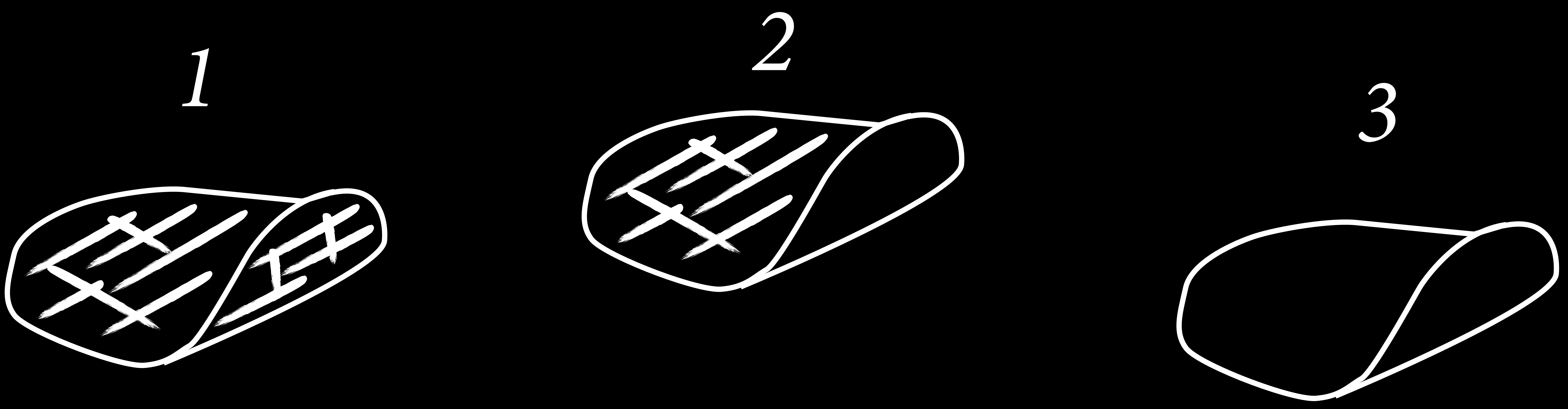
Statistical Rethinking

2023

17. Measurement & Misclassification


ONE SIDE
PERFECT
BOTH SIDES



You are served:

What's the probability the other side is also burnt?

$$\Pr(\text{burnt down} \mid \text{burnt up}) = \frac{\Pr(\text{burnt up}, \text{burnt down})}{\Pr(\text{burnt up})}$$

$$\Pr(\text{burnt down} \mid \text{burnt up}) = \frac{\Pr(\text{burnt up}, \text{burnt down})}{\Pr(\text{burnt up})}$$

$$\Pr(\text{burnt up}) = (1/3)(1) + (1/3)(0.5) + (1/3)(0) = 1/2$$

$$\Pr(\text{burnt down} \mid \text{burnt up}) = \frac{\Pr(\text{burnt up}, \text{burnt down})}{\Pr(\text{burnt up})}$$

$$\Pr(\text{burnt up}) = (1/3)(1) + (1/3)(0.5) + (1/3)(0) = 1/2$$

$$\Pr(\text{burnt down} \mid \text{burnt up}) = \frac{1/3}{1/2} = \frac{2}{3}$$

The Importance of Not Being Clever

Being clever is unreliable and opaque

Better to follow the axioms

Probability theory provides solutions to challenging problems, if only we'll follow the rules

Often nothing else to "understand"

Ye Olde Causal Alchemy

The Four Elemental Confounds

The Fork

$$X \leftarrow Z \rightarrow Y$$

The Pipe

$$X \rightarrow Z \rightarrow Y$$

The Collider

$$X \rightarrow Z \leftarrow Y$$

The Descendant

$$\begin{tikzcd} X \arrow[r] & Z \arrow[r] \arrow[d] & Y \\ & A & \end{tikzcd}$$

Measurement Error

Many variables are proxies of the causes of interest

Common to ignore measurement

Many ad hoc procedures

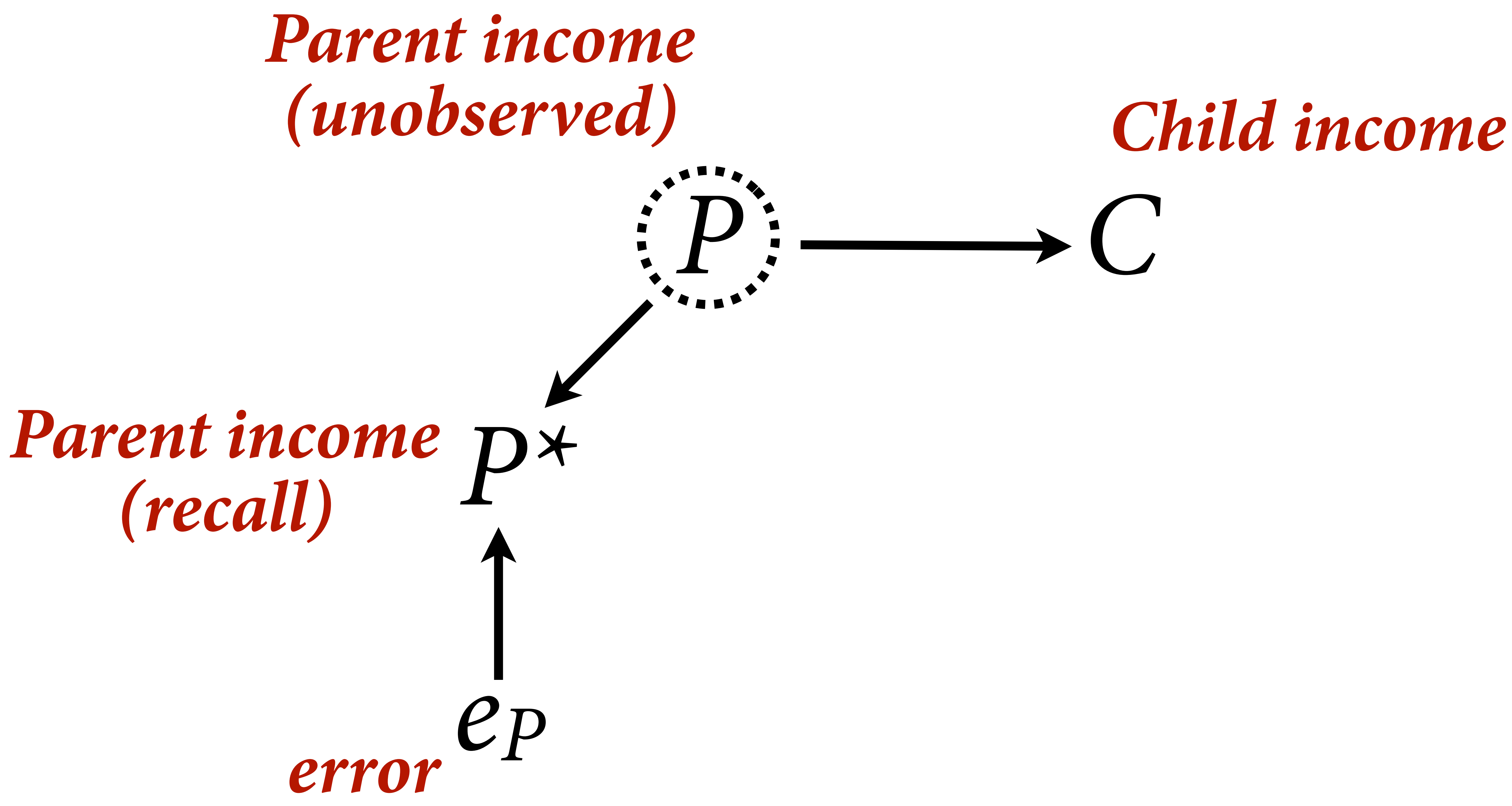
Think causally, lean on probability theory

The Descendant

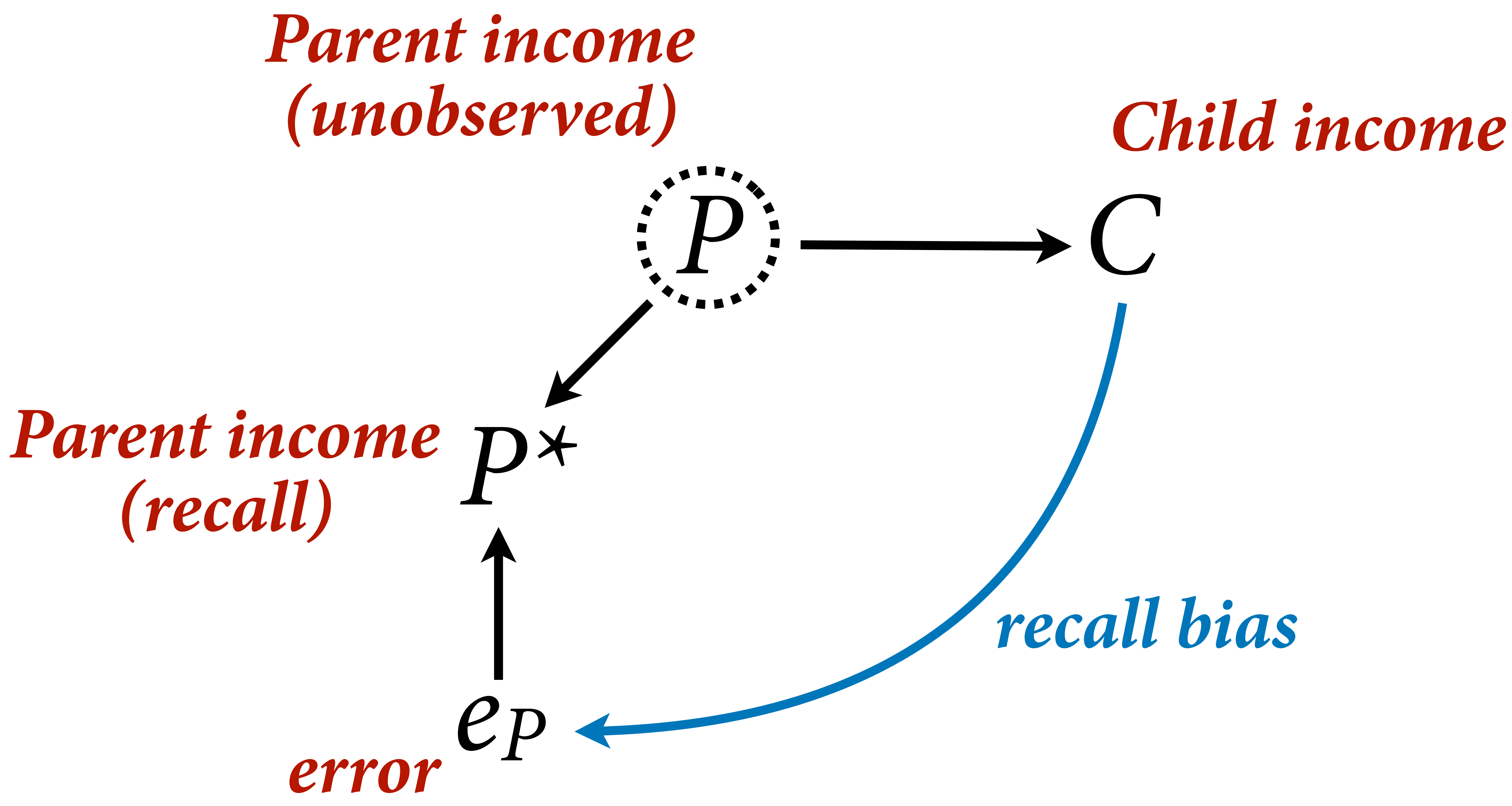
Myth: Measurement error only reduces effect estimates, never increases them

Parent income ***Child income***

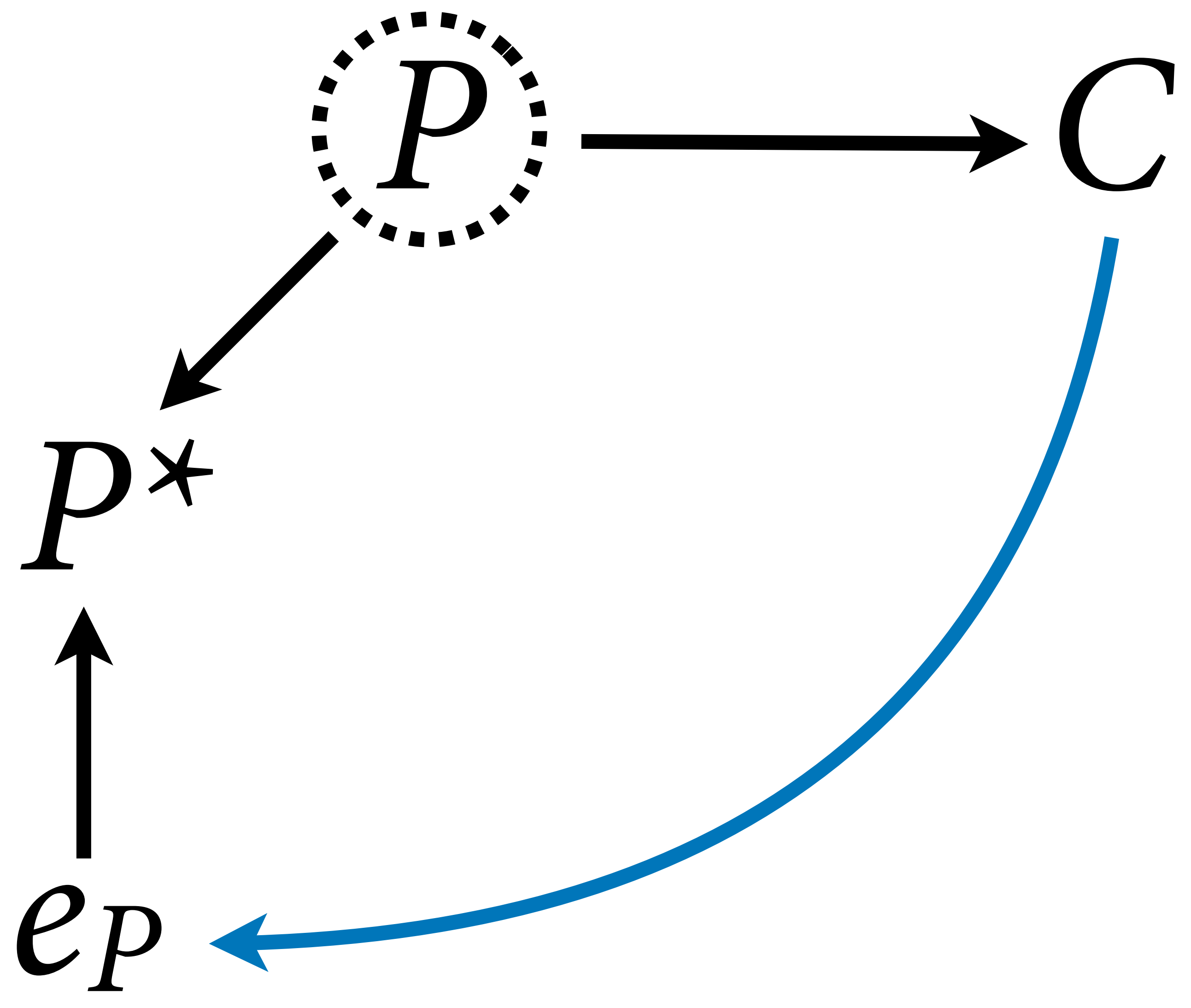
$$P \longrightarrow C$$



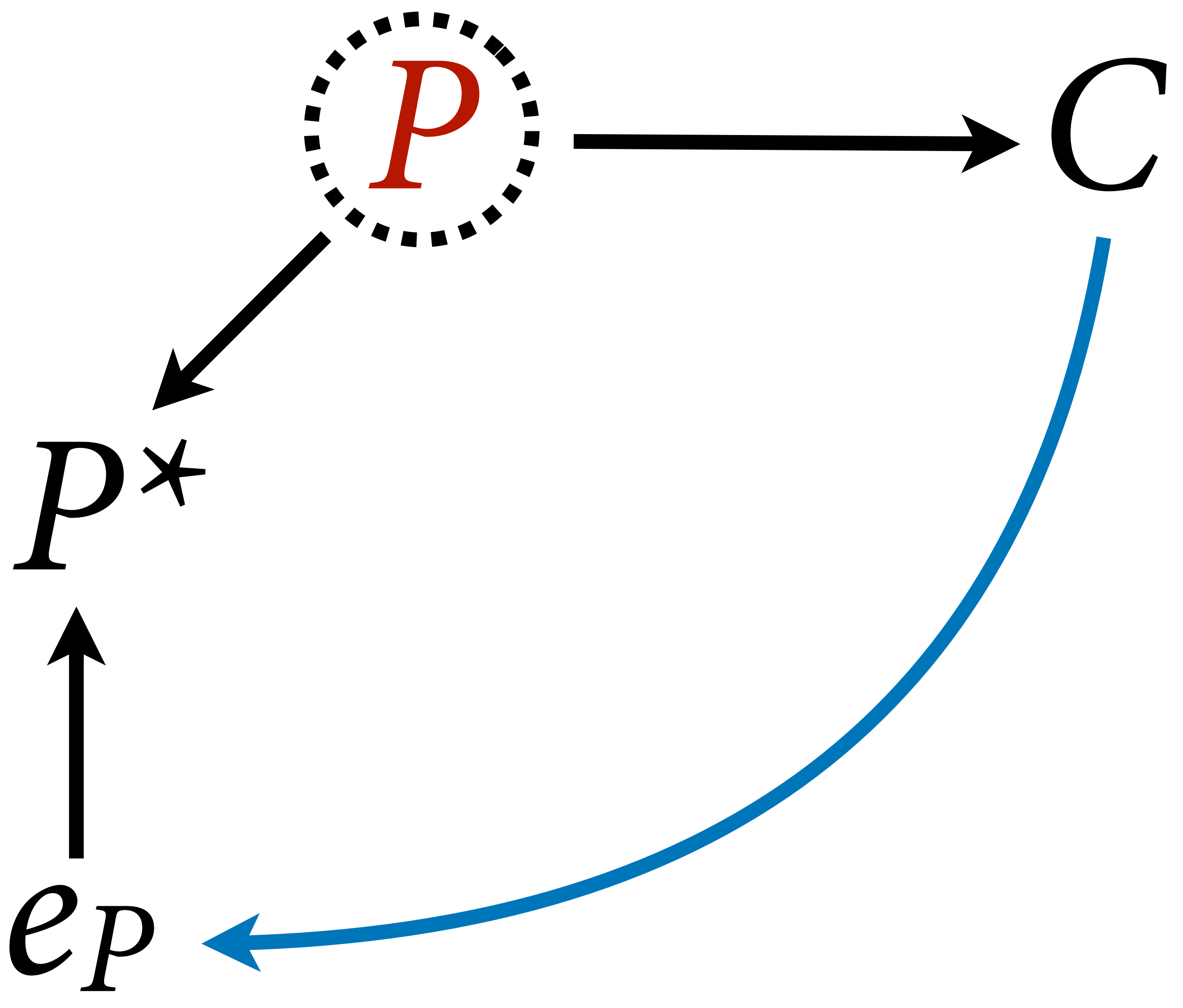
Parent income
(unobserved)
P
Child income
C
Parent income
(recall)
P*
e_P
error


Parent income
(unobserved)
Child income
P
C
Parent income
(recall)
P^*
recall bias
e_P
error

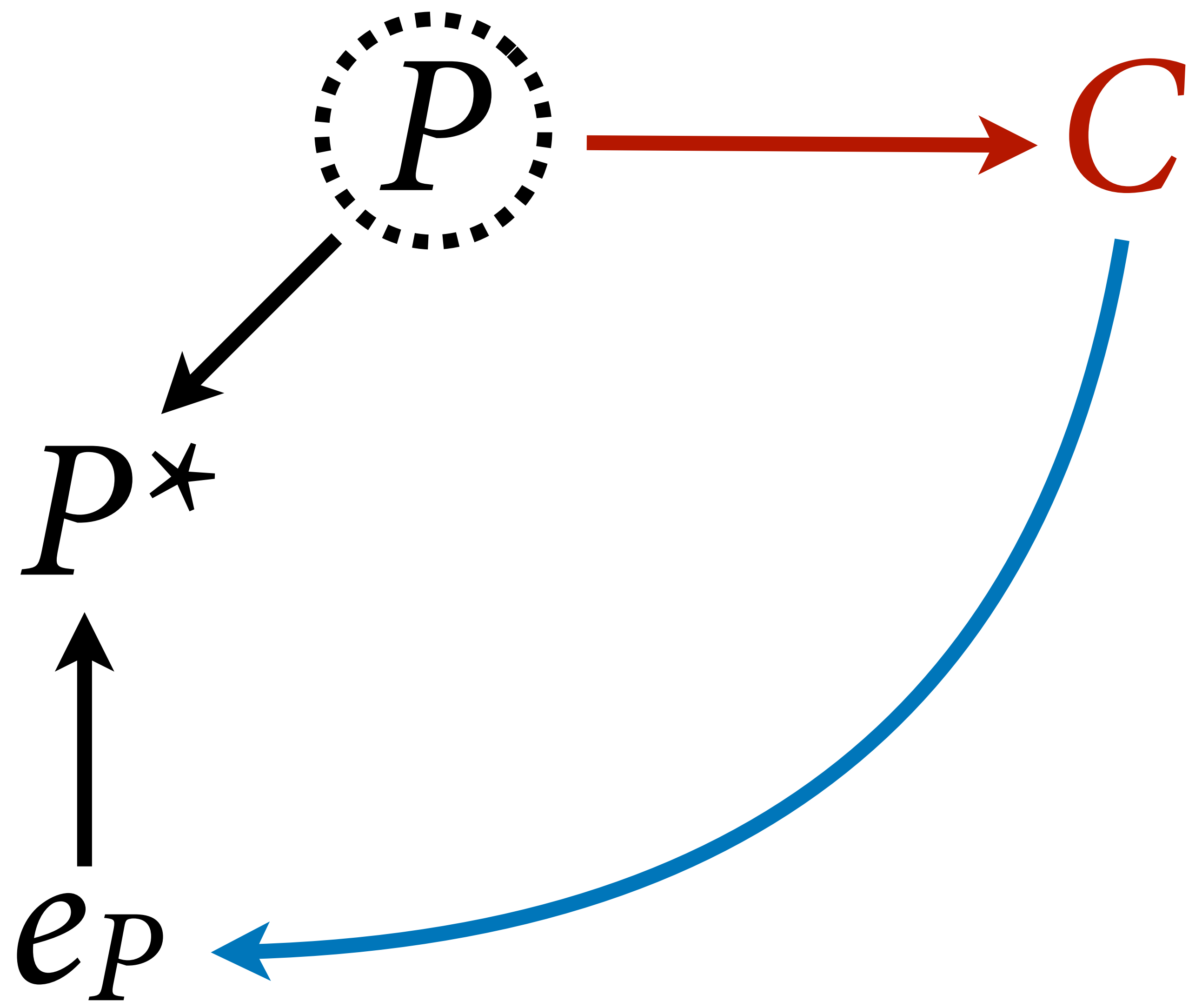
```r
# simple parent-child income example
# recall bias on parental income

N <- 500
P <- rnorm(N)
C <- rnorm(N,0*P)
Pstar <- rnorm(N, 0.8*P + 0.2*C )

mCP <- ulam(
    alist(
        C ~ normal(mu,sigma),
        mu <- a + b*P,
        a ~ normal(0,1),
        b ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=list(P=Pstar,C=C) , chains=4 )

precis(mCP)
```



