Activity 11: Statistical reasoning 3: multiple regression and DAGs

Tyler Beaman and Jacob Chung

Welcome! This is the third statistical reasoning activity. The goals of this activity are to understand how to implement DAGs in the context of multiple regression. Specifically, you will:

1. Build and interpret the relationships in DAGs
2. Use prior predictive simulation to adjust priors
3. Apply your undertanding of DAG structure to a multiple regression problem

---

You will submit one output for this activity:

1. A **PDF** of a rendered Quarto document with all of your R code. Please create a new Quarto document (e.g. don't use this `README.qmd`) and include all of the code that appears in this document, your own code, and **answers to all of the questions** in the "Q#" sections. Submit this PDF through Gradescope.

A reminder: **Please label the code** in your final submission in two ways:

1. denote your answers to each question using headers that correspond to the question you're answering, and
2. thoroughly "comment" your code: remember, this means annotating your code directly by typing descriptions of what each line does after a `#`. This will help future you!

Let's start by reading in the relevant packages

```
# Run this to install some data packages
#install.packages(c("coda","mvtnorm","devtools","loo","dagitty","shape"))
#install.packages("cmdstanr", repos = c('https://stan-dev.r-universe.dev', getOption("repos")
#devtools::install_github("rmcelreath/rethinking")
#devtools::install_github("rmcelreath/rethinking")
library(rethinking)
```

```
Loading required package: cmdstanr
```

```
This is cmdstanr version 0.9.0
```

```
- CmdStanR documentation and vignettes: mc-stan.org/cmdstanr
```

```
- Use set_cmdstan_path() to set the path to CmdStan
```

```
- Use install_cmdstan() to install CmdStan
```

```
Loading required package: posterior
```

```
This is posterior version 1.6.1
```

```

Attaching package: 'posterior'
```

```
The following objects are masked from 'package:stats':

    mad, sd, var
```

```
The following objects are masked from 'package:base':

    %in%, match
```

```
Loading required package: parallel
```

```
rethinking (Version 2.42)
```

```

Attaching package: 'rethinking'
```

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

    rstudent
```

```
library(brms) # for statistics
```

```
Loading required package: Rcpp

Loading 'brms' package (version 2.23.0). Useful instructions
can be found by typing help('brms'). A more detailed introduction
to the package is available through vignette('brms_overview').


Attaching package: 'brms'

The following objects are masked from 'package:rethinking':

    LOO, stancode, WAIC

The following object is masked from 'package:stats':

    ar
```

```
library(tidyverse) # for data wrangling
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.2.0     v readr     2.1.6
v forcats   1.0.1     v stringr   1.6.0
v ggplot2   4.0.2     v tibble    3.3.1
v lubridate 1.9.4     v tidyr     1.3.2
v purrr     1.2.1

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
x purrr::map()    masks rethinking::map()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

