Assignment 2 - Methods 4

Alexander (Alex) Juby-Rasmussen & Vilma Bejer Kristiansen

2025-03-18

Second assignment

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

```
#Loading in packages
pacman::p_load(ggplot2, rethinking, tidyverse, dplyr)
```

Chapter 3: Causal Confussion - Alex

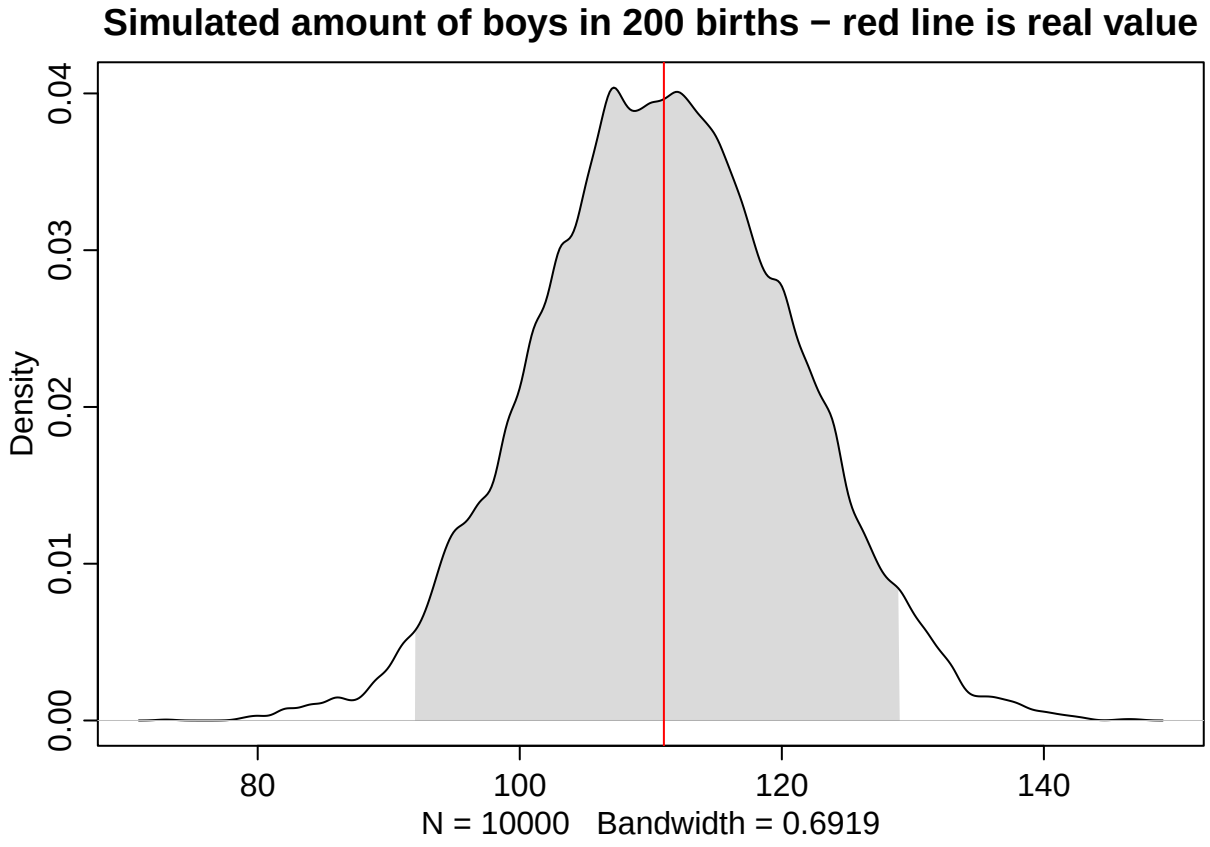
Reminder: We are tying to estimate the probability of giving birth to a boy I have pasted a working solution to questions 6.1-6.3 so you can continue from here:)

3H3 - Alex

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).

```
# 3H1
#Setting seed
set.seed(123)
# Find the posterior probability of giving birth to a boy:
pacman::p_load(rethinking)
data(homeworkch3)
set.seed(1)
W <- sum(birth1) + sum(birth2)
N <- length(birth1) + length(birth2)
p_grid <-seq(from =0, to = 1, len =1000)
prob_p <- rep(1,1000)
prob_data <- dbinom(W,N,prob=p_grid)
posterior <-prob_data * prob_p
posterior <- posterior / sum(posterior)

# 3H2
# Sample probabilities from posterior distribution:
samples <- sample (p_grid, prob = posterior, size =1e4, replace =TRUE)
```

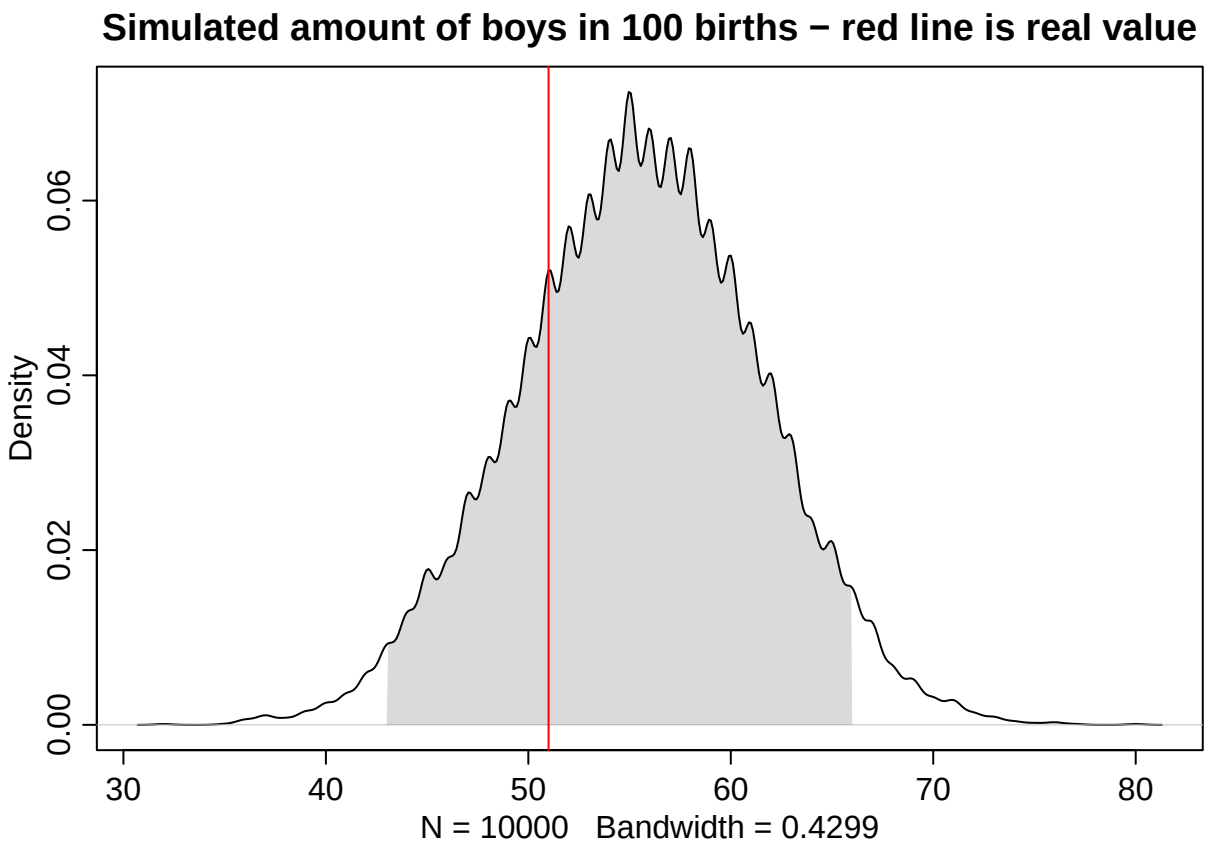
```r
# 3H3
# Simulate births using sampled probabilities as simulation input, and check if they allign with real v
simulated_births <- rbinom(n = 1e4, size = N, prob = samples)
rethinking::dens(simulated_births,show.HPDI = 0.95)
abline(v=W, col="red")
title("Simulated amount of boys in 200 births - red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



3H4.- Alex

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?

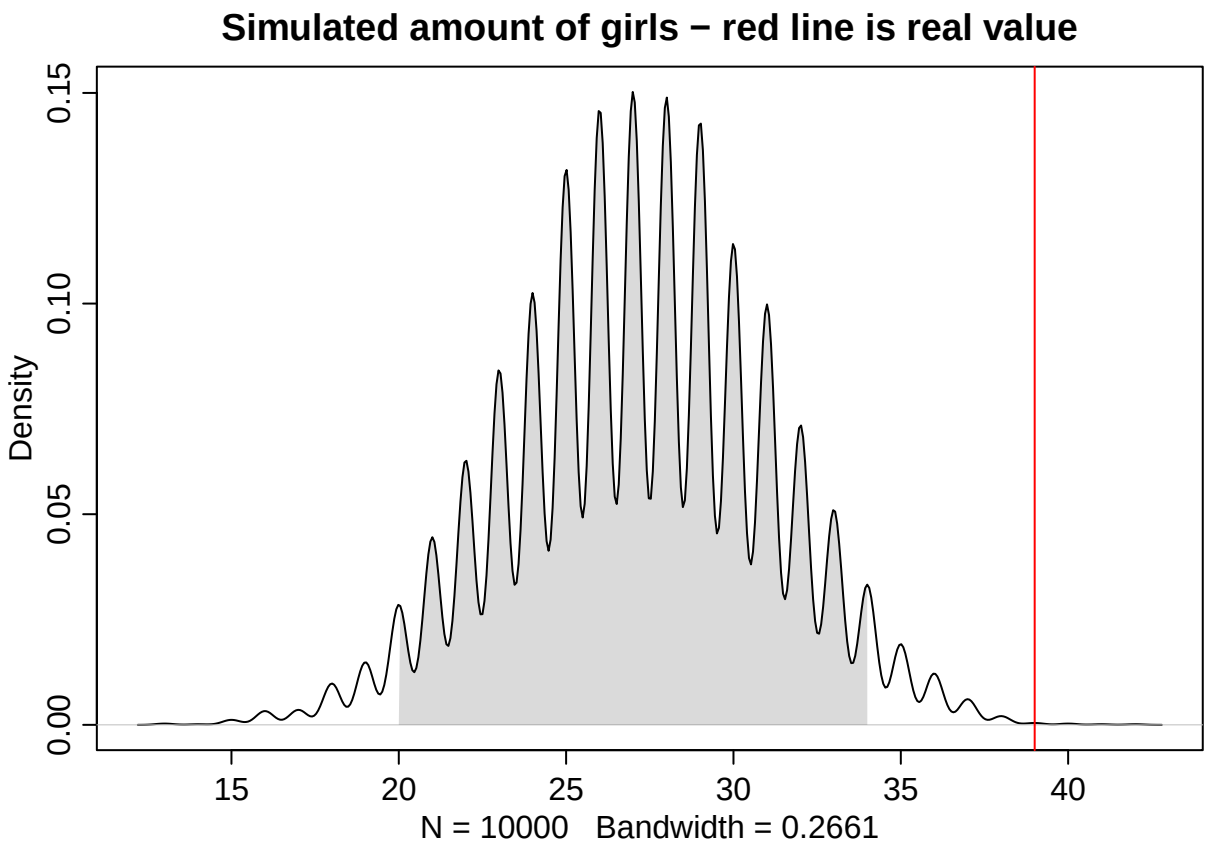
```r
#Firstly setting seed
set.seed(123)

