Easy

E1

2,4

E2

$$Diversity_{animal} = \alpha + \beta_{plant} Diversity_{plant} + \beta_{latitude} Latitude$$

E3

I would expect each slope to be negative.

$$TTD = \alpha + \beta_{funding} Funding + \beta_{size} LabSize$$

E4

I think they are all equivalent?

Medium

M1

Simulate a random variable x that is positively correlated with z and y, this will create a spurious correlations between y and z. But after accounting for the affect of x on y, adding z doesn't.

```
n <- 100
# simulate x
x <- rnorm(n)
# simulate y so that it is positively correlated with x
y <- rnorm(n, x)
# simulate z so that it is positively coreelated with x
z <- rnorm(n, x)
# visualize the pairs
pairs(data.frame(x,y,z))
```
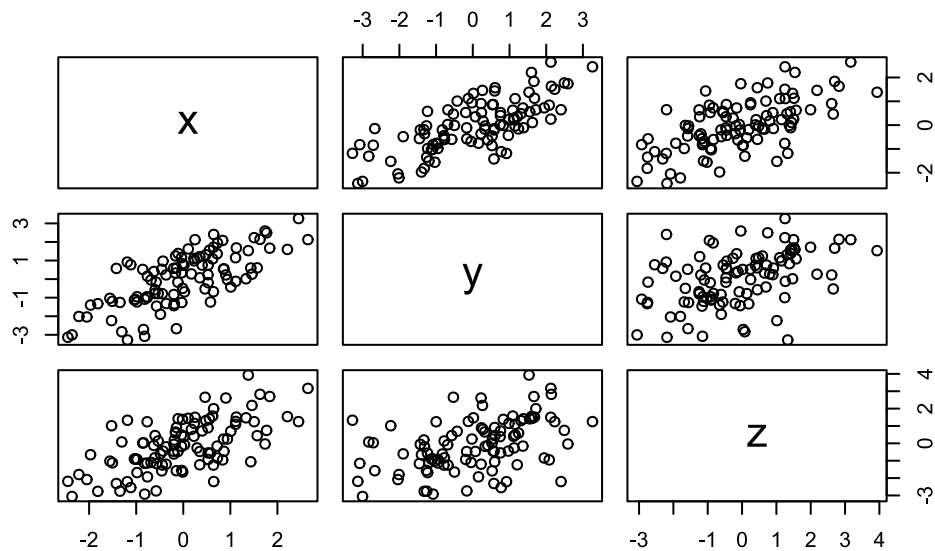
```
# look at the raw correlations
cor(data.frame(x,y,z))
```

```
          x         y         z
x 1.0000000 0.7202251 0.6392192
y 0.7202251 1.0000000 0.4436060
z 0.6392192 0.4436060 1.0000000
```

```
# least squares regression
X <- matrix(c(rep(1, length(y)),x,z), ncol = 3)
solve(t(X) %*% X) %*% (t(X) %*% y)
```

```
            [,1]
[1,]  0.04521925
[2,]  0.99940694
[3,] -0.02787139
```

```
lm(y ~ x + z)
```

```
Call:
lm(formula = y ~ x + z)
```

```
Coefficients:
(Intercept)              x              z
    0.04522        0.99941      -0.02787
```