1. DAG practice

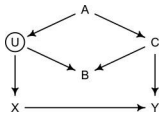


Figure 1: example DAG

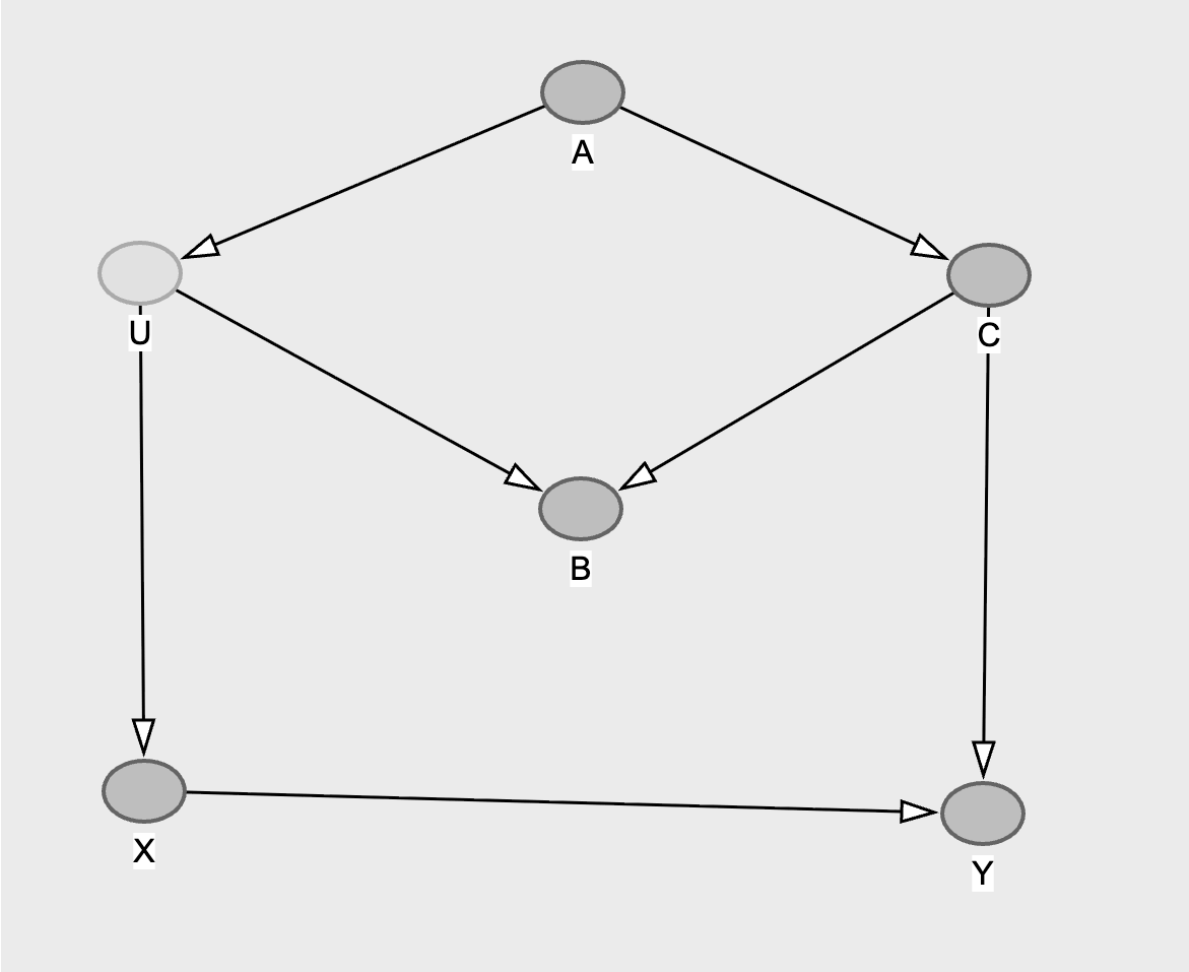
Directed Acyclic Graphs (DAGs) represent our understanding of causal influences in a system, with arrows connecting causes to effects. Consider the DAG above.

Now recreate the DAG above on dagitty.net. Leave the window open, as we'll be using it more.

Q1.1 Make a DAG

Please paste either your DAG image from the website or the DAG model code here.

Answer Q1.1)



There are four fundamental relations in a DAG: the fork, the pipe, the collider, and the descendent. This image shows them:

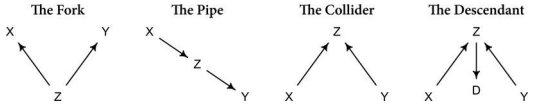


Figure 2: elemental confounds

Q1.2 Identify forks

Which forks do you see in the DAG you made on dagitty.net? Please write them out in a Quarto list (look up how to write a list if you don't remember!) in the form L ← M → N.

Answer Q1.2)

1. U <- A -> C
2. B <- C -> Y
3. X <- U -> B

Q1.3 Identify colliders

Which colliders do you see? Please write them out in a Quarto list in the form L → M ← N. Hint: there is more than one!