set.seed(1)
W <- sum(birth1)
N <- length(birth1)

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 100 births - red line is real value")
```



3H5. - Alex

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 cound 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?

```
#Setting seed
set.seed(123)
W <- sum(birth2[birth1==0])
N <- length(birth2[birth1==0])

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 girls - red line is real value")
```



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. - Vilma

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).

Example = More sunscreen, more ice cream? outcome variable (Y): Number of ice creams sold per day

predictor 1 (X1): Amount of sunscreen sold per day

Predictor 2 (X2): Daily temperature

The setup is as follows:

- On hot days, people buy more sunscreen.
- On hot days, people also buy more ice cream.
- So, sunscreen and ice cream appear to be correlated.

Model 1: Bivariate (spurious) - ice_cream_sales ~ sunscreen_sales

Model 2: Multiple regression (adjusted) - ice_cream_sales ~ sunscreen_sales + temperature

```
#Setting seed
set.seed(123)

#Simulating temperature (X2)
temperature <- rnorm(100, mean = 25, sd = 5)  # e.g. degrees Celsius

#Sunscreen sales (X1) increase with temperature + noise
sunscreen_sales <- 0.5 * temperature + rnorm(100, sd = 2)

#Ice cream sales (Y) also increase with temperature + noise
ice_cream_sales <- 0.8 * temperature + rnorm(100, sd = 2)

# Putting it in a data frame
df <- data.frame(
  temperature,
  sunscreen_sales,
  ice_cream_sales
)
```

```
#Fitting models

#Model 1: Spurious bivariate model
model_spurious <- lm(ice_cream_sales ~ sunscreen_sales, data = df)
summary(model_spurious)
```

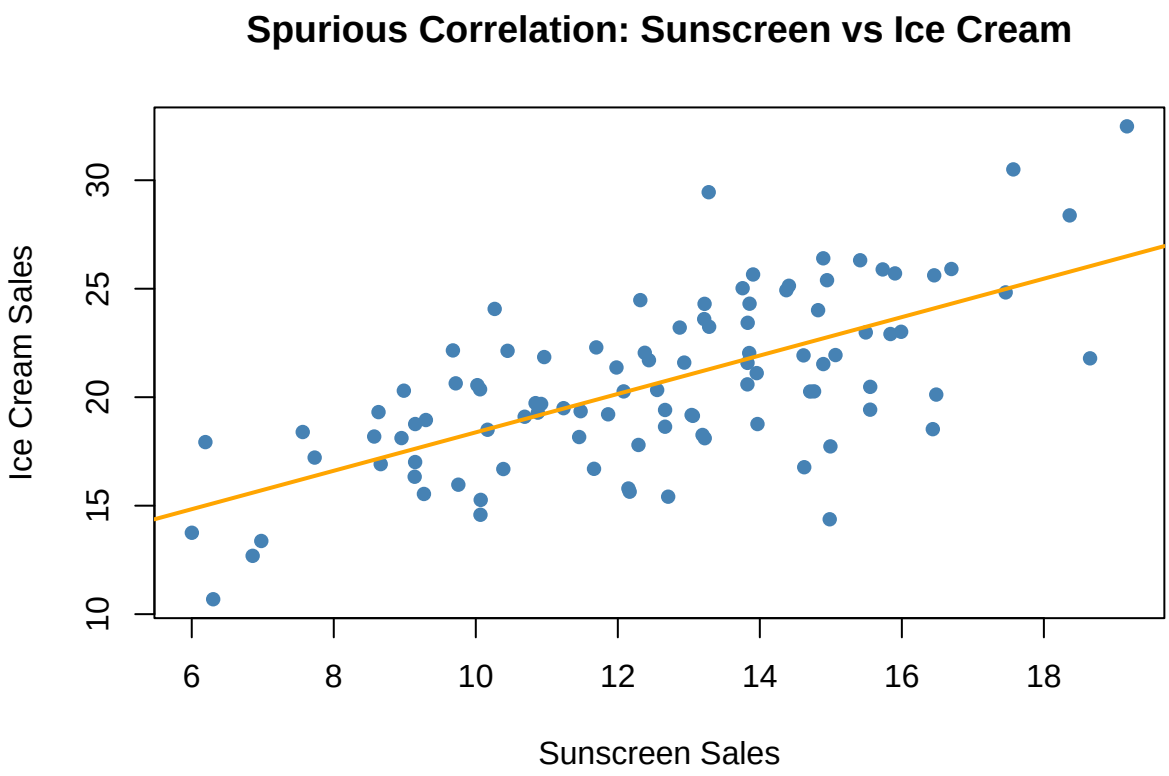
```
##
## Call:
## lm(formula = ice_cream_sales ~ sunscreen_sales, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4193 -2.1235  0.0316  2.2009  8.1641
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       9.5262     1.2949   7.357 5.81e-11 ***
## sunscreen_sales   0.8853     0.1008   8.781 5.31e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.926 on 98 degrees of freedom
## Multiple R-squared:  0.4404, Adjusted R-squared:  0.4347
## F-statistic: 77.11 on 1 and 98 DF,  p-value: 5.314e-14
```

```
#Model 2: Adjusted model
model_adjusted <- lm(ice_cream_sales ~ sunscreen_sales + temperature, data = df)
summary(model_adjusted)
```

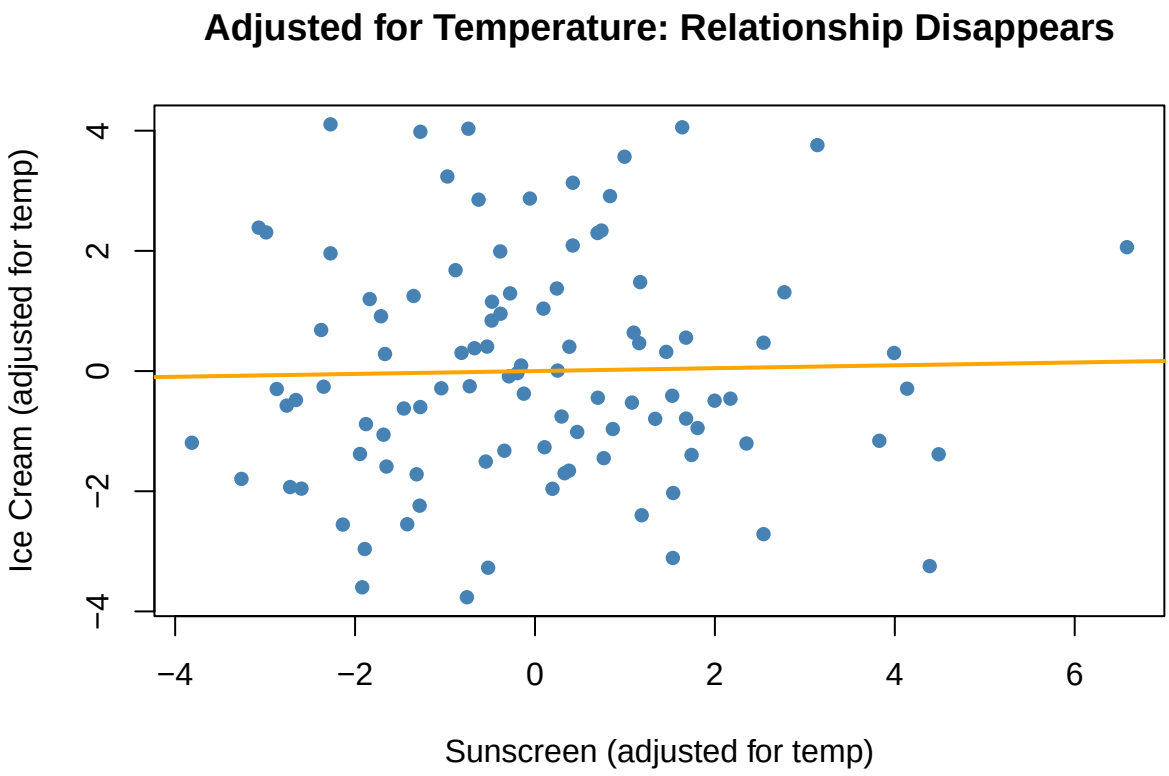
```
##
```

