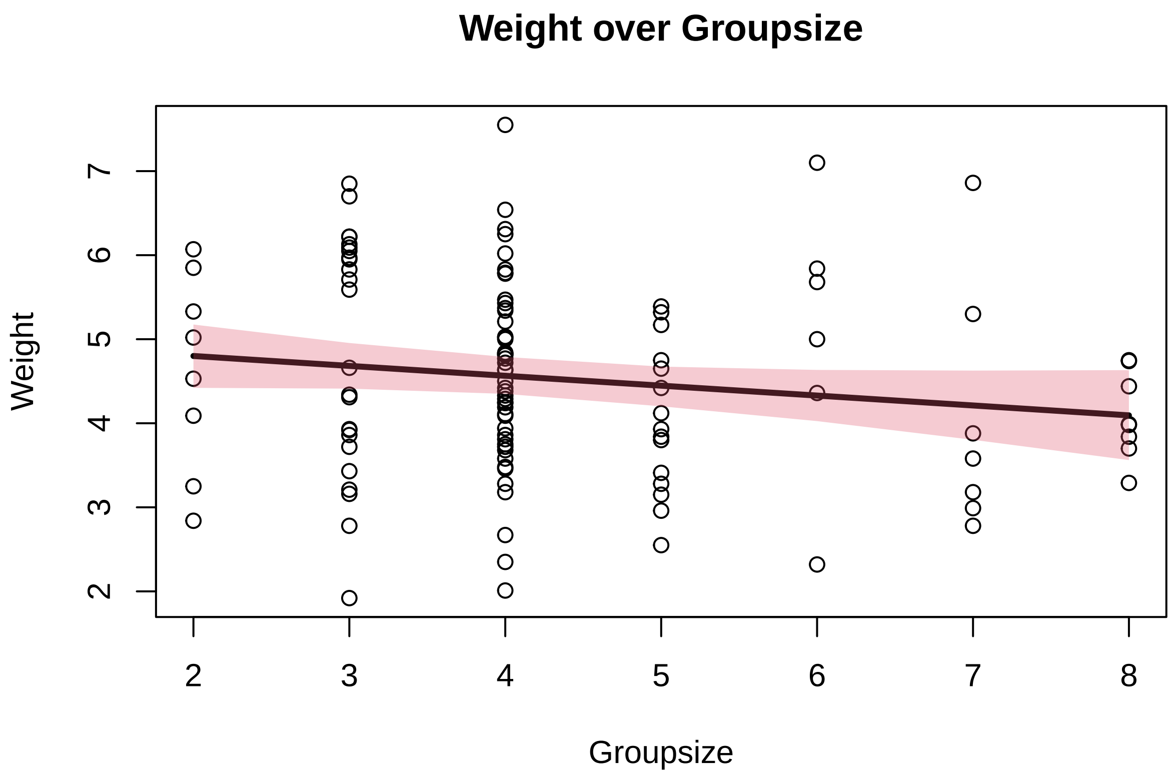


```
# - - -


### Regression Model 2

## quap formula   (priors are not well-founded but somewhat arbitrary)
foxes5H1_quapFormula2 <- alist(
    weight ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_g*groupsize,
    alpha ~ dnorm( 0 , 5 ) , # Prior
    beta_g ~ dnorm( 0 , 1 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


foxes5H1_model2 <- quap(foxes5H1_quapFormula2, data=foxes_data)
precis(foxes5H1_model2)
```

```
##              mean         sd       5.5%        94.5%
## alpha   5.0442971 0.32354013  4.5272174  5.561376662
## beta_g -0.1189817 0.07022527 -0.2312152 -0.006748139
## sigma   1.1636224 0.07640529  1.0415120  1.285732855
```

```
## PLOTTING
```

```
plot(foxes_data$groupsize, foxes_data$weight,
     xlab = "Groupsize", ylab = "Weight", main = "Weight over Groupsize")


## sequence of groupsize_ used for plotting = all unique groupsize
xseq <- sort(unique(foxes_data$groupsize))



## superimposing the MAP regression line
mu <- link( foxes5H1_model2, data=list(groupsize=xseq) ) ## retrieving the model's posterior to then ge

lines( xseq, apply(mu, 2, mean), lwd=3 ) ## adding the MAP regression line,    i.e. the mean

## 95% interval showing uncertainty about the regression line,   i.e. the 89% interval for the mean
shade( apply(mu, 2, PI, prob=0.95), xseq, col=col.alpha(2, 0.3) )
```



