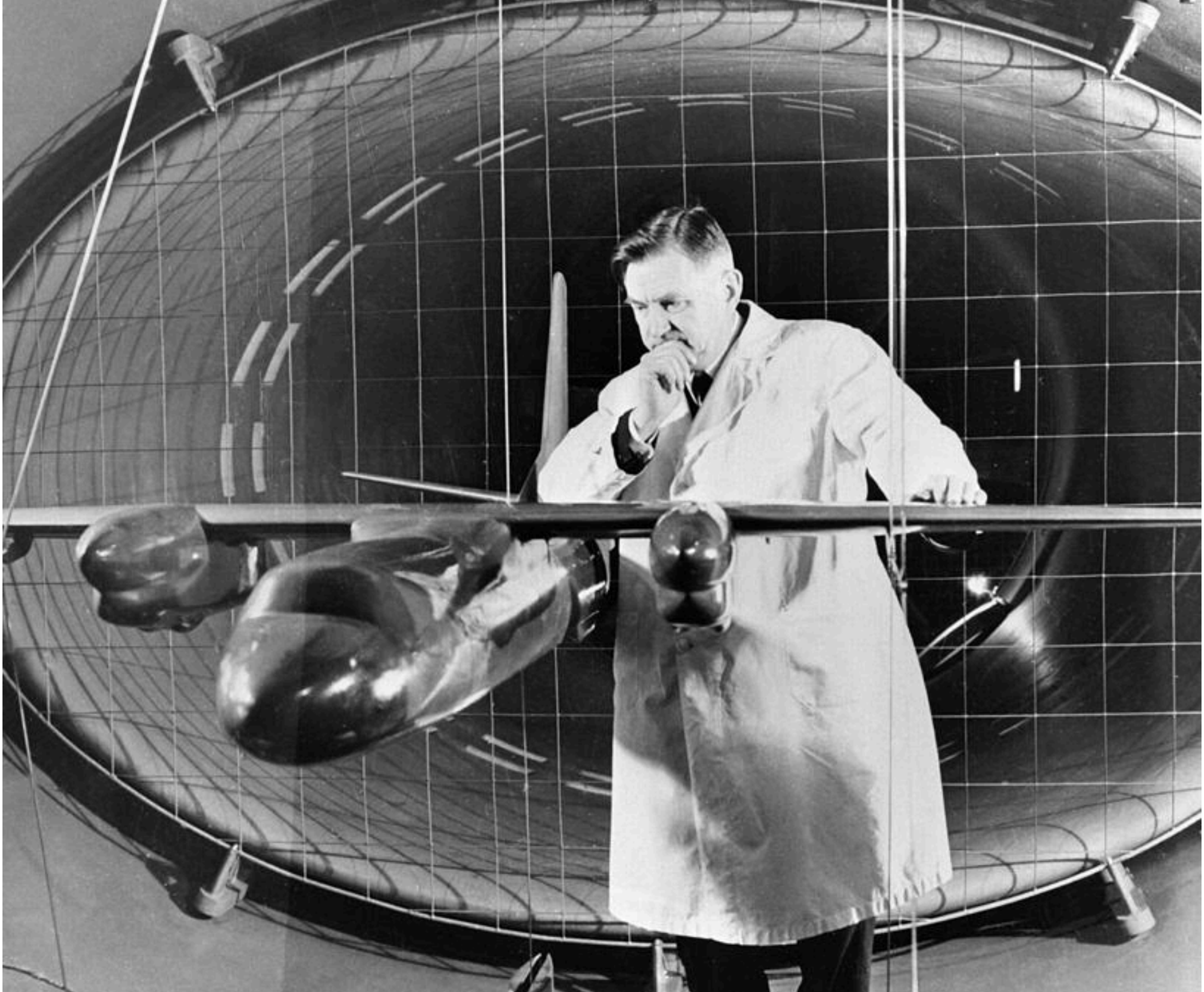
Practical Principles in Applied Bayesian Data Analysis

Richard McElreath
Max Planck Institute for
Evolutionary Anthropology
Leipzig



Responsible Belief Engineering

- Inside the Model: Prior predictive simulation
- Outside the Model: Causal inference
- Within the Computer: Building models for better computation

Materials

@rlmcelreath

- Examples from draft 2nd edition of Statistical Rethinking
  - https://xcelab.net/rm/sr2/
  - Password: tempest

- Experimental branch of rethinking R package
  - `install_github("rmcelreath/rethinking",ref="Experimental")`

- Everything linked on course page:
  - https://github.com/rmcelreath/statrethinking_winter2019

Everybody overfits

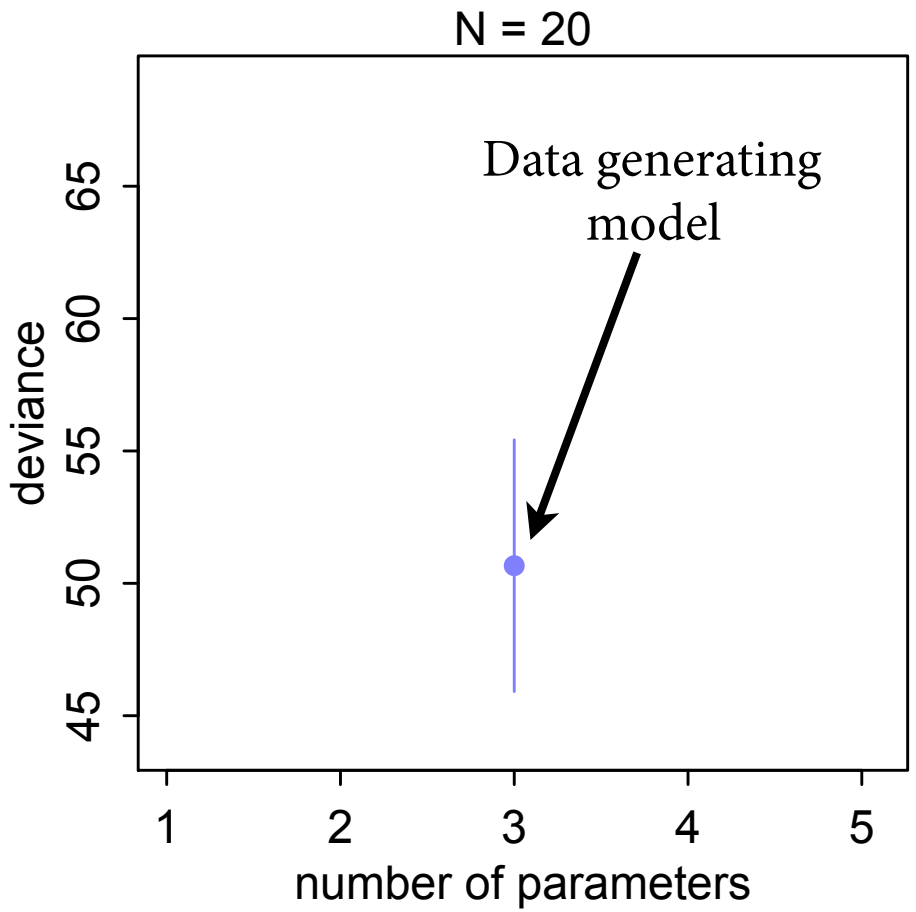


Figure 7.7

Everybody overfits

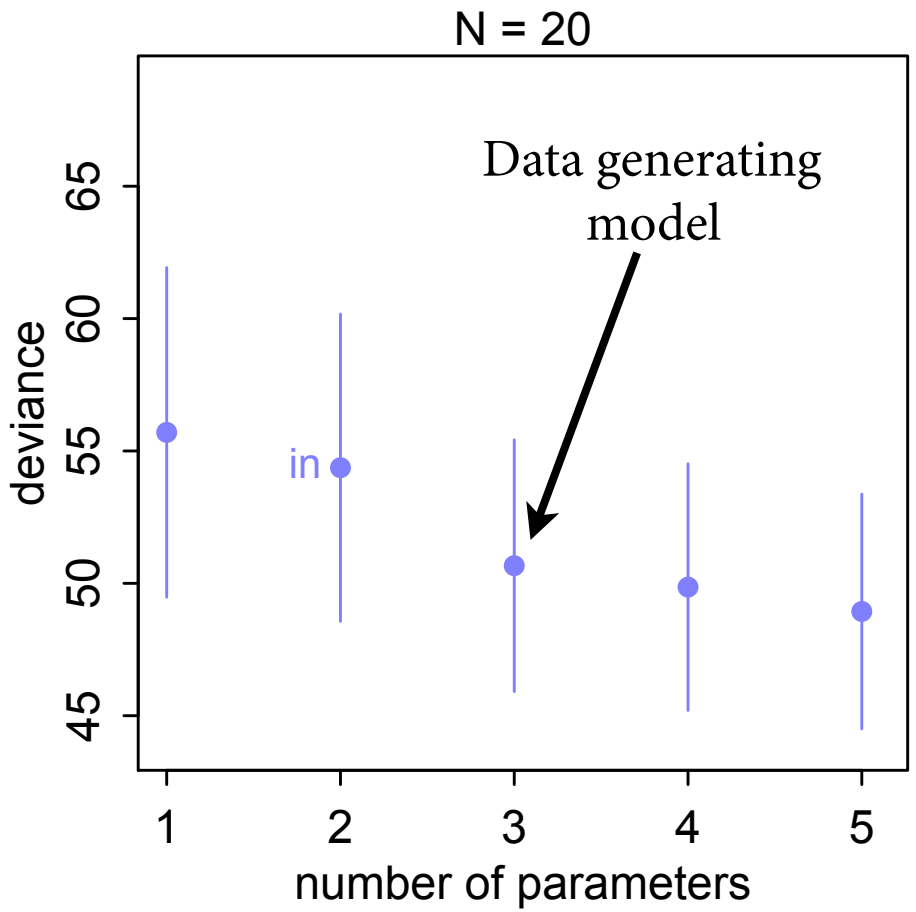


Figure 7.7

Everybody overfits

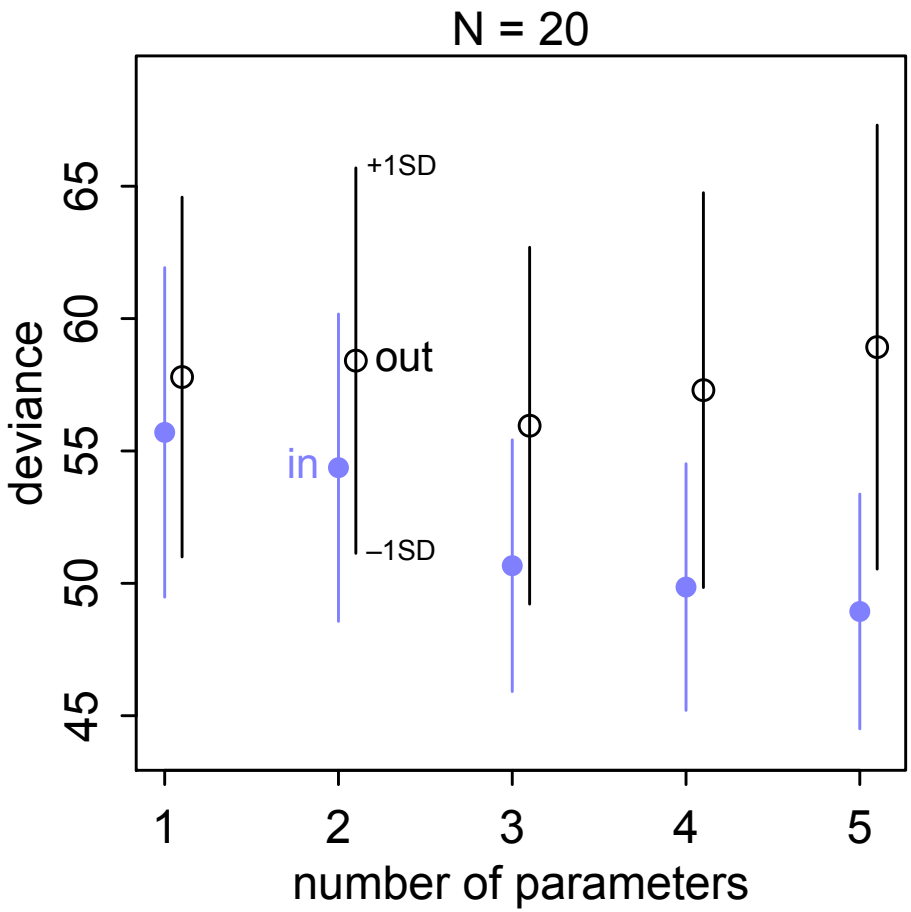


Figure 7.7

Everybody overfits

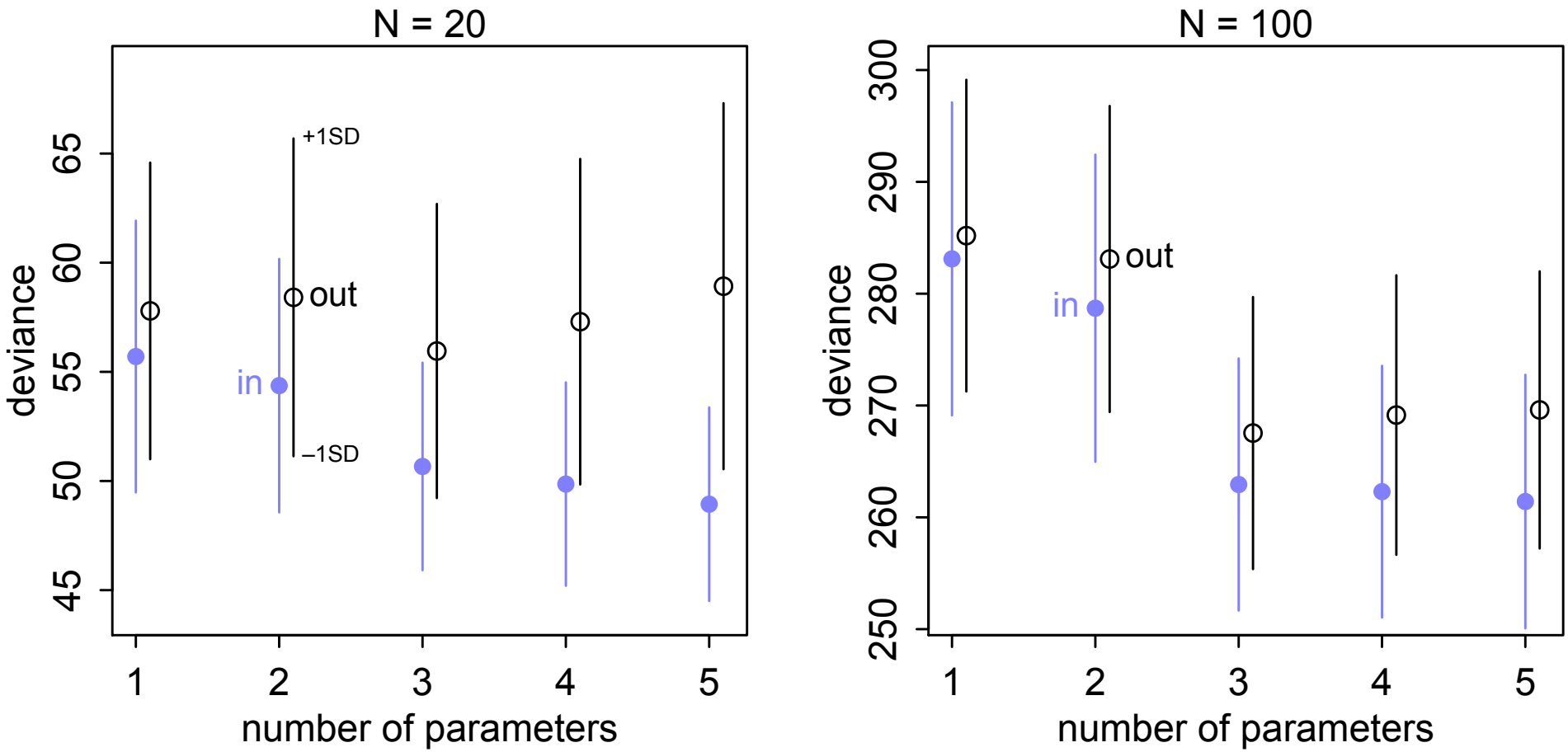


Figure 7.7

Regularization

- **Must be skeptical of the sample!**
- Use informative, conservative priors to reduce overfitting => model learns less from sample
- But if too skeptical, model learns too little
- Such priors are *regularizing*

Regularization

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta x_i$$
$$\alpha \sim \text{Normal}(0, 100)$$
prior ⇢ $$\beta \sim \text{Normal}(0, 1)$$
$$\sigma \sim \text{Uniform}(0, 10)$$

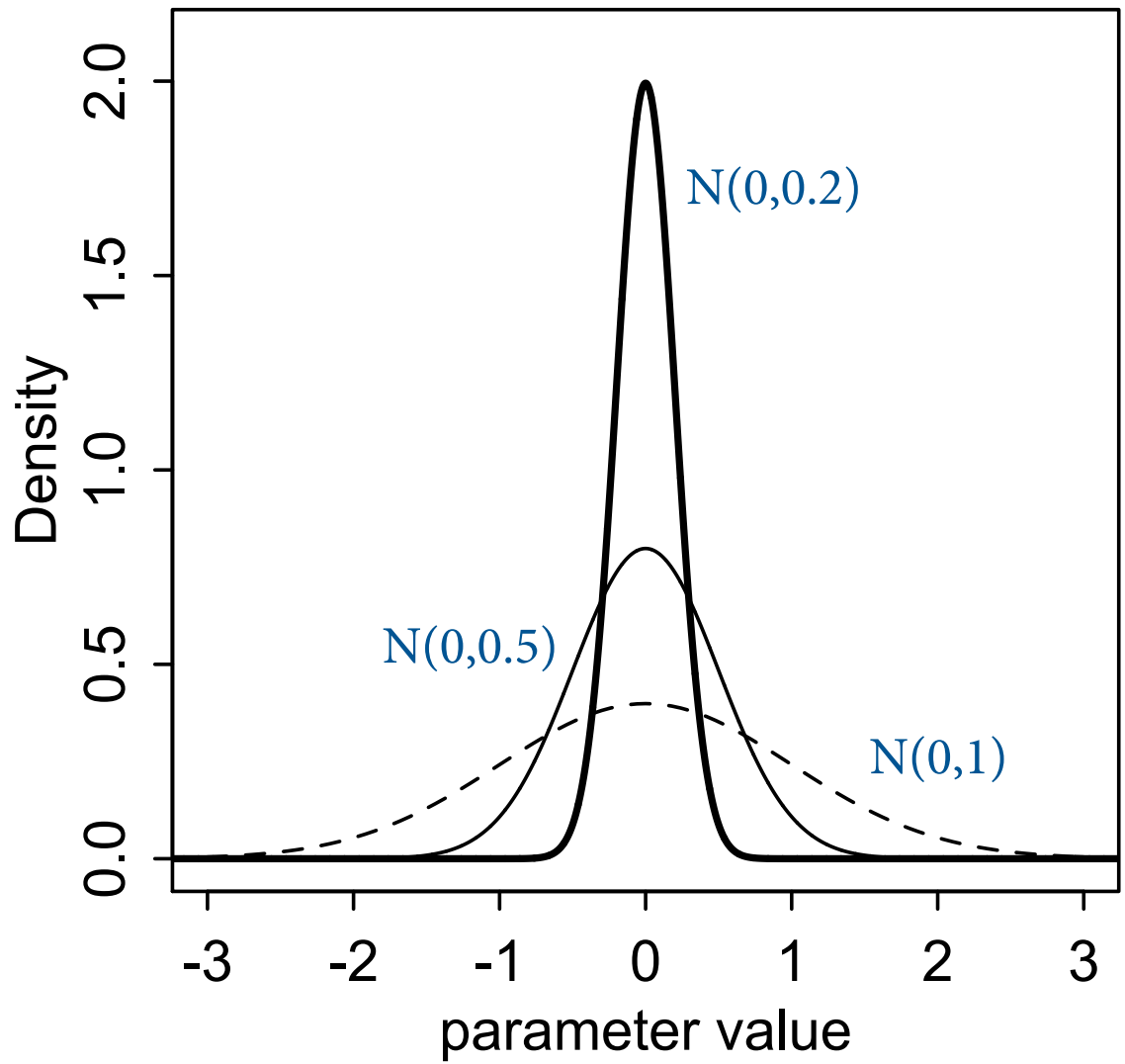
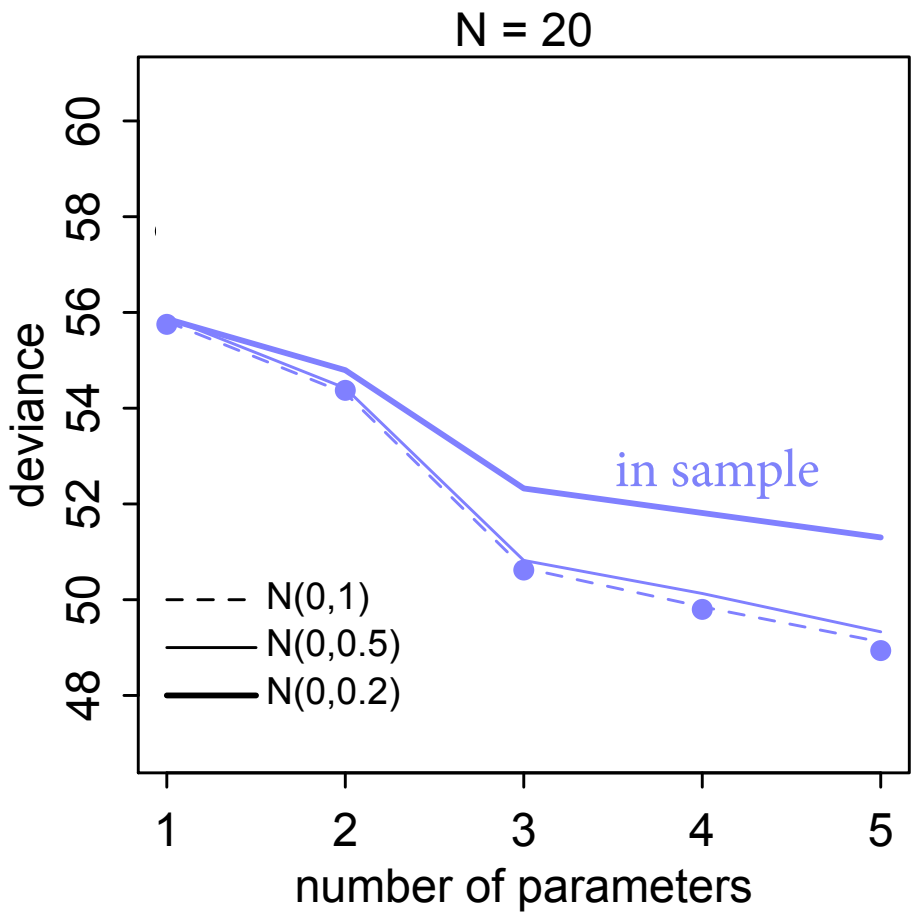


Figure 7.8

Regularization



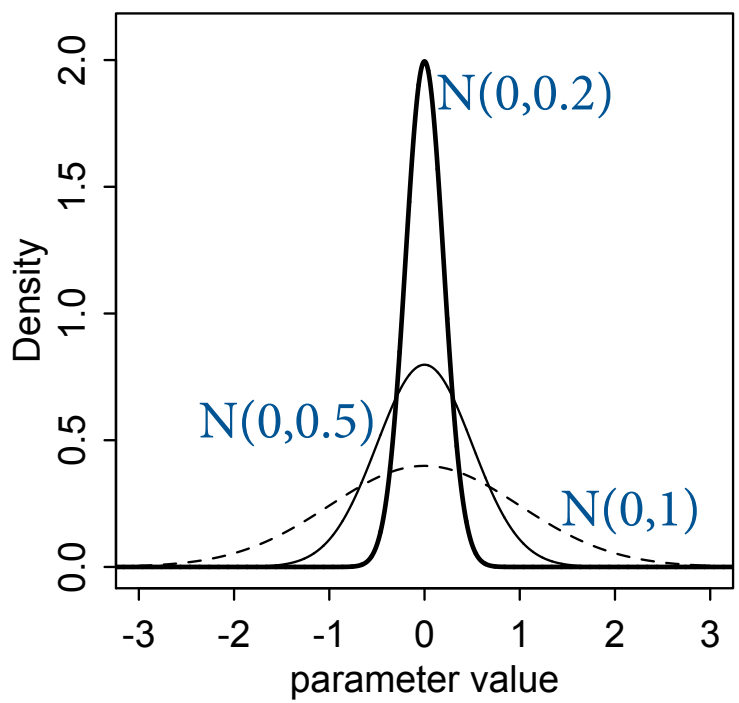
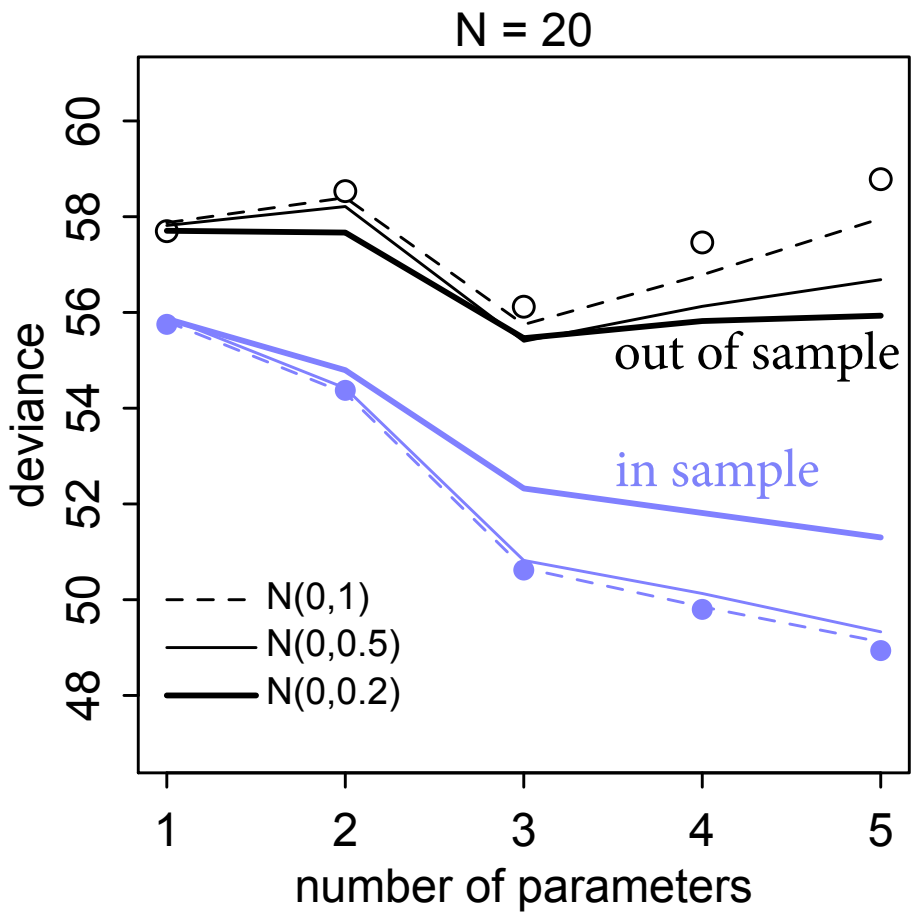


