# statsreasoning3

Paige Gardner

## Libraries

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

```
Warning: package 'brms' was built under R version 4.4.3
```

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

    ar
```

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

```
Warning: package 'ggplot2' was built under R version 4.4.3
```

```
Warning: package 'tibble' was built under R version 4.4.3
```

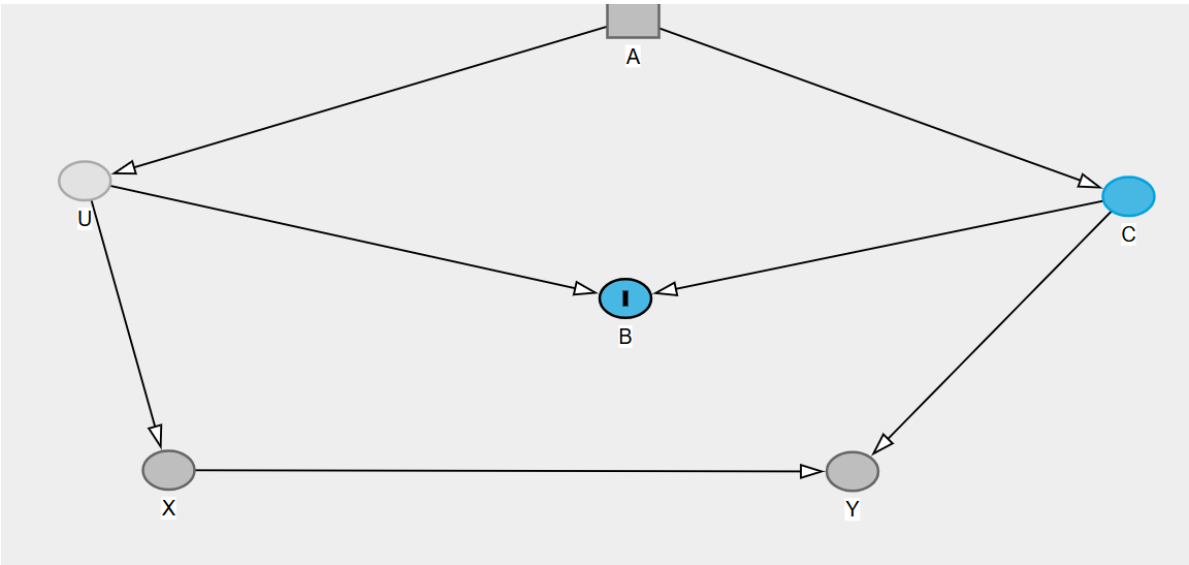
```
Warning: package 'purrr' was built under R version 4.4.3
```

```
-- 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   4.0.1     v tibble    3.3.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.2.1

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

```r
# a function to scale and center. from rethinking package
standardize <- function(x) {
    x <- scale(x)
    z <- as.numeric(x)
    attr(z,"scaled:center") <- attr(x,"scaled:center")
    attr(z,"scaled:scale") <- attr(x,"scaled:scale")
    return(z)
}
```

