

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.
$$\text{Income} \sim \text{Normal}(\mu, \sigma) \quad \mu = \alpha + \beta_A I_A + \beta_B I_B$$
 - Why don't we have I_C ?

Non-Identifiability

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

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

Non-Identifiability (cont.)

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

- 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: $\alpha' = 0.5$, $\beta_A = 2.5$, $\beta_B = 3.5$, $\beta_C = 4.5$ $\left[\begin{array}{l} \mu = 1 + 2 I_A + 3 I_B + 4 I_C \\ \mu' = 0.5 + 2.5 I_A + 3.5 I_B + 4.5 I_C \end{array} \right]$
 $\begin{array}{l} \mu = (1 + 4) + (2 - 4) I_A + (3 - 4) I_B \\ \mu' = (0.5 + 4.5) + (2.5 - 4.5) I_A + (3.5 - 4.5) I_B \end{array}$ $\Rightarrow \mu = 5 - 2 I_A - 1 I_B$ $\Rightarrow \mu' = 5 - 2 I_A - 1 I_B$

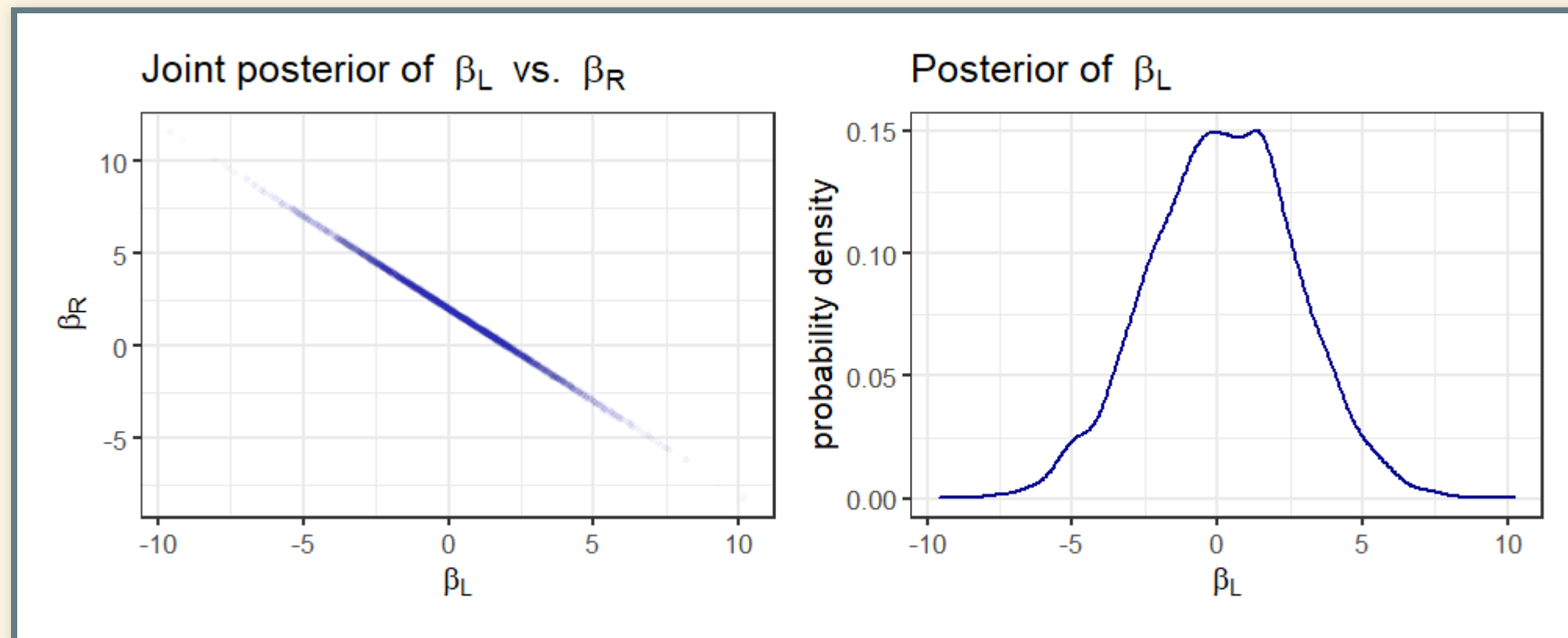
Multicollinearity

Multicollinearity

- Height versus length of legs: $H \sim \text{Normal}(\mu, \sigma)$ where $H = \alpha + \beta_R R + \beta_L L$, 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 (β_L) and (β_R) ,
 - then for some number (δ) , consider
 - $(\beta'_L = \beta_L + \delta)$
 - $(\beta'_R = \beta_R - \delta)$
 - On average $(L = R)$, so $(\mu' = \mu)$.
 - (β_L) and (β_R) are not identifiable.

