Designing and Analyzing Statistical Models

EES 5891-03

Bayesian Statistical Methods

Jonathan Gilligan

Class #8: Tuesday, September 20 2022

More Categories of Confounding

General Principle: Identifiability

- **Identifiable Models:** Each set of *model parameters* makes different predictions
- **Non-Identifiable Models:** For any set of parameters, there are many other sets of parameters that make the same prediction
- Example: Categorical variables
  - x has three possible values: **Architect**, **Baker**, or **Carpenter**, and your regression will connect profession to income.
  - Represent x with two variables I_A and I_B, which are 1 if x has that value, and 0 otherwise. \[\begin{align} \text{Income} &\sim \text{Normal}(\mu, \sigma) \\ \mu &= \alpha + \beta_A I_A + \beta_B I_B \end{align} \]
  - Why don’t we have I_C?

Non-Identifiability

$$\begin{align} \mu &= \alpha + \beta_A I_A + \beta_B I_B + \beta_C I_C \\ \rlap{I_A + I_B + I_C = 1}{\quad} & \\ I_C &= 1 - (I_A + I_B) \\ \mu &= \alpha + \beta_A I_A + \beta_B I_B + \beta_C (1 - (I_A + I_B)) \\ &= \alpha + (\beta_A - \beta_C) I_A + (\beta_B - \beta_C) I_B + \beta_C \\ &= (\alpha + \beta_C) + (\beta_A - \beta_C) I_A + (\beta_B - \beta_C) I_B \end{align}$$

- Now pick any number δ and let $$\begin{align} \alpha' &= \alpha - \delta \\ \beta_A' &= \beta_A + \delta \\ \beta_B' &= \beta_B + \delta \\ \beta_C' &= \beta_C + \delta \end{align}$$ And $$\mu' = \alpha' + \beta_A' I_A + \beta_B' I_B + \beta_C' I_C$$

Non-Identifiability (cont.)

$$\begin{align} \require{cancel} \mu &= \alpha + \beta_A I_A + \beta_B I_B + \beta_C I_C \\ &= (\alpha + \beta_C) + (\beta_A - \beta_C) I_A + (\beta_B - \beta_C) I_B \\ \mu' &= \alpha' + \beta_A' I_A + \beta_B' I_B + \beta_C' I_C \\ &= (\alpha' + \beta_C) + (\beta_A' - \beta_C') I_A + (\beta_B' - \beta_C') I_B \\ &= [(\alpha - \delta) + (\beta_C + \delta)] + [(\beta_A + \delta) - (\beta_C + \delta)] I_A + [(\beta_B + \delta) - (\beta_C + \delta)] I_B \\ &= [(\alpha - \cancel{\delta}) + (\beta_C + \cancel{\delta})] + [(\beta_A + \cancel{\delta}) - (\beta_C + \cancel{\delta})] I_A + [(\beta_B + \cancel{\delta}) - (\beta_C + \cancel{\delta})] I_B \\ &= (\alpha + \beta_C) + (\beta_A - \beta_C) I_A + (\beta_B - \beta_C) I_B \\ &= \mu \end{align}$$

- So for any δ, $\mu' = \mu$.
    - This means that there isn’t a **best** set of values for α, β_A, β_B, and β_C.
    - The problem is if you know I_A and I_B, then you also know I_C.
    - If you don’t have an I_C variable, then this problem doesn’t come up.
- There should be one fewer indicator variables than there are levels of the category variable.

Worked Example

- Pick values: \(\alpha = 1\), \(\beta_A = 2\), \(\beta_B = 3\), \(\beta_C = 4\)
- \(\delta = 0.5\)
- Alternate values: \(\alpha' = 0.5\), \(\beta_A = 2.5\), \(\beta_B = 3.5\), \(\beta_C = 4.5\) \[\begin{align} \require{cancel} \mu &= 1 + 2 I_A + 3 I_B + 4 I_C \\ &= (1 + 4) + (2 - 4) I_A + (3 - 4) I_B \\ &= 5 - 2 I_A - 1 I_B \\ \mu' &= 0.5 + 2.5 I_A + 3.5 I_B + 4.5 I_C \\ &= (0.5 + 4.5) + (2.5 - 4.5) I_A + (3.5 - 4.5) I_B \\ &= 5 - 2 I_A - 1 I_B \\ &= \mu \end{align} \]