M2

By simulating a random variable x and a random variable z that is correlated with x you get a positive correlation between x and z, then by simulating a random variable y that is positively correlated with x and negatively correlated with y. This makes no correlation between x and y and a reduced correlation between z and y. By adding them both into the multiple regression, the true values are recovered.

```
set.seed(120414)
# simulate x
x <- rnorm(n)
# simulate z so that it is positively correlated with x
z <- rnorm(n, x)
# simulate y so that it is positively correlated with x but negatively correlated
wity z
y <- rnorm(n, x - z)
# look at pairs plots to visualize
pairs(data.frame(x,z,y))
```
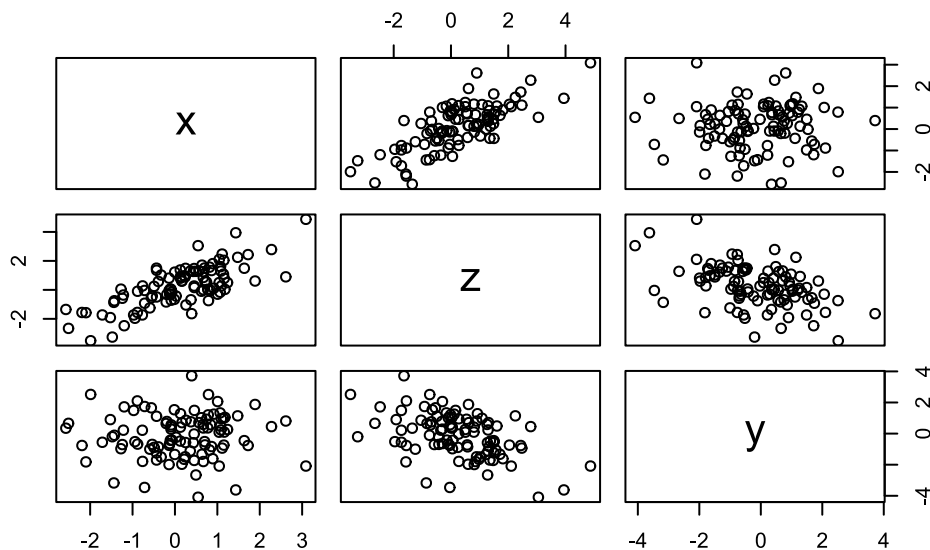
```
# look at the raw correlations
cor(data.frame(x,z,y))
```

```
            x          z          y
x 1.00000000  0.7426733  0.02266509
z 0.74267326  1.0000000 -0.47133835
y 0.02266509 -0.4713383  1.00000000
```

```
# least squares to recover the regression coefficients
X <- matrix(c(rep(1,length(y)), x, z), ncol = 3)
solve(t(X) %*% X) %*% t(X) %*% y
```

```
            [,1]
[1,]  0.004811758
[2,]  1.092155887
[3,] -1.075275811
```

M3

I guess you could try to predict marriage from divorce and age, presumably states with a younger
median age at marriage would have a higher marriage rate and a lower divorce rate would lead
to a higher marriage rate–I think

```
library(rethinking)
library(tidyverse)
library(cmdstanr)
library(bayesplot)

data("WaffleDivorce")
d <- WaffleDivorce

# function to standardize variables
stn <- function(x) (x - mean(x))/sd(x)

X <- matrix(c(rep(1, nrow(d)), stn(d$MedianAgeMarriage), stn(d$Divorce)),
            ncol = 3)
dat <- list(
  N = nrow(d),
  K = ncol(X),
  marriage = stn(d$Marriage),
  X = X
)

mod.3 <- cmdstan_model("stan_models/married_pred.stan")
```

```
Warning in readLines(stan_file): incomplete final line found on
'stan_models/married_pred.stan'
```

```
fit.3 <- mod.3$sample(
  data = dat,
  chains = 4,
  parallel_chains = 4,
  show_message = F
)

fit.3$diagnostic_summary()
```
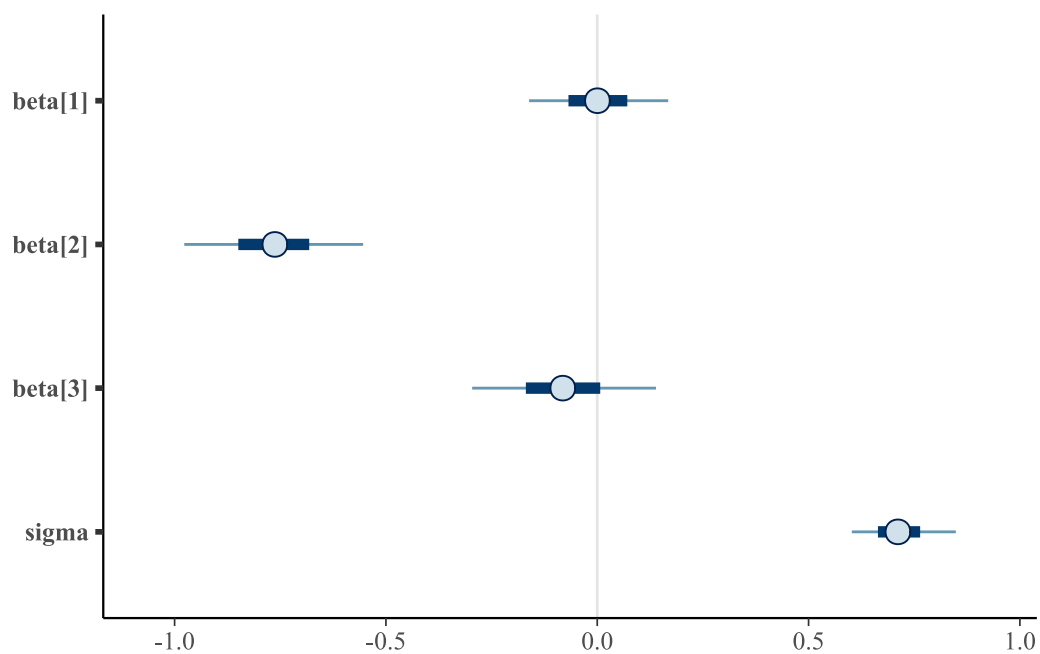
```
$num_divergent
[1] 0 0 0 0

$num_max_treedepth
[1] 0 0 0 0

$ebfmi
[1] 0.9973762 1.0960696 0.9772913 1.0830302
```

```
mcmc_intervals(fit.3$draws(c("beta", "sigma")))
```