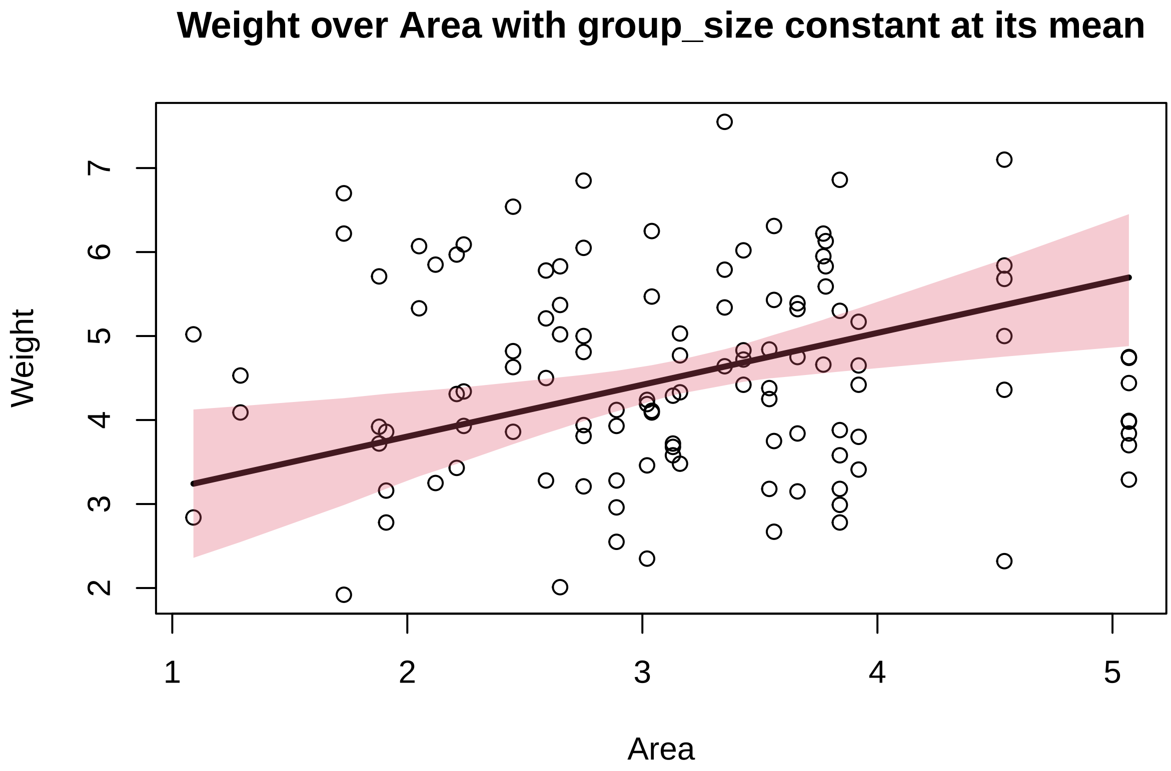
*Answer* - Neither predictor variable seems particularly important for predicting the weights of individual foxes as the slopes are relatively small indicating little effect of either predictor variable on fox weight. An increase of 1 unit in a territory's area has an effect of 0.02 (beta_a) on the wolves' weight within that territory - the 89 percentile interval does include both pos and negative values (-0.16-0.21) indicating that the direction of the effect isn't entirely stable. Whereas an increase of 1 unit in the groupsize of a social group has an effect of -0.12 on the wolves' weight.

**5H2.** Now fit a multiple linear regression with weight as the outcome and both area and groupsize as predictor variables. Plot the predictions of the model for each predictor, holding the other predictor constant at its mean. What does this model say about the importance of each variable? Why do you get different results than you got in the exercise just above?