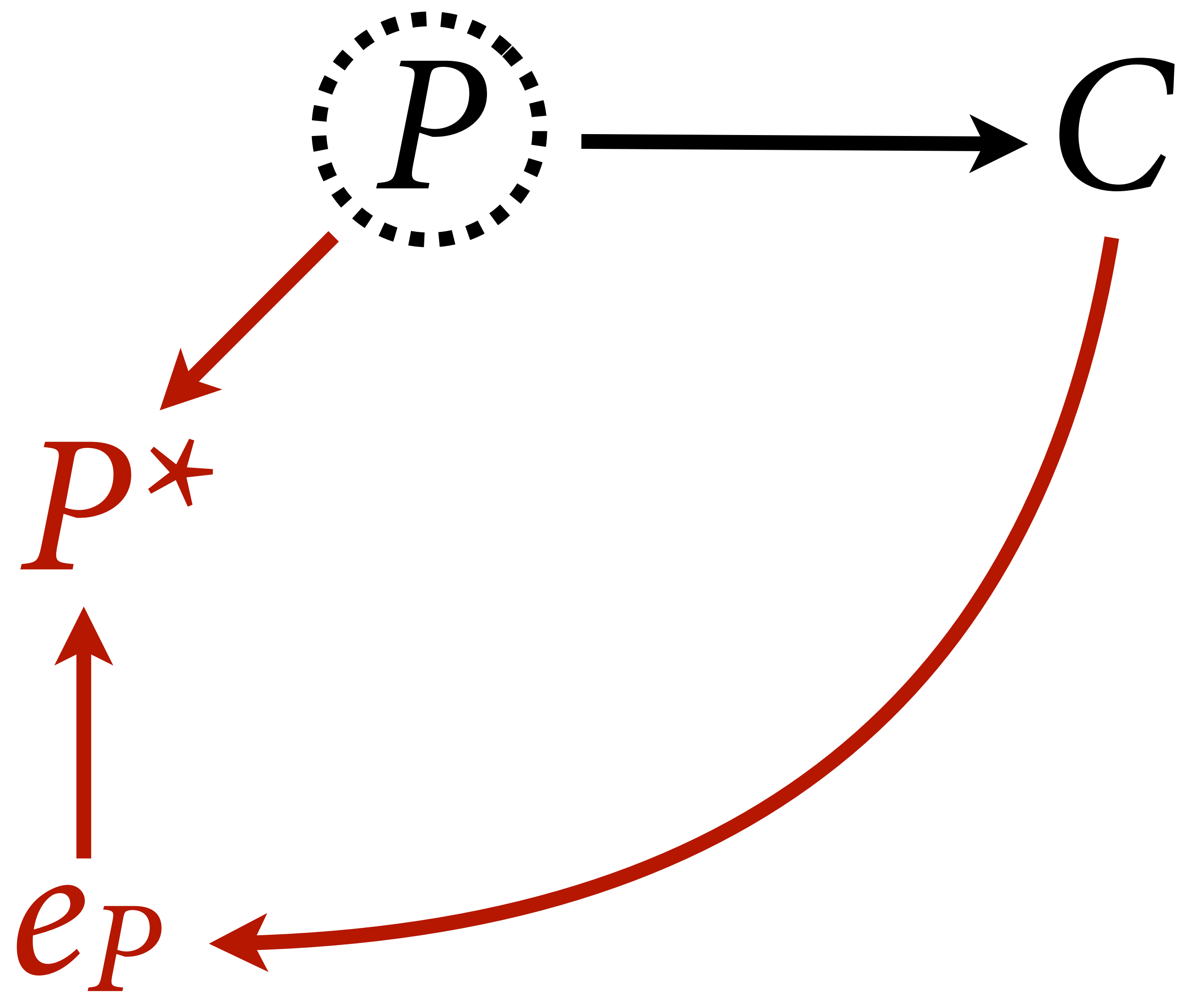
```
# simple parent-child income example
# recall bias on parental income

N <- 500
P <- rnorm(N)
C <- rnorm(N,0*P)
Pstar <- rnorm(N, 0.8*P + 0.2*C )

mCP <- ulam(
    alist(
        C ~ normal(mu,sigma),
        mu <- a + b*P,
        a ~ normal(0,1),
        b ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=list(P=Pstar,C=C) , chains=4 )

precis(mCP)
```



```
# simple parent-child income example
# recall bias on parental income

N <- 500
P <- rnorm(N)
C <- rnorm(N,0*P)
Pstar <- rnorm(N, 0.8*P + 0.2*C )

mCP <- ulam(
    alist(
        C ~ normal(mu,sigma),
        mu <- a + b*P,
        a ~ normal(0,1),
        b ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=list(P=Pstar,C=C) , chains=4 )

precis(mCP)
```

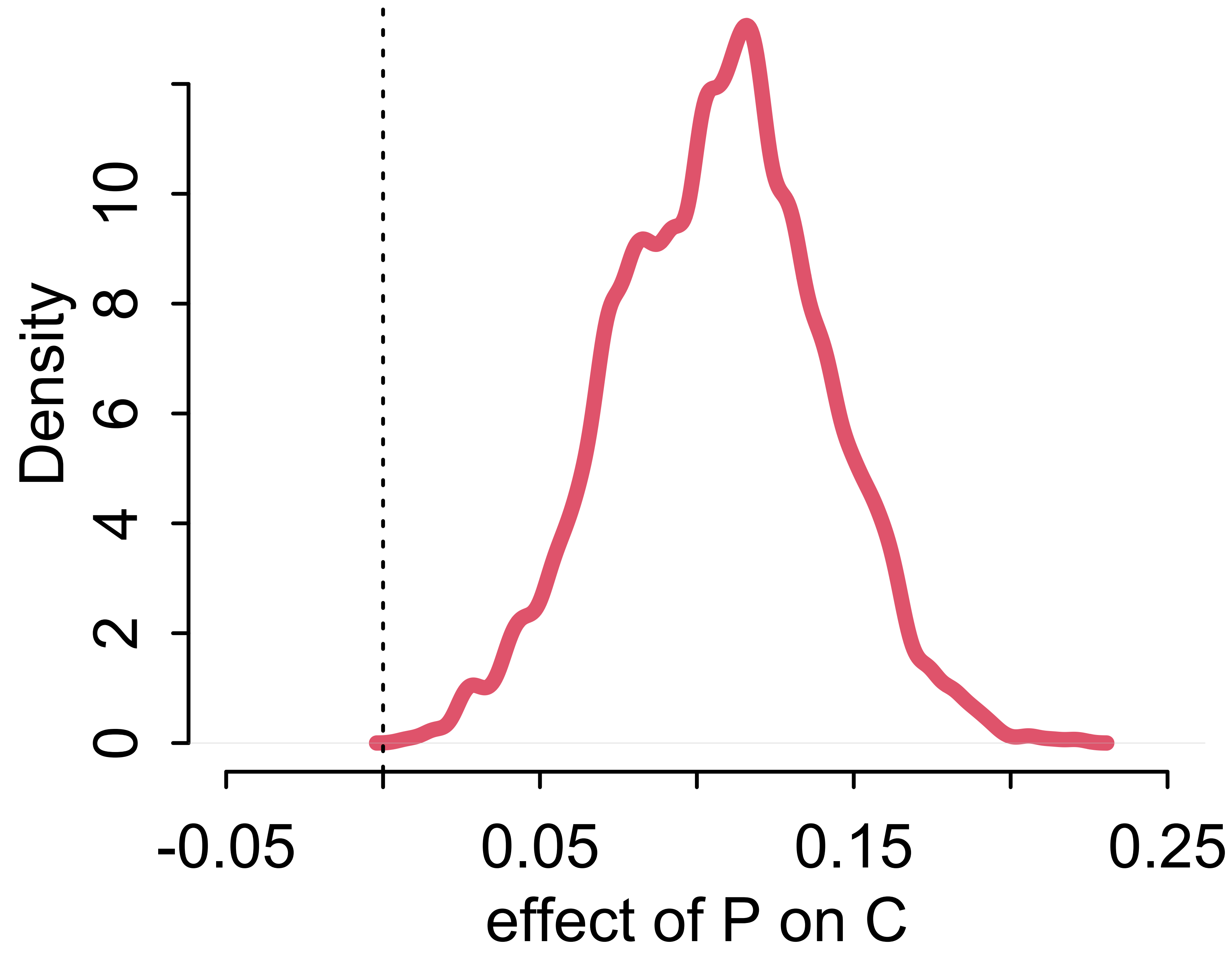


```r
# simple parent-child income example
# recall bias on parental income

N <- 500
P <- rnorm(N)
C <- rnorm(N,0*P)
Pstar <- rnorm(N, 0.8*P + 0.2*C )

mCP <- ulam(
    alist(
        C ~ normal(mu,sigma),
        mu <- a + b*P,
        a ~ normal(0,1),
        b ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=list(P=Pstar,C=C) , chains=4 )

precis(mCP)
```



```r
# simple parent-child income example
# recall bias on parental income

N <- 500
P <- rnorm(N)
C <- rnorm(N,0*P)
Pstar <- rnorm(N, 0.8*P + 0.2*C )

mCP <- ulam(
    alist(
        C ~ normal(mu,sigma),
        mu <- a + b*P,
        a ~ normal(0,1),
        b ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=list(P=Pstar,C=C) , chains=4 )

precis(mCP)
```

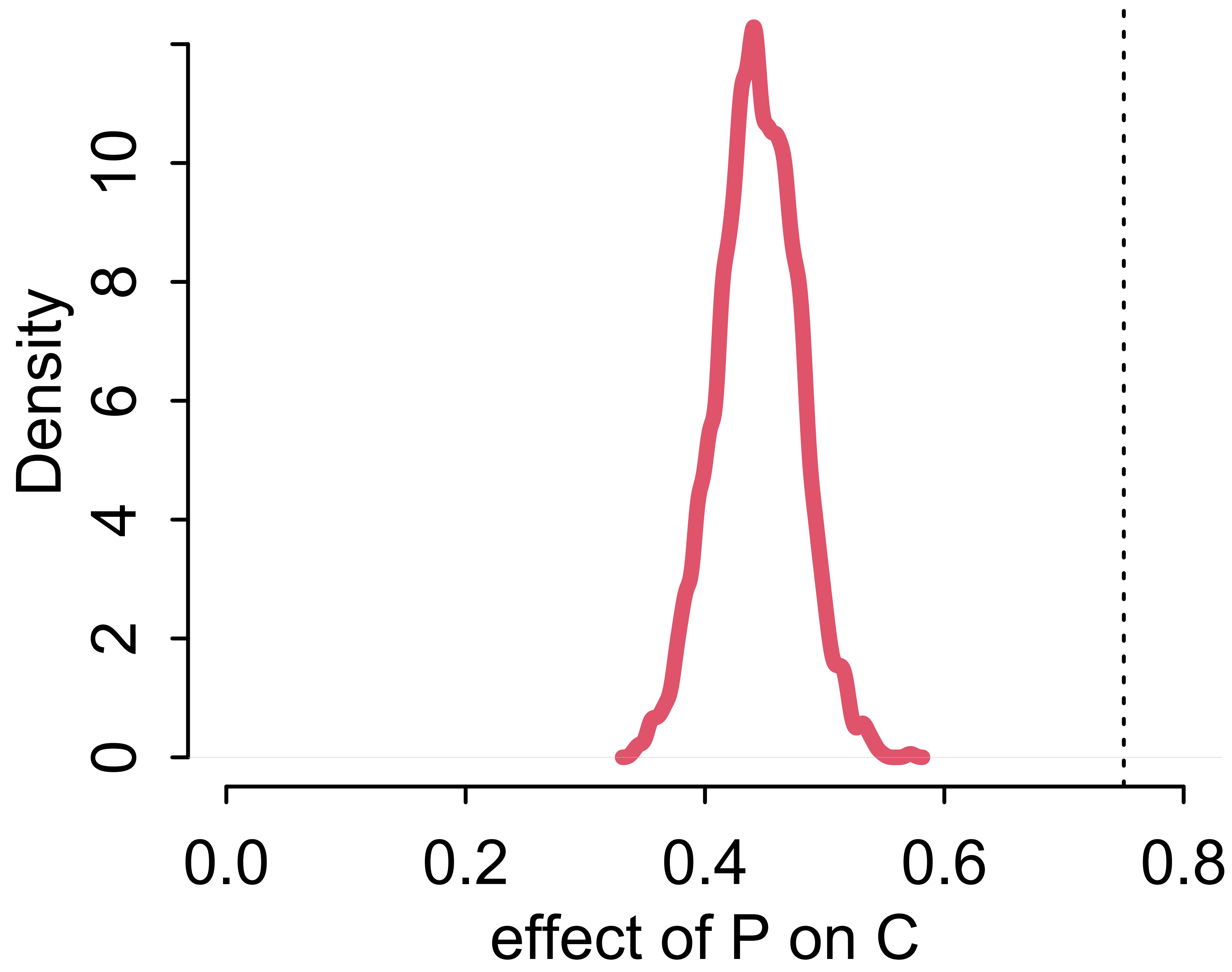
```
> precis(mCP)
      mean   sd  5.5% 94.5% n_eff Rhat4
a     0.01 0.04 -0.06  0.07  1764     1
b     0.11 0.03  0.05  0.16  1993     1
sigma 0.98 0.03  0.93  1.03  1929     1
```



```
# simple parent-child income example
# recall bias on parental income

N <- 500
P <- rnorm(N)
C <- rnorm(N,0.75*P)
Pstar <- rnorm(N, 0.8*P + 0.2*C )

mCP <- ulam(
    alist(
        C ~ normal(mu,sigma),
        mu <- a + b*P,
        a ~ normal(0,1),
        b ~ normal(0,1),
        sigma ~ exponential(1)
    ) , data=list(P=Pstar,C=C) , chains=4 )

precis(mCP)
```

```
> precis(mCP)
```

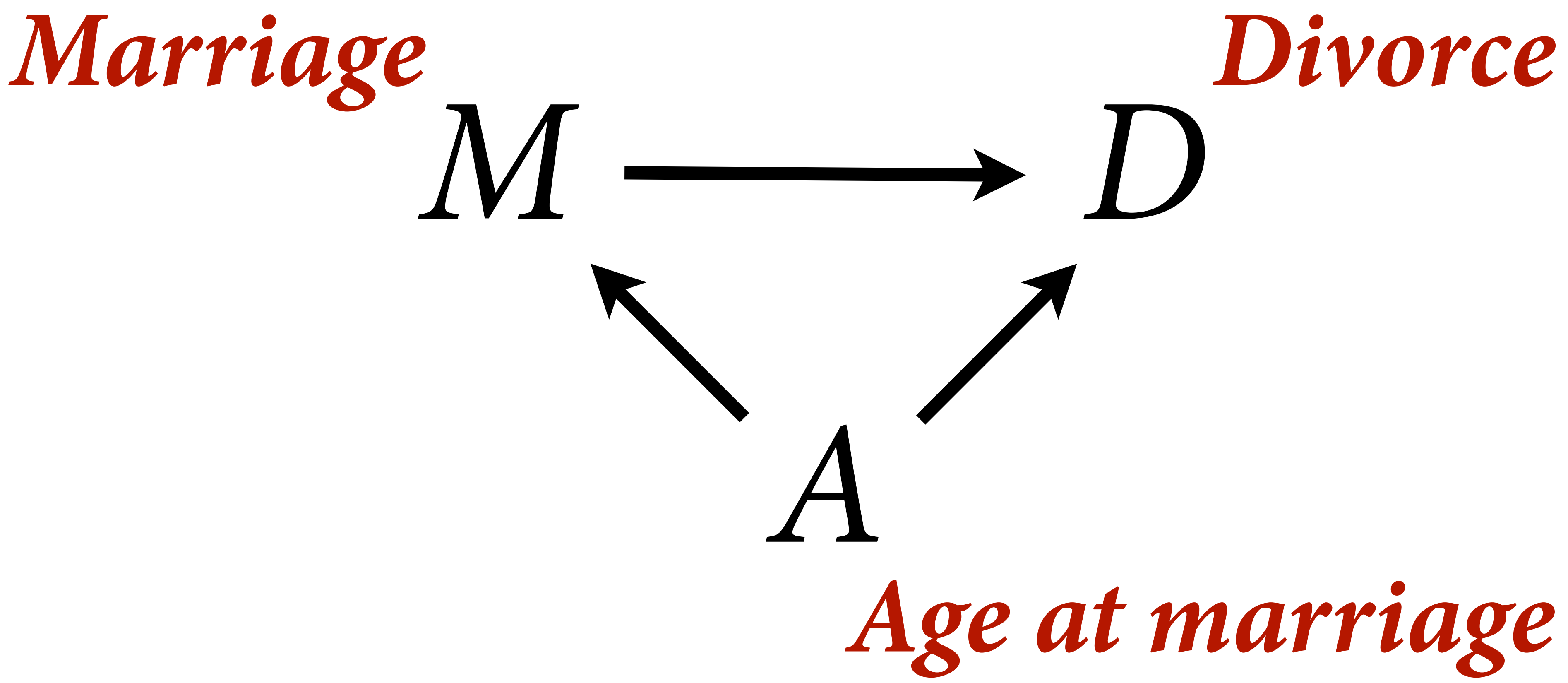
	mean	sd	5.5%	94.5%	n_eff	Rhat4
a	-0.09	0.05	-0.17	-0.02	1782	1
b	0.44	0.03	0.39	0.50	1698	1
sigma	1.05	0.03	1.00	1.11	1947	1



Modeling Measurement

data(WaffleDivorce)

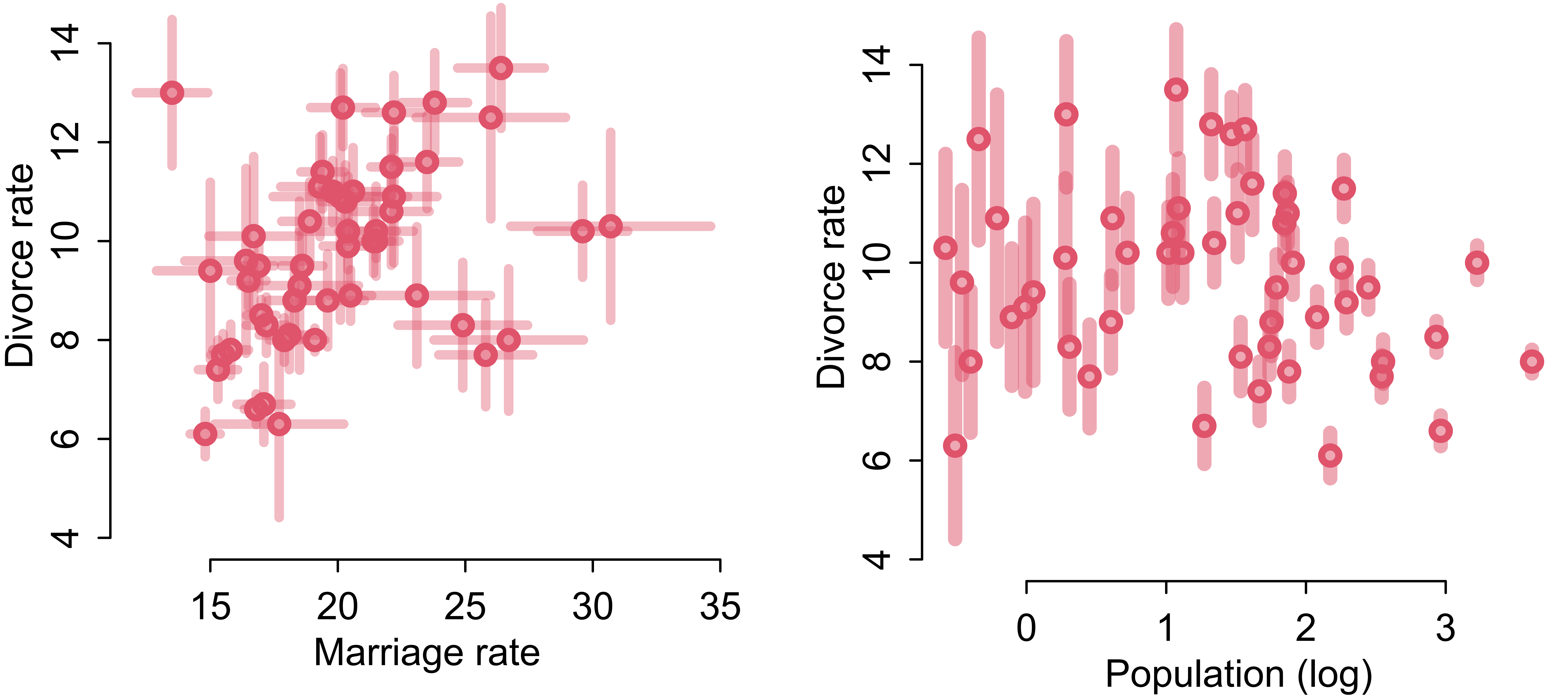
State estimates D, M, A measured with error, error varies by State

