Statistical_Reasoning_3_Multiple_Regression_And_DA

Seleonalela & Veronika

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

```r
# Run this to install some data packages
# devtools::install_github("rmcelreath/rethinking")
library(rethinking)
```

```
Loading required package: cmdstanr

This is cmdstanr version 0.9.0.9000

- 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
```

```r
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
```

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

```
Warning: package 'tibble' was built under R version 4.5.2
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.3.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
```

```
-- 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 becom
```

1. DAG practice

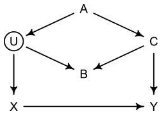


Figure 1: example DAG

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.

dag { A [pos="-0.707,-0.655"] B [pos="-0.703,0.070"] C [pos="0.224,-0.395"] U [latent,pos="-1.583,-0.297"] X [pos="-1.554,0.574"] Y [pos="0.232,0.584"] A -> C A -> U C -> B C -> Y U -> B U -> X X -> Y }

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

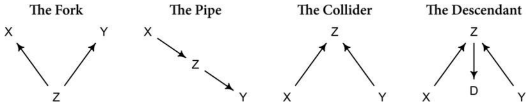


Figure 2: elemental confounds

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.

- fork: U <- A -> C

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!

- collider:
  - U -> B <- C
  - 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.

dag { A [pos=“-0.707,-0.655”] B [pos=“-0.703,0.070”] C [pos=“0.224,-0.395”] U [latent,pos=“-1.583,-0.297”] V [latent,pos=“-0.359,0.137”] X [pos=“-1.554,0.574”] Y [pos=“0.232,0.584”] A -> C A -> U C -> B C -> Y U -> B U -> X V -> C V -> Y X -> Y }

Q1.5 Identify paths

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

- paths connecting X to Y
  - X <- U <- A -> C -> Y
  - X <- U <- A -> C <- V -> Y
  - X <- U -> B <- C -> Y
  - X <- U -> B <- C <- V -> Y
  - X -> Y

Q1.6 Identify open backdoor paths

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

- paths that must be closed
  - X <- U -> A -> C -> Y
  - X <- U -> A -> C <- V -> 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?

We should condition on C because it is closer to Y in our path from X to Y.

---

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:

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

```r
# 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:

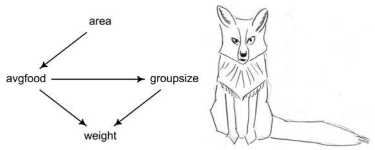


Figure 4: fox DAG

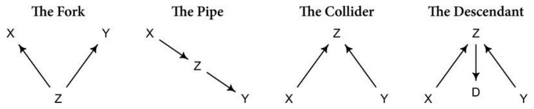


Figure 5: elemental confounds

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 Q1.2:

- Forks
  - groupsize <- avgfood -> weight

- Pipes
  - area -> avgfood -> weight
  - avgfood -> groupsize -> weight
  - area -> avgfood -> groupsize -> weight
- Colliders
  - avgfood -> weight <- groupsize

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

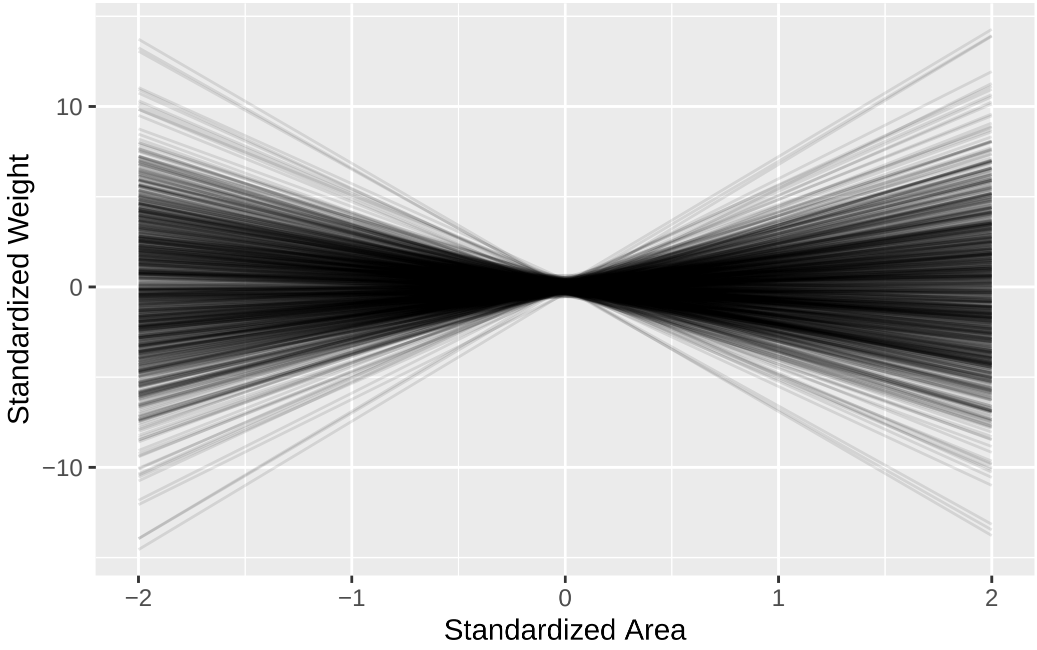
```
#standardize weight to mean zero and sd of 1
fox_dat <- foxes %>%
  as_tibble() %>%
  select(area, avgfood, weight, groupsize) %>%
  mutate(across(everything(), standardize))
```

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

