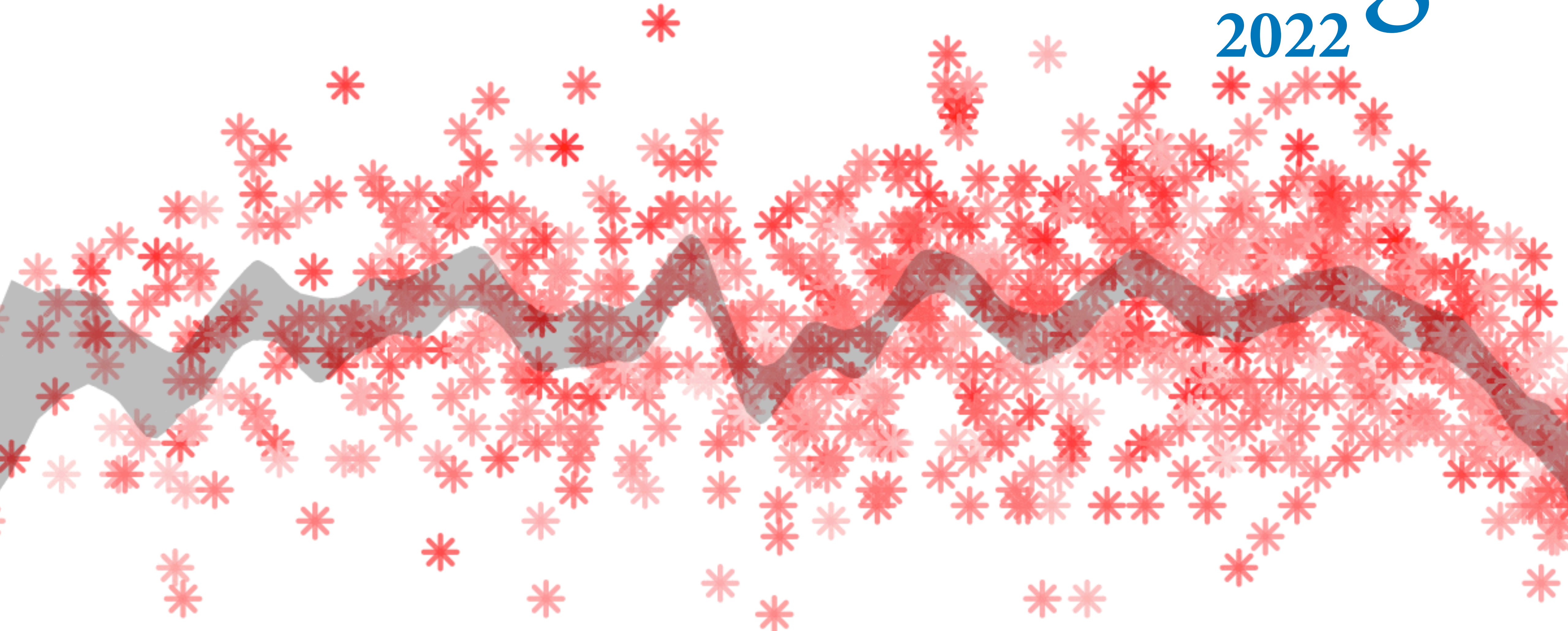


Statistical Rethinking

2022



13: Multi-Multilevel Models

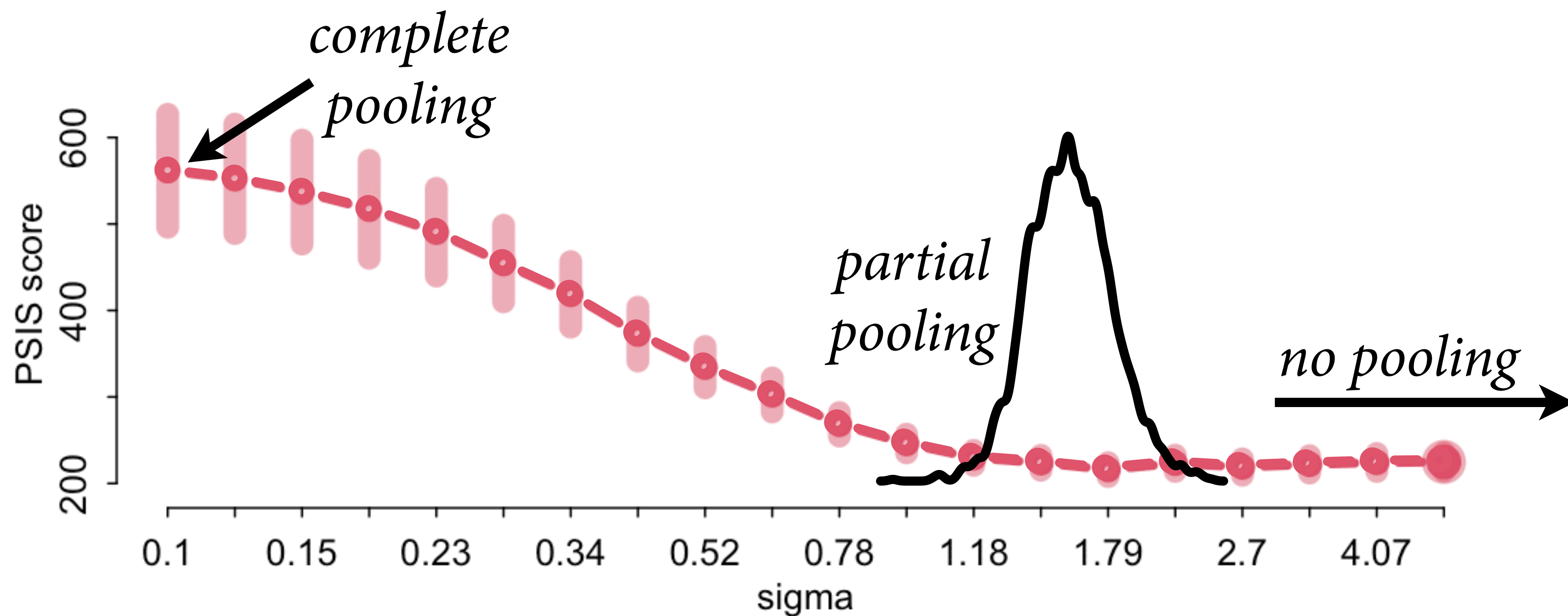
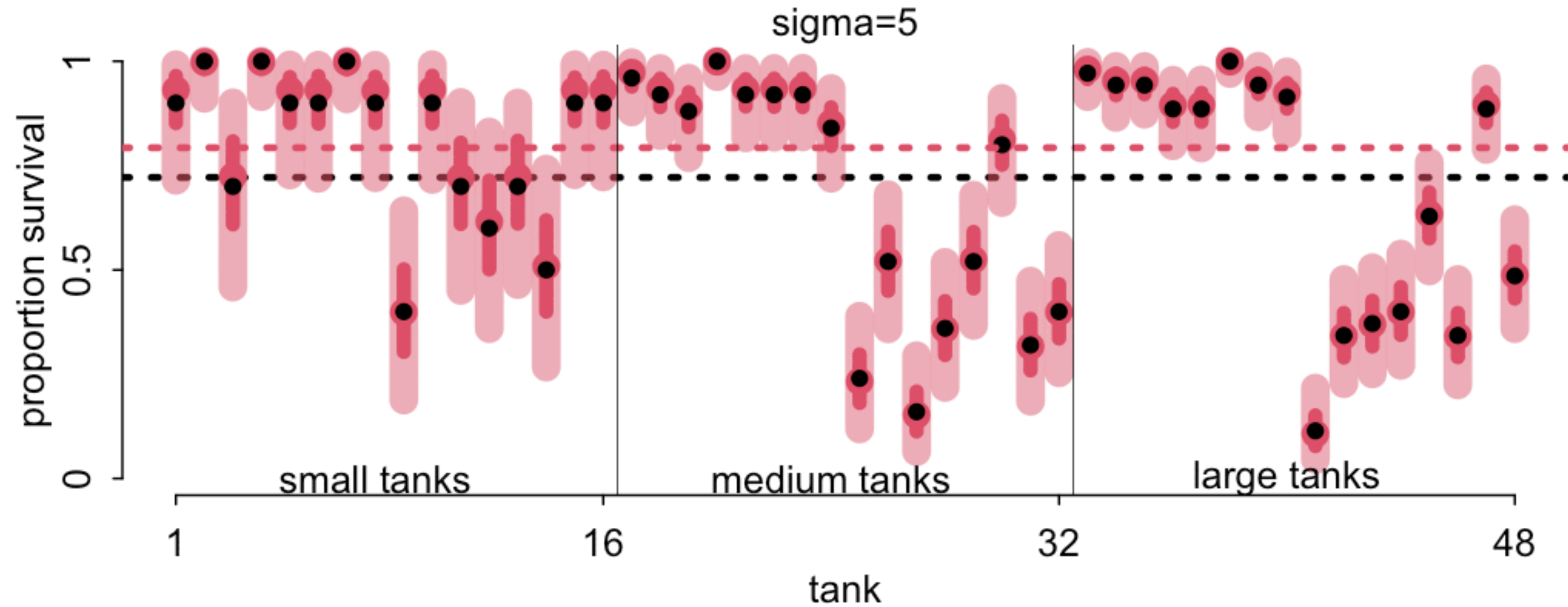