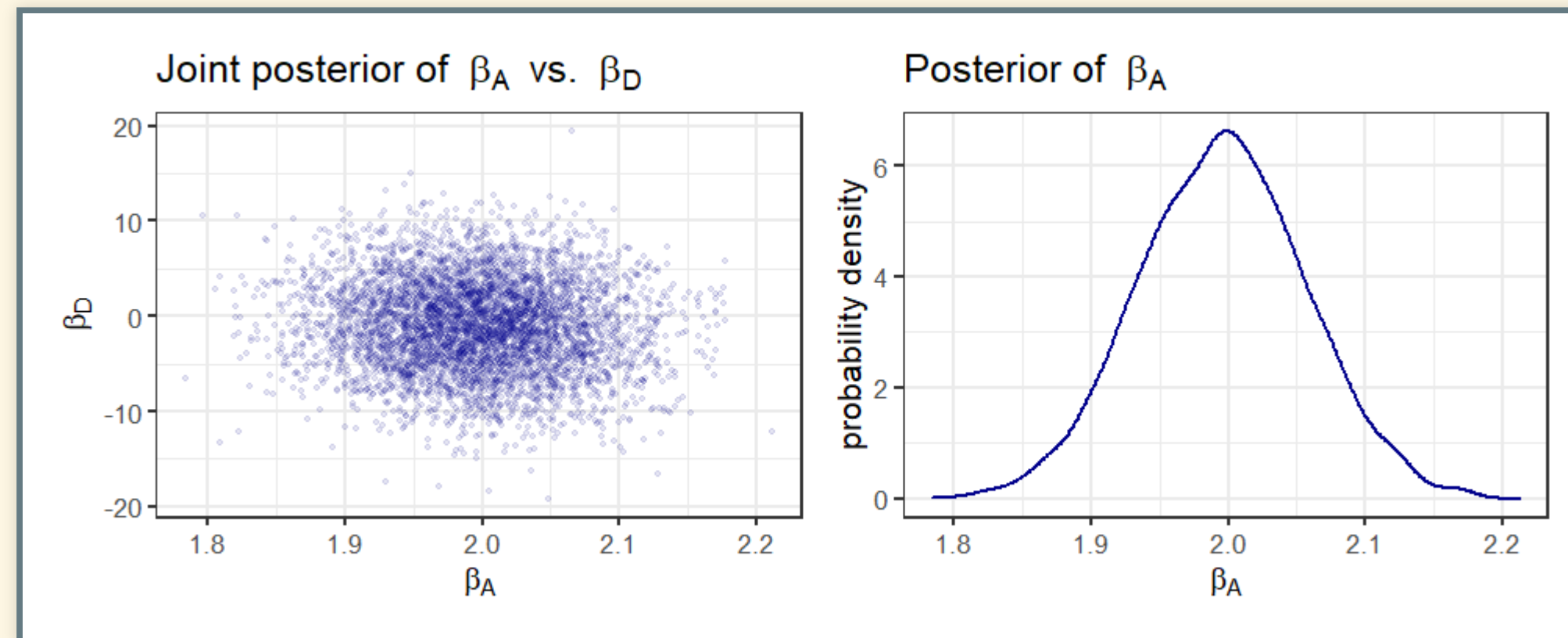
Does Multicollinearity Matter?

- 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 β_L and β_R are just as probable.