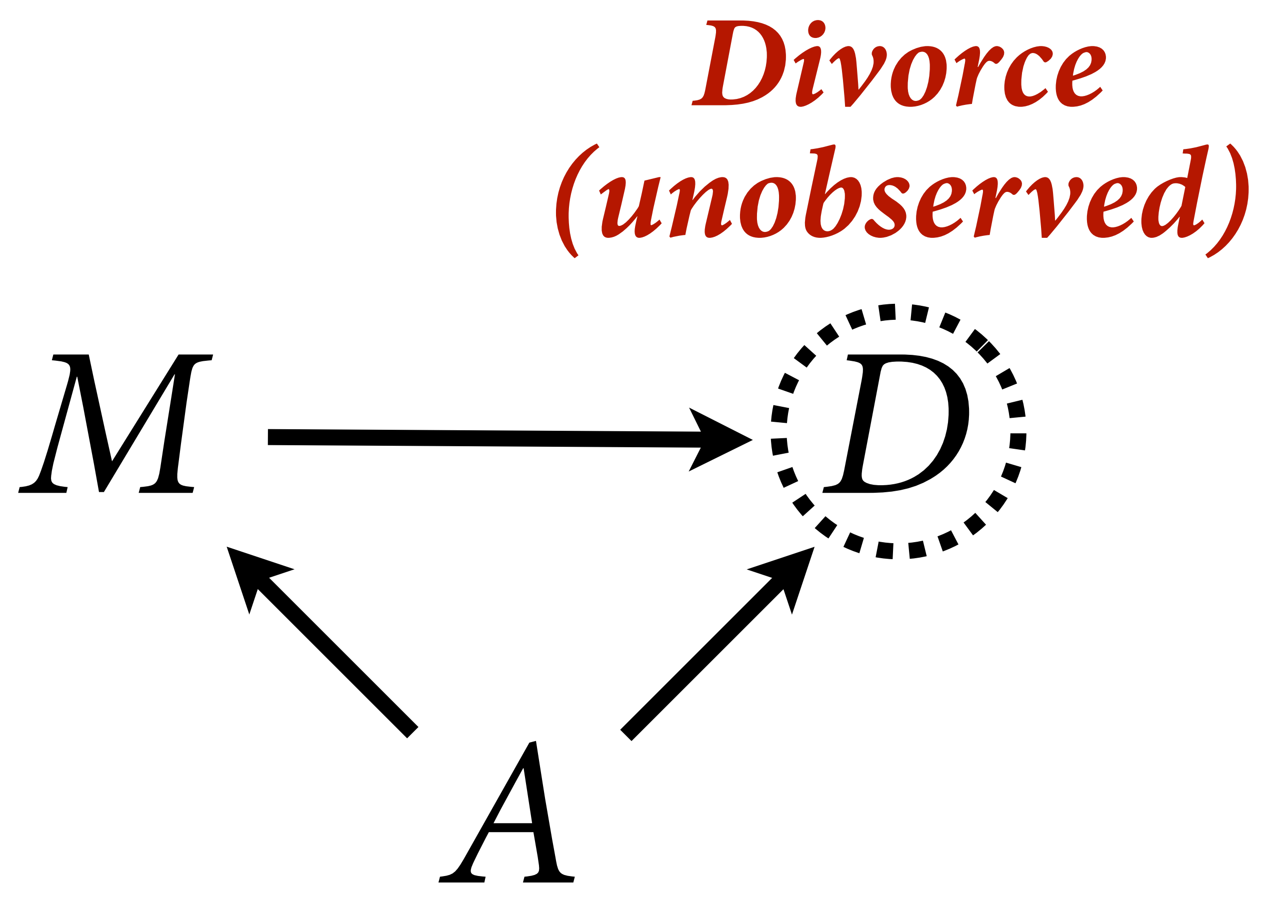


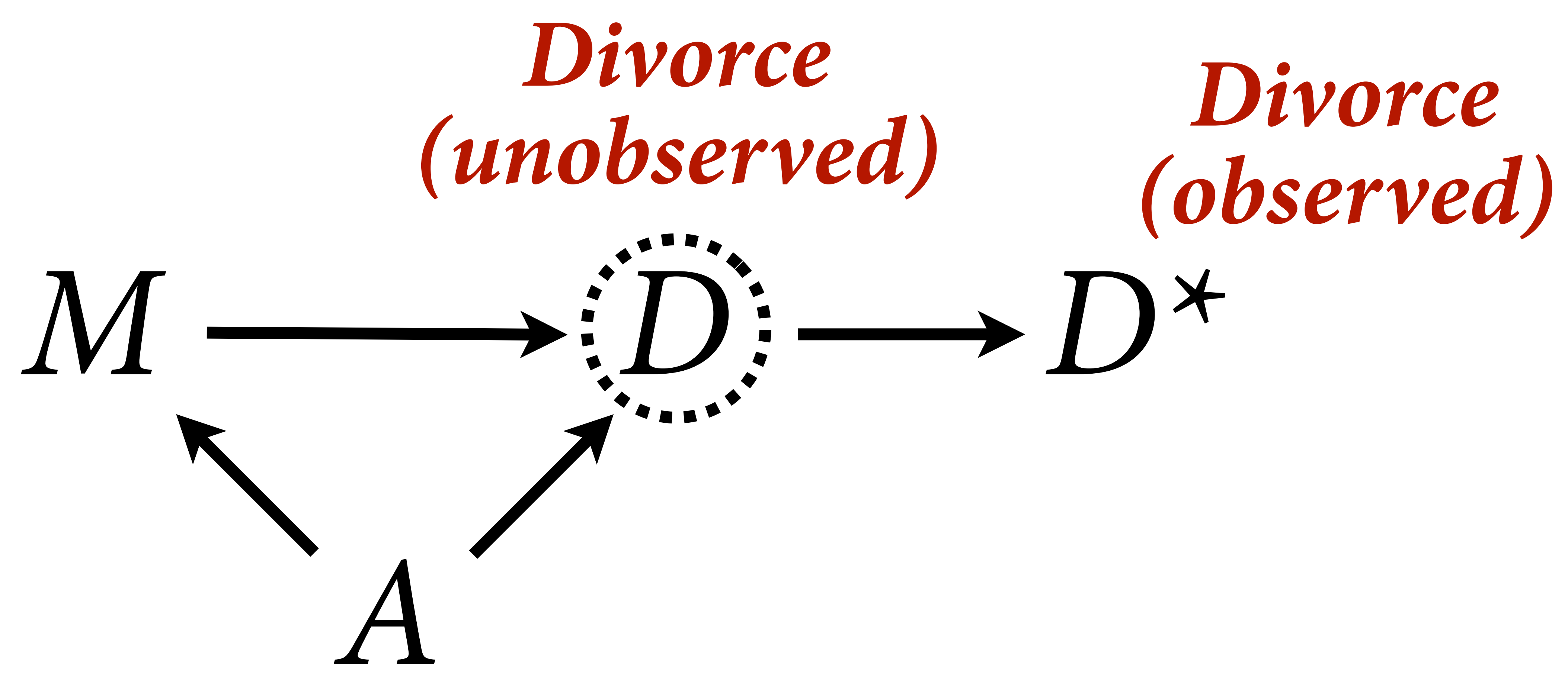
Two problems:

(1) Imbalance in evidence

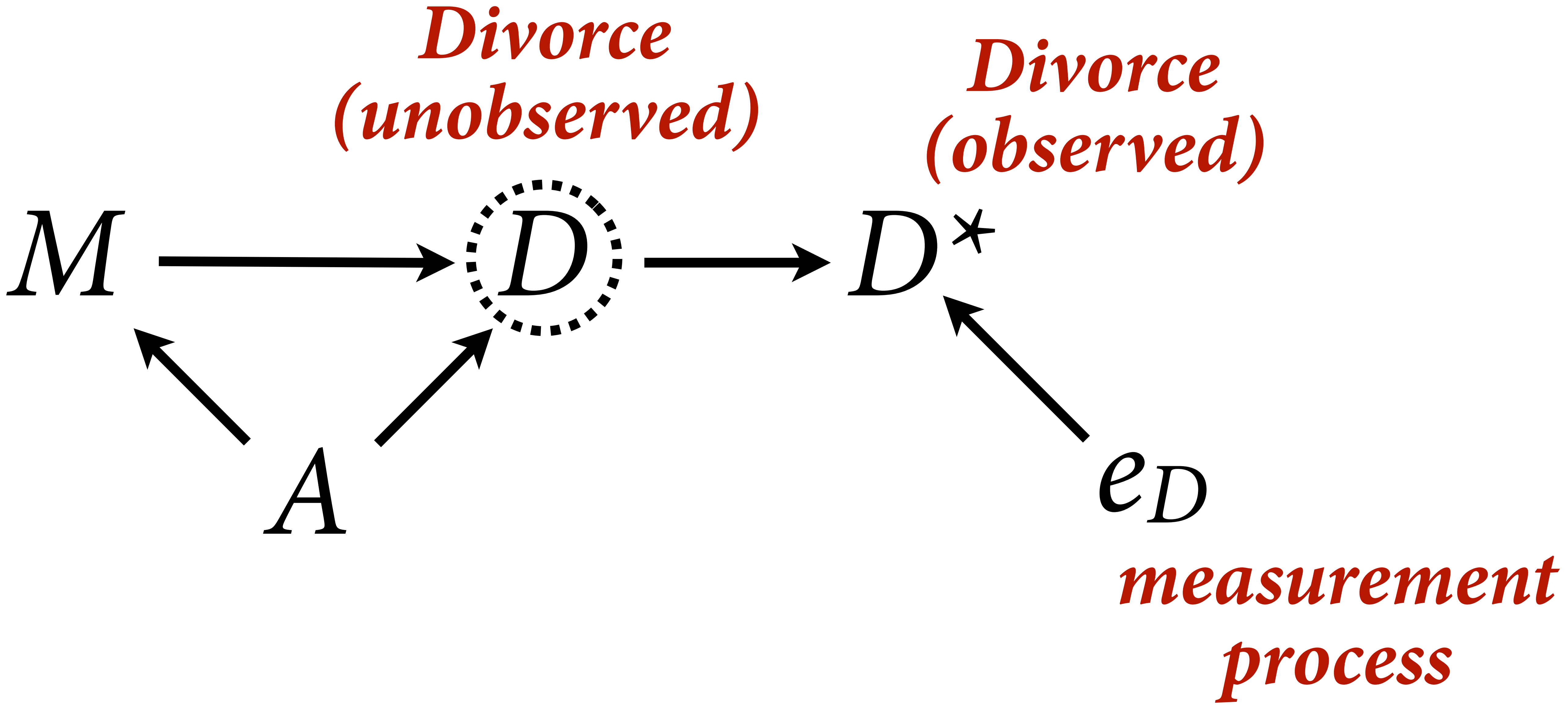
(2) Potential confounding

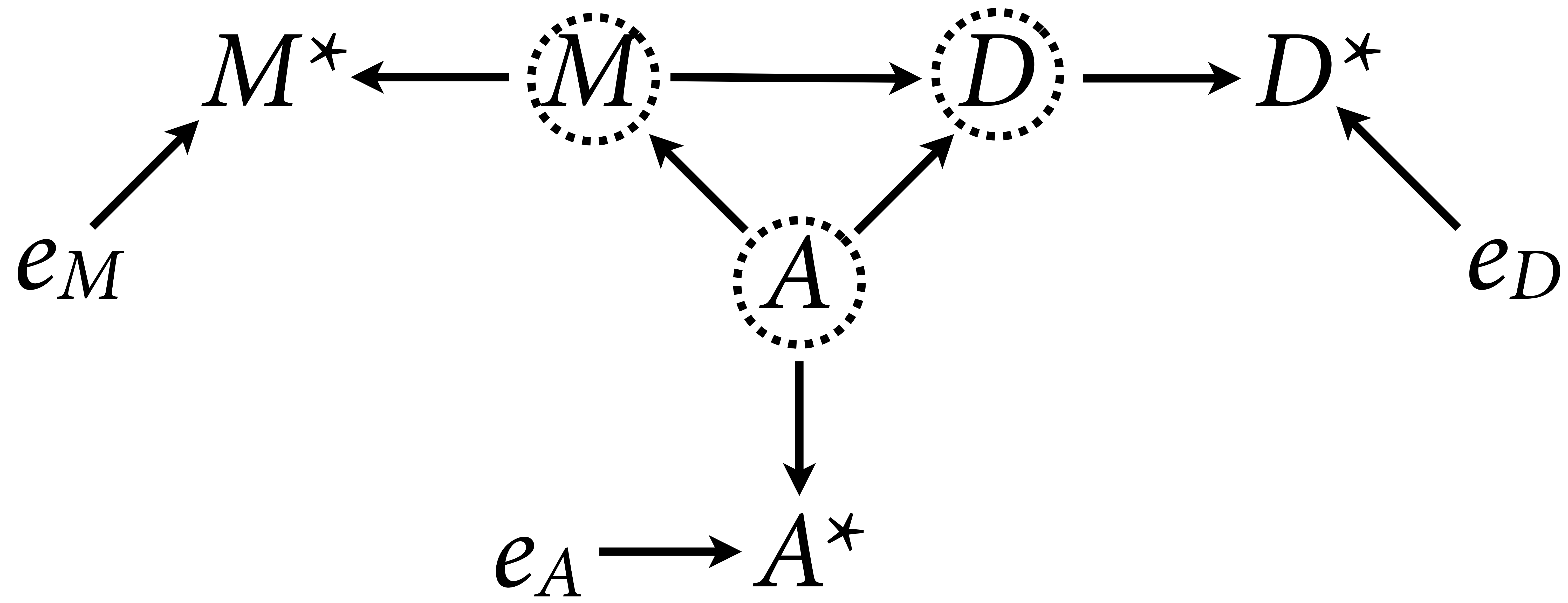

Divorce rate
Marriage rate
4
6
8
10
12
14
15
20
25
30
35
Divorce rate
Population (log)
4
6
8
10
12
14
0
1
2
3


Divorce
(unobserved)
M
D
A

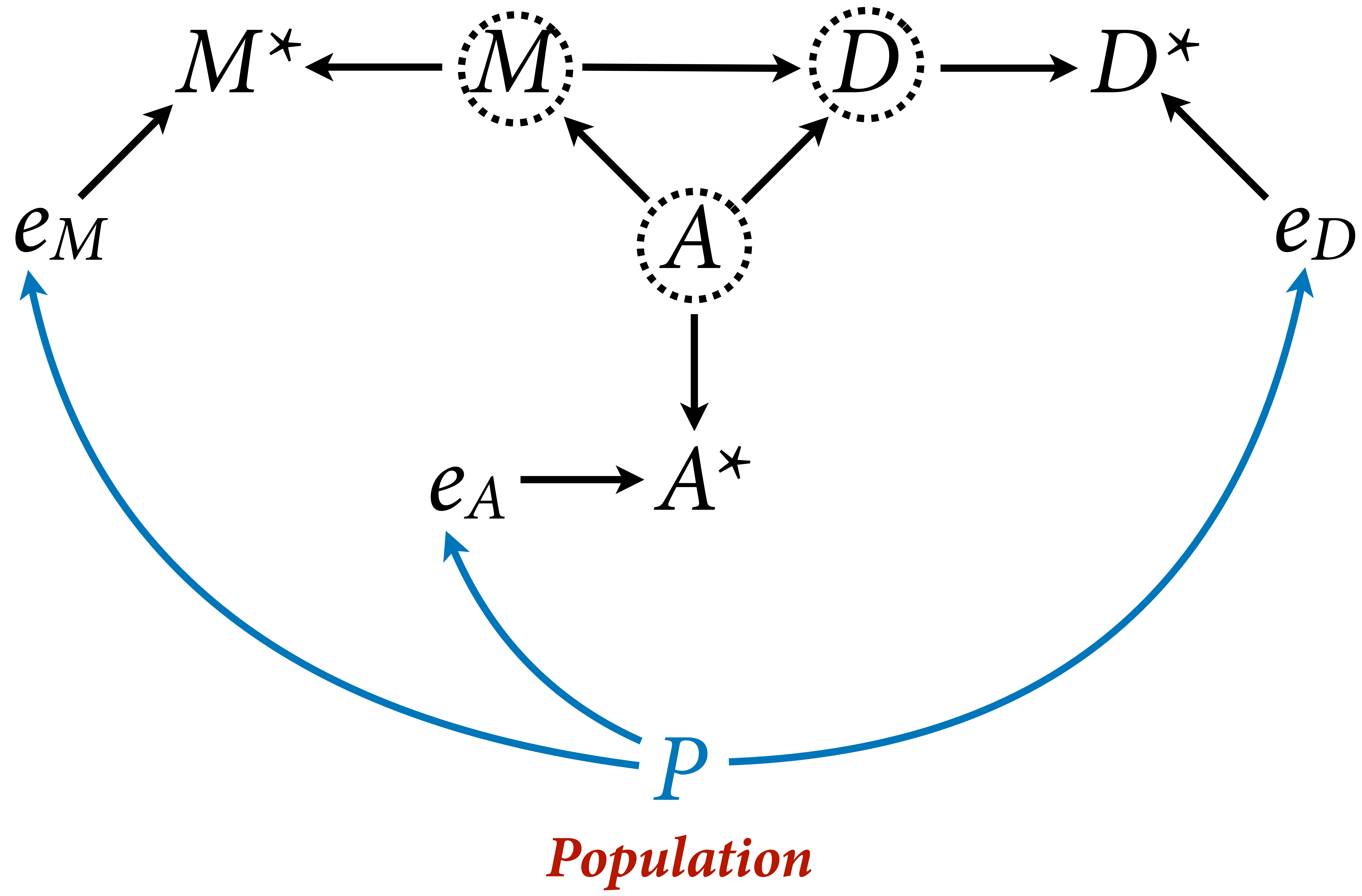


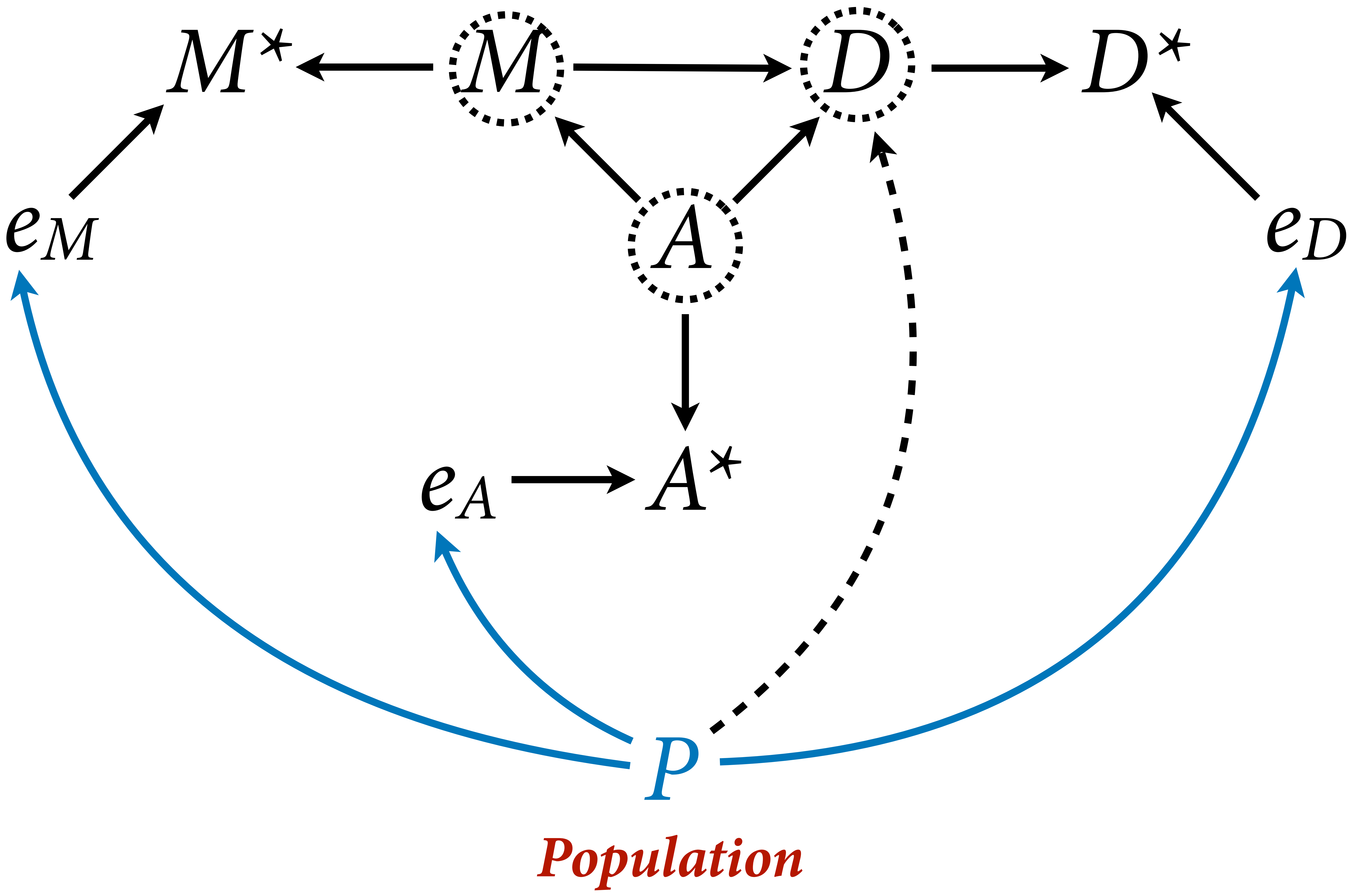
Divorce
(unobserved)
Divorce
(observed)
M
D
D*
A