$$S_i \sim \text{Binomial}(D_i, p_i)$$

$$\text{logit}(p_i) = \alpha_{T[i]}$$

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

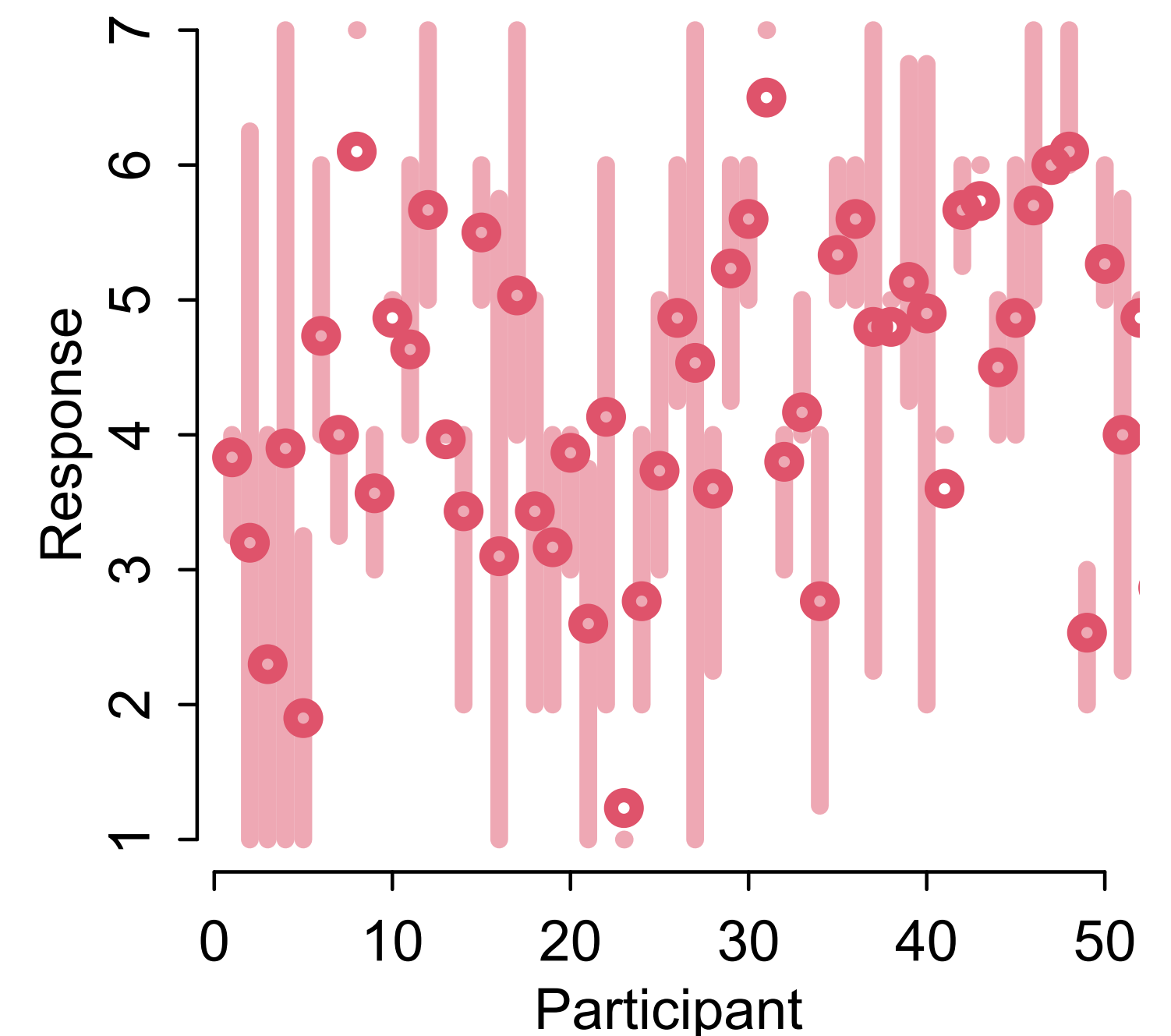
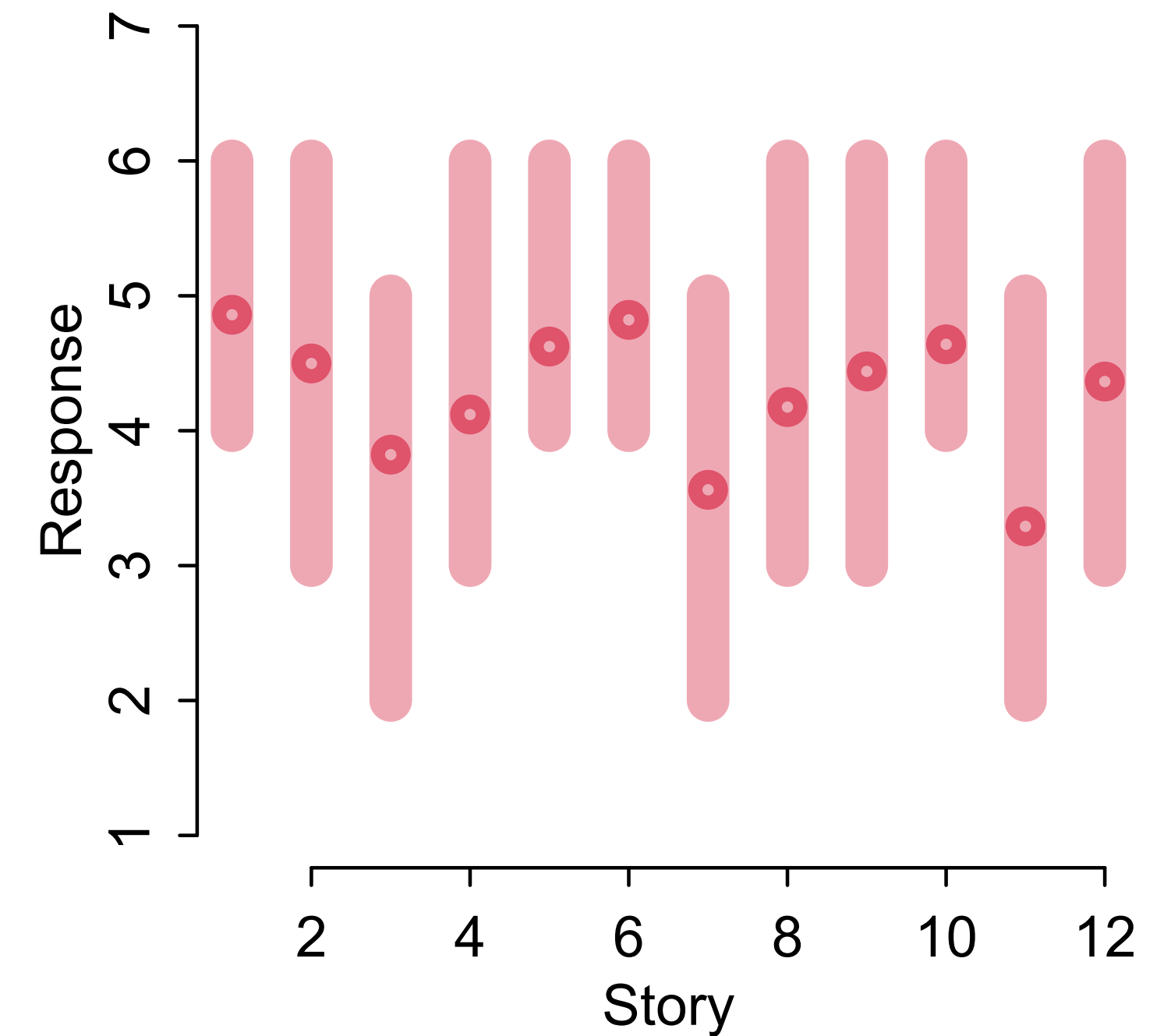
$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$



Practical Difficulties

Varying effects are a good default, but...

- (1) How to use **more than one** cluster type at the same time? For example **stories** and **participants**
- (2) How to calculate predictions
- (3) How to sample chains efficiently

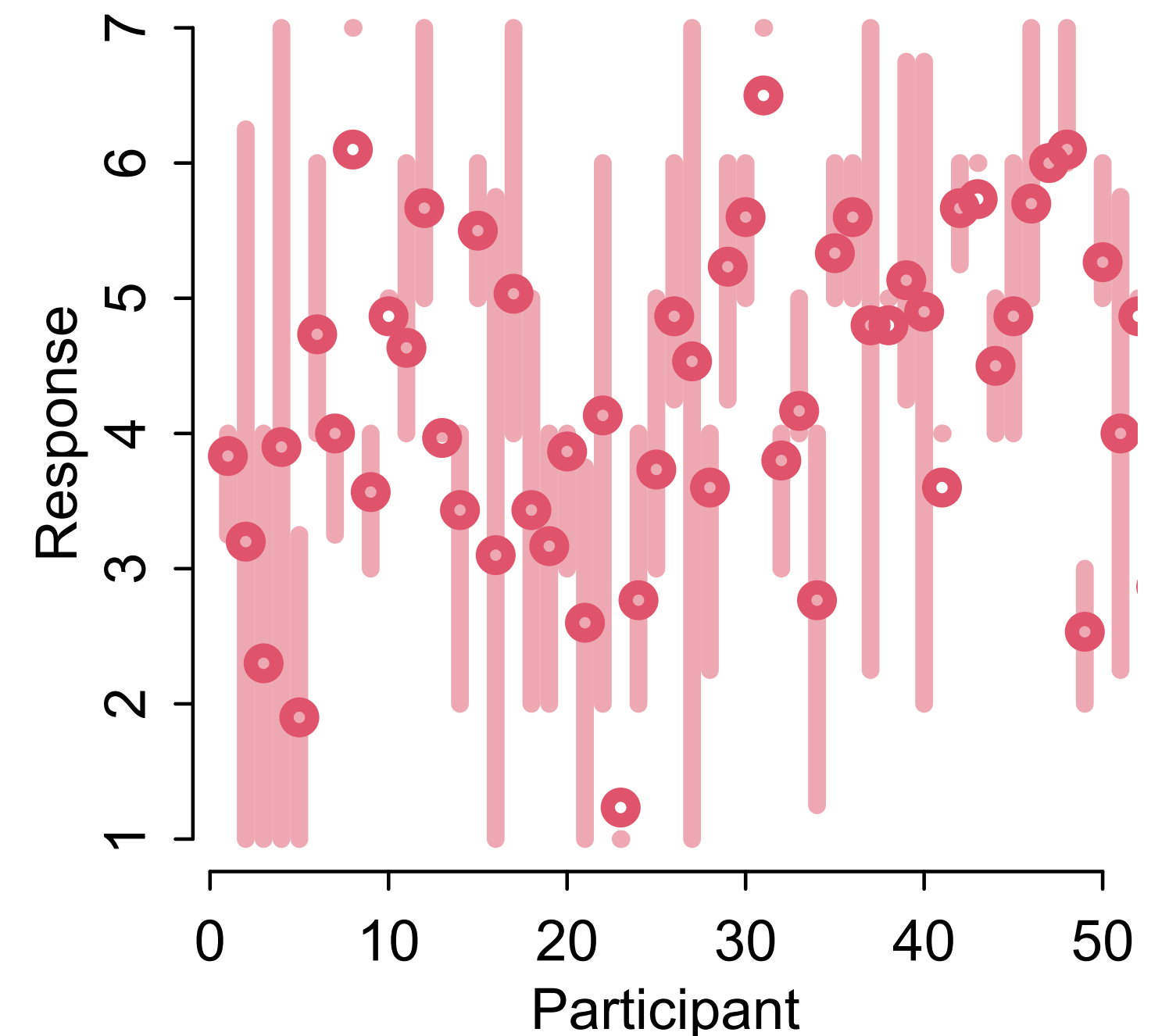
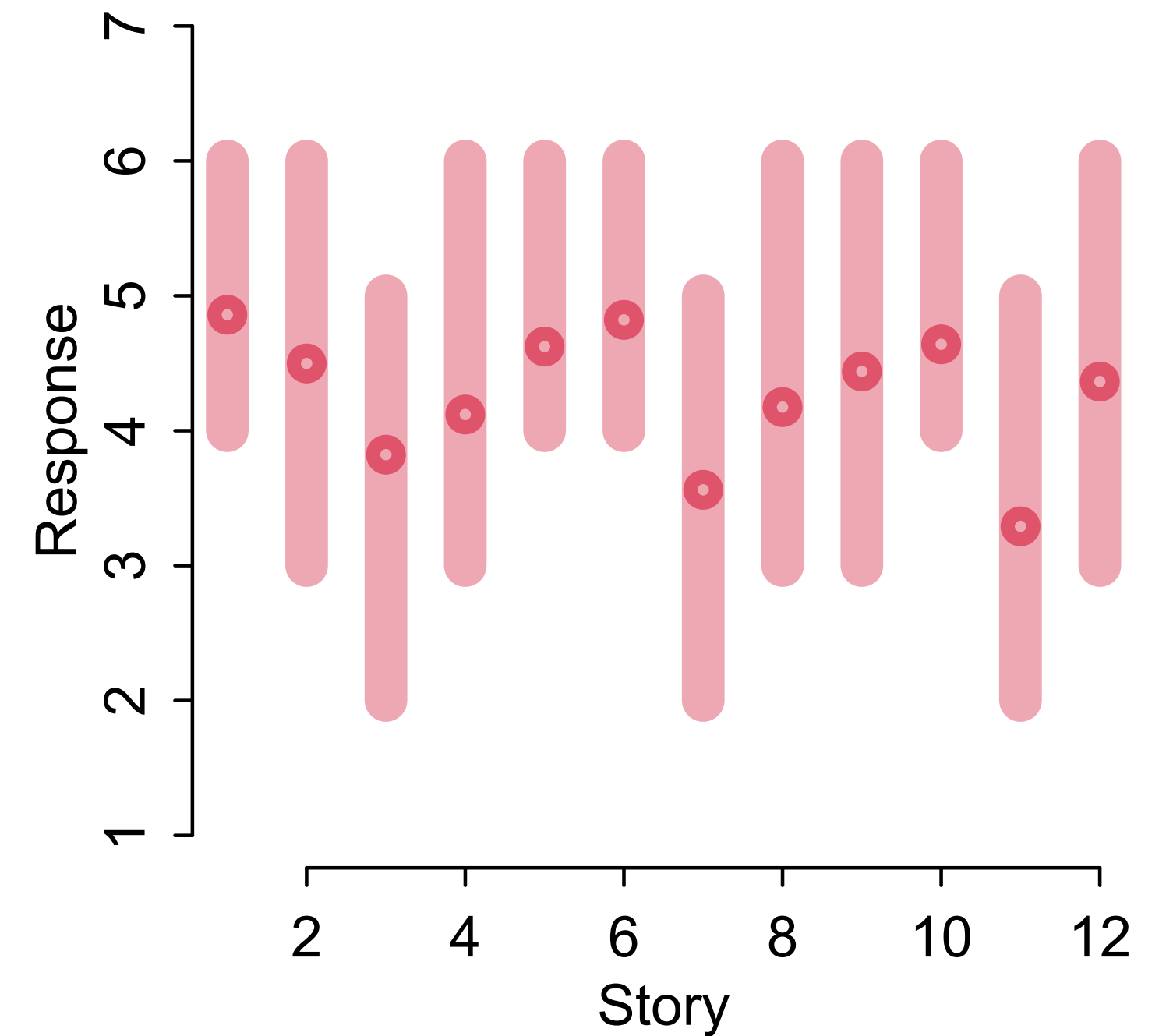