Figure 7.9

Regularization



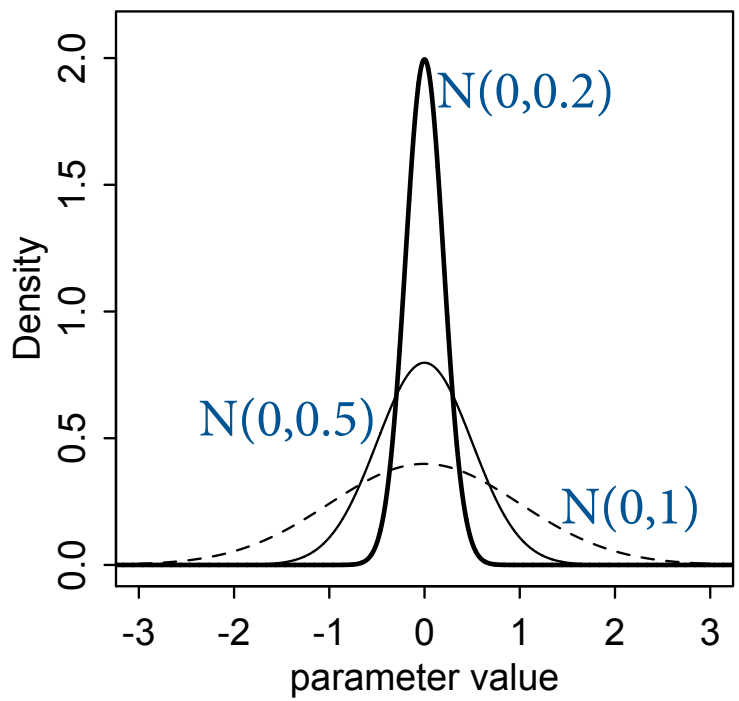


Figure 7.9

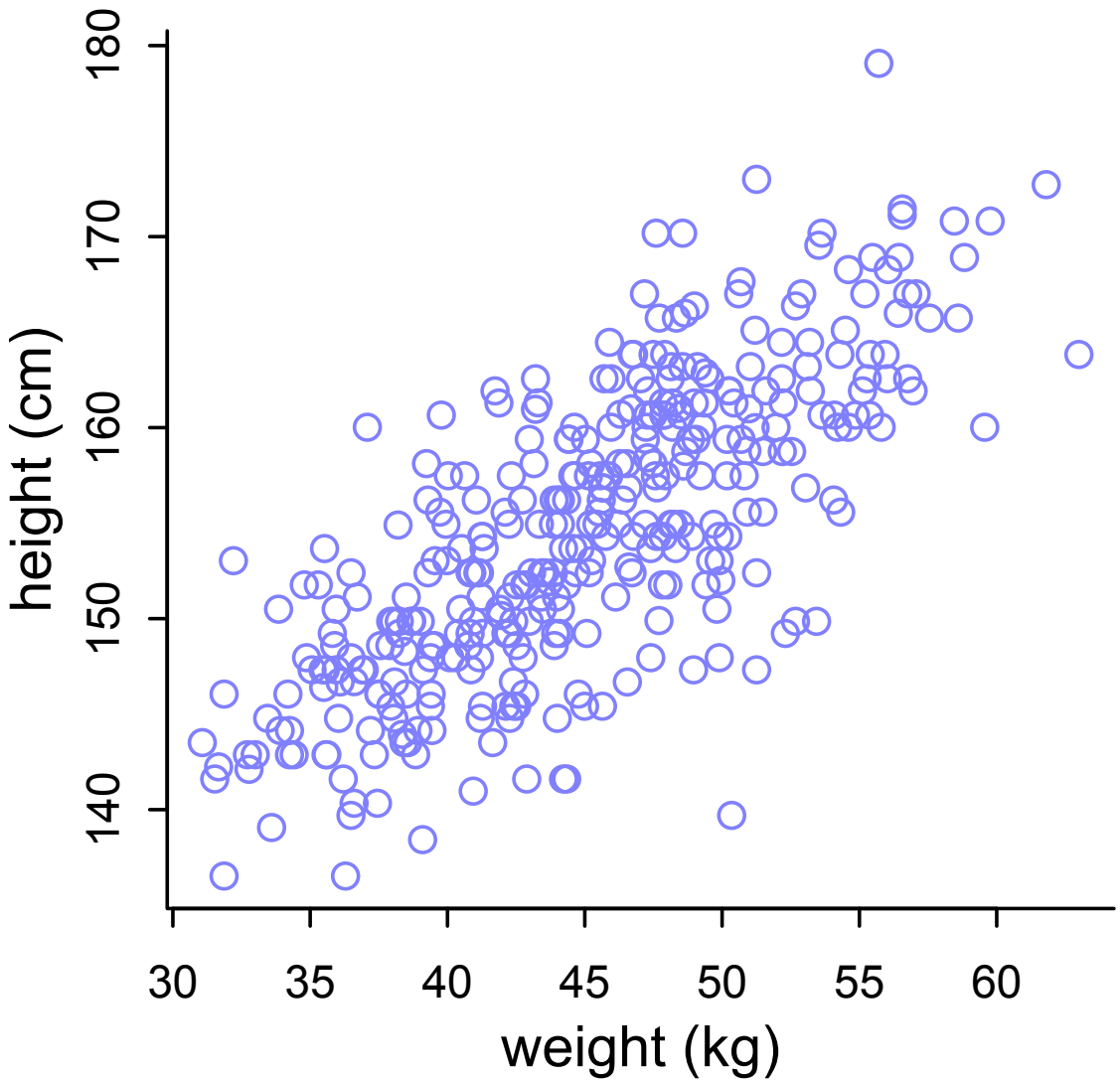
Prior predictive simulation

- Minimum standard: Model predictions should make sense **even before updating**
  - Find priors that achieve this, but are blind to training
- Helps in understanding meaning of parameters
- Helps in regularizing inference
- Helps in updating model (better mixing)

Prior predictive simulation

```
library(rethinking)
data(Howell1)
```

$$h_i \sim \text{Normal}(\mu, \sigma)$$
$$\mu \sim \text{Normal}(178, 20)$$
$$\sigma \sim \text{Uniform}(0, 50)$$



```
sample_mu <- rnorm( 1e4 , 178 , 100 )
sample_sigma <- runif( 1e4 , 0 , 50 )
prior_h <- rnorm( 1e4 , sample_mu , sample_sigma )
dens( prior_h )
```

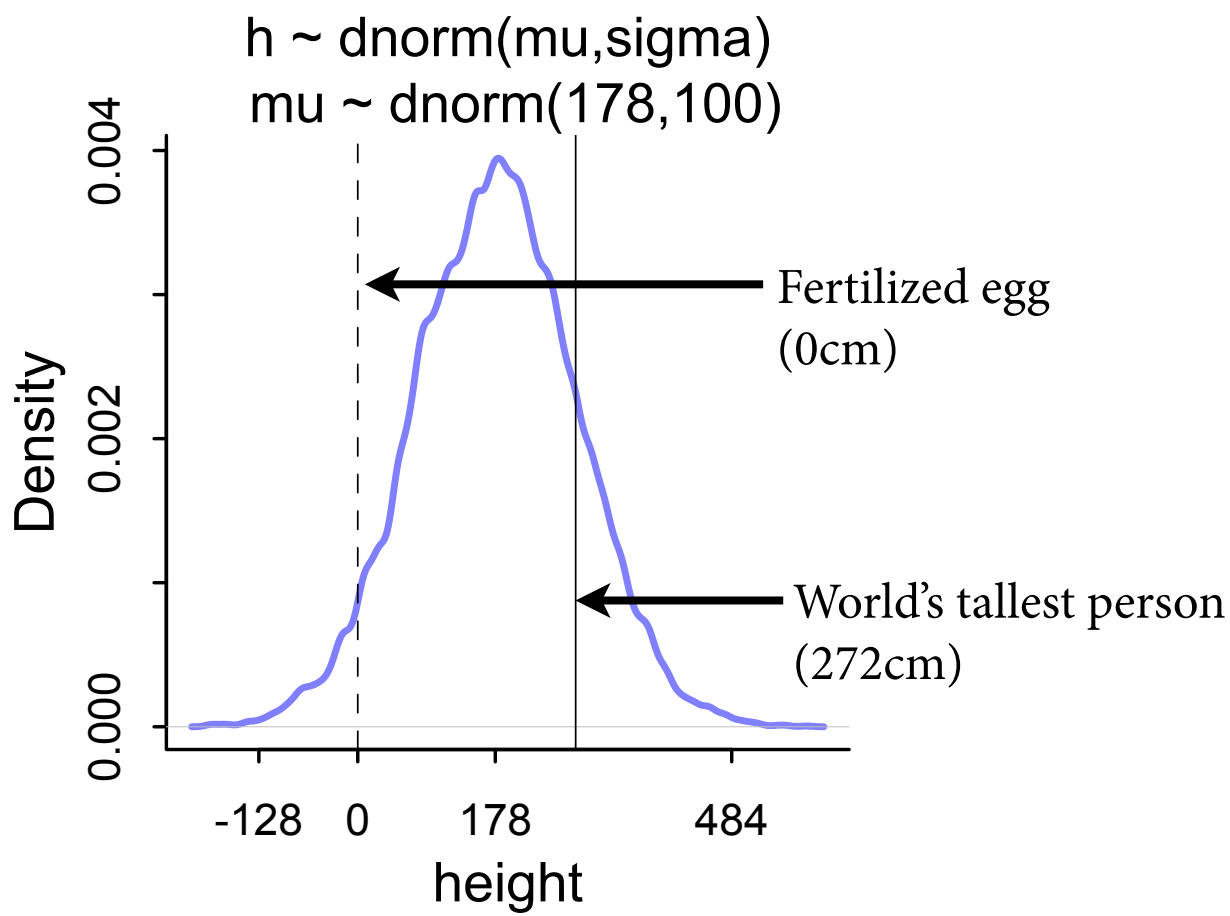


Figure 4.3

Prior predictive distribution

```
set.seed(2971)
N <- 100                    # 100 lines
a <- rnorm( N , 178 , 20 )
b <- rnorm( N , 0 , 10 )
```

$$\alpha \sim \text{Normal}(178, 20)$$
$$\beta \sim \text{Normal}(0, 10)$$

R code 4.38

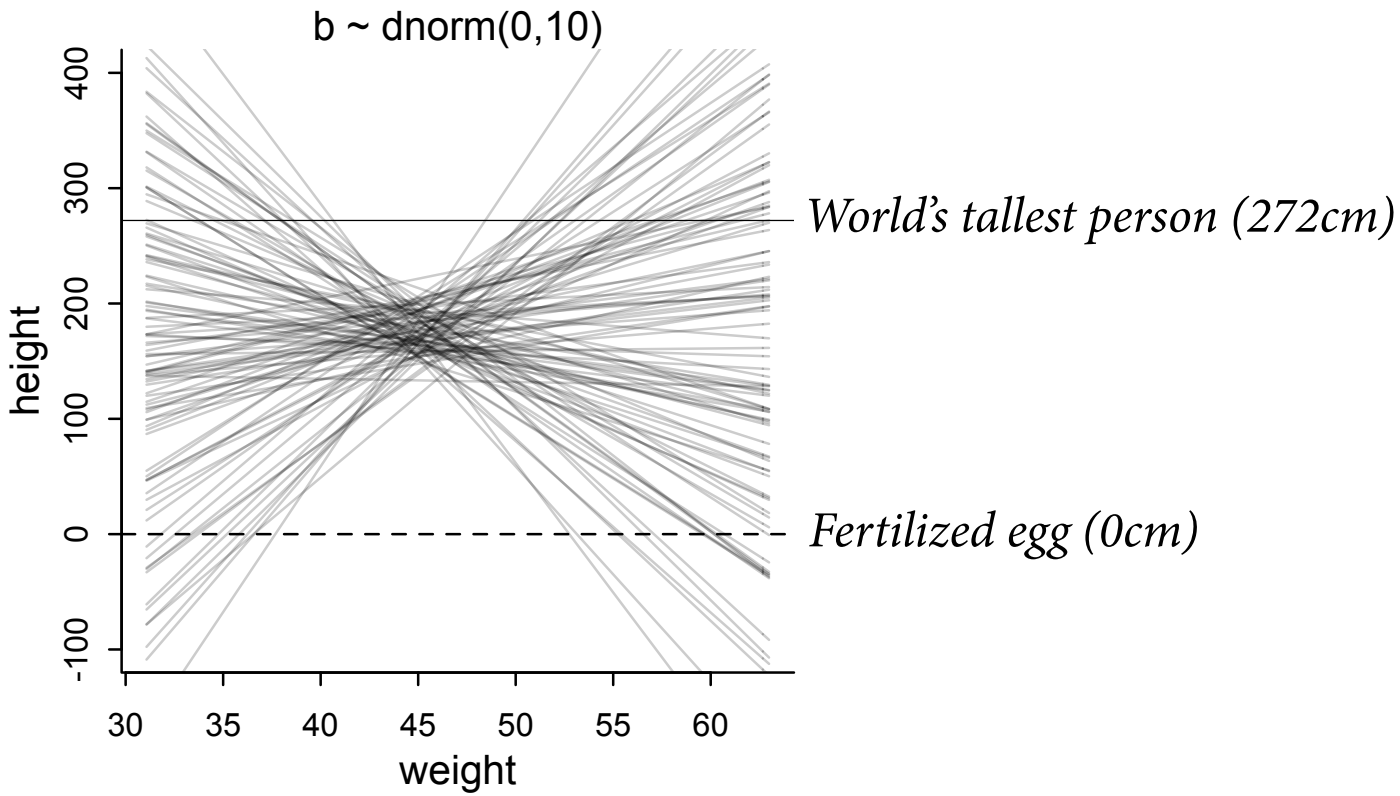


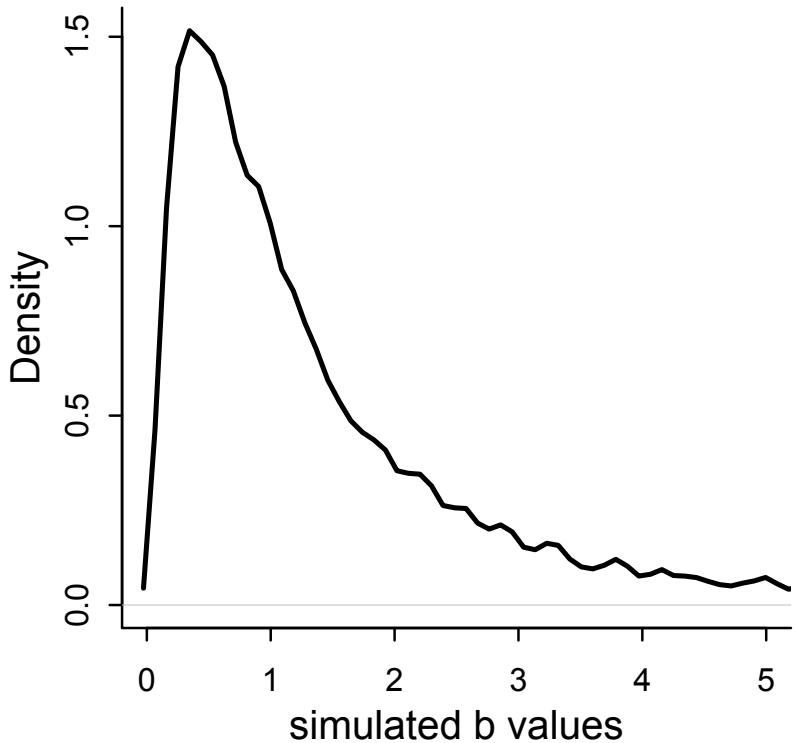
Figure 4.5

Prior predictive distribution

R code 4.40

```
b <- rlnorm( 1e4 , 0 , 1 )
dens( b , xlim=c(0,5) , adj=0.1 )
```

$$\beta \sim \text{Log-Normal}(0, 1)$$



Prior predictive distribution

R code 4.41

```
set.seed(2971)
N <- 100                    # 100 lines
a <- rnorm( N , 178 , 20 )
b <- rlnorm( N , 0 , 1 )
```

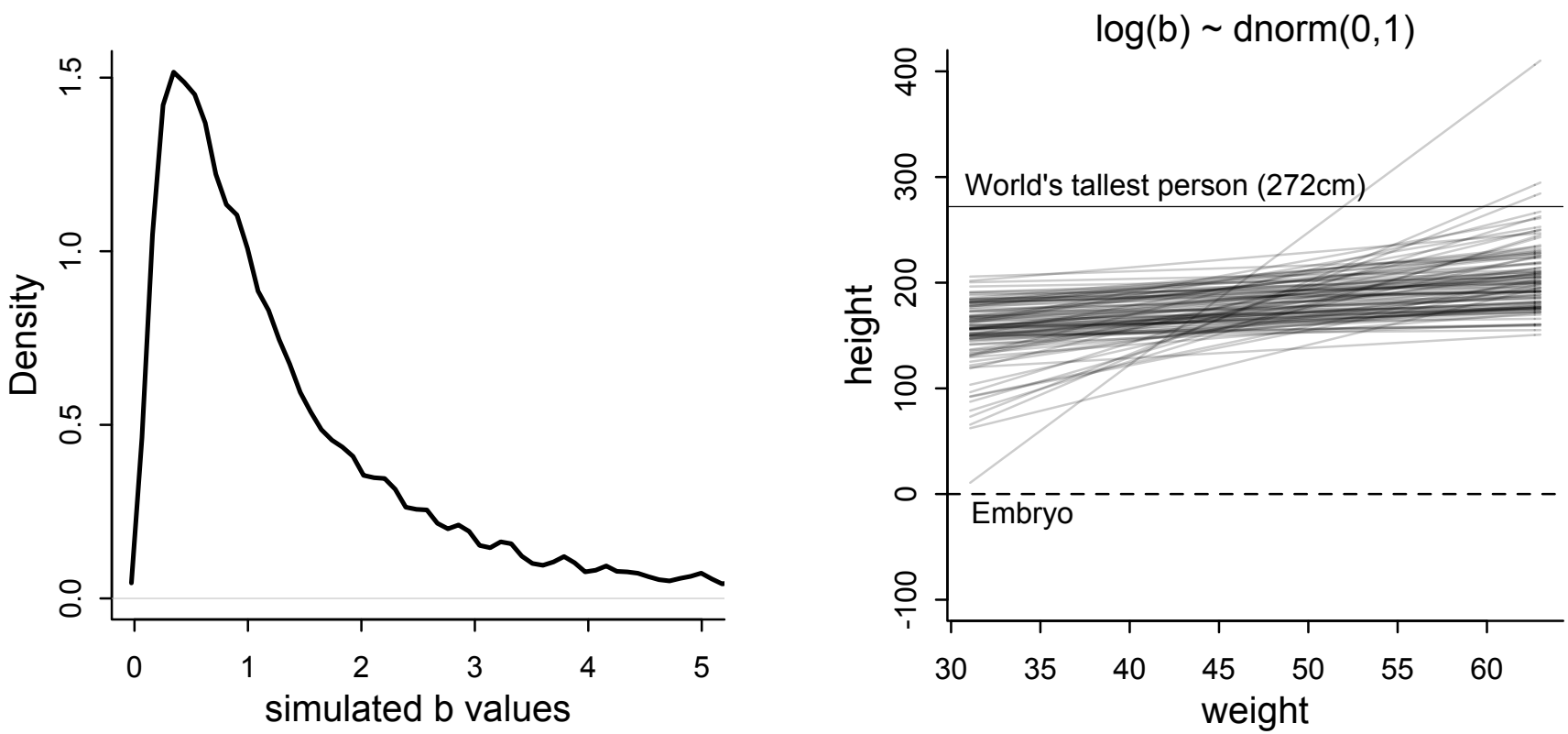


Figure 4.5

```
library(rethinking)
data(rugged)
d <- rugged
```

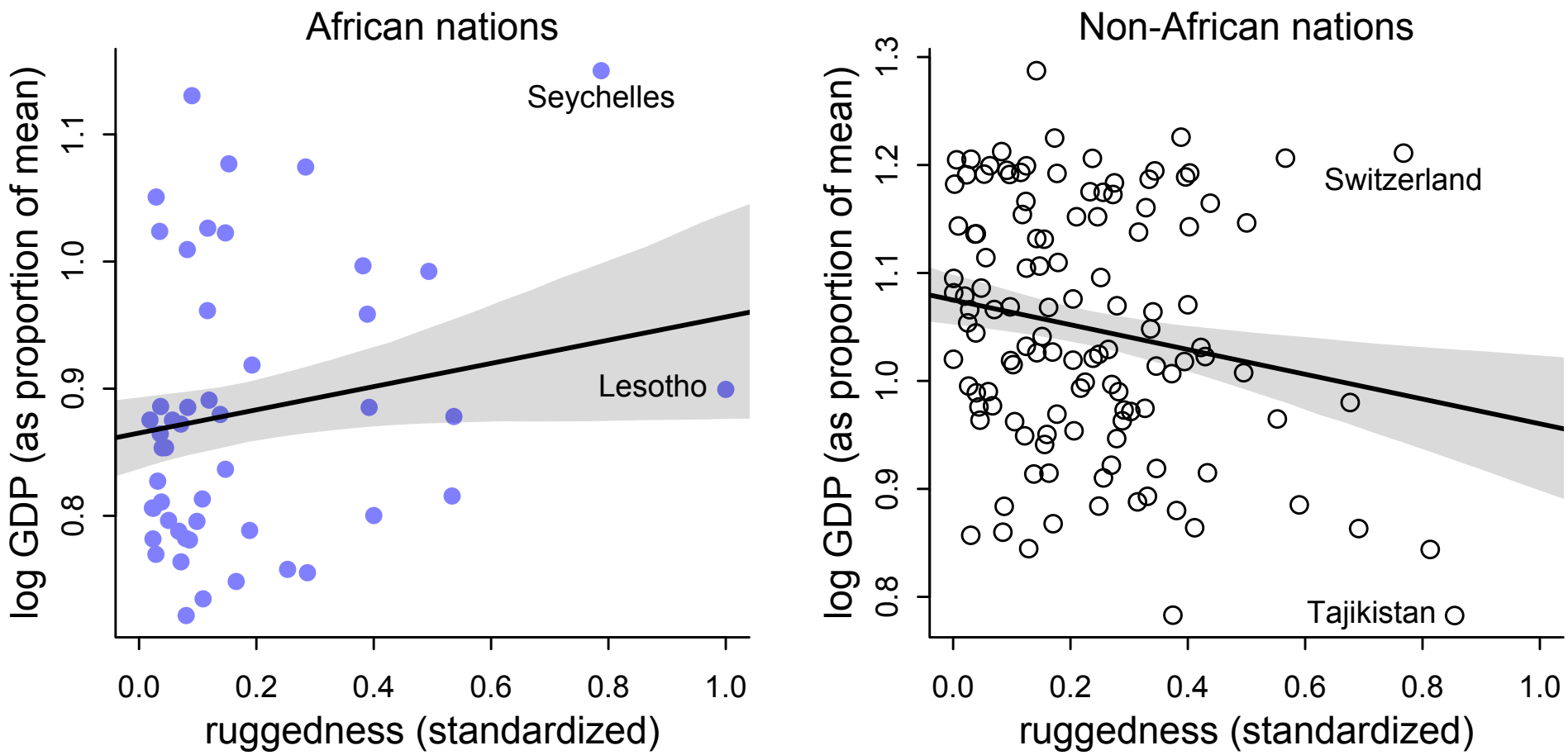


Figure 8.2

The sermon on priors

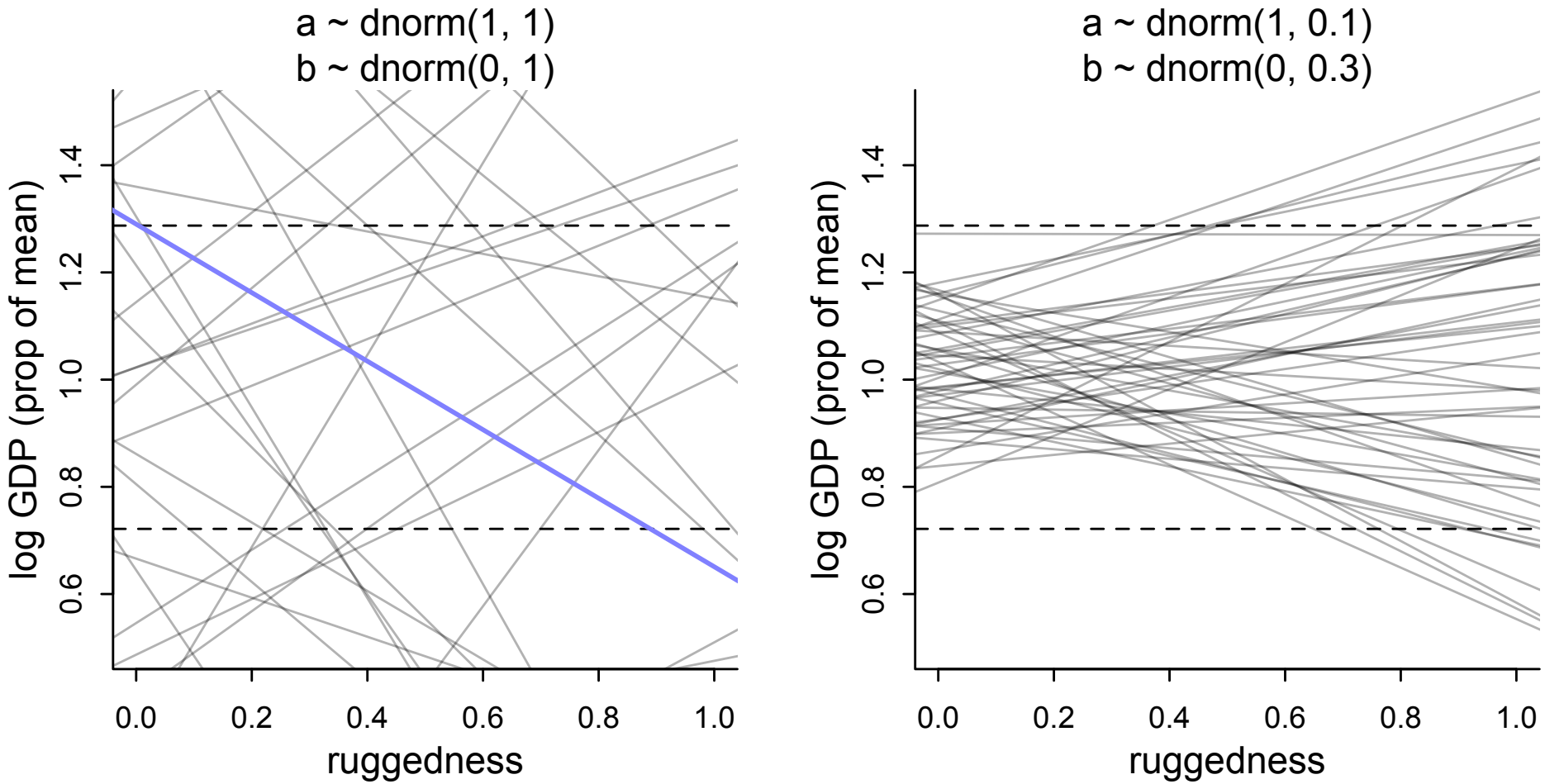
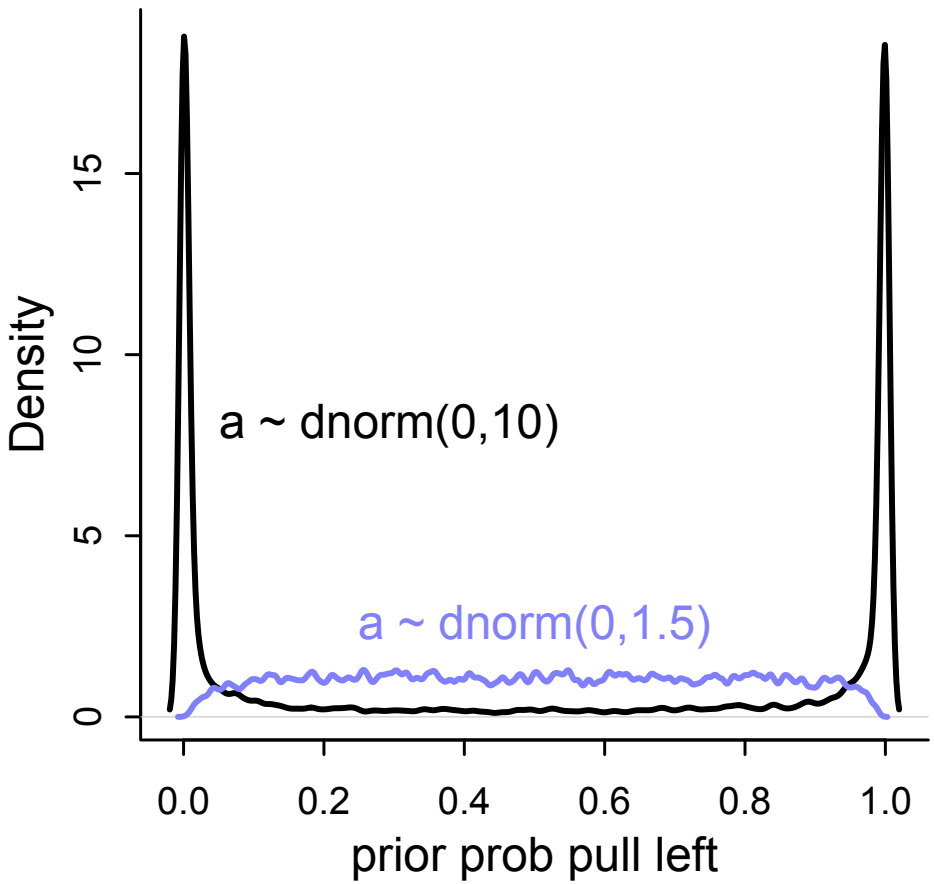


Figure 8.3

Logit link priors

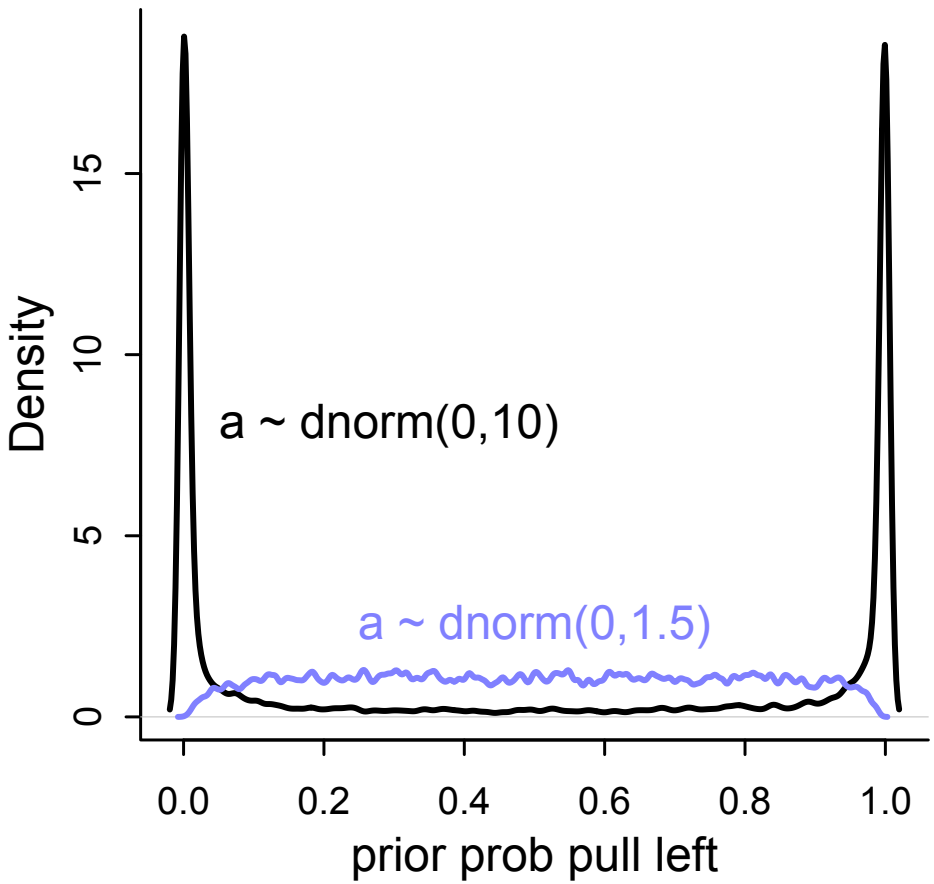
- Prior on logit scale not same shape as prior on probability scale
- Use prior simulation to understand