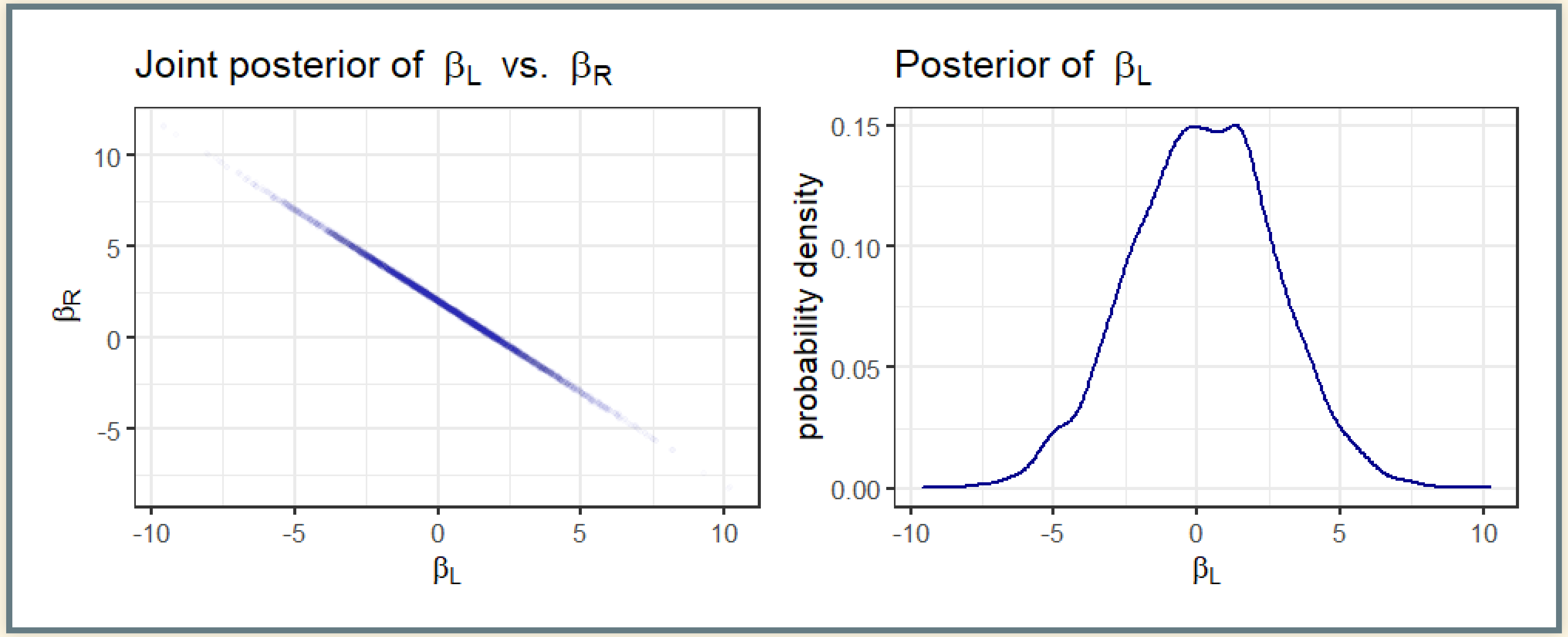
Multicollinearity

Multicollinearity

- Height versus length of legs: \[\begin{align} H &\sim \text{Normal}(\mu, \sigma) \\ \mu &= \alpha + \beta_R R + \beta_L L, \end{align} \] where
  - *H* is the person’s height,
  - *R* is the length of the right leg,
  - *L* is the length of the left leg.
- The legs don’t have identical length, but they are highly correlated.
- This creates a problem of identifiability:
  - Start with \(\beta_L\) and \(\beta_R\),
    - then for some number \(\delta\), consider
      - \(\beta_L' = \beta_L + \delta\)
      - \(\beta_R' = \beta_R - \delta\)
  - On average \(L = R\), so \(\mu' = \mu\).
    - \(\beta_L\) and \(\beta_R\) are not identifiable.

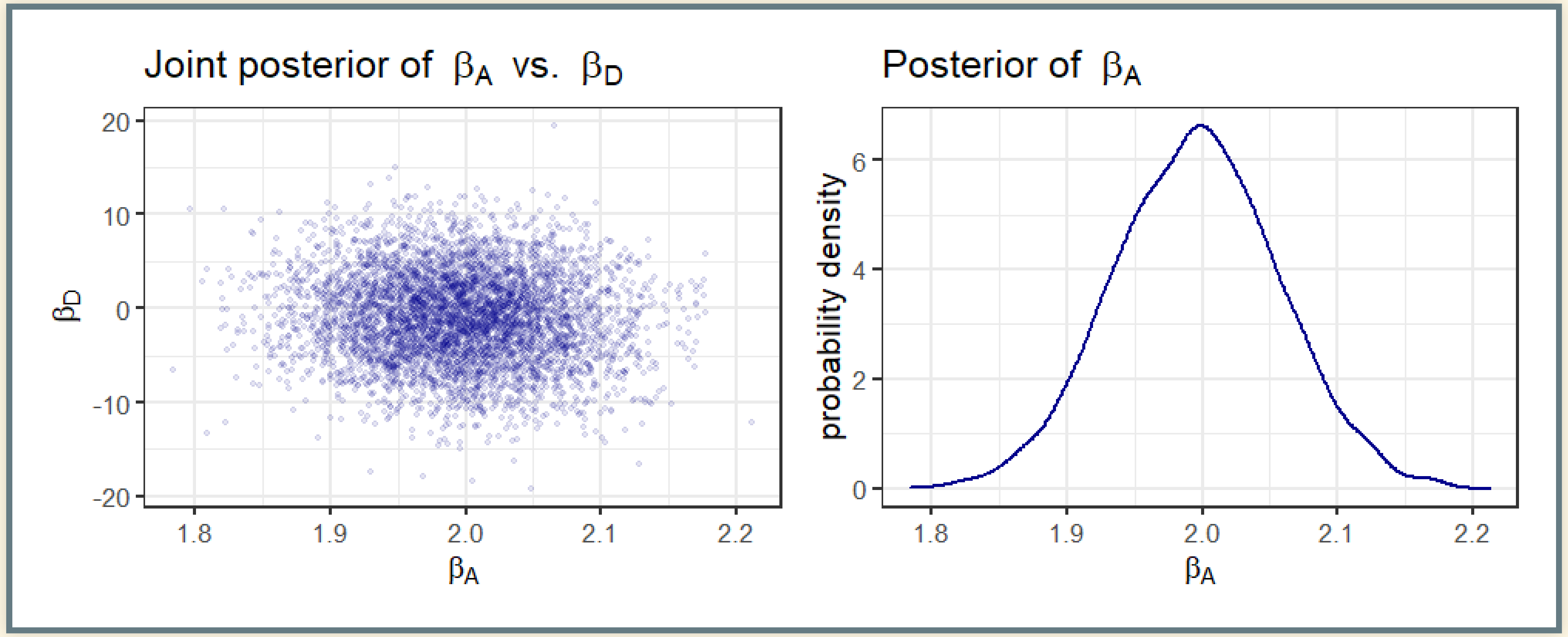
Does Multicollinearity Matter?

- McElreath says it doesn’t matter for model predictions
  - Only matters for interpreting model.
  - Large uncertainty in posteriors for parameters when considered,
    - Because many values of \(\beta_L\) and \(\beta_R\) are just as probable.
  - The *joint posterior* for \(\beta_L\) and \(\beta_R\) is very narrow.



Another perspective

- Multicollinearity can make computational analysis difficult
- One response:
  - Define new variables:
    - \(A = \text{average} = (L + R) / 2\)
    - \(D = \text{difference} = (L - R) / 2\)
    - \(L = A + D\), \(R = A - D\).