Clusters & features

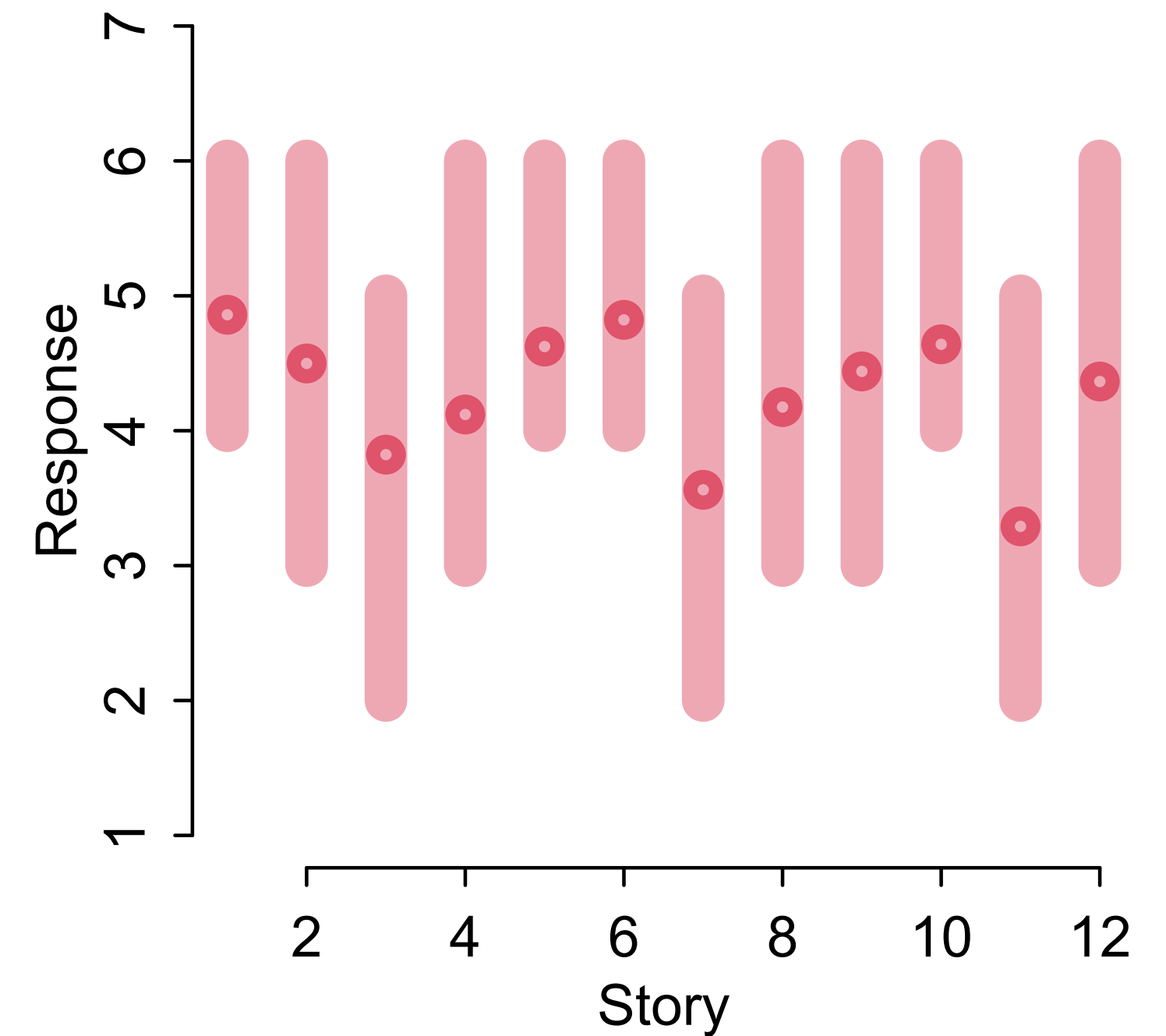
Clusters: Kinds of groups in the data

Features: Aspects of the model (parameters) that vary by cluster

Cluster	Features
tanks	→ survival
stories	→ treatment effect
individuals	→ average response
departments	→ admission rate, bias

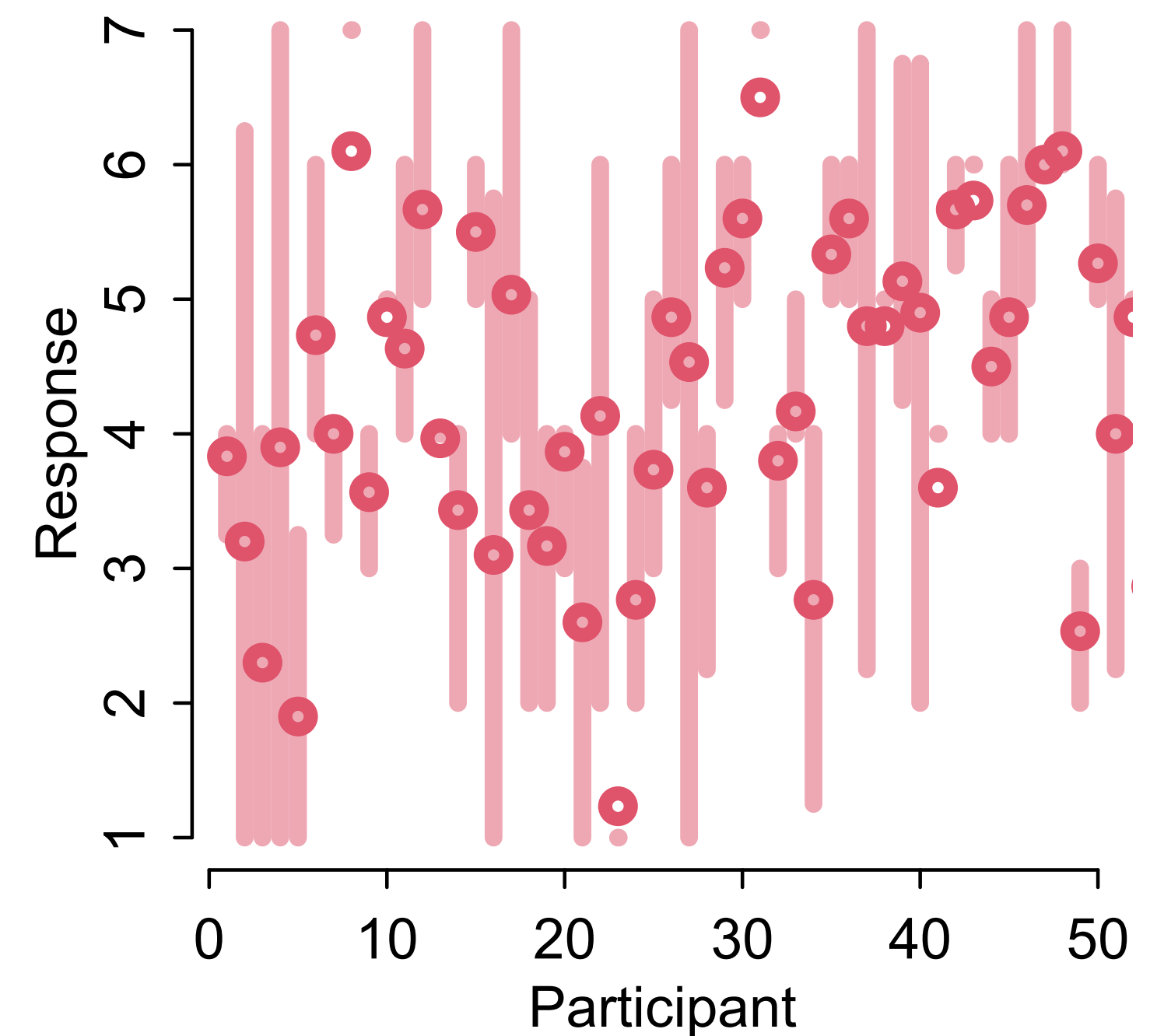


Cluster		Features
tanks	→	survival
stories	→	treatment effect
individuals	→	average response
departments	→	admission rate, bias



Add clusters: More index variables, more population priors (this lecture)

Add features: More parameters, more dimensions *in each* population prior (next lecture)



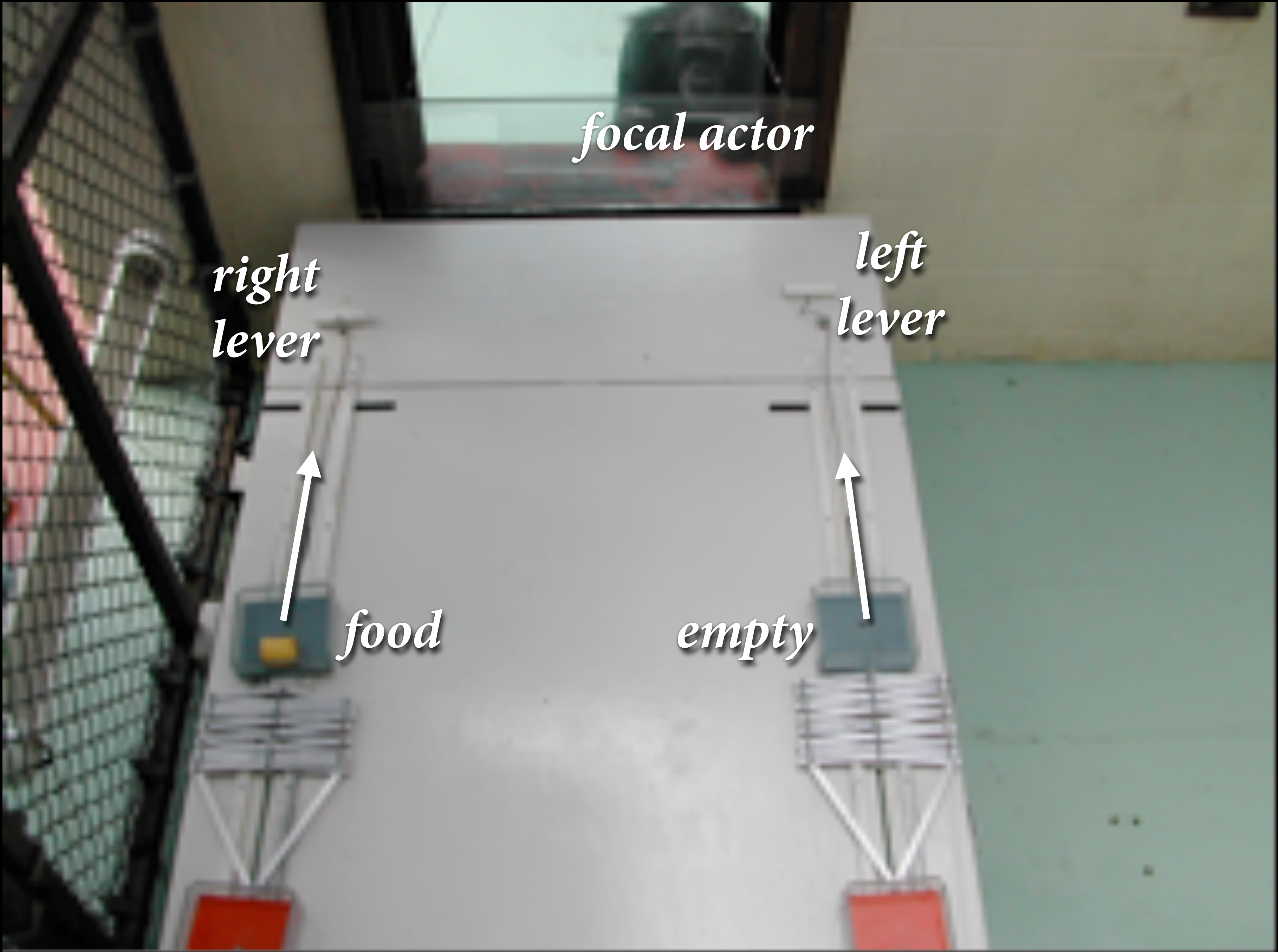
focal actor

*right
lever*

*left
lever*

food

empty



partner

to partner

food

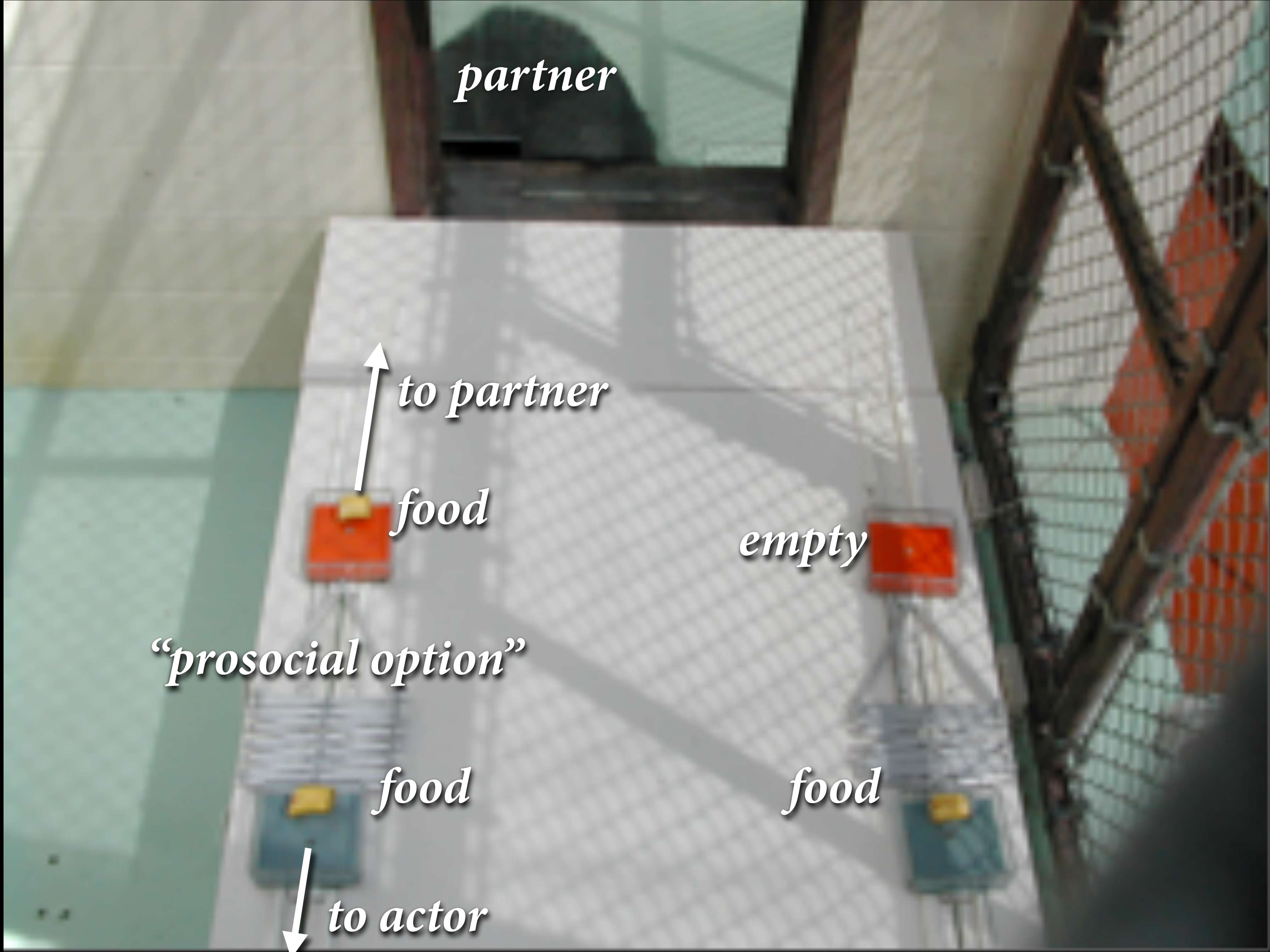
empty

“prosocial option”