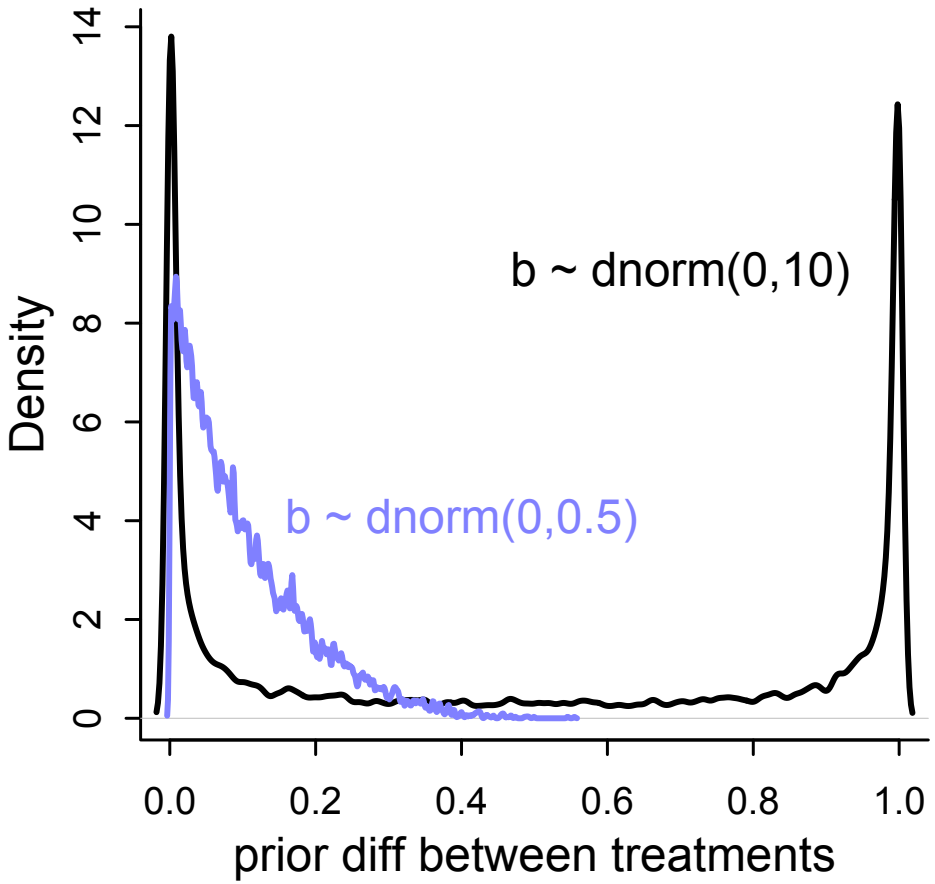
$$y_i \sim \text{Bernoulli}(p_i)$$
$$\text{logit}(p_i) = \alpha + \beta_{\text{ID}[i]}$$




```
a <- rnorm( 1e4 , 0 , 10 )
p <- inv_logit( a )
```

Figure 11.3





$$y_i \sim \text{Bernoulli}(p_i)$$
$$\text{logit}(p_i) = \alpha + \beta_{\text{ID}[i]}$$

Figure 11.3

Poisson GLMs also funky

total_tools (*outcome*)

expected tools for case i

log link

$$T_i \sim \text{Poisson}(\lambda_i)$$

$$\log \lambda_i = \alpha_{\text{CID}[i]} + \beta_{\text{CID}[i]} \log P_i$$

$$\alpha_j \sim \text{to be determined}$$

$$\beta_j \sim \text{to be determined}$$

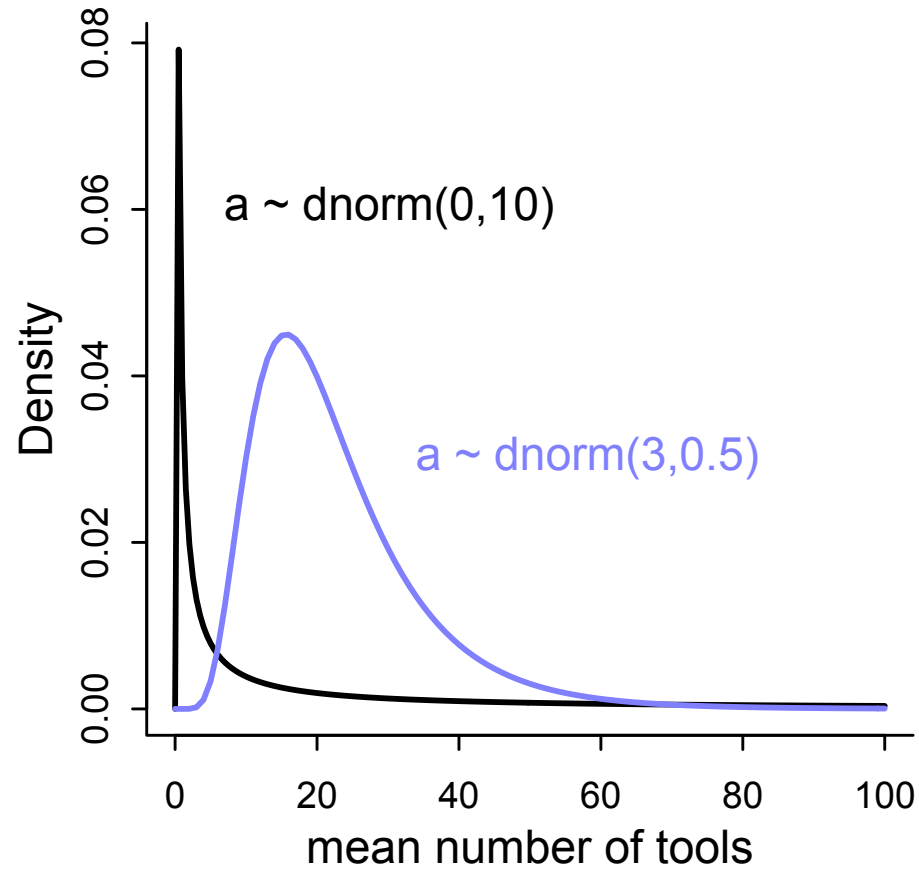
Priors & the log link

- Log link not intuitive — simulate

$$T_i \sim \text{Poisson}(\lambda_i)$$
$$\log \lambda_i = \alpha$$
$$\alpha \sim \text{Normal}(0, 10)$$

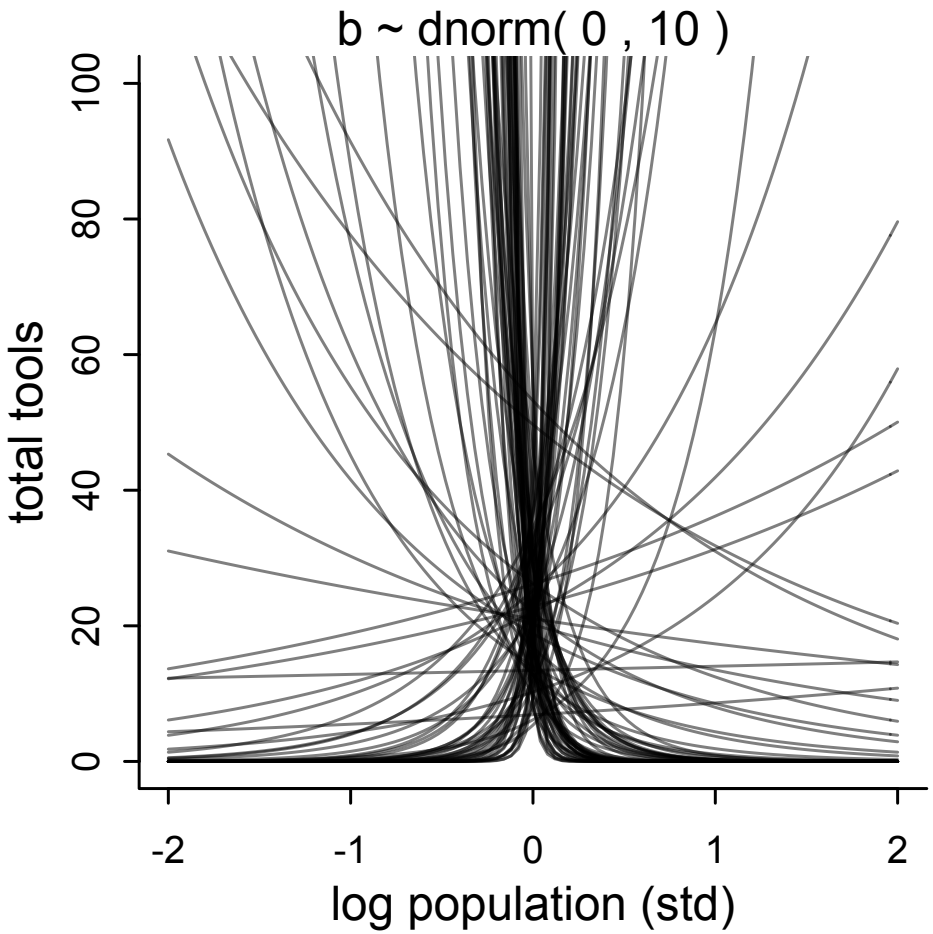
```
a <- rnorm(1e4,0,10)
lambda <- exp(a)
mean( lambda )
```

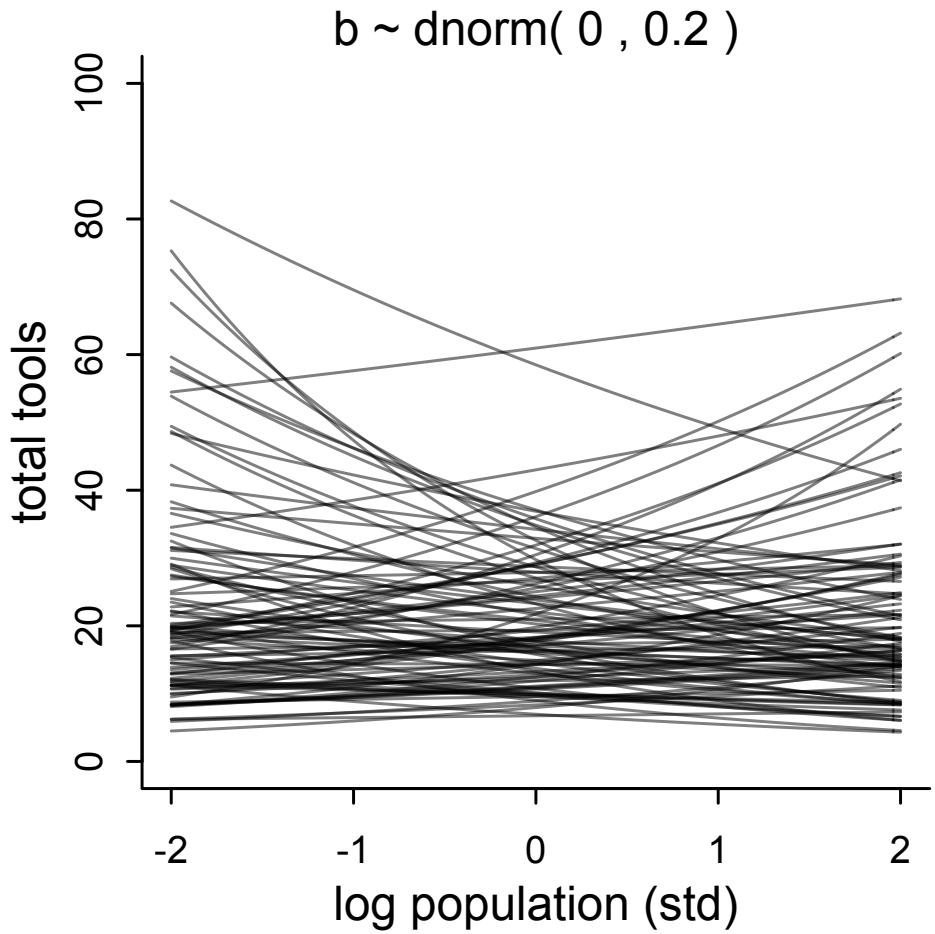
```
[1] 9.622994e+12
```



Priors & the log link

- Slopes equally unintuitive





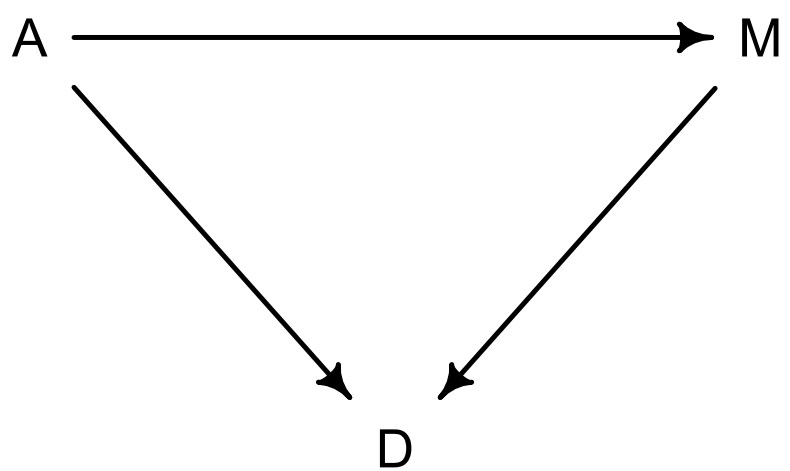

FACTOR
factor
FACTORS
FACTOR
FACTOR
FACTOR
CAUSAL
SALAD

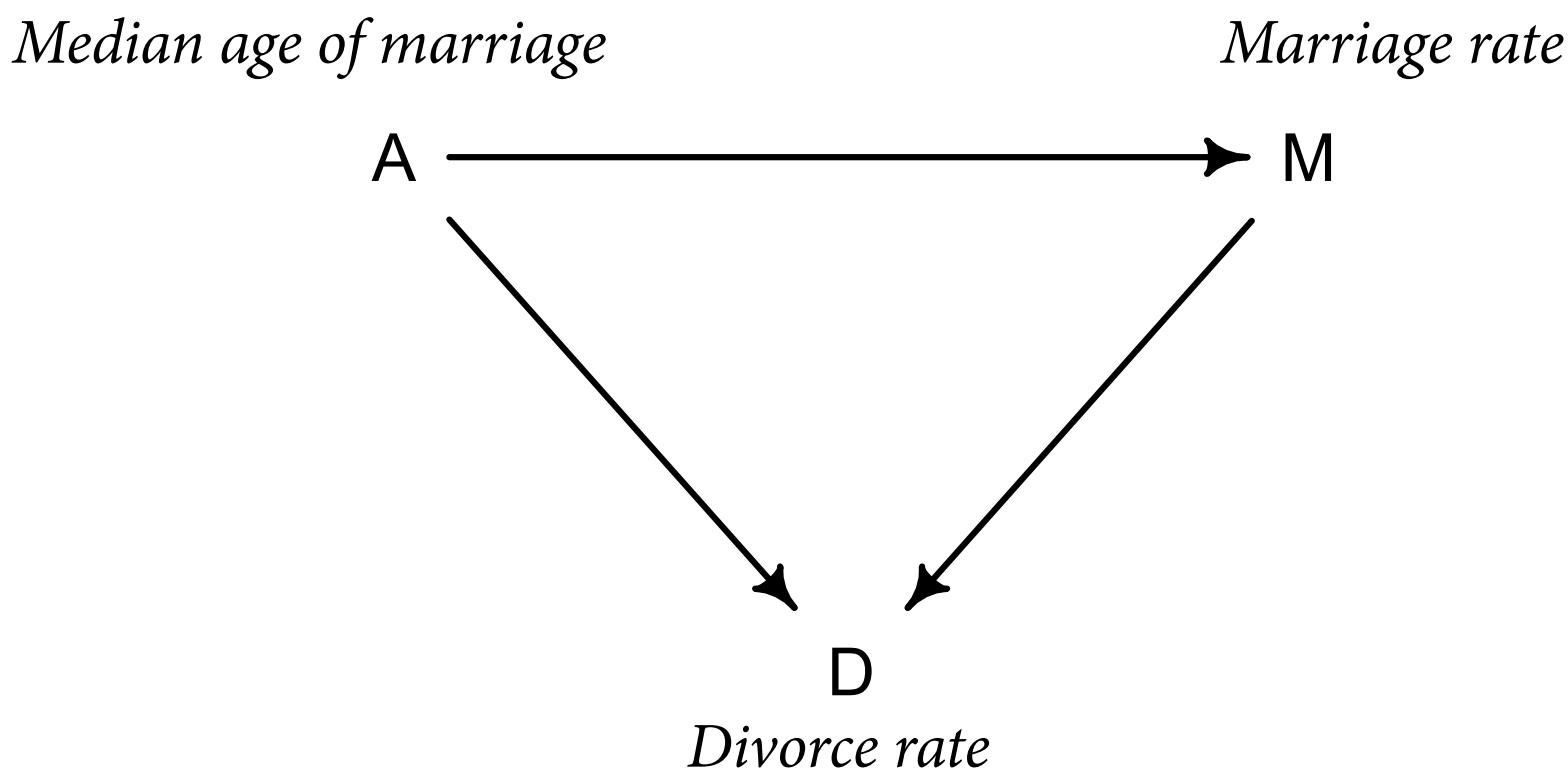
Causal Inference

- Effective inference requires
  - (1) Statistical model
  - (2) Causal model
  - (3) Decision model
- Examples of causal phenomena:
  - Hidden mediation
  - Collider bias
  - M-bias
  - Instrumental variables

They're good DAGs, Brent

- Directed Acyclic Graphs — tools for causal models
  - Directed: Arrows
  - Acyclic: Arrows don't make loops
  - Graphs: Nodes and edges
- Unlike statistical model, has causal implications





Implications:

(1) M is a function of A

(2) D is a function of A and M

(3) The total causal effect of A has two *paths*:

(a) A –> M –> D

(b) A –> D

Ye Olde Causal Alchemy

The Four Elemental Confounds

The Fork

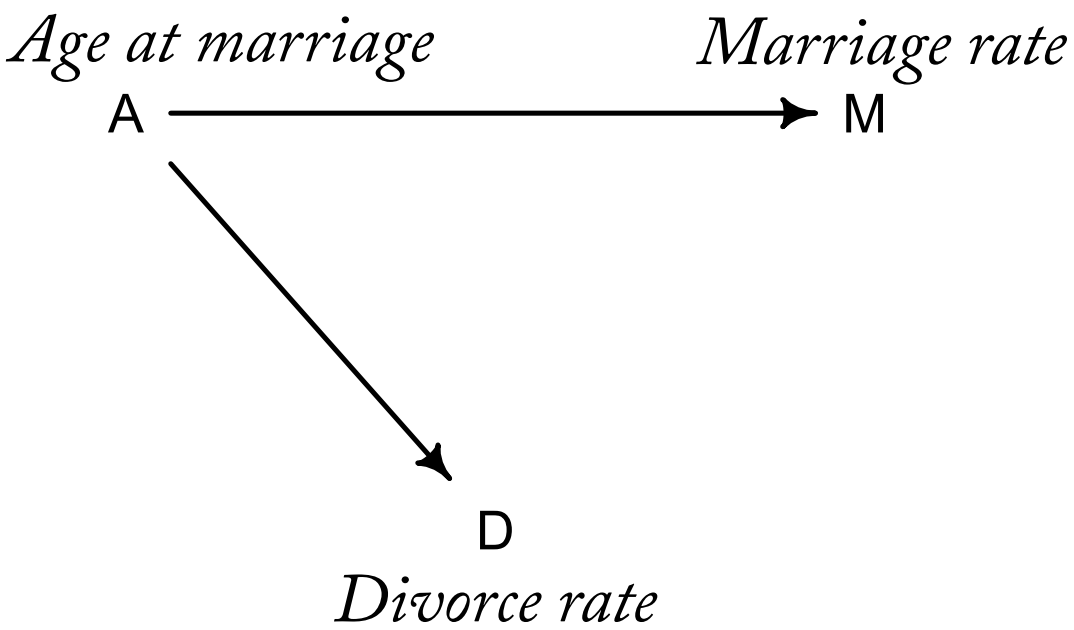
The Pipe

The Collider

The Descendant

The Confounding Fork

Z IS A COMMON CAUSE OF X AND Y



DE-CONFOUNDING!
CONDITIONING ON Z REMOVES DEPENDENCY BETWEEN X AND Y

$X \perp\!\!\!\perp Y \mid Z$

The Perplexing Pipe

X CAUSES Z CAUSES Y

Z MEDIATES ASSOCIATION BETWEEN X AND Y

CONDITIONING ON Z REMOVES DEPENDENCY BETWEEN X AND Y:
$X \perp\!\!\!\perp Y \mid Z$

DATA DO NOT DISTINGUISH FROM FORK!
$X \perp\!\!\!\perp Y \mid Z$ IN BOTH

The Explosive Collider

X AND Y JOINTLY CAUSE Z

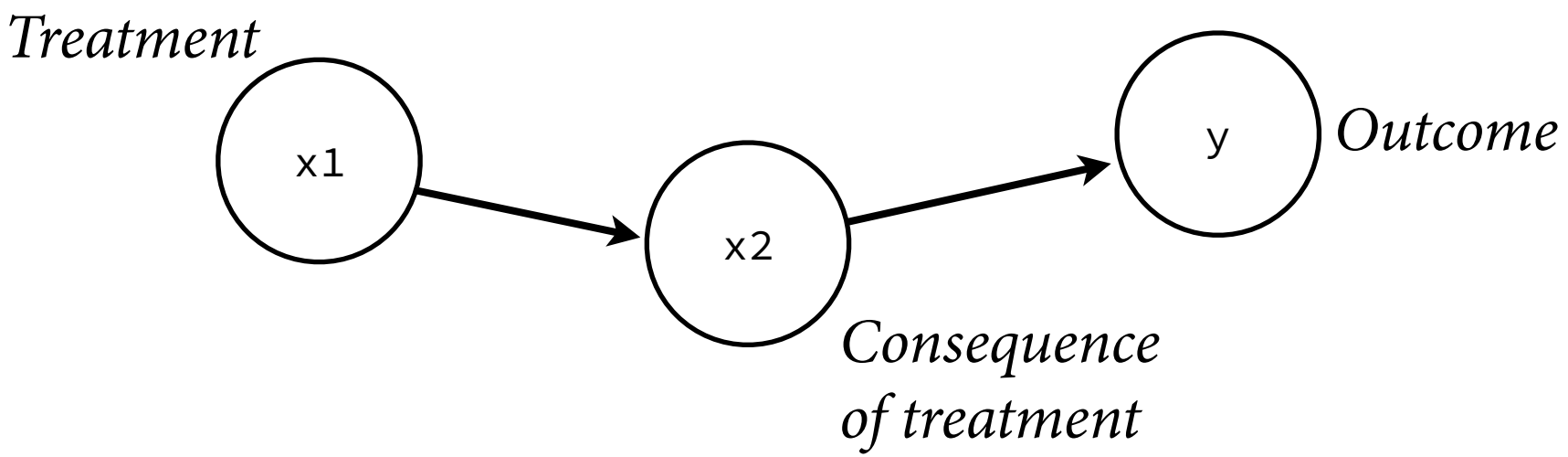
X AND Y INDEPENDENT

CONDITIONING ON Z
CREATES DEPENDENCY
BETWEEN X AND Y

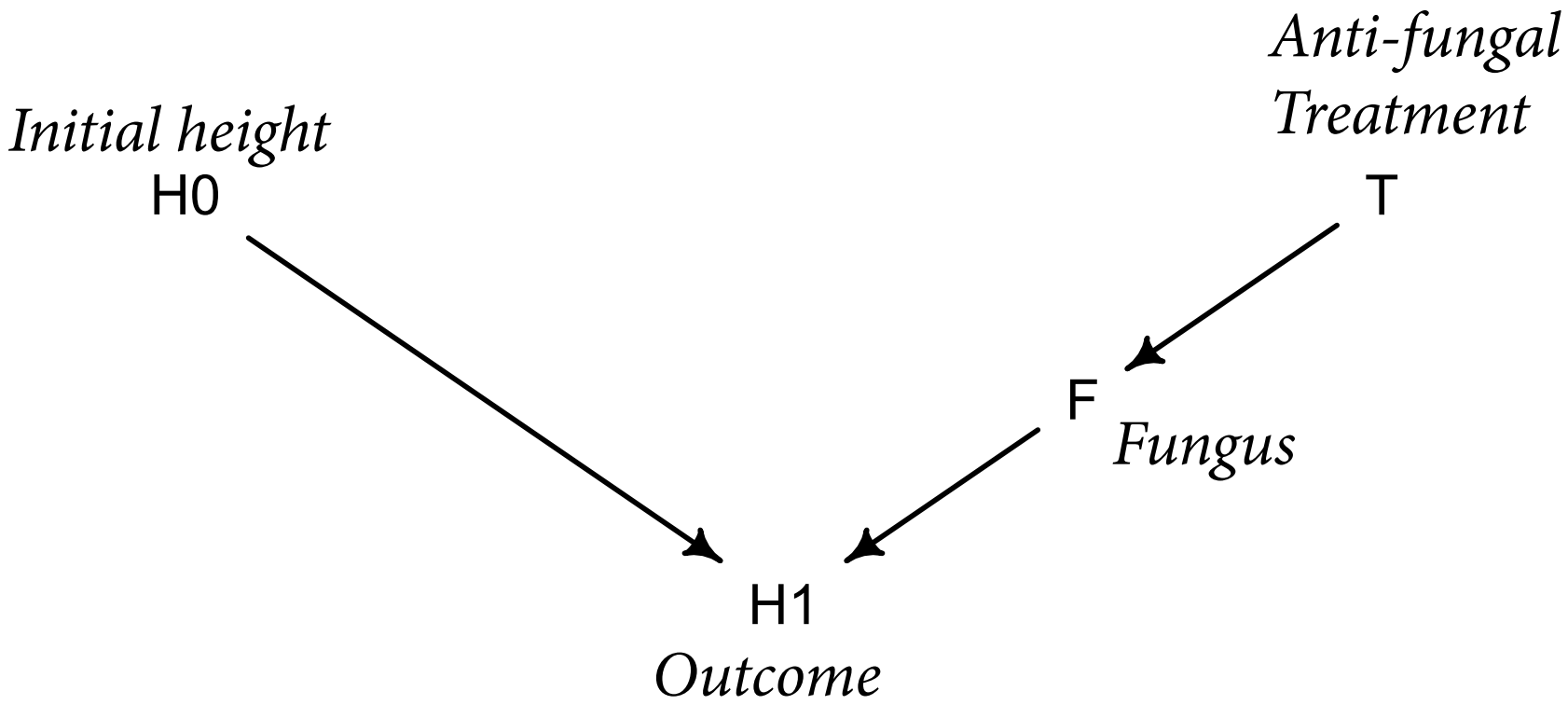
LEARNING X AND Z REVEALS Y

Post-treatment bias

- The pipe confounds when we ignore it
- *Post-treatment bias:* Controlling for consequence of treatment statistically knocks out treatment

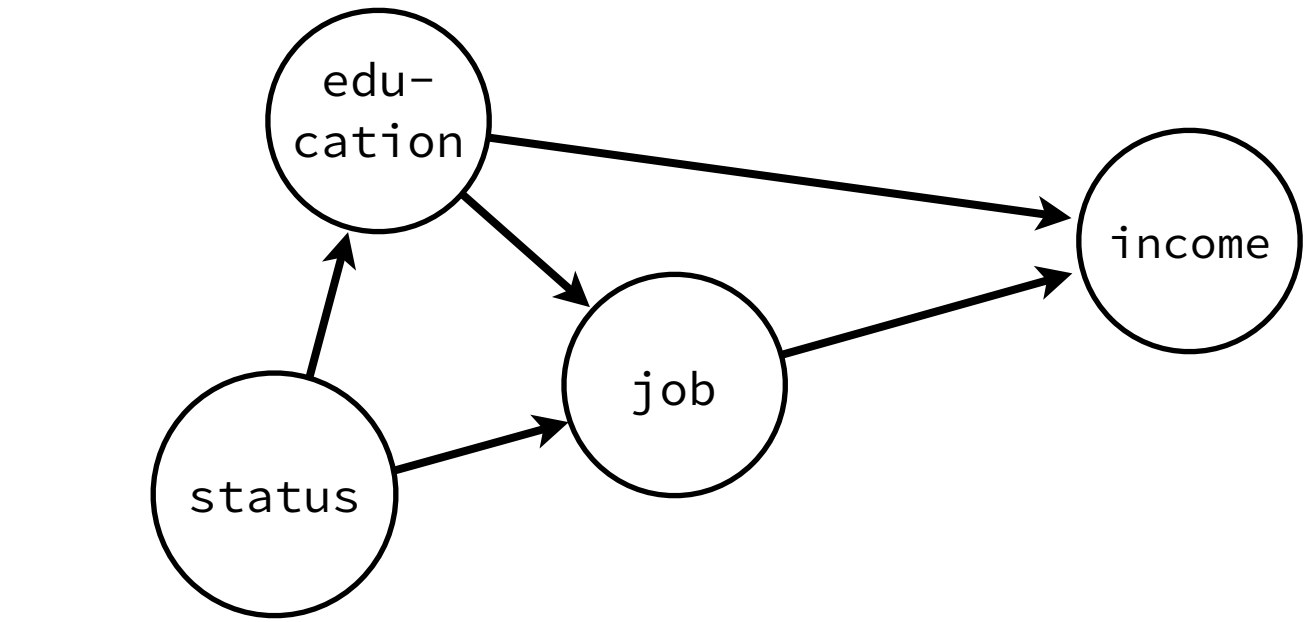


Post-treatment bias



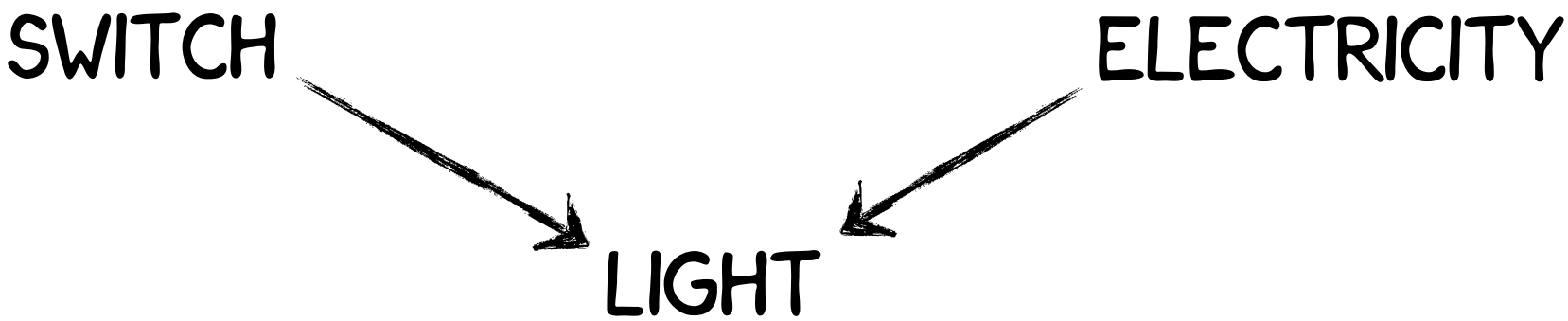
Post-treatment bias

Observational studies harder

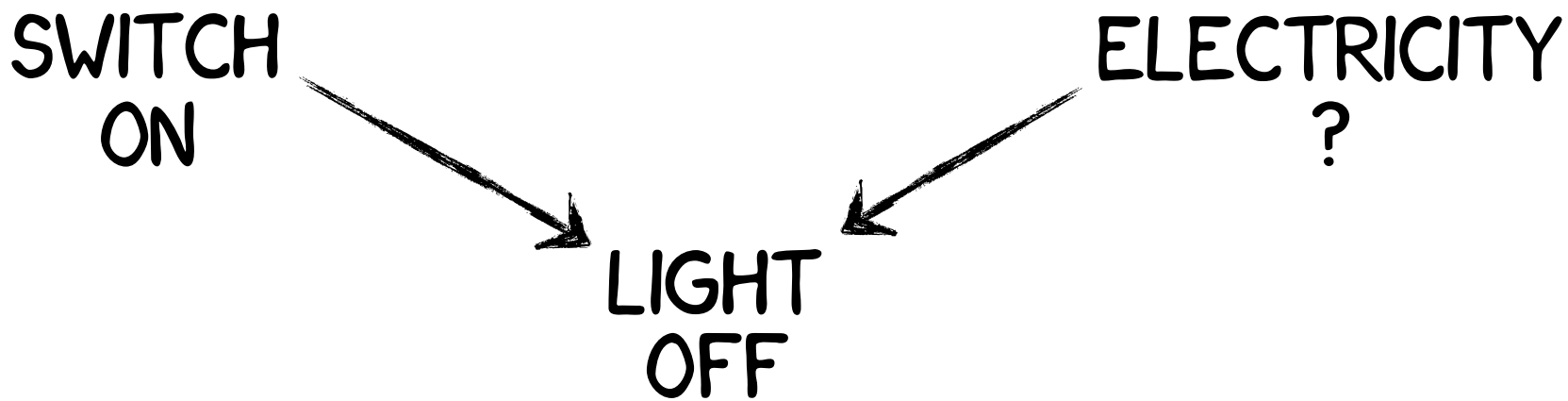


Controlling for every available variable likely to block a pipe someplace.