 - The *joint posterior* for β_L and β_R is very narrow.



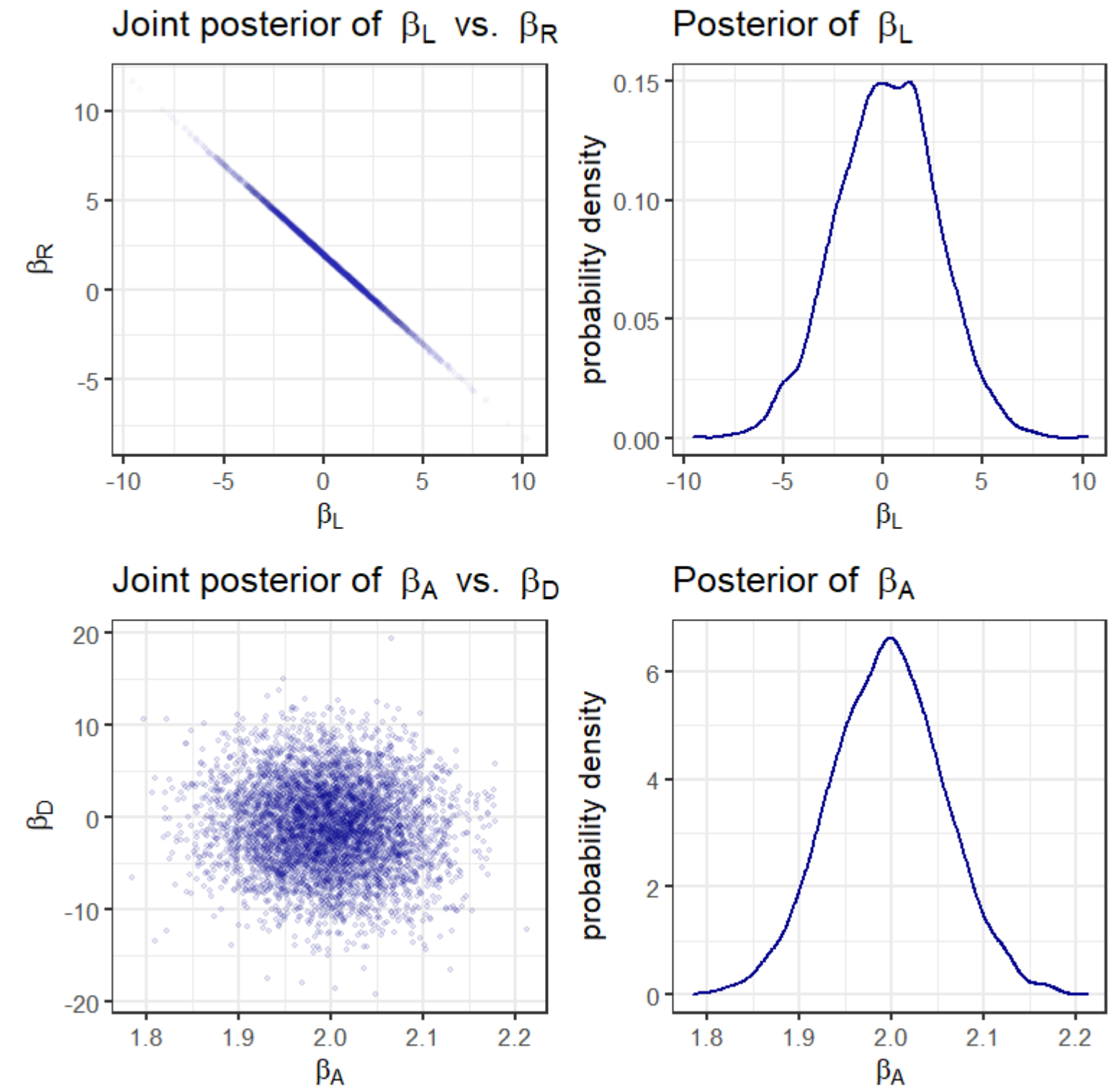
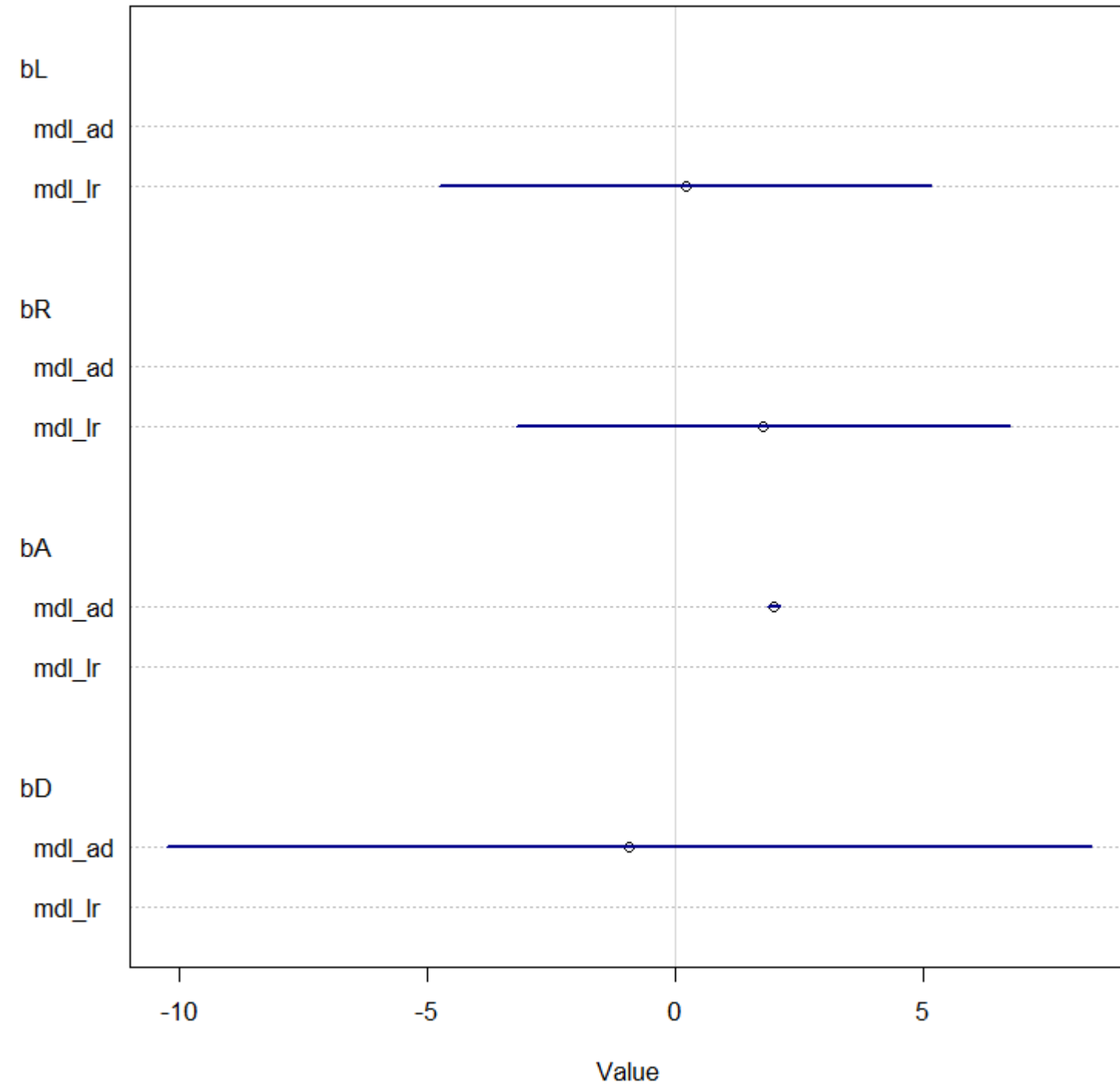
Another perspective