food

food

to actor



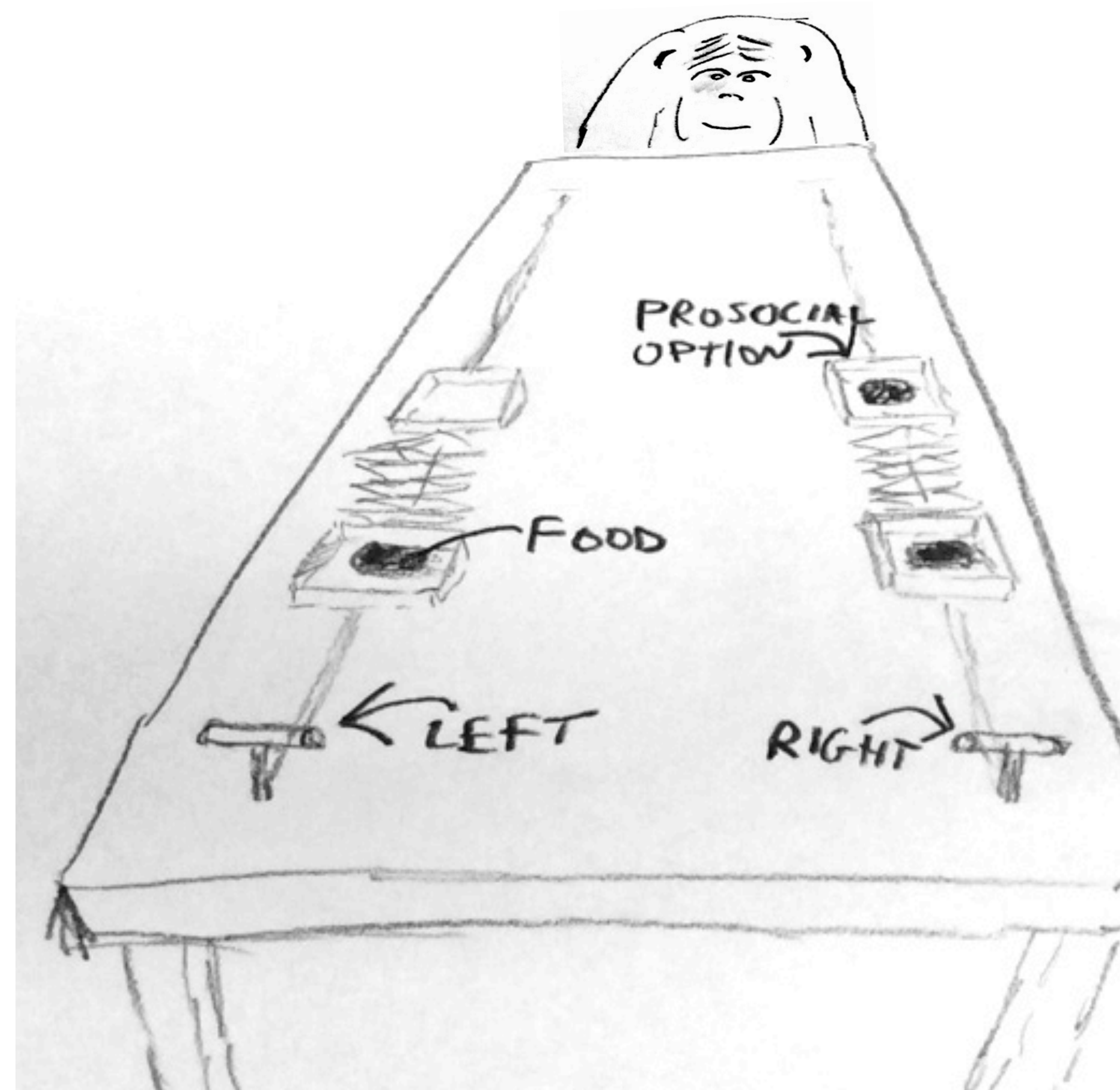
Prosocial chimpanzees

data(chimpanzees)

504 trials, 7 actors, 6 blocks

4 treatments:

- (1) right, no partner
- (2) left, no partner
- (3) right, partner
- (4) left, partner



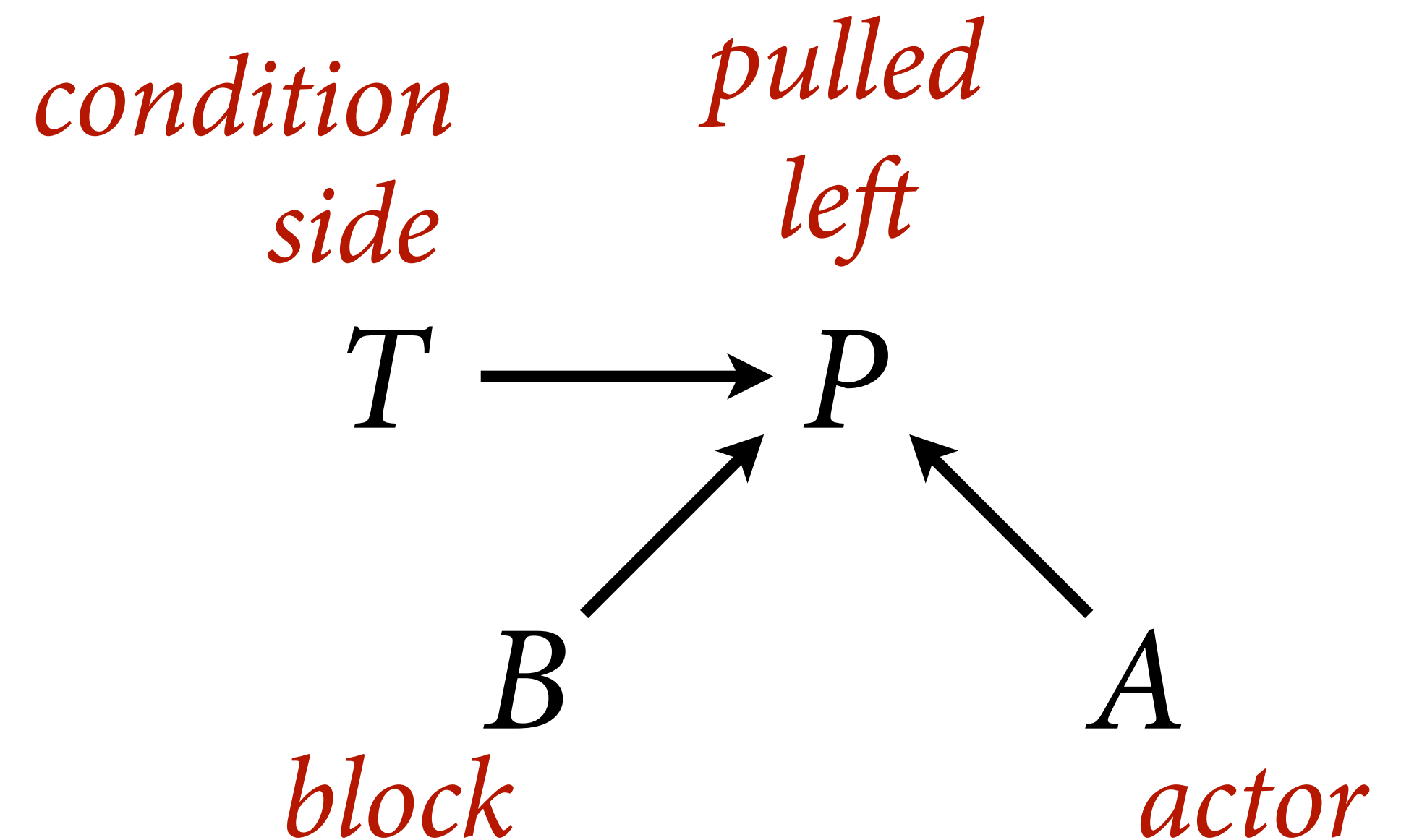
Prosocial chimpanzees

data(chimpanzees)

504 trials, 7 actors, 6 blocks

4 treatments:

- (1) right, no partner
- (2) left, no partner
- (3) right, partner
- (4) left, partner