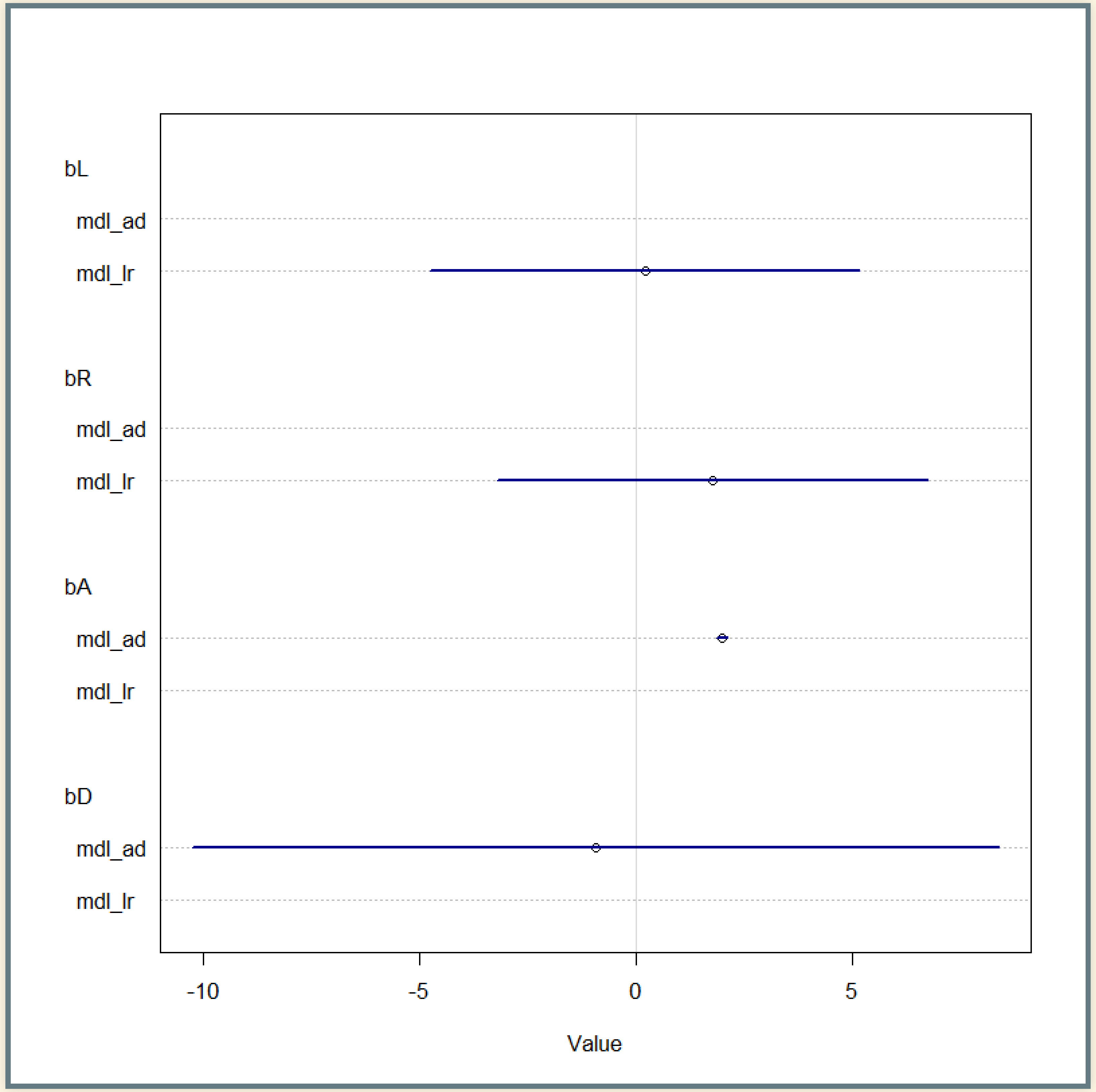


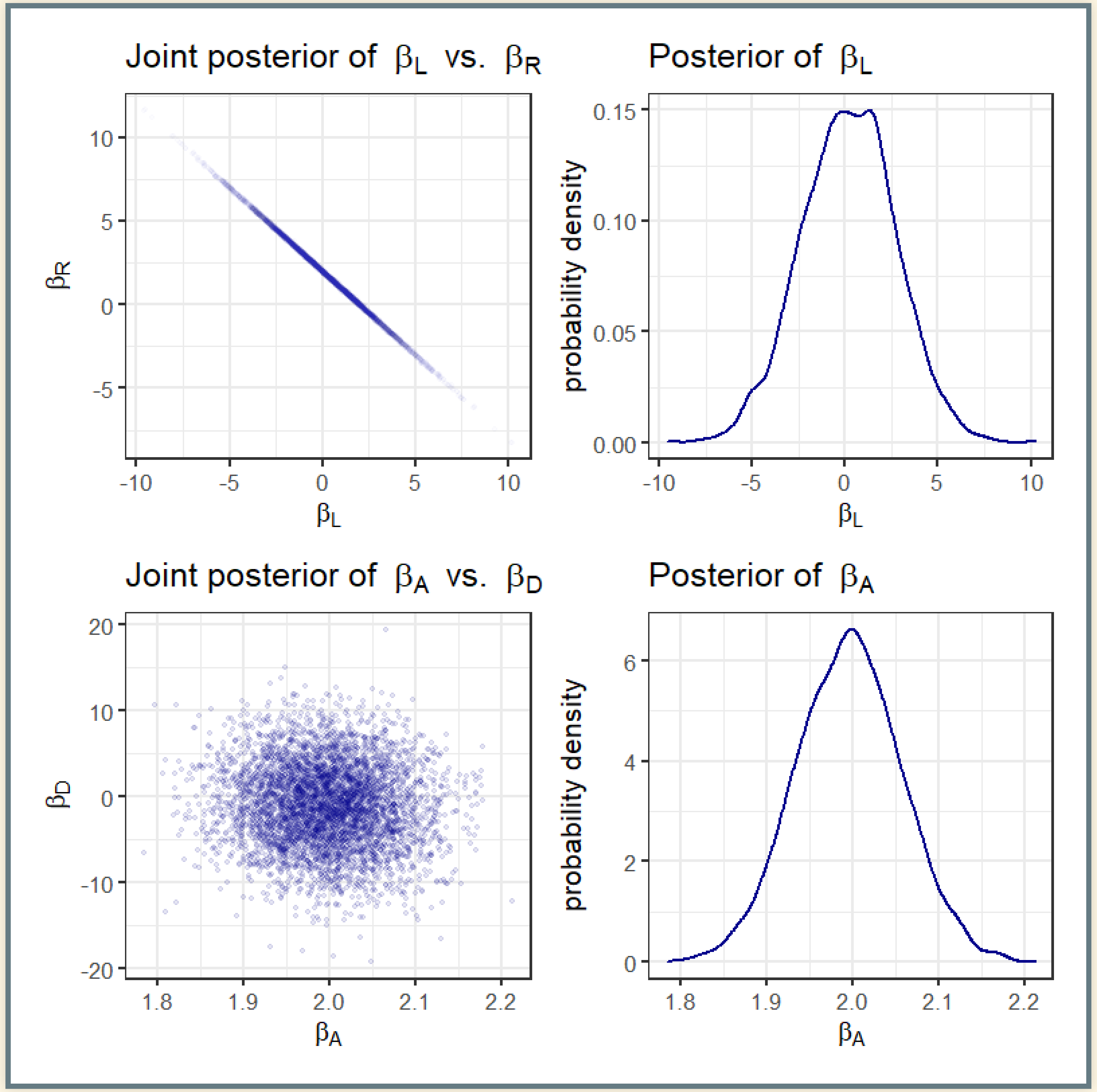
Joint posterior of β_A vs. β_D

Posterior of β_A

Summary

- Note how the different scales for \(\beta_A\) vs. \(\beta_L\).





Multicollinearity with Milk Data