```
## Call:
## lm(formula = ice_cream_sales ~ sunscreen_sales + temperature,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7460 -1.3215 -0.2489  1.2427  4.1597
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.60185    1.08365   1.478    0.143
## sunscreen_sales  0.02381    0.09899   0.241    0.810
## temperature      0.73483    0.06328  11.613   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.903 on 97 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.761
## F-statistic: 158.7 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
#Plotting it, just for fUN!
plot(df$sunscreen_sales, df$ice_cream_sales,
     xlab = "Sunscreen Sales",
     ylab = "Ice Cream Sales",
     main = "Spurious Correlation: Sunscreen vs Ice Cream",
     col = "steelblue", pch = 16)
abline(model_spurious, col = "orange", lwd = 2)
```




```r
#Getting residuals of both sunscreen and ice cream after regressing on temperature
res_sunscreen <- resid(lm(sunscreen_sales ~ temperature, data = df))
res_icecream <- resid(lm(ice_cream_sales ~ temperature, data = df))

#plotting:
plot(res_sunscreen, res_icecream,
     xlab = "Sunscreen (adjusted for temp)",
     ylab = "Ice Cream (adjusted for temp)",
     main = "Adjusted for Temperature: Relationship Disappears",
     col = "steelblue", pch = 16)
abline(lm(res_icecream ~ res_sunscreen), col = "orange", lwd = 2)
```



Final thoughts:

- SO, as we see in both the comparisons of the models, and the plotting, the initial look of the spurious correlation looks very promising: Sunscreen sales look like a strong predictor of ice cream sales. The effect size is highly significant and and $R^2 = 44\%$. But as we know, correlation is not equal to causation. So when we look at model 2: ice_cream_sales ~ sunscreen_sales + temperature, we see that as soon as we start to control for temperature, the sunscreen becomes irrelevant. The effect size drops from $0.89 \rightarrow 0.02$, and p-value greatly increases to 0.81. This means that the temperature is the real causal driver between these variables, while the spurious correlation is simply just a proxy.

5M2. - Vilma

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.

Example: Coffee and sleep quality A masked relationship happens when an outcome variable is influenced by two predictors, but in opposite directions e.g. one is positive the other negative.The predictors are correlated with each other, and in a bivariate regression, their effects cancel out, which makes the outcome look unrelated to either, if or until you adjust for them.

Outcome variable (Y): Daytime energy levels

Predictor 1 (X1): hours of sleep

- More sleep → more energy

Predictor 2 (X2): Coffee Consumption

- More coffee → more energy
- Coffee is negatively correlated with sleep (people who sleep less drink more coffee)

Let's simulate it:

```r
set.seed(123)

#Simulating sleep hours (X1)
sleep <- rnorm(100, mean = 7, sd = 1.5)

#Coffee consumption (X2), negatively correlated with sleep
coffee <- -0.6 * (sleep - mean(sleep)) + rnorm(100, sd = 1)

#Daytime energy (Y): positively affected by both sleep and coffee
energy <- 0.6 * sleep + 0.5 * coffee + rnorm(100, sd = 1)

#Creating a dataframe
df_masked <- data.frame(
  sleep,
  coffee,
  energy
)

#Fitting the models
#Model A: Bivariate regression - energy ~ sleep
model_sleep <- lm(energy ~ sleep, data = df_masked)
summary(model_sleep)
```

```
##
## Call:
## lm(formula = energy ~ sleep, data = df_masked)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.27943 -0.73809 -0.00409  0.58150  2.67586
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.97163    0.57289   5.187 1.15e-06 ***
## sleep        0.19290    0.07886   2.446   0.0162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 98 degrees of freedom
## Multiple R-squared:  0.05754,    Adjusted R-squared:  0.04792
## F-statistic: 5.983 on 1 and 98 DF,  p-value: 0.01623
```

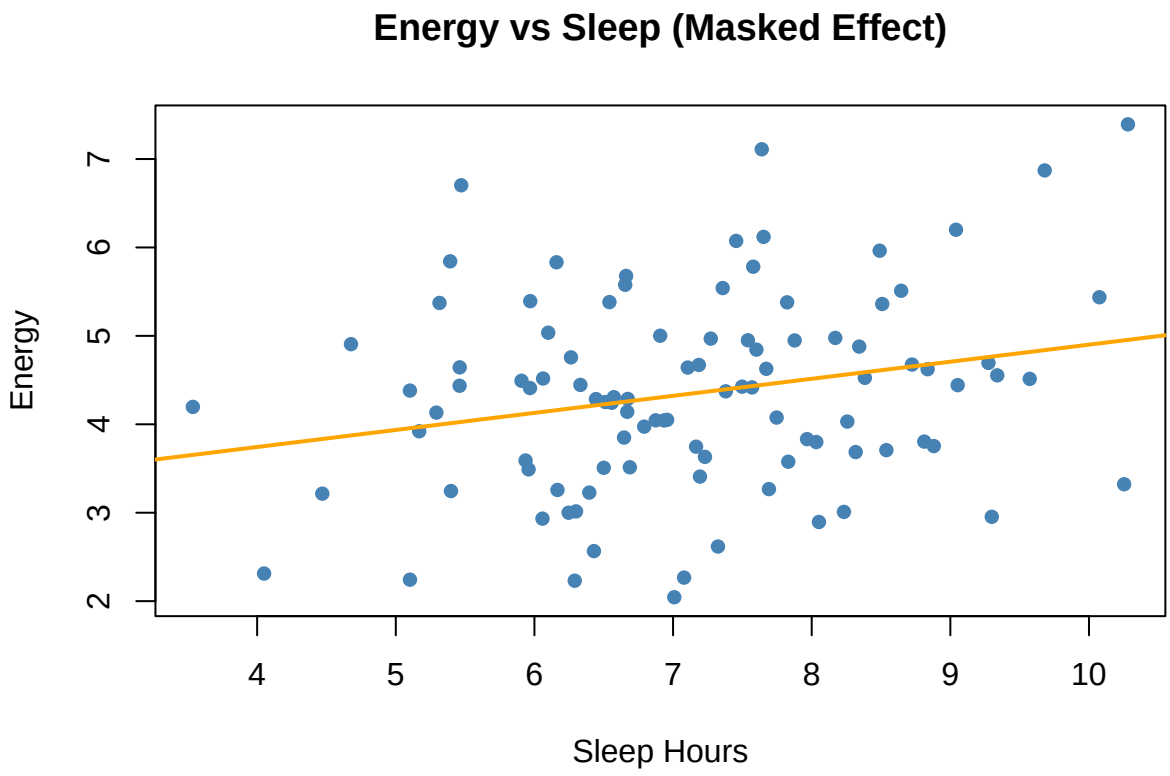
```r
#Model B: Bivariate regression - energy ~ coffee
model_coffee <- lm(energy ~ coffee, data = df_masked)
summary(model_coffee)
```

```
##
## Call:
## lm(formula = energy ~ coffee, data = df_masked)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2379 -0.6602 -0.0897  0.5547  3.2249
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.36455    0.10922  39.962   <2e-16 ***
## coffee       0.15335    0.08418   1.822   0.0715 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.088 on 98 degrees of freedom
## Multiple R-squared:  0.03276,    Adjusted R-squared:  0.02289
## F-statistic: 3.319 on 1 and 98 DF,  p-value: 0.07154
```

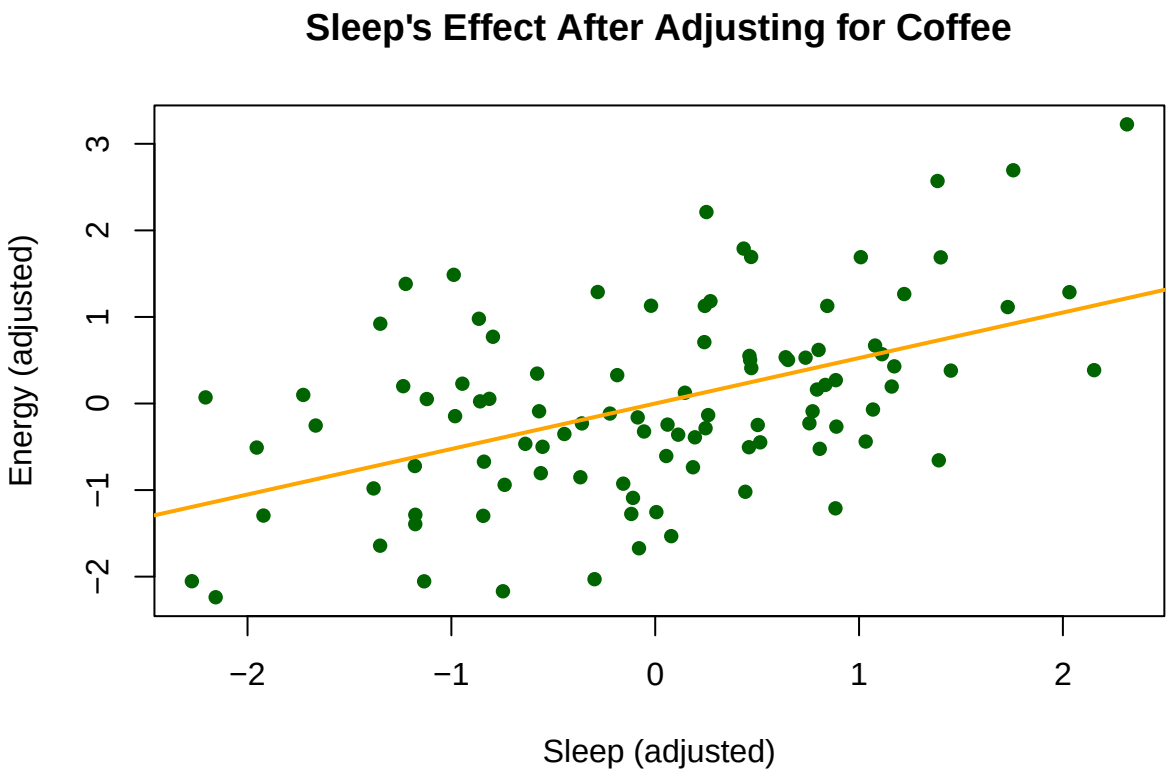
```r
#Model C: Multiple regression - energy ~ sleep + coffee
model_both <- lm(energy ~ sleep + coffee, data = df_masked)
summary(model_both)
```