Prosocial chimpanzees

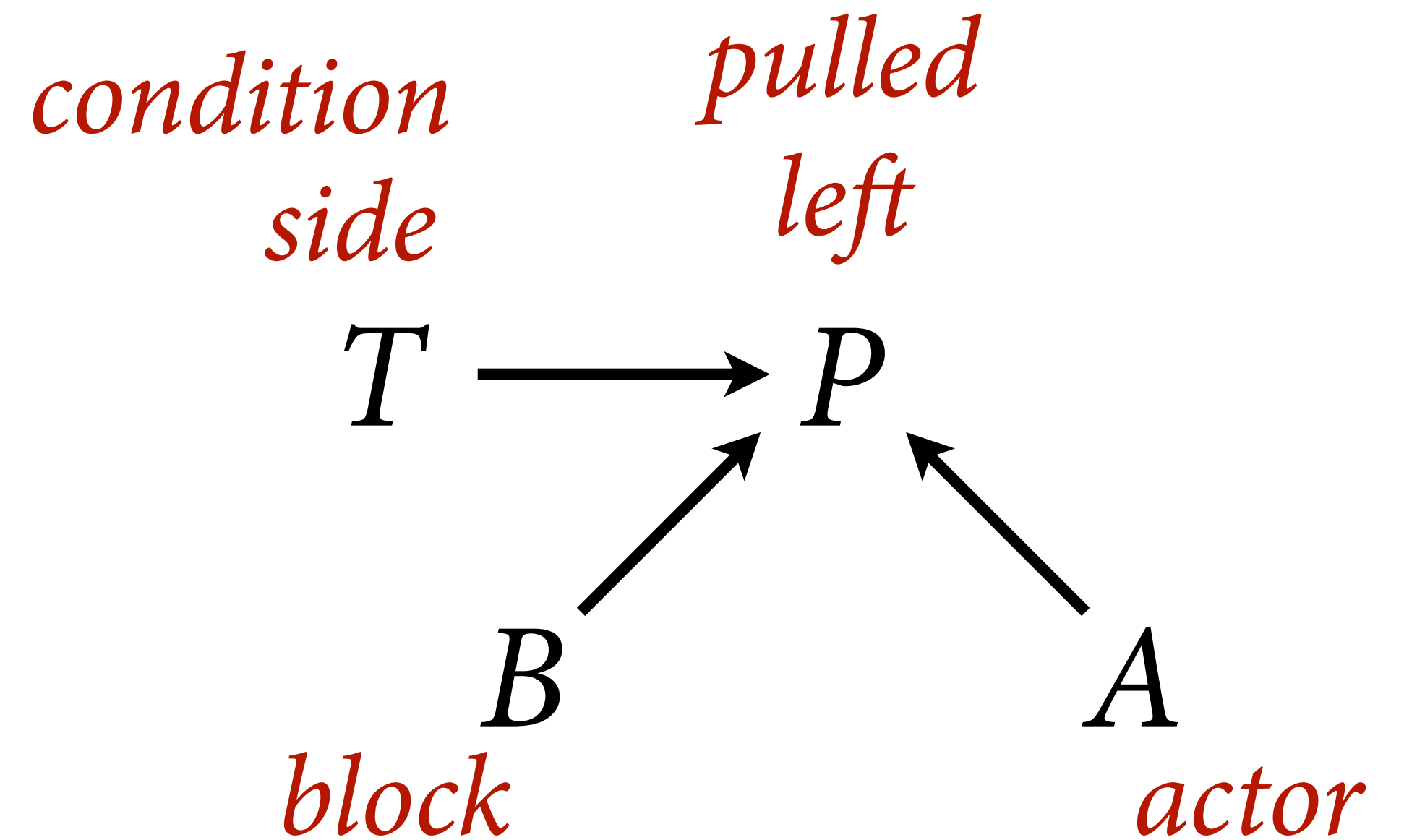
pulled

left

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

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

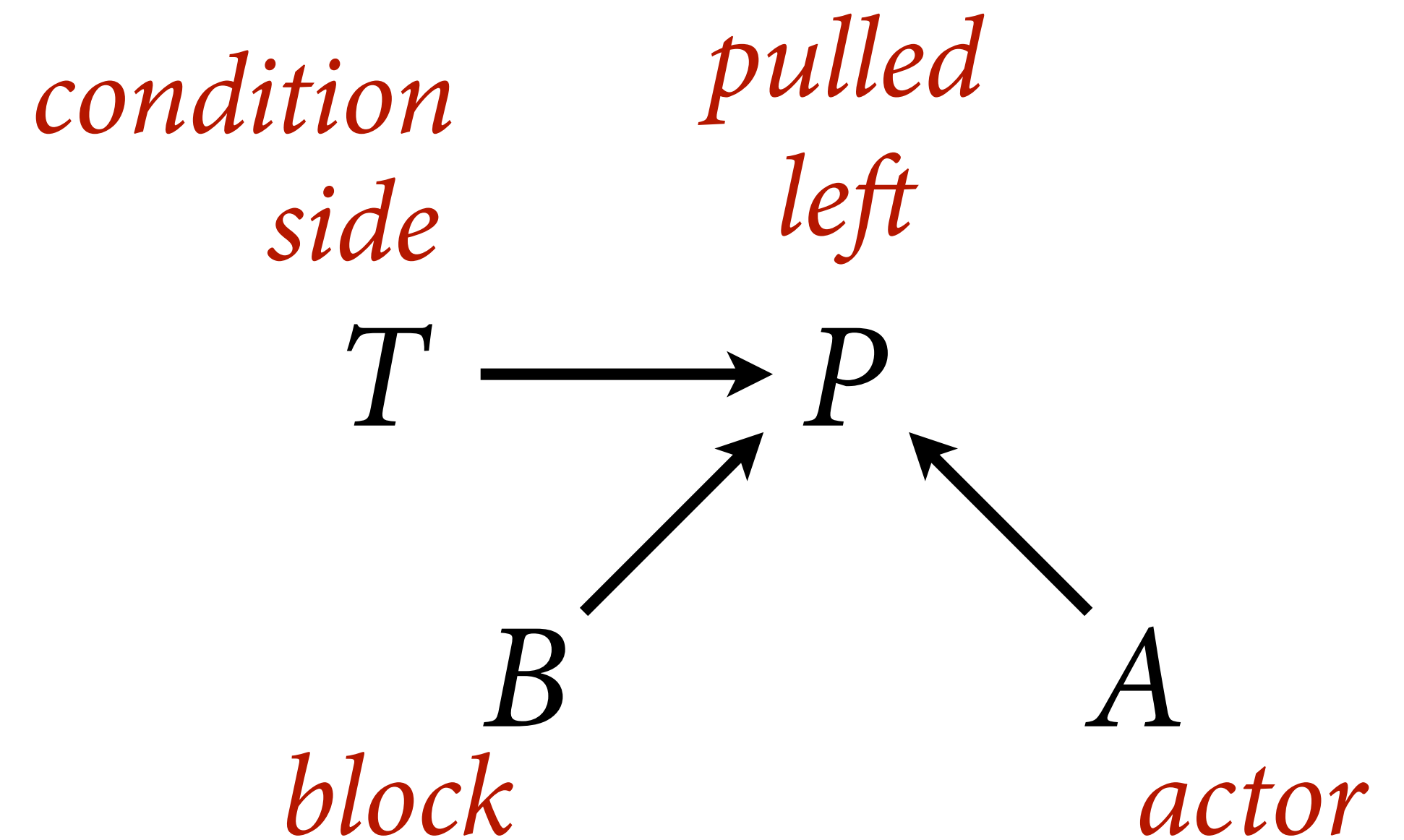
treatment *block* *actor*



Prosocial chimpanzees

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

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$



Prosocial chimpanzees

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

Probability of left lever

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

log-odds of left lever

Prosocial chimpanzees

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

Probability of left lever

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

log-odds of left lever

$$\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma_A)$$

Prior for actor effects (handedness)

Prosocial chimpanzees

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

Probability of left lever

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

log-odds of left lever

$$\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma_A)$$

Prior for actor effects (handedness)

$$\beta_{j,k} \sim \text{Normal}(0, \sigma_B)$$

Prior treatment/block effects

Prosocial chimpanzees

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

Probability of left lever

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

log-odds of left lever

$$\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma_A)$$

Prior for actor effects (handedness)

$$\beta_{j,k} \sim \text{Normal}(0, \sigma_B)$$

Prior treatment/block effects

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

Prior for “variance components”

Pooling treatment effects?!

Why is it reasonable to partially pool treatment effects?

$$\beta_{j,k} \sim \text{Normal}(0, \sigma_B)$$

Because treatments are not completely different

Because there are many possible treatments, you used a few

Because it results in better estimates

If parameters get the same prior, usually better to learn the prior from the sample



```

data(chimpanzees)
d <- chimpanzees
d$treatment <- 1 + d$prosoc_left + 2*d$condition
dat <- list(
  P = d$pulled_left,
  A = d$actor,
  B = d$block,
  T = d$treatment )

# block interactions
mBT <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- b[T,B] + a[A],
    ## adaptive priors
    matrix[T,B]:b ~ dnorm( 0 , sigma_B ),
    a[A] ~ dnorm( a_bar , sigma_A ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1)
  ) , data=dat , chains=4 , cores=4 )

```

```

> precis(mBT,3)

```

	mean	sd	5.5%	94.5%	n_eff	Rhat4
b[1,1]	-0.23	0.37	-0.85	0.34	1301	1.00
b[1,2]	-0.01	0.34	-0.56	0.51	2716	1.00
b[1,3]	0.32	0.36	-0.22	0.92	919	1.00
b[1,4]	0.11	0.35	-0.44	0.68	1790	1.00
b[1,5]	-0.36	0.37	-0.98	0.17	922	1.00
b[1,6]	-0.24	0.35	-0.85	0.28	1327	1.01
b[2,1]	0.09	0.35	-0.44	0.67	2071	1.00
b[2,2]	-0.01	0.36	-0.58	0.56	1689	1.00
b[2,3]	-0.12	0.34	-0.69	0.41	1738	1.00
b[2,4]	0.32	0.38	-0.21	0.98	1120	1.00
b[2,5]	0.20	0.35	-0.32	0.79	1769	1.00
b[2,6]	0.67	0.45	0.02	1.42	333	1.01
b[3,1]	-0.36	0.38	-1.01	0.17	735	1.00
b[3,2]	-0.03	0.36	-0.61	0.53	2061	1.00
b[3,3]	-0.21	0.35	-0.79	0.33	1464	1.00
b[3,4]	-0.46	0.38	-1.11	0.06	527	1.01
b[3,5]	0.03	0.37	-0.53	0.62	2478	1.00
b[3,6]	-0.34	0.37	-0.99	0.19	883	1.00
b[4,1]	-0.37	0.40	-1.06	0.20	873	1.01
b[4,2]	0.28	0.36	-0.24	0.88	1205	1.00
b[4,3]	0.27	0.35	-0.24	0.86	1105	1.00
b[4,4]	0.08	0.37	-0.49	0.68	1828	1.00
b[4,5]	0.06	0.34	-0.48	0.60	2025	1.00
b[4,6]	0.45	0.41	-0.13	1.17	597	1.01
a[1]	-0.35	0.27	-0.78	0.09	1408	1.00
a[2]	4.70	1.25	3.09	7.01	1214	1.00
a[3]	-0.62	0.27	-1.08	-0.20	1481	1.00
a[4]	-0.64	0.27	-1.07	-0.20	1424	1.00
a[5]	-0.36	0.27	-0.78	0.06	1237	1.00
a[6]	0.60	0.26	0.18	1.04	1654	1.00
a[7]	2.15	0.39	1.56	2.82	1436	1.00
a_bar	0.62	0.69	-0.47	1.71	1635	1.00
sigma_A	2.02	0.66	1.17	3.17	1482	1.00
sigma_B	0.46	0.18	0.18	0.74	198	1.02

```

data(chimpanzees)
d <- chimpanzees
d$treatment <- 1 + d$prosoc_left + 2*d$condition
dat <- list(
  P = d$pulled_left,
  A = d$actor,
  B = d$block,
  T = d$treatment )

```

```

# block interactions
mBT <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- b[T,B] + a[A]
    ## adaptive priors
    matrix[T,B]:b ~ dnorm( 0 , sigma_B )
    a[A] ~ dnorm( a_bar , sigma_A )
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 )
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1)
  ) , data=dat , chains=4 , cores=4 )

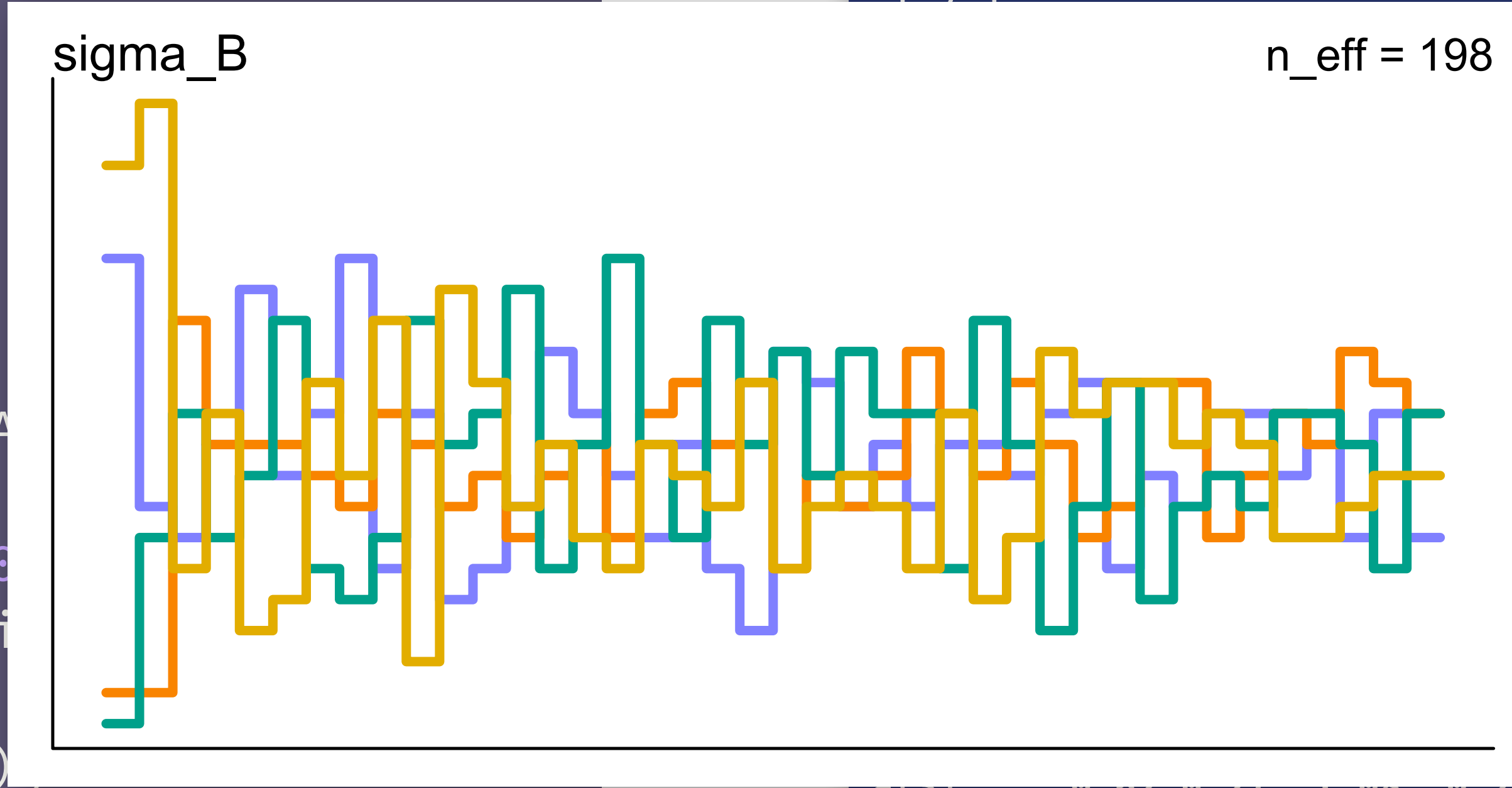
```

```

> precis(mBT,3)