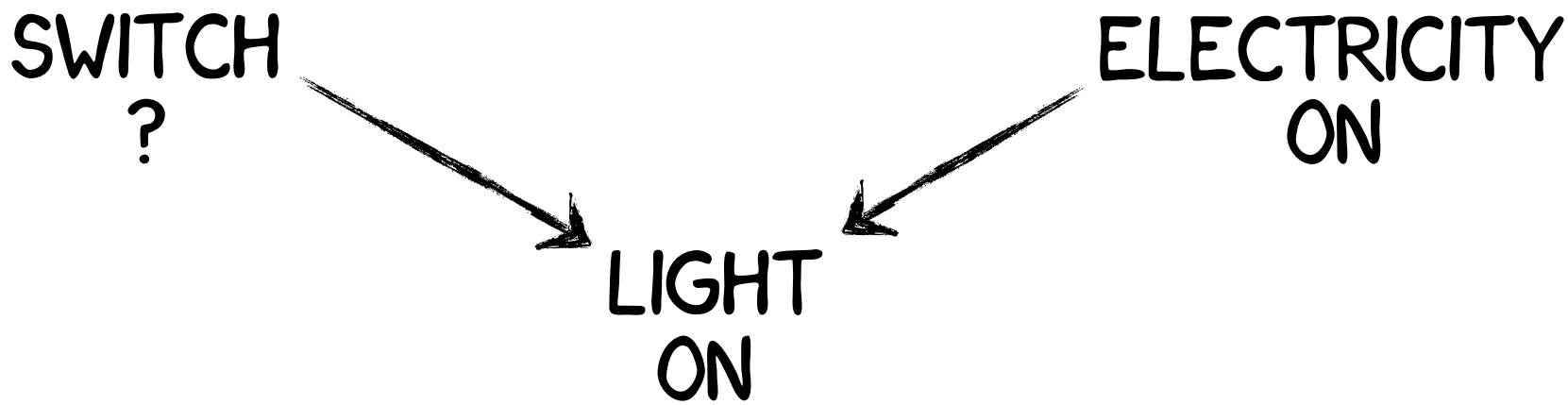
Conditioning on a Collider



Conditioning on a Collider



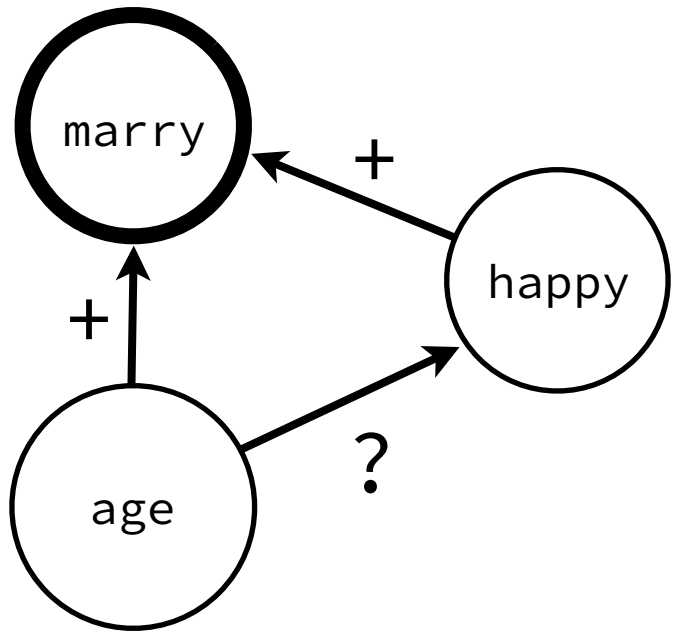
Conditioning on a Collider



Collider confounding

Conditioning on collider is like selecting on sub-population.

Are older people less happy? Should we control for marriage status?



Collider simulation

- Assumptions:
  - 20 people born each year
  - Uniform happiness at birth, never changes
  - At 18 years old, eligible to marry. Probability of marriage in each year proportional to happiness.
  - Married people remain married until death.
  - At age 65, move to south coast of Spain.

R code 6.22

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

```
'data.frame': 1300 obs. of 3 variables:
           mean    sd  5.5% 94.5%     histogram
age       33.0 18.77  4.00 62.00 ▇▇▇▇▇▇▇▇▇▇
married    0.3  0.46  0.00  1.00 ▇▁▁▁▁▁▁▁▂
happiness  0.0  1.21 -1.79  1.79   ▅▇▅▅▇▅
```

Collider of sorrow

R code 6.24

```
d2$mid <- d2$married + 1
m6.9 <- 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(m6.9,depth=2)
```

single

married

```
        mean   sd  5.5% 94.5%
a[1]  -0.23 0.06 -0.34 -0.13
a[2]   1.26 0.08  1.12  1.40
bA    -0.75 0.11 -0.93 -0.57
sigma  0.99 0.02  0.95  1.03
```

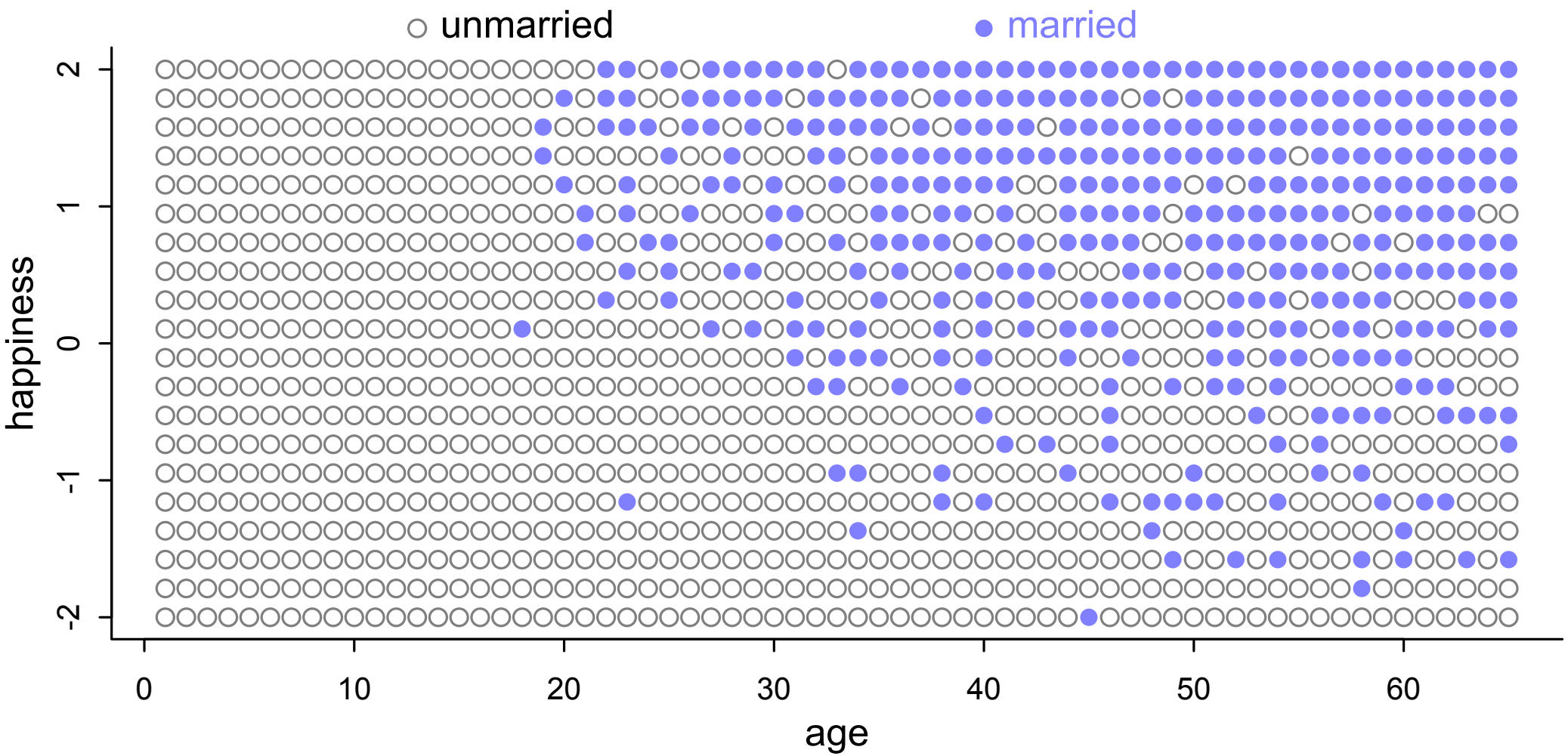


Figure 6.5

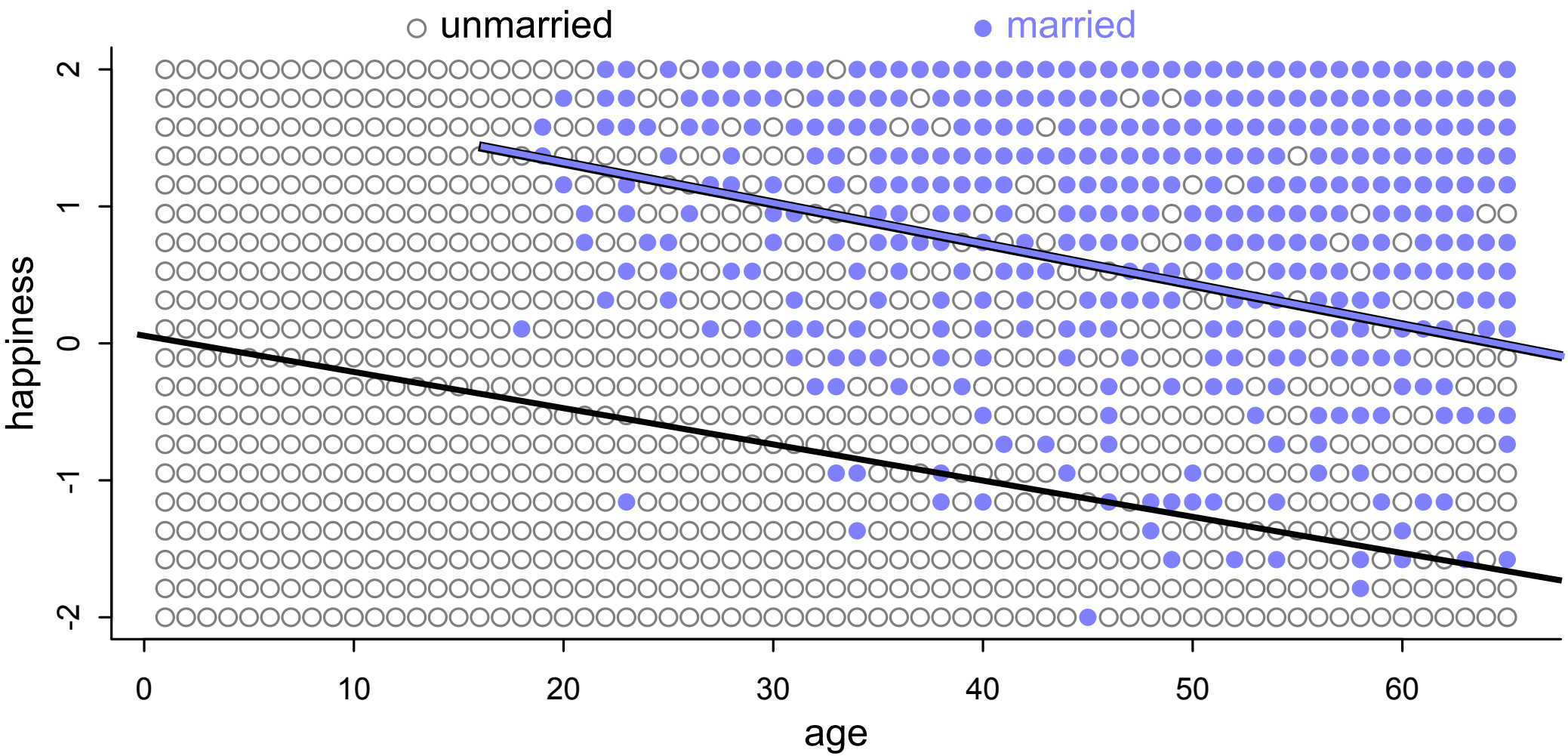
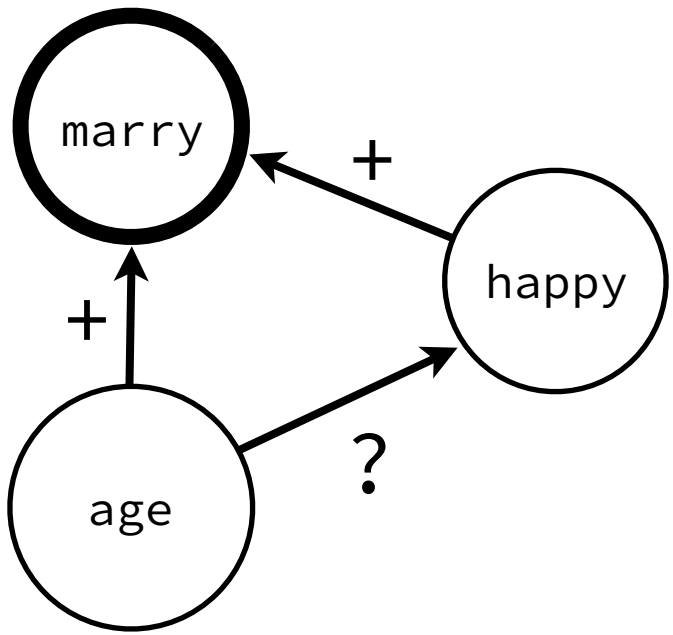


Figure 6.5

Collider confounding

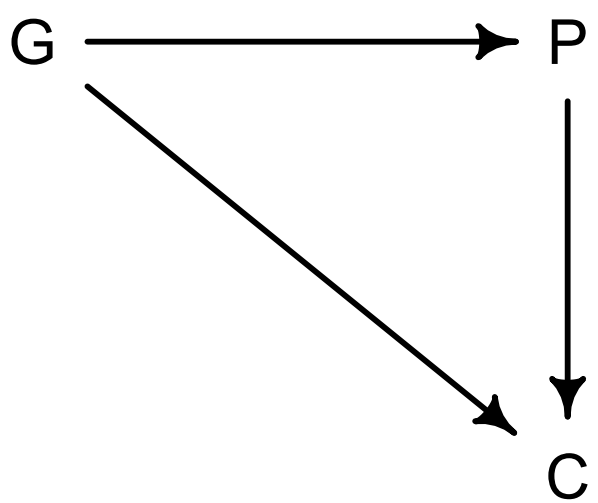
Are older people less happy? Controlling for marriage status creates a confound.

Cannot know whether to control for some variable, without a causal model.



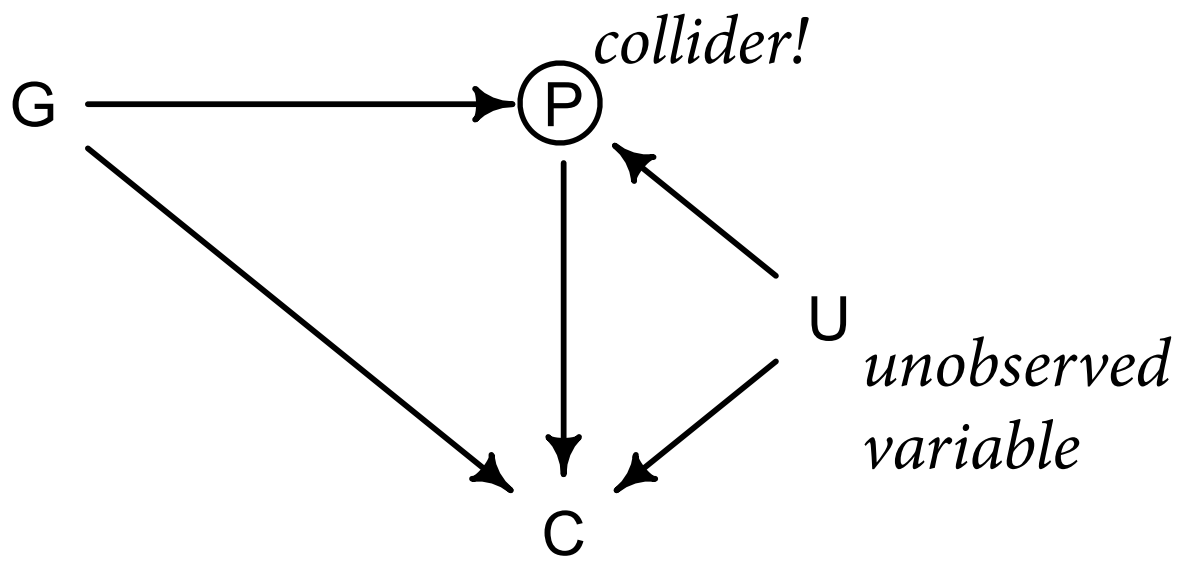
The Haunted DAG

- Unmeasured variables can also create colliders
- Example: Influence of grandparents (G) and parents (P) on education of children (C)



The Haunted DAG

- Unmeasured variables can also create colliders
- Example: Influence of grandparents (G) and parents (P) on education of children (C)



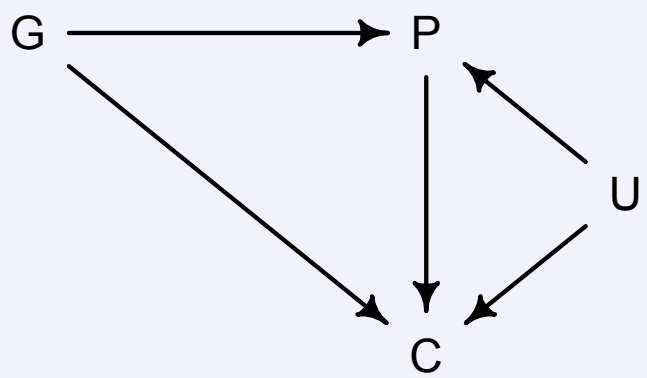
Simulated haunting

R code 6.26

```
N <- 200  # number of grandparent-parent-child triads
b_GP <- 1 # direct effect of G on P
b_GC <- 0 # direct effect of G on C
b_PC <- 1 # direct effect of P on C
b_U <- 2  # direct effect of U on P and C
```

R code 6.27

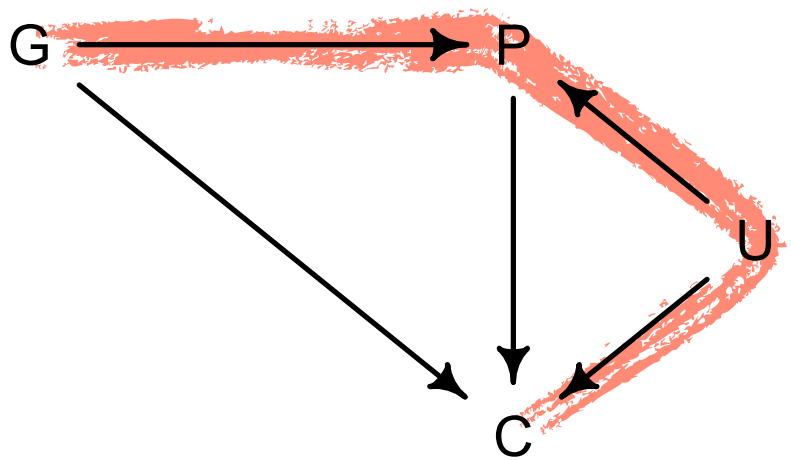
```
set.seed(1)
U <- 2*rbern( N , 0.5 ) - 1
G <- rnorm( N )
P <- rnorm( N , b_GP*G + b_U*U )
C <- rnorm( N , b_PC*P + b_GC*G + b_U*U )
d <- data.frame( C=C , P=P , G=G , U=U )
```



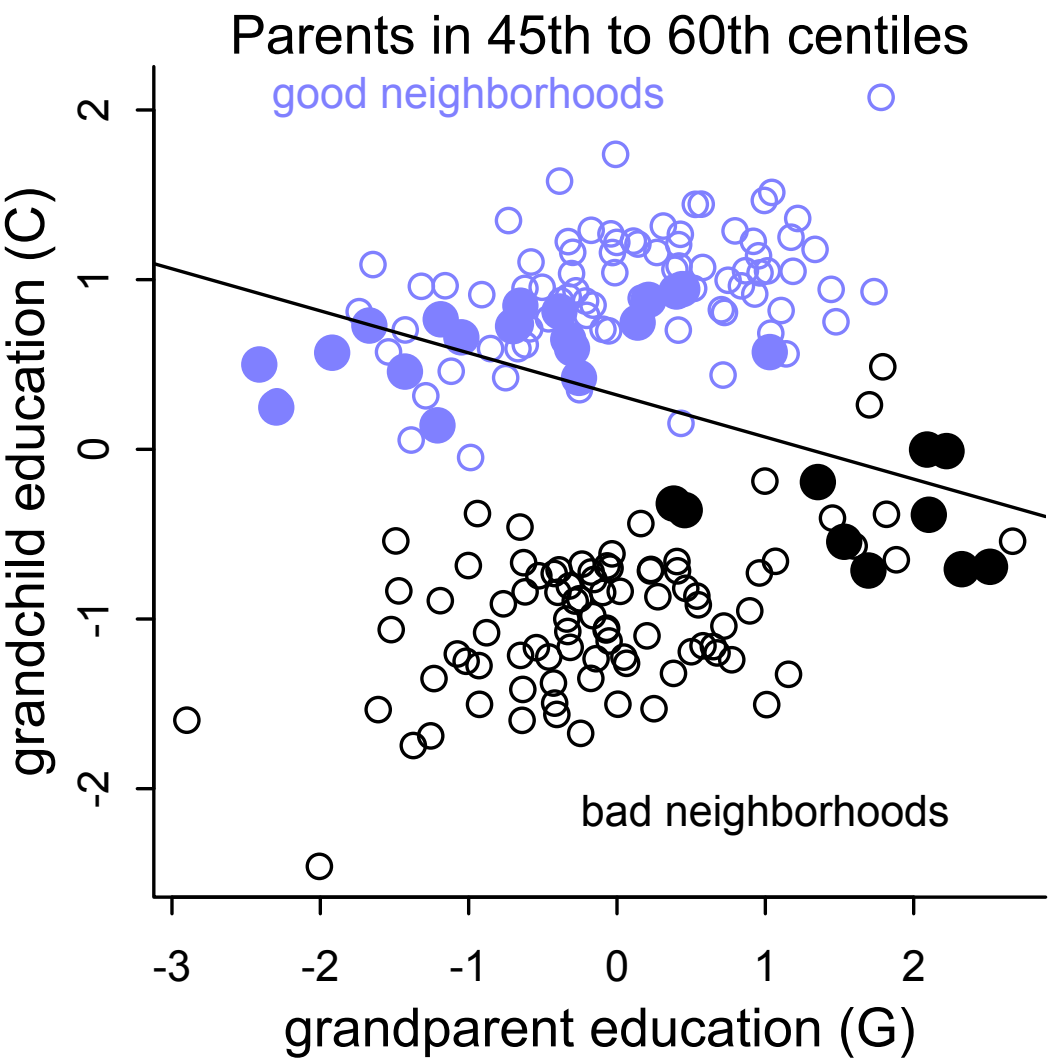
Simulated haunting

- Conditioning on parents distorts inference about grandparents
- Reason: Opens a “backdoor” through U to C

```
m6.11 <- quap(
    alist(
        C ~ dnorm( mu , sigma ),
        mu <- a + b_PC*P + b_GC*G,
        a ~ dnorm( 0 , 1 ),
        c(b_PC,b_GC) ~ dnorm( 0 , 1 ),
        sigma ~ dexp( 1 )
    ), data=d )
precis(m6.11)
```




```
        mean   sd  5.5% 94.5%
a      -0.12 0.10 -0.28  0.04
b_PC    1.79 0.04  1.72  1.86
b_GC   -0.84 0.11 -1.01 -0.67
sigma   1.41 0.07  1.30  1.52
```



Consider those P in 45-60th centile of education.

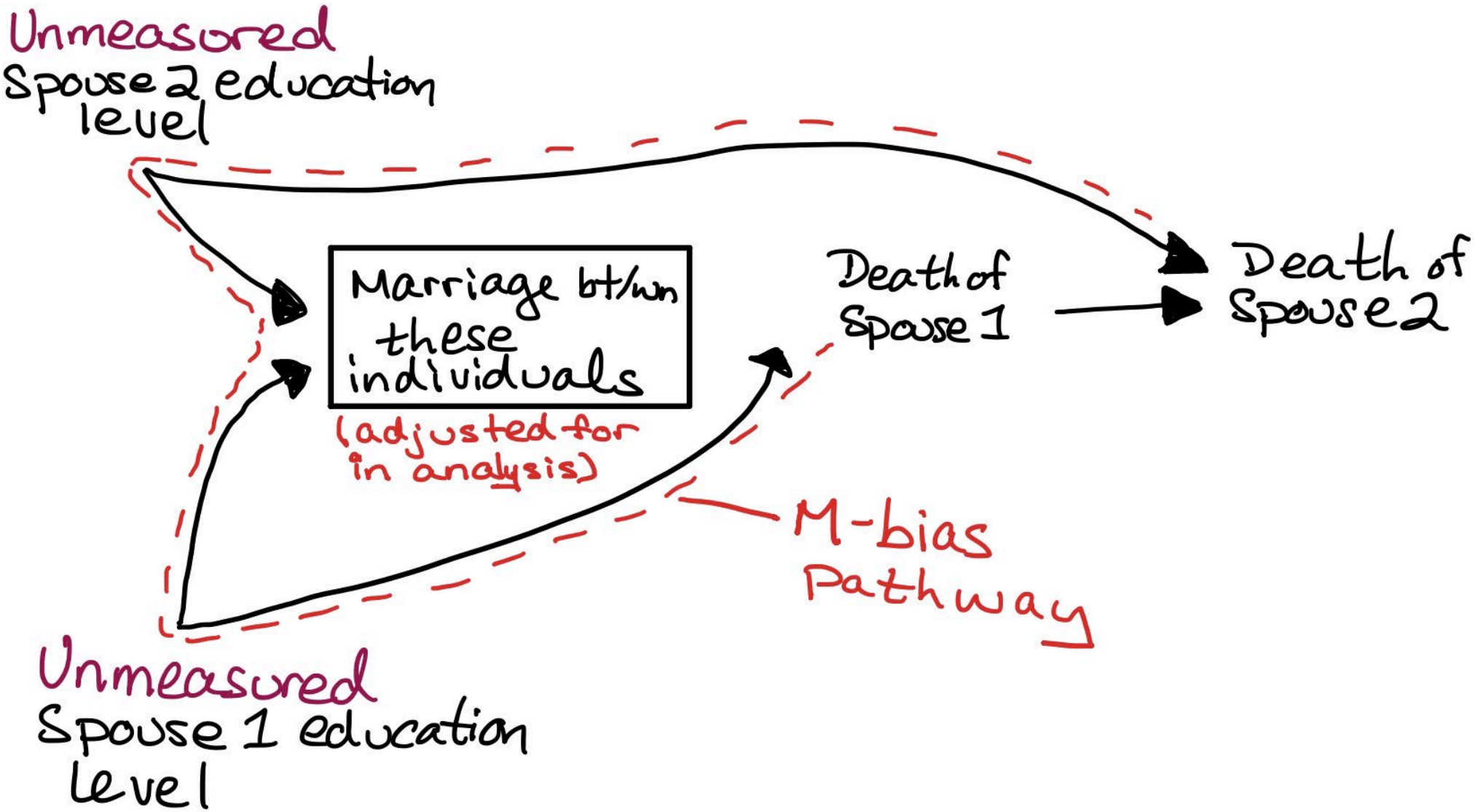
P in good neighborhoods must have had *less* educated G.