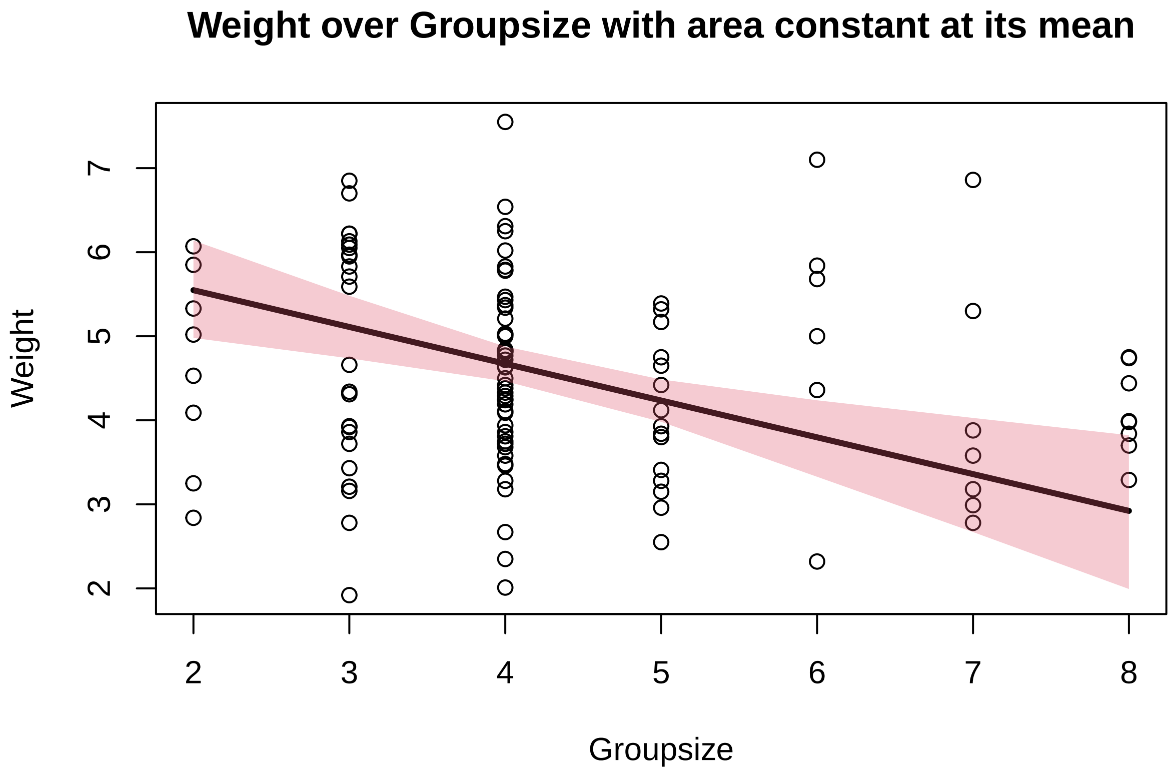
```r
set.seed(13)

## quap formula   (priors are not well-founded but somewhat arbitrary)
foxes5H2_quapFormula <- alist(
    weight ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_a*area + beta_g*groupsize,
    alpha ~ dnorm( 0 , 100 ) , # Prior
    beta_g ~ dnorm( 0 , 10 ) , # Prior
    beta_a ~ dnorm( 0 , 10 ) , # Prior
    sigma ~ dunif( 0 , 50 ) # Prior
    )

foxes5H2_model <- quap(foxes5H2_quapFormula, data=foxes_data)
precis(foxes5H2_model)
```

```
##               mean         sd       5.5%      94.5%
## alpha   4.4503637 0.37084800  3.8576769  5.0430504
## beta_g -0.4324060 0.12074246 -0.6253758 -0.2394362
## beta_a  0.6178386 0.20009157  0.2980536  0.9376235
## sigma   1.1184493 0.07342945  1.0010949  1.2358038
```

```r
## PLOTTING

# plot over Area
plot(foxes_data$area, foxes_data$weight,
     xlab = "Area", ylab = "Weight", main = "Weight over Area with group_size constant at its mean")


## sequence of areas used for plotting = all unique areas
xseq <- sort(unique(foxes_data$area))



## superimposing the MAP regression line
mu <- link( foxes5H2_model, data=data.frame(area=xseq, groupsize=mean(foxes_data$groupsize)) ) ## retri

lines( xseq, apply(mu, 2, mean), lwd=3 ) ## adding the MAP regression line,    i.e. the mean

## 95% interval showing uncertainty about the regression line,   i.e. the 89% interval for the mean
shade( apply(mu, 2, PI, prob=0.95), xseq, col=col.alpha(2, 0.3) )
```



```r
# - - -

# plot over Groupsize
plot(foxes_data$groupsize, foxes_data$weight,
     xlab = "Groupsize", ylab = "Weight", main = "Weight over Groupsize with area constant at its mean"


## sequence of groupsize used for plotting = all unique groupsize
xseq <- sort(unique(foxes_data$groupsize))


## superimposing the MAP regression line
mu <- link( foxes5H2_model, data=data.frame(groupsize=xseq, area=mean(foxes_data$area)) ) ## retrieving

lines( xseq, apply(mu, 2, mean), lwd=3 ) ## adding the MAP regression line,    i.e. the mean

## 95% interval showing uncertainty about the regression line,   i.e. the 89% interval for the mean
shade( apply(mu, 2, PI, prob=0.95), xseq, col=col.alpha(2, 0.3) )
```