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



Summary

- Note how the different scales for β_A vs. β_L .



Multicollinearity with Milk Data

Multicollinearity with Milk Data

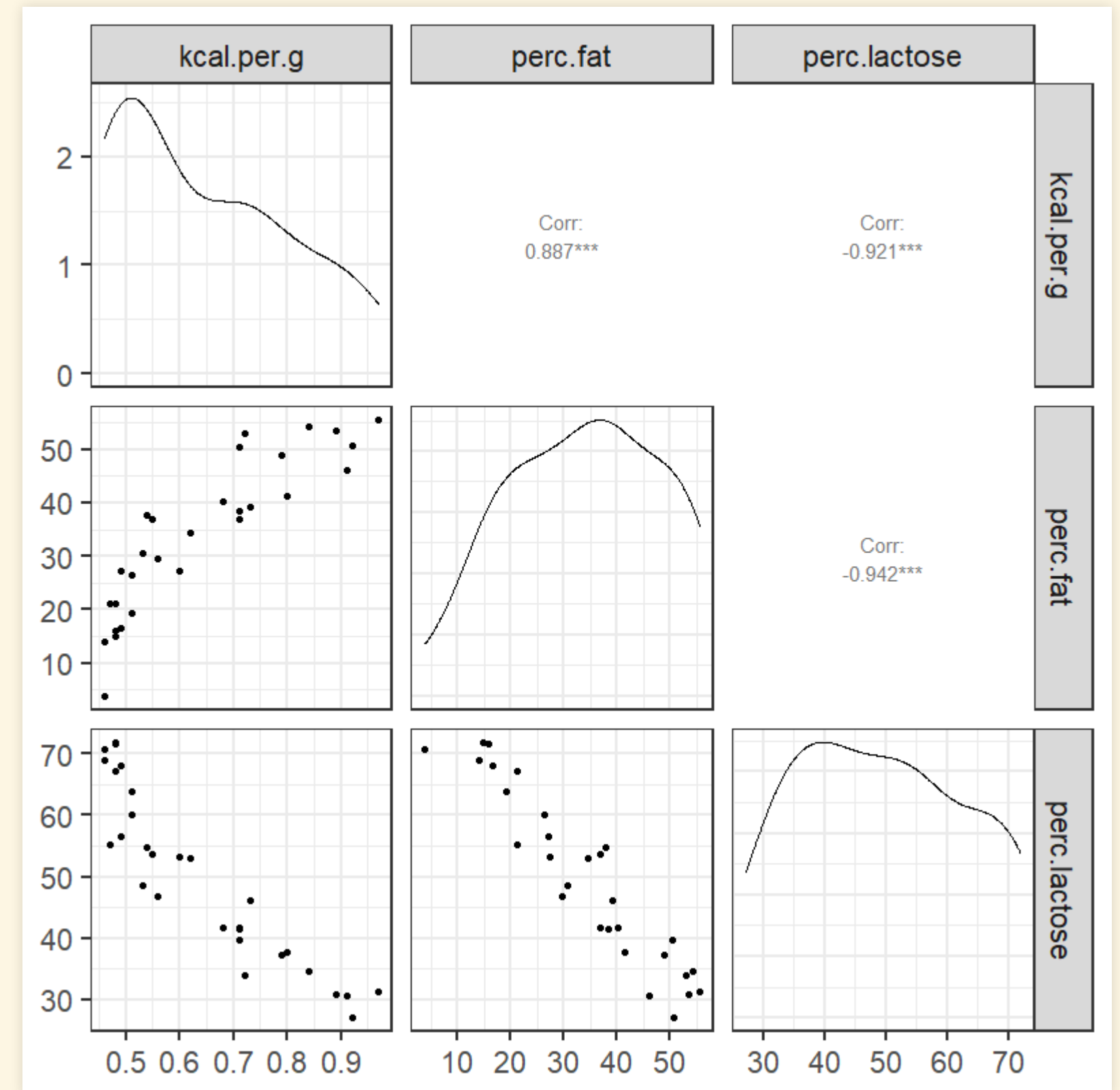
- Calories come from fat and sugar (lactose):

```
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)

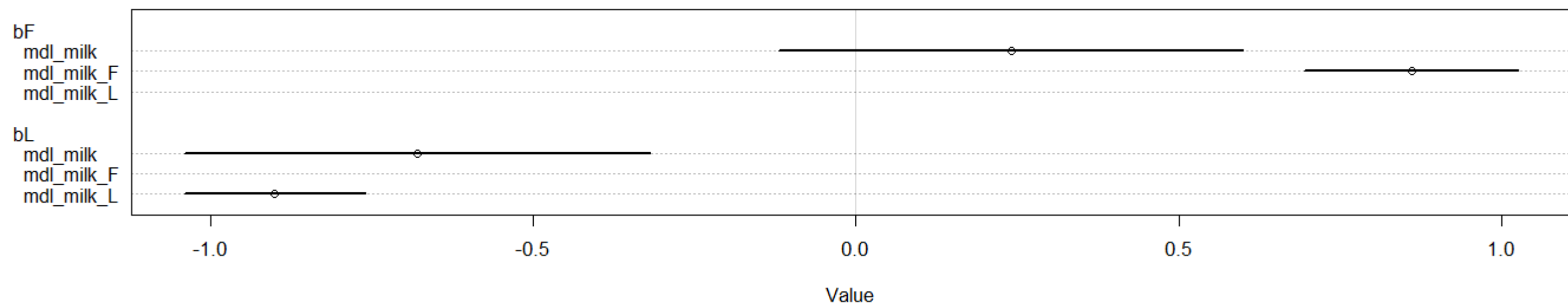
d %>% select(kcal.per.g, perc.fat, perc.lactose) %>%
  ggpairs()
```



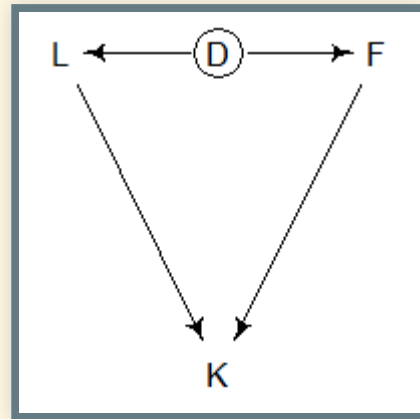
Making a model

```
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 ($p \geq 0$) 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)

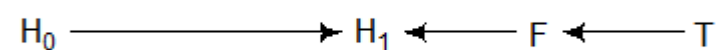
precis_show(precis(mdl_fungus, digits = 2))
```

##		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?
 - $\text{mean}(bT) = 0$.

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



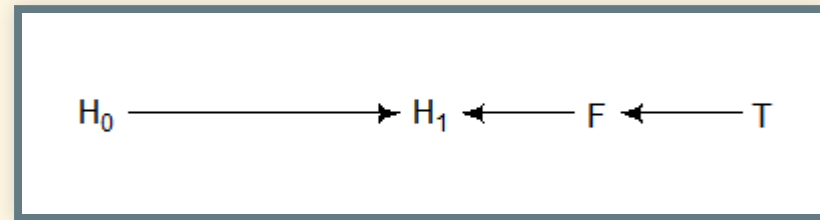
```
## Implied Conditional Independencies
```

```
## 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



```
## Implied Conditional  
Independencies
```

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

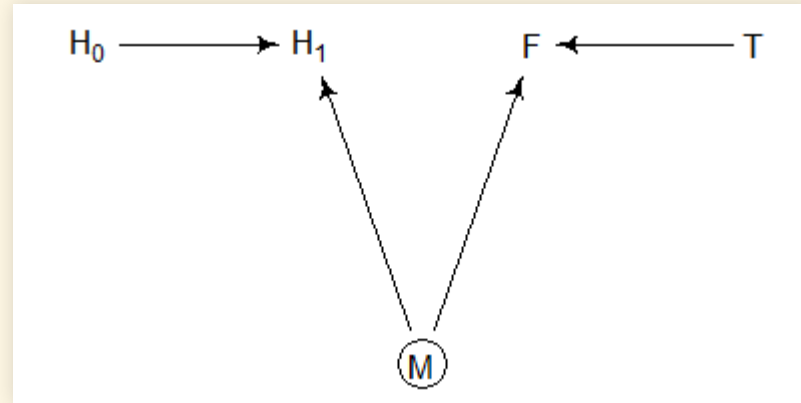
- Conditioning on F induces a *D-separation* (*directional* separation) between T and $H1$.
- Remove fungus data from the model.