```
# stimulate some priors for a linear regression w intercept alpha and slope beta
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.

```r
# plot results
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?

Answer Q2.2:

We think a reasonable minimum weight for a fox would be 1 kg (~ 2 pounds).

Q2.3 Maximum fox weight

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

Answer Q2.3:

We think a reasonable maximum weight for a fox would be 8 kg.

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).

```r
# finding mean fox weight
mean(foxes$weight)
```

```
[1] 4.529655
```

```r
# finding standard deviation for fox weight
sd(foxes$weight)
```

```
[1] 1.184023
```

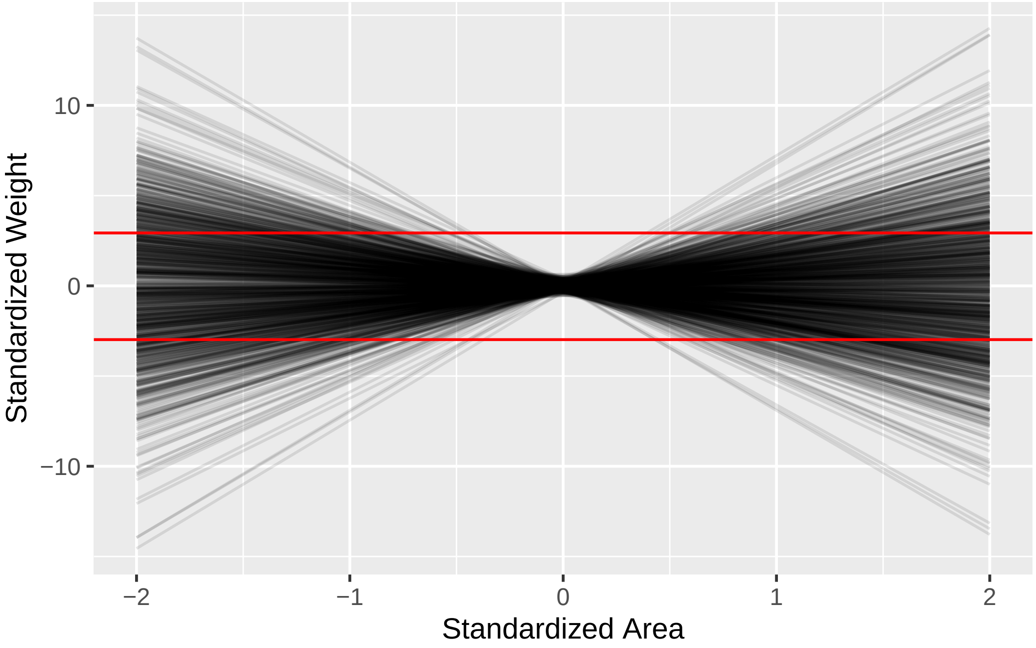
```r
# standardizing our predicted reasonable min and max values
z <- (8-4.529655)/1.184023
z
```