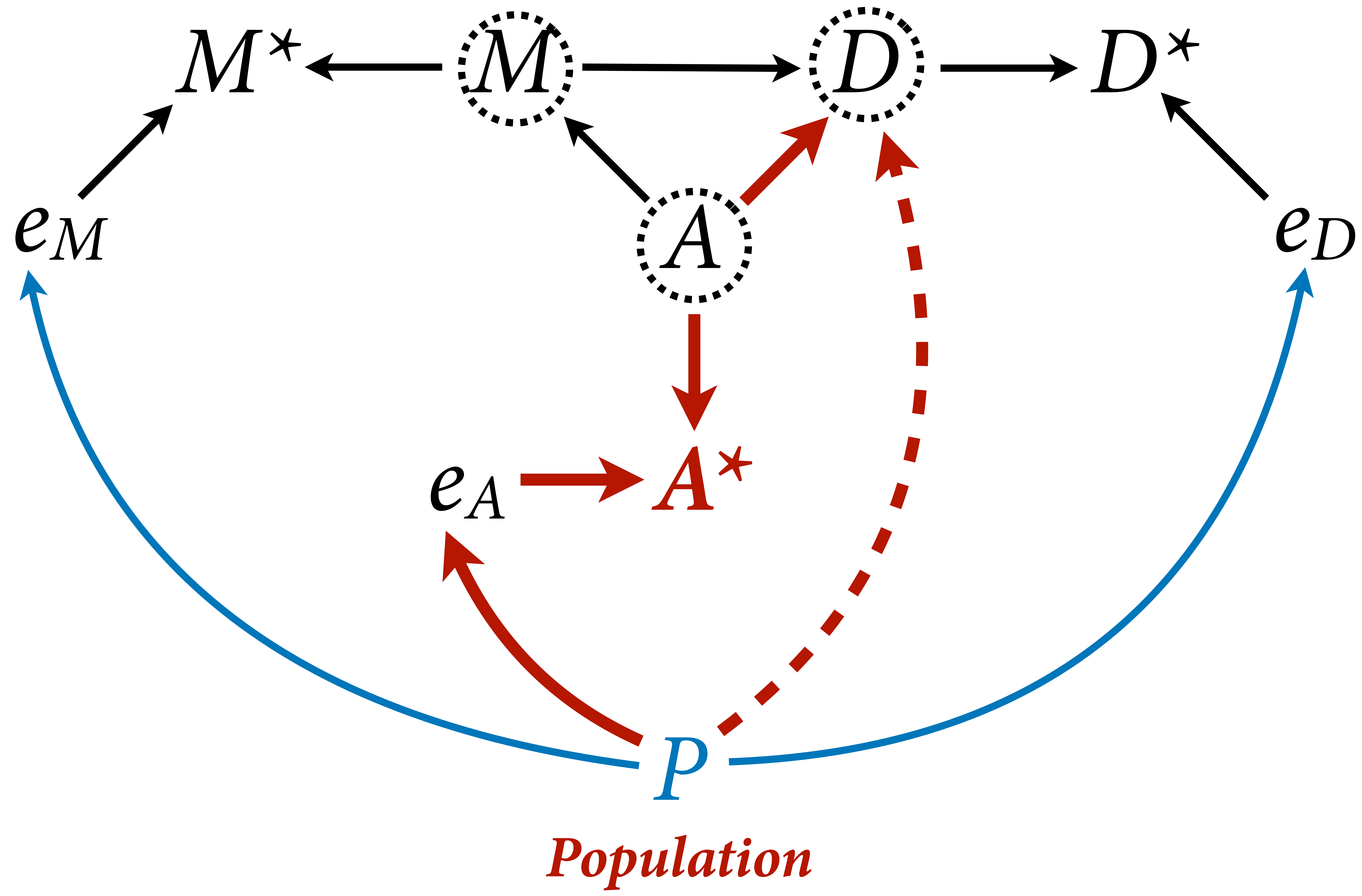

Divorce
(unobserved)
Divorce
(observed)
M
D
D*
A
eD
measurement
process



M^*
M
D
D^*
e_M
A
e_D
e_A
A^*


M*
M
D
D*
e_M
A
e_D
e_A
A*
P
Population


M^*
M
D
D^*
e_M
A
e_D
e_A
A^*
P
Population

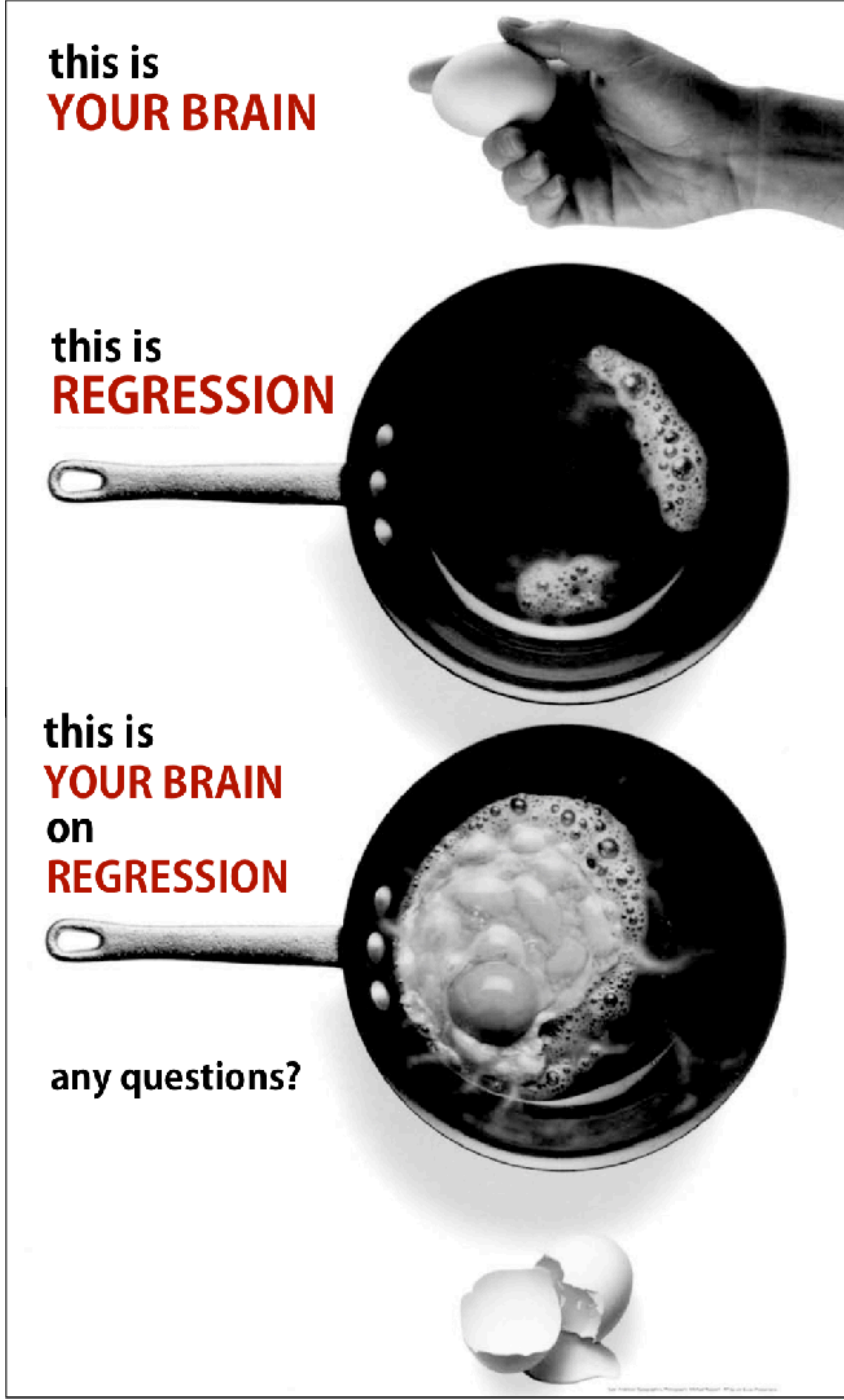

M*
M
D
D*
e_M
A
e_D
e_A
A*
P
Population

Thinking Like a Graph

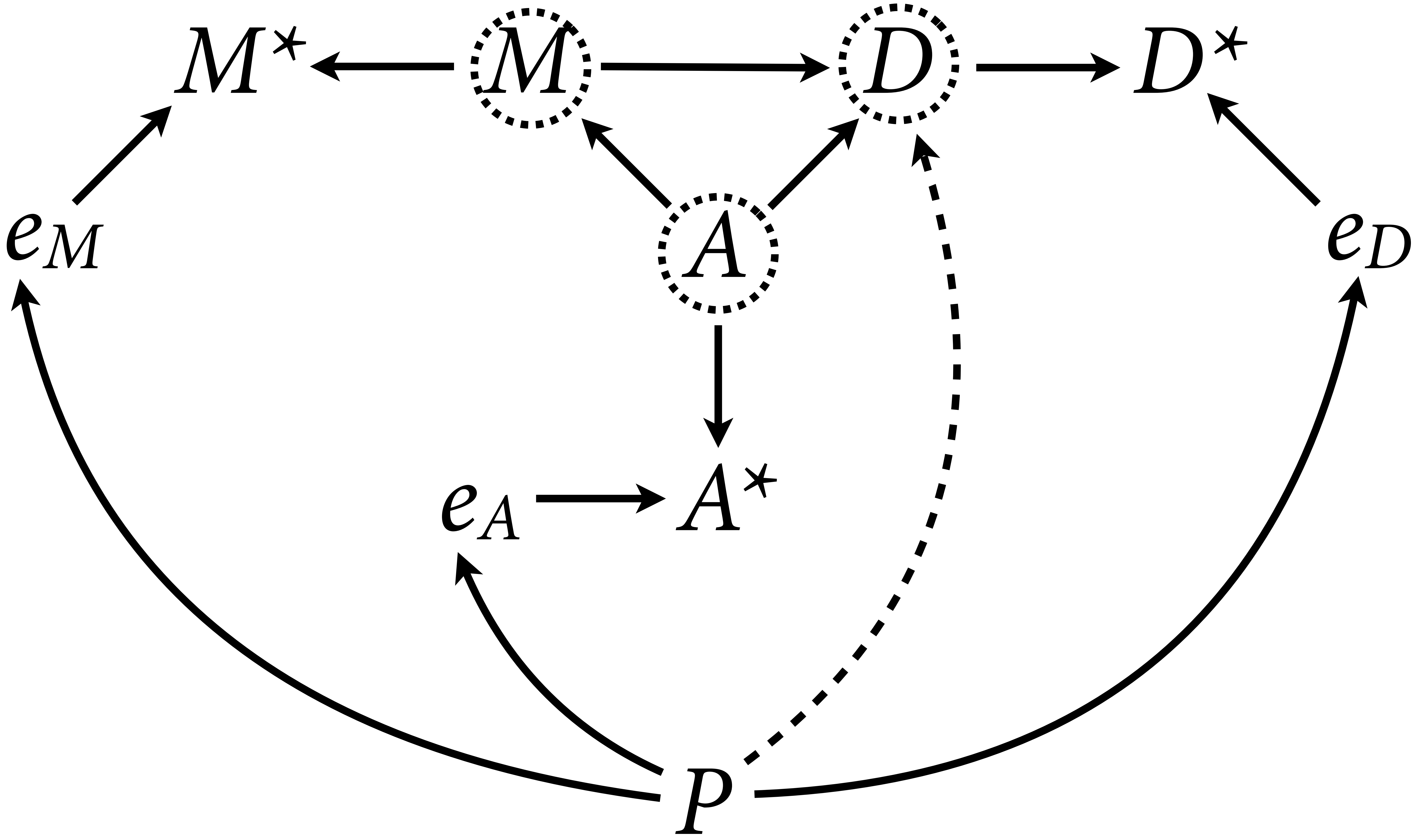
Regressions (GLMMs) are special case machines

Thinking like a regression: Which predictor variables do I use?

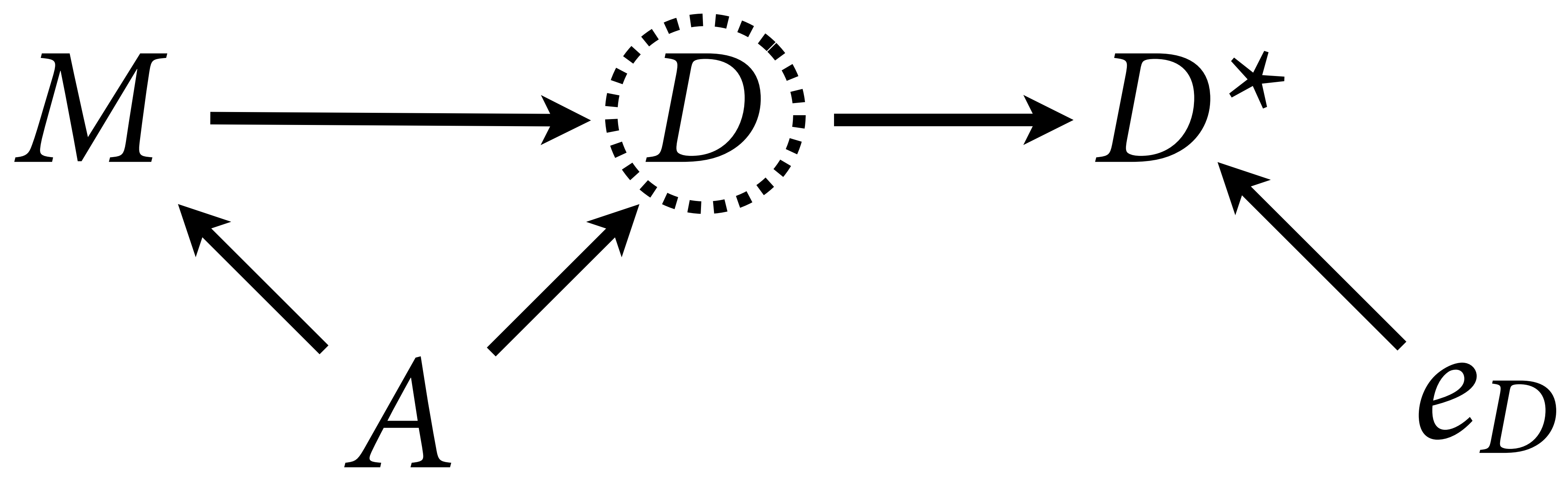
Thinking like a graph: How do I model the network of causes?



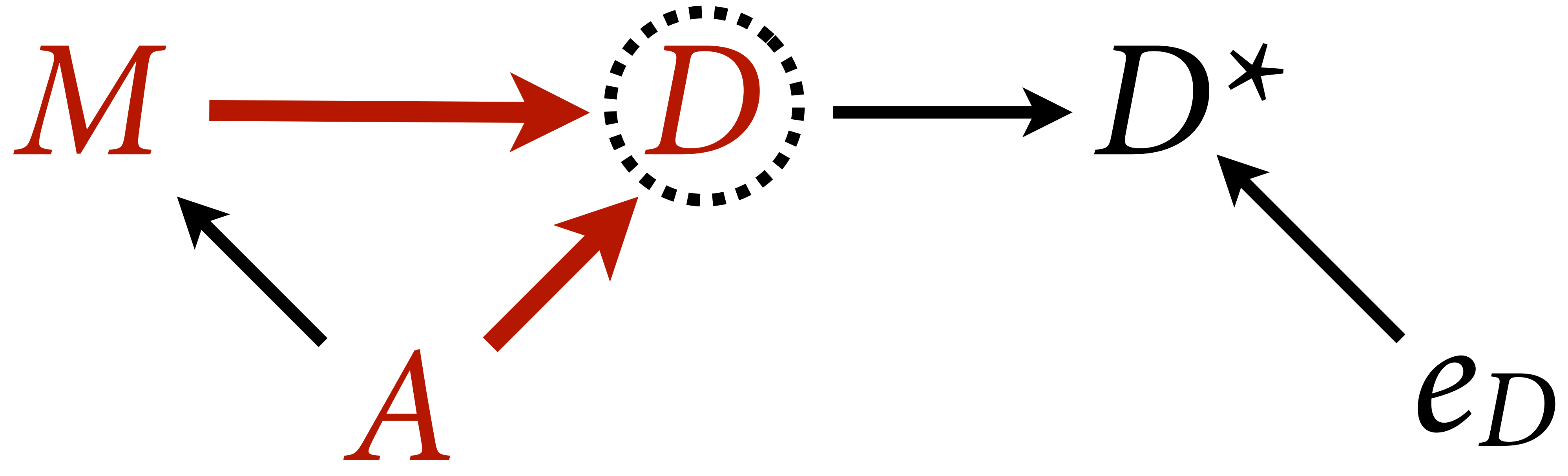
Thinking Like a Graph



Thinking Like a Graph

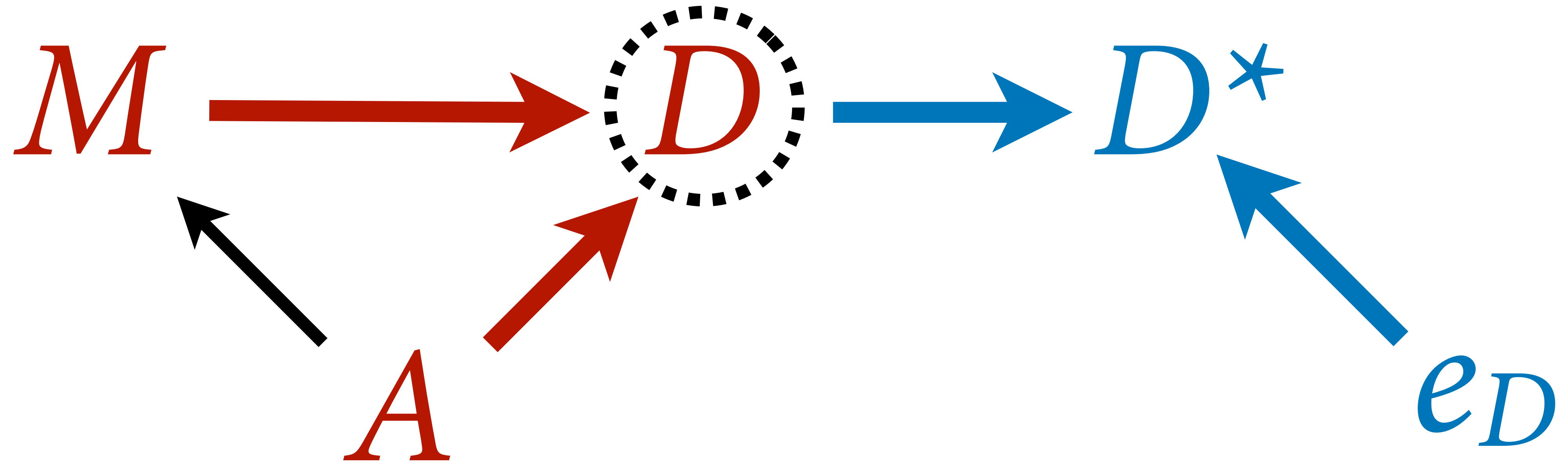


Thinking Like a Graph



Model for D

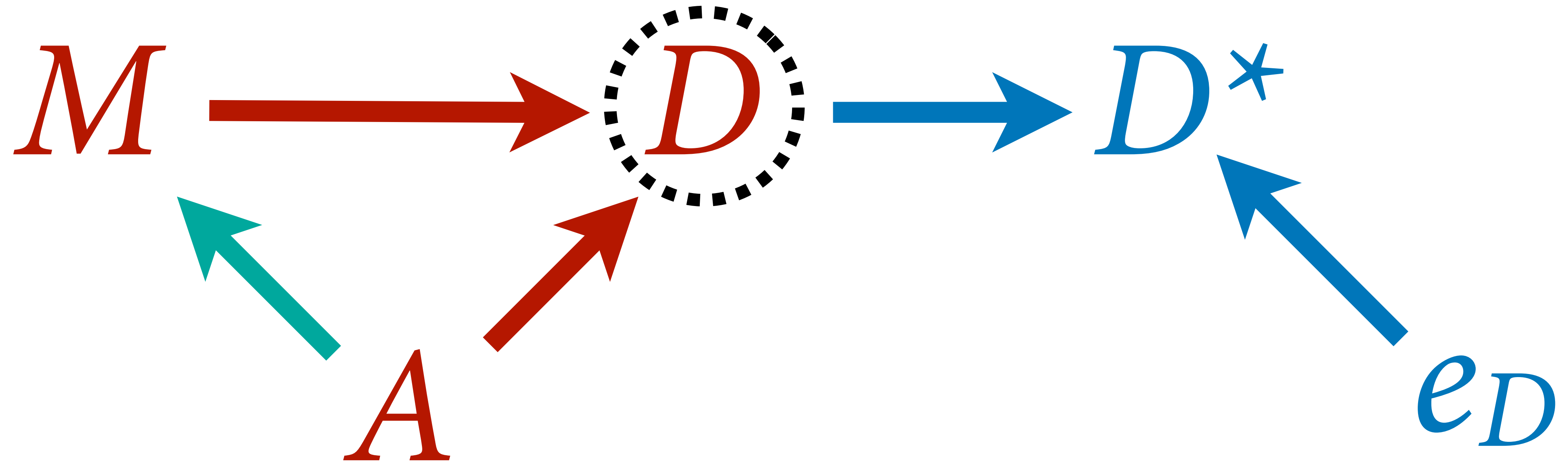
Thinking Like a Graph



Model for D

*Model for D**

Thinking Like a Graph



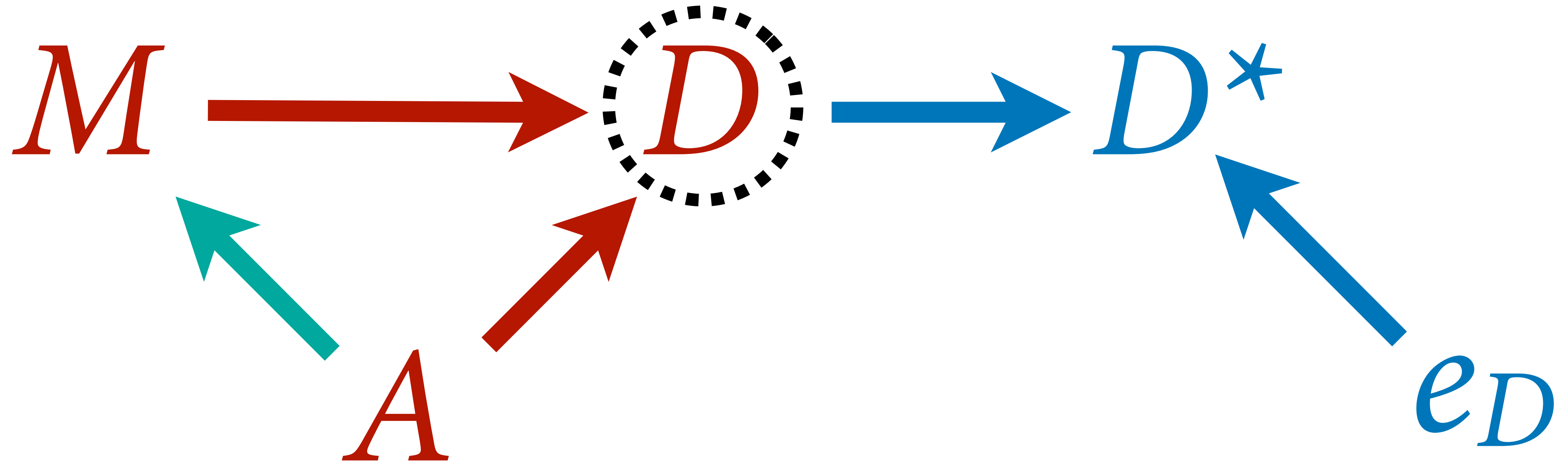
Model for D

*Model for D**

Model for M

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$

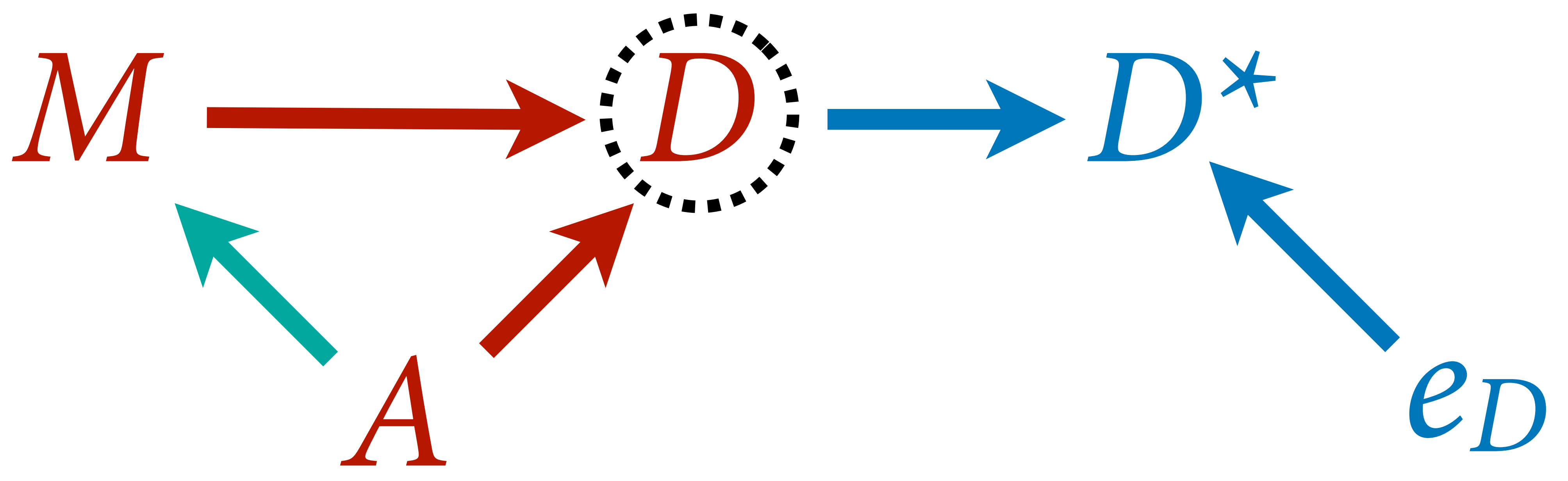
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$



$$D_i^\star = D_i + e_{D,i}$$

$$e_{D,i} \sim \text{Normal}(0, S_i)$$

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

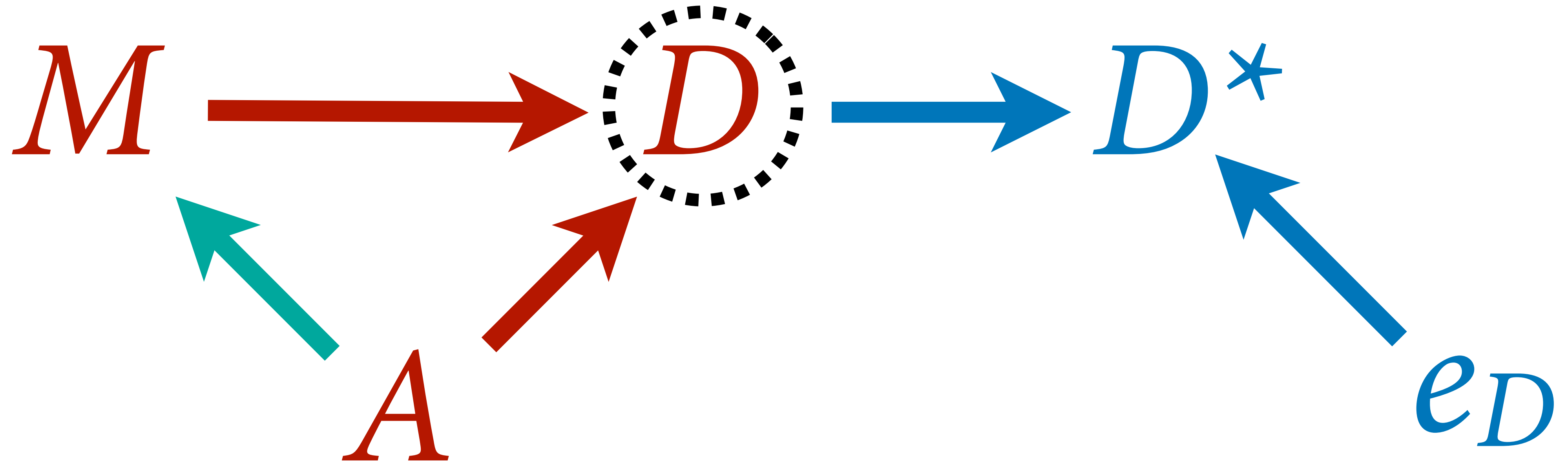


$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

PAUSE

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$



$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

```r
## R code 15.3
dlist <- list(
    D_obs = standardize( d$Divorce ),
    D_sd = d$Divorce.SE / sd( d$Divorce ),
    M = standardize( d$Marriage ),
    A = standardize( d$MedianAgeMarriage ),
    N = nrow(d)
)

m15.1 <- ulam(
    alist(
        # model for D* (observed)
        D_obs ~ dnorm( D_true , D_sd ),

        # model for D (unobserved)
        vector[N]:D_true ~ dnorm( mu , sigma ),
        mu <- a + bA*A + bM*M,
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        bM ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ) , data=dlist , chains=4 , cores=4 )

precis( m15.1 , depth=2 )
```