```
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:

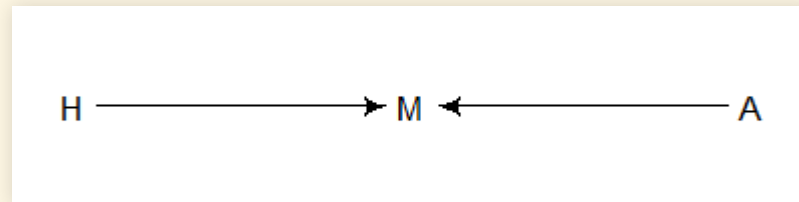


- Fungus F does not influence plant growth.
- Moisture M influences both plant growth $H1$ and fungus F
- Fitting our original model falsely implies that treatment benefits plants.
- This is a kind of *collider* effect.

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*

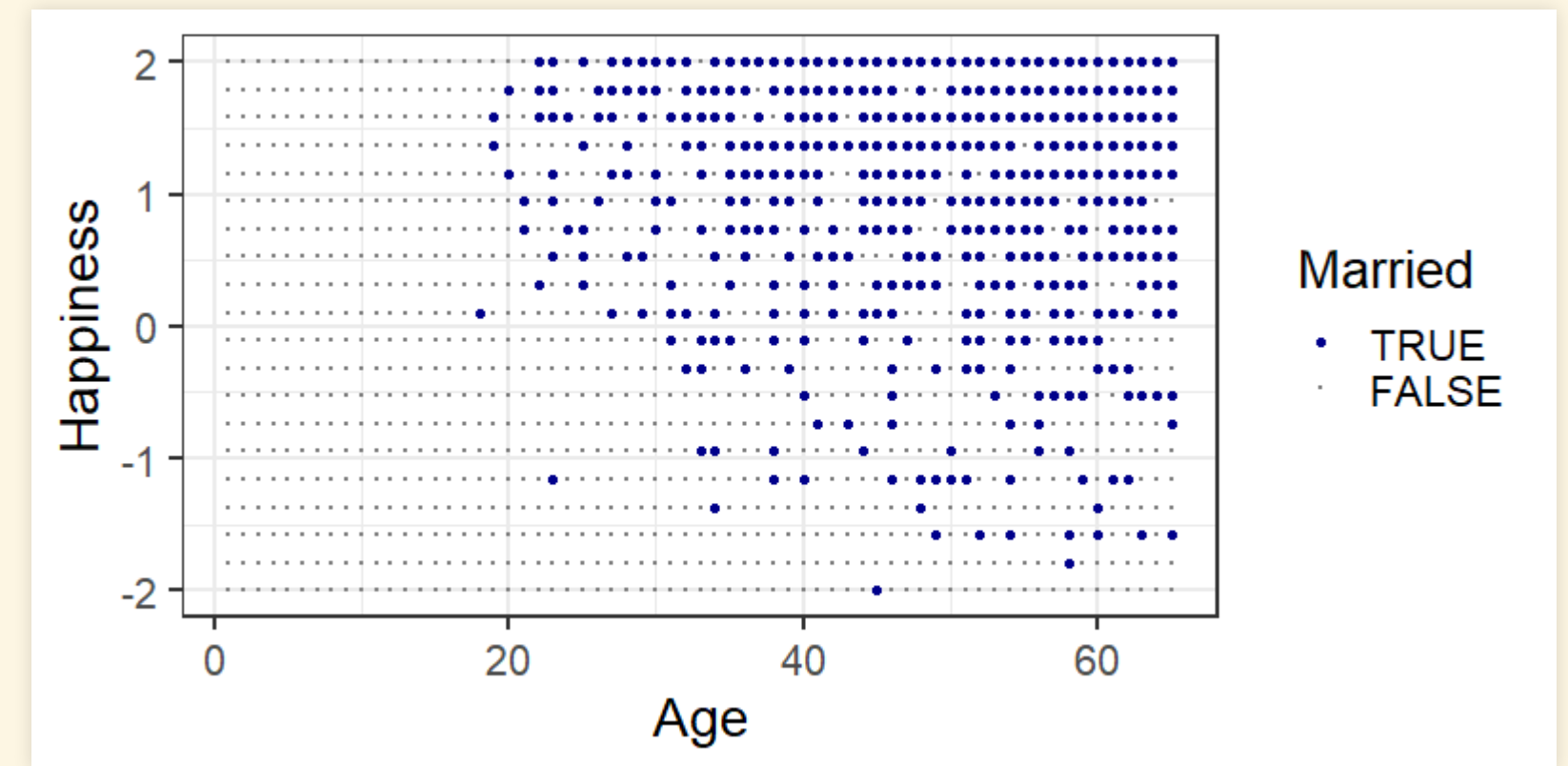
Analyze Happiness Data

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

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



```
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))
```