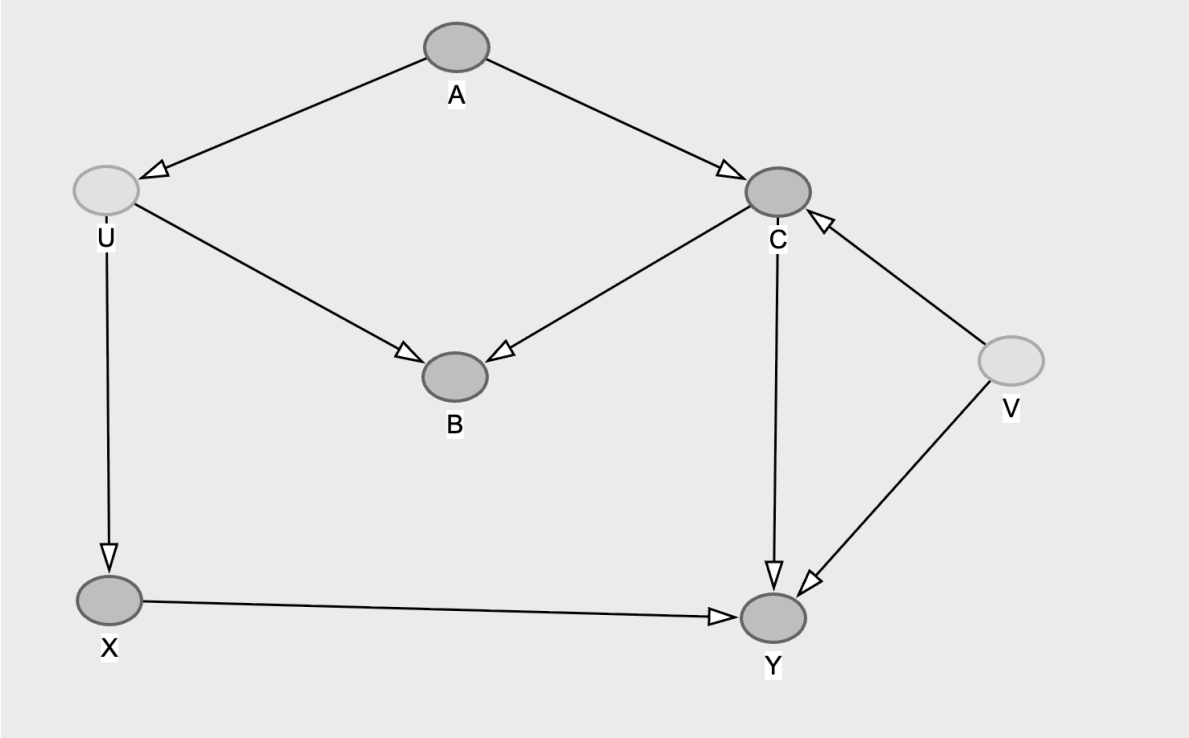
Answer Q1.3)

1. U -> B <- C
2. X <- Y -> C

Q1.4 Modify the DAG

Now modify the DAG (it should still be open on dagitty.net) to include the variable V, an unobserved cause of C and Y: C ← V → Y. Please paste either your DAG image from the website or the DAG model code here.

Answer Q1.4)



Q1.5 Identify paths

Reanalyze this new DAG. How many paths connect X to Y? Please list them in a Quarto list here:

AnswerQ1.5)

1. X <- Y
2. X<- U -> B <- C -> Y
3. X<- U -> B <- C <- V -> Y
4. X <- U <- A -> C -> Y
5. X <- U <- A -> C <- V ->Y

Q1.6 Identify open backdoor paths

Which paths must be closed to estimate the direct effect of X on Y? List the paths

Answer Q1.6)

1. X <- U <- A -> C -> Y

Q1.7 Identify variables to close the backdoor(s)

Given what you just wrote about paths to close, which variables should you condition on to estimate the direct effect of X on Y in your new DAG?

Answer Q1.7)

We would condition on A since it is the source of the 'open' backdoor

---

2. Foxes: Regression practice informed by DAGs

Figure 3: urban fox, pestuk.com

For this section, we are going to implement what we learned about DAGs into an example about urban fox territories from the `rethinking` package. Let's load in the data:

```
# Load in the fox data
data(foxes)
```

```
# Check out the fox data
?foxes
head(foxes)
```

```
  group avgfood groupsize area weight
1     1    0.37         2 1.09   5.02
2     1    0.37         2 1.09   2.84
3     2    0.53         2 2.05   5.33
4     2    0.53         2 2.05   6.07
5     3    0.49         2 2.12   5.85
6     3    0.49         2 2.12   3.25
```

From the Rethinking textbook: "The data in data(foxes) are 116 foxes from 30 different urban groups in England. These foxes are like street gangs. `Group size` varies from 2 to 8 individuals. Each group maintains its own urban territory. Some territories are larger than others. The `area` variable encodes this information. Some territories also have more `avgfood` than others. We want to model the `weight` of each fox [in kg]." For the questions below, we will assume the following DAG is appropriate for this system:

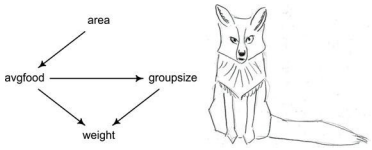


Figure 4: fox DAG

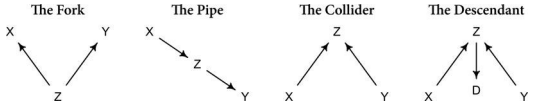


Figure 5: elemental confounds

Q2.1 Identify the fundamental relations in the fox DAG

Which of the first three fundamental relations above (Fork, Pipe, and Collider) do you see in the Fox DAG? List the names of the relations you see AND the particular paths (e.g. "Pipe1: X->Z->Y, Pipe2: X->Z->C and Fork1: X<-Z->Y")

Answer 2.1)

1. Pipe1 : Area -> avgfood -> Weight
2. Pipe2: avgfood -> groupsize -> weight
3. Collider1: avgfood -> weight <- groupsize
4. Fork1: groupsize <- avgfood -> weight
5. Pipe3: area -> avgfood -> groupsize -> weight

Total causal influence of area on weight

In this first part we are going to infer the total causal influence of area on weight. Would increasing the area available to each fox make it heavier (healthier)?

- First, we will standardize the variables.
- Second, we will use prior predictive simulation to check that our model's prior predictions stay within a reasonable outcome range.
- Third, we will run and interpret the models.

Standardize weight to mean zero and standard deviation of 1

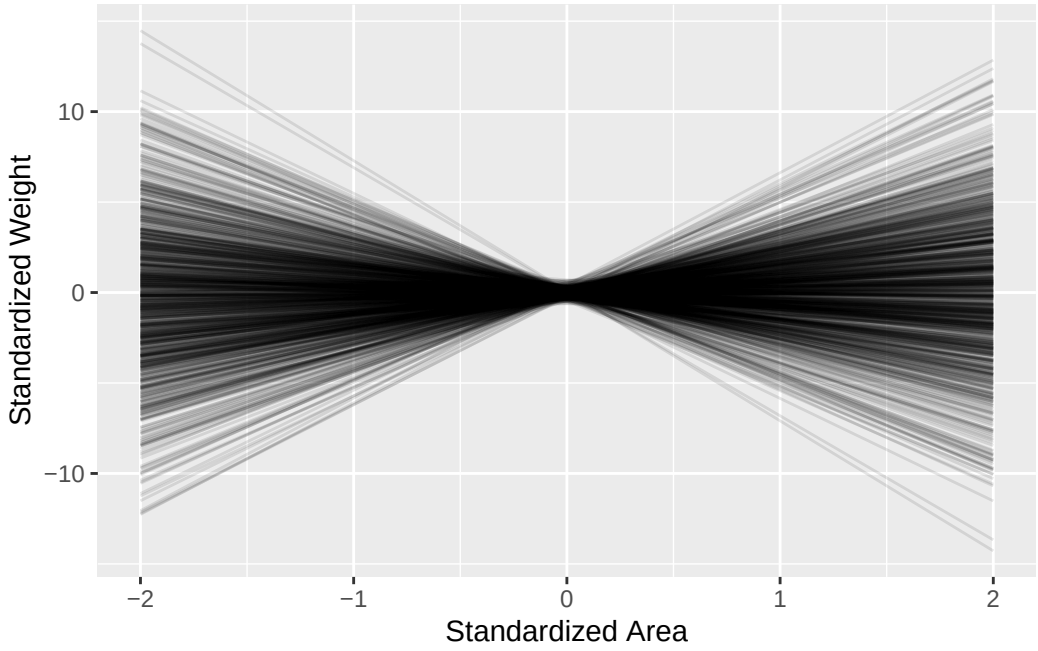
```
fox_dat <- foxes %>%
  as_tibble() %>%
  select(area, avgfood, weight, groupsize) %>%
  mutate(across(everything(), standardize))
```

Simulate from some priors for a linear regression with intercept *alpha* and slope *beta*: *alpha* ~ Gaussian(0, 0.2), *beta* ~ Gaussian(0, 2)

```
n <- 1000
priorsims <- tibble(group = seq_len(n),
       alpha = rnorm(n, 0, 0.2), # prior for alpha
       beta = rnorm(n, 0, 2)) %>% # prior for beta
  expand(nesting(group, alpha, beta), # the expand function gives us all possible combinatio
         area = seq(from = -2, to = 2, length.out = 100)) %>% # set up a range of areas
  mutate(weight = alpha + beta * area) # calculate weight from the parameters and area
```

Make a plot of what these priors imply.

```
ggplot(priorsims, aes(x = area, y = weight, group = group)) +
  geom_line(alpha = 1 / 10) +
  labs(x = "Standardized Area", y = "Standardized Weight")
```



It's pretty hard to understand what a "reasonable" fox weight is when it is in standardized units. Let's logic our way through this slowly.