```
##
## Call:
## lm(formula = energy ~ sleep + coffee, data = df_masked)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8730 -0.6607 -0.1245  0.6214  2.0798
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.65459    0.67012   0.977    0.331
## sleep        0.52551    0.09395   5.593 2.06e-07 ***
## coffee       0.52381    0.09899   5.291 7.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9513 on 97 degrees of freedom
## Multiple R-squared:  0.2686, Adjusted R-squared:  0.2536
## F-statistic: 17.81 on 2 and 97 DF,  p-value: 2.573e-07
```

```r
#Plotting it:
plot(df_masked$sleep, df_masked$energy,
     xlab = "Sleep Hours", ylab = "Energy",
     main = "Energy vs Sleep (Masked Effect)", col = "steelblue", pch = 16)
abline(model_sleep, col = "orange", lwd = 2)
```




```r
#Residual plot to visualize unmasking
# Residuals of energy ~ coffee
res_energy <- resid(lm(energy ~ coffee, data = df_masked))
res_sleep <- resid(lm(sleep ~ coffee, data = df_masked))

plot(res_sleep, res_energy,
     xlab = "Sleep (adjusted)", ylab = "Energy (adjusted)",
     main = "Sleep's Effect After Adjusting for Coffee",
     col = "darkgreen", pch = 16)
abline(lm(res_energy ~ res_sleep), col = "orange", lwd = 2)
```



Final thoughts - Explanation

- If we start looking at model 1 energy ~ sleep, sleep appears to help energy a little $R^2 = 5.8\%$, which is surprising, giving we know exactly how important sleep is in our daily life to function. If we then look at model 2 energy ~ coffee, the effect looks even weaker with an estimate of $R^2 = 3.3\%$, which is also confusing, given the many studies and research papers being made and have been made about coffee increasing (given caffeine) energy. But, when looking at model 3: energy ~ sleep + coffee, we suddenly see a very different result when including both predictors. The model suddenly just from the 3-5% to 26.9%, which is a major improvement.

This shows the masked relationship caused by collinearity and the compensating predictors. People who sleep less tend to drink more coffee, and when you look at either variable in isolation its effect is masked by the missing context. However, when you include both in the model the effect unmakes and they contribute is clear.

5M3. - Vilma

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 5m3

- A higher divorce rate increases the pool of single individuals who are eligible to remarry. In some cases, divorces may even occur with the intent of entering a new marriage. To explore this relationship with multiple regression, we could include variables such as the total number of marriages or an indicator for remarriage. If the initial link between divorce and marriage rates is driven by remarriage, the coefficient for marriage rate should shrink toward zero once this factor is controlled for.

5M5. - Vilma

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 5m5

- Let's say we have access to data on obesity rates (O), gas prices (G), driving frequency or distance (D), levels of physical activity (E), and frequency of dining out (R). There are a couple of ways gas prices might be tied to obesity, and we can test both using multiple regression (but without actually coding and simulating it).

First idea: when gas gets expensive, people might drive less. If they're not driving as much, maybe they walk or bike more instead—which bumps up exercise and helps lower obesity. So the steps would be:

Gas price (G) goes up → Driving (D) goes down

Driving goes down → Exercise (E) goes up

Exercise goes up → Obesity (O) goes down

Second idea: again, gas gets pricey, people cut back on driving—but this time the result is fewer restaurant trips. That means fewer giant portions and fast food, which could also lead to less obesity. So in this case:

Gas price (G) goes up → Driving (D) goes down

Driving goes down → Eating out (R) goes down

Eating out goes down → Obesity (O) goes down

So basically, by running regressions on these different links—like G on D, D on E or R, and then E or R on O—we can start to untangle which paths are actually driving the relationship between gas prices and obesity.

Chapter 5: Foxes and Pack Sizes - Vilma

All five exercises below use the same data, data(foxes) (part of rethinking).84 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. - Vilma

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?

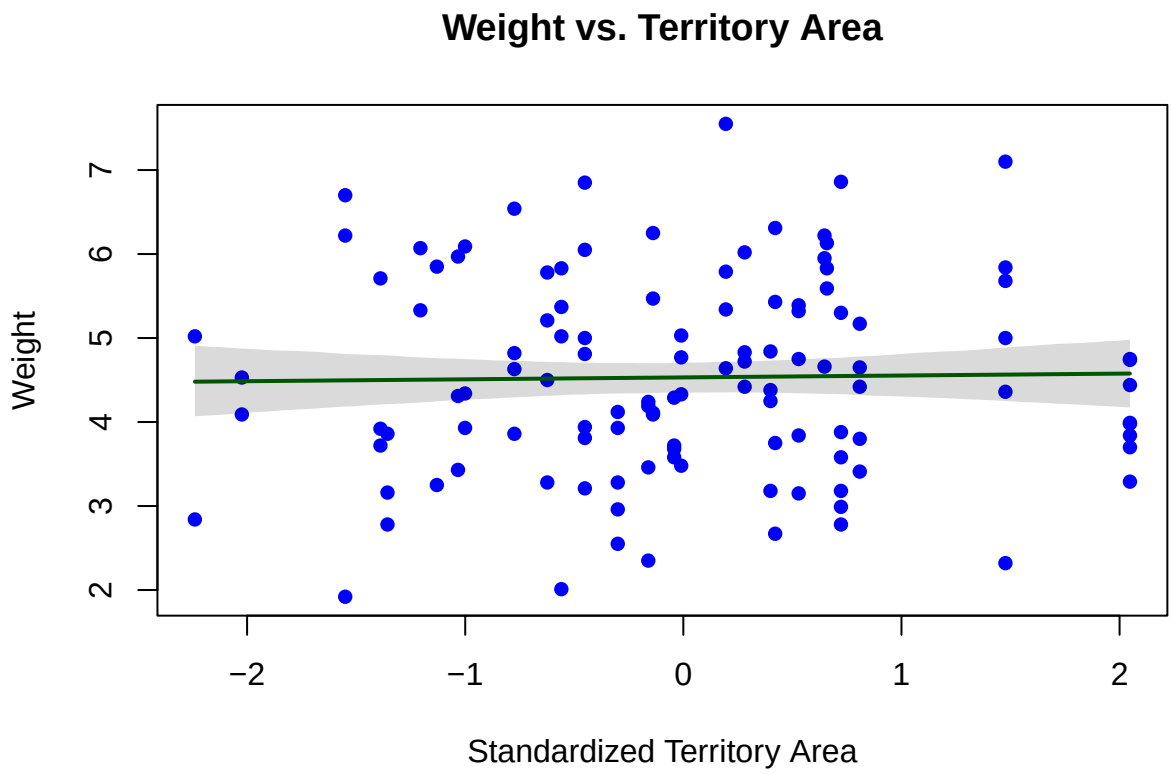
```
#loading in the dataset
data(foxes)

# Standardize predictors for better interpretation
foxes$area_s <- standardize(foxes$area)
foxes$groupsize_s <- standardize(foxes$groupsize)

# Model 1: weight ~ area
m_area <- quap( #Using the quap function
  alist( #Definigning variables under
    weight ~ dnorm(mu, sigma),
    mu <- a + b * area_s,
    a ~ dnorm(0, 10),
    b ~ dnorm(0, 1),
    sigma ~ dunif(0, 10)
  ),
  data = foxes
)

precis(m_area)
```

```
##             mean         sd       5.5%     94.5%
## a     4.52910560 0.10943760  4.3542032 4.7040080
## b     0.02278514 0.10926088 -0.1518348 0.1974051
## sigma 1.17874941 0.07739485  1.0550575 1.3024413
```

```
# Model 2: weight ~ Group_Size
m_group <- quap(#Using the quap function
  alist( #Defining variables under
    weight ~ dnorm(mu, sigma),
    mu <- a + b * groupsize_s,
    a ~ dnorm(0, 10),
    b ~ dnorm(0, 1),
    sigma ~ dunif(0, 10)
  ),
  data = foxes
)

precis(m_group)
```

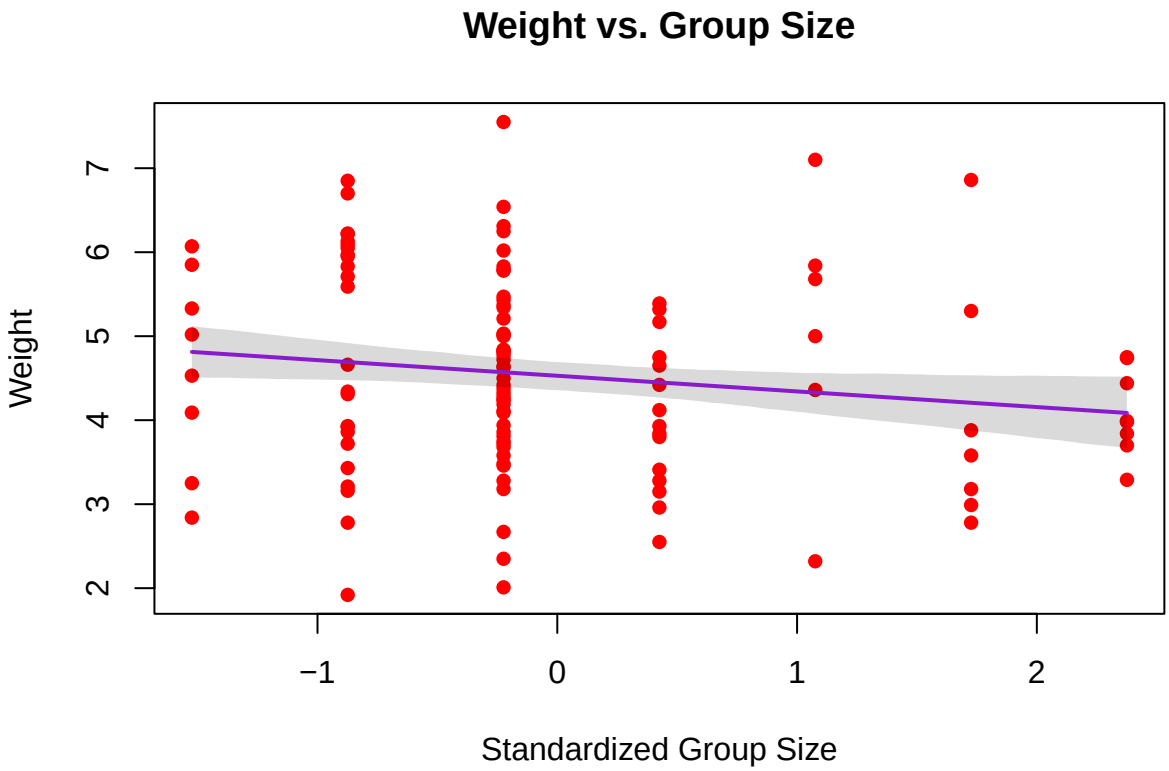
```
##             mean         sd       5.5%       94.5%
## a      4.5290759 0.10802423  4.3564323  4.70171950
## b     -0.1883721 0.10786653 -0.3607637 -0.01598054
## sigma  1.1635242 0.07638838  1.0414408  1.28560754
```