$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

```r
## R code 15.3
dlist <- list(
    D_obs = standardize( d$Divorce ),
    D_sd = d$Divorce.SE / sd( d$Divorce ),
    M = standardize( d$Marriage ),
    A = standardize( d$MedianAgeMarriage ),
    N = nrow(d)
)

m15.1 <- ulam(
    alist(
        # model for D* (observed)
        D_obs ~ dnorm( D_true , D_sd ),

        # model for D (unobserved)
        vector[N]:D_true ~ dnorm( mu , sigma ),
        mu <- a + bA*A + bM*M,
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        bM ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ) , data=dlist , chains=4 , cores=4 )

precis( m15.1 , depth=2 )
```

$$\boxed{D_i^\star \sim \text{Normal}(D_i, S_i)}$$

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

```r
## R code 15.3
dlist <- list(
    D_obs = standardize( d$Divorce ),
    D_sd = d$Divorce.SE / sd( d$Divorce ),
    M = standardize( d$Marriage ),
    A = standardize( d$MedianAgeMarriage ),
    N = nrow(d)
)

m15.1 <- ulam(
    alist(
        # model for D* (observed)
        D_obs ~ dnorm( D_true , D_sd ),

        # model for D (unobserved)
        vector[N]:D_true ~ dnorm( mu , sigma ),
        mu <- a + bA*A + bM*M,
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        bM ~ dnorm(0,0.5),
        sigma ~ dexp(1)
    ) , data=dlist , chains=4 , cores=4 )

precis( m15.1 , depth=2 )
```

$$D_i^{\star} \sim \text{Normal}(D_i, S_i)$$

$$\boxed{D_i \sim \text{Normal}(\mu_i, \sigma)}$$

$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

```
## R code 15.3
dlist <- list
    D_obs = s
    D_sd = d$
    M = stand
    A = stand
    N = nrow(
)

m15.1 <- ulam
    alist(
        # mod
        D_obs

        # mod
        vecto
        mu <
        a ~ d
        bA ~
        bM ~
        sigma
    ) , data=

precis( m15.1
```

```
> precis( m15.1 , depth=2 )
```

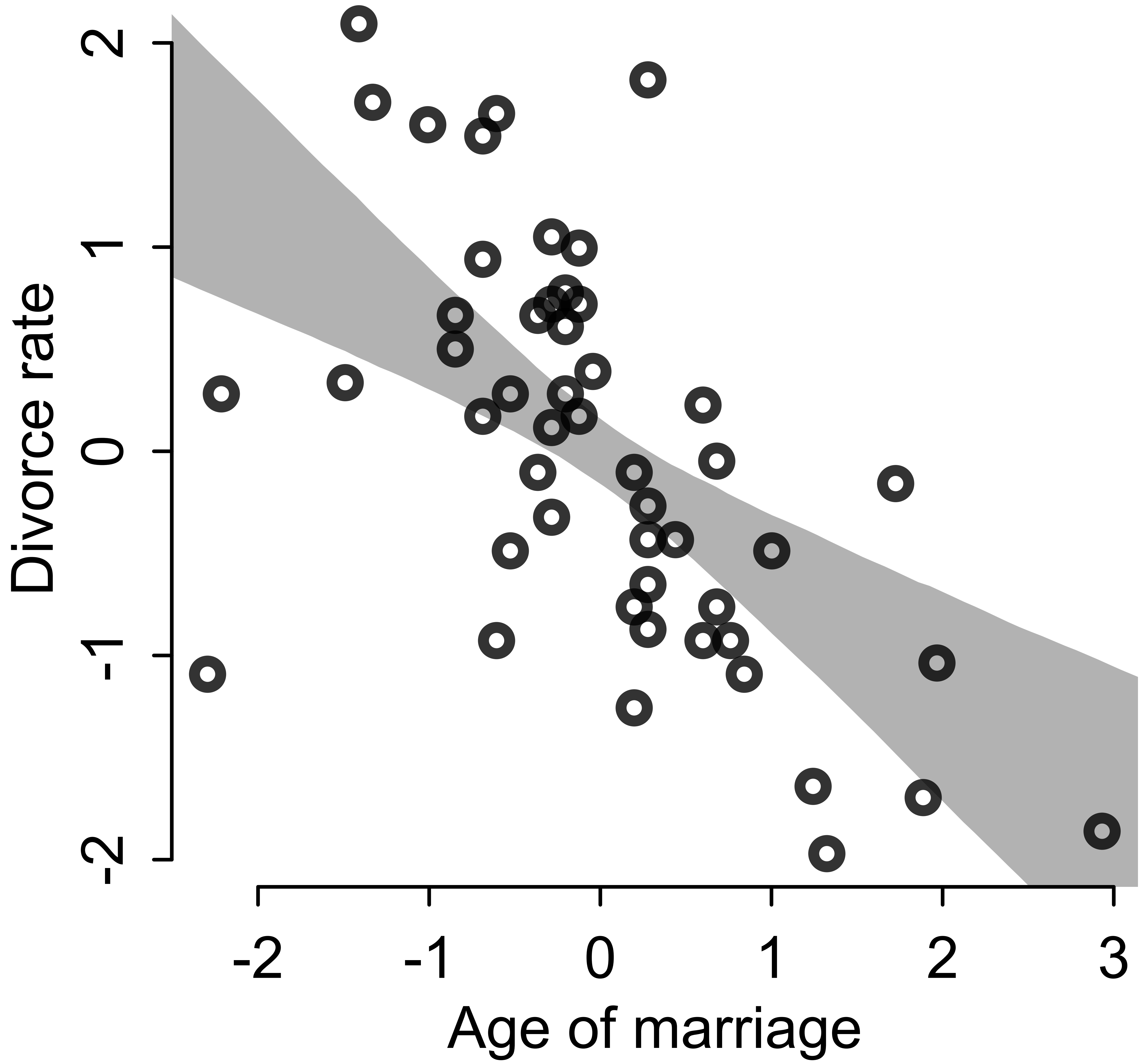
	mean	sd	5.5%	94.5%	n_eff	Rhat4
D_true[1]	1.17	0.37	0.59	1.78	2219	1
D_true[2]	0.69	0.57	-0.23	1.59	2895	1
D_true[3]	0.42	0.34	-0.13	0.97	2582	1
D_true[4]	1.42	0.48	0.66	2.21	1938	1
D_true[5]	-0.90	0.13	-1.10	-0.70	2855	1
D_true[6]	0.65	0.40	0.02	1.28	2398	1
D_true[7]	-1.37	0.36	-1.95	-0.83	2432	1
D_true[8]	-0.35	0.48	-1.10	0.40	2515	1
D_true[9]	-1.90	0.59	-2.84	-0.94	2053	1
D_true[10]	-0.62	0.16	-0.87	-0.36	3370	1
D_true[11]	0.77	0.28	0.34	1.20	2748	1
D_true[12]	-0.57	0.47	-1.29	0.20	2154	1
D_true[13]	0.19	0.51	-0.63	0.99	1356	1
D_true[14]	-0.87	0.23	-1.22	-0.49	3601	1
D_true[15]	0.56	0.30	0.07	1.06	3677	1
D_true[16]	0.28	0.38	-0.33	0.88	2677	1
D_true[17]	0.51	0.41	-0.15	1.17	2972	1
D_true[18]	1.25	0.36	0.70	1.83	2443	1
D_true[19]	0.42	0.39	-0.21	1.03	3487	1
D_true[20]	0.41	0.52	-0.42	1.27	1558	1
D_true[21]	-0.56	0.32	-1.06	-0.04	3063	1
D_true[22]	-1.10	0.27	-1.53	-0.68	2963	1
D_true[23]	-0.27	0.26	-0.67	0.14	2819	1
D_true[24]	-1.00	0.30	-1.45	-0.53	2529	1
D_true[25]	0.43	0.40	-0.19	1.09	2989	1
D_true[26]	-0.04	0.31	-0.55	0.45	3276	1
D_true[27]	-0.02	0.48	-0.81	0.72	3308	1

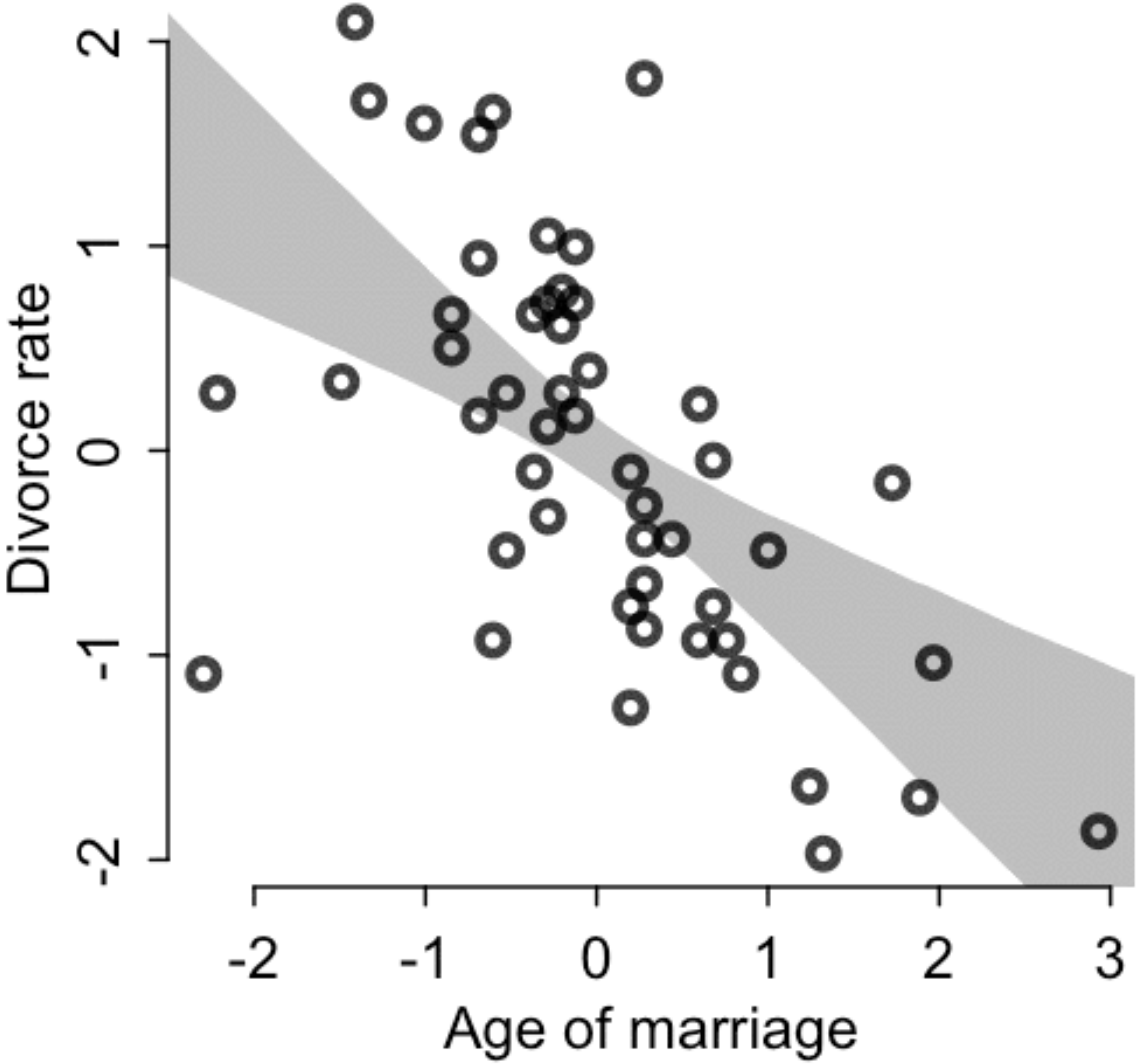
$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

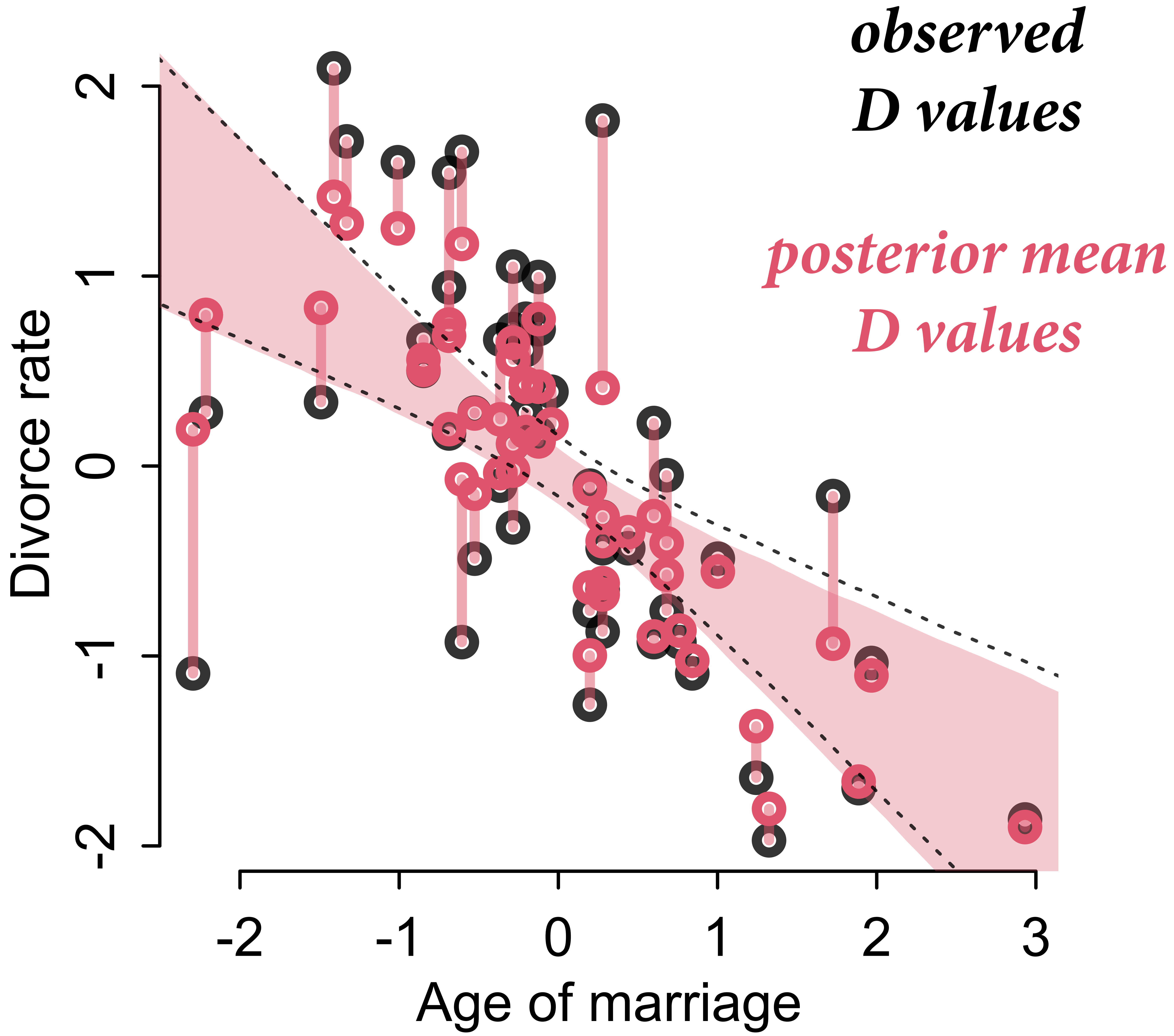
$$\boxed{D_i \sim \text{Normal}(\mu_i, \sigma)}$$

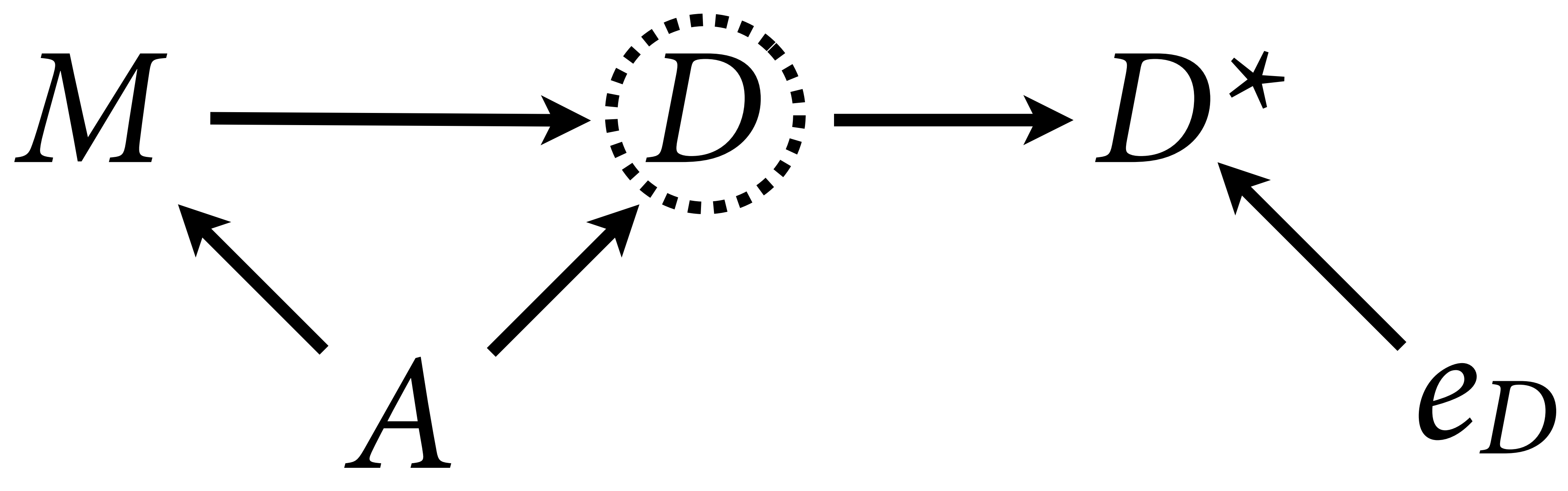
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

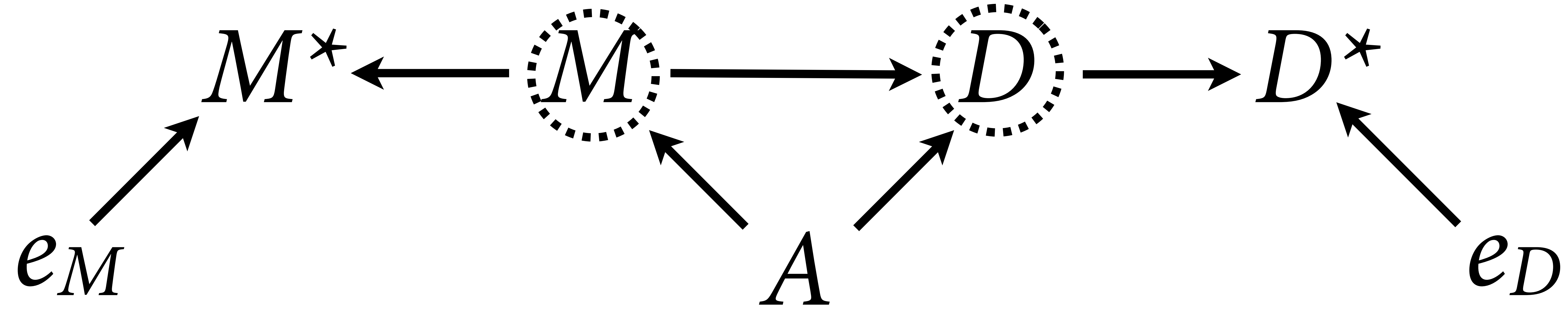

Divorce rate
2
1
0
-1
-2
-2
-1
0
1
2
3
Age of marriage


Divorce rate
2
1
0
-1
-2
-2
-1
0
1
2
3
Age of marriage


Divorce rate
2
1
0
-1
-2
-2
-1
0
1
2
3
Age of marriage


observed
D values
posterior mean
D values
Divorce rate
Age of marriage
-2
-1
0
1
2
3
2
1
0
-1
-2