This model thinks that median marriage age is a better predictor of marriage rate than divorce rate.

M4

```r
# get the data into order
d <- WaffleDivorce
lds <- read.csv("lds_prop.csv")
lds <- lds %>%
  select(Location = "X.State", lds_prop)
d <- d %>%
  left_join(lds)

dat <- d %>%
  select(Divorce, age = MedianAgeMarriage, mar_rate = Marriage, lds = lds_prop)
%>%
  mutate(divorce = stn(Divorce),
         age = stn(age),
         mar_rate = stn(mar_rate),
         lds = stn(lds))

mod.4 <- cmdstan_model("stan_models/divorce_lds.stan")

# lets try it first without the lds data

# create a design matrix for the covariates, with a column of 1's for the
intercept
X = matrix(c(rep(1, nrow(dat)), dat$age, dat$mar_rate), ncol = 3)

stan_dat <- list(
  N = nrow(dat),
  K = ncol(X),
  X = X,
  divorce = dat$divorce
)

fit.4_nolds <- mod.4$sample(
  data = stan_dat,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)

fit.4_nolds$diagnostic_summary()
```
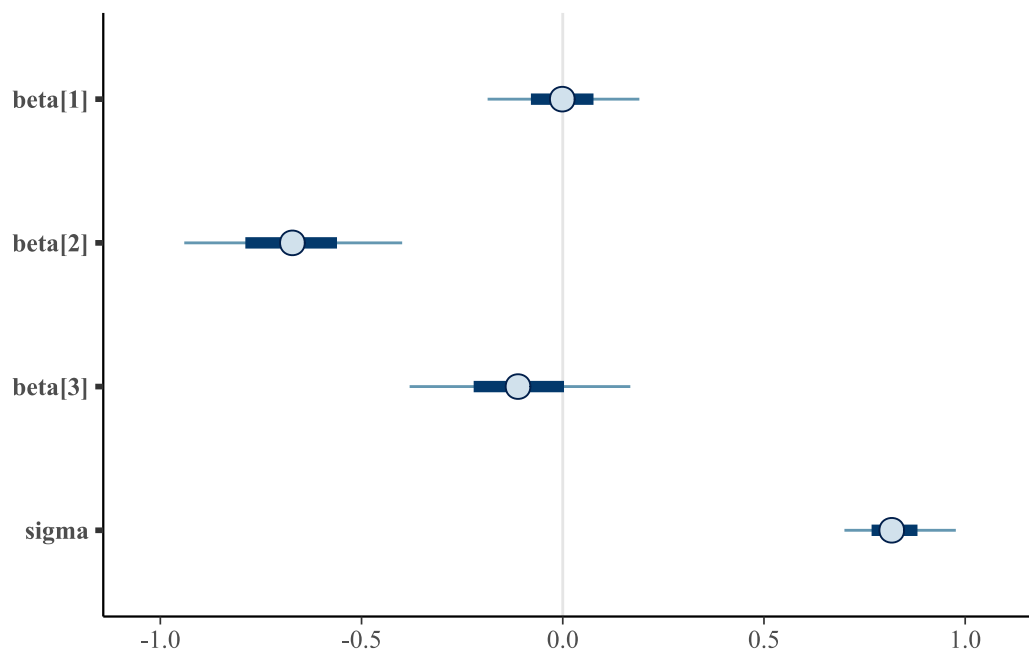
```