Multicollinearity with Milk Data

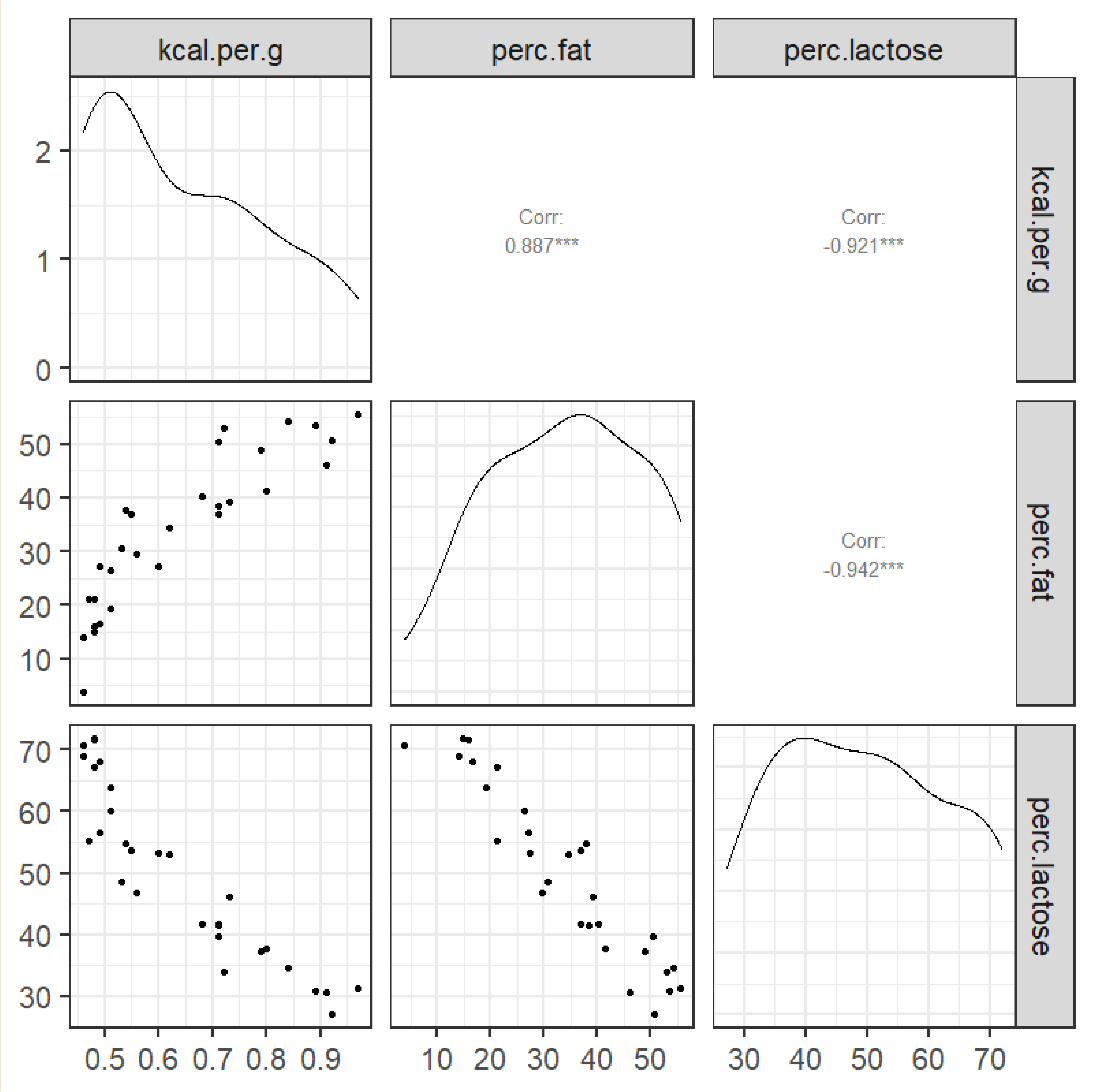
- Calories come from fat and sugar (lactose):

```
data(milk)
d <- milk
d$K <- standardize( d$kcal.per.g )
d$F <- standardize( d$perc.fat )
d$L <- standardize( d$perc.lactose )
```