## Q1.1



### Q1.2 Forks

U ← A → B

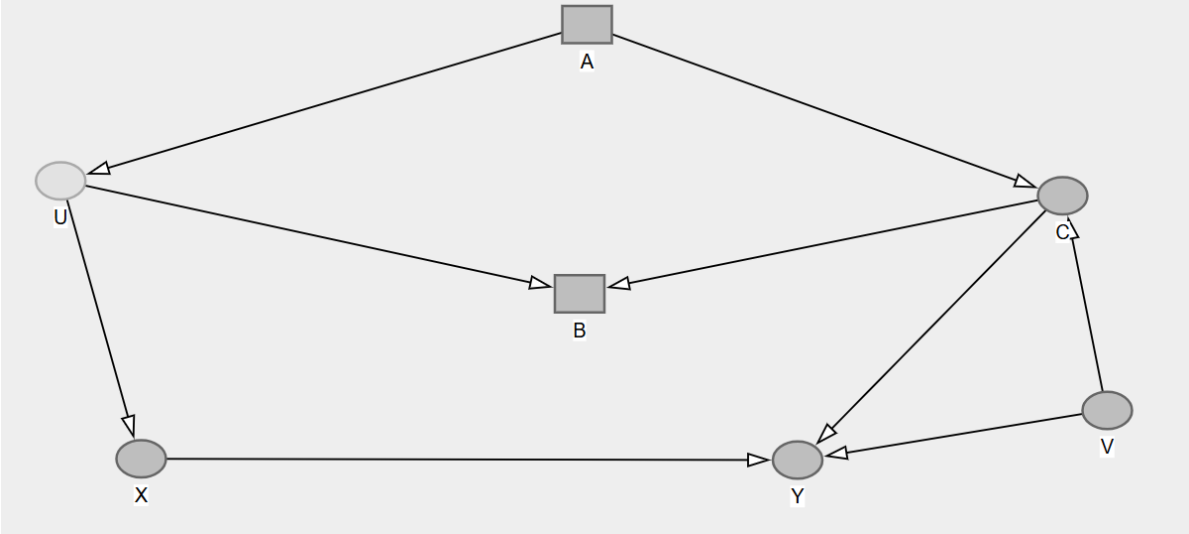
B← C → Y

### Q1.3 Colliders

X→ Y← C

U ← B → C

### Q1.4




### Q1.5

X <- U -> B <- C -> Y

X <- U -> B <- C <- V -> Y

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

X <- U <-A -> C <- V -> Y

### Q1.6 Identify open backdoor paths

A path is open if it has no collider, and closed if it has a collider.

The only open door path:

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

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

Which variables to condition on?

I don't fully understand what it means to condition on a variable

Condition on variables U and C?

# Part 2

## Load in data

```
library(rethinking)
```

```
Loading required package: cmdstanr
```

```
This is cmdstanr version 0.9.0.9000
```

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

```
- CmdStan path: C:/Users/jgard/.cmdstan/cmdstan-2.37.0
```

```
- CmdStan version: 2.37.0
```

```
A newer version of CmdStan is available. See ?install_cmdstan() to install it.
To disable this check set option or environment variable cmdstanr_no_ver_check=TRUE.
```

```
Loading required package: posterior
```

```
Warning: package 'posterior' was built under R version 4.4.3
```

```
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 _by_ '.GlobalEnv':

    standardize

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

    map

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

    LOO, stancode, WAIC

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

    rstudent
```

```
foxes <- read.csv('https://raw.githubusercontent.com/rmcelreath/rethinking/refs/heads/master/
```

```
?foxes
```

```
starting httpd help server ... done
```

```
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
```

## Q 2.1

Answer: I see a pipe: Area -> avgfood -> weight

This pipe can even be larger: Area -> avgfood -> groupsize -> weight

```
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
```

### Q2.2 Minimum fox weight

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

**Answer**: 5 kg

---

### Q2.3 Maximum fox weight

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

**Answer:** 15 kg

## Q 2.4

### Step 1: Standardize the min and max values

```
# I asked co-pilot for help
# I'm making a new function for single values, since you can't standardize a single value (no
standardize_single <- function(x, center, scale) {
  z <- (x - center) / scale
  attr(z, "scaled:center") <- center
  attr(z, "scaled:scale") <- scale
  z
}

standardize_single(5,10,2) # simulated mean of 10, SD of 2
```

```
[1] -2.5
attr(,"scaled:center")
[1] 10
attr(,"scaled:scale")
[1] 2
```