$num_divergent
[1] 0 0 0 0
```

```
$num_max_treedepth
[1] 0 0 0 0

$ebfmi
[1] 1.087619 1.091299 1.023643 1.086551
```

```
mcmc_intervals(fit.4_nolds$draws(c("beta", "sigma")))
```



```
# now with the lds data


X = matrix(c(rep(1, nrow(dat)), dat$age, dat$mar_rate, dat$lds), ncol = 4)

stan_dat <- list(
  N = nrow(dat),
  K = ncol(X),
  X = X,
  divorce = dat$divorce
)



fit.4 <- mod.4$sample(
  data = stan_dat,
  chains = 4,
```

```
    parallel_chains = 4,
    show_messages = F
)

fit.4$diagnostic_summary()
```
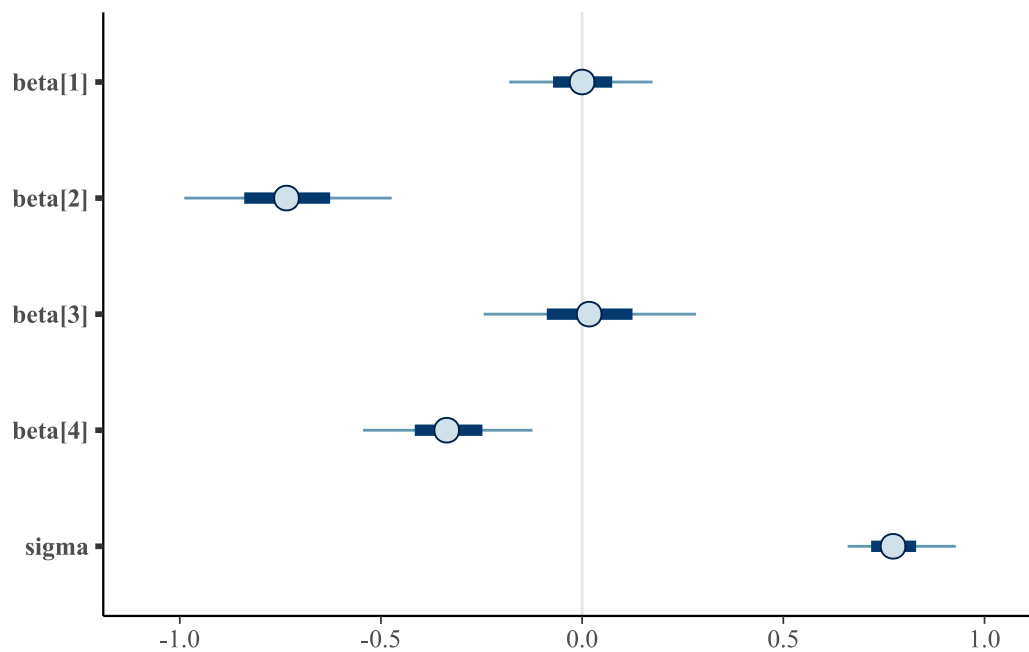
```
$num_divergent
[1] 0 0 0 0

$num_max_treedepth
[1] 0 0 0 0

$ebfmi
[1] 0.9556615 1.0051774 1.0035906 0.9720306
```

```
mcmc_intervals(fit.4$draws(c("beta", "sigma")))
```


M5

$$obesity = \alpha + \beta_{gas}GasPrice + \beta_{eat}RestaurantVisits$$

$$obesity = \alpha + \beta_{gas}GasPrice + \beta_{exercise}ExerciseTime$$

Hard

Make a stan file to fit all these so it is general.

```r
lin_mod <- cmdstan_model("stan_models/linear_mod.stan")
```