```r
#Plotting weight ~ Area
# Predict over a sequence of area values
area_seq <- seq(from = min(foxes$area_s), to = max(foxes$area_s), length.out = 100)
mu_area <- link(m_area, data = data.frame(area_s = area_seq))
mu_mean_area <- apply(mu_area, 2, mean)
mu_PI_area <- apply(mu_area, 2, PI)

# Plot
plot(foxes$area_s, foxes$weight, col = "blue", pch = 16,
     xlab = "Standardized Territory Area", ylab = "Weight",
     main = "Weight vs. Territory Area")
lines(area_seq, mu_mean_area, col = "darkgreen", lwd = 2)
shade(mu_PI_area, area_seq)
```




```r
#Plotting Weight ~ Group Size
# Predict over a sequence of group sizes
group_seq <- seq(from = min(foxes$groupsize_s), to = max(foxes$groupsize_s), length.out = 100)
mu_group <- link(m_group, data = data.frame(groupsize_s = group_seq))
mu_mean_group <- apply(mu_group, 2, mean)
mu_PI_group <- apply(mu_group, 2, PI)

# Plot
plot(foxes$groupsize_s, foxes$weight, col = "red", pch = 16,
     xlab = "Standardized Group Size", ylab = "Weight",
     main = "Weight vs. Group Size")
```

```
lines(group_seq, mu_mean_group, col = "purple", lwd = 2)
shade(mu_PI_group, group_seq)
```



Intepreting the results

- Model 1 weight ~ Area: The slope b = 0.02, and the 95% interval is [-0.15, 0.20]. As the slope is very close to zero, and the 95% interval includes zero, suggest that the territory size is *not* a strong or reliable predictor of a fox's body weight in this particular model.
- Model 2 Weight ~ Group Size: Here the slope, b = -0.19 and the 95% interval is [-0.36, -0.02]. As the slope is negative and the 92% interval does no include zero, suggests that a larger group size is somewhat associated with lower body weight. This might make sense, in terms of group dynamic. More foxes in the group means less for for all the foxes and mroe competition for the food there is, resulting in an average of lower body weight.

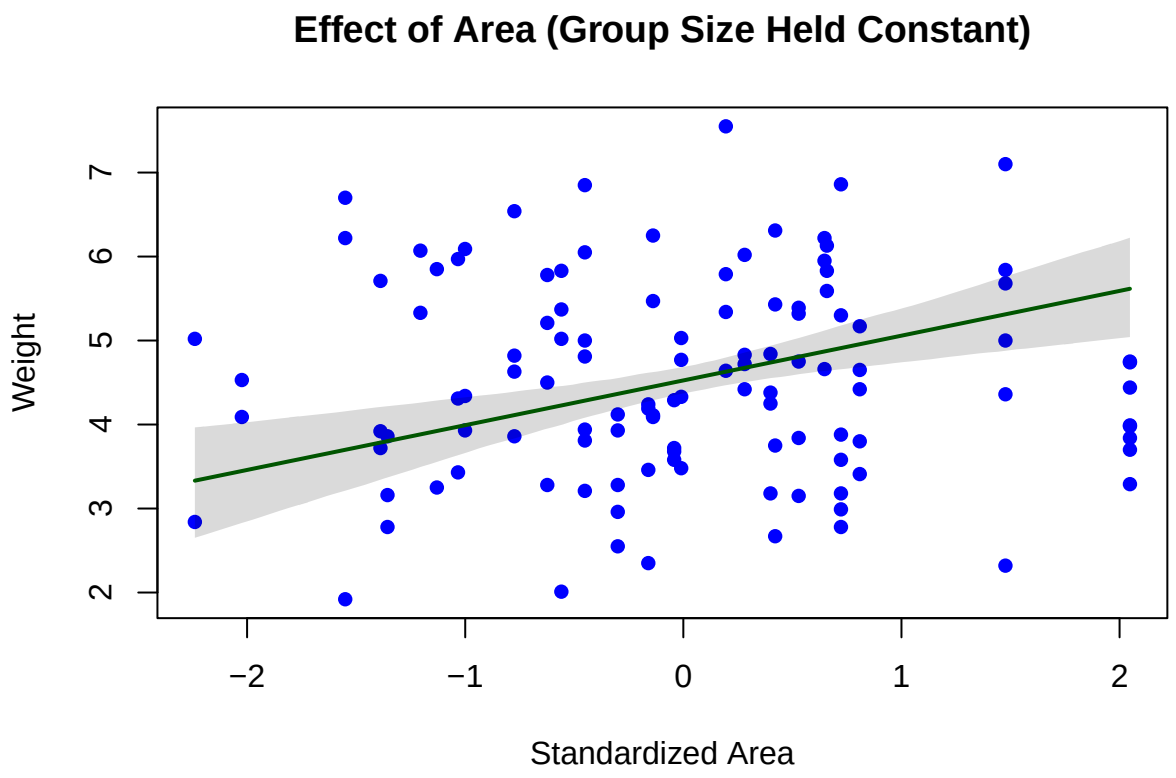
5H2.- Vilma

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?

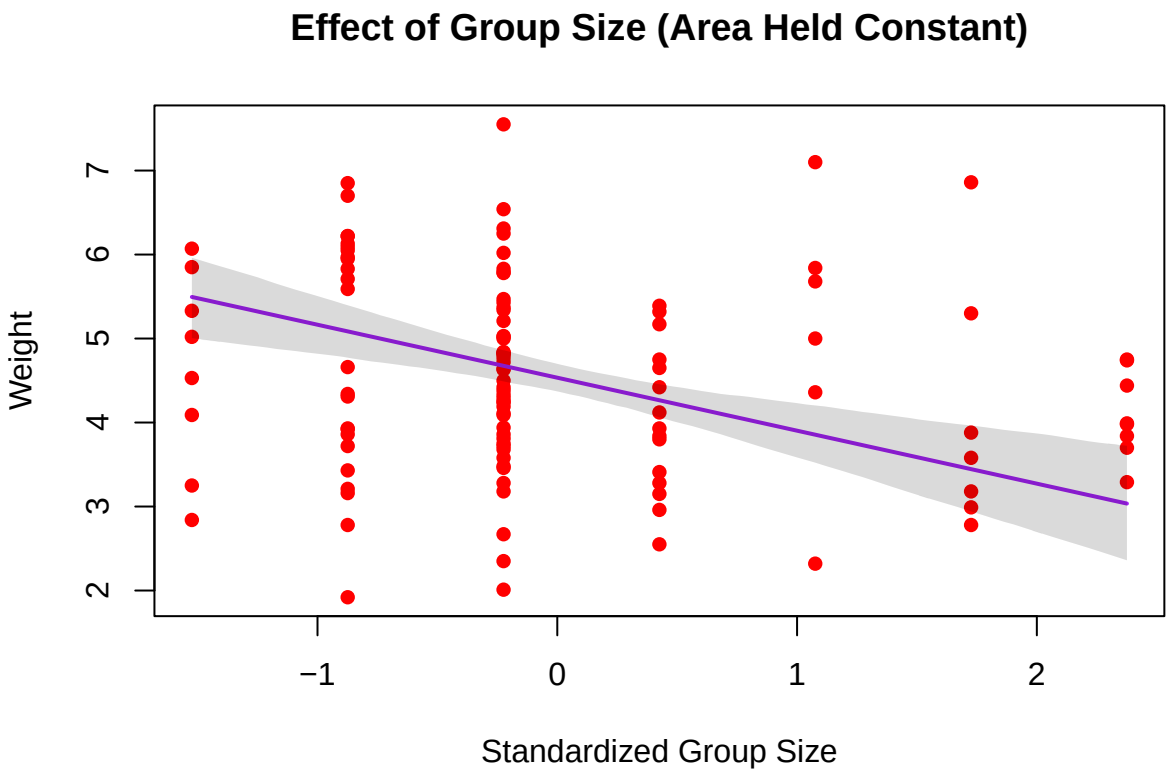
```r
#Firstly fitting the model
m_multi <- quap(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + bA * area_s + bG * groupsize_s,
    a ~ dnorm(0, 10),
    bA ~ dnorm(0, 1),
    bG ~ dnorm(0, 1),
    sigma ~ dunif(0, 10)
  ),
  data = foxes
)

precis(m_multi)
```

```
##             mean         sd       5.5%      94.5%
## a      4.5291553 0.10386498  4.3631590  4.6951516
## bA     0.5375723 0.18074234  0.2487112  0.8264335
## bG    -0.6287090 0.18074412 -0.9175731 -0.3398450
## sigma  1.1187201 0.07347904  1.0012864  1.2361538
```