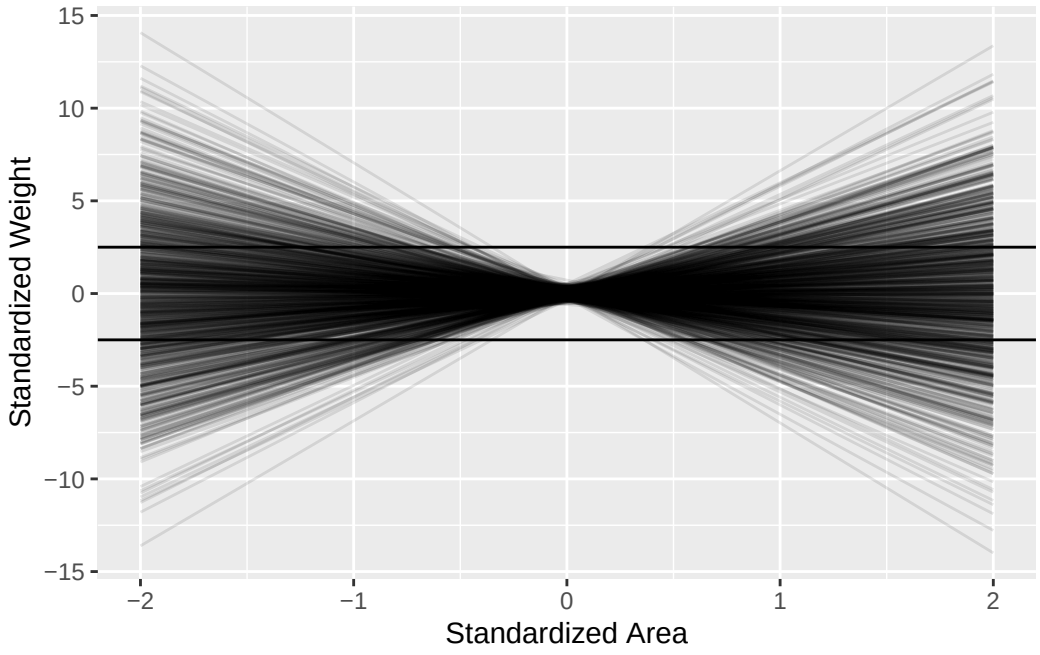
```
standardize_single(15,10,2)
```

```
[1] 2.5
attr(,"scaled:center")
[1] 10
attr(,"scaled:scale")
[1] 2
```

```
# Min: -2.5
# Max: 2.5
```

**Step 2: Plot**

```
ggplot(priorsims, aes(x = area, y = weight, group = group)) +
  geom_line(alpha = 1 / 10) +
  geom_hline(yintercept= c(-2.5, 2.5))+
  labs(x = "Standardized Area", y = "Standardized Weight")
```




## Q2.5

Answer: The priors seem much too large in the tails. I set the maximum size as 15 kg and the minimum size as 5 kg, assuming that foxes are actually pretty small animals. I was

fairly conservative in this assumption, but the priors set in the first part of this exercise have extremely large tails.