H1

```r
data(foxes)

# area size model
X <- matrix(c(rep(1, nrow(foxes)), stn(foxes$area)), ncol = 2)

dat_area <- list(
  N = nrow(foxes),
  K = ncol(X),
  y = stn(foxes$weight),
  X = X
)

fit_area <- lin_mod$sample(
  data = dat_area,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)
```

```
Chain 3 Informational Message: The current Metropolis proposal is about to be
rejected because of the following issue:
```

```
Chain 3 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'C:/Users/jjk06/AppData/Local/Temp/RtmpmOslxs/model-419c7fa0536e.stan',    line
17, column 2 to column 28)
```

```
Chain 3 If this warning occurs sporadically, such as for highly constrained
variable types like covariance matrices, then the sampler is fine,
```

```
Chain 3 but if this warning occurs often then your model may be either severely
ill-conditioned or misspecified.
```

```
Chain 3
```

```
Chain 4 Informational Message: The current Metropolis proposal is about to be
rejected because of the following issue:
```
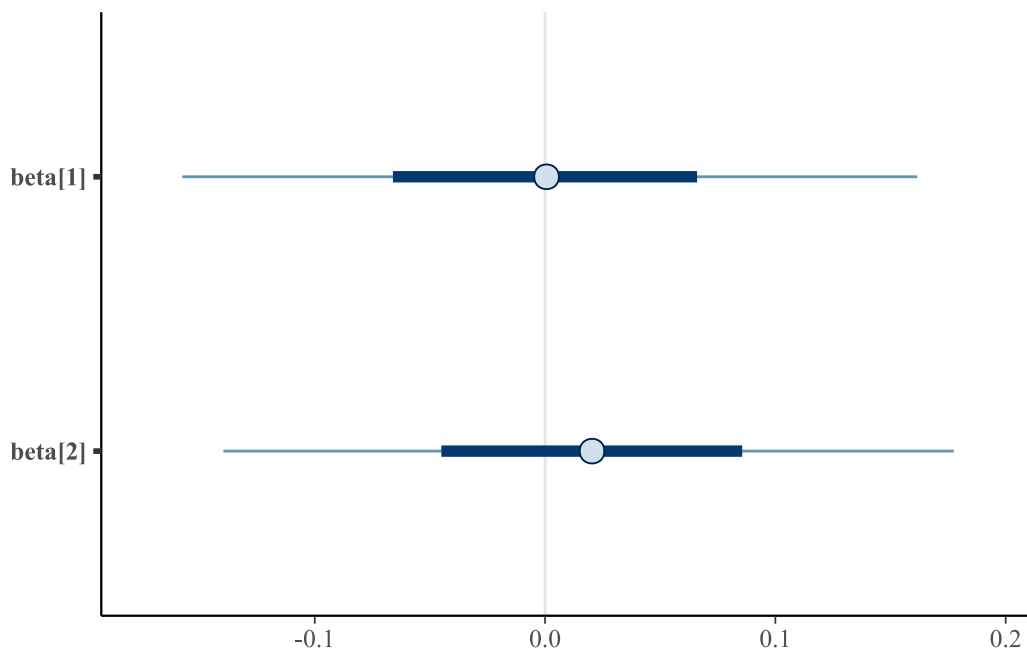
```
Chain 4 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'C:/Users/jjk06/AppData/Local/Temp/RtmpmOslxs/model-419c7fa0536e.stan',    line
17, column 2 to column 28)
```

```
Chain 4 If this warning occurs sporadically, such as for highly constrained
variable types like covariance matrices, then the sampler is fine,
```

```
Chain 4 but if this warning occurs often then your model may be either severely
ill-conditioned or misspecified.
```

```
Chain 4
```

```r
mcmc_intervals(fit_area$draws("beta"))
```



```r
X <- matrix(c(rep(1, nrow(foxes)), stn(foxes$groupsize)), ncol = 2)

dat_group <- list(
  N = nrow(foxes),
  K = ncol(X),
  y = stn(foxes$weight),
  X = X
)
```

```
fit_group <- lin_mod$sample(
  data = dat_group,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)
```

Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of the following issue:

Chain 3 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in 'C:/Users/jjk06/AppData/Local/Temp/RtmpmOslxs/model-419c7fa0536e.stan',   line 17, column 2 to column 28)

Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like covariance matrices, then the sampler is fine,

Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or misspecified.