*Answer* - This model that combines the predictors for groupsize and area increases the magnitude of the predictors' effect on wolf weight - for area the estimated beta-coefficient goes from 0.02 tp 0.62 and for groupsize it goes from -0.12-(-0.43). The changes in these estimated beta-coefficients suggest that the predictors were suppressed by each other, so including both in the model at the same time allows their correlation to increase the beta-coefficients (i.e. their effects on wolf weight).

**5H3.** Finally, consider the avgfood variable. Fit two more multiple regressions: (1) body weight as an additive function of avgfood and groupsize, and (2) body weight as an additive function of all three variables, avgfood and groupsize and area. Compare the results of these models to the previous models you've fit, in the first two exercises. (a) Is avgfood or area a better predictor of body weight? If you had to choose one or the other to include in a model, which would it be? Support your assessment with any tables or plots you choose. (b) When both avgfood or area are in the same model, their effects are reduced (closer to zero) and their standard errors are larger than when they are included in separate models. Can you explain this result?

```
### Regression Model 1

## quap formula   (priors are not well-founded but somewhat arbitrary)
foxes5H3_quapFormula1 <- alist(
    weight ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_g*groupsize + beta_f*avgfood,
    alpha ~ dnorm( 5 , 5 ) , # Prior
    beta_g ~ dnorm( 0 , 10 ) , # Prior
    beta_f ~ dnorm( 0 , 10 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )
```

```r
foxes5H3_model1 <- quap(foxes5H3_quapFormula1, data=foxes_data)
precis(foxes5H3_model1)
```

```
##              mean         sd       5.5%      94.5%
## alpha   4.1451891 0.42911103  3.4593868  4.8309914
## beta_g -0.5611279 0.15533389 -0.8093815 -0.3128744
## beta_f  3.7553749 1.20179781  1.8346699  5.6760799
## sigma   1.1163567 0.07326866  0.9992592  1.2334541
```