Q2.2 Minimum fox weight

What to you seems like a reasonable minimum weight for a fox, in kg?

Answer2.2)

3.17kg (or 7lbs) seems reasonable weight for a very unhealthy fox

---

Q2.3 Maximum fox weight

What to you seems like a reasonable minimum weight for a fox, in kg?

Answer2.3)

11.3 kg (or 25 lbs) seems a reasonable weight for a heavy fox

---

Q2.4 Modify simulation plot

Remake your prior predictive simulation plot and add two horizontal lines, one each for the minimum and maximum weights that you just provided. Before plotting, make sure to *standardize* your values in kg so that they are plotted as centered values in units of standard deviation (i.e., subtract the mean and divide by the standard deviation of foxes$weight).

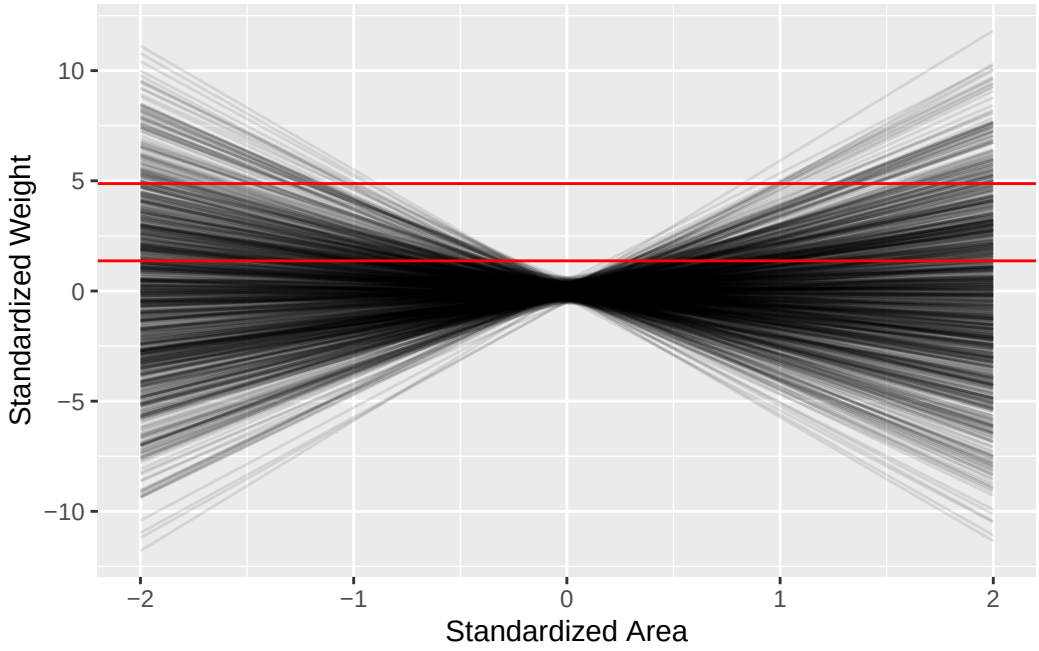
Answer2.4)

```
n <- 1000
foxes_w<- tibble(group = seq_len(n),
       alpha = rnorm(n, 0, 0.2), # prior for alpha
       beta = rnorm(n, 0, 2)) %>% # prior for beta
  expand(nesting(group, alpha, beta), # the expand function gives us all possible combinatio
         area = seq(from = -2, to = 2, length.out = 100)) %>% # set up a range of areas
  mutate(weight = alpha + beta * area) # calculate weight from the parameters and area

foxes_w_lines <- foxes_w |>
  mutate(foxes_min = (3.17 - (mean(foxes_w$weight)))/sd(foxes_w$weight)) |>
  mutate(foxes_max = (11.3 - (mean(foxes_w$weight)))/sd(foxes_w$weight))

foxes_w_lines_plot = ggplot(foxes_w_lines, aes(x = area, y = weight, group = group)) +
  geom_line(alpha = 1/10) +
  geom_hline(yintercept = foxes_w_lines$foxes_min , color = 'red') +
  geom_hline(yintercept = foxes_w_lines$foxes_max , color = 'red') +
  labs(x = "Standardized Area", y = "Standardized Weight")

foxes_w_lines_plot
```



Q2.5 Evaluate prior predictive simulation

Do your priors seem reasonable? You haven't seen any data yet, though you have marked out the minimum and maximum weights you expect foxes to be. Do your priors greatly exceed those values? Please explain your thinking.