```
[1] 2.930978
```

```r
v <- (1-4.529655)/1.184023
v
```

```
[1] -2.98107
```

```
# modified simulation plot by adding horizontal lines at our assumed reasonable min and max
ggplot(priorsims, aes(x = area, y = weight, group = group)) +
  geom_line(alpha = 1 / 10) +
  labs(x = "Standardized Area", y = "Standardized Weight")+
  # geom_hline() adds the line!
  geom_hline(yintercept = c(v, z), color = "red", linetype = 1)
```



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:

Our predicted priors seem unreasonable. Our priors appear to greatly subceed the values of weight on the upper bound and exceed the values of weight on lower bound, and are thereby excluding a large amount of data. Therefore we think we should broaden our priors, so we choose a smaller min (0.5) and a larger max weight (10) values.

```
# Adjusted min and max values after generating last plot => Min = 0.5 and Max = 10
# Standardizing adjusted min and max values:
a <- (10-4.529655)/1.184023
a # Standardized adjusted max
```

```
[1] 4.620134
```

```
b <- (0.5-4.529655)/1.184023
b # Standardized adjusted min
```

```
[1] -3.403359
```

Q2.6 Refine priors

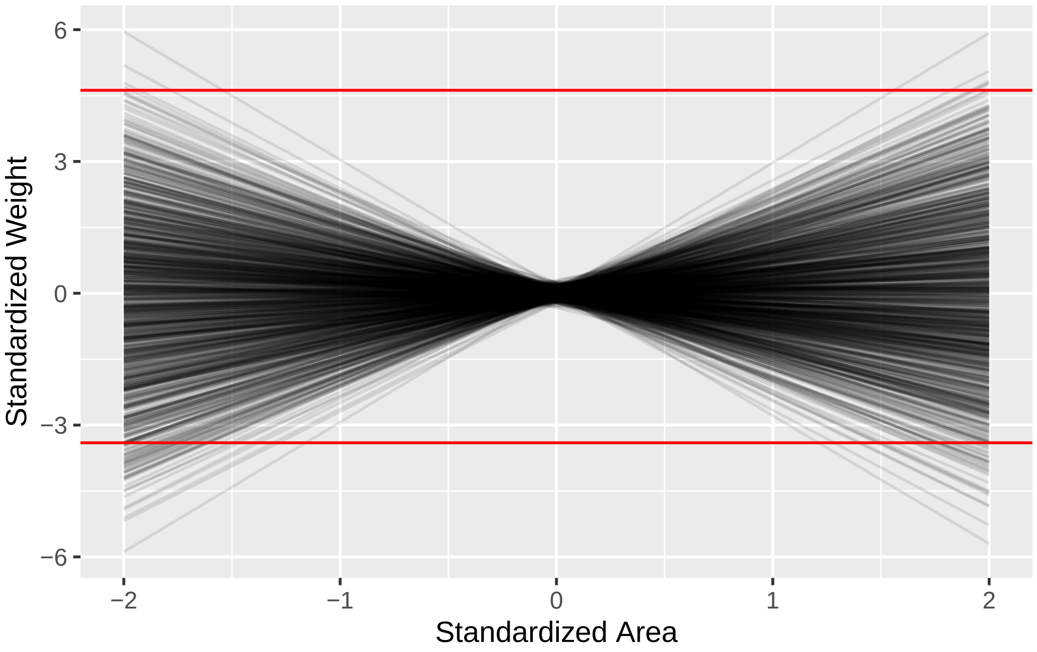
Remake and plot a set of prior simulations that use priors you think are reasonable (adjusting the code from above would work well for this). Be sure to include the minimum and maximum fox weights that you expect. You can iterate on this a few times (simulate, plot, adjust, etc.) until you arrive at priors that make sense to you.

Answer Q2.6:

We were able to arrive at priors that make sense to us after the first adjustment from our first predicted priors.