```

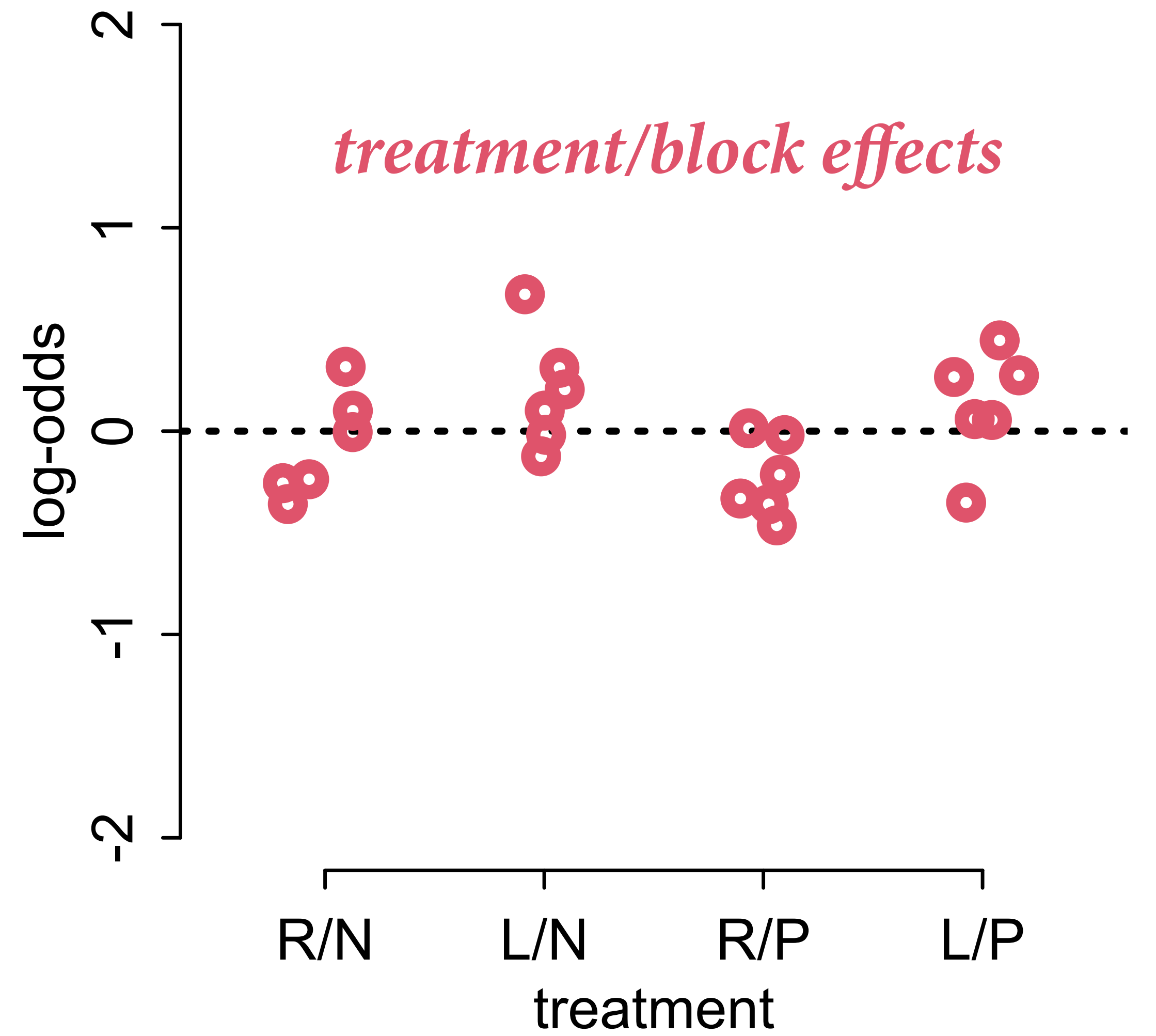
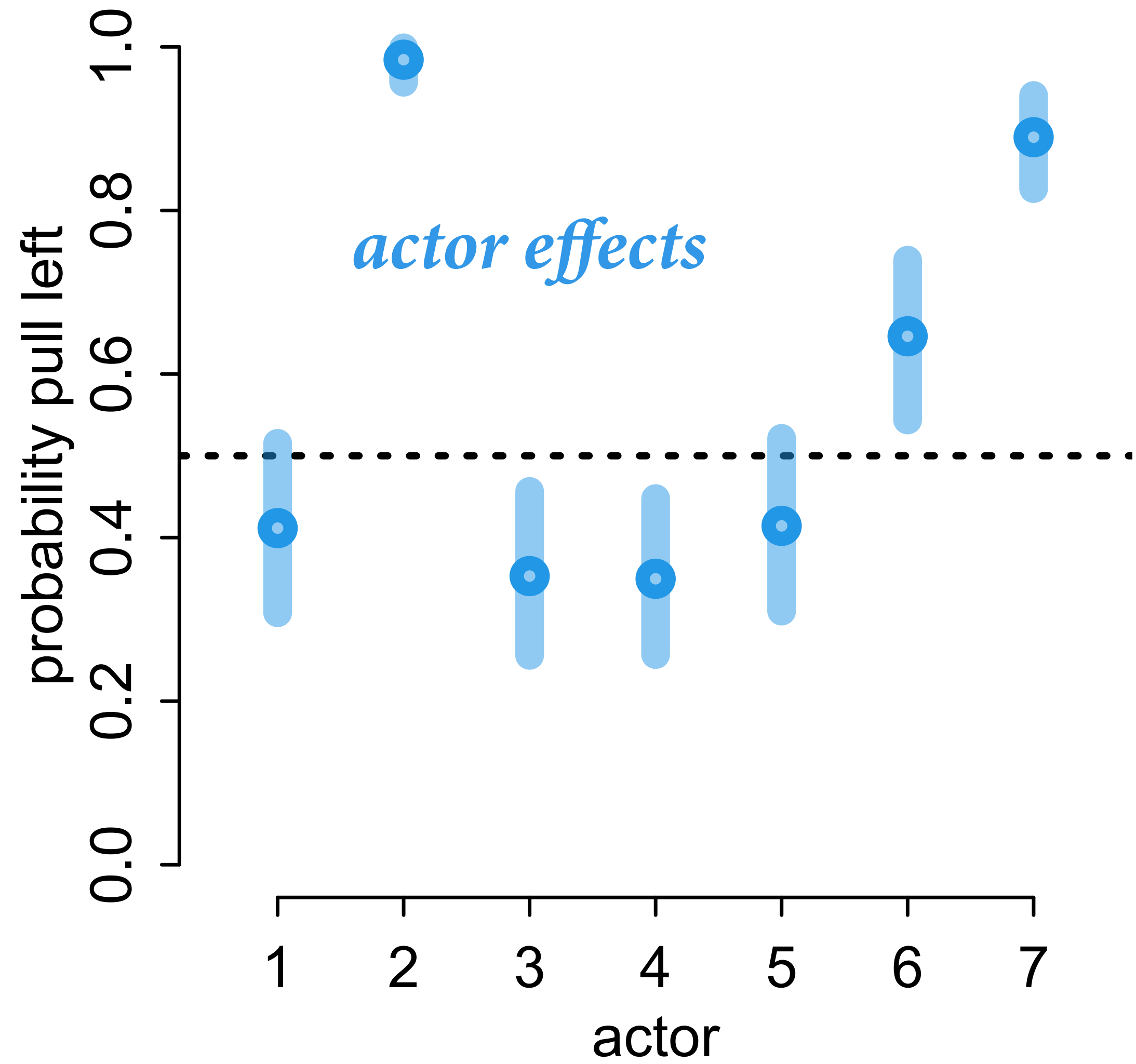
	mean	sd	5.5%	94.5%	n_eff	Rhat4
b[1,1]	-0.23	0.37	-0.85	0.34	1301	1.00
b[1,2]	-0.01	0.34	-0.56	0.51	2716	1.00
b[1,3]	0.32	0.36	-0.22	0.92	919	1.00
b[1,4]	0.11	0.35	-0.44	0.68	1790	1.00
b[1,5]	-0.36	0.37	-0.98	0.17	922	1.00
b[1,6]	-0.24	0.35	-0.85	0.28	1327	1.01
b[2,1]	0.09	0.35	-0.44	0.67	2071	1.00
b[2,2]	-0.01	0.36	-0.58	0.56	1689	1.00
b[2,3]	-0.12	0.34	-0.69	0.41	1738	1.00
b[2,4]	0.32	0.38	-0.21	0.98	1120	1.00