P in bad neighborhoods must have had *more* educated G.

Otherwise they wouldn't all be in same quantile.

Figure 6.6

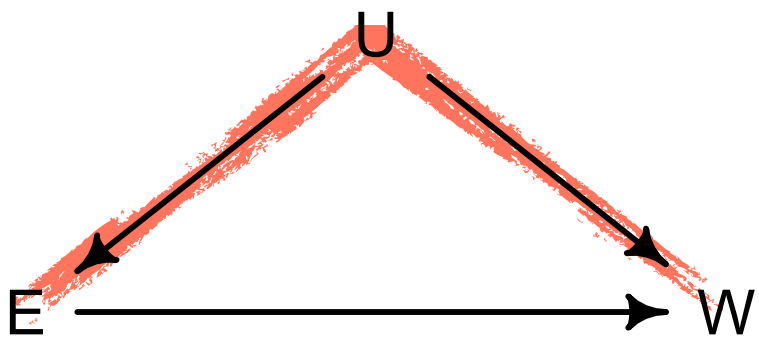
M-bias

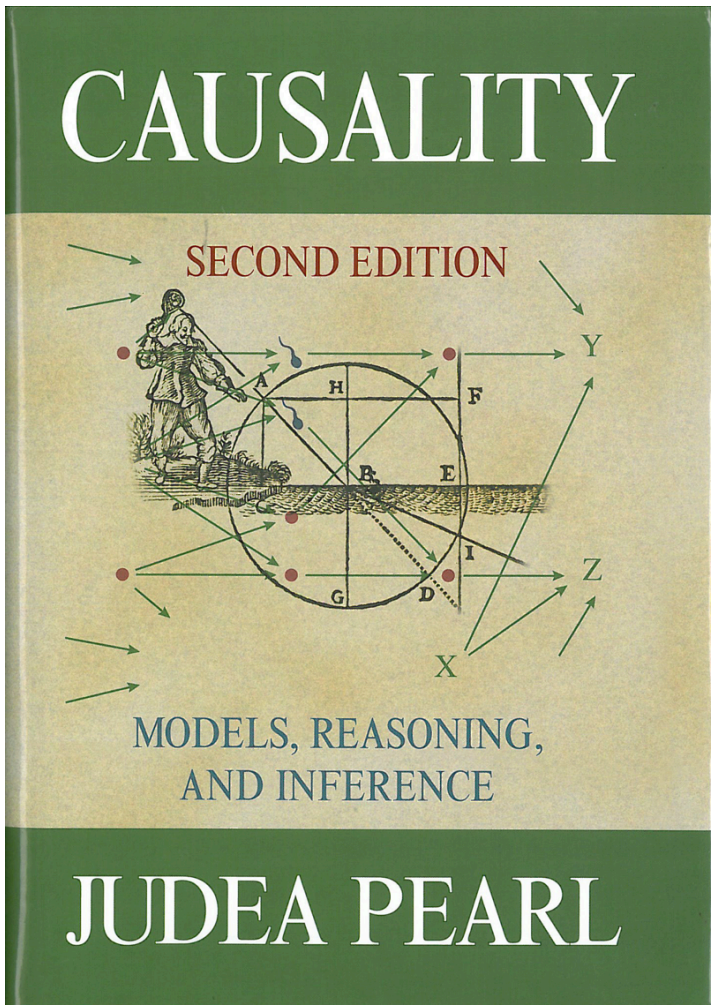


https://twitter.com/EpiEllie/status/1109656087958040577

Shutting the back door

- What ties these examples together:
- The **back-door criterion**: Confounding caused by existence of open back door paths from X to Y
- If you know your elements, you know how to open/close each of them





Ye Olde Causal Alchemy

The Four Elemental Confounds

The Fork

The Pipe

The Collider

The Descendant

The Fork

Open unless you
condition on Z

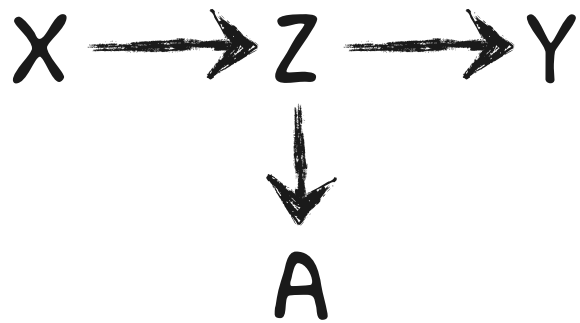
The Pipe

Open unless you
condition on Z

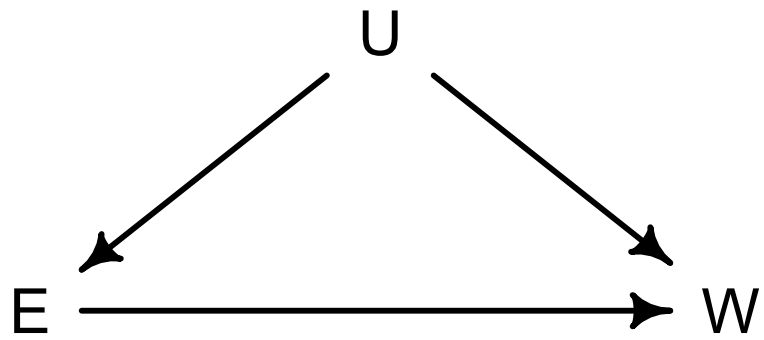
The Collider

Closed until you
condition on Z

The Descendant



Conditioning on A is
like conditioning on Z

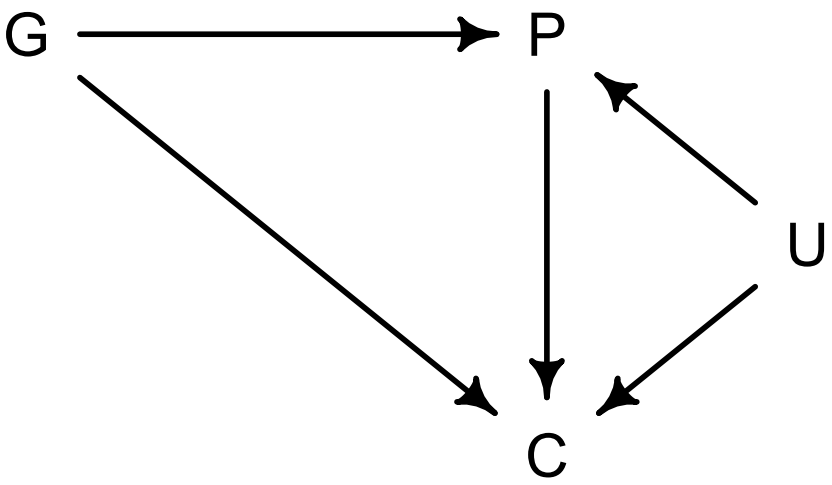


Two paths from E to W:

(1) E → W

(2) E ← U → W

Close 2nd path by conditioning on U, closing the pipe.



3 paths from G to C:

(1) G → C

(2) G → P → C

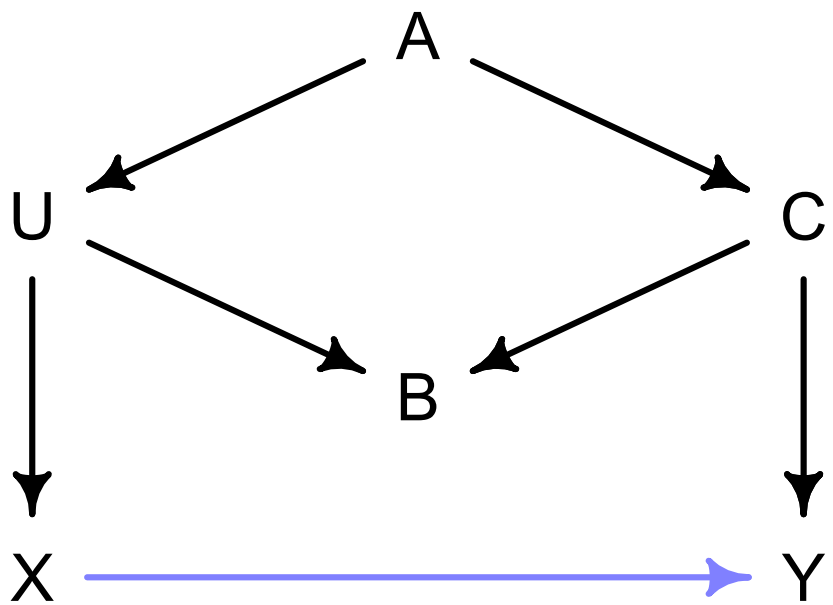
(3) G → P ← U → C

Condition on P:

Closes (2) but opens (3)

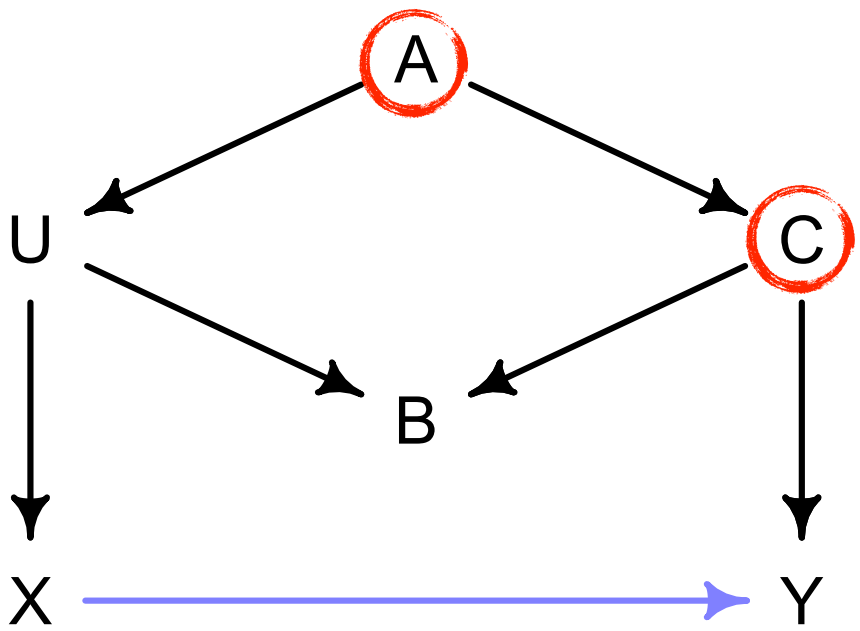
Something more interesting

- Which variables, if any, should you condition on to infer X → Y?
- Procedure: (1) Find all paths. (2) Open/close as necessary.



Something more interesting

- Which variables, if any, should you condition on to infer X → Y?
- Condition on A or C. Do not condition on B.



(1) $X \leftarrow U \leftarrow A \rightarrow C \rightarrow Y$

This path is open.

(2) $X \leftarrow U \rightarrow B \leftarrow C \rightarrow Y$

This path is closed.

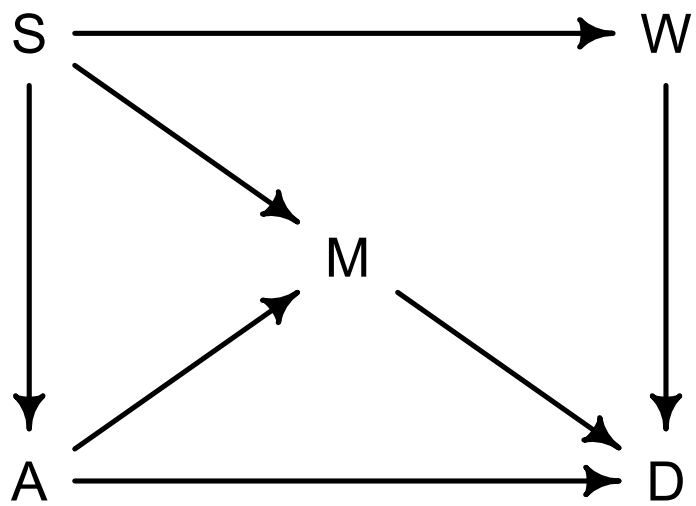
Implied conditional independence

- Given DAG, can test some implications

```
impliedConditionalIndependencies( dag_6.2 )
```

R code 6.36

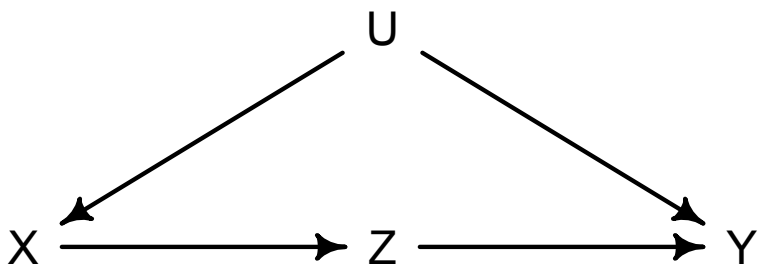
```
A _||_ W | S
D _||_ S | A, M, W
M _||_ W | S
```



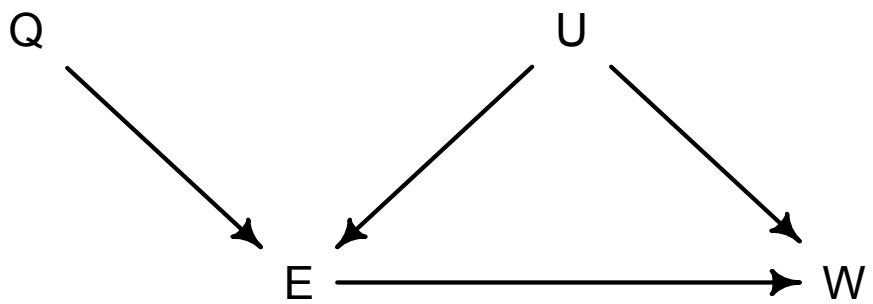
(1) A and W independent, conditioning on S
(2) D and S independent, conditioning on A, M, & W
(3) M and W independent, conditioning on S

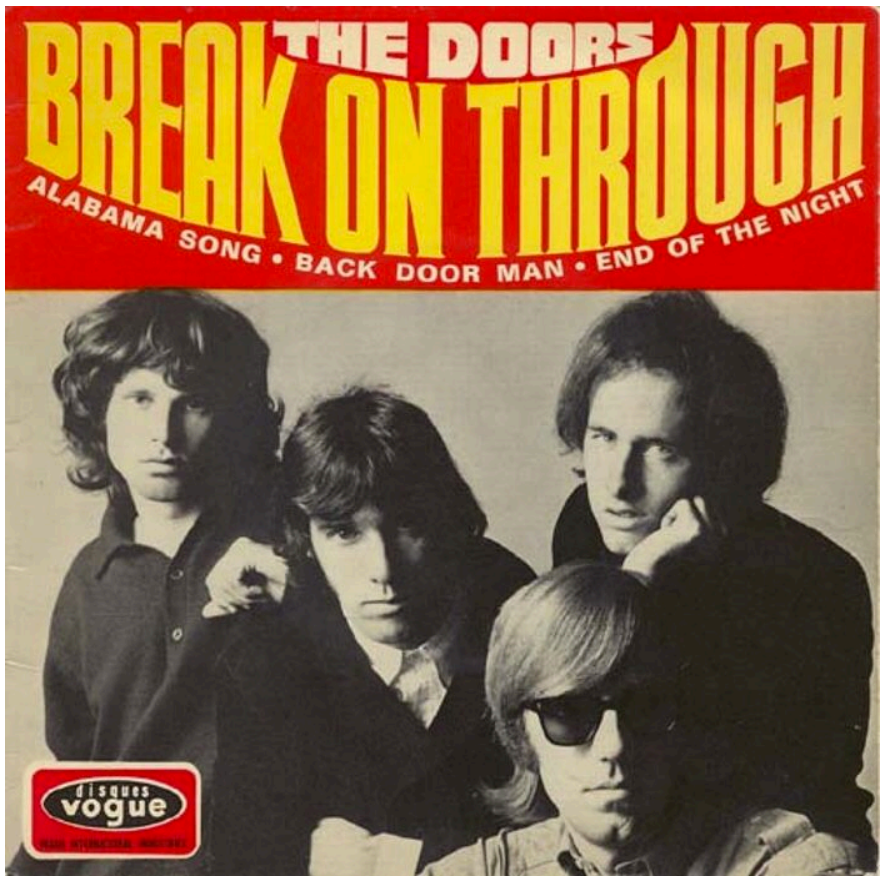
More than the Back Door

- Closing back doors is not the only option
- Front-door criterion



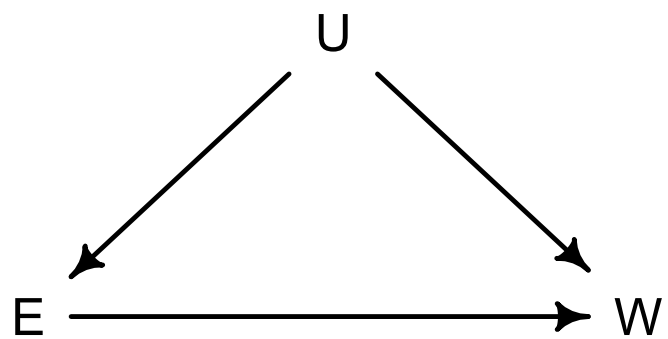
- Instrumental variables





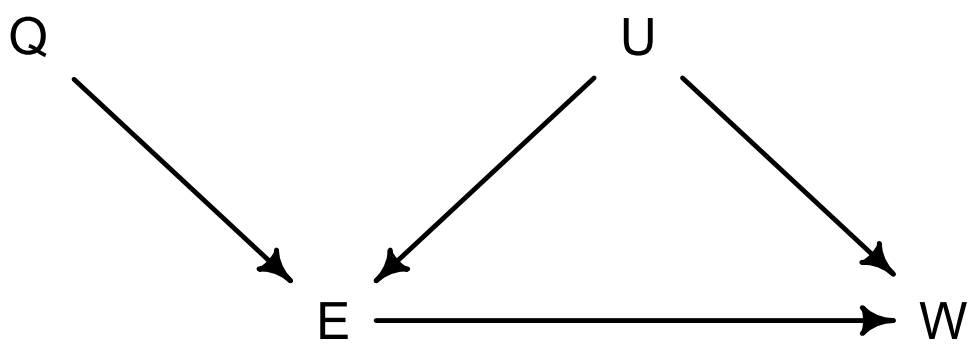
Instrumental variables

- Imagine trying to estimate influence of education on wages — lots of unmeasured confounds.



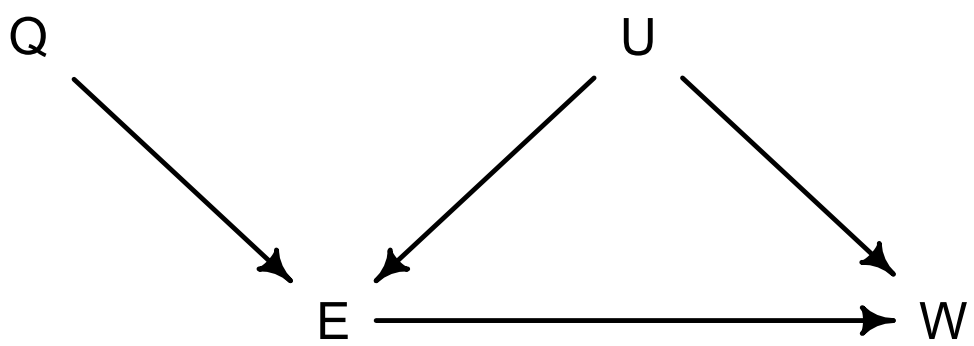
Instrumental variables

- Instrument: A variable that influences exposure (E) but not outcome (W)
- Here: Birthday position in year (Q). People born earlier in year consume less education.
  - Start school later (biologically)
  - Eligible to quit school earlier (biologically)



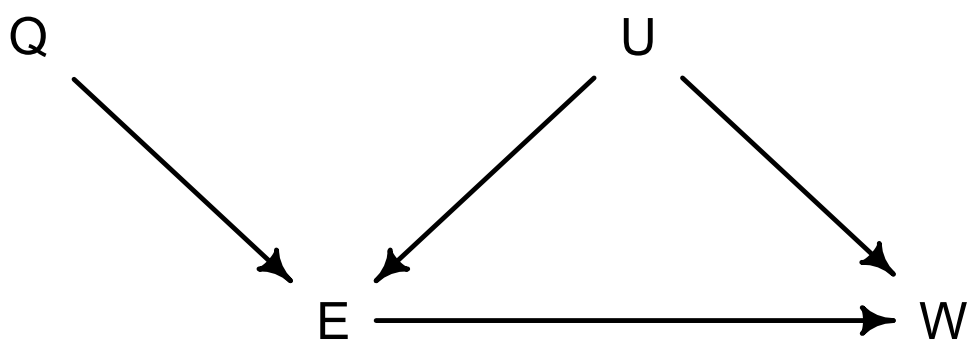
Instrumental variables

- Instrument: A variable that influences exposure (E) but not outcome (W)
- How could this help us?
- Gives us information about U
- E and W correlated, due to U
- Q helps us measure that correlation



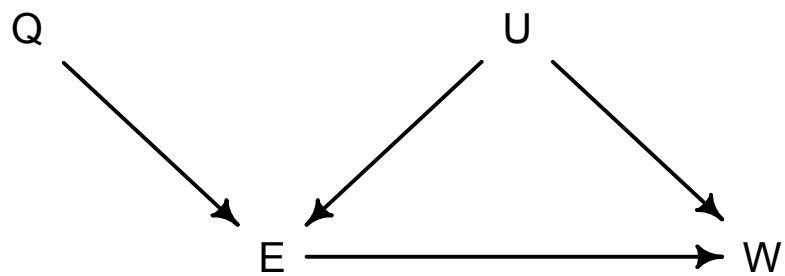
Instrumental variables

- Example:
- People born in 1st quarter (Q1) of year consume 10 years of education on average
- A specific person born in Q1 consumed 12 years
- Gives us information about unmeasured U

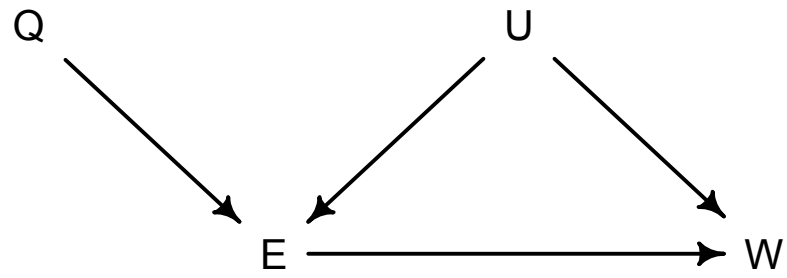


Instrumental variables

- Another perspective:
- Q is a “natural experiment”
- Q assigns E, as if by experimenter giving education pills
- But individuals are uncooperative and don’t always take their pills => imperfect randomization
- Many (most?) real “experiments” are actually like this, have *intent to treat*



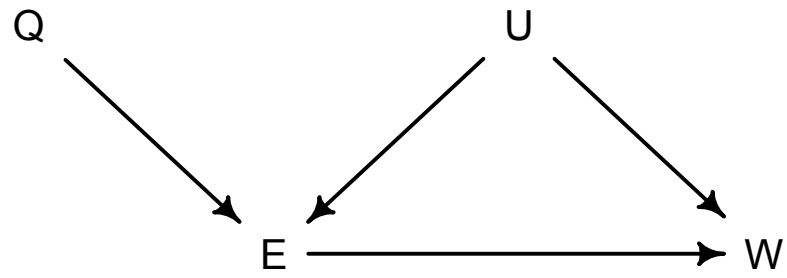
Simulated instrument



$$W_i \sim \text{Normal}(\mu_{\text{W},i}, \sigma_{\text{W}})$$
$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i + U_i$$

[Wage model]

Simulated instrument



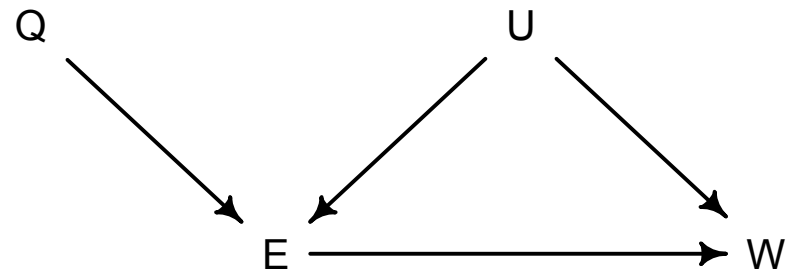
$$W_i \sim \text{Normal}(\mu_{\text{W},i}, \sigma_{\text{W}})$$
$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i + U_i$$

[Wage model]

$$E_i \sim \text{Normal}(\mu_{\text{E},i}, \sigma_{\text{E}})$$
$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i + U_i$$

[Education model]

Simulated instrument



$$W_i \sim \text{Normal}(\mu_{\text{W},i}, \sigma_{\text{W}})$$
$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i + U_i$$

[Wage model]

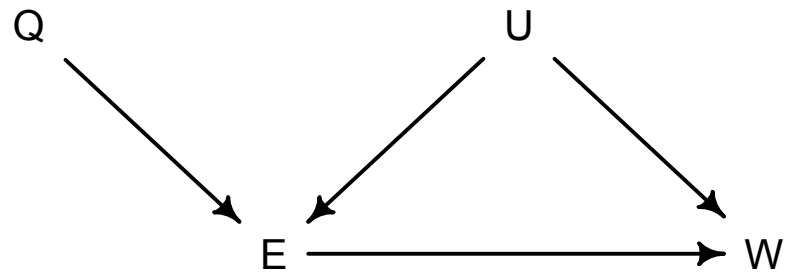
$$E_i \sim \text{Normal}(\mu_{\text{E},i}, \sigma_{\text{E}})$$
$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i + U_i$$

[Education model]

$$Q_i \sim \text{Bernoulli}(0.25)$$

[Birth model]

Simulated instrument



$$W_i \sim \text{Normal}(\mu_{\text{W},i}, \sigma_{\text{W}})$$
$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i + U_i$$

[Wage model]

$$E_i \sim \text{Normal}(\mu_{\text{E},i}, \sigma_{\text{E}})$$
$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i + U_i$$

[Education model]

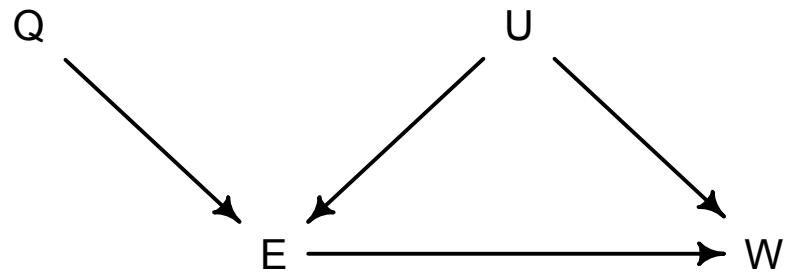
$$Q_i \sim \text{Bernoulli}(0.25)$$

[Birth model]

$$U_i \sim \text{Normal}(0, 1)$$

[Confound model]

Simulated instrument




```
set.seed(73)
N <- 500
U_sim <- rnorm( N )
Q_sim <- sample( 1:4 , size=N , replace=TRUE )
E_sim <- rnorm( N , U_sim + Q_sim )
W_sim <- rnorm( N , U_sim + 0*E_sim )
dat_sim <- list(
    W=standardize(W_sim) ,
    E=standardize(E_sim) ,
    Q=standardize(Q_sim) )
```

Simulated instrument

- E —> W confounded

R code 14.24

```
m14.4 <- ulam(
    alist(
        W ~ dnorm( mu , sigma ),
        mu <- aW + bEW*E,
        aW ~ dnorm( 0 , 0.2 ),
        bEW ~ dnorm( 0 , 0.5 ),
        sigma ~ dexp( 1 )
    ) , data=dat_sim , chains=4 , cores=4 )
precis( m14.4 )
```

```
      mean   sd  5.5% 94.5% n_eff Rhat
aW    0.00 0.04 -0.07  0.07  2028    1
bEW   0.39 0.04  0.32  0.45  2032    1
sigma 0.93 0.03  0.88  0.97  1999    1
```

Instrumentality

- Think of pairs of (W,E) values as sampled from a common distribution with some covariance structure:

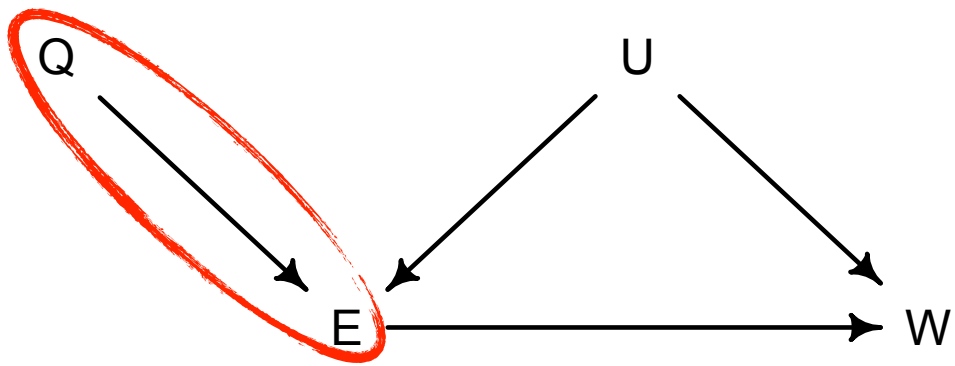
$$\begin{pmatrix} W_i \\ E_i \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} \mu_{\text{W},i} \\ \mu_{\text{E},i} \end{pmatrix}, \text{S} \right)$$
$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i$$
$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i$$