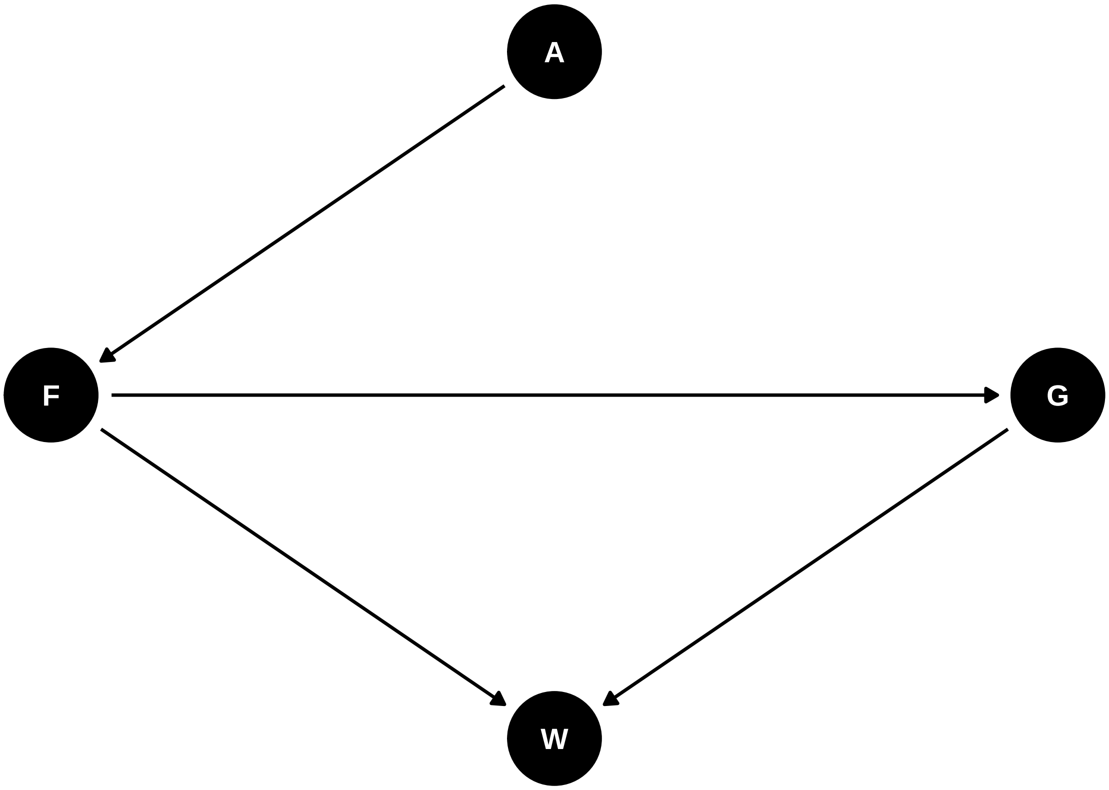
```r
# - - -


### Regression Model 2

## quap formula   (priors are not well-founded but somewhat arbitrary)
foxes5H3_quapFormula2 <- alist(
    weight ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_a*area + beta_g*groupsize + beta_f*avgfood,
    alpha ~ dnorm( 0 , 5 ) , # Prior
    beta_a ~ dnorm( 0 , 10 ) , # Prior
    beta_g ~ dnorm( 0 , 10 ) , # Prior
    beta_f ~ dnorm( 0 , 10 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


foxes5H3_model2 <- quap(foxes5H3_quapFormula2, data=foxes_data)
precis(foxes5H3_model2)
```

```
##              mean         sd        5.5%      94.5%
## alpha   4.0423827 0.42638248  3.36094114  4.7238242
## beta_a  0.3911406 0.23846778  0.01002302  0.7722582
## beta_g -0.6075320 0.15578391 -0.85650479 -0.3585592
## beta_f  2.5083745 1.43630930  0.21287482  4.8038742
## sigma   1.1043651 0.07250715  0.98848472  1.2202456
```

*Answer* - a) If assessing which is the better predictor of avgfood or area by comparing their effect on weight when included in models controlling for groupsize (G), then avgfood would be the better predictor with an estimated beta-coefficient of 3.76 (beta_f) whereas the coefficient for area is 0.62 (beta_a). - b) Including both avgfood or area in a model seems to reduce their effects and make their standard errors larger, which might be because the two predictors are correlated, so that including one of them adds little predictive power when already knowing the other. This correlation seems plausible too as bigger areas would probably also tend to have more food. To "solve" this correlation, one could use a combination of both predictors in one, such as 'food available per 100mˆ2' or something of that sort.

**Defining our theory with explicit DAGs** Assume this DAG as an causal explanation of fox weight:

```r
pacman::p_load(dagitty,
               ggdag)
dag <- dagitty('dag {
A[pos="1.000,0.500"]
F[pos="0.000,0.000"]
G[pos="2.000,0.000"]
W[pos="1.000,-0.500"]
A -> F
F -> G
F -> W
G -> W
}')

# Plot the DAG
ggdag(dag, layout = "circle")+
  theme_dag()
```



where A is area, F is avgfood,G is groupsize, and W is weight.