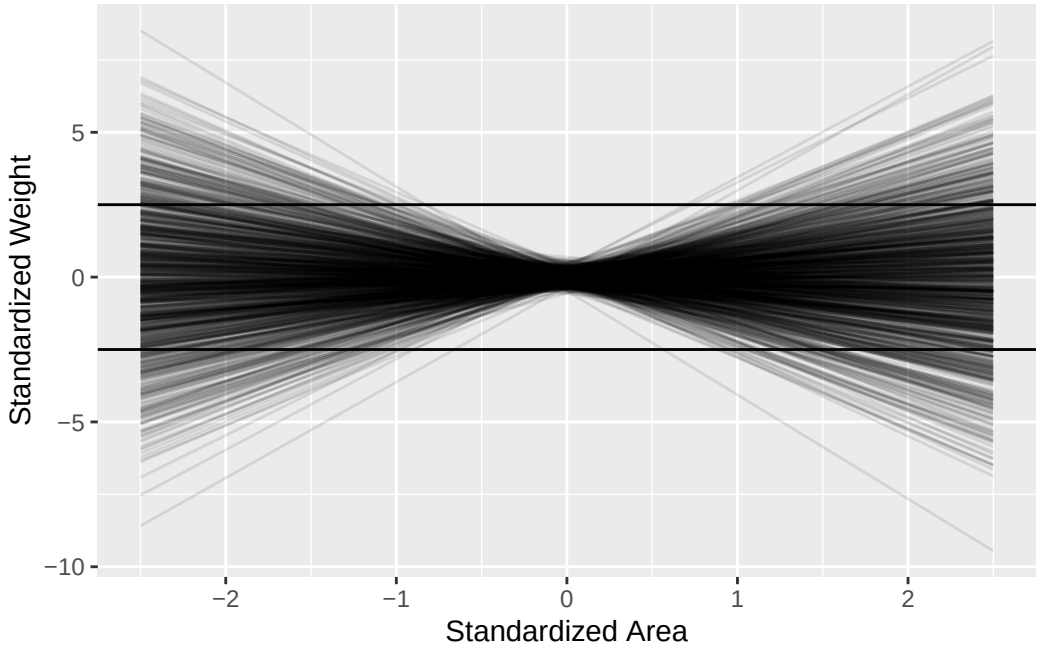
### Q 2.6

I think that the effect of area on weight is being inflated. I am going to make the standard deviation 1.5 instead of 2.

*Update: I ran it with 1.5, then updated that to 1 because the range of weights still felt too broad.*

```
# this is a prior predictive simulation, and NOT the same as a brm
n <- 1000
priorsims_updated <- tibble(group = seq_len(n),
       alpha = rnorm(n, 0, 0.2), # 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 = -2.5, to = 2.5, length.out = 100)) %>% # set up a range of areas
  mutate(weight = alpha + beta * area) # calculate weight from the parameters and area
```

```
ggplot(priorsims_updated, aes(x = area, y = weight, group = group)) +
  geom_line(alpha = 1 / 10) +
  geom_hline(yintercept= c(-2.5, 2.5))+
  labs(x = "Standardized Area", y = "Standardized Weight")
```




**Q2.7**

```
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 for the inte
                                # of weight when avgfood) I would think this would be negative
                                prior(normal(0, 1), class = b,), # prior for slope of the effe
                                prior(exponential(1), class = sigma)), # prior for how much da
                      # asked co-pilot for help on this: set exponential to make positive valu
                      # variation around regression for larger values of area
                      iter = 4000, warmup = 2000, chains = 4, cores = 4, seed = 1234,
                      file = "output/food_on_weight")
```

```
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     7968     5853
avgfood      -0.03      0.09    -0.21     0.16 1.00     7991     5468

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     1.01      0.07     0.89     1.14 1.00     7678     5661

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).
```

**Summary:** The effect of food appears to be negative, but the confidence interval is really poor. It contains 0, so that says to me that the effect of avg_food on weight is either not important (which doesn't make biological sense), or we are not capturing the effect of food with the priors / model set up.

### Q2.8

**Answer:** We should condition on groupsize, which is in the middle of the pipe from avgfood -> groupsize -> weight

### Q2.9

**Hypothesis:** Increasing the number of foxes will increase competition for resources. Therefore, increasing groupsize would decrease fox weight.

```r
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")
```

```r
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    -0.16     0.15 1.00     6399     5493
avgfood       0.47      0.18     0.10     0.84 1.00     3962     4290
groupsize    -0.57      0.19    -0.93    -0.20 1.00     4145     4524

Further Distributional Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.96      0.06     0.84     1.10 1.00     5389     4998

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).
```

### Q 2.10a

The effects of food: 0.47 (95% CI 0.1-0.84)

The effects of groupsize: -0.57 (95% CI -0.93 - -0.20)

### Q2.10b

The univariate regressions showed no effect of avg food or group size on weight, which biologically did not make sense. When conditioning on groupsize, we see that the effect of group size on weight is strongly negative, and the effect of avg food is strongly positive.

### Q2.10c

Fox territories are a factor of three main variables: area, amount of food present, and group size, and these variables vary as a function of each other. Because group size is a function of the amount of food present, we conditioned the model on this variable to estimate the direct

effect of average food on fox size. When conditioning on group size, the effect of food increases fox weight by 0.47 unit of body weight per unit of food. However, group size decreases fox weight by 0.57 unit weight per individual added to the group.

A univariate model does not account for how different variables change as a function of another, so a univariate model masks the true relationships.