Instrumentality

$$\begin{pmatrix} W_i \\ E_i \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} \mu_{\text{W},i} \\ \mu_{\text{E},i} \end{pmatrix}, \text{S} \right)$$

$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i$$

$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i$$

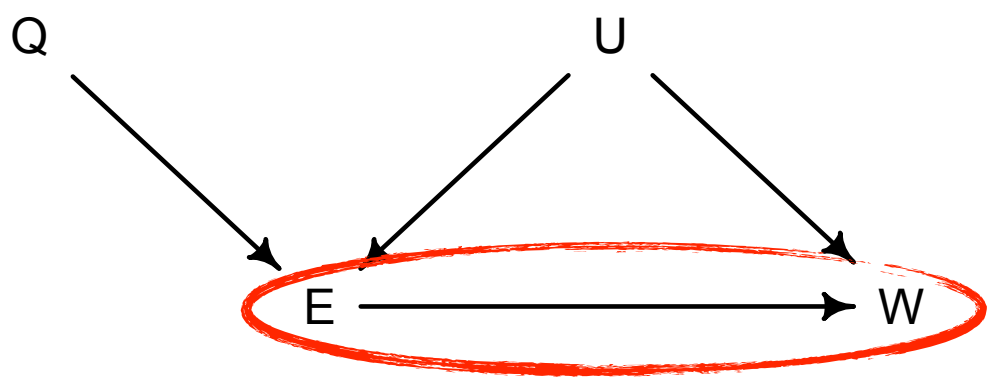


Instrumentality

$$\begin{pmatrix} W_i \\ E_i \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} \mu_{\text{W},i} \\ \mu_{\text{E},i} \end{pmatrix}, \text{S} \right)$$

$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i$$

$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i$$

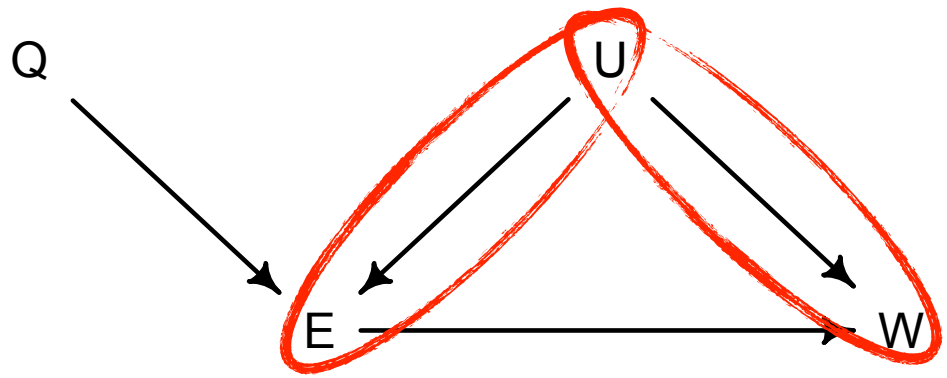


Instrumentality

$$\begin{pmatrix} W_i \\ E_i \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} \mu_{\text{W},i} \\ \mu_{\text{E},i} \end{pmatrix}, \text{S} \right)$$

$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i$$

$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i$$



$$\begin{pmatrix} W_i \\ E_i \end{pmatrix} \sim \text{MVNormal}\left(\begin{pmatrix} \mu_{\text{W},i} \\ \mu_{\text{E},i} \end{pmatrix} , \text{S} \right)$$
$$\mu_{\text{W},i} = \alpha_{\text{W}} + \beta_{\text{EW}} E_i$$
$$\mu_{\text{E},i} = \alpha_{\text{E}} + \beta_{\text{QE}} Q_i$$

R code 14.25

```
m14.5 <- ulam(
    alist(
        c(W,E) ~ multi_normal( c(muW,muE) , Rho , Sigma ),
        muW <- aW + bEW*E,
        muE <- aE + bQE*Q,
        c(aW,aE) ~ normal( 0 , 0.2 ),
        c(bEW,bQE) ~ normal( 0 , 0.5 ),
        Rho ~ lkj_corr( 2 ),
        Sigma ~ exponential( 1 )
    ), data=dat_sim , chains=4 , cores=4 )
precis( m14.5 , depth=3 )
```

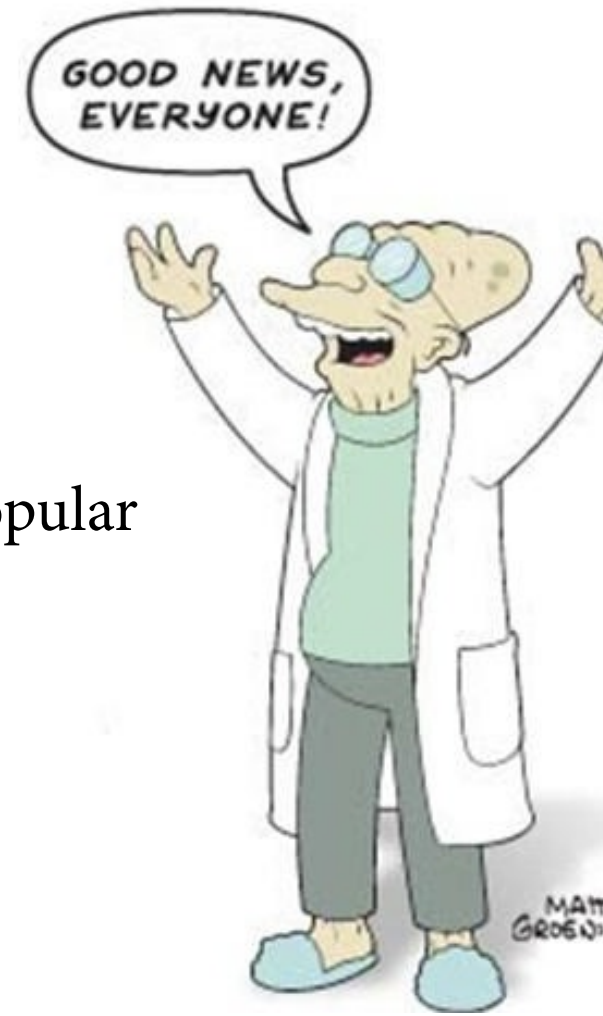
R code 14.25

```
m14.5 <- ulam(
    alist(
        c(W,E) ~ multi_normal( c(muW,muE) , Rho , Sigma ),
        muW <- aW + bEW*E,
        muE <- aE + bQE*Q,
        c(aW,aE) ~ normal( 0 , 0.2 ),
        c(bEW,bQE) ~ normal( 0 , 0.5 ),
        Rho ~ lkj_corr( 2 ),
        Sigma ~ exponential( 1 )
    ), data=dat_sim , chains=4 , cores=4 )
precis( m14.5 , depth=3 )
```

```
           mean   sd  5.5% 94.5% n_eff Rhat
aE         0.00 0.03 -0.05  0.05  1158    1
aW         0.00 0.04 -0.07  0.07  1400    1
bQE        0.63 0.03  0.58  0.69  1557    1
bEW       -0.03 0.07 -0.14  0.08  1010    1
Rho[1,1]   1.00 0.00  1.00  1.00   NaN  NaN
Rho[1,2]   0.53 0.05  0.45  0.60   987    1
Rho[2,1]   0.53 0.05  0.45  0.60   987    1
Rho[2,2]   1.00 0.00  1.00  1.00  1714    1
Sigma[1]   1.01 0.04  0.95  1.08  1028    1
Sigma[2]   0.77 0.03  0.73  0.81  1478    1
```

Causal inference hard but possible

- Explicitly state assumptions
- Experiments not required!
- Experiments not always practical & ethical
  - Disease, evolution, development, dynamics of popular music, global climate, economics, war
- Experiments must choose an intervention
  - Interventions influence many variables at once
  - Experimentally manipulate obesity?



Directed Acyclic Gaffes

- Don't get cocky
- DAGs are small world constructs
- Residual confounding:
  - Misclassification
  - Measurement error
  - Missingness
- DAGs can accommodate these problems, but maybe tell us there are no solutions
- Eventually need *real* models of the system

...USING A NON-CENTERED PARAMETRISATION IMPROVED R-HAT AND ESS FOR THAT PARAMETER AND ALLOWED ME TO GET RID OF THOSE DIVERGENCES!

I. — NOTICES SCIENTIFIQUES

Commandant BENOIT[1].

NOTE SUR UNE MÉTHODE DE RÉSOLUTION DES ÉQUATIONS NORMALES PROVENANT DE L'APPLICATION DE LA MÉTHODE DES MOINDRES CARRÉS A UN SYSTÈME D'ÉQUATIONS LINÉAIRES EN NOMBRE INFÉRIEUR A CELUI DES INCONNUES. — APPLICATION DE LA MÉTHODE A LA RÉSOLUTION D'UN SYSTÈME DEFINI D'ÉQUATIONS LINÉAIRES.

(Procédé du Commandant CHOLESKY[2].)

Le Commandant d'Artillerie Cholesky, du Service géographique de l'Armée, tué pendant la grande guerre, a imaginé, au cours de recherches sur la compensation des réseaux géodésiques, un procédé très ingénieux de résolution des équations dites *normales*, obtenues par application de la méthode des moindres carrés à des équations linéaires en nombre inférieur à celui des inconnues. Il en a conclu une méthode générale de résolution des équations linéaires.

Nous suivrons, pour la démonstration de cette méthode, la progression même qui a servi au Commandant Cholesky pour l'imaginer.

1. De l'Artillerie coloniale, ancien officier géodésien au Service géographique de l'Armée et au Service géographique de l'Indo-Chine, Membre du Comité national français de Géodésie et Géophysique.

2. Sur le Commandant Cholesky, tué à l'ennemi le 31 août 1918, voir la notice biographique insérée dans le volume du *Bulletin géodésique* de 1922 intitulé : *Union géodésique et géophysique internationale, Première Assemblée générale, Rome, mai 1922, Section de Géodésie*, Toulouse, Privat, 1922, in-8°, 241 p., pp. 159 à 161.

André-Louis Cholesky
(1875–1918)

I. — NOTICES SCIENTIFIQUES

Commandant BENOIT[1].

NOTE SUR UNE MÉTHODE DE RÉSOLUTION DES ÉQUATIONS NORMALES PROVENANT DE L'APPLICATION DE LA MÉTHODE DES MOINDRES CARRÉS A UN SYSTÈME D'ÉQUATIONS LI… CELUI DES INCO… THODE A LA R… D'ÉQUATIONS LI…

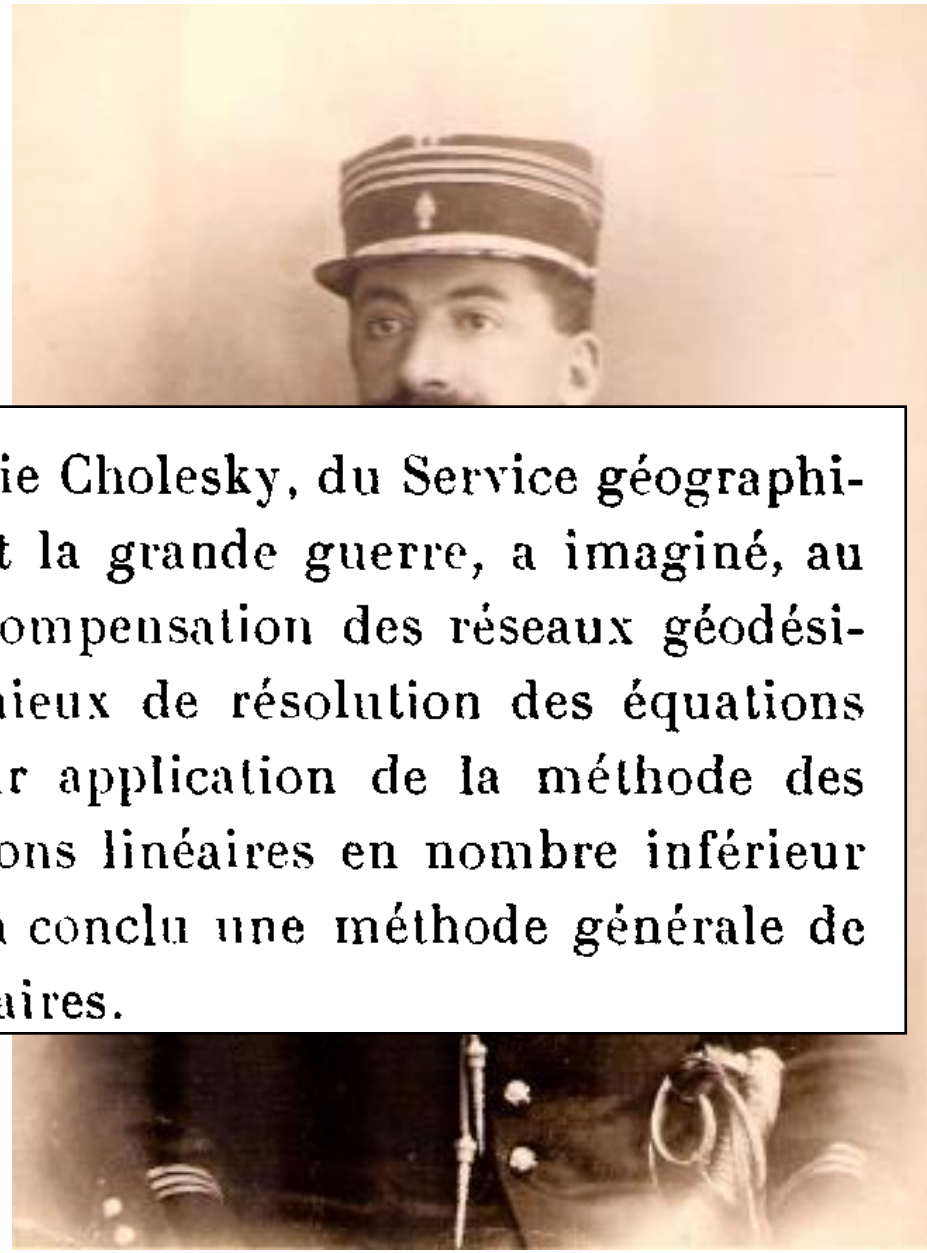
(Procédé…

Le Commandant d'… que de l'Armée, tué … cours de recherches s… ques, un procédé trè… dites *normales*, obte… moindres carrés à des… à celui des inconnues… résolution des équatio…

Nous suivrons, pour la démonstration de cette méthode, la progression même qui a servi au Commandant Cholesky pour l'imaginer.

1. De l'Artillerie coloniale, ancien officier géodésien au Service géographique de l'Armée et au Service géographique de l'Indo-Chine, Membre du Comité national français de Géodésie et Géophysique.

2. Sur le Commandant Cholesky, tué à l'ennemi le 31 août 1918, voir la notice biographique insérée dans le volume du *Bulletin géodésique* de 1922 intitulé : *Union géodésique et géophysique internationale, Première Assemblée générale, Rome, mai 1922, Section de Géodésie*, Toulouse, Privat, 1922, in-8°, 241 p., pp. 159 à 161.

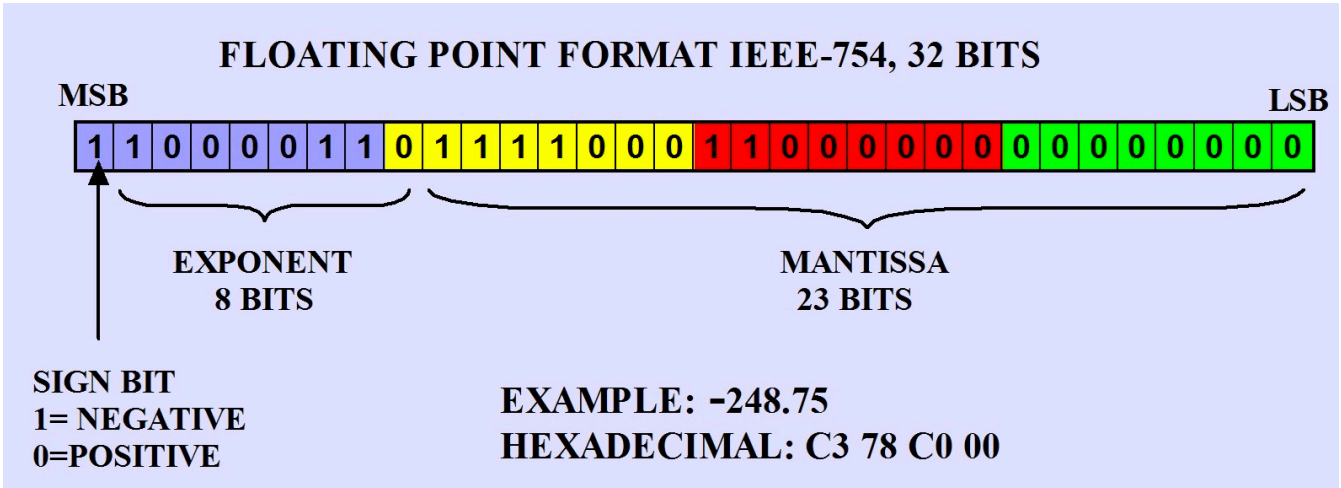


Le Commandant d'Artillerie Cholesky, du Service géographique de l'Armée, tué pendant la grande guerre, a imaginé, au cours de recherches sur la compensation des réseaux géodésiques, un procédé très ingénieux de résolution des équations dites *normales*, obtenues par application de la méthode des moindres carrés à des équations linéaires en nombre inférieur à celui des inconnues. Il en a conclu une méthode générale de résolution des équations linéaires.

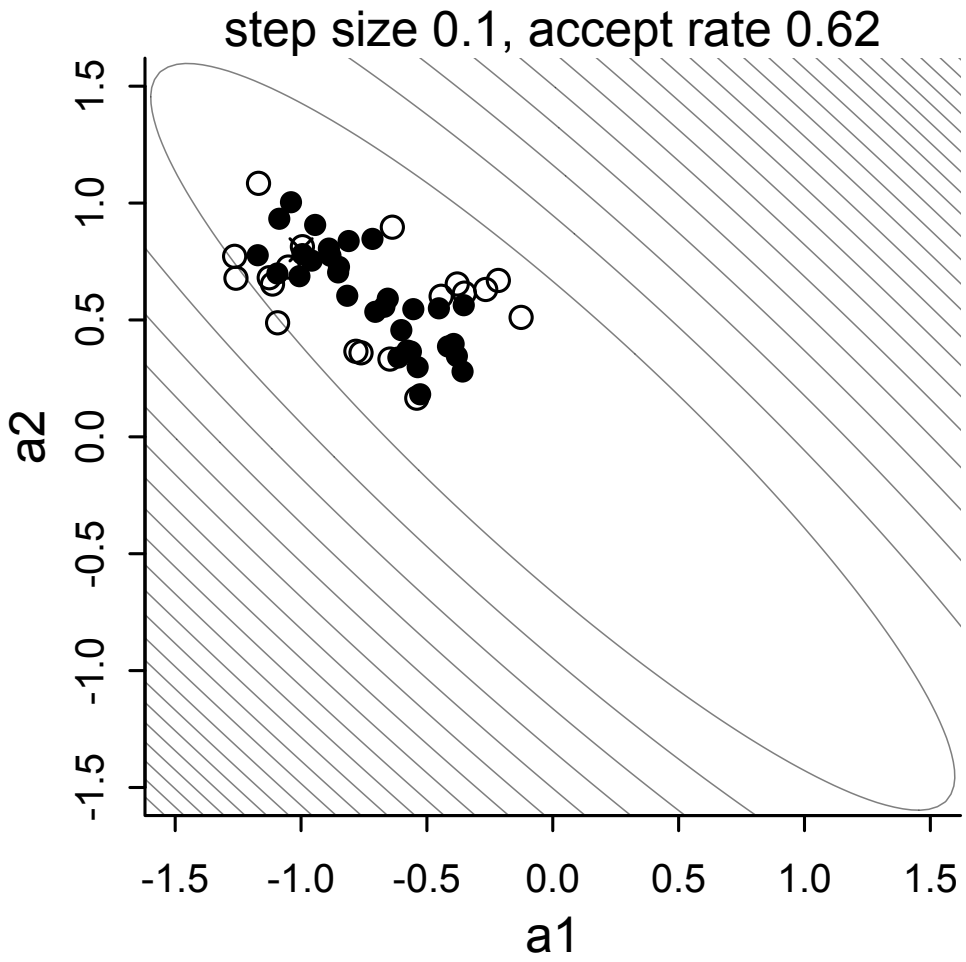
André-Louis Cholesky
(1875–1918)

Re-parameterization

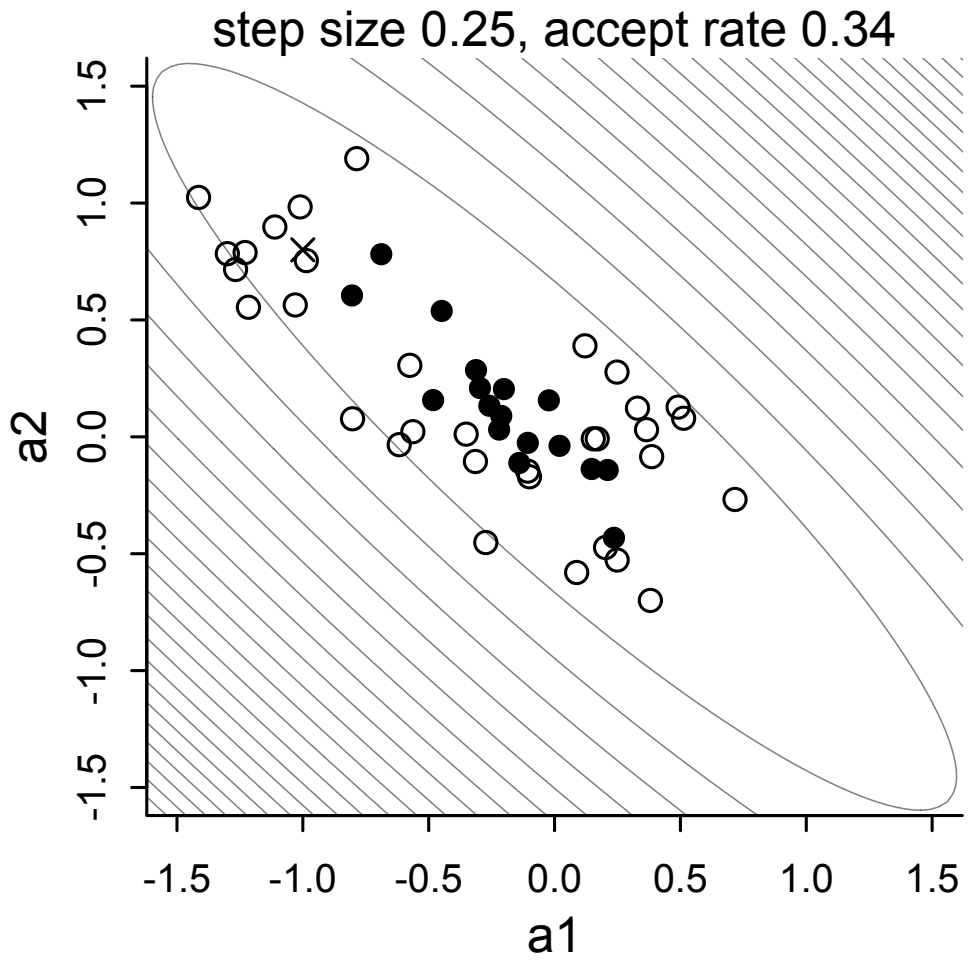
- Math inside the computer is not *real* math
- Numerical algorithms sometimes subtle
- Example: Re-parameterizing hierarchical models
  - Introduce hierarchical model
  - What is a divergent transition?
  - Non-centered parameterization
  - Coding details



Metropolis gets stuck



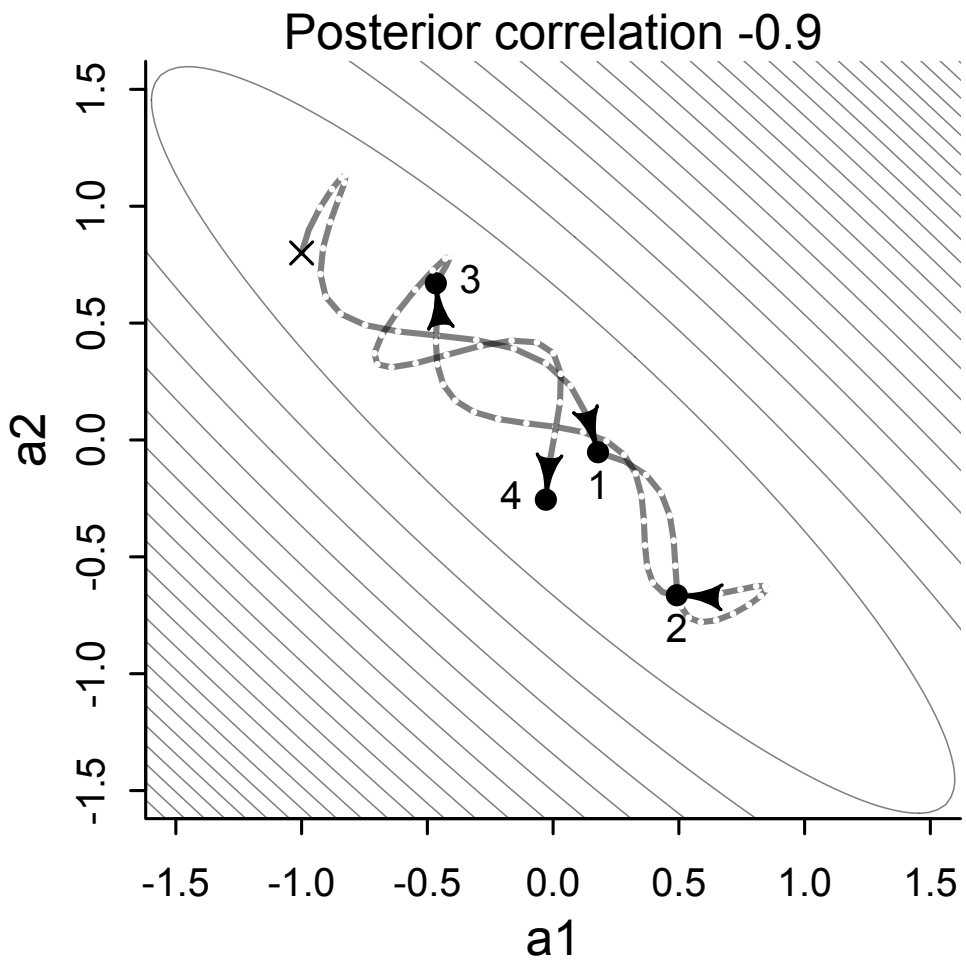
small steps, slow walk



bigs steps, low accept rate

Figure 9.3

Hamiltonian dynamics better



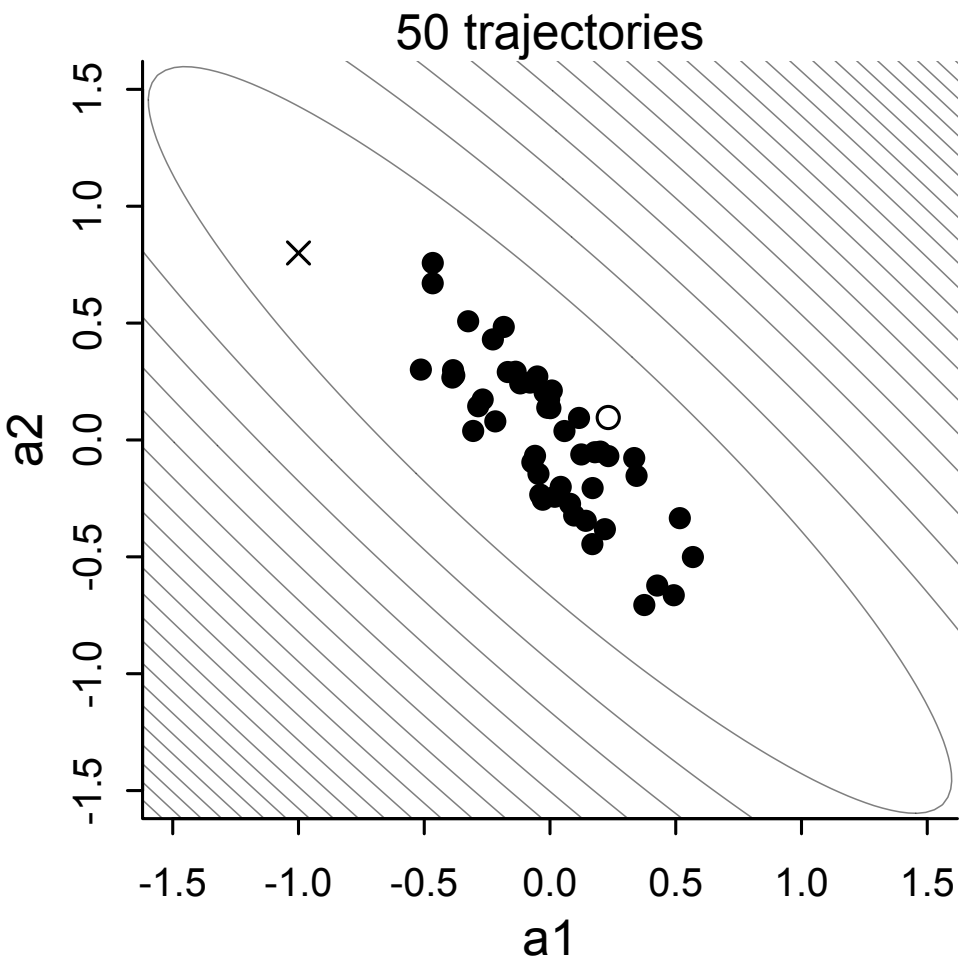
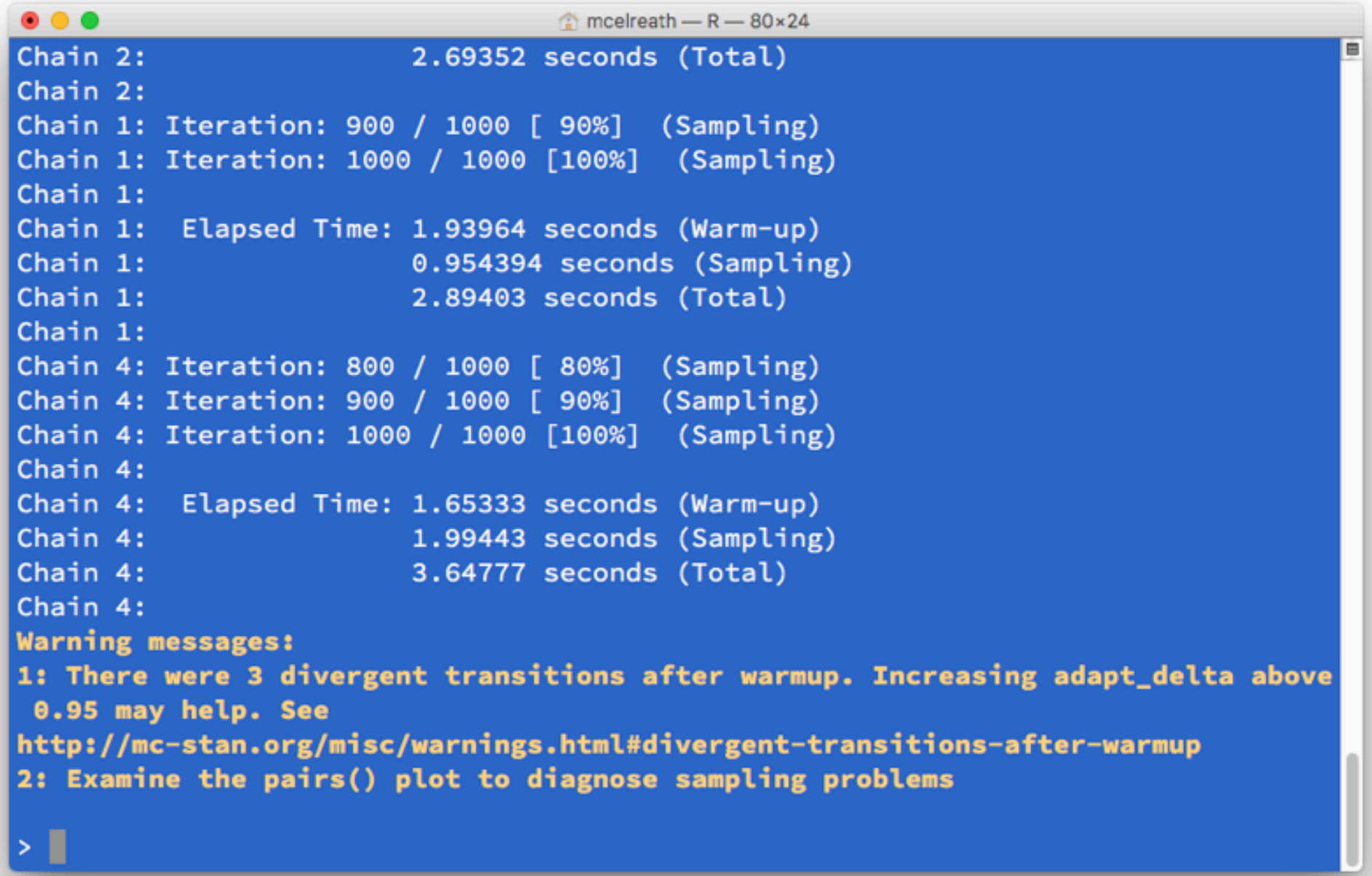