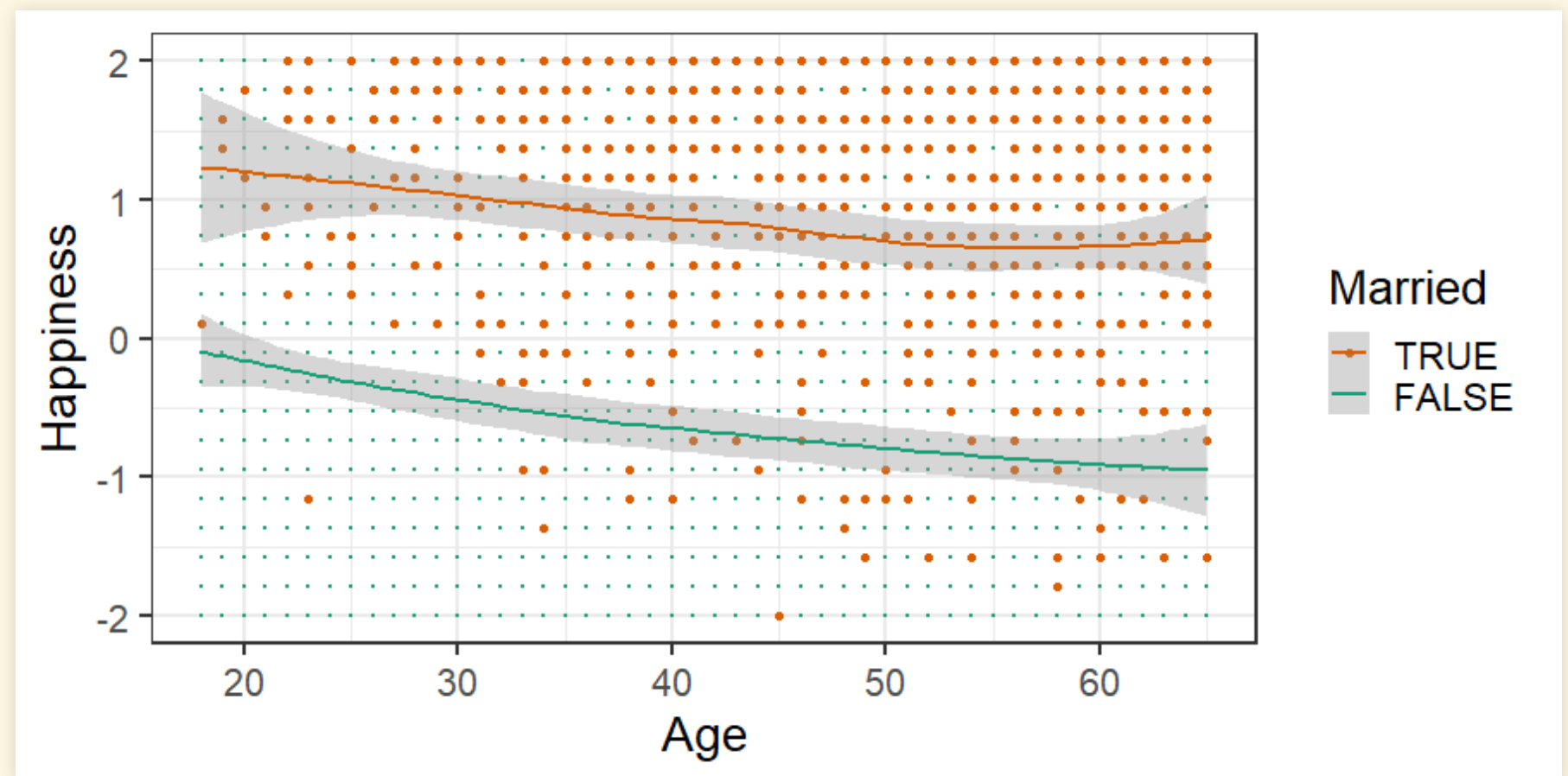
##		mean	sd	5.5%	94.5%
##	bA	-0.75	0.11	-0.93	-0.57
##	sigma	0.99	0.02	0.95	1.03

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

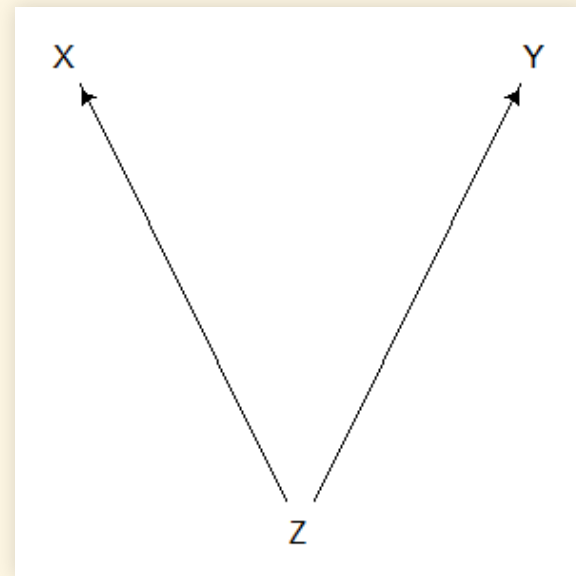


Categories of Confounding Relationships

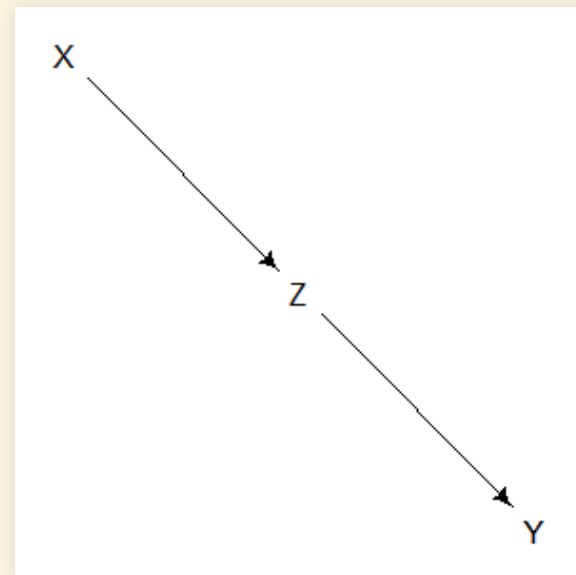
Categories of Confounding Relationships

- There are four major categories of confounding relationships:

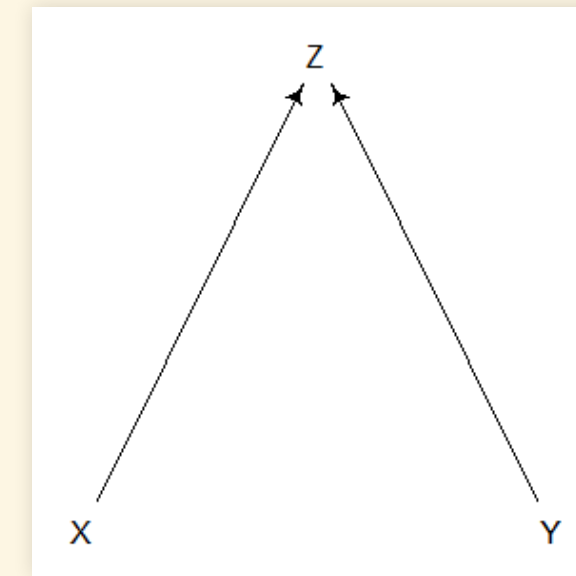
1. Fork



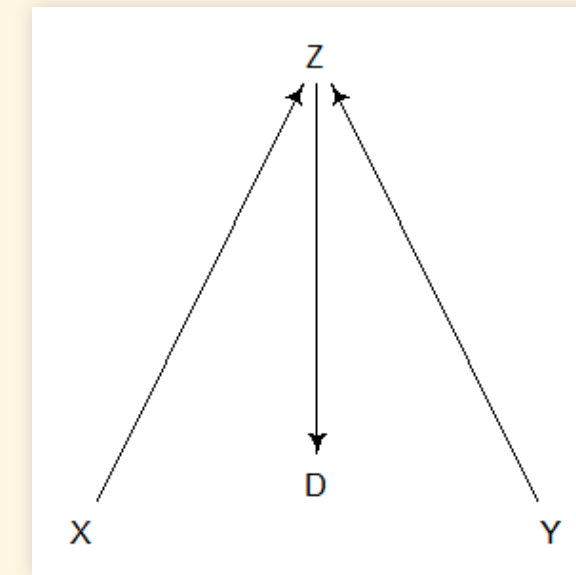
2. Pipe



3. Collider



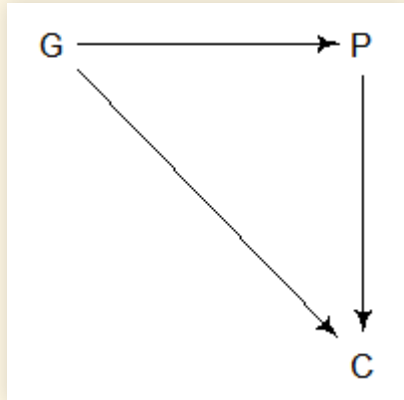
4. Descendant



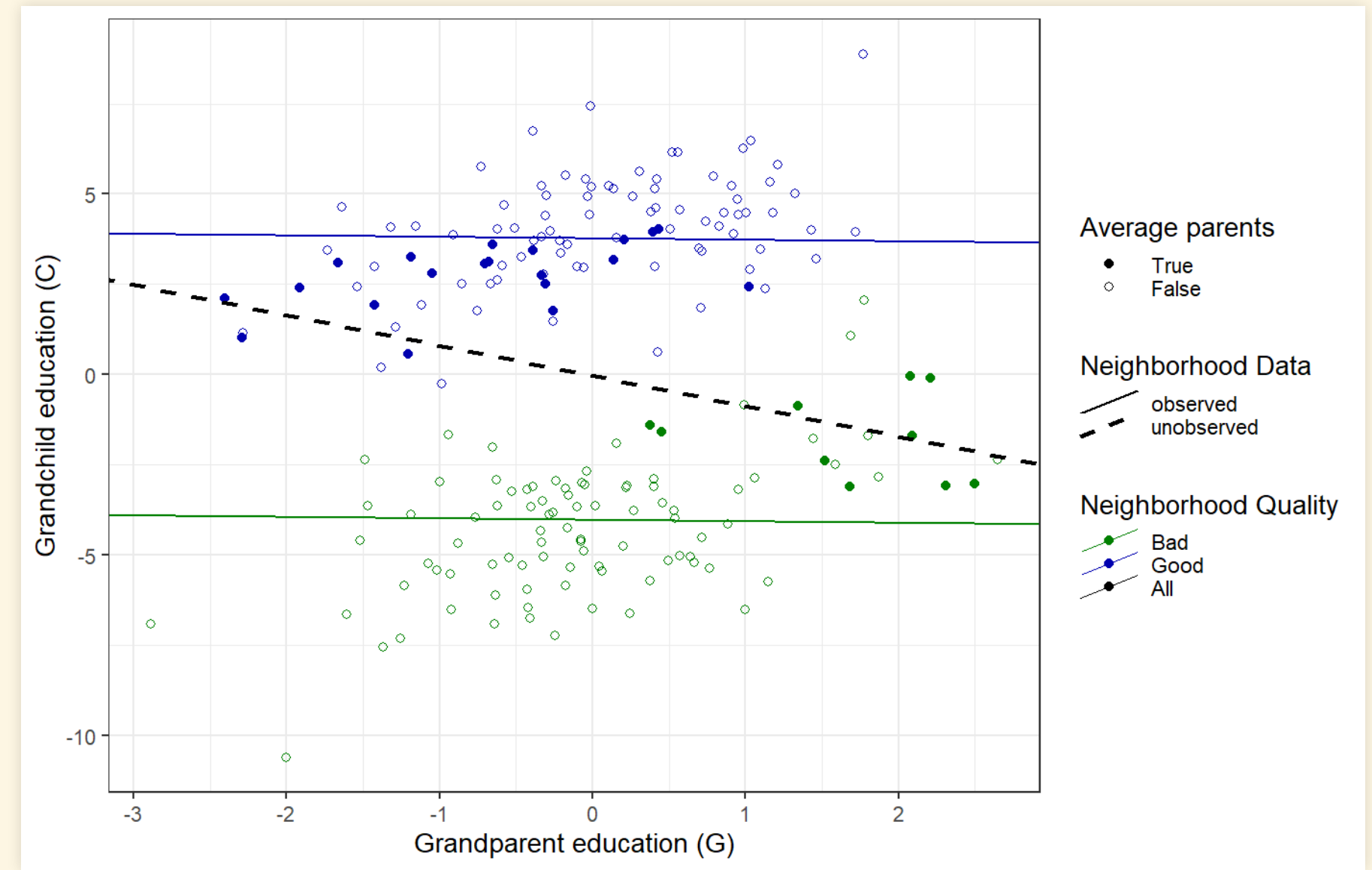
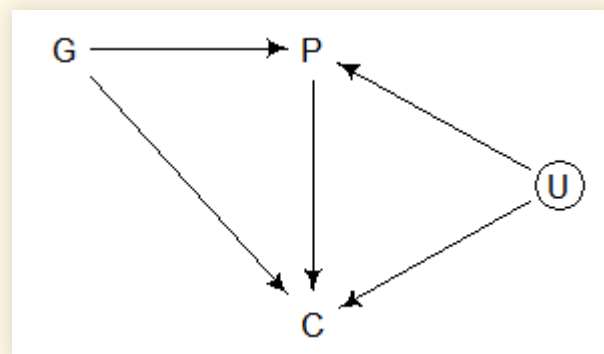
- All causal DAGs are build of combinations of these four patterns.