  - Make a pairwise correlation plot

```
library(tidyverse)
library(GGally)

d %>% select(kcal.per.g, perc.fat, perc.lactose) %>%
  ggpairs()
```

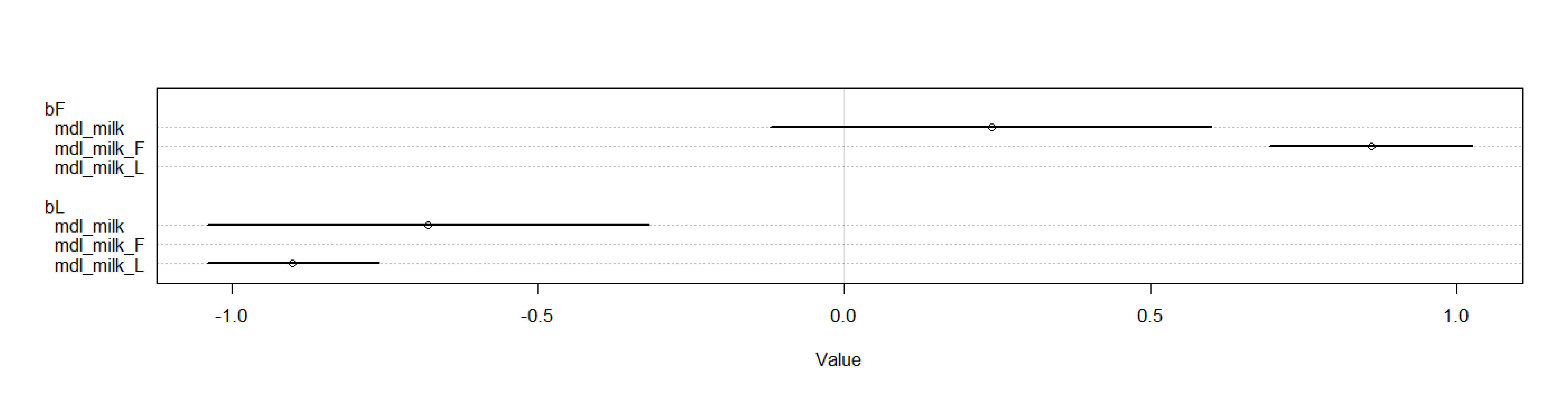


Making a model

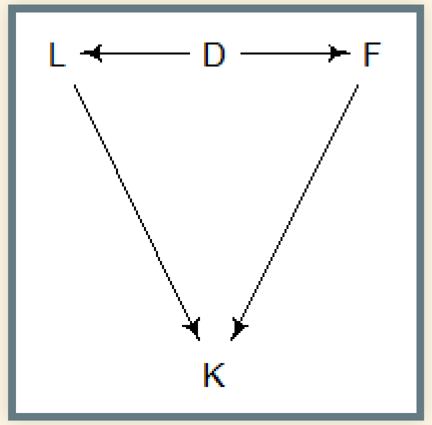
```
mdl_milk <- quap(
    alist(
        K ~ dnorm(mu, sigma),
        mu <- a + bF * F + bL * L,
        a ~ dnorm(0, 0.2),
        bF ~ dnorm(0, 0.5),
        bL ~ dnorm(0, 0.5),
        sigma ~ dexp(1)
    ), data=d )

precis_show(precis(mdl_milk, digits = 2))
```

```
##          mean   sd  5.5% 94.5%
## a        0.00 0.07 -0.11  0.11
## bF       0.24 0.18 -0.05  0.54
## bL      -0.68 0.18 -0.97 -0.38
## sigma    0.38 0.05  0.30  0.46
```



Explaining the multicollinearity



- Knowledge of biology
- Density D is important
  - Frequent nursing: watery, low-energy milk, high in sugar (lactose)
  - Infrequent nursing: rich, dense, high-energy milk, high in fat

Post-Treatment Bias

Anti-Fungal Treatment Experiment

- You do an experiment
  - Divide plants in 2 groups
    - Apply anti-fungal treatment to one group (\(T = 1\))
    - The other is a control (\(T = 0\))
    - Observe whether there is fungus after treatment (\(F\))
    - Compare height before treatment (\(H_0\)) to height some time after treatment (\(H_1\)).
      - Growth rate \(p \ge 0\) unless fungus is very bad.

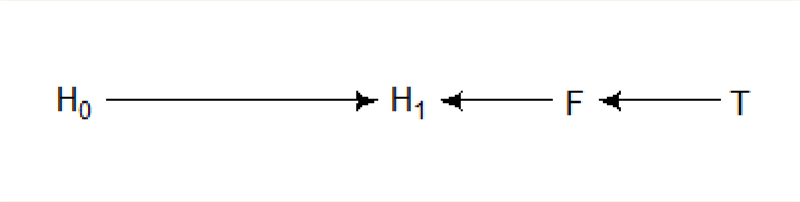
```
mdl_fungus <- quap(
    alist(
        H1 ~ dnorm(mu, sigma),
        mu <- H0 * p,
        # p is growth rate
        p <- a + bT * T + bF * F,
        a ~ dlnorm(0, 0.2),
        bT ~ dnorm(0, 0.5),
        bF ~ dnorm(0, 0.5),
        sigma ~ dexp(1)
    ), data=d)
precis_show(precis(mdl_fungus, digits = 2))
```

```
##         mean   sd  5.5% 94.5%
## a       1.48 0.02  1.44  1.52
## bT      0.00 0.03 -0.05  0.05
## bF     -0.27 0.04 -0.33 -0.21
## sigma   1.41 0.10  1.25  1.57
```

- Why doesn’t the treatment have an effect?
  - `mean(bT)` = 0.