**Using what you know about DAGs from chapter 5 and 6, solve the following three questions:**

1) Estimate the total causal influence of A on F. What effect would increasing the area of a territory have on the amount of food inside of it?

```
model <- lm(avgfood ~ area, data = foxes_data)
summary(model)$coefficients
```

```
##              Estimate  Std. Error  t value     Pr(>|t|)
## (Intercept) 0.1538679 0.030988398  4.96534 2.426603e-06
## area        0.1886495 0.009387042 20.09680 2.870329e-39
## the above should give about the same as this
set.seed(13)

modelFormula <- alist(
    avgfood ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_A*area,
    alpha ~ dnorm( 0 , 5 ) , # Prior
    beta_A ~ dnorm( 0 , 1 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


model1 <- quap(modelFormula, data=foxes_data)
precis(model1)
```

```
##              mean          sd       5.5%     94.5%
## alpha  0.15391332 0.030717570 0.10482071 0.2030059
## beta_A 0.18863274 0.009304985 0.17376157 0.2035039
## sigma  0.09264127 0.006078795 0.08292619 0.1023564
```

*Answer* - The total causal influence of A on F is the same as the direct causal influence in this case, since there is no other (indirect) paths going from A and ending in F. The total effect of increasing a territory's area by 1 unit on the amount of food inside it is 0.19 (beta_A).

2) Infer the **total** causal effect of adding food F to a territory on the weight W of foxes. Can you calculate the causal effect by simulating an intervention on food?

```
modelFormula <- alist(
    weight ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_F*avgfood + beta_A*area,
    alpha ~ dnorm( 0 , 5 ) , # Prior
    beta_F ~ dnorm( 0 , 1 ) , # Prior
    beta_A ~ dnorm( 0 , 1 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


model2 <- quap(modelFormula, data=foxes_data)
precis(model2)
```

```
##              mean         sd       5.5%     94.5%
## alpha   4.4959954 0.40499348  3.8487376 5.1432532
## beta_F -0.4485498 0.75547678 -1.6559476 0.7588480
## beta_A  0.1164316 0.18298447 -0.1760129 0.4088761
## sigma   1.1756592 0.07728108  1.0521491 1.2991693
```

*Answer* - The total causal influence of F on W includes both the direct path and the indirect path from F -> G -> W. The latter path is kept open by not stratifying / conditioning on G. The model does include A as a predictor as it influences F but isn't part of an indirect path from F to W. The total effect of increasing a territory's food by 1 unit on the weight of the wolves inside the territory is -0.45 (beta_F) - although the 89 percentage interval includes both negative and positive values (-1.66-0.76) indicating that this coefficient is not stable in the direction of its effect.

3) Infer the **direct** causal effect of adding food F to a territory on the weight W of foxes. In light of your estimates from this problem and the previous one, what do you think is going on with these foxes?

```
modelFormula <- alist(
    weight ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_F*avgfood + beta_A*area + beta_G*groupsize,
    alpha ~ dnorm( 0 , 5 ) , # Prior
    beta_F ~ dnorm( 0 , 1 ) , # Prior
    beta_A ~ dnorm( 0 , 1 ) , # Prior
    beta_G ~ dnorm( 0 , 1 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


model3 <- quap(modelFormula, data=foxes_data)
precis(model3)
```

```
##               mean         sd       5.5%      94.5%
## alpha   4.3194655 0.38742755  3.7002814  4.9386495
## beta_F  0.8294164 0.82424014 -0.4878785  2.1467113
## beta_A  0.5178128 0.20779748  0.1857123  0.8499133
## beta_G -0.4732487 0.13124687 -0.6830066 -0.2634909
## sigma   1.1106679 0.07318368  0.9937062  1.2276295
```