Example: Haunted DAG

- How do parents' P and grandparents' G educational attainment influence educational attainment of children C ?



- But there are unmeasured effects here, such as the character of the neighborhood.
 - Grandparents moved into the neighborhood after they finished school,
 - Parents and children grew up in the neighborhood and are affected by it.

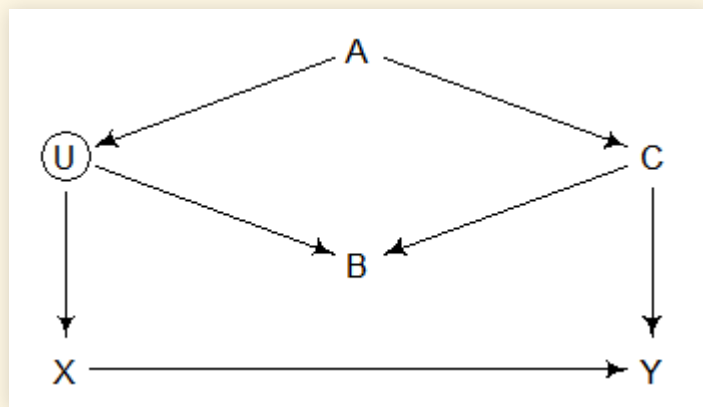


- There is a no correlation between G and C in each neighborhood
 - This is the correct answer.
- But when we don't account for the neighborhood effect, the collider bias makes it look like there's a negative correlation
 - more educated grandparents have less educated grandchildren

Backdoor Effects

- In the age and happiness example, conditioning on the marriage variable created bias,
- But in the grandparent, parent, and children example, we needed to condition on the neighborhood to avoid bias.
 - How can we tell when to condition on a variable?

- Consider this DAG:



- How does X affect Y ?

- Backdoor paths:
 1. $X \leftarrow U \leftarrow A \rightarrow C \rightarrow Y$
 2. $X \leftarrow U \rightarrow B \leftarrow C \rightarrow Y$
- Which backdoor path is open?
 1. This path is open because it has no internal collider
 2. This path is closed because (B) is a collider.
 - If we condition on (B) , it will open the backdoor and introduce a collider effect.
- Closing backdoors:
 - We don't observe (U) , so we can't condition on it.
 - To close the backdoor path #1, condition on (A) or (C) .

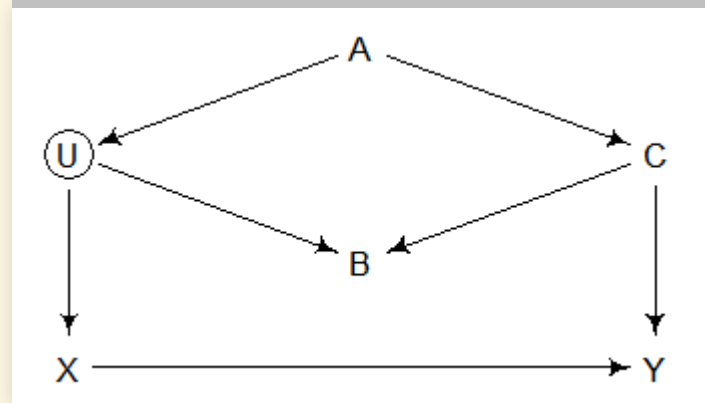
Automated Analysis

- Define the DAG

```
library(dagitty)
dag_two_roads <- dagitty("dag {
  U [unobserved]
  X -> Y
  X <- U <- A -> C -> Y
  U -> B <- C
}")
```

- Optionally, draw the DAG diagram

```
coordinates(dag_two_roads) <- list(
  x = c(U = 0, X = 0, A = 1, B = 1, C = 2, Y = 2),
  y = c(U = 0, X = 1, A = -0.5, B = 0.5, C = 0, Y = 1)
)
drawdag(dag_two_roads)
```



- Analyze to identify which variables to condition on

```
adjustmentSets(dag_two_roads, exposure = "X", outcome = "Y")
```

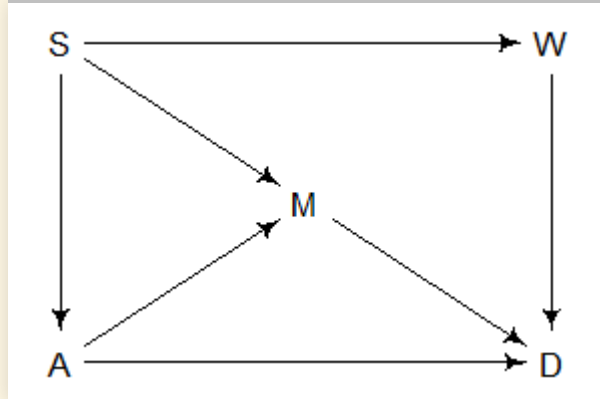
```
## { C }
## { A }
```

- Condition on A or C

Backdoors in Waffle-House and Divorce

- Waffle-House and Divorce

```
dag_waffles <- dagitty("dag {  
  A -> D  
  A -> M -> D  
  A <- S -> M  
  S -> W -> D  
}")
```



S = state, W = waffle-house restaurants,
 A = median age at marriage, M = marriage rate,
and D = divorce rate.

- Identify which variables to condition on

```
adjustmentSets(dag_waffles, exposure="W", outcome="D")
```

```
## { A, M }  
## { S }
```

- What does this mean?

- Backdoors:

- $(W \leftarrow S \rightarrow M \rightarrow D)$
- $(W \leftarrow S \rightarrow A \rightarrow D)$
- $(W \leftarrow S \rightarrow A \rightarrow M \rightarrow D)$

- All of these pass through S .
- To close the backdoors, either
 - Condition on S , or
 - Condition on both A and M .

- Further analysis: *conditional independencies*

```
impliedConditionalIndependencies(dag_waffles)
```

```
## A _||_ W | S  
## D _||_ S | A, M, W  
## M _||_ W | S
```

- If we condition on S , then A and M should both be independent of W
- If we simultaneously condition on A , M , and W , then D should be independent of S .