```r
#Secondly, plotting predictions, whiel holding other predictors constant

area_seq <- seq(from = min(foxes$area_s), to = max(foxes$area_s), length.out = 100)
mu_area <- link(m_multi, data = data.frame(
  area_s = area_seq,
  groupsize_s = mean(foxes$groupsize_s)  # hold constant
))
mu_mean_area <- apply(mu_area, 2, mean)
mu_PI_area <- apply(mu_area, 2, PI)

plot(foxes$area_s, foxes$weight, col = "blue", pch = 16,
     xlab = "Standardized Area", ylab = "Weight",
     main = "Effect of Area (Group Size Held Constant)")
lines(area_seq, mu_mean_area, col = "darkgreen", lwd = 2)
shade(mu_PI_area, area_seq)
```




```r
#Thirdly, plotting Vary Group Size, hold Area at its mean
group_seq <- seq(from = min(foxes$groupsize_s), to = max(foxes$groupsize_s), length.out = 100)
mu_group <- link(m_multi, data = data.frame(
  groupsize_s = group_seq,
  area_s = mean(foxes$area_s)  # hold constant
))
mu_mean_group <- apply(mu_group, 2, mean)
mu_PI_group <- apply(mu_group, 2, PI)

plot(foxes$groupsize_s, foxes$weight, col = "red", pch = 16,
     xlab = "Standardized Group Size", ylab = "Weight",
     main = "Effect of Group Size (Area Held Constant)")
lines(group_seq, mu_mean_group, col = "purple", lwd = 2)
shade(mu_PI_group, group_seq)
```



Interpreting the result - Why different answers/plots?

- So as we see in the dataset precis(m_mult), the territory size (bA) has a positive and statistically meaningful effect on the body weight of the fox. THe 95% interval [0.25, 0.83] does not include zero, meaning we are relatively confident that this effect is real. More territory –> heavier foxes, likely due to more resources available for them.
- The group size (bG) haws a negative effect on the foxes body weight. The interval is [-0.92, -0.34] and stays below zero. Meaning that more foxes in a group –> lighter indivdual foxes, probably due to competion (as mentioned earlier).
- The information is different, as when we did the bivariate regressions, each predictor was explaining the shared variance. The territory size and group size are correlated, meaning that a larger area can hold more foxes. The shared variance was therefore confoundng the true relationsship. SO when we here isolate each effect by controlling for the other (in the multiple regression), the picture suddenly becomes: More area is realyl helpful, but only when you account for the number of foxes sharing it. **and** more foxes really hurt, but only when you account for how big the space is.

5H3. - Vilma

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?

```
#First I standardize the avgFood for a fairer comparison
foxes$avgfood_s <- standardize(foxes$avgfood)

#Now fitting model A weight ~ avgFood + Group Size
m_food_group <- quap(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + bF * avgfood_s + bG * groupsize_s,
    a ~ dnorm(0, 10),
    bF ~ dnorm(0, 1),
    bG ~ dnorm(0, 1),
    sigma ~ dunif(0, 10)
  ),
  data = foxes
)


#Fittning model B weight ~ avgFood + groupSize + Area
m_food_group_area <- quap(
  alist(
    weight ~ dnorm(mu, sigma),
    mu <- a + bF * avgfood_s + bG * groupsize_s + bA * area_s,
    a ~ dnorm(0, 10),
    bF ~ dnorm(0, 1),
    bG ~ dnorm(0, 1),
    bA ~ dnorm(0, 1),
    sigma ~ dunif(0, 10)
  ),
  data = foxes
)


#Comparing the co-Efficients
precis(m_food_group)
```

```
##             mean         sd       5.5%      94.5%
## a      4.5291682 0.10372170  4.3634009  4.6949355
## bF     0.6783206 0.22922573  0.3119736  1.0446676
## bG    -0.7935026 0.22922933 -1.1598553 -0.4271498
## sigma  1.1171767 0.07341417  0.9998467  1.2345068
```

```
precis(m_food_group_area)
```

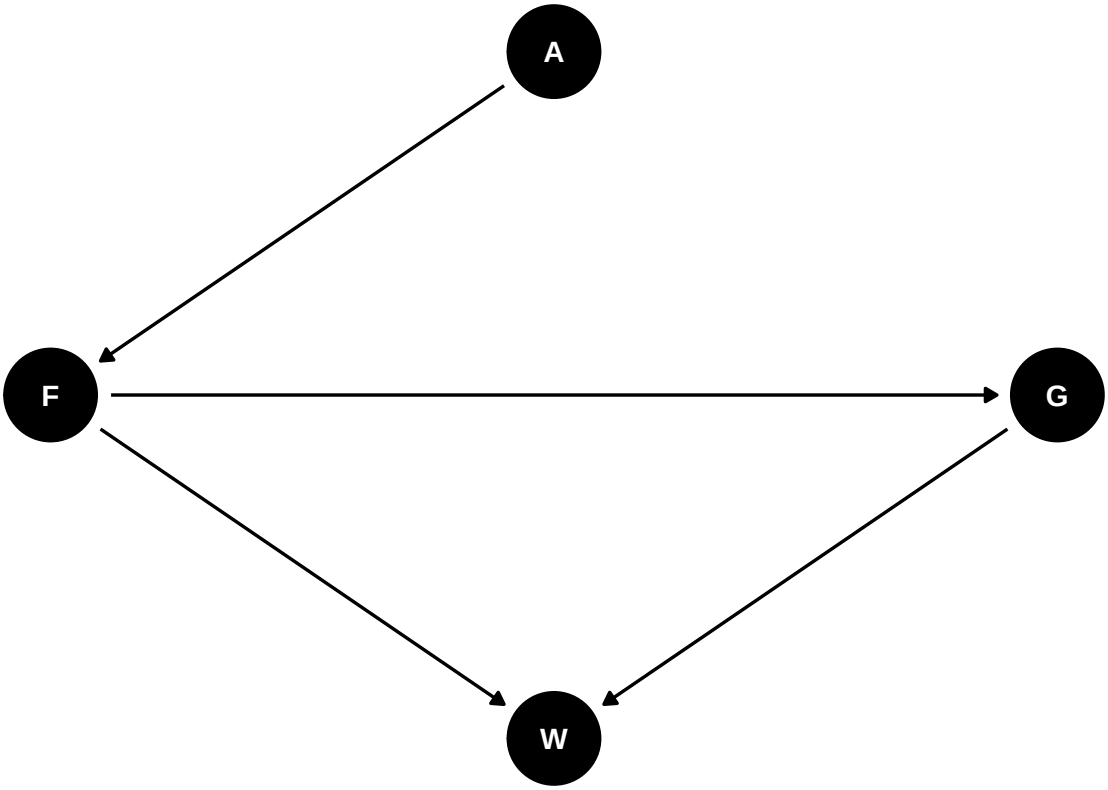
```
##             mean         sd          5.5%      94.5%
## a      4.5291786 0.10257178  4.3652490813  4.6931081
## bF     0.4351941 0.27212653  0.0002833787  0.8701049
## bG    -0.8624469 0.23062653 -1.2310326027 -0.4938611
## bA     0.3487942 0.21424326  0.0063920820  0.6911963
## sigma  1.1047898 0.07258567  0.9887839085  1.2207957
```

Intepreting the results and answering reflection questions

- Which is a better predictor? Avg Food or Area? So, when looking at the models separate both variableds have a strong, clear effect on the body weight. The avgFood has a slope of 0.68, and an interval of [0.31, 1.04] strongly excludes 0. While the area has a slope of 0.54, and an interval of [0.25, 0.83]. In the full model, with both, the avgFood drops to 0.44, and the interval just includes 0, resulting in weaker support. And the area drops to 0.35, while the interval still holds above zero, meaning it is still credible. I will therefore conclude that **area** is the more robust across the models, and even when the avgFood is in the model, it still holds up. So if I had to pick only one predictor, it would be Area.
- Why do the effects weaken when both are in the model? We see clear sign of multicollinearity, so when avgFood and area are modeled separately, their effects are stronger and more precise. When modeled togther, both coefficients becomes closer to zero, and the standard deviations increases in the bF SD from 0.23 –> 0.27 and for bA SD it keeps at 0.21. This happens because the two variables are correlated. They both try and explain the same part of variation in the body weight. However the model isn't that good at dividing the "credit", so it instead minimizes both coefficients and inflates the uncertainty.

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

```
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: - Vilma

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? - Vilma

```
#Standardizing the variables (again, just to be sure)
foxes$area_s <- standardize(foxes$area)
foxes$avgfood_s <- standardize(foxes$avgfood)

#Modelling area (A)/ AvgFood (F)
m_af <- quap(
  alist(
    avgfood_s ~ dnorm(mu, sigma),
    mu <- a + bA * area_s,
    a ~ dnorm(0, 1),
    bA ~ dnorm(0, 1),
    sigma ~ dunif(0, 1)
  ),
  data = foxes
)

precis(m_af)
```

```
##                mean         sd        5.5%      94.5%
## a     1.909455e-06 0.04333434 -0.06925474 0.06925856
## bA    8.814287e-01 0.04352255  0.81187126 0.95098613
## sigma 4.671640e-01 0.03067157  0.41814493 0.51618310
```