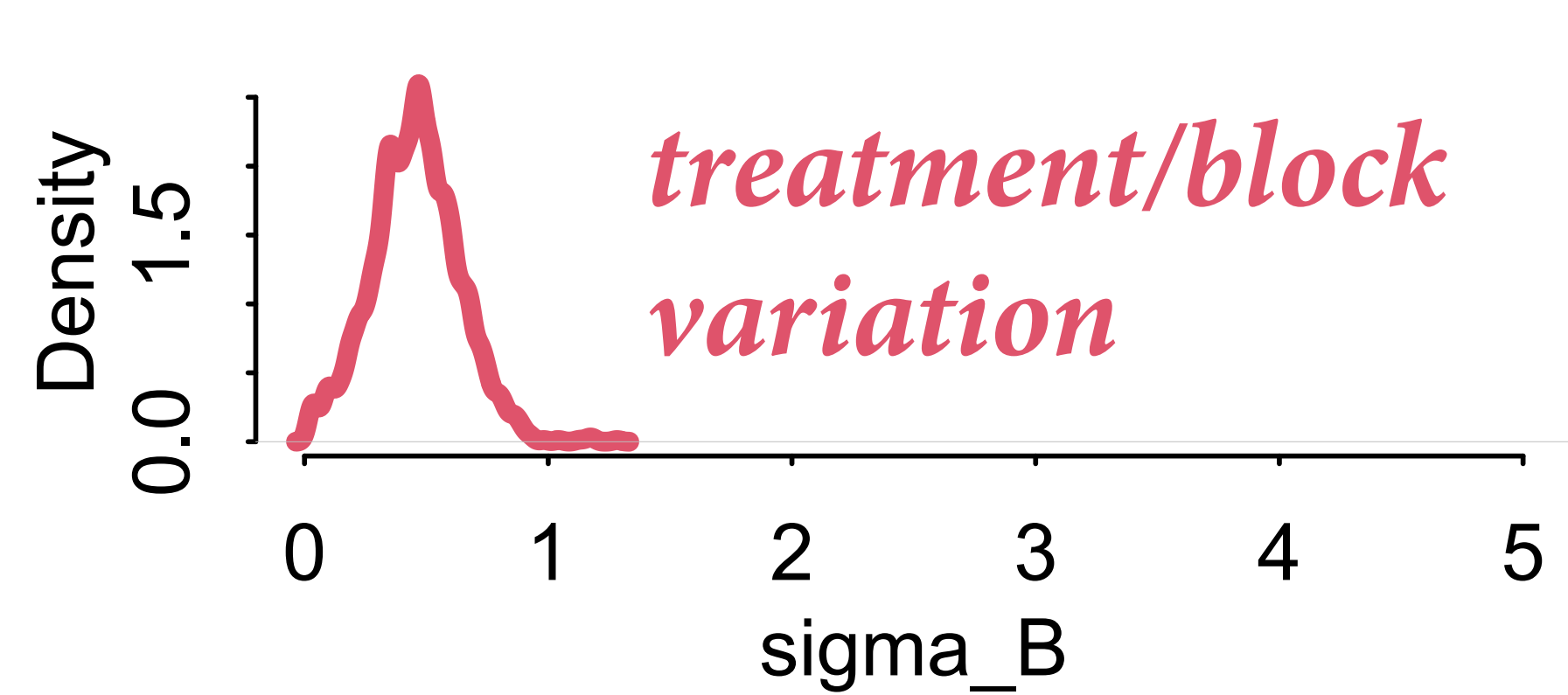
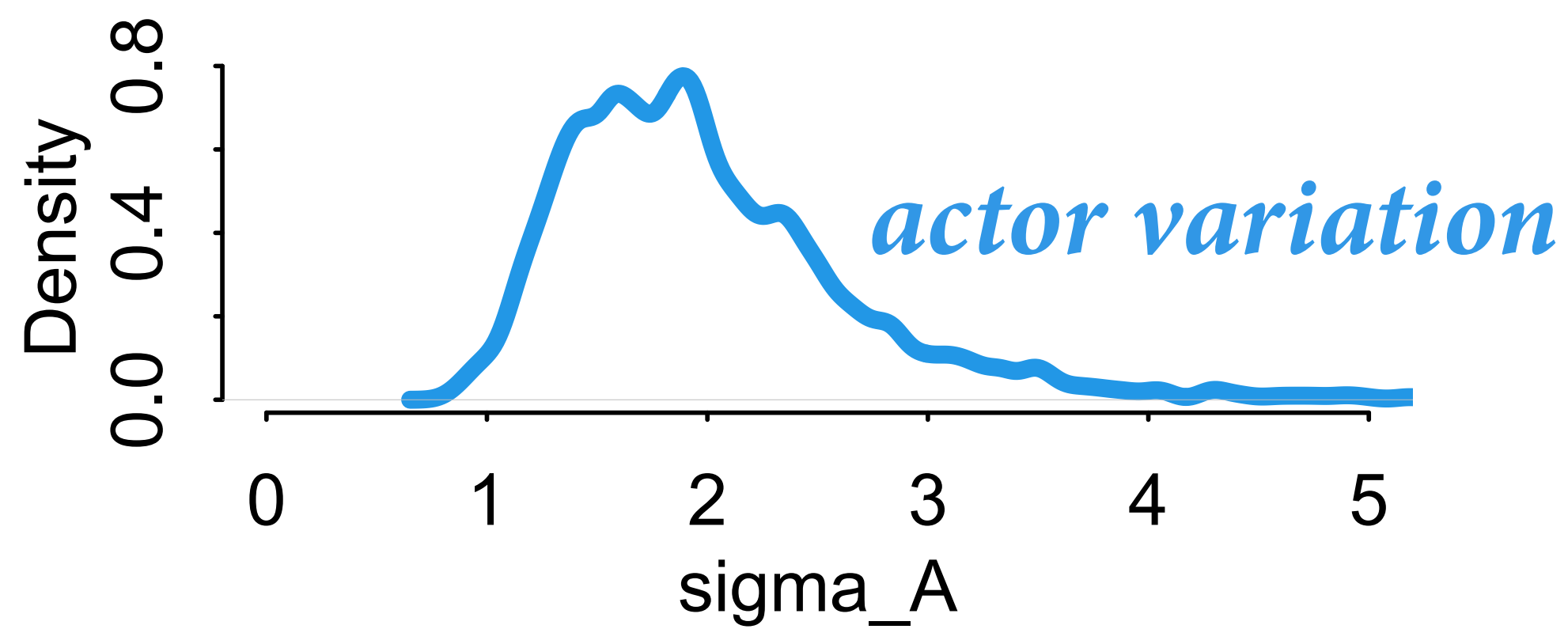
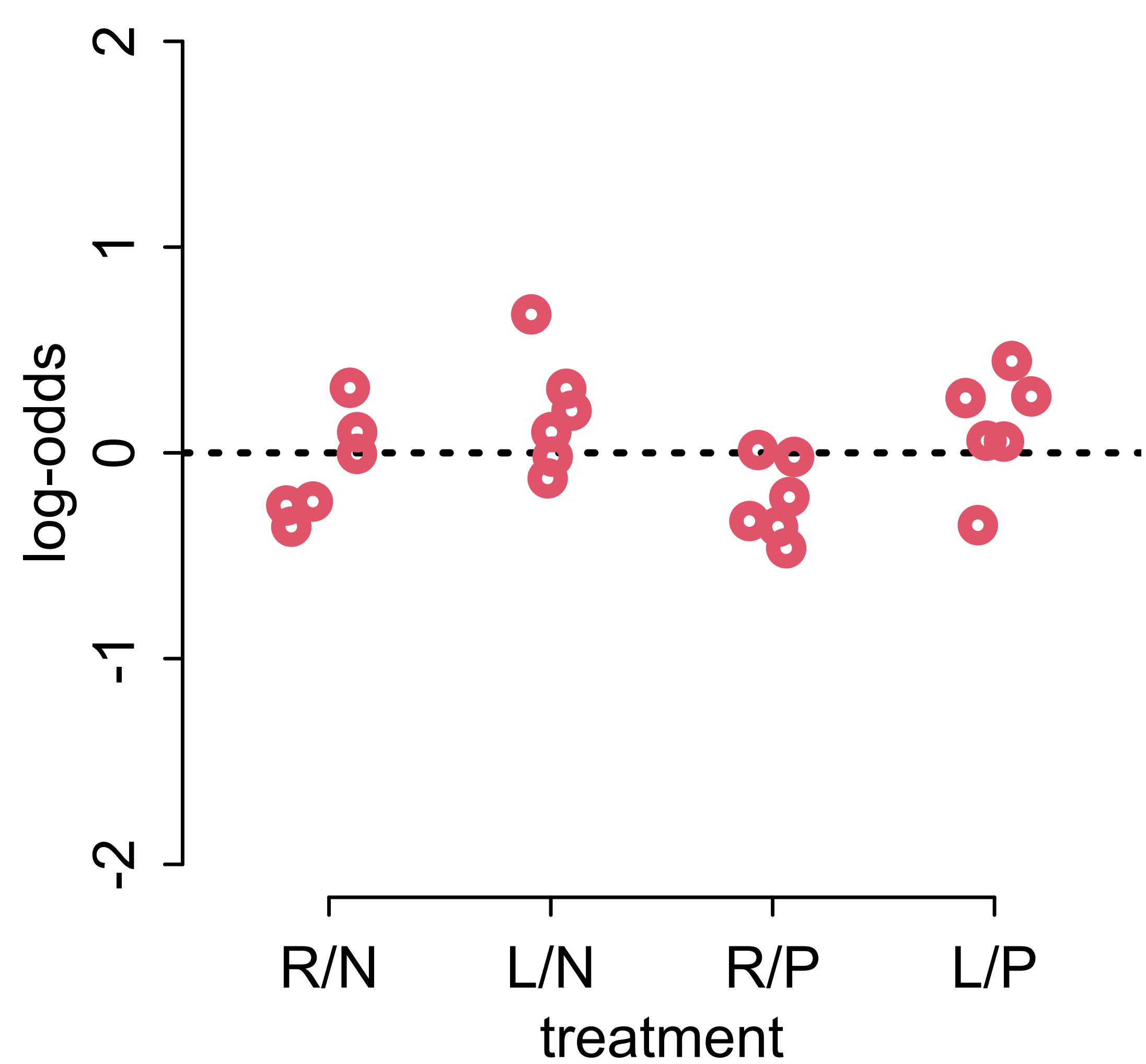
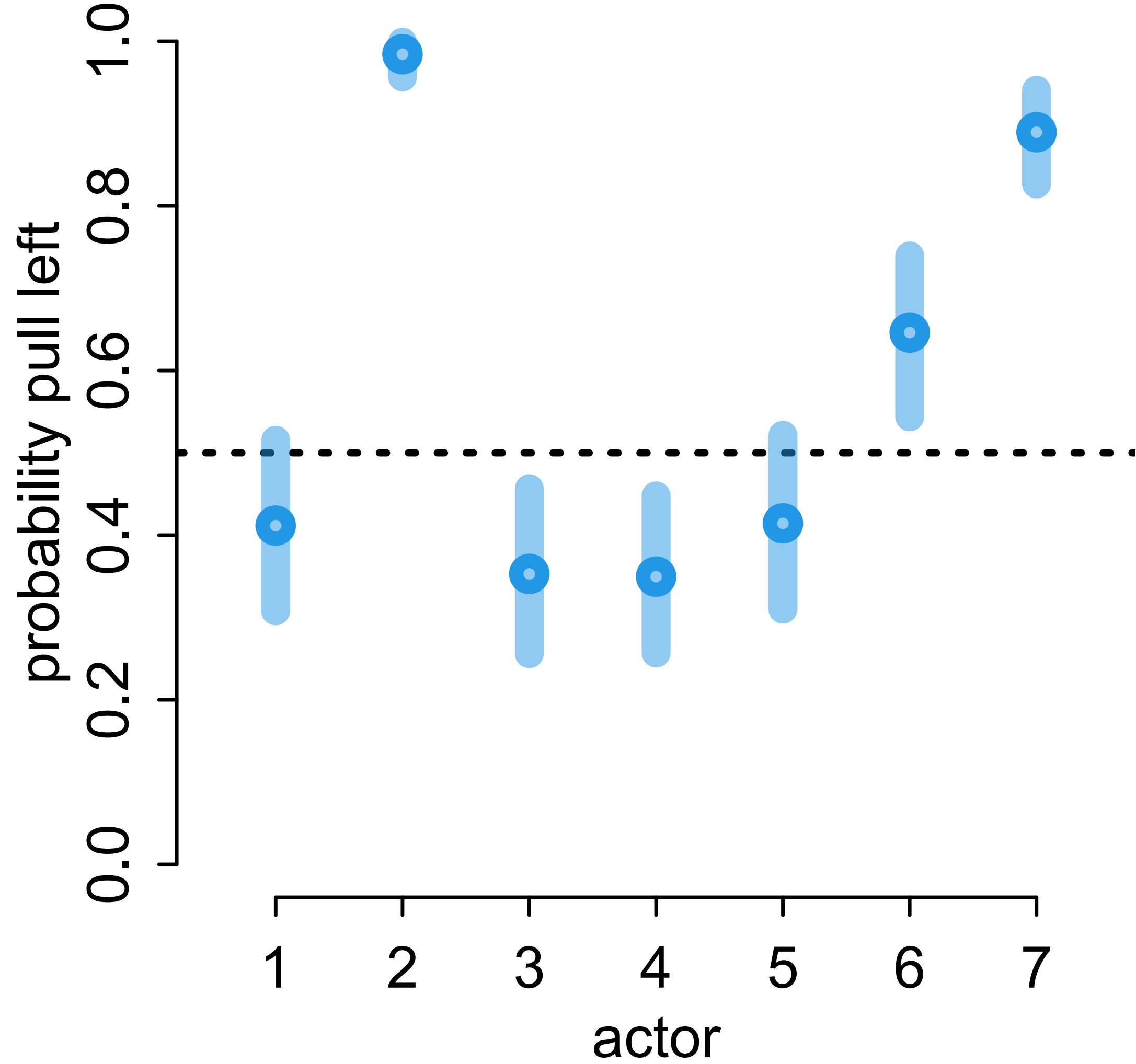


```


```

					1769	1.00
					333	1.01
					735	1.00
					2061	1.00
					1464	1.00
					527	1.01
					2478	1.00
					883	1.00
					873	1.01
					1205	1.00
					1105	1.00
					1828	1.00
					2025	1.00
					597	1.01
					1408	1.00
					1214	1.00
					1481	1.00
a[5]	0.02	0.27	-1.08	0.20		
a[4]	-0.64	0.27	-1.07	-0.20	1424	1.00
a[5]	-0.36	0.27	-0.78	0.06	1237	1.00
a[6]	0.60	0.26	0.18	1.04	1654	1.00
a[7]	2.15	0.39	1.56	2.82	1436	1.00
a_bar	0.62	0.69	-0.47	1.71	1635	1.00
sigma_A	2.02	0.66	1.17	3.17	1482	1.00
sigma_B	0.46	0.18	0.18	0.74	198	1.02





Variance does not add!