Answer Q2.5)

Looking at the priors, it seems that our maximum weight is about right, however, our minimum seems to be a little too high. In general, the priors do seem reasonable.

Q2.6 Refine priors

Remake and plot a set of prior simulations that use priors you think are reasonable (adjusting the code from above would work well for this). Be sure to include the minimum and maximum

fox weights that you expect. You can iterate on this a few times (simulate, plot, adjust, etc.) until you arrive at priors that make sense to you.

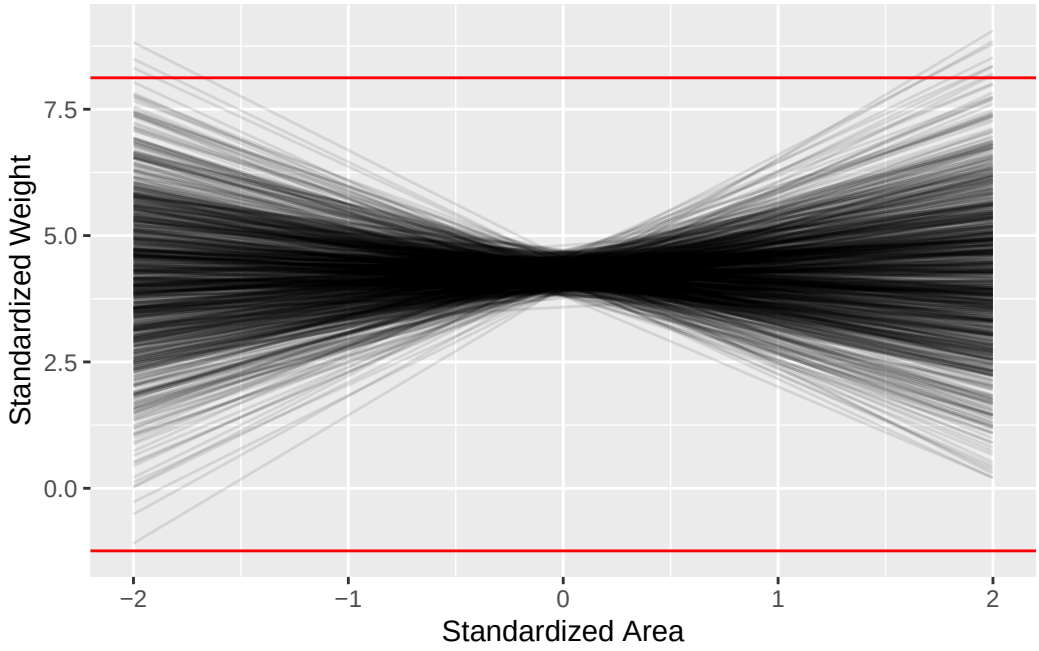
Answer Q2.6)

```
n <- 1000
foxes_w<- tibble(group = seq_len(n),
       alpha = rnorm(n, 4.25, 0.2), # prior for alpha
       beta = rnorm(n, 0, 0.75)) %>% # prior for beta
  expand(nesting(group, alpha, beta), # the expand function gives us all possible combinatio
         area = seq(from = -2, to = 2, length.out = 100)) %>% # set up a range of areas
  mutate(weight = alpha + beta * area) # calculate weight from the parameters and area



foxes_w_lines <- foxes_w |>
  mutate(foxes_min = (3.17 - (mean(foxes_w$weight)))/sd(foxes_w$weight)) |>
  mutate(foxes_max = (11.3 - (mean(foxes_w$weight)))/sd(foxes_w$weight))

foxes_w_lines_plot2 = ggplot(foxes_w_lines, aes(x = area, y = weight, group = group)) +
  geom_line(alpha = 1/10) +
  geom_hline(yintercept = foxes_w_lines$foxes_min, color = 'red') +
  geom_hline(yintercept = foxes_w_lines$foxes_max, color = 'red') +
  labs(x = "Standardized Area", y = "Standardized Weight")
foxes_w_lines_plot2
```