Understanding the problem

- Fungus is the big thing that affects the plants' growth
- Treatment affects fungus.
  - Doesn't affect plants directly
  - Doesn't always eliminate all fungus
- Fungus is a better predictor
  - But we don't know how bad fungus will be until *after* we treat.
- DAG



```
## Implied Conditional Independencies
```

```
## F _||_ H_0
## H_0 _||_ T
## H_1 _||_ T | F
```

```
mdl_fungus <- quap(
    alist(
        H1 ~ dnorm(mu, sigma),
        mu <- H0 * p,
        # p is growth rate
        p <- a + bT * T + bF *
         F,
        a ~ dlnorm(0, 0.2),
        bT ~ dnorm(0, 0.5),
        bF ~ dnorm(0, 0.5),
        sigma ~ dexp(1)
    ), data=d)
```

A Better Model

- DAG
- Conditioning on *F* induces a *D-separation* (*directional* separation) between *T* and *H1*.
- Remove fungus data from the model.

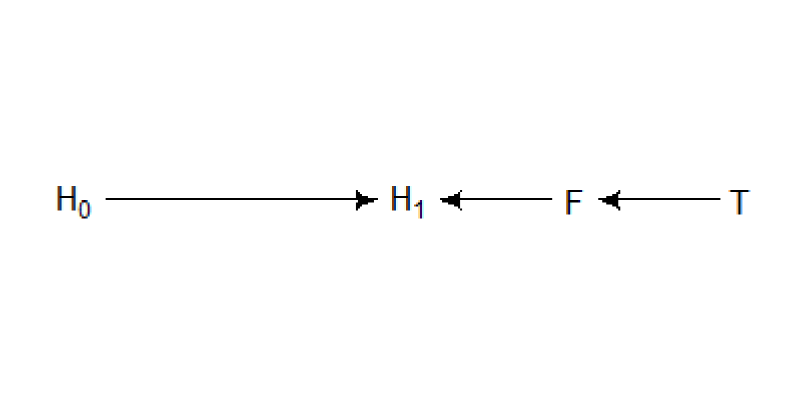
```r
mdl_fungus_2 <- quap(
    alist(
        h1 ~ dnorm( mu , sigma ),
        mu <- h0 * p,
        p <- a + bt*treatment,
        a ~ dlnorm( 0 , 0.2 ),
        bt ~ dnorm( 0 , 0.5 ),
        sigma ~ dexp( 1 )
    ), data=d )

precis_show(precis(mdl_fungus_2, digits = 2))
```

```
##       mean   sd 5.5% 94.5%
## a     1.38 0.03 1.34  1.42
## bt    0.08 0.03 0.03  0.14
## sigma 1.75 0.12 1.55  1.94
```

Other Post-Treatment Bias Problems

- Suppose we have this DAG:

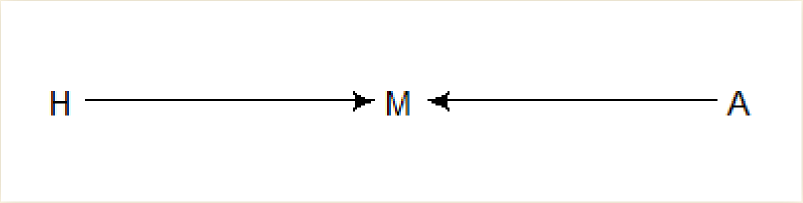


- Fungus does not influence plant growth.
- Moisture influences both plant growth and fungus
- Fitting our original model falsely implies that treatment benefits plants.
- This is a kind of *collider* effect.

Collider Bias

Happiness and Age

- Do people get happier as they get older?
- Suppose:
    - Everyone's happiness is something they are born with and it doesn't change.
    - Happier people are more likely to get married
    - Older people are more likely to be married.
    - DAG:



This diagram is a **collider**: Causal paths from *H* and *A collide* at *M*

Analyze Happiness Data

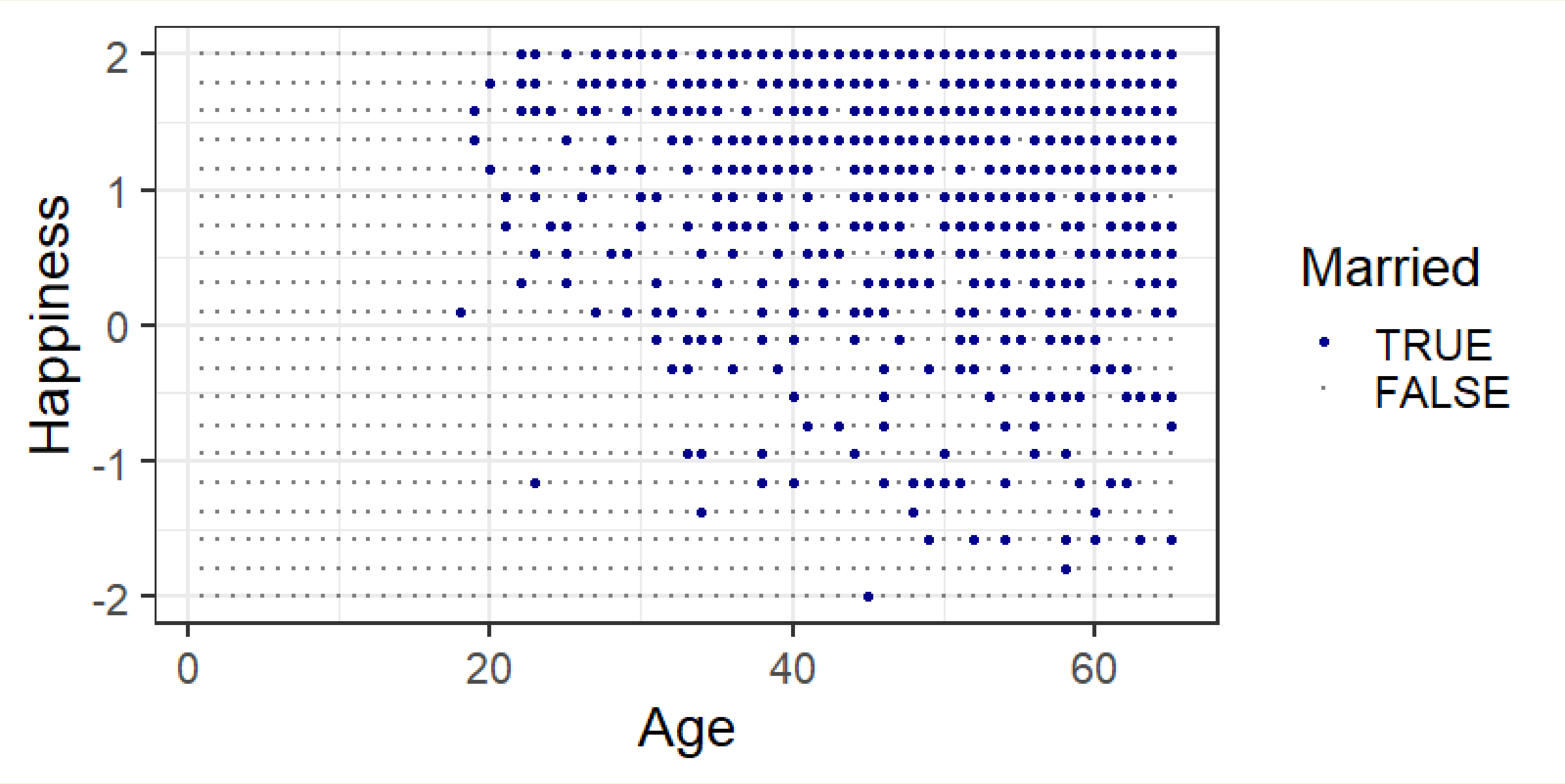
- Load data

```
d <- sim_happiness( seed=1977 , N_years=1000 )
```