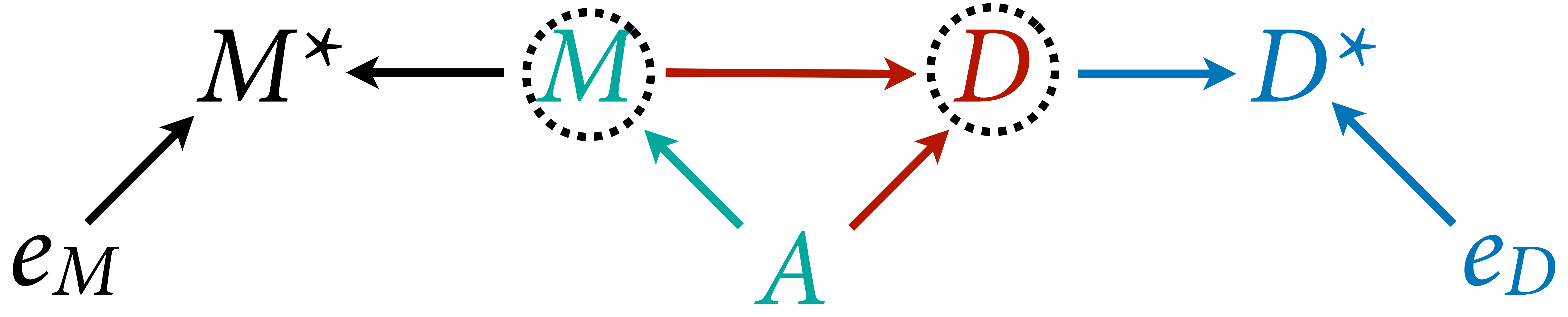

M
D
D*
A
e_D



M^*
M
D
D^*
e_M
A
e_D

$$D_i \sim \mathrm{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

$$D_i^\star \sim \mathrm{Normal}(D_i, S_i)$$



$$D_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

$$M_i^\star \sim \text{Normal}(M_i, T_i)$$

$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

$$\begin{tikzcd}
 & M^\star & M \arrow[l] \arrow[rr] & & D \arrow[r] & D^\star & \\
e_M \arrow[ur] & & & A \arrow[ul] \arrow[ur] & & & e_D \arrow[ul]
\end{tikzcd}$$

$$M_i \sim \text{Normal}(\nu_i, \tau)$$
$$\nu_i = \alpha_M + \beta_{AM} A_i$$

```r
m15.2 <- ulam(
    alist(
        # D* model (observed)
        D_obs ~ dnorm( D_true , D_sd ),

        # D model (unobserved)
        vector[N]:D_true ~ dnorm( mu , sigma ),
        mu <- a + bA*A + bM*M_true[i],
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        bM ~ dnorm(0,0.5),
        sigma ~ dexp( 1 ),

        # M* model (observed)
        M_obs ~ dnorm( M_true , M_sd ),

        # M model (unobserved)
        vector[N]:M_true ~ dnorm( nu , tau ),
        nu <- aM + bAM*A,
        aM ~ dnorm(0,0.2),
        bAM ~ dnorm(0,0.5),
        tau ~ dexp( 1 )

    ) , data=dlist2 , chains=4 , cores=4 )
```

$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

$$M_i^\star \sim \text{Normal}(M_i, T_i)$$

$$M_i \sim \text{Normal}(\nu_i, \tau)$$
$$\nu_i = \alpha_M + \beta_{AM} A_i$$

```r
m15.2 <- ulam(
    alist(
        # D* model (observed)
        D_obs ~ dnorm( D_true , D_sd ),

        # D model (unobserved)
        vector[N]:D_true ~ dnorm( mu , sigma ),
        mu <- a + bA*A + bM*M_true[i],
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        bM ~ dnorm(0,0.5),
        sigma ~ dexp( 1 ),

        # M* model (observed)
        M_obs ~ dnorm( M_true , M_sd ),

        # M model (unobserved)
        vector[N]:M_true ~ dnorm( nu , tau ),
        nu <- aM + bAM*A,
        aM ~ dnorm(0,0.2),
        bAM ~ dnorm(0,0.5),
        tau ~ dexp( 1 )

    ) , data=dlist2 , chains=4 , cores=4 )
```

$$\boxed{D_i^\star \sim \text{Normal}(D_i, S_i)}$$

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

$$M_i^\star \sim \text{Normal}(M_i, T_i)$$

$$M_i \sim \text{Normal}(\nu_i, \tau)$$
$$\nu_i = \alpha_M + \beta_{AM} A_i$$

```
m15.2 <- ulam(
    alist(
        # D* model (observed)
        D_obs ~ dnorm( D_true , D_sd ),

        # D model (unobserved)
        vector[N]:D_true ~ dnorm( mu , sigma ),
        mu <- a + bA*A + bM*M_true[i],
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        bM ~ dnorm(0,0.5),
        sigma ~ dexp( 1 ),

        # M* model (observed)
        M_obs ~ dnorm( M_true , M_sd ),

        # M model (unobserved)
        vector[N]:M_true ~ dnorm( nu , tau ),
        nu <- aM + bAM*A,
        aM ~ dnorm(0,0.2),
        bAM ~ dnorm(0,0.5),
        tau ~ dexp( 1 )

    ) , data=dlist2 , chains=4 , cores=4 )
```

$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

$$\boxed{\begin{aligned} D_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &= \alpha + \beta_A A_i + \beta_M M_i \end{aligned}}$$

$$M_i^\star \sim \text{Normal}(M_i, T_i)$$

$$M_i \sim \text{Normal}(\nu_i, \tau)$$

$$\nu_i = \alpha_M + \beta_{AM} A_i$$

```
m15.2 <- ulam(
    alist(
        # D* model (observed)
        D_obs ~ dnorm( D_true , D_sd ),

        # D model (unobserved)
        vector[N]:D_true ~ dnorm( mu , sigma ),
        mu <- a + bA*A + bM*M_true[i],
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        bM ~ dnorm(0,0.5),
        sigma ~ dexp( 1 ),

        # M* model (observed)
        M_obs ~ dnorm( M_true , M_sd ),

        # M model (unobserved)
        vector[N]:M_true ~ dnorm( nu , tau ),
        nu <- aM + bAM*A,
        aM ~ dnorm(0,0.2),
        bAM ~ dnorm(0,0.5),
        tau ~ dexp( 1 )

    ) , data=dlist2 , chains=4 , cores=4 )
```

$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

$$\boxed{M_i^\star \sim \text{Normal}(M_i, T_i)}$$

$$M_i \sim \text{Normal}(\nu_i, \tau)$$
$$\nu_i = \alpha_M + \beta_{AM} A_i$$

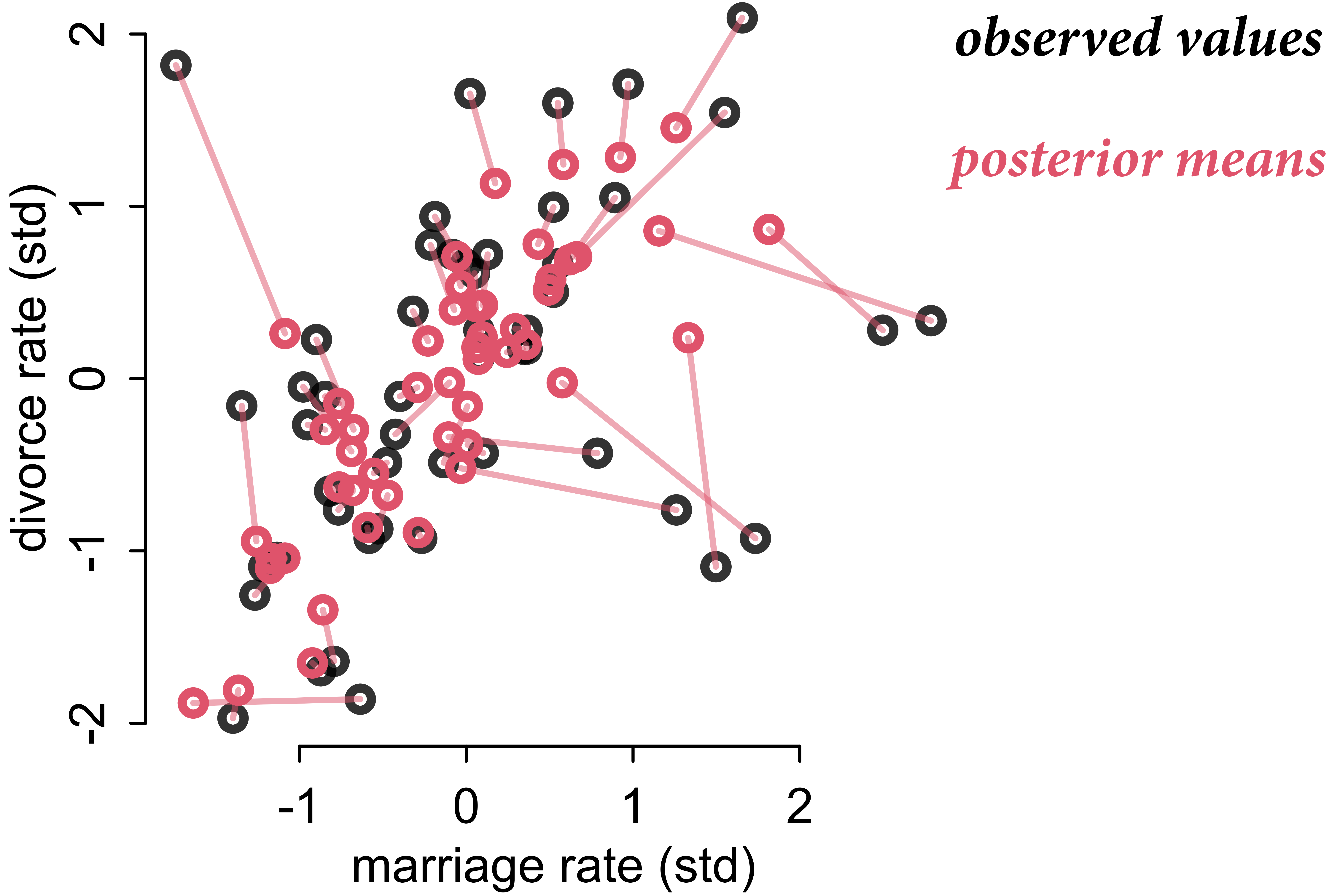
```
m15.2 <- ulam(
    alist(
        # D* model (observed)
        D_obs ~ dnorm( D_true , D_sd ),

        # D model (unobserved)
        vector[N]:D_true ~ dnorm( mu , sigma ),
        mu <- a + bA*A + bM*M_true[i],
        a ~ dnorm(0,0.2),
        bA ~ dnorm(0,0.5),
        bM ~ dnorm(0,0.5),
        sigma ~ dexp( 1 ),

        # M* model (observed)
        M_obs ~ dnorm( M_true , M_sd ),

        # M model (unobserved)
        vector[N]:M_true ~ dnorm( nu , tau ),
        nu <- aM + bAM*A,
        aM ~ dnorm(0,0.2),
        bAM ~ dnorm(0,0.5),
        tau ~ dexp( 1 )

    ) , data=dlist2 , chains=4 , cores=4 )
```

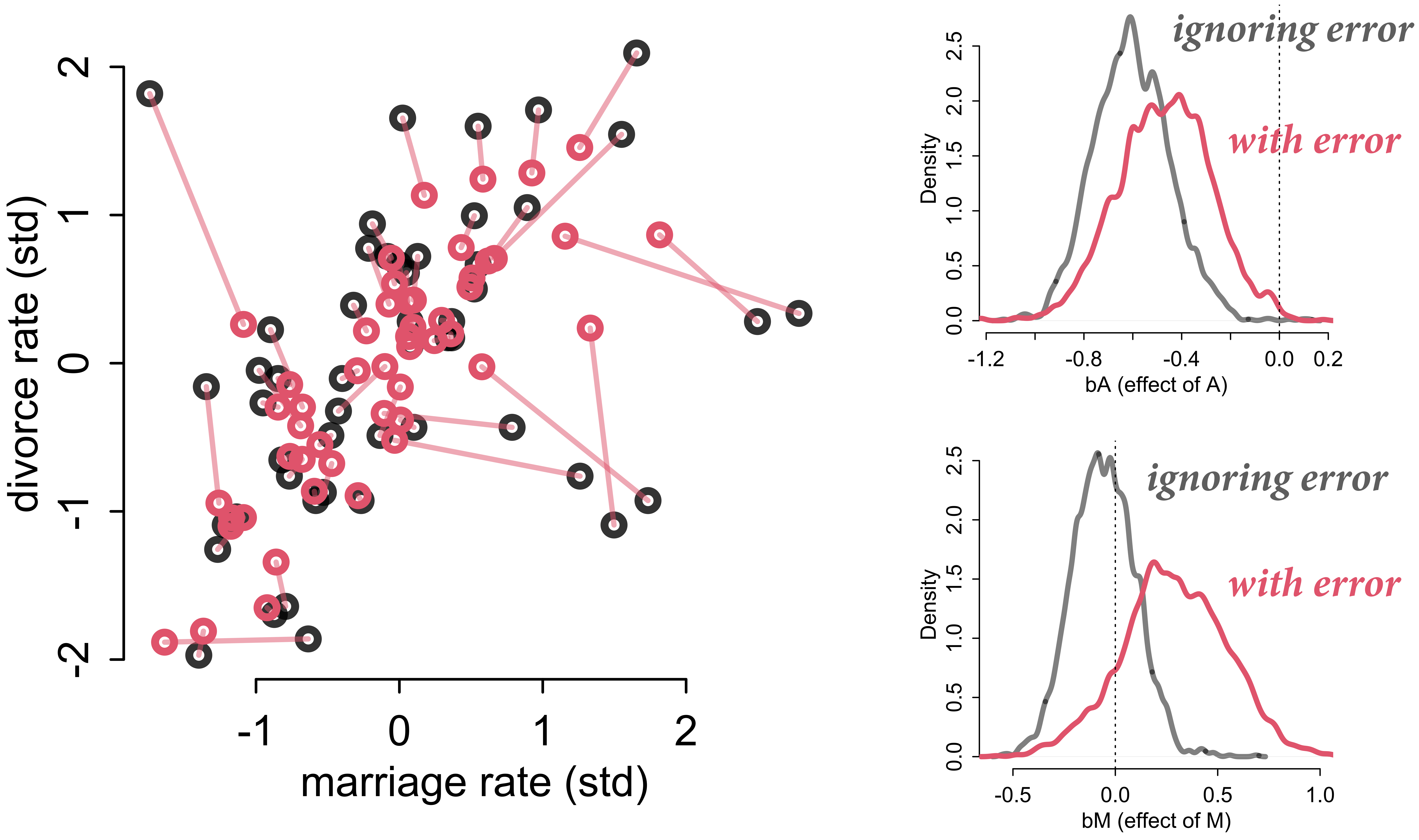
$$D_i^\star \sim \text{Normal}(D_i, S_i)$$

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + \beta_A A_i + \beta_M M_i$$

$$M_i^\star \sim \text{Normal}(M_i, T_i)$$

$$M_i \sim \text{Normal}(\nu_i, \tau)$$
$$\nu_i = \alpha_M + \beta_{AM} A_i$$


observed values
posterior means
divorce rate (std)
marriage rate (std)
-2
-1
0
1
2
-1
0
1
2



divorce rate (std)
marriage rate (std)
-2
-1
0
1
2
-1
0
1
2
ignoring error
with error
Density
bA (effect of A)
-1.2
-0.8
-0.4
0.0
0.2
0.0
0.5
1.0
1.5
2.0
2.5
ignoring error
with error
Density
bM (effect of M)
-0.5
0.0
0.5
1.0

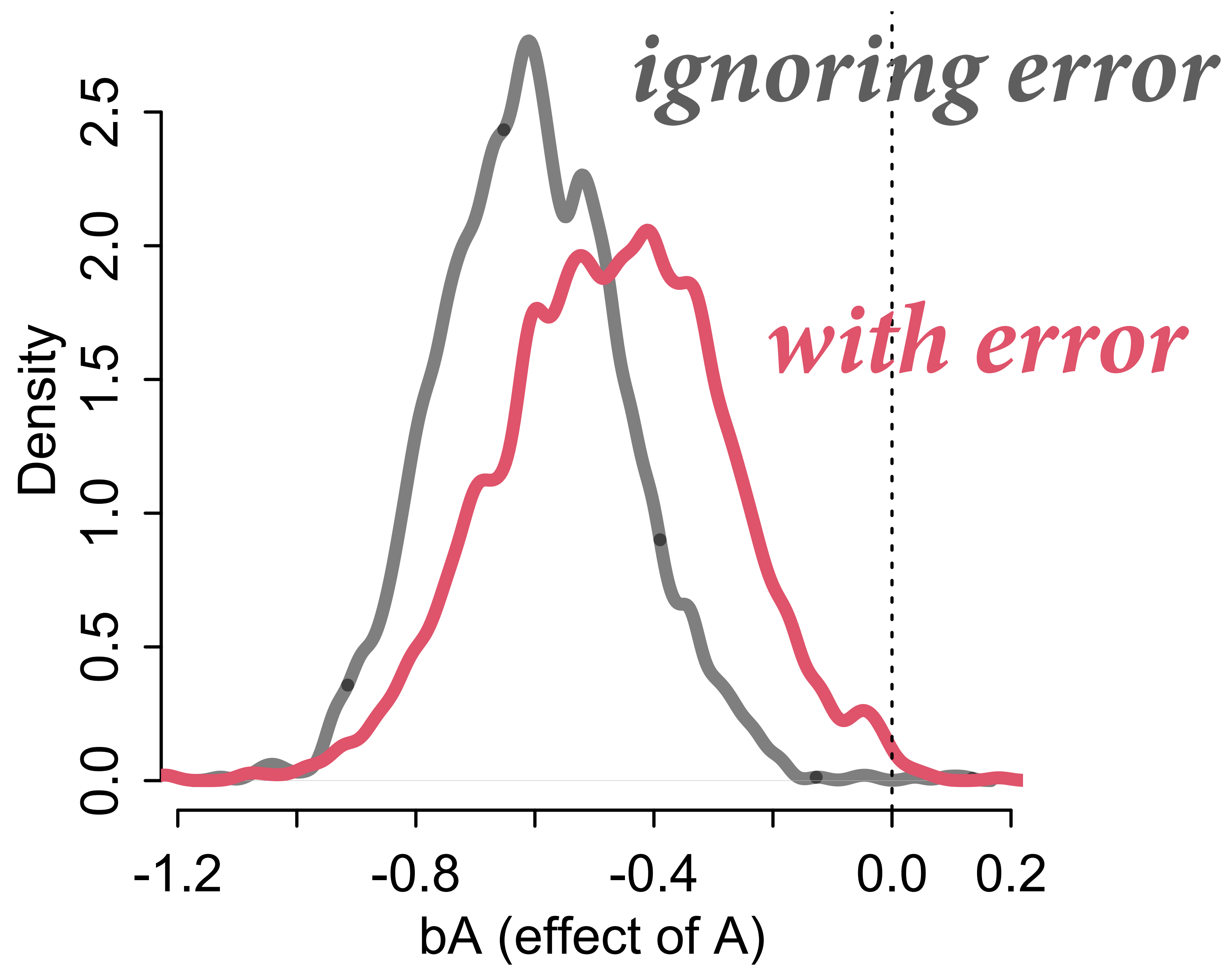
Unpredictable errors

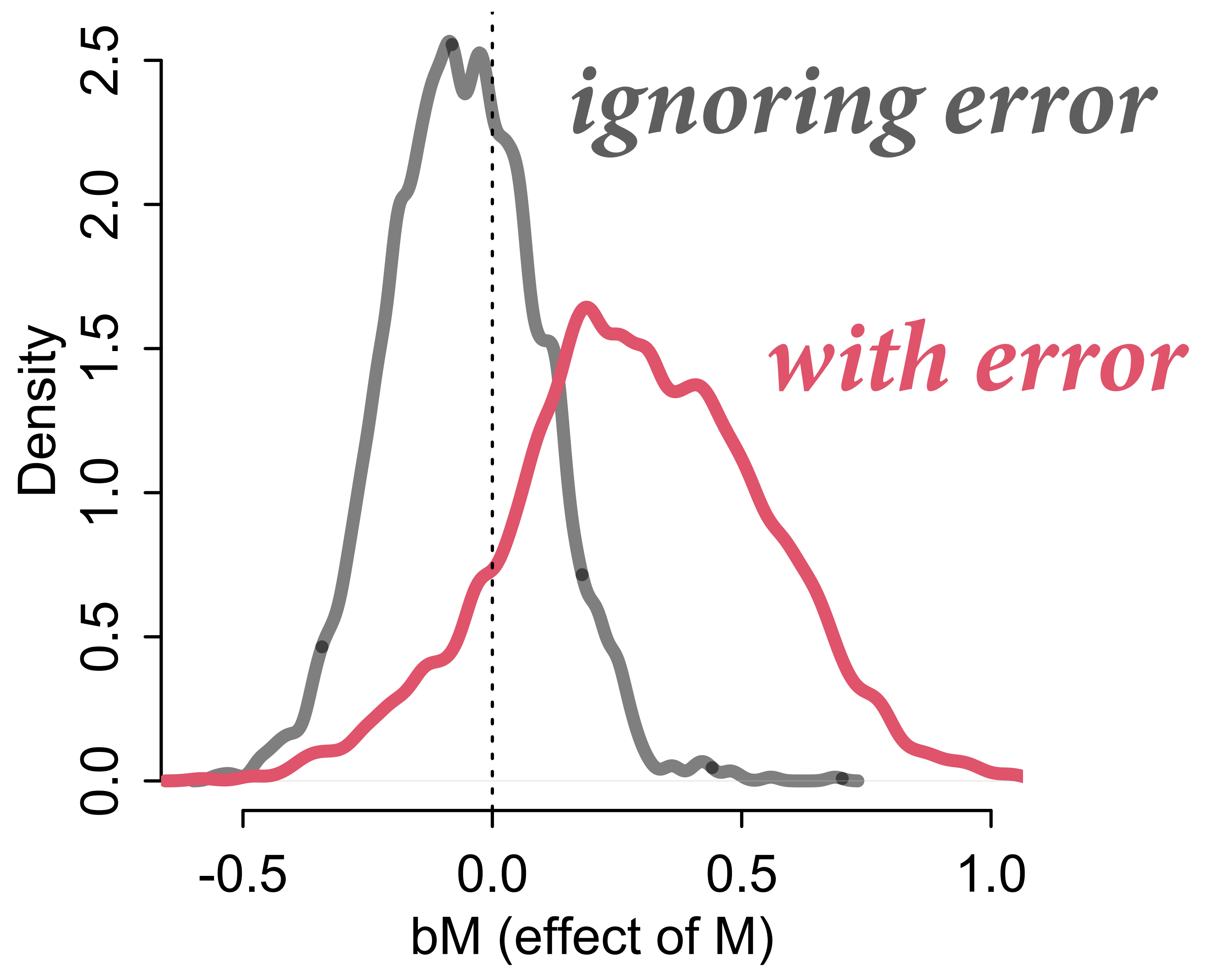
Including error on M increases evidence for an effect of M on D

Why?