Run models

Run a model predicting average food as a function of area. Modify the code for the priors below to match the priors you just chose.

```
food_on_area <- brm(avgfood ~ 1 + area,
                    data = fox_dat,
                    family = gaussian,
                    # Here we set the priors that we investigated earlier
                    prior = c(prior(normal(4.25, 0.2), class = Intercept),
                              prior(normal(0, 0.75), class = b,),
                              prior(exponential(1), class = sigma)),
                    iter = 4000, warmup = 2000, chains = 4, cores = 4, seed = 1234,
                    file = "output/food_on_area")
```

Check out the summary:

```
summary(food_on_area)
```

```
 Family: gaussian
  Links: mu = identity
Formula: avgfood ~ 1 + area
   Data: fox_dat (Number of observations: 116)
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup draws = 8000

Regression Coefficients:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.07      0.04    -0.02     0.16 1.00     7099     5673
area          0.88      0.05     0.79     0.96 1.00     7947     6057

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.48      0.03     0.42     0.55 1.00     6292     5731

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

We see a fairly strong effect of area on the average amount of food. Because we standardized each variable by standard deviations, our units are now in "standard deviations". (*We can backtransform these value to translate this back to the normal units! We won't do that here, as we'll get a lot more practice with that when we get to generalized linear models, but just know that if you are annoyed by the unitless values, there's a way out!*)

We find that for an increase of 1 standard deviation in area, we expect to see a 0.88 standard deviation increase in food. The 95% compatibility interval for the area parameter is 0.79 to 0.96, which does not include zero. Logically this makes sense, as a greater area would have more prey available.

Q2.7 Run a model for the impact of food on fox weight

Now infer the total impact of adding food to a territory. Run a model with `weight` as a function of `avgfood`. Based on your results, does more food make foxes heavier? In your opinion, is this expected or unexpected? Please explain in two (2) or more sentences.

```r
avgfood_on_weight <- brm(weight ~ 1 + avgfood,
                    data = fox_dat,
                    family = gaussian,
                    # Here we set the priors that we investigated earlier
                    prior = c(prior(normal(1.5, 0.2), class = Intercept),
                              prior(normal(0, 0.5), class = b,),
                              prior(exponential(1), class = sigma)),
                    iter = 4000, warmup = 2000, chains = 4, cores = 4, seed = 1234,
                    file = "output/avgfood_on_weight")
```

```r
summary(avgfood_on_weight)
```

```
 Family: gaussian
  Links: mu = identity
Formula: weight ~ 1 + avgfood
   Data: fox_dat (Number of observations: 116)
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup draws = 8000

Regression Coefficients:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.29      0.10     0.11     0.48 1.00     5974     5911
avgfood      -0.02      0.10    -0.21     0.16 1.00     7319     6010

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.05      0.07     0.92     1.21 1.00     6195     5797

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Answer Q2.7)

Based on our results (an estimate of -0.02) more food does not make foxes heavier which is extremely unexpected. However, the 95% credibility interval overlaps with zero, thus that leads us to suggest that, given our model, the effect of average food on fox weight is not different from zero. Looking at the DAG retrospectively, it makes sense that our model would come to this result, because there is a path avgfood -> groupsize -> weight, that is unaccounted for in our model.

Q2.8 Is there a variable we should condition upon?

We just estimated the total impact of `avgfood` on `weight`, which includes both direct and indirect paths. Think back to your DAG elemental confounds. If we want to estimate only the direct impact of `avgfood` on `weight`, which variable should we condition upon?

Answer Q2.8)

We believe that we need to condition on the variable group size because it is the indirect path mentioned in the question.

Add in `groupsize`

In the previous model we saw no effect of `avgfood` on fox `weight`, but we have an extra path that we need to account for, since `avgfood` flows to `weight` through `groupsize`.

First, let's look at the separate effect of `groupsize` in a univariate regression, just like with `avgfood`.

Q2.9: What's your hypothesis about how group size affects fox weight?

Before running the model, how do you think the number of foxes in a group `groupsize` would affect fox weight? Why?

Answer Q2.9)

Theoretically, group size should have an effect on fox weight. This is because, the larger a group, the less food each individual can have. So we would expect a negative effect of group size on weight.