- Look for an association between *age* and *happiness*.
  - We suspect that the relationship between age and happiness may be different for married people, so we include marriage as a variable.
- Clean the data: Select adults and convert age to a variable that goes from 0 to 1, and create a marriage index:

```
d2 <- d[ d$age>17 , ] # only adults
d2$A <- ( d2$age - 18 ) / ( 65 - 18 )
d2$mid <- d2$married + 1
```

- The model says that people become unhappy as they get older



```
mdl_happy <- quap(
  alist(
    happiness ~ dnorm(mu, sigma),
    mu <- a[mid] + bA * A,
    a[mid] ~ dnorm(0, 1),
    bA ~ dnorm(0, 2),
    sigma ~ dexp(1)
  ), data=d2)
precis_show(precis(mdl_happy, digits = 2))
```

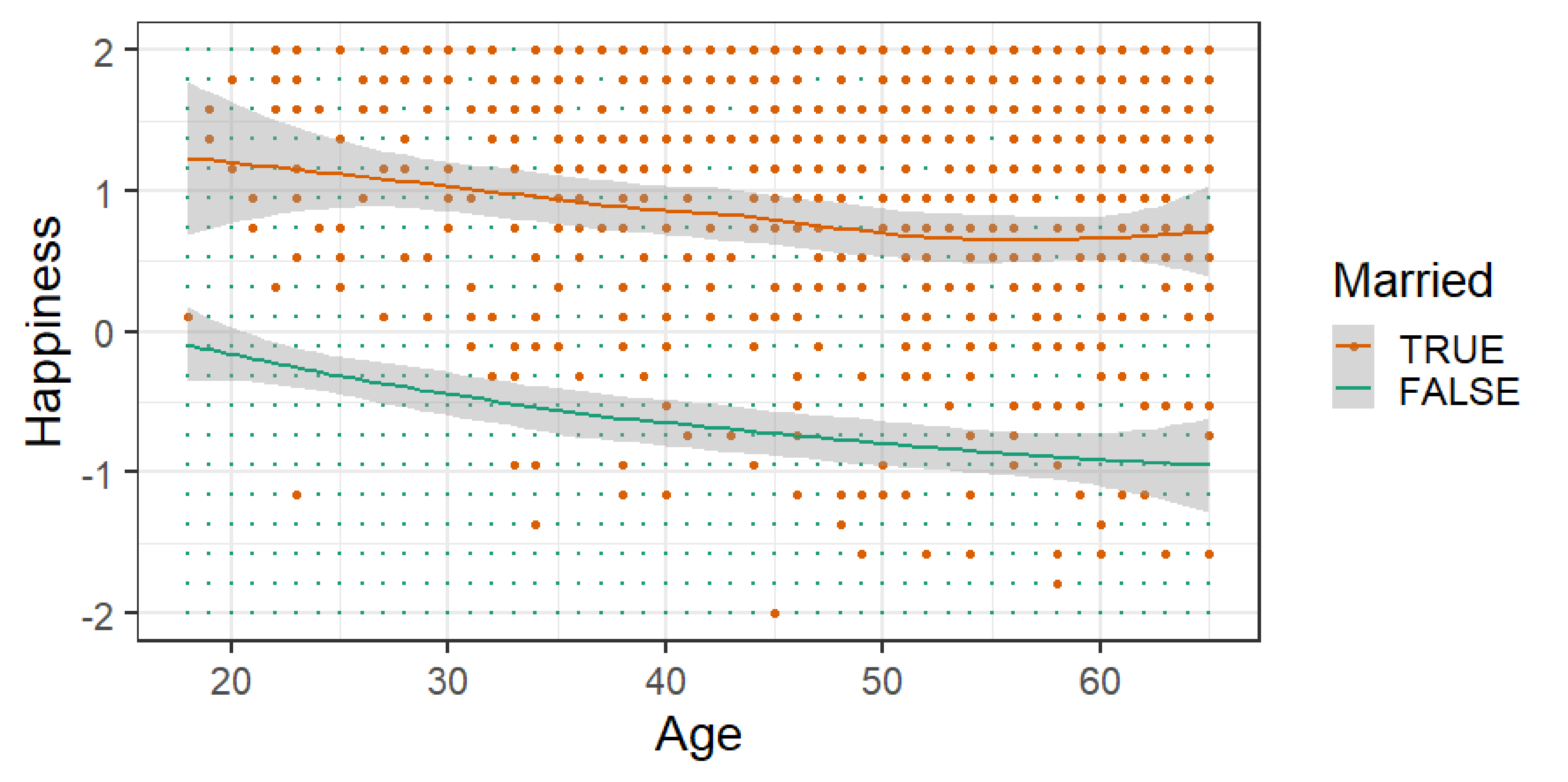
```
##          mean   sd  5.5% 94.5%
## bA      -0.75 0.11 -0.93 -0.57
## sigma    0.99 0.02  0.95  1.03
```