```
# changed the priors: alpha to 0.1, and beta to 1
n <- 1000
priorsims <- tibble(group = seq_len(n),
       alpha = rnorm(n, 0, 0.1), # prior for alpha
       beta = rnorm(n, 0, 1)) %>% # 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
```

```
# changed the lines to contain more of the data
ggplot(priorsims, aes(x = area, y = weight, group = group)) +
  geom_line(alpha = 1 / 10) +
  labs(x = "Standardized Area", y = "Standardized Weight")+
  geom_hline(yintercept = c(b, a), color = "red", linetype = 1)
```



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.

```
# model predicting avgfood as a function of area
food_on_area <- brm(avgfood ~ 1 + area,
                    data = fox_dat,
                    family = gaussian,
                    # Here we set the priors that we investigated earlier. (alpha = 0.1, beta
                    prior = c(prior(normal(0, 0.1), class = Intercept),
                              prior(normal(0, 1), 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:

```r
summary(food_on_area) # checking Rhat value is 1
```

```
 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     0.00      0.04    -0.08     0.08 1.00     7350     5390
area          0.88      0.04     0.79     0.97 1.00     7494     5957

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.48      0.03     0.42     0.54 1.00     8491     6467

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
# change the model to be weight as a function of avgfood
food_on_weight <- brm(weight ~ 1 + avgfood,
                    data = fox_dat,
                    family = gaussian,
                    # Here we set the priors that we investigated earlier
                    prior = c(prior(normal(0, 0.1), class = Intercept),
                              prior(normal(0, 1), class = b,),
                              prior(exponential(1), class = sigma)),
                    iter = 4000, warmup = 2000, chains = 4, cores = 4, seed = 1234,
                    file = "output/food_on_weight")
```

```
Compiling Stan program...

Start sampling

Found more than one class "stanfit" in cache; using the first, from namespace 'rethinking'

Also defined by 'rstan'

Found more than one class "stanfit" in cache; using the first, from namespace 'rethinking'

Also defined by 'rstan'

Found more than one class "stanfit" in cache; using the first, from namespace 'rethinking'

Also defined by 'rstan'

Found more than one class "stanfit" in cache; using the first, from namespace 'rethinking'

Also defined by 'rstan'
```

```r
# check it worked
# check rhat
summary(food_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    -0.00      0.07    -0.14     0.13 1.00     8273     5825
avgfood      -0.02      0.09    -0.20     0.16 1.00     8199     5804

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.01      0.07     0.89     1.15 1.00     8163     5841

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).
```

More food does not make foxes heavier.

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:

In order to estimate only the direct impact of avgfood on weight we need to condition on groupsize in order to close the back door. By holding groupsize constant, we can isolating the direct path from avgfood to weight, allowing us to see the true impact of avgfood on weight without any influence from groupsize (if there is any).

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:

We predict that group size and weight will have an inversely proportional relationship, in that group size (as group size increases) would have a negative effect on weight (weight decreases). Because the more mouths to feed in a group the less food there is per individual per kill, leading to a lower average weight amongst members of a larger groups. In smaller groups there are fewer individuals, therefore you have larger portions of food per individual per kill.

Now let's run the model:

```r
# change the model to be weight as a function of groupsize
groupsize_on_weight <- brm(weight ~ 1 + groupsize,
                    data = fox_dat,
                    family = gaussian,
                    # Here we set the priors that we investigated earlier
                    prior = c(prior(normal(0, 0.1), class = Intercept),
                              prior(normal(0, 1), class = b,),
                              prior(exponential(1), class = sigma)),
                    iter = 4000, warmup = 2000, chains = 4, cores = 4, seed = 1234,
                    file = "output/groupsize_on_weight")
```

```
Compiling Stan program...
```

```
Start sampling
```

```
Found more than one class "stanfit" in cache; using the first, from namespace 'rethinking'
```

```
Also defined by 'rstan'

Found more than one class "stanfit" in cache; using the first, from namespace 'rethinking'

Also defined by 'rstan'

Found more than one class "stanfit" in cache; using the first, from namespace 'rethinking'

Also defined by 'rstan'

Found more than one class "stanfit" in cache; using the first, from namespace 'rethinking'

Also defined by 'rstan'
```

```
# checking Rhat value = 1
summary(groupsize_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      0.07    -0.13     0.14 1.00     8344     5840
groupsize    -0.16      0.09    -0.34     0.02 1.00     7170     5304

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.99      0.07     0.88     1.13 1.00     7387     5753

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`:

::: {.cell}