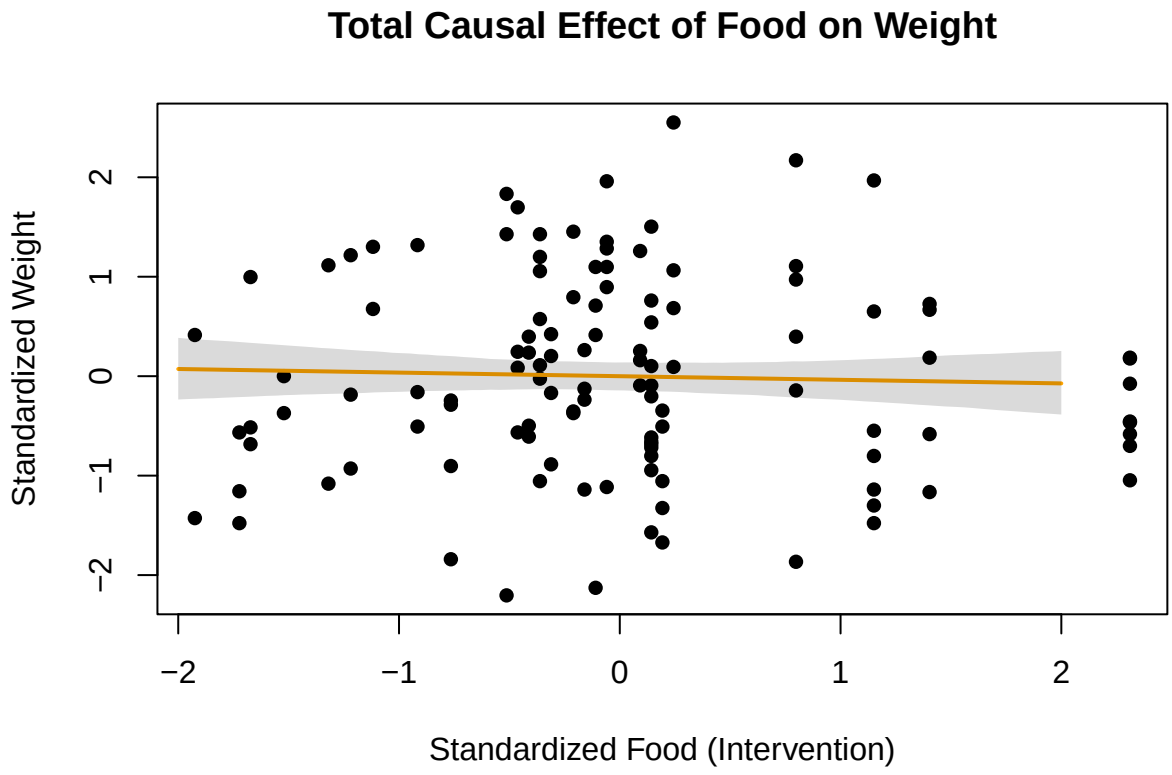
Intepreting results

- So as seen in the DAG model above, we clearly see a direct arrow between A –> F, meaning that the causal influence of A on F is a *direct effect*, as there is no other path from A to F. We therefore simulate it (to further support this claim), that increasing the area A, will directly increase food F. As we see in the precis(m_af), the etimated effect of area on food is strong, as we see the bA (effect of area on food) is significantly positive. The 95% interval is tight and well above 0, statistically convincing, and looking at the SD (in standardized terms), ever 1 SD increases in territory area leads to a 0.88 SD increase in available food.

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? - Vilma

```r
# Standardize variables (again)
foxes$avgfood_s <- standardize(foxes$avgfood)
foxes$groupsize_s <- standardize(foxes$groupsize)
foxes$weight_s <- standardize(foxes$weight)

#Firstly, modeling group size ~ Food
m_g <- quap(
  alist(
    groupsize_s ~ dnorm(mu, sigma),
    mu <- a + bF * avgfood_s,
    a ~ dnorm(0, 1),
    bF ~ dnorm(0, 1),
    sigma ~ dunif(0, 1)
  ),
  data = foxes
)

#Secondly, modelling weight ~ Food + Group Size
m_w <- quap(
  alist(
    weight_s ~ dnorm(mu, sigma),
    mu <- a + bF * avgfood_s + bG * groupsize_s,
    a ~ dnorm(0, 1),
    bF ~ dnorm(0, 1),
    bG ~ dnorm(0, 1),
    sigma ~ dunif(0, 1)
  ),
  data = foxes
)

#Now, simulating an intervention on Food

#Sequence of food values to simulate
food_seq <- seq(from = -2, to = 2, length.out = 100)
```

```r
#Predicting group size for each food level
mu_g <- link(m_g, data = data.frame(avgfood_s = food_seq))
group_pred <- apply(mu_g, 2, mean)

#Predicting weight using both food and predicted group size
mu_w <- link(m_w, data = data.frame(
  avgfood_s = food_seq,
  groupsize_s = group_pred
))

# Getting means and 95% intervals
mu_w_mean <- apply(mu_w, 2, mean)
mu_w_PI <- apply(mu_w, 2, PI)

#Finally, plotting the total effect of Food on Weight
plot(foxes$avgfood_s, foxes$weight_s, col = "black", pch = 16,
     xlab = "Standardized Food (Intervention)", ylab = "Standardized Weight",
     main = "Total Causal Effect of Food on Weight")

lines(food_seq, mu_w_mean, col = "orange", lwd = 2)
shade(mu_w_PI, food_seq)
```



Intepreting the results

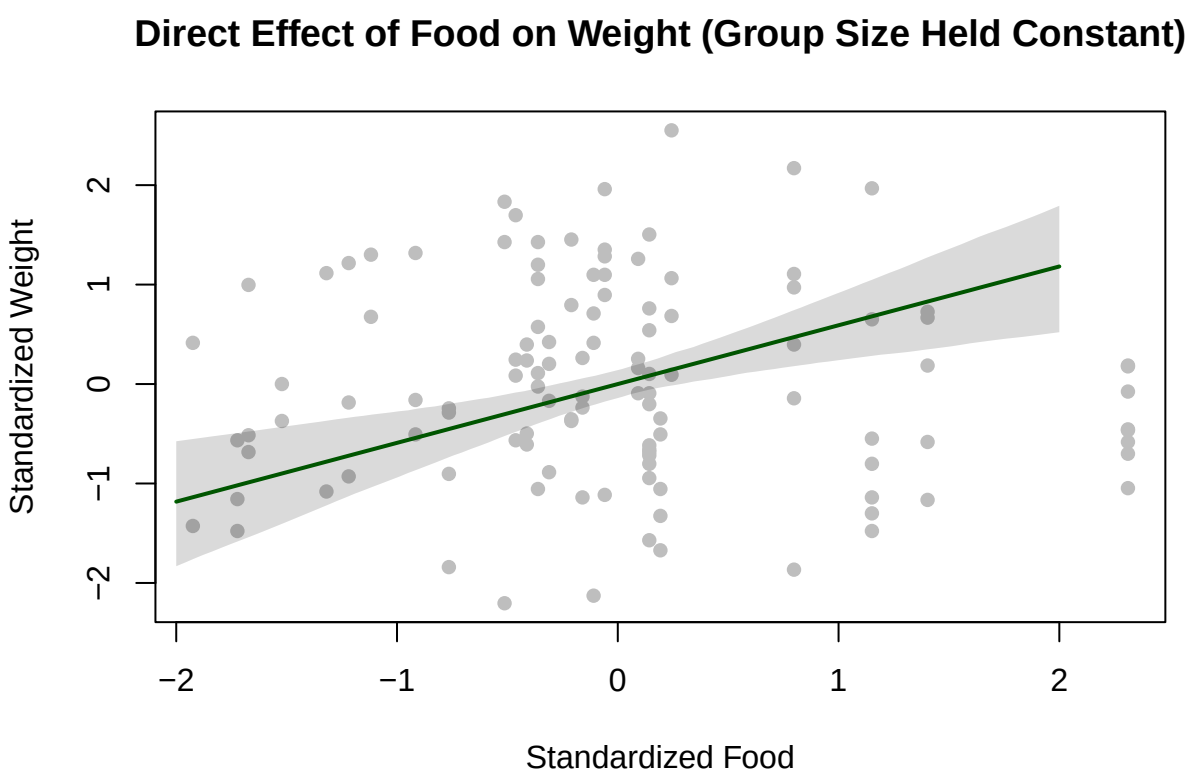
- Our orange regression line is nearly flat, and actually sloping slightly downward, which means that:

The total causal effect of food on weight is close to zero. So when we here simulate increasing food levels, the expected change in fox weight is very small. Our 95% interval, which includes 0, suggests that there is no strong causal signal. As we learned from the question before, the direct effect food has on weight is strongly positive, meaning the foxes gain more weight, while here, the indirect effect instead means more food –> more foxes –> more competition –> less weight. The two effects are therefore canceling each other out, which explains why the total effect is almost zero, despite the fact that both paths are active.

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? - Vilma

- F → W (direct path)
- F → G → W (indirect path)

```
# Simulating direct effect of food on weight
food_seq <- seq(from = -2, to = 2, length.out = 100)

# Predicting weight, holding group size constant
mu_direct <- link(m_w, data = data.frame(
  avgfood_s = food_seq,
  groupsize_s = 0  # fixed
))

mu_direct_mean <- apply(mu_direct, 2, mean)
mu_direct_PI <- apply(mu_direct, 2, PI)

#Plotting the direct effect of food on weight
plot(foxes$avgfood_s, foxes$weight_s, col = "gray", pch = 16,
     xlab = "Standardized Food", ylab = "Standardized Weight",
     main = "Direct Effect of Food on Weight (Group Size Held Constant)")

lines(food_seq, mu_direct_mean, col = "darkgreen", lwd = 2)
shade(mu_direct_PI, food_seq)
```



Intepreting the result – Explaing the plot: The direct causal effect of food on weight is positive: Adding food does help the foxes gaining weight, but only when we hold group size constant and stop it from dragging the benefit down.

The direct causal effect of food on weight is positive: if we increase food without changing group size, the foxes gain weight. However, in the real world, more food draws more foxes to the area, which increases competition — negating the benefit. So while food seems like it should help, in real settings its total impact is minimal because of the indirect crowding effect.