Chain 3

Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of the following issue:
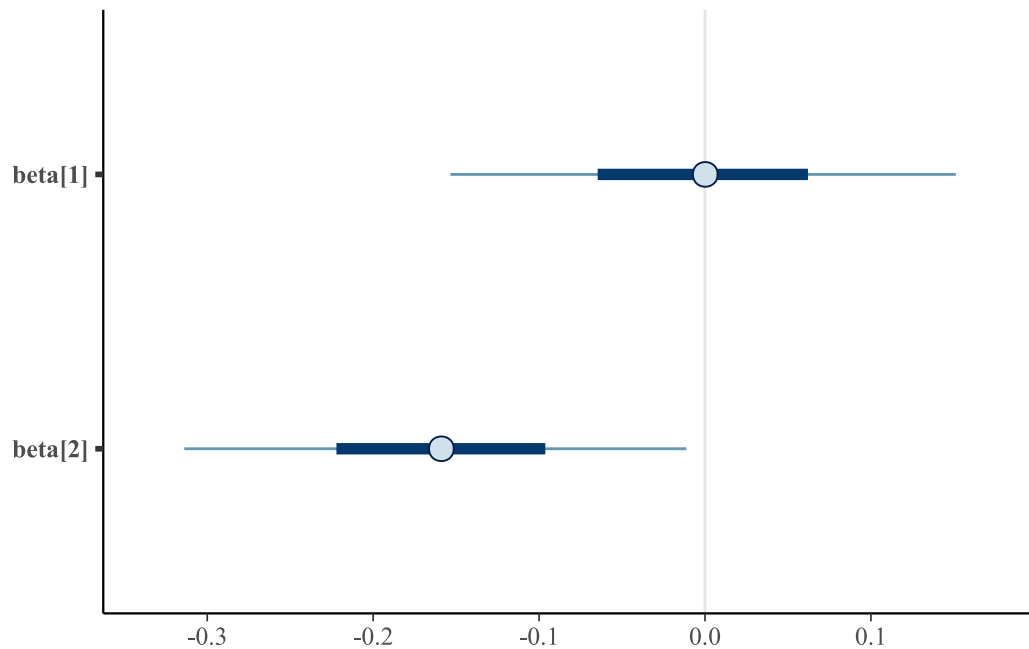
Chain 4 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in 'C:/Users/jjk06/AppData/Local/Temp/RtmpmOslxs/model-419c7fa0536e.stan',   line 17, column 2 to column 28)

Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like covariance matrices, then the sampler is fine,

Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or misspecified.

Chain 4

```
mcmc_intervals(fit_group$draws("beta"))
```


H2

```
X <- matrix(c(rep(1, nrow(foxes)), stn(foxes$area),stn(foxes$groupsize)), ncol
= 3)

dat_both <- list(
  N = nrow(foxes),
  K = ncol(X),
  y = stn(foxes$weight),
  X = X
)

fit_both <- lin_mod$sample(
  data = dat_both,
  chains = 4,
  parallel_chains = 4,
  show_messages = F
)
```

```
Chain 1 Informational Message: The current Metropolis proposal is about to be
rejected because of the following issue:
```

```
Chain 1 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'C:/Users/jjk06/AppData/Local/Temp/RtmpmOslxs/model-419c7fa0536e.stan',    line
17, column 2 to column 28)
```

```
Chain 1 If this warning occurs sporadically, such as for highly constrained
variable types like covariance matrices, then the sampler is fine,
```

```
Chain 1 but if this warning occurs often then your model may be either severely
ill-conditioned or misspecified.
```

```
Chain 1
```

```
Chain 2 Informational Message: The current Metropolis proposal is about to be
rejected because of the following issue:
```

```
Chain 2 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in
'C:/Users/jjk06/AppData/Local/Temp/RtmpmOslxs/model-419c7fa0536e.stan',    line
17, column 2 to column 28)
```

```
Chain 2 If this warning occurs sporadically, such as for highly constrained
variable types like covariance matrices, then the sampler is fine,
```

```
Chain 2 but if this warning occurs often then your model may be either severely
ill-conditioned or misspecified.
```

```
Chain 2
```

```r
mcmc_intervals(fit_both$draws("beta"))
```