Down-weighting of unreliable estimates

Errors can hurt or help, but only honest option is to attend to them





PAUSE

Bantu pastoralists

50k people, 50k km^2

Bilineal descent

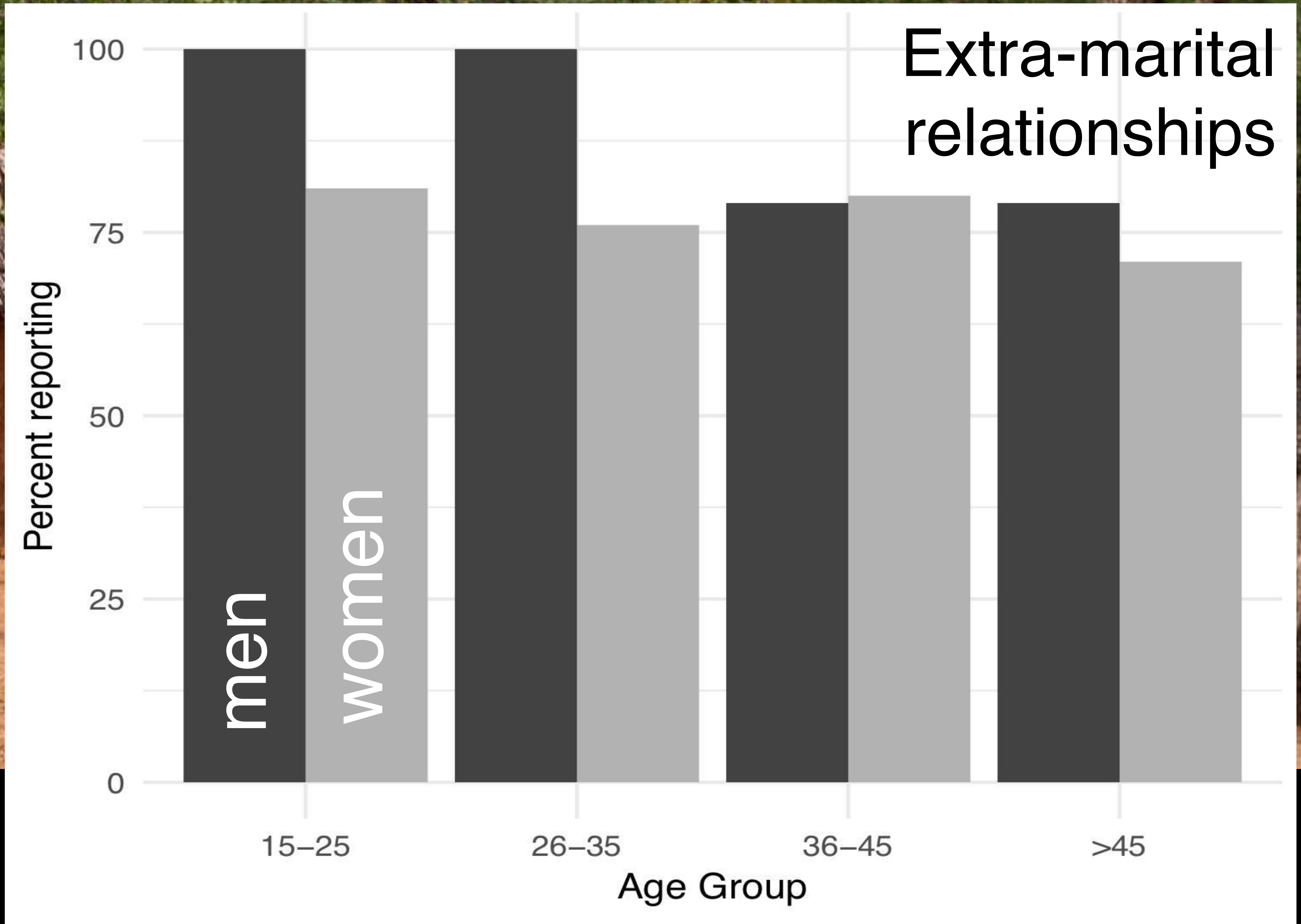
Matrilineal inheritance

Patrilineal status

Patrilocality

Polygyny common

"Open" marriages



Misclassification

Estimand: Proportion of children fathered by extra-pair men

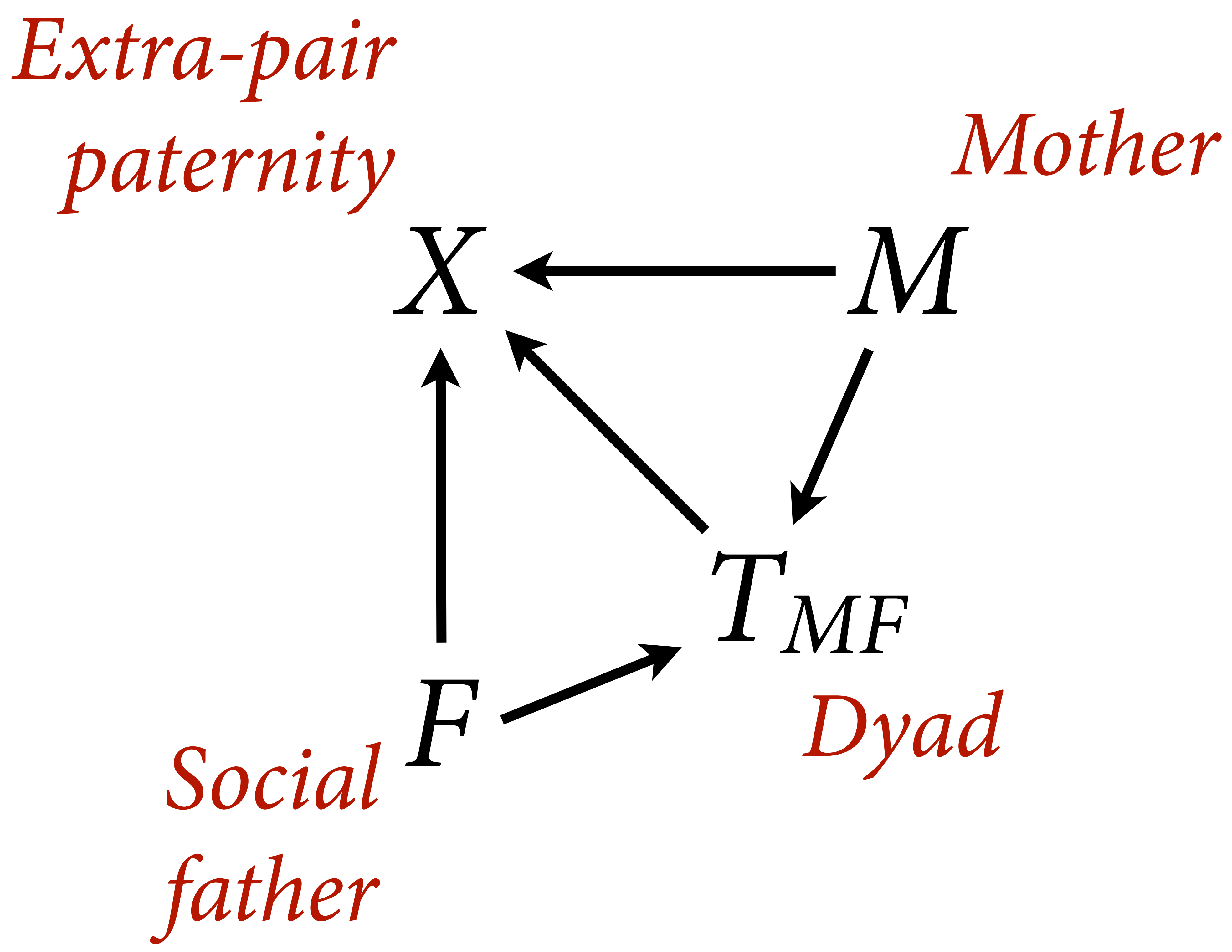
Problem: 5% false-positive rate

Misclassification: Categorical version of measurement error

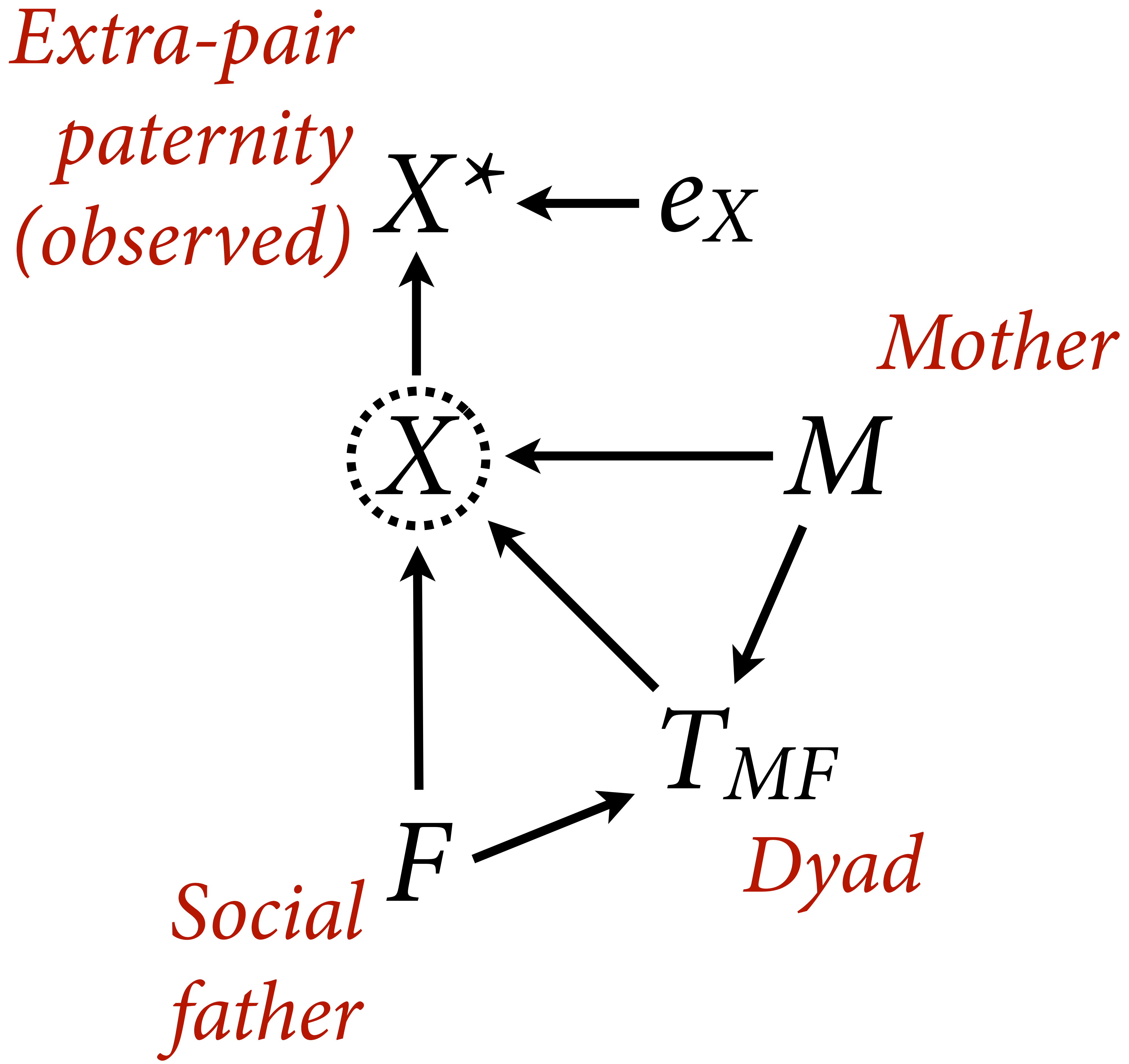
How do we include misclassification?

When my colleagues ask me for help with misclassification

Scelza et al 2020 High rate of extrapair paternity in a human population…



Extra-pair paternity
X
Mother
M
T_{MF}
Dyad
F
Social father


Extra-pair paternity (observed)
X*
e_X
Mother
X
M
T_MF
F
Dyad
Social father

Generative model

$$X_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha + \underbrace{\mu_{M[i]}}_{\textit{Mom}} + \underbrace{\delta_{D[i]}}_{\textit{Dyad}}$$

Generative model

$$X_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha + \underbrace{\mu_{M[i]}}_{\text{Mom}} + \underbrace{\delta_{D[i]}}_{\text{Dyad}}$$

Measurement model

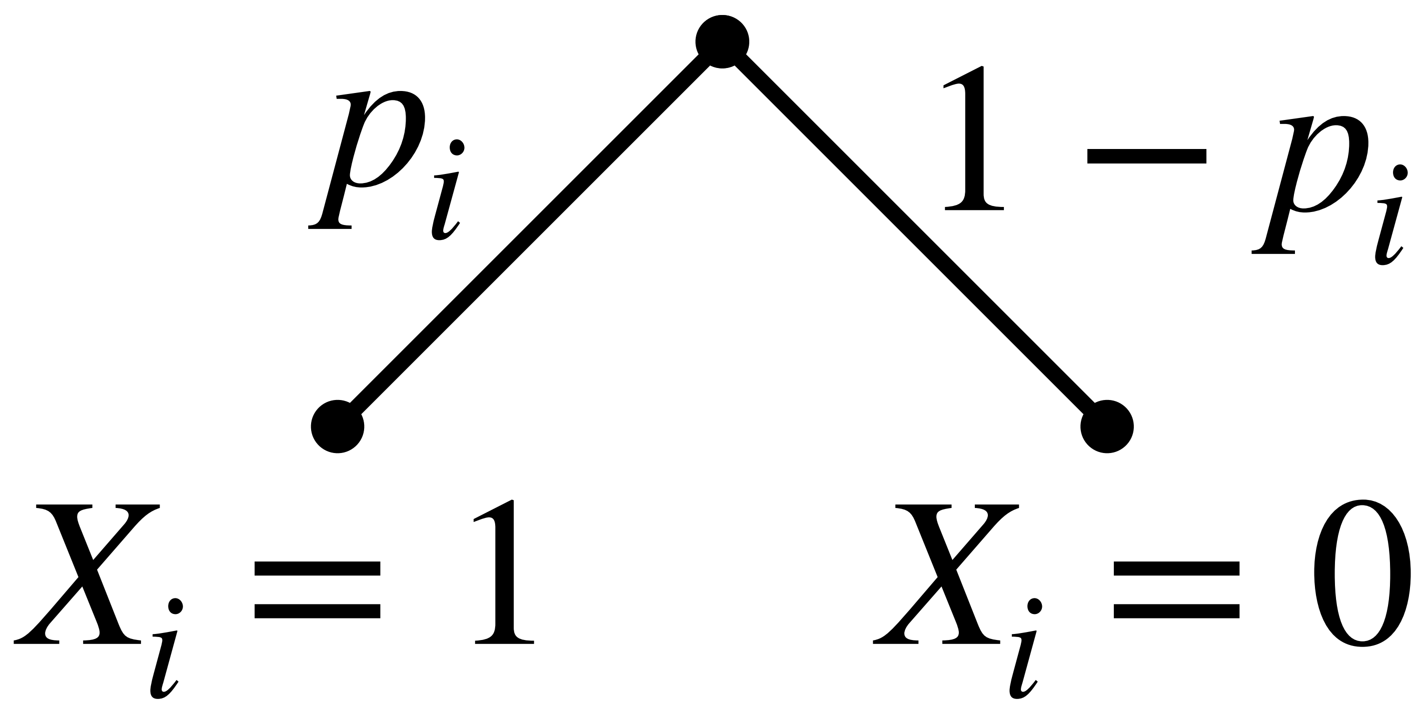
$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \mid p_i) = (1 - p_i)(1 - f)$$

Measurement model

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

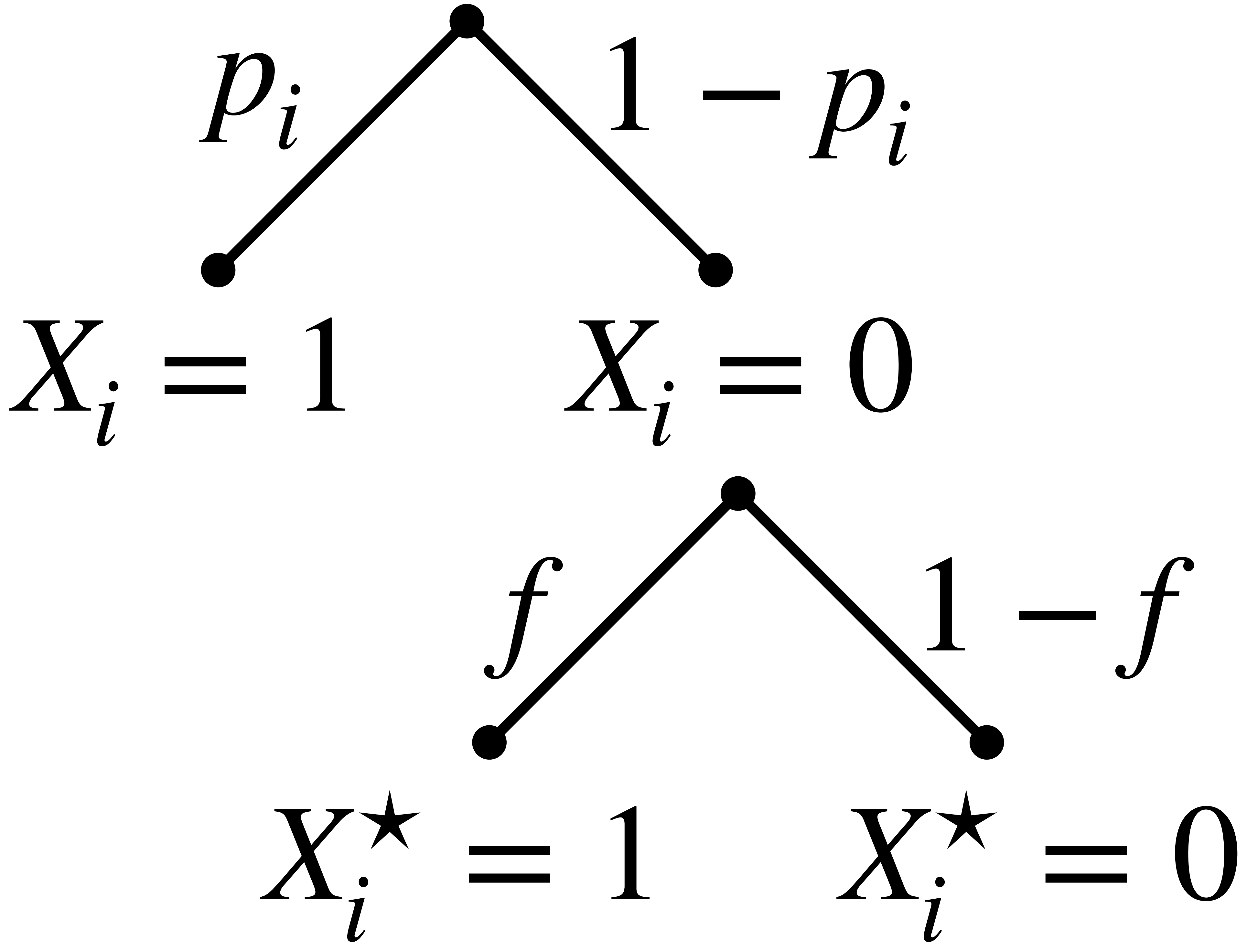
$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$



Measurement model

$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

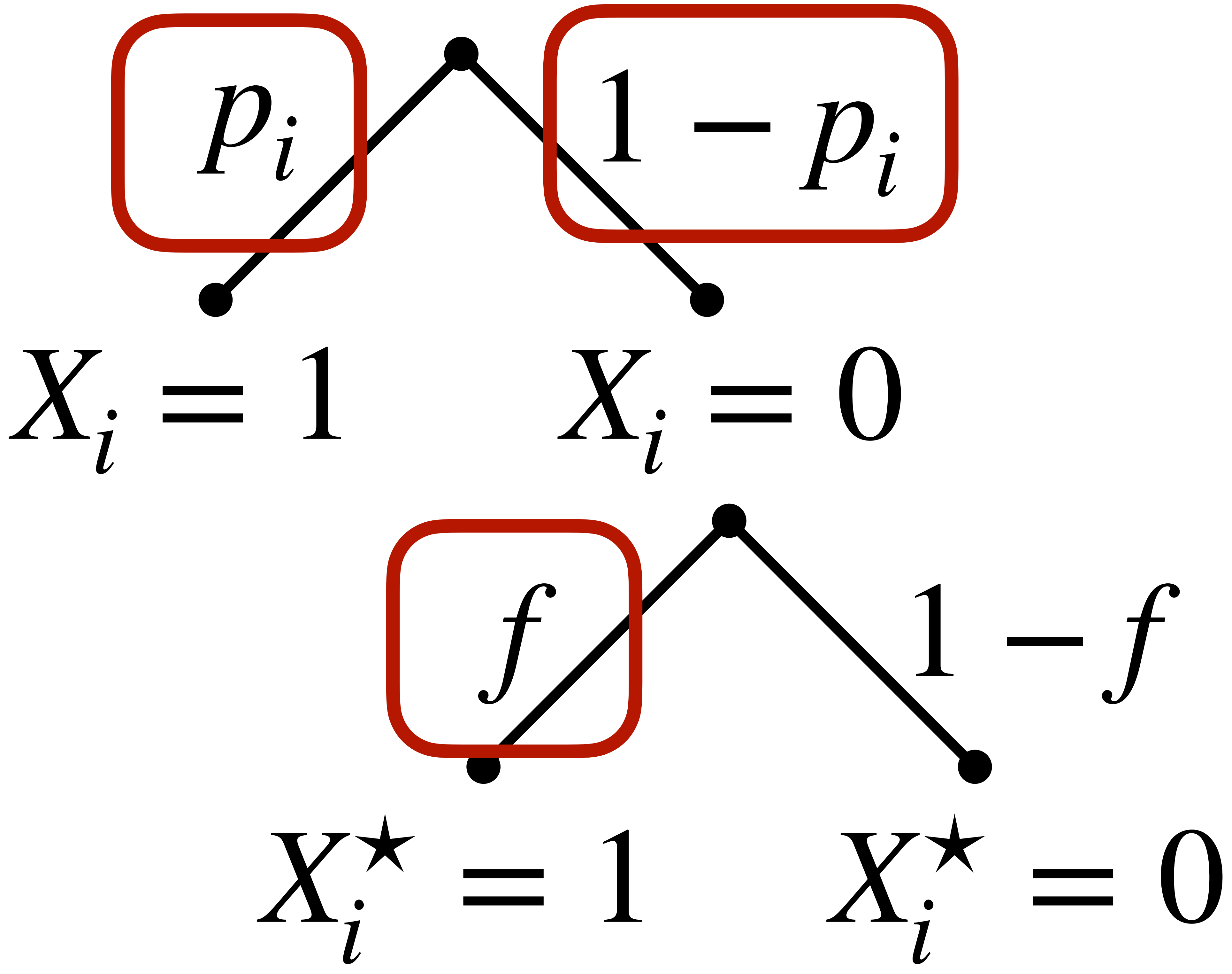
$$\Pr(X_i^\star = 0 \mid p_i) = (1 - p_i)(1 - f)$$