Figure 9.6


mcelreath — R — 80×24
Chain 2:   2.69352 seconds (Total)
Chain 2:
Chain 1: Iteration: 900 / 1000 [90%] (Sampling)
Chain 1: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 1.93964 seconds (Warm-up)
Chain 1:   0.954394 seconds (Sampling)
Chain 1:   2.89403 seconds (Total)
Chain 1:
Chain 4: Iteration: 800 / 1000 [80%] (Sampling)
Chain 4: Iteration: 900 / 1000 [90%] (Sampling)
Chain 4: Iteration: 1000 / 1000 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 1.65333 seconds (Warm-up)
Chain 4:   1.99443 seconds (Sampling)
Chain 4:   3.64777 seconds (Total)
Chain 4:
Warning messages:
1: There were 3 divergent transitions after warmup. Increasing adapt_delta above
 0.95 may help. See
http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
2: Examine the pairs() plot to diagnose sampling problems
>

Multilevel chimpanzees

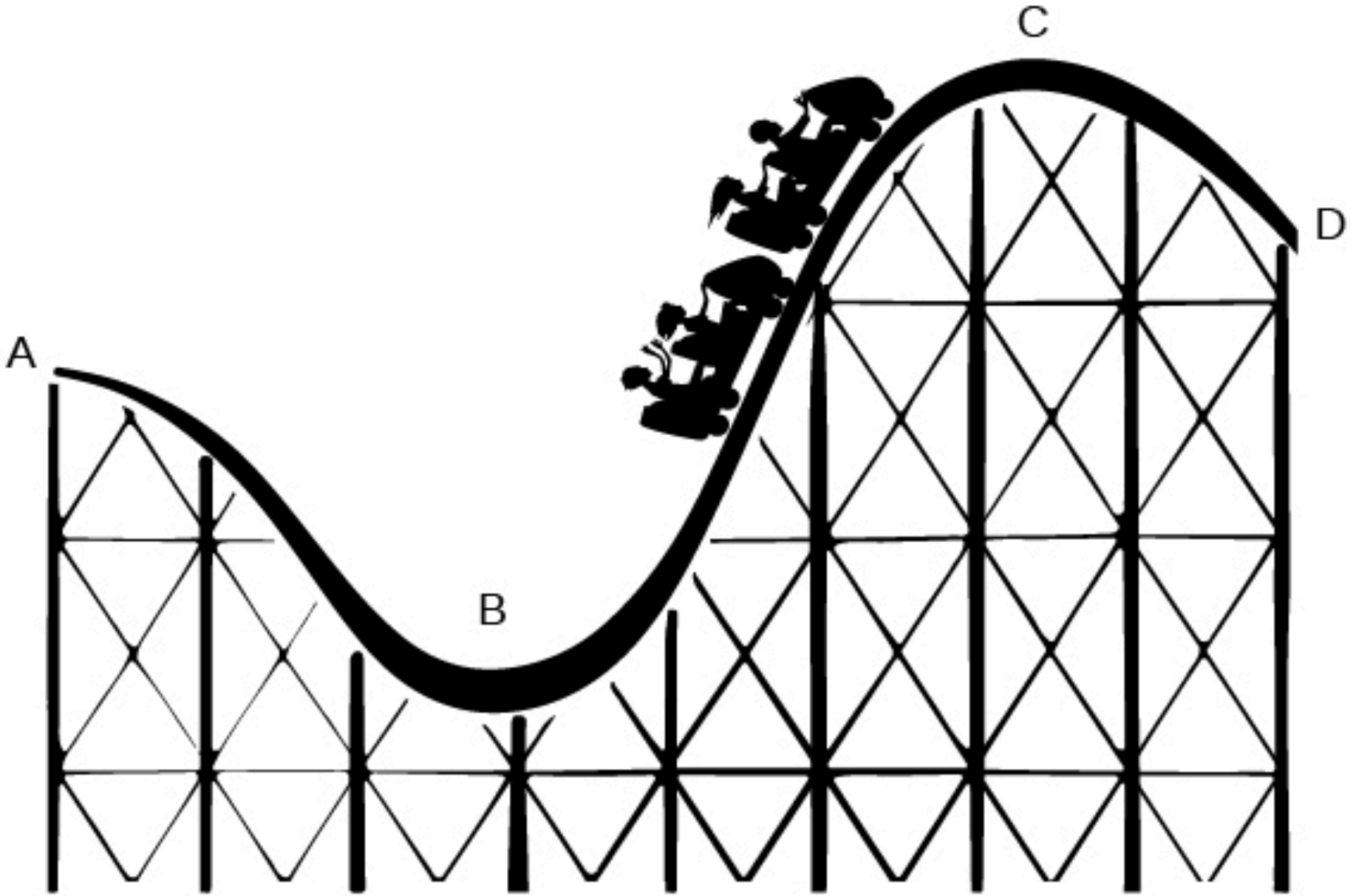
$$
\begin{aligned}
L_i &\sim \text{Binomial}(1, p_i) \\
\text{logit}(p_i) &= \alpha_{\text{ACTOR}[i]} + \gamma_{\text{BLOCK}[i]} + \beta_{\text{TREATMENT}[i]} \\
\beta_j &\sim \text{Normal}(0, 0.5) \quad \text{, for } j = 1..4 \\
\alpha_j &\sim \text{Normal}(\bar{\alpha}, \sigma_\alpha) \quad \text{, for } j = 1..7 \\
\gamma_j &\sim \text{Normal}(0, \sigma_\gamma) \quad \text{, for } j = 1..6 \\
\bar{\alpha} &\sim \text{Normal}(0, 1.5) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\gamma &\sim \text{Exponential}(1)
\end{aligned}
$$

varying intercepts on actor ⟶ $\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma_\alpha)$, for $j = 1..7$

varying intercepts on block ⟶ $\gamma_j \sim \text{Normal}(0, \sigma_\gamma)$, for $j = 1..6$

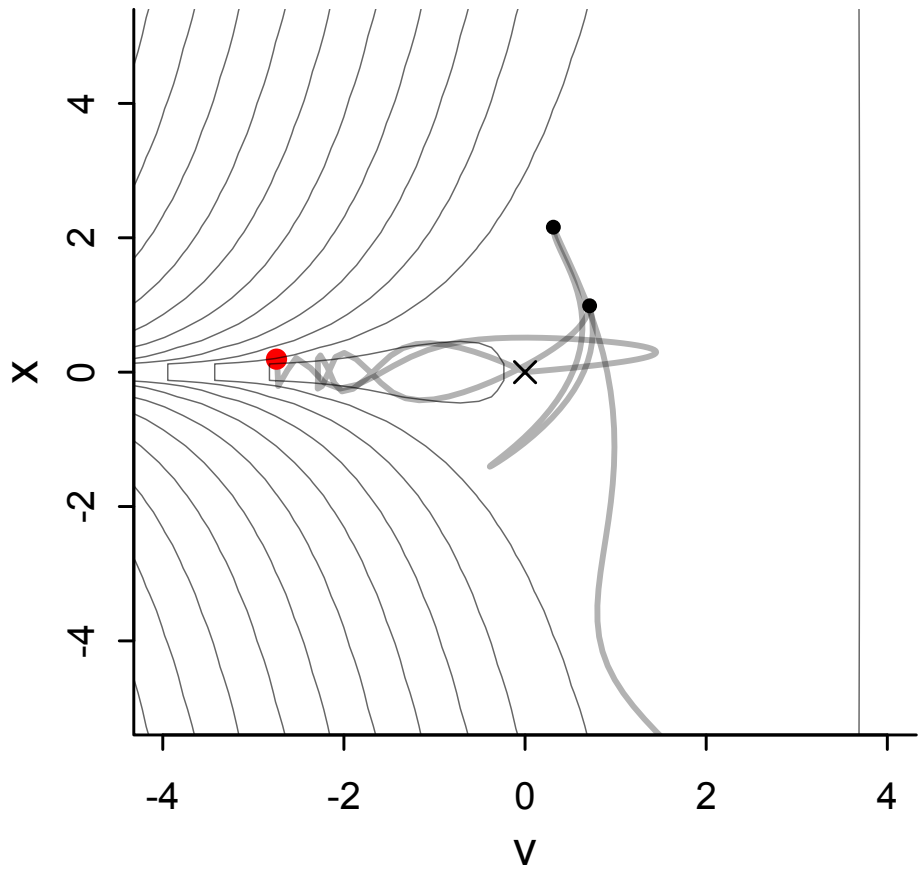
Multilevel chimpanzees

```
m13.4 <- ulam(
    alist(
        pulled_left ~ dbinom( 1 , p ) ,
        logit(p) <- a[actor] + g[block_id] + b[treatment] ,
        b[treatment] ~ dnorm( 0 , 0.5 ),

        # adaptive priors
        a[actor] ~ dnorm( a_bar , sigma_a ),
        g[block_id] ~ dnorm( 0 , sigma_g ),

        # hyper-priors
        a_bar ~ dnorm( 0 , 1.5 ),
        sigma_a ~ dexp(1),
        sigma_g ~ dexp(1)
    ) , data=dat_list , chains=4 , cores=4 , log_lik=TRUE )
```


C
D
A
B

Divergent transitions

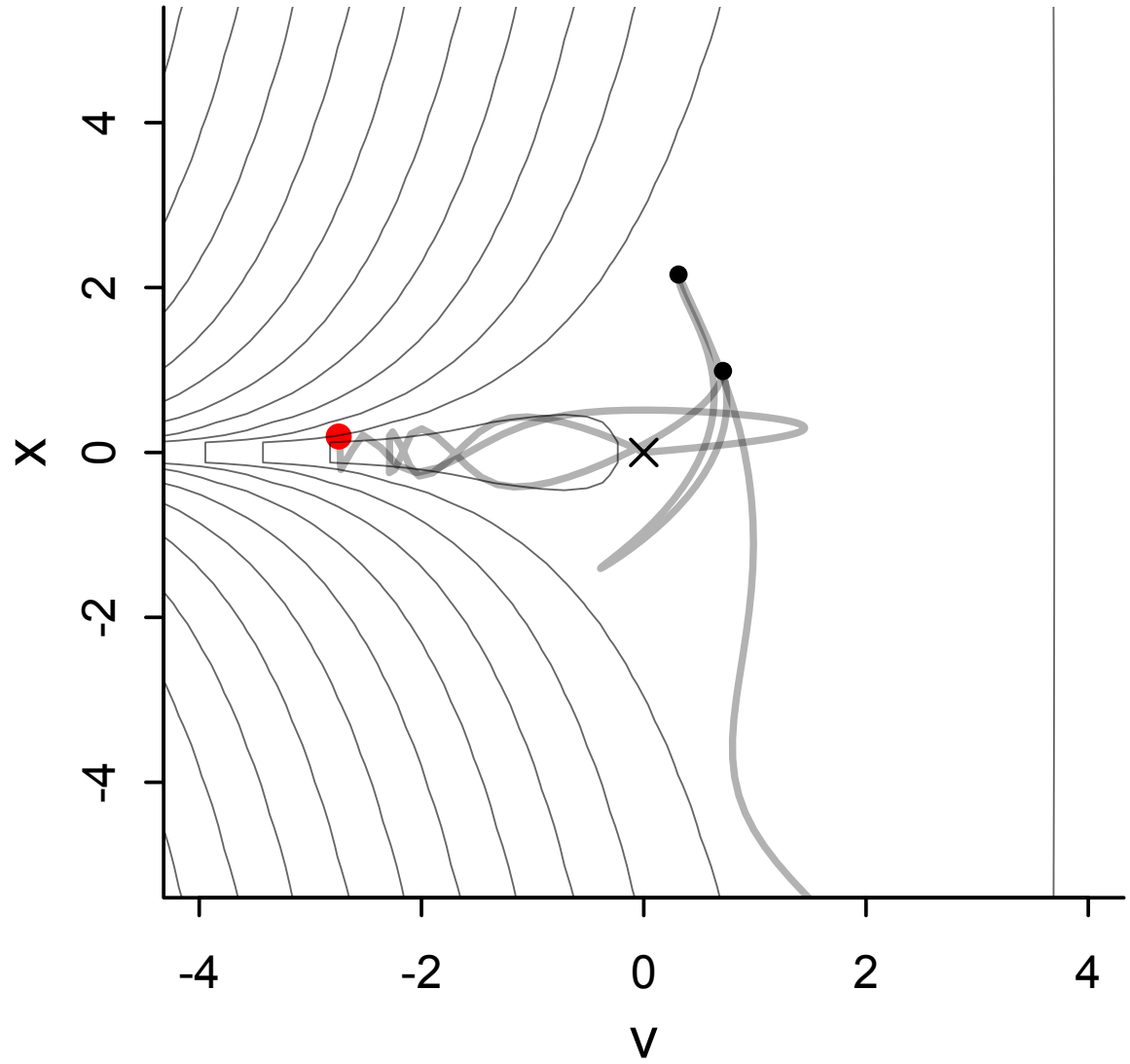
- HMC runs a physics simulation
- Each ***transition*** is a sample path
- In real physics, energy is conserved
- If energy at end of transition is not equal to energy at start, transition is ***divergent***
- Indicates inaccurate approximation
- Tends to happen in regions of strong curvature of log-posterior
- **Other sampling strategies also bad in these cases, but produce no warnings!**



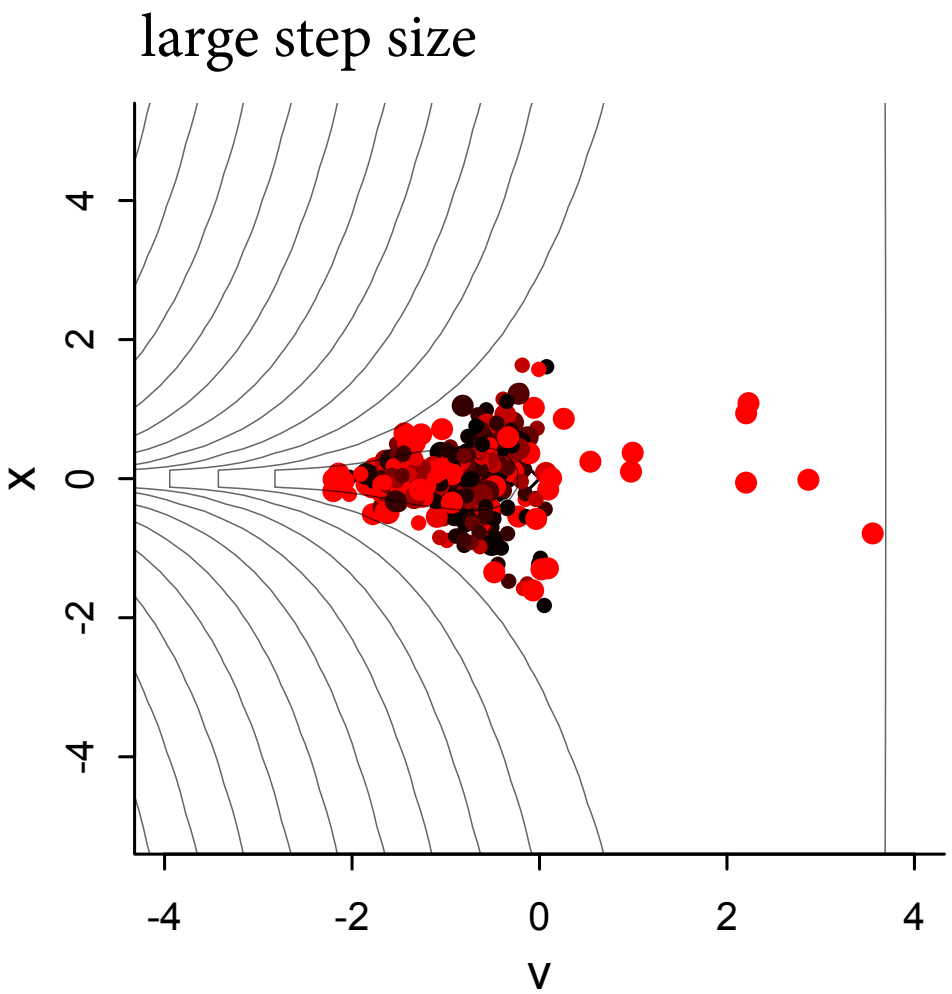
Divergent transitions

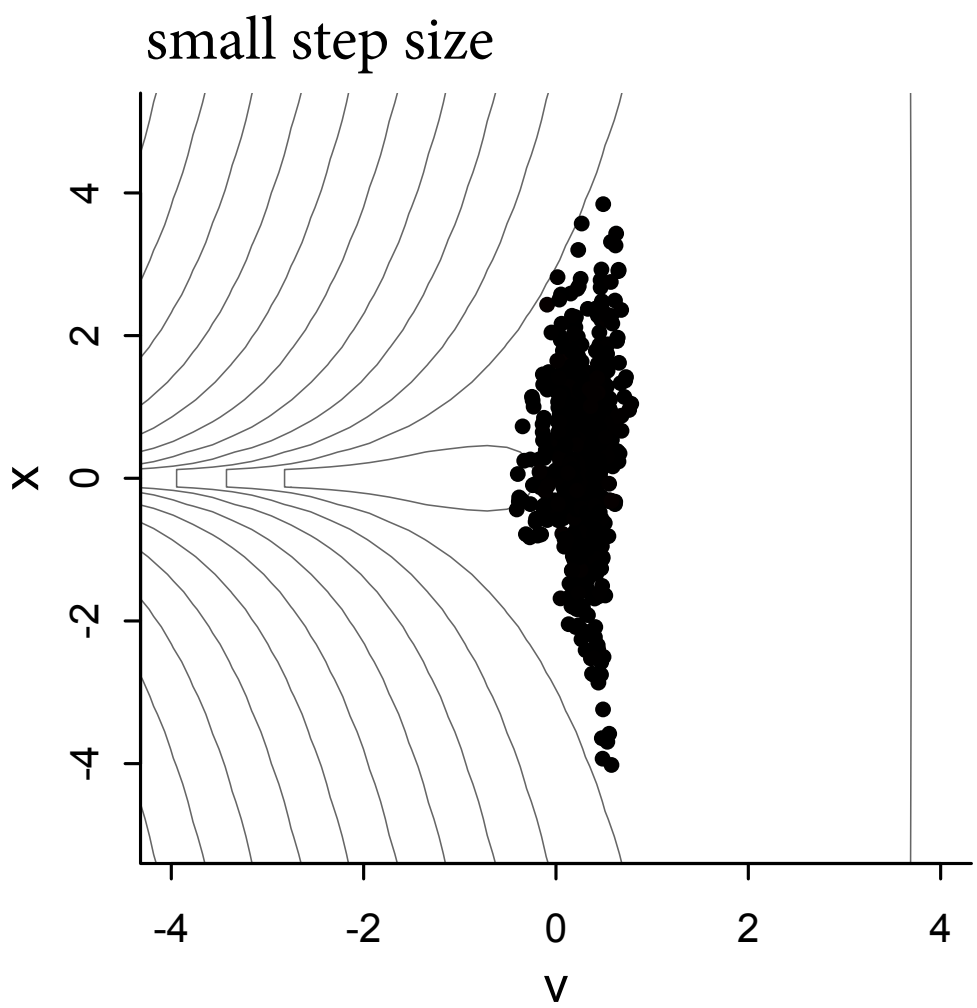
$$v \sim \text{Normal}(0, 3)$$

$$x \sim \text{Normal}(0, \exp(v))$$



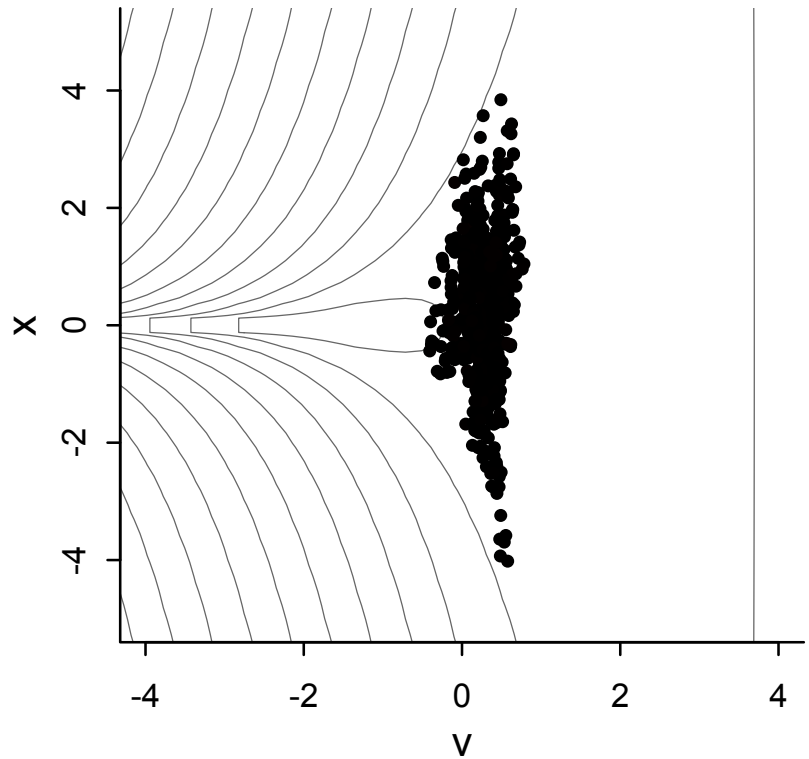
Divergent transitions





Divergent transitions

- Two basic strategies:
- (1) Increase Stan's adapt_delta control parameter => better step size adaptation + slower exploration
- (2) Re-parameterize!



Re-parameterize!

- Most any statistical model can be expressed in several mathematically identical ways

$$\alpha \sim \text{Normal}(\mu, \sigma)$$

Re-parameterize!

- Most any statistical model can be expressed in several mathematically identical ways

$$\alpha \sim \text{Normal}(\mu, \sigma)$$

$$\alpha = \mu + \beta$$
$$\beta \sim \text{Normal}(0, \sigma)$$

Re-parameterize!

- Most any statistical model can be expressed in several mathematically identical ways

$$\alpha \sim \text{Normal}(\mu, \sigma)$$ *Centered*

$$\alpha = \mu + \beta$$
$$\beta \sim \text{Normal}(0, \sigma)$$

$$\alpha = \mu + z\sigma$$
$$z \sim \text{Normal}(0, 1)$$ *Non-centered*

Re-parameterize!

- Why would this madness help with sampling?
- HMC sees a different geometry!