A different Model

- Try a different model that does not control for marriage.
- This model shows no association between age and happiness.
- What happened?
- Consider married people:
    - Older people are more likely to get married
    - Happier people are more likely to get married
    - Happy people get married younger
    - Unhappy people get married older
    - Thus, among married people, younger people are happier, and older ones are unhappier.
- Consider single people
    - As people age, happier ones marry,
    - So the older someone is, if they are still single, they're more likely to be unhappy.

```
mdl_happy_2 <- quap(
    alist(
        happiness ~ dnorm( mu , sigma ),
        mu <- a + bA*A,
        a ~ dnorm( 0 , 1 ),
        bA ~ dnorm( 0 , 2 ),
        sigma ~ dexp(1)
    ) , data=d2 )
precis_show(precis(mdl_happy_2, digits = 2))
```

```
##       mean   sd  5.5% 94.5%
## a     0.00 0.08 -0.12  0.12
## bA    0.00 0.13 -0.21  0.21
## sigma 1.21 0.03  1.17  1.26
```

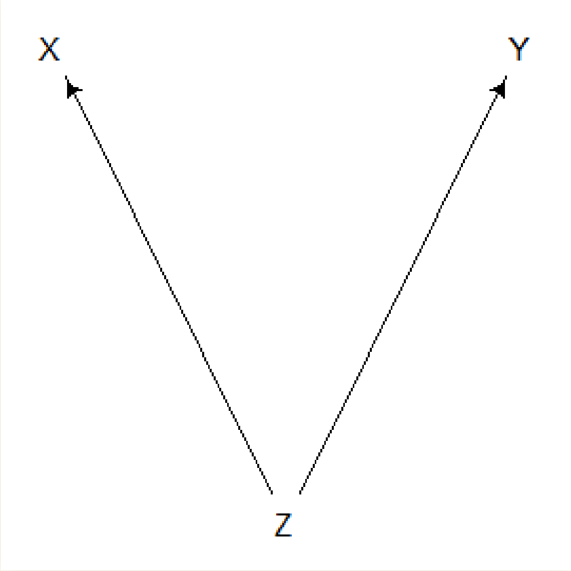


Categories of Confounding Relationships

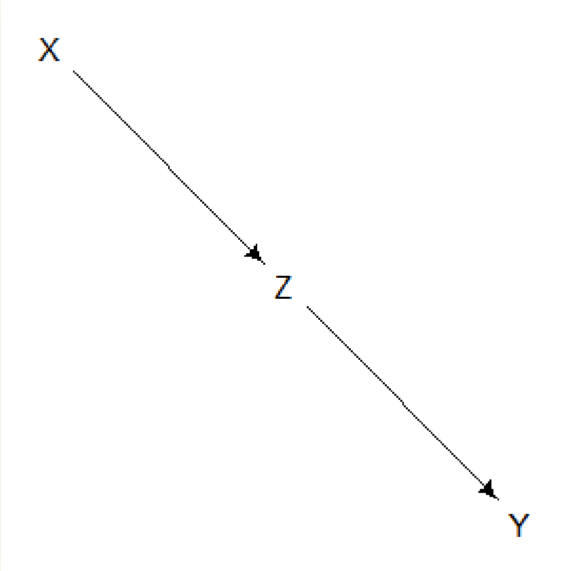
Categories of Confounding Relationships

- There are four major categories of confounding relationships:

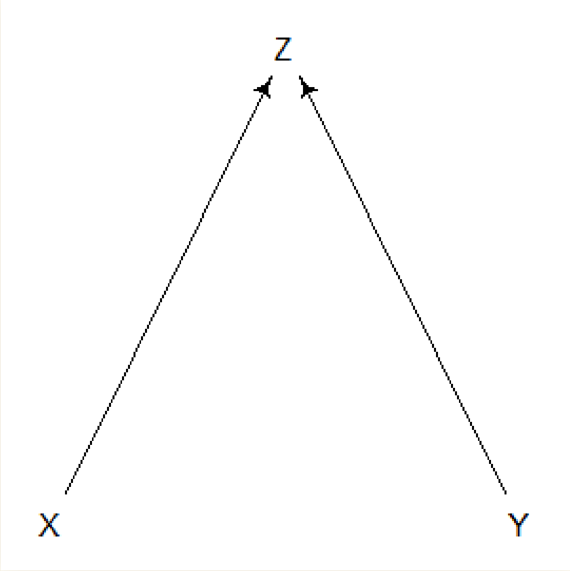
1. Fork



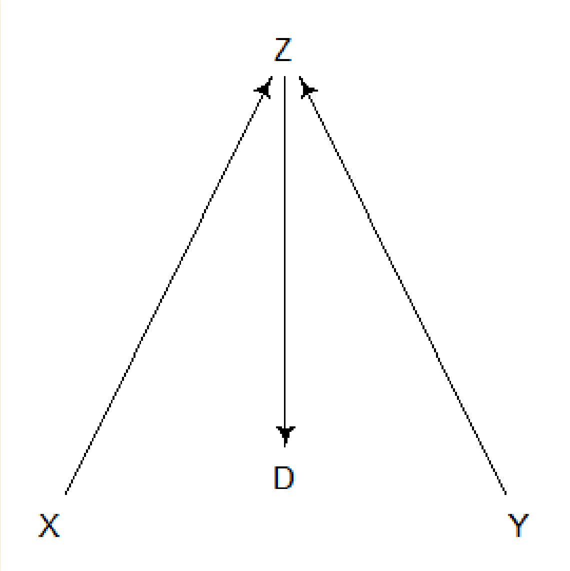
2. Pipe



3. Collider



4. Descendant



- All causal DAGs are build of combinations of these four patterns.