Measurement model

$$\Pr(X_i^\star = 1 \mid p_i) = \boxed{p_i + (1 - p_i)f}$$

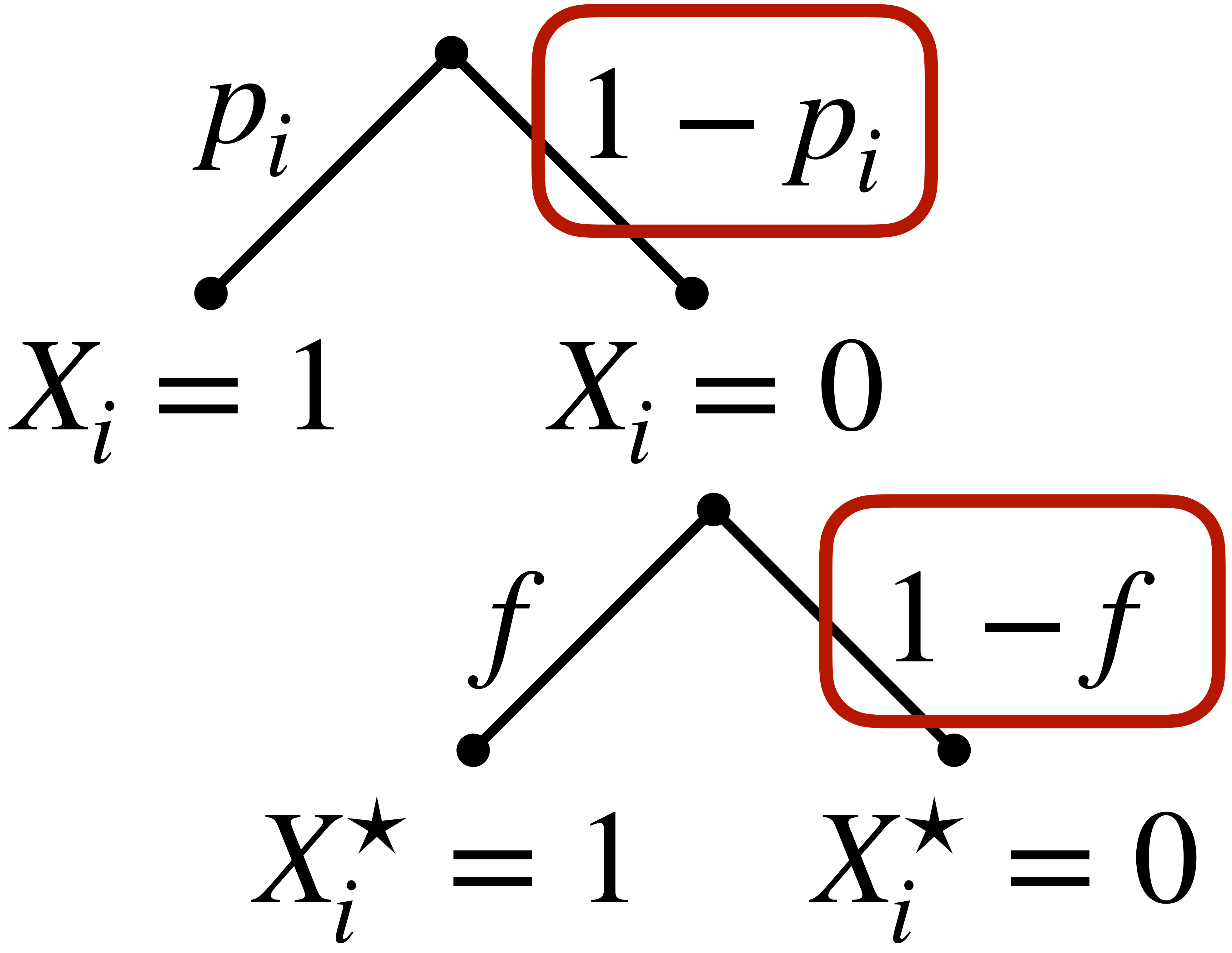
$$\Pr(X_i^\star = 0 \mid p_i) = (1 - p_i)(1 - f)$$



Measurement model

$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \mid p_i) = \boxed{(1 - p_i)(1 - f)}$$



```r
# with false positive rate
dat$f <- 0.05
mX <- ulam(
    alist(
        X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        logit(p) <- a + z[mom_id]*sigma + x[dyad_id]*tau,
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$X_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha + \mu_{M[i]} + \delta_{D[i]}$$

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

```
# with false positive rate
dat$f <- 0.05
mX <- ulam(
    alist(
        X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        logit(p) <- a + z[mom_id]*sigma + x[dyad_id]*tau,
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$X_i \sim \text{Bernoulli}(p_i)$$

$$\text{logit}(p_i) = \alpha + \mu_{M[i]} + \delta_{D[i]}$$

$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \mid p_i) = (1 - p_i)(1 - f)$$

Floating Point Monsters

Probability calculations tend to underflow (round to zero) and overflow (round to one)

Solution: Calculate on log scale

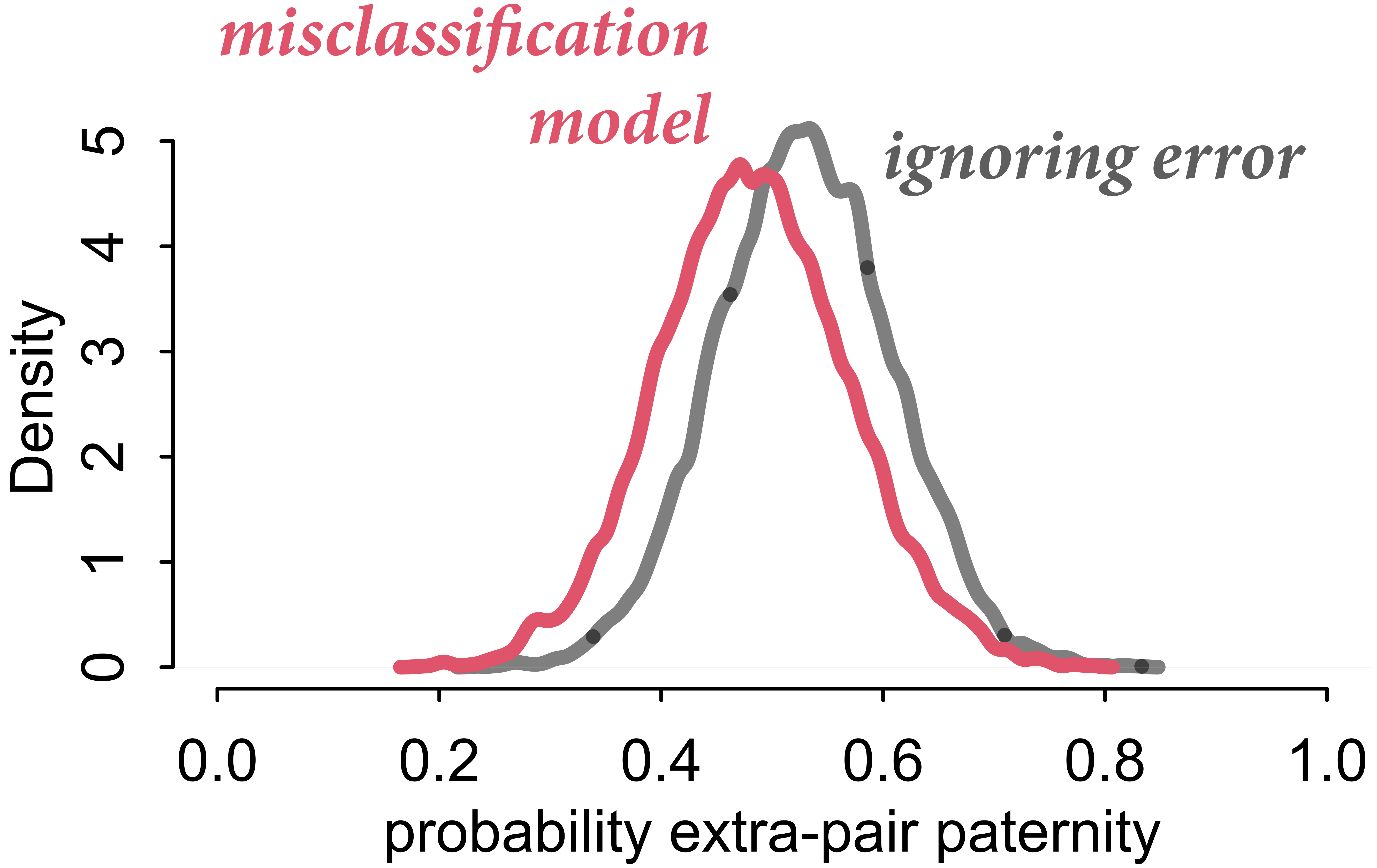
Ancient weapons:

`log_sum_exp`, `log1m`, `log1m_exp`

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

```
# numerically stable version of model - see BONUS for explanation
mX3 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log_p , log1m_exp(log_p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m_exp(log_p) + log1m(f) ),

        log_p <- log_inv_logit( a + z[mom_id]*sigma + x[dyad_id]*tau ),
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```



$$X_i \sim \text{Bernoulli}(p_i)$$
$$\text{logit}(p_i) = \alpha + \mu_{M[i]} + \delta_{D[i]}$$

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$
$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

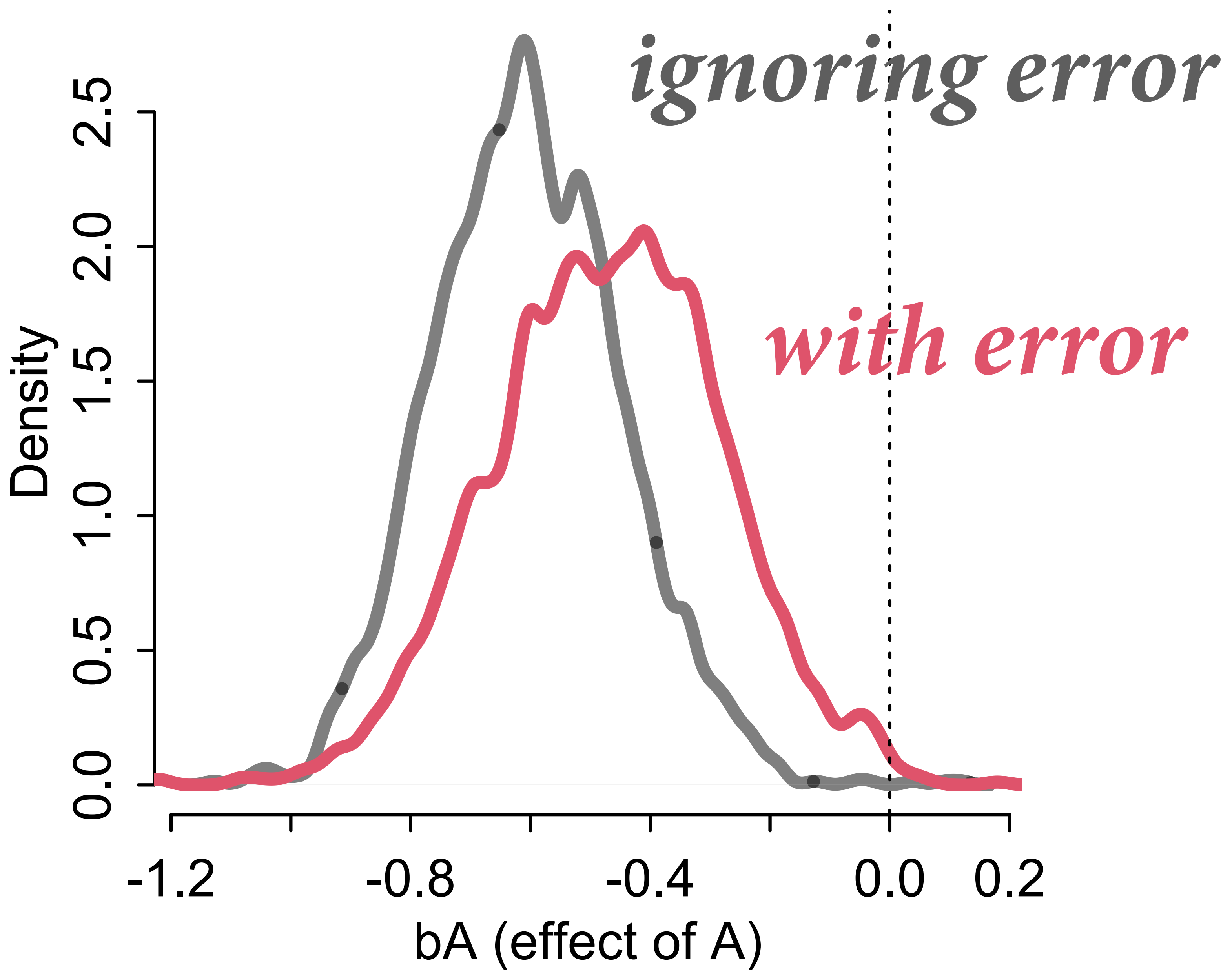
Measurement Horizons

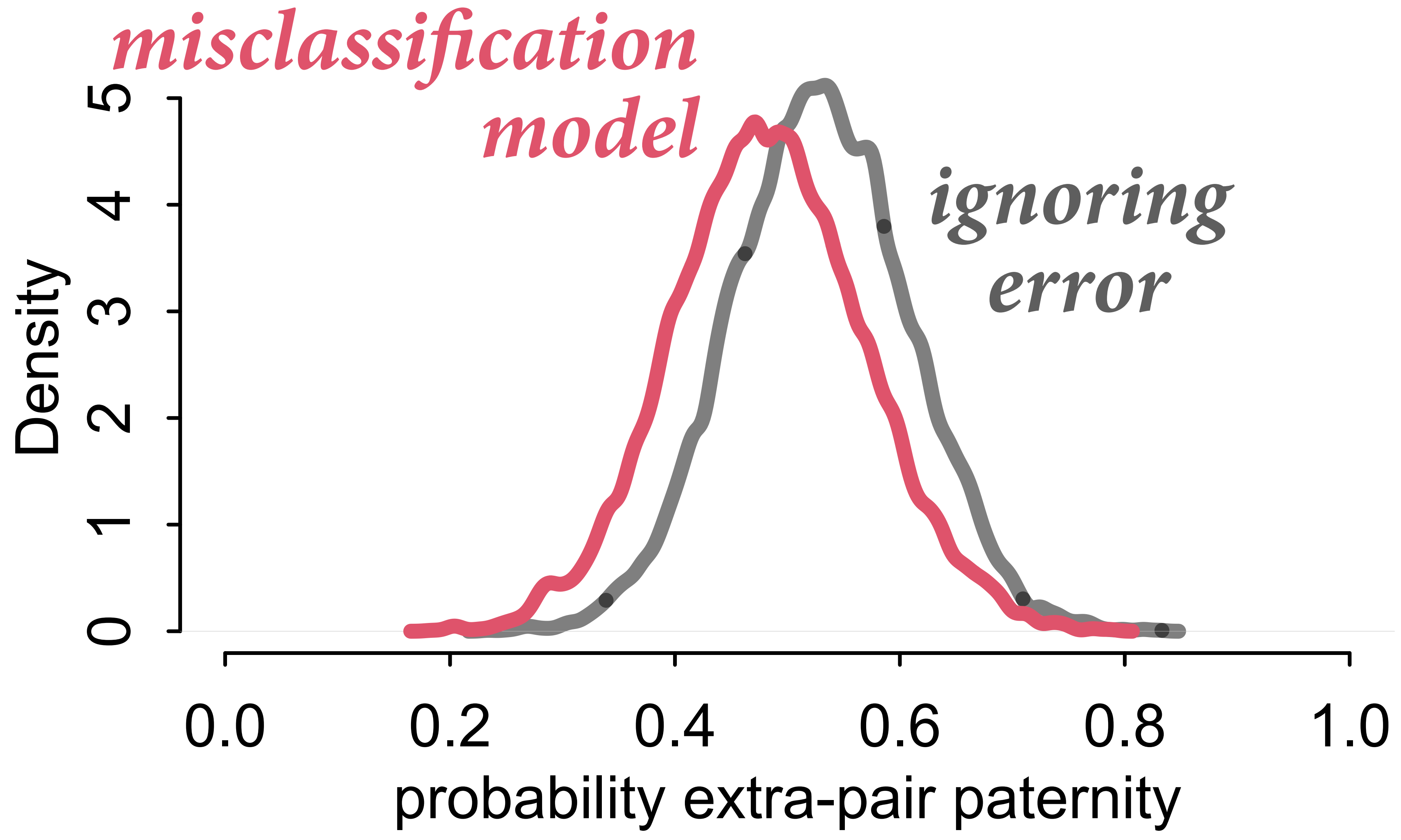
Plenty of related problems and solutions

Rating and assessment: Judges and tests are noisy — item response theory and factor analysis

Hurdle models: Thresholds for detection

Occupancy models: Not detecting something doesn't mean it isn't there





Course Schedule

Week 1	Bayesian inference	Chapters 1, 2, 3
Week 2	Linear models & Causal Inference	Chapter 4
Week 3	Causes, Confounds & Colliders	Chapters 5 & 6
Week 4	Overfitting / MCMC	Chapters 7, 8, 9
Week 5	Generalized Linear Models	Chapters 10, 11
Week 6	Ordered categories & Multilevel models	Chapters 12 & 13
Week 7	More Multilevel models	Chapters 13 & 14
Week 8	Social Networks & Gaussian Processes	Chapter 14
Week 9	Measurement & Missingness	Chapter 15
Week 10	Generalized Linear Madness	Chapter 16

https://github.com/rmcelreath/stat_rethinking_2023

BONUS

Floating Point Monsters

Probability calculations tend to underflow (round to zero) and overflow (round to one)

Solution: Calculate on log scale

Ancient weapons:

`log_sum_exp`, `log1m`, `log1m_exp`

Logarithms make sense

$$\Pr(X_i^\star = 0 \mid p_i) = (1 - p_i)(1 - f)$$

Logarithms make sense

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

$$\log \Pr(X_i^\star = 0 \,|\, p_i) = \log[(1 - p_i)(1 - f)]$$

Logarithms make sense

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

$$\begin{aligned}\log \Pr(X_i^\star = 0 \,|\, p_i) &= \log[(1 - p_i)(1 - f)] \\ &= \log(1 - p_i) + \log(1 - f)\end{aligned}$$

Devil in the details

$$\log \Pr(X_i^\star = 0 \mid p_i) = \log(1 - p_i) + \log(1 - f)$$

If p_i is close to zero, log(1–p_i) could evaluate to zero

Devil in the details

$$\log \Pr(X_i^\star = 0 \,|\, p_i) = \log(1 - p_i) + \log(1 - f)$$

If p_i is close to zero, log(1–p_i) could evaluate to zero

```
> log( 1 - 0.01 )
[1] -0.01005034
> log( 1 - 1e-10 )
[1] -1e-10
> log( 1 - 1e-90 )
[1] 0
```

Devil in the details

$$\log \Pr(X_i^\star = 0 \,|\, p_i) = \log(1 - p_i) + \log(1 - f)$$

If p_i is close to zero, log(1–p_i) could evaluate to zero

```
> log( 1 - 0.01 )
[1] -0.01005034
> log( 1 - 1e-10 )
[1] -1e-10
> log( 1 - 1e-90 )
[1] 0
```

```
> log1p( -1e-90 )
[1] -1e-90
```

$$\mathrm{log1p}(-x) = \mathrm{log1m}(x)$$

Devil in the details

Okay, but what is log1p(x)? And how do we program it?

When x > 1e–4, no reason to worry

When x < 1e–4, use a Taylor series approximation:

$$\log(1+x) = \boxed{x - \frac{x^2}{2}} + \frac{x^3}{3} + \ldots$$

second-order approximation, for x near zero

log_sum_exp

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

$$\log \Pr(X_i^\star = 1 \,|\, p_i) = \log[p_i + (1 - p_i)f]$$

log_sum_exp

$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

$$\log \Pr(X_i^\star = 1 \mid p_i) = \log[p_i + (1 - p_i)f]$$

$$= \text{log\_sum\_exp}(\log p_i, \log(1 - p_i) + \log f)$$

log_sum_exp

$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

$$\log \Pr(X_i^\star = 1 \mid p_i) = \log[p_i + (1 - p_i)f]$$

$$= \text{log\_sum\_exp}(\log p_i, \log(1 - p_i) + \log f)$$

$$= \text{log\_sum\_exp}(\log p_i, \text{log1m}(p_i) + \log f)$$

$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \mid p_i) = (1 - p_i)(1 - f)$$

```
# numerically kosher version
mX2 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log(p) , log1m(p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m(p) + log1m(f) ),

        logit(p) <- a + z[mom_id]*sigma + x[dyad_id]*tau,
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \mid p_i) = (1 - p_i)(1 - f)$$

```
# numerically kosher version
mX2 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log(p) , log1m(p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m(p) + log1m(f) ),

        logit(p) <- a + z[mom_id]*sigma + x[dyad_id]*tau,
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

```
# numerically kosher version
mX2 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log(p) , log1m(p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m(p) + log1m(f) ),

        logit(p) <- a + z[mom_id]*sigma + x[dyad_id]*tau,
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

```
# numerically kosher version
mX2 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log(p) , log1m(p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m(p) + log1m(f) ),

        logit(p) <- a + z[mom_id]*sigma + x[dyad_id]*tau,
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

```
# numerically kosher version
mX2 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log(p) , log1m(p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m(p) + log1m(f) ),

        logit(p) <- a + z[mom_id]*sigma + x[dyad_id]*tau,
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

```
# numerically kosher version
mX2 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log(p) , log1m(p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m(p) + log1m(f) ),

        logit(p) <- a + z[mom_id]*sigma + x[dyad_id]*tau,
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$\Pr(X_i^\star = 1 \,|\, p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \,|\, p_i) = (1 - p_i)(1 - f)$$

```
# double kosher
mX3 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log_p , log1m_exp(log_p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m_exp(log_p) + log1m(f) ),

        log_p <- log_inv_logit( a + z[mom_id]*sigma + x[dyad_id]*tau ),
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

$$\Pr(X_i^\star = 1 \mid p_i) = p_i + (1 - p_i)f$$

$$\Pr(X_i^\star = 0 \mid p_i) = (1 - p_i)(1 - f)$$

```
# double kosher
mX3 <- ulam(
    alist(
        #X|X==1 ~ custom( log( p + (1-p)*f ) ),
        X|X==1 ~ custom( log_sum_exp( log_p , log1m_exp(log_p)+log(f) ) ),

        #X|X==0 ~ custom( log( (1-p)*(1-f) ) ),
        X|X==0 ~ custom( log1m_exp(log_p) + log1m(f) ),

        log_p <- log_inv_logit( a + z[mom_id]*sigma + x[dyad_id]*tau ),
        a ~ normal(0,1.5),
        z[mom_id] ~ normal(0,1),
        sigma ~ normal(0,1),
        x[dyad_id] ~ normal(0,1),
        tau ~ normal(0,1)
    ) , data=dat , chains=4 , cores=4 , iter=4000 ,
    constraints=list(sigma="lower=0",tau="lower=0") )
```

Floating Point Monsters

Ancient weapons:

`log_sum_exp`, `log1m`, `log1m_exp`

Built into most math libraries

Often unnecessary, but when you need them, no better defense

https://mc-stan.org/docs/functions-reference/composed-functions.html