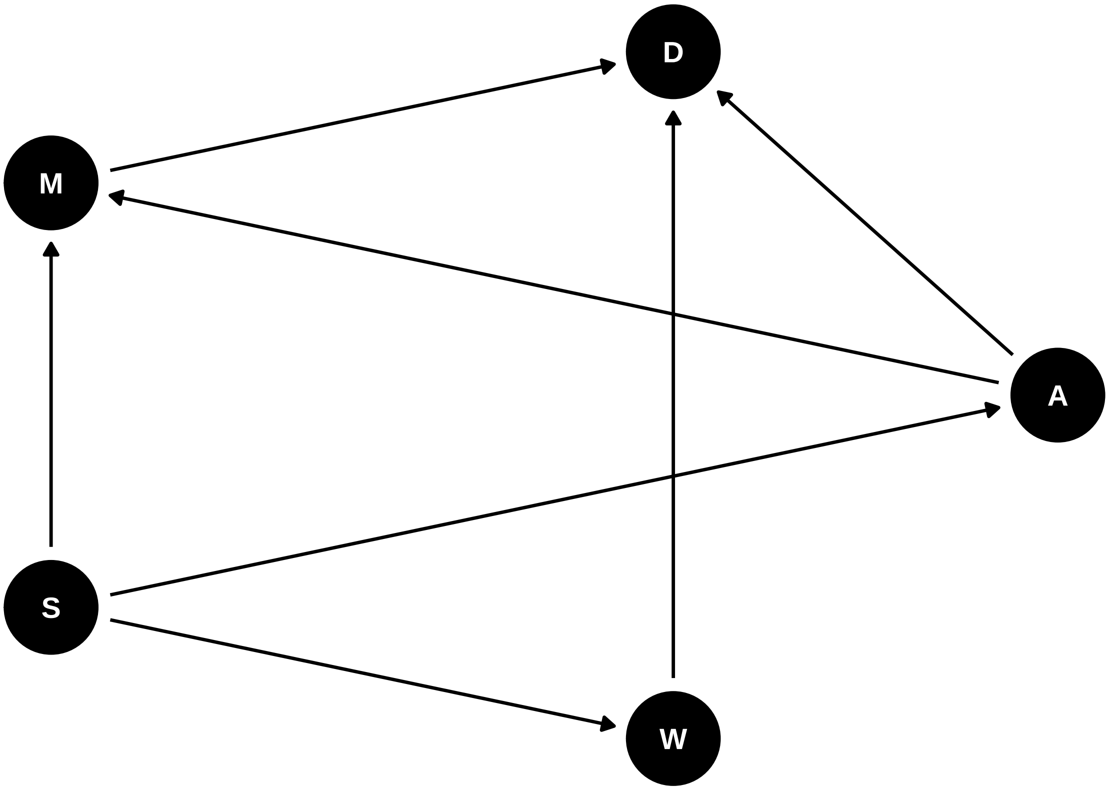
*Answer* - The direct causal influence of F on W should block the indirect path from F -> G -> W - by stratifying / conditioning on G. Additionally, the model again controls for A as by including it as a predictor, since it isn't part of the direct path. The direct effect of increasing a territory's food by 1 unit on the weight of the wolves inside the territory is 0.83 (beta_F).

- In terms of what might be going on with these foxes, it seems that increasing the average amount of food available in the territory (F) has two effects influencing the weights of the wolves in a territory (W) differently. First of all, this direct pathway from F to W shows that increasing amounts of food has a positive effect of the wolves' weight. However, as shown by the DAG, increasing the amounts of food also affects G (the number of foxes in the social group), which then affects W. If causally interpreting 2) 'the total causal effect of F on W', then the negative effect of increasing amounts of food on the wolves' weight likely comes from the increasing number of wolves resulting from increasing amounts of food, which then results in less food for the individual wolves.

## Chapter 6: Investigating the Waffles and Divorces

**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.

```
data(WaffleDivorce)
WD_data <- WaffleDivorce

## Causal Graph (DAG)
dag_fromBook <- dagitty( "dag {
 A-> D
 A-> M-> D
 A <- S-> M
 S-> W-> D
 }")

# Plot the DAG
ggdag(dag_fromBook, layout = "circle") +
  theme_dag()
```



```
## Model for the total causal influence of number of Waffle Houses on divorce rate
modelFormula6H1 <- alist(
    Divorce ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_W*WaffleHouses + beta_S*South,
    alpha ~ dnorm( 0 , 10 ) , # Prior
    beta_W ~ dnorm( 0 , 10 ) , # Prior
    beta_S ~ dnorm( 0 , 10 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


model6H1 <- quap(modelFormula6H1, data=WD_data)
precis(model6H1)
```

```
##                 mean          sd         5.5%       94.5%
## alpha   9.2894428914 0.282797708  8.837477535 9.741408248
## beta_W  0.0009721041 0.005045707 -0.007091911 0.009036119
## beta_S  1.2918369060 0.730823398  0.123839965 2.459833847
## sigma   1.6925178635 0.169455500  1.421695246 1.963340481
```

*Answer* - The total causal influence of W on D is the same as the direct causal influence in this case, since there is no other (indirect) paths going from W and ending in D. However, assessing this effect does require the blocking of the backdoor pathway from S to W, which is done by controlling for S in the model. The total effect of number of Waffle Houses increasing by 1 unit on divorce rate is 0.00 (beta_W).