```
{.r .cell-code} # added group size to close the back door path from weight ->
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")
```

::: {.cell-output .cell-output-stderr}

```
Compiling Stan program...
```

:::

::: {.cell-output .cell-output-stderr}

```
Start sampling
```

:::

::: {.cell-output .cell-output-stderr}

```
Found more than one class "stanfit" in cache; using the first, from namespace
'rethinking'
```

:::

::: {.cell-output .cell-output-stderr}

```
Also defined by 'rstan'
```

:::

::: {.cell-output .cell-output-stderr}

```
Found more than one class "stanfit" in cache; using the first, from namespace
'rethinking'
```

:::

::: {.cell-output .cell-output-stderr}

```
Also defined by 'rstan'
```

:::

::: {.cell-output .cell-output-stderr}

```
Found more than one class "stanfit" in cache; using the first, from namespace
'rethinking'
```

:::

::: {.cell-output .cell-output-stderr}

```
Also defined by 'rstan'
:::
::: {.cell-output .cell-output-stderr}
Found more than one class "stanfit" in cache; using the first, from namespace
'rethinking'
:::
::: {.cell-output .cell-output-stderr}
Also defined by 'rstan'
:::
{.r .cell-code} summary(food_direct) # checking Rhat = 1 (if 1 means model
fits) :::
```

Interpret the multiple regression output

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.

Q2.10a

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

Answer Q2.10a:

Given the model that now accounts for `avgfood` and `groupsize`, the effect of `avgfood` on `weight` is 0.47, therefore for an increase of 1 standard deviation in `avgfood` there is an expected increase of 0.47 standard deviations in weight; while the effect of `groupsize`on `weight` is -0.57, meaning that for an increase of 1 standard deviation in `groupsize`, weight is expected to decrease by 0.57 standard deviations.

Q2.10b

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

Answer Q2.20b:

Univariant regressions: * `avgfood` on `weight` = -0.02 * `groupsize`on `weight`= -0.16 Multivariant regression: * `avgfood` on `weight` = 0.47 * `groupsize`on `weight`= -0.57

When `avgfood` was being considered individually for its influence on `weight`, the relationship between the variables was described by a slope of -0.02 (little correlation), while when `avgfood` and `groupsize` are both being accounted for (multivarient regression), the influence of `avgfood` on weight is 0.47. This means for an increase of 1 standard deviation in `avgfood` you have an increase of 0.47 standard deviations in `weight`. This means that `groupsize` was supressing the true influence of average food on `weight`.

When `groupsize` was being considered individually for its influence on `weight`, the relationship between the variables was described by a slope of -0.16, while when `avgfood` and `groupsize` are both being accounted for, the influence of `groupsize` on `weight` is -0.57, meaning for an increase of 1 standard deviation in `groupsize` you have an decrease of 0.57 standard deviations in `weight`. This means that `avgfood` was masking the true negative effect of group size on `weight`, since `avgfood` positively effects weight, so it brought up the effect of `groupsize` on `weight`.

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 Q2.10c:

Having a larger territory means there are more resources such as food available to the foxes. So individuals or groups of foxes that have larger territories will have more food. Now when we look at how "groupsize" impacts the weight of foxes using a multiple regression model, we see a negative association represented by a slope of -0.57, indicating that for an increase in 1 standard deviation in group size we have a decrease of 0.57 standard deviations in weight. Now if we look at how "avgfood" impacts the weight of foxes, using a multiple regression model, we see a positive association represented by a slope of 0.47 when we hold "groupsize" constant (to prevent "groupsize" from supressing the true effects of "avgfood" on weight) unlike what we saw when we ran the linear regression model (slope -0.02). We believe that the univarient regression suggested no relationship because the "groupsize" variable was supressing the true impacts of "avgfood" on weight since larger groups have lower weights; therefore, by holding "groupsize" constant using the multivarient regression, we were able prevent "groupsize" from bringing down the magnitude to which food impactes weight, allowing us to see the true positive effect of “avgfood” on weight.

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!