Now let's run the model:

```
group_on_weight <- brm(weight ~ 1 + groupsize,
                       data = fox_dat,
                       family = gaussian,
                       prior = c(prior(normal(0, 0.2), class = Intercept),
                                 prior(normal(0, 0.5), class = b,),
                                 prior(exponential(1), class = sigma)),
                       iter = 4000, warmup = 2000, chains = 4, cores = 4, seed = 1234,
                       file = "output/group_on_weight")
```

```
summary(group_on_weight)
```

```
 Family: gaussian
  Links: mu = identity
Formula: weight ~ 1 + groupsize
   Data: fox_dat (Number of observations: 116)
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup draws = 8000

Regression Coefficients:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    -0.00      0.08    -0.16     0.16 1.00     7636     5822
groupsize    -0.15      0.09    -0.33     0.02 1.00     7918     5474

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.00      0.07     0.88     1.14 1.00     8166     6280

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Similar to the total effect of `avgfood` on `weight` in a univariate regression, we see no effect; the estimate for the slope of `groupsize` on `weight` is -0.16, but the 95% CI are between -0.33 and 0.02, which includes 0. This suggests the effect of `groupsize` on `weight` could very well be zero, *given this model*.

To estimate the **direct effect** of `avgfood` on `weight`, we need to block the indirect path through `groupsize`. To do that, we include `groupsize` in a multiple regression (along with our main interest, `avgfood`). (By coincidence, this will also give us the direct effect of `groupsize` on `weight`. Look hard at the DAG and ask Calvin or Malin if the reasoning here isn't clear).

Let's add in `groupsize` to block the pipe `weight->groupsize->avgfood`:

```
food_direct <- brm(weight ~ 1 + avgfood + groupsize,
                   data = fox_dat,
                   family = gaussian,
                   prior = c(prior(normal(0, 0.2), class = Intercept),
                             prior(normal(0, 0.5), class = b,),
                             prior(exponential(1), class = sigma)),
                   iter = 4000, warmup = 2000, chains = 4, cores = 4, seed = 1234,
                   file = "output/food_direct")

summary(food_direct)
```

```
 Family: gaussian
  Links: mu = identity
Formula: weight ~ 1 + avgfood + groupsize
   Data: fox_dat (Number of observations: 116)
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup draws = 8000

Regression Coefficients:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.00      0.08    -0.15     0.16 1.00     6422     4722
avgfood       0.47      0.18     0.11     0.83 1.00     4288     4614
groupsize    -0.57      0.18    -0.92    -0.22 1.00     4165     4943

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.96      0.06     0.85     1.10 1.00     5952     4955

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Interpret the multiple regression output

Q2.10a

What are the effects of `avgfood` and `groupsize` now that you have accounted for both variables?

Answer 2.10A)

1. The effect of avgfood on weight is 0.47
2. The effect of groupsize on weight is -0.57

Q2.10b

How does this interpretation change your interpretation from the univariate regressions of each variable separately with `weight`?

Answer 2.10B)

This contradicts our interpretations from both the univariate regressions that we ran previously. Both of the univariate models' credibility intervals overlapped with zero, suggesting that given our models the effects were not different from zero. But now our multiple regression shows that each have an effect on weight that is different from zero.

Q2.10c

Provide a small discussion (2-4 sentences) explaining in your own words why these results turned out the way they did, in the context of the ecological system of fox territories. Include why you think that the univariate regressions may have suggested no relationship while the multiple regression suggests a different answer.

Answer 2.10c)

We found weight increases 0.47kg with every 1 unit increase in average food. Our 95% credibility interval is between 0.11 and 0.83, suggesting that given our model, the effects of average food on weight is different than zero. Additionally, we found that weight decreases 0.57kg with every 1 unit increase in group size. Our 95% credibility interval is between -0.92 and -0.22, suggesting that given our model, the effects of group size on weight is different from zero. When thinking about the context of the ecological system, our results make sense. While knowing the average amount of food available to an individual should allow us to understand the weight of the individual, without knowing about the individual's group size, we cannot be sure how much food that individual is actually getting. Vice versa, while knowing a fox's group size should allow us to get an idea of it's weight, without knowing the average food available to the group, we can't actually know how much food the individual is getting.

Render to PDF

When you have finished, remember to pull, stage, commit, and push with GitHub:

- Pull to check for updates to the remote branch
- Stage your edits (after saving your document!) by checking the documents you'd like to push
- Commit your changes with a commit message
- Push your changes to the remote branch

Then submit the well-labeled PDF on Gradescope. Thanks!