**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.

```r
impliedConditionalIndependencies( dag_fromBook )
```

```
## A _||_ W | S
## D _||_ S | A, M, W
## M _||_ W | S
```

```r
# 1st implied conditional in-dependency = A & W when controlling for S
modelFormula6H2 <- alist(
    WaffleHouses ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_A*MedianAgeMarriage + beta_S*South,
    alpha ~ dnorm( 0 , 50 ) , # Prior
    beta_A ~ dnorm( 0 , 5 ) , # Prior
    beta_S ~ dnorm( 0 , 5 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


#model6H2 <- quap(modelFormula6H2, data=WD_data)
#precis(model6H2)



# 2nd implied conditional in-dependency = S & D when controlling for W, M A
modelFormula6H2 <- alist(
    South ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_D*Divorce + beta_W*WaffleHouses + beta_M*Marriage + beta_A*MedianAgeMarriage,
    alpha ~ dnorm( 0 , 50 ) , # Prior
    beta_A ~ dnorm( 0 , 0.5 ) , # Prior
    beta_W ~ dnorm( 0 , 0.5 ) , # Prior
    beta_D ~ dnorm( 0 , 0.5 ) , # Prior
    beta_M ~ dnorm( 0 , 0.5 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


model6H2 <- quap(modelFormula6H2, data=WD_data)
precis(model6H2)
```

```
##                mean           sd        5.5%       94.5%
## alpha   1.854712841 1.9580786877 -1.27467508 4.984100768
## beta_A -0.066417658 0.0598102936 -0.16200606 0.029170743
## beta_W  0.004401232 0.0007047784  0.00327486 0.005527604
## beta_D  0.028560793 0.0314575867 -0.02171451 0.078836093
## beta_M -0.013090444 0.0170450162 -0.04033167 0.014150784
## sigma   0.312999240 0.0312990056  0.26297738 0.363021096
```

```
# 3rd implied conditional in-dependency = M & W when controlling for S
modelFormula6H2 <- alist(
    Marriage ~ dnorm( mu , sigma ), # Predictive distribution
    mu <- alpha + beta_W*WaffleHouses + beta_S*South,
    alpha ~ dnorm( 0 , 50 ) , # Prior
    beta_W ~ dnorm( 0 , 0.5 ) , # Prior
    beta_S ~ dnorm( 0 , 0.5 ) , # Prior
    sigma ~ dunif( 0 , 10 ) # Prior
    )


model6H2 <- quap(modelFormula6H2, data=WD_data)
precis(model6H2)
```

```
##                mean          sd        5.5%      94.5%
## alpha  20.049558613 0.596104983 19.09686772 21.0022495
## beta_W  0.001173057 0.008462456 -0.01235158  0.0146977
## beta_S  0.086582464 0.478176467 -0.67763589  0.8508008
## sigma   3.756000577 0.375804830  3.15539187  4.3566093
```

*Answer* - I cannot seem to find priors for the first that allows me to fit the model. However, the second test implies that there isn't absolute independence between S & D when controlling for W, M A, since the estimated beta-coefficient for D isn't exactly 0, but instead is very slightly positive 0.03 (beta_D) - however the standard error is 0.03, so the "real" beta_D might very well be 0 and thus indicating absolute independence between S & D when controlling for W, M A. If we regard this test as failed, it would suggest that the DAG-implied conditional in-dependency is more complex in reality than in our causal DAG graph. In other words, it suggests that there is either missing variables with causal effects or there is a missing arrow (i.e. causal path) drawn between the S & D. The third test suggests that there is absolute independence between M & W when controlling for S, since the mean beta-coefficient for W is exactly 0 - additionally it has a small standard error of 0.01.