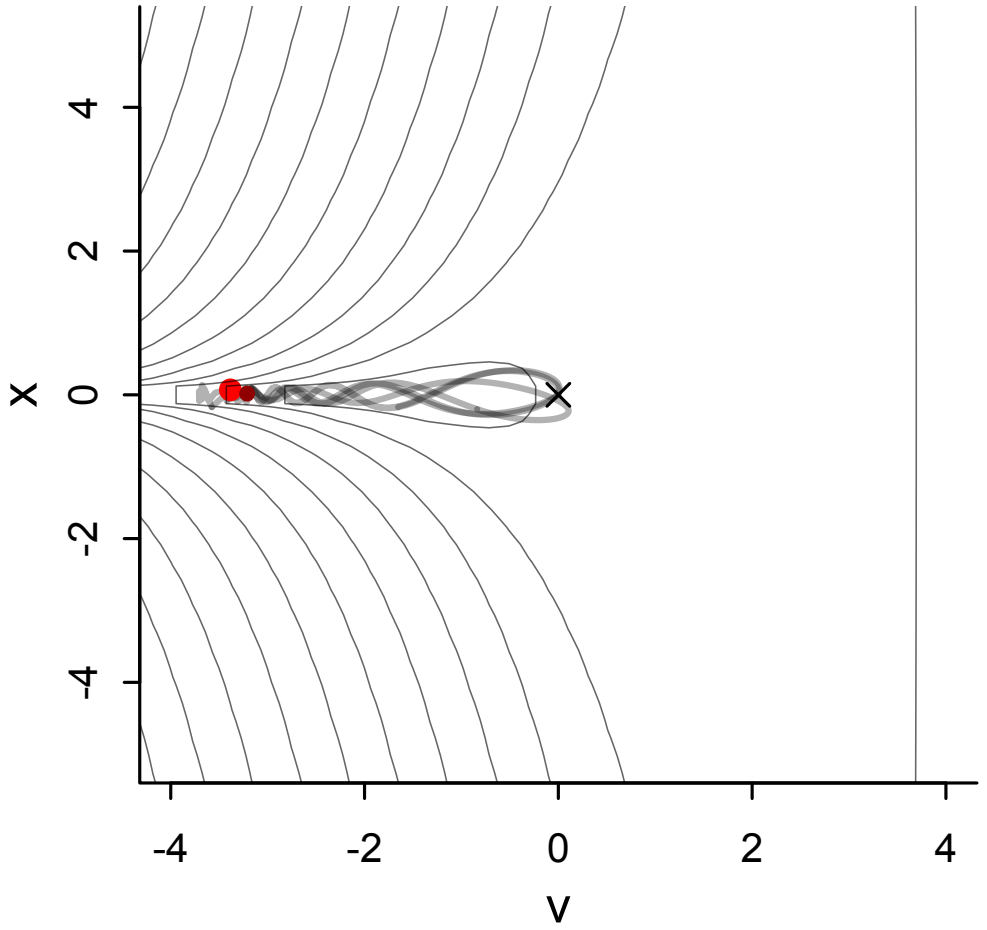
Centered

$$v \sim \text{Normal}(0, 3)$$
$$x \sim \text{Normal}(0, \exp(v))$$

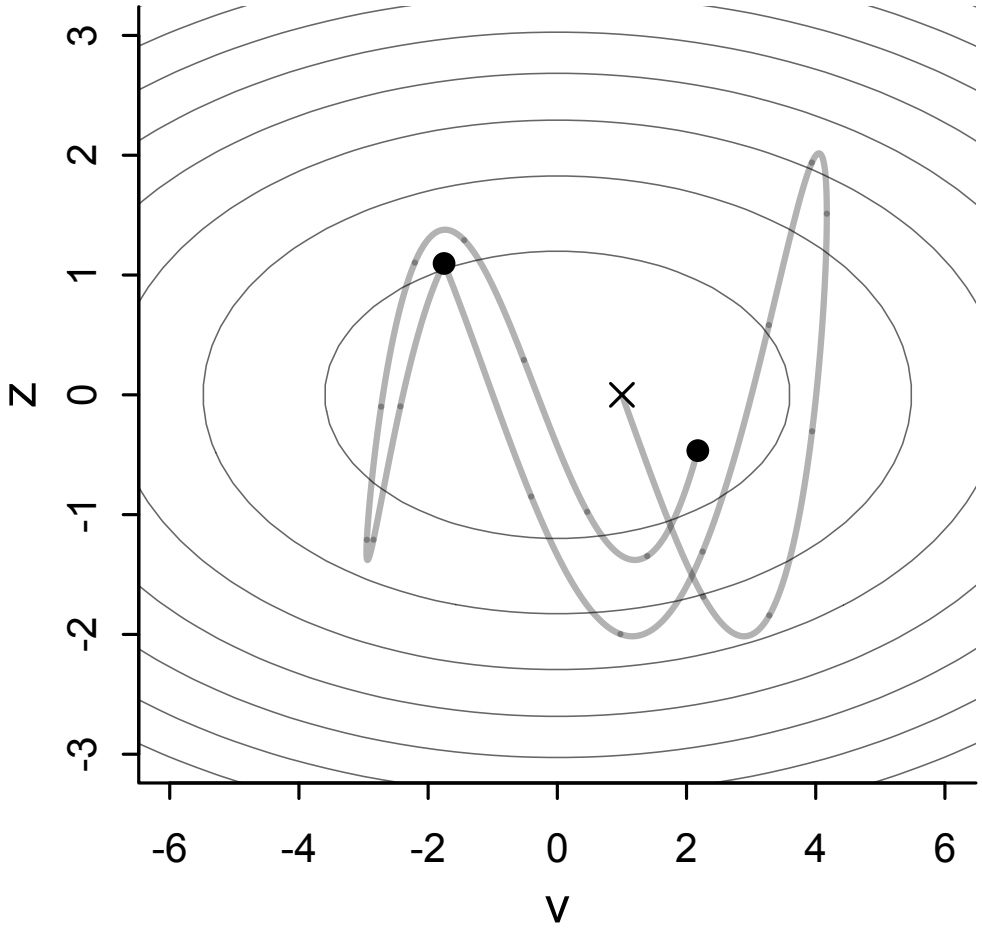
Non-centered

$$v \sim \text{Normal}(0, 3)$$
$$x = z \exp(v)$$
$$z \sim \text{Normal}(0, 1)$$

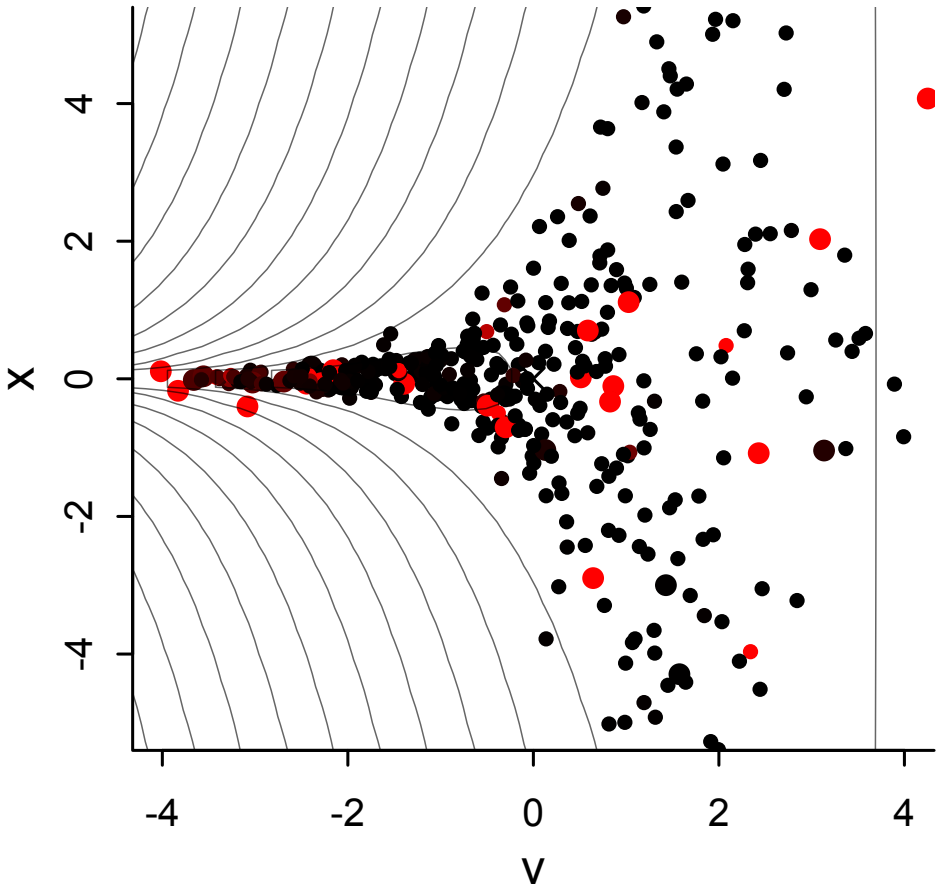
$$v \sim \text{Normal}(0, 3)$$
$$x \sim \text{Normal}(0, \exp(v))$$



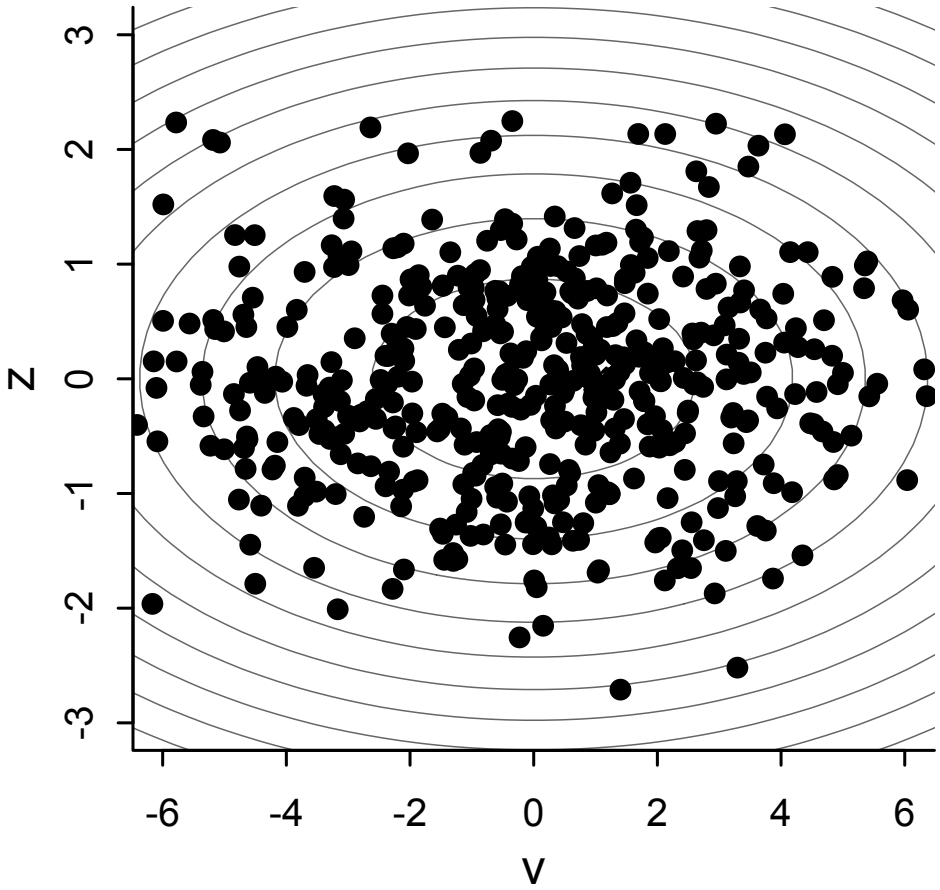
$$v \sim \text{Normal}(0, 3)$$
$$x = z \exp(v)$$
$$z \sim \text{Normal}(0, 1)$$



$$v \sim \text{Normal}(0, 3)$$
$$x \sim \text{Normal}(0, \exp(v))$$



$$v \sim \text{Normal}(0, 3)$$
$$x = z \exp(v)$$
$$z \sim \text{Normal}(0, 1)$$



$$
\begin{aligned}
L_i &\sim \text{Binomial}(1, p_i) \\
\text{logit}(p_i) &= \alpha_{\text{ACTOR}[i]} + \gamma_{\text{BLOCK}[i]} + \beta_{\text{TREATMENT}[i]}
\end{aligned}
$$

$$
\begin{aligned}
L_i &\sim \text{Binomial}(1, p_i) \\
\text{logit}(p_i) &= \bar{\alpha} + z_{\text{ACTOR}[i]}\sigma_\alpha + x_{\text{BLOCK}[i]}\sigma_\gamma + \beta_{\text{TREATMENT}[i]}
\end{aligned}
$$

$$
\begin{aligned}
L_i &\sim \text{Binomial}(1, p_i) \\
\text{logit}(p_i) &= \alpha_{\text{ACTOR}[i]} + \gamma_{\text{BLOCK}[i]} + \beta_{\text{TREATMENT}[i]}
\end{aligned}
$$

$$
\begin{aligned}
L_i &\sim \text{Binomial}(1, p_i) \\
\text{logit}(p_i) &= \bar{\alpha} + z_{\text{ACTOR}[i]}\sigma_\alpha + x_{\text{BLOCK}[i]}\sigma_\gamma + \beta_{\text{TREATMENT}[i]} \\
\beta_j &\sim \text{Normal}(0, 0.5) \quad , \text{for } j = 1..4 \\
z_j &\sim \text{Normal}(0, 1) \quad , \text{for } j = 1..7 \\
x_j &\sim \text{Normal}(0, 1) \quad , \text{for } j = 1..6 \\
\bar{\alpha} &\sim \text{Normal}(0, 1.5) \\
\sigma_\alpha &\sim \text{Exponential}(1) \\
\sigma_\gamma &\sim \text{Exponential}(1)
\end{aligned}
$$

$$L_i \sim \text{Binomial}(1, p_i)$$

$$\text{logit}(p_i) = \bar{\alpha} + z_{\text{ACTOR}[i]}\sigma_\alpha + x_{\text{BLOCK}[i]}\sigma_\gamma + \beta_{\text{TREATMENT}[i]}$$

R code 13.28

```
set.seed(13)
m13.4nc <- ulam(
    alist(
        pulled_left ~ dbinom( 1 , p ) ,
        logit(p) <- a_bar + z[actor]*sigma_a + x[block_id]*sigma_g + b[treatment] ,
        b[treatment] ~ dnorm( 0 , 0.5 ),
        z[actor] ~ dnorm( 0 , 1 ),
        x[block_id] ~ dnorm( 0 , 1 ),
        a_bar ~ dnorm( 0 , 1.5 ),
        sigma_a ~ dexp(1),
        sigma_g ~ dexp(1)
    ) , data=dat_list , chains=4 , cores=4 )
```

Non-centered vs centered

R code 13.29

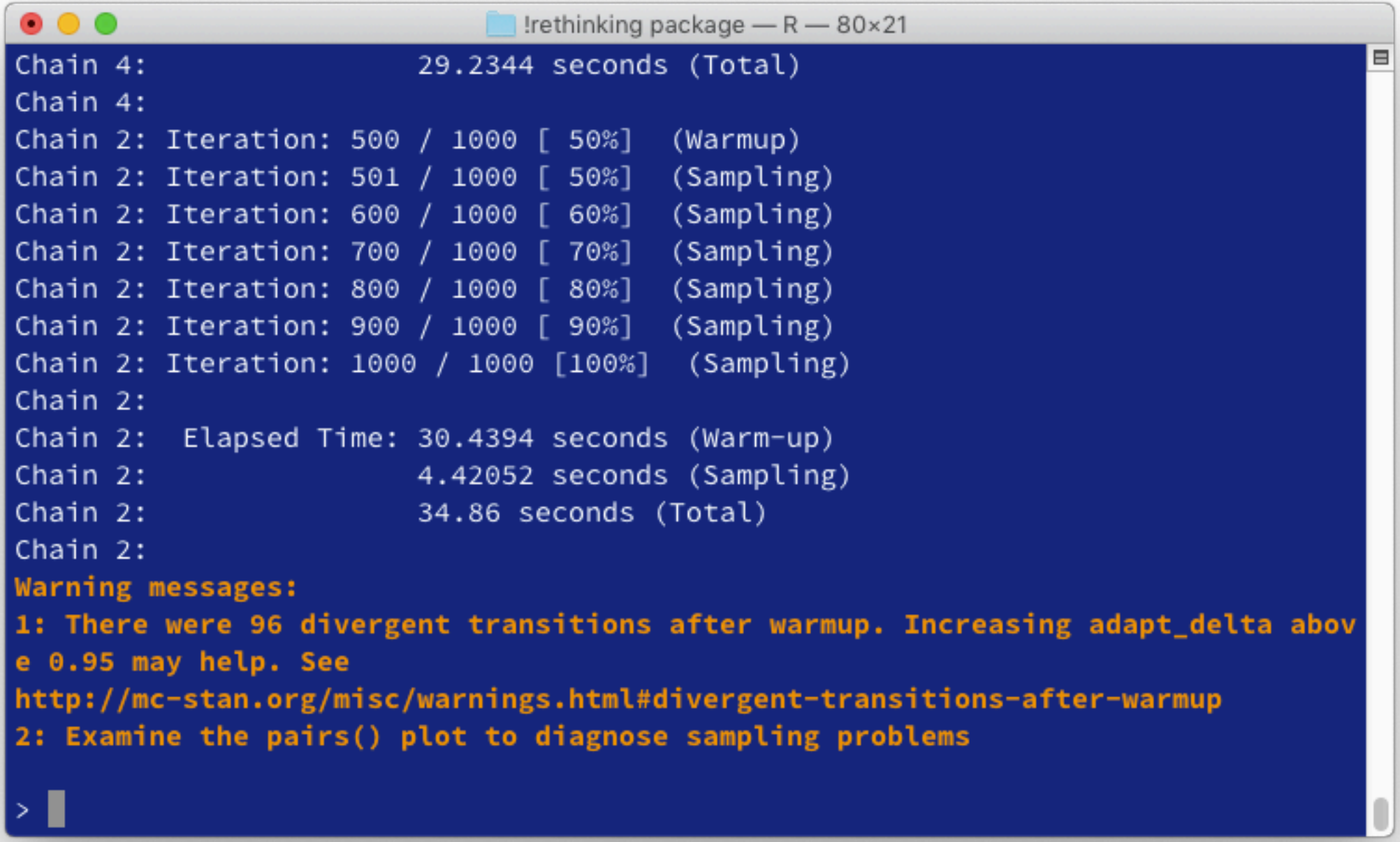
```
neff_c <- precis( m13.4 , depth=2 )[['n_eff']]
neff_nc <- precis( m13.4nc , depth=2 )[['n_eff']]
par_names <- rownames( precis( m13.4 , depth=2 ) )
neff_table <- cbind( neff_c , neff_nc )
rownames(neff_table) <- par_names
round(t(neff_table))
```

```
        b[1] b[2] b[3] b[4] a[1] a[2] a[3] a[4] a[5] a[6] a[7] g[1] g[2] g[3]
neff_c   584  572  587  523  562  759  541  547  582  601  677  418  793  784
neff_nc 1144 1233 1144 1115  592  962  614  614  611  627  772 1811 2137 1748
        g[4] g[5] g[6] a_bar sigma_a sigma_g
neff_c   605  934  603   824     898     243
neff_nc 1575 2127 1429   573     777     970
```

Maximally random chimps

```
m14.2 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors
        vector[4]:alpha[actor] ~ multi_normal(0,Rho_actor,sigma_actor),
        vector[4]:beta[block_id] ~ multi_normal(0,Rho_block,sigma_block),

        # fixed priors
        g[tid] ~ dnorm(0,1),
        sigma_actor ~ dexp(1),
        Rho_actor ~ dlkjcorr(4),
        sigma_block ~ dexp(1),
        Rho_block ~ dlkjcorr(4)
    ) , data=dat , chains=4 , cores=4 )
```

Divergence, my old friend



Non-centered form

- Non-centered easy for uni-variate priors: Just factor out sigma
- But now need to factor correlation matrix out of the prior and smuggle into linear model
- Can be done: Cholesky factor

André-Louis Cholesky (1875–1918)

Cholesky magic

```
N <- 1e4
sigma1 <- 2
sigma2 <- 0.5
rho <- 0.6
z1 <- rnorm( N )
z2 <- rnorm( N )
a1 <- z1 * sigma1
a2 <- ( rho*z1 + sqrt( 1-rho^2 )*z2 )*sigma2
```

```
> cor(z1,z2)
[1] -0.0005542644
> cor(a1,a2)
[1] 0.5999334
> sd(a1)
[1] 1.997036
> sd(a2)
[1] 0.4989456
```

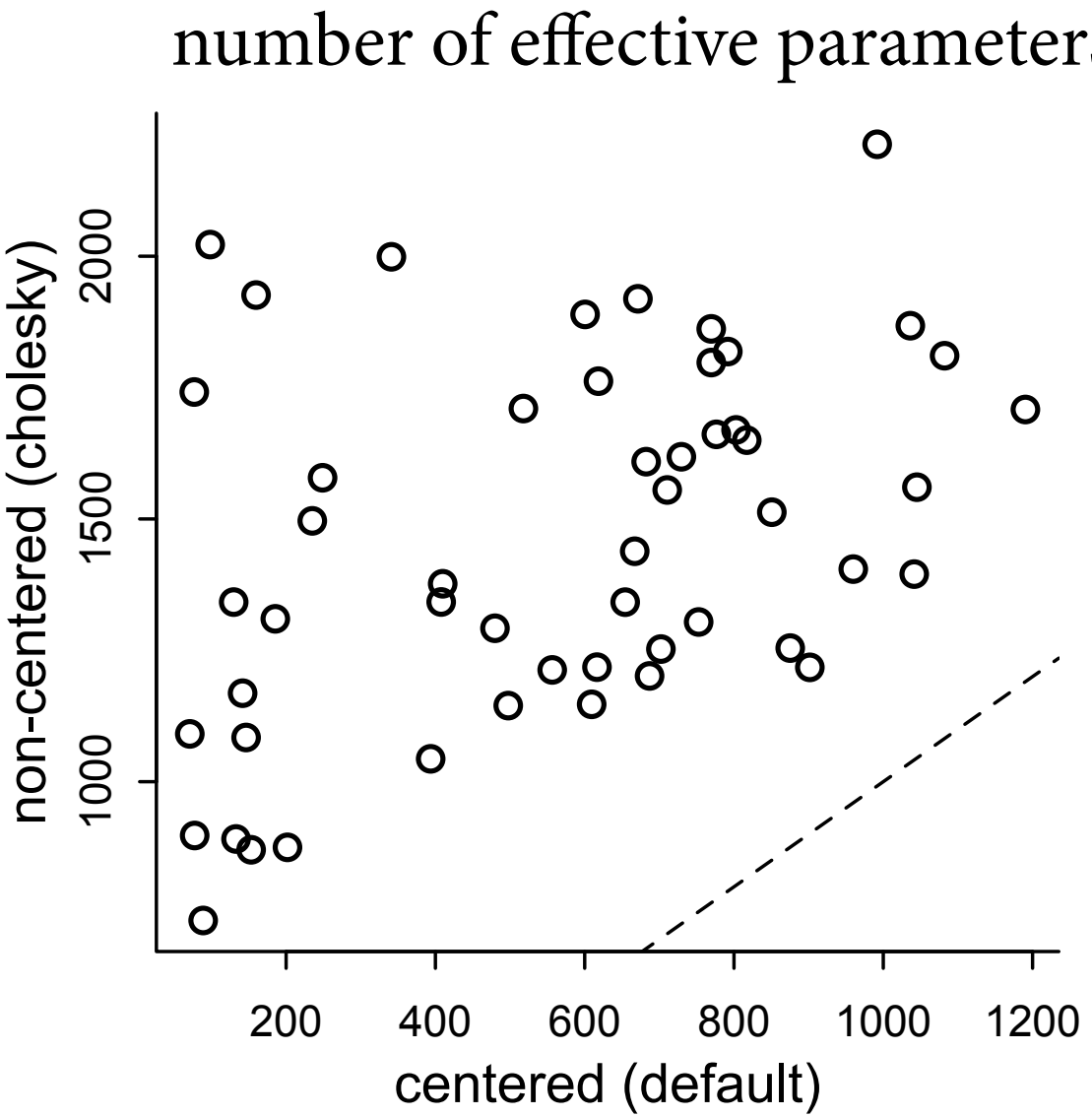
Non-centered random chimps

```
m14.3 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors - non-centered
        transpars> matrix[actor,4]:alpha <-
                compose_noncentered( sigma_actor , L_Rho_actor , z_actor ),
        transpars> matrix[block_id,4]:beta <-
                compose_noncentered( sigma_block , L_Rho_block , z_block ),
        matrix[4,actor]:z_actor ~ normal( 0 , 1 ),
        matrix[4,block_id]:z_block ~ normal( 0 , 1 ),

        # fixed priors
        g[tid] ~ normal(0,1),
        vector[4]:sigma_actor ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_actor ~ lkj_corr_cholesky( 2 ),
        vector[4]:sigma_block ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_block ~ lkj_corr_cholesky( 2 )
    ) , data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

Non-centered random chimps

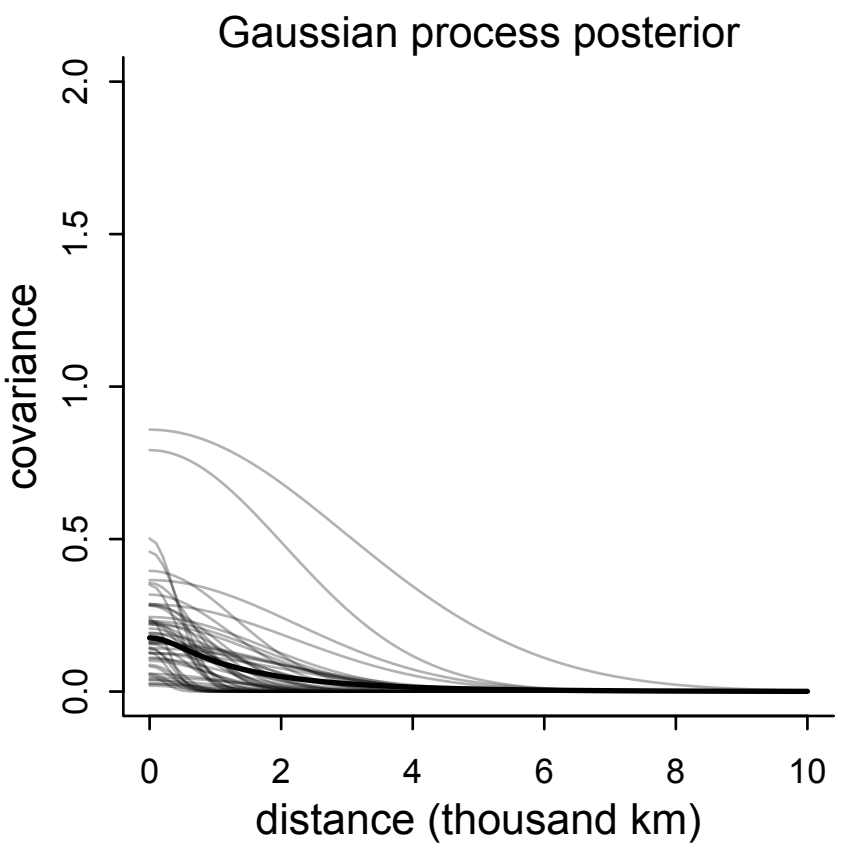
```
m14.3 <- ulam(
    alist(
        L ~ binomial(1,p),
        logit(p) <- g[tid] + alpha[actor,tid] + beta[block_id,tid],

        # adaptive priors - non-centered
        transpars> matrix[actor,4]:alpha <-
                compose_noncentered( sigma_actor , L_Rho_actor , z_actor ),
        transpars> matrix[block_id,4]:beta <-
                compose_noncentered( sigma_block , L_Rho_block , z_block ),
        matrix[4,actor]:z_actor ~ normal( 0 , 1 ),
        matrix[4,block_id]:z_block ~ normal( 0 , 1 ),

        # fixed priors
        g[tid] ~ normal(0,1),
        vector[4]:sigma_actor ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_actor ~ lkj_corr_cholesky( 2 ),
        vector[4]:sigma_block ~ dexp(1),
        cholesky_factor_corr[4]:L_Rho_block ~ lkj_corr_cholesky( 2 )
    ) , data=dat , chains=4 , cores=4 , log_lik=TRUE )
```

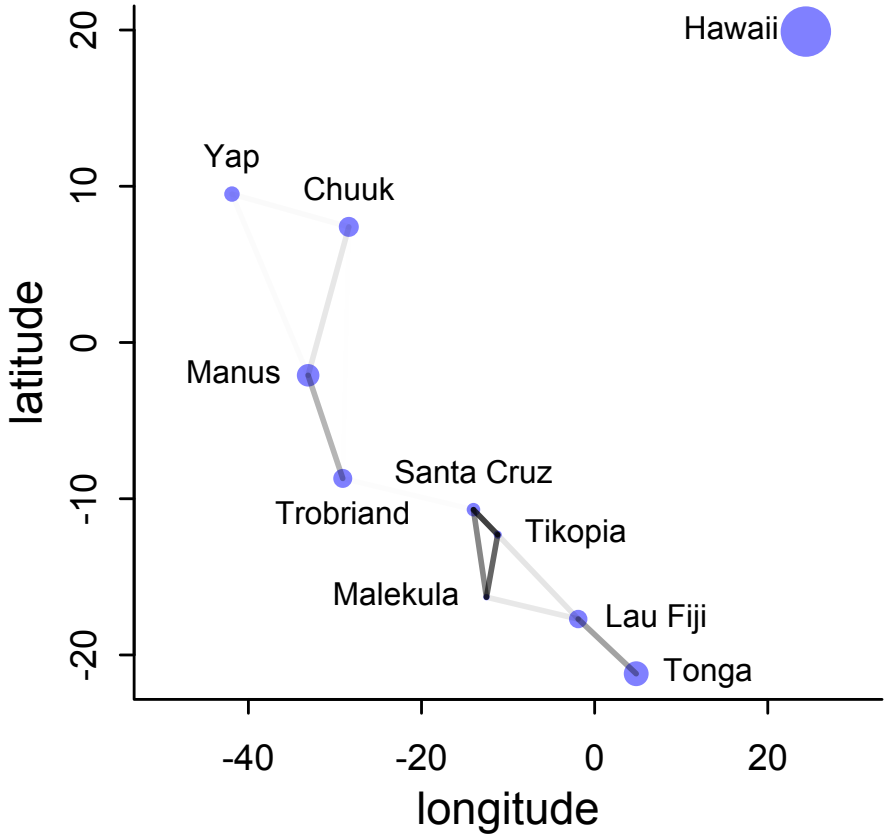
Non-centered random chimps



Works for many model types

- Gaussian processes e.g.





$$
\begin{aligned}
T_i &\sim \text{Poisson}(\lambda_i) \\
\lambda_i &= \exp(k_{\text{SOCIETY}[i]})\alpha P_i^{\beta}/\gamma \\
\mathbf{k} &\sim \text{MVNormal}\big((0,\ldots,0),\mathbf{K}\big) \\
\mathbf{K}_{ij} &= \eta^2 \exp(-\rho^2 D_{ij}^2) + \delta_{ij}(0.01) \\
\alpha &\sim \text{Exponential}(1) \\
\beta &\sim \text{Exponential}(1) \\
\eta^2 &\sim \text{Exponential}(2) \\
\rho^2 &\sim \text{Exponential}(0.5)
\end{aligned}
$$

$$
\begin{aligned}
T_i &\sim \text{Poisson}(\lambda_i) \\
\lambda_i &= \exp(k_{\text{SOCIETY}[i]})\alpha P_i^{\beta}/\gamma \\
\mathbf{k} &\sim \text{MVNormal}\big((0,\ldots,0),\mathbf{K}\big) \\
\mathbf{K}_{ij} &= \eta^2 \exp(-\rho^2 D_{ij}^2) + \delta_{ij}(0.01) \\
\alpha &\sim \text{Exponential}(1) \\
\beta &\sim \text{Exponential}(1) \\
\eta^2 &\sim \text{Exponential}(2) \\
\rho^2 &\sim \text{Exponential}(0.5)
\end{aligned}
$$

R code 14.38

```
m14.7 <- ulam(
    alist(
        T ~ dpois(lambda),
        lambda <- (a*P^b/g)*exp(k[society]),
        vector[10]:k ~ multi_normal( 0 , SIGMA ),
        matrix[10,10]:SIGMA <- cov_GPL2( Dmat , etasq , rhosq , 0.01 ),
        c(a,b,g) ~ dexp( 1 ),
        etasq ~ dexp( 2 ),
        rhosq ~ dexp( 0.5 )
    ), data=dat_list , chains=4 , cores=4 , iter=2000 )
```

R code 14.45

Non-centered

```
m14.7nc <- ulam(
    alist(
        T ~ dpois(lambda),
        lambda <- (a*P^b/g)*exp(k[society]),

        # non-centered Gaussian Process prior
        transpars> vector[10]: k <<- L_SIGMA * z,
        vector[10]: z ~ normal( 0 , 1 ),
        transpars> matrix[10,10]: L_SIGMA <<- cholesky_decompose( SIGMA ),
        transpars> matrix[10,10]: SIGMA <- cov_GPL2( Dmat , etasq , rhosq , 0.01 ),

        c(a,b,g) ~ dexp( 1 ),
        etasq ~ dexp( 2 ),
        rhosq ~ dexp( 0.5 )
    ), data=dat_list , chains=4 , cores=4 , iter=2000 )
```

R code 14.38

Centered

```
m14.7 <- ulam(
    alist(
        T ~ dpois(lambda),
        lambda <- (a*P^b/g)*exp(k[society]),
        vector[10]:k ~ multi_normal( 0 , SIGMA ),
        matrix[10,10]:SIGMA <- cov_GPL2( Dmat , etasq , rhosq , 0.01 ),
        c(a,b,g) ~ dexp( 1 ),
        etasq ~ dexp( 2 ),
        rhosq ~ dexp( 0.5 )
    ), data=dat_list , chains=4 , cores=4 , iter=2000 )
```

Re-parameterization

- Non-centered approach: Factor everything out of the prior for better sampling
- Steps:
  - (1) matrix of z-scores ~ Normal(0,1)
  - (2) Cholesky factor for covariance
  - (3) Multiply Cholesky covariance and z-scores to get matrix of scaled and correlated parameters

- Sometimes having everything inside the prior is better — typically when scale is well-identified

THANK YOU!