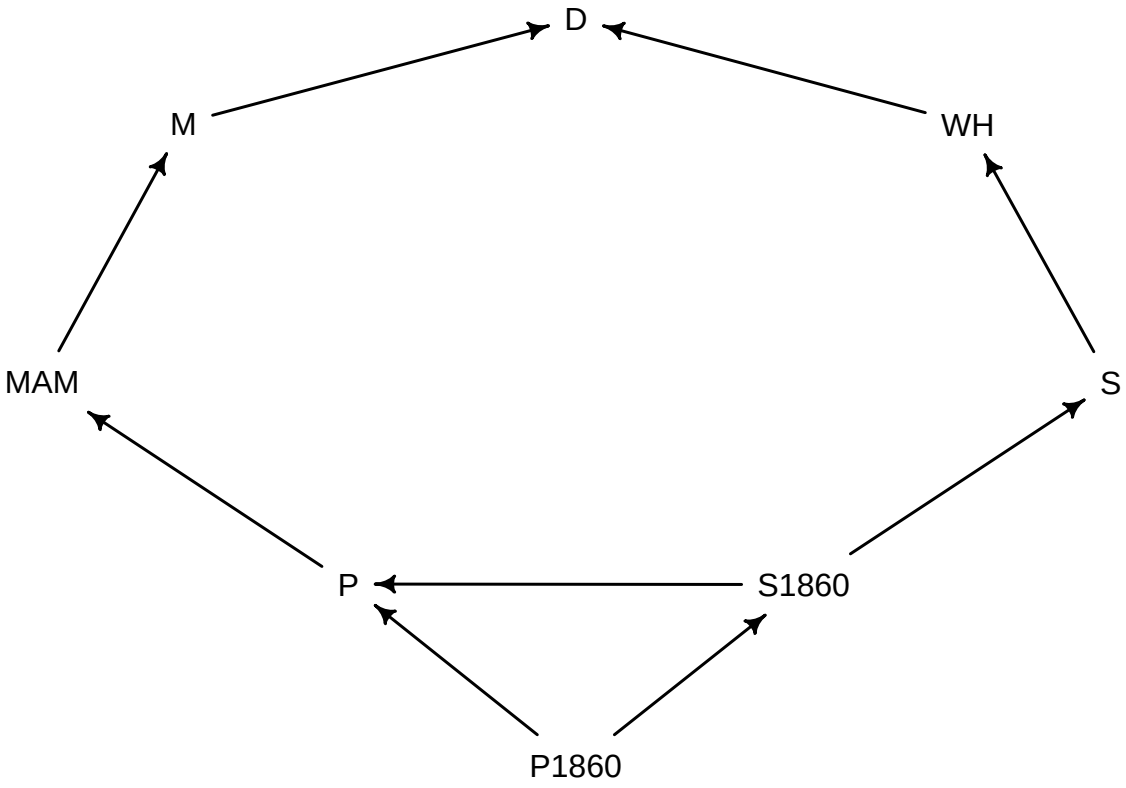
Chapter 6: Investigating the Waffles and Divorces - Alex

6H1. - Alex

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.

```
# DAG for WaffleDivorce data

library(dagitty)
WaffleDivorce_dag <- dagitty( "dag {
P1860 -> S1860 ->  S -> WH
P1860 -> P
S1860 -> P
P -> MAM -> M -> D
WH -> D
```

```
}")
drawdag( WaffleDivorce_dag )
```




```
adjustmentSets( WaffleDivorce_dag , exposure="WH" , outcome="D" )
```

```
## { M }
## { MAM }
## { P }
## { S1860 }
## { S }
```

```
impliedConditionalIndependencies( WaffleDivorce_dag )
```

```
## D _||_ MAM | M, P
## D _||_ MAM | M, S186
## D _||_ MAM | M, S
## D _||_ MAM | M, WH
## D _||_ P | MAM, S186
## D _||_ P | MAM, S
## D _||_ P | MAM, WH
## D _||_ P | M, S186
## D _||_ P | M, S
## D _||_ P | M, WH
## D _||_ P186 | P, S186
```

```
## D _||_ P186 | MAM, S186
## D _||_ P186 | P, S
## D _||_ P186 | MAM, S
## D _||_ P186 | P, WH
## D _||_ P186 | MAM, WH
## D _||_ P186 | M, S186
## D _||_ P186 | M, S
## D _||_ P186 | M, WH
## D _||_ S | S186, WH
## D _||_ S | P, WH
## D _||_ S | MAM, WH
## D _||_ S | M, WH
## D _||_ S186 | P, S
## D _||_ S186 | MAM, S
## D _||_ S186 | P, WH
## D _||_ S186 | MAM, WH
## D _||_ S186 | M, S
## D _||_ S186 | M, WH
## M _||_ P | MAM
## M _||_ P186 | P
## M _||_ P186 | MAM
## M _||_ S | S186
## M _||_ S | P
## M _||_ S | MAM
## M _||_ S186 | P
## M _||_ S186 | MAM
## M _||_ WH | S
## M _||_ WH | S186
## M _||_ WH | P
## M _||_ WH | MAM
## MAM _||_ P186 | P
## MAM _||_ S | S186
## MAM _||_ S | P
## MAM _||_ S186 | P
## MAM _||_ WH | S
## MAM _||_ WH | S186
## MAM _||_ WH | P
## P _||_ S | S186
## P _||_ WH | S
## P _||_ WH | S186
## P186 _||_ S | S186
## P186 _||_ WH | S
## P186 _||_ WH | S186
## S186 _||_ WH | S
```

Answering 6H1

- M = Marriage Rates
- MAM = Median Age of Marriage
- P = Population
- D = Divorce Rates
- WH = Waffle Houses
- S = South

- S1860 = Number of Slaves in 1860
- P1860 = Population in 1860

6H2. - Alex

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 int he data.

```
data(WaffleDivorce)

# Model 1 modelling Waffle Houses against Divorce Rates

model_6.1 <- quap(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <-  a + b1*WaffleHouses,
    a ~ dnorm(9,2),
    b1 ~ dnorm(0,0.5),
    sigma ~ dexp(1)),
  data = WaffleDivorce
  )


# Model 2 including South States

model_6.2 <- quap(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <-  a + b1*WaffleHouses + b2*South,
    a ~ dnorm(9,2),
    b1 ~ dnorm(0,0.5),
    b2 ~ dnorm(2,1),
    sigma ~ dexp(1)),
  data = WaffleDivorce
  )


# Model 3 Including Slaves1860

WaffleDivorce$Slaves1860_scaled <- WaffleDivorce$Slaves1860 / 100000

model_6.3 <- quap(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <-  a + b1*WaffleHouses + b2*South + b3*Slaves1860_scaled,
    a ~ dnorm(9,2),
    b1 ~ dnorm(0,0.5),
    b2 ~ dnorm(2,1),
    b3 ~ dnorm(0, 1),
    sigma ~ dexp(1)),
  data = WaffleDivorce
  )
```

```r
# Model 4 Including Population

log_mean <- log(mean(WaffleDivorce$Population))
log_sd <- sd(log(WaffleDivorce$Population))

model_6.4 <- quap(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <-  a + b1*WaffleHouses + b2*South + b3*Slaves1860_scaled + b4*Population,
    a ~ dnorm(9,2),
    b1 ~ dnorm(0,0.5),
    b2 ~ dnorm(2,1),
    b3 ~ dnorm(0, 1),
    b4 ~ dlnorm(log_mean, log_sd),
    sigma ~ dexp(1)),
  data = WaffleDivorce
  )

# Model 5 Including Median Age of Marriage

model_6.5 <- quap(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <-  a + b1*WaffleHouses + b2*South + b3*Slaves1860_scaled + b4*Population + b5*MedianAgeMarriage
    a ~ dnorm(9,2),
    b1 ~ dnorm(0,0.5),
    b2 ~ dnorm(2,1),
    b3 ~ dnorm(0, 1),
    b4 ~ dlnorm(log_mean, log_sd),
    b5 ~ dnorm(26, 1),
    sigma ~ dexp(1)),
  data = WaffleDivorce
  )

# Model 6 Including Marriage Rates

model_6.6 <- quap(
  alist(
    Divorce ~ dnorm(mu, sigma),
    mu <-  a + b1*WaffleHouses + b2*South + b3*Slaves1860_scaled + b4*Population + b5*MedianAgeMarriage
    a ~ dnorm(9,2),
    b1 ~ dnorm(0,0.5),
    b2 ~ dnorm(2,1),
    b3 ~ dnorm(0, 1),
    b4 ~ dlnorm(log_mean, log_sd),
    b5 ~ dnorm(26, 2),
    b6 ~ dnorm(20, 6),
    sigma ~ dexp(1)),
  data = WaffleDivorce
  )
```

```
compare(model_6.1, model_6.2, model_6.3, model_6.4, model_6.5, model_6.6)
```

```
##               WAIC        SE     dWAIC      dSE    pWAIC      weight
## model_6.2 202.2529  9.615206  0.000000       NA 3.492024 0.489454767
## model_6.1 203.7738  9.537513  1.520863 3.883417 2.822135 0.228802765
## model_6.6 204.7842 12.944549  2.531337 9.265000 7.234638 0.138051032
## model_6.3 204.8413  9.778821  2.588415 1.283285 4.773860 0.134166877
## model_6.4 210.8552  9.470279  8.602319 3.018056 4.835536 0.006633499
## model_6.5 212.5162  9.524158 10.263337 2.892182 4.917924 0.002891061
```

```
precis(model_6.2)
```

```
##                mean          sd         5.5%      94.5%
## a      9.2603325432 0.270802390  8.827538021 9.69312707
## b1    -0.0002063233 0.004554383 -0.007485106 0.00707246
## b2     1.5383179618 0.585346064  0.602821898 2.47381403
## sigma  1.6655388163 0.162777268  1.405389303 1.92568833
```

Answering 6H3 The best-fitting model is Model 6.2, which suggests that WaffleHouses have a strong influence on Divorce Rates. But when looking at the distribution of the variance of the data we can see that regional effects (South) are also important and that WaffleHouses have if not no effect, next to none. Adding other variables like Slaves1860_scaled, Population, MedianAgeMarriage and Marriage Rates does not seem to substantially improve the model fit.

The implied conditional independencies from the DAG are mostly supported by the model comparison results. South is a crucial factor to account for, and further variables like Slaves1860_scaled, Population, and Marriage Rates do not drastically alter the relationship between WaffleHouses and Divorce. The graph does not need significant revision but could be refined for future analysis by the addition of more arrows between variables but also including alternate variables such as Religion, Happiness, Welfare, Health and Income.