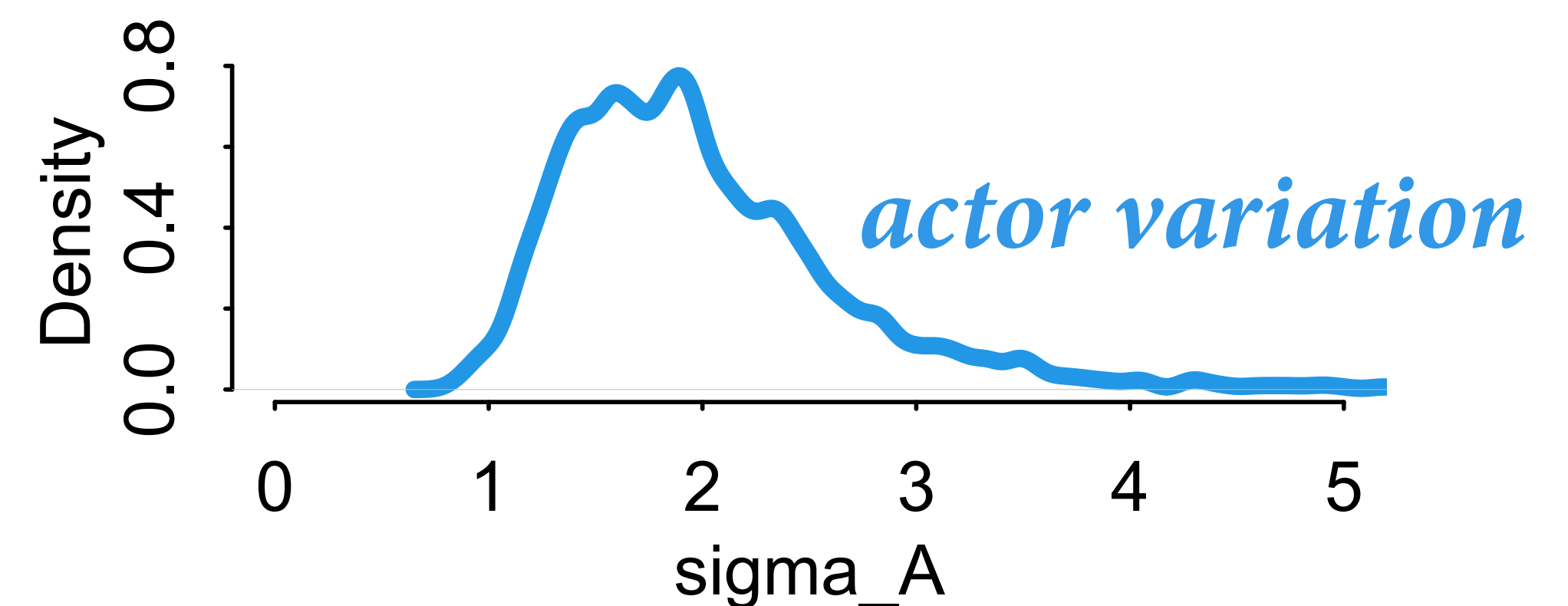
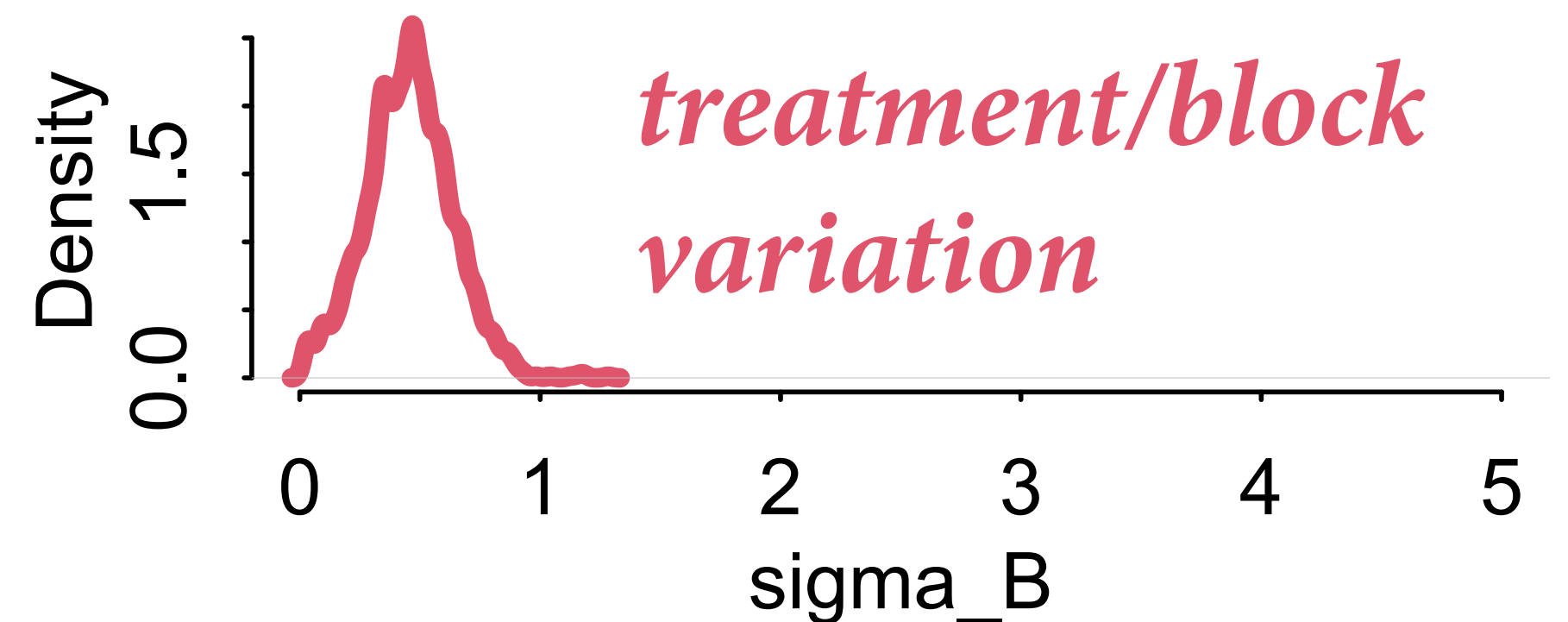
In linear models, variance components are **additive**

Total variation in outcome is sum of the components

Not true for generalized linear models

Link function breaks additivity

Variation in one component **moderates** variation in the others

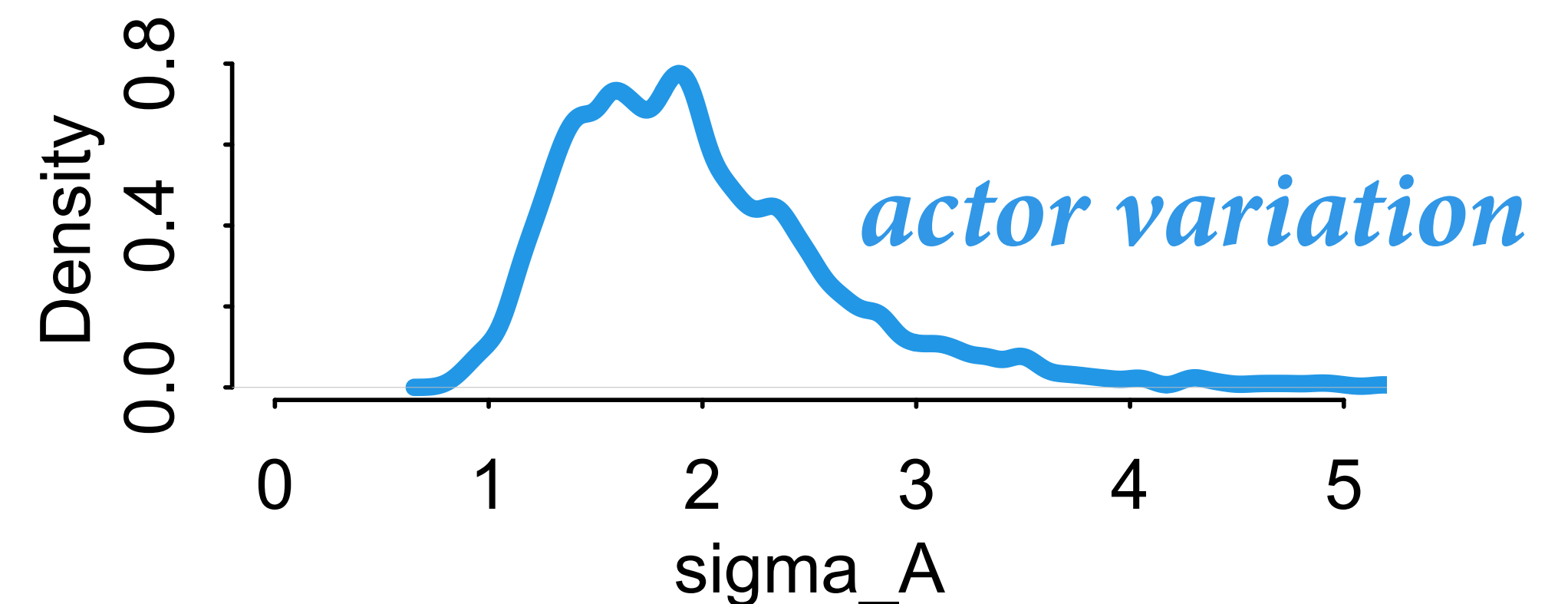
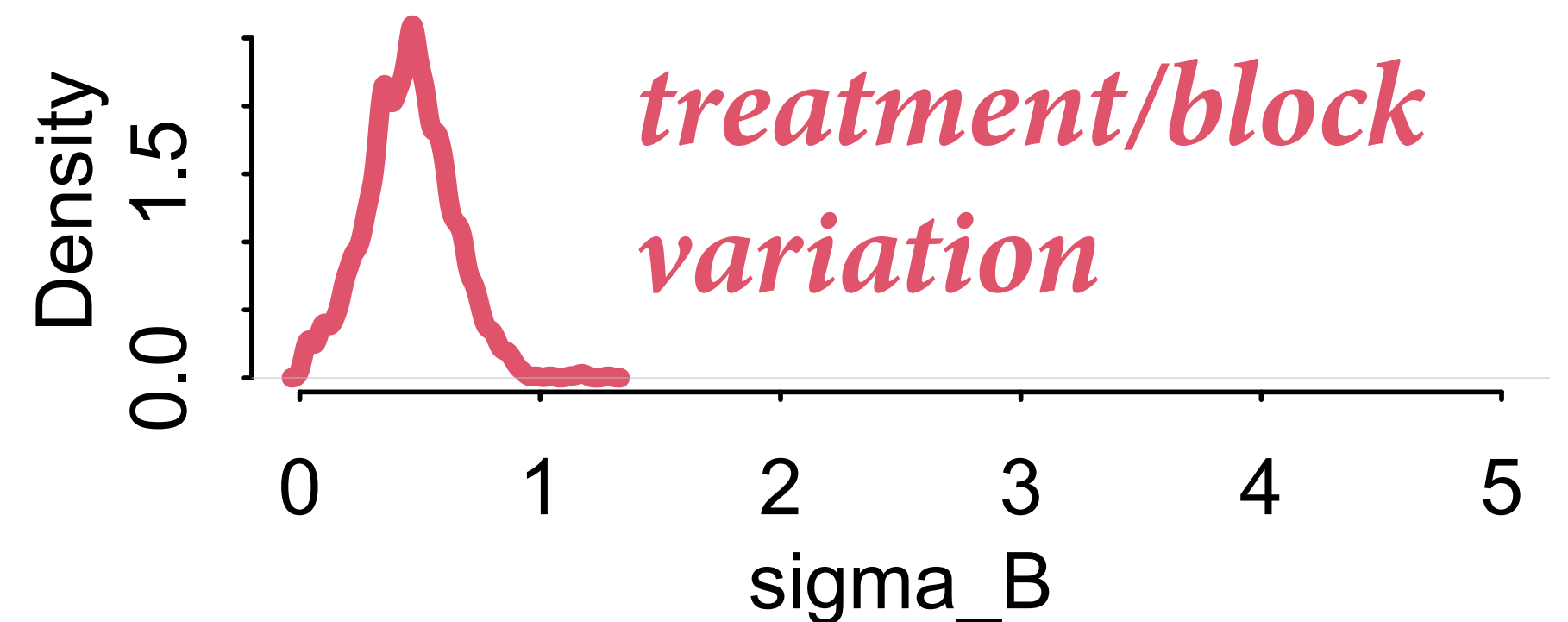


Multilevel predictions & effects

How to compute predictions and interventions (causal effects)?

Predict for same groups: Use varying effect estimates for each group

Predict for new groups: Ignore varying effect estimates, marginalize over population distribution



New groups

Reedfrog intervention

Target population 50% predation,
25% large tadpoles

What is causal effect of increasing
size to 75% large?

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

dat <- list(
  S = d$surv,
  D = d$density,
  T = 1:nrow(d),
  P = ifelse(d$pred=="no", 1L, 2L),
  G = ifelse(d$size=="small", 1L, 2L)
)

mSPG <- ulam(
  alist(
    S ~ binomial( D , p ),
    logit(p) <- a[T] + b[P,G],
    a[T] ~ normal( 0 , sigma ),
    matrix[P,G]:b ~ normal( 0 , 1 ),
    sigma ~ exponential( 1 )
  ), data=dat , chains=4 , cores=4 )
```



```
post <- extract.samples(mSPG)

# sim under status quo
n_groups <- 1000
n_samples <- 2000
S1 <- matrix(0,nrow=n_samples,ncol=n_groups)
for ( s in 1:n_groups ) {
  # sim a tank from posterior population
  aT <- rnorm(n_samples,0,post$sigma)
  # sample P and G for this group
  P <- sample( 1:2 , size=1 , prob=c(0.5,0.5) ) # 50% pred
  G <- sample( 1:2 , size=1 , prob=c(0.75,0.25) ) # 25% large
  # sim survival
  p <- inv_logit( aT + post$b[,P,G] )
  S1[,s] <- rbinom(2000,35,p)
}
```

```
post <- extract.samples(mSPG)

# sim under status quo
n_groups <- 1000
n_samples <- 2000
S1 <- matrix(0,nrow=n_samples,ncol=n_groups)
for ( s in 1:n_groups ) {
  # sim a tank from posterior population
  aT <- rnorm(n_samples,0,post$sigma)
  # sample P and G for this group
  P <- sample( 1:2 , size=1 , prob=c(0.5,0.5) ) # 50% pred
  G <- sample( 1:2 , size=1 , prob=c(0.75,0.25) ) # 25% large
  # sim survival
  p <- inv_logit( aT + post$b[,P,G] )
  S1[,s] <- rbinom(2000,35,p)
}
```

```
post <- extract.samples(mSPG)

# sim under status quo
n_groups <- 1000
n_samples <- 2000
S1 <- matrix(0,nrow=n_samples,ncol=n_groups)
for ( s in 1:n_groups ) {
  # sim a tank from posterior population
  aT <- rnorm(n_samples,0,post$sigma)
  # sample P and G for this group
  P <- sample( 1:2 , size=1 , prob=c(0.5,0.5) ) # 50% pred
  G <- sample( 1:2 , size=1 , prob=c(0.75,0.25) ) # 25% large
  # sim survival
  p <- inv_logit( aT + post$b[,P,G] )
  S1[,s] <- rbinom(2000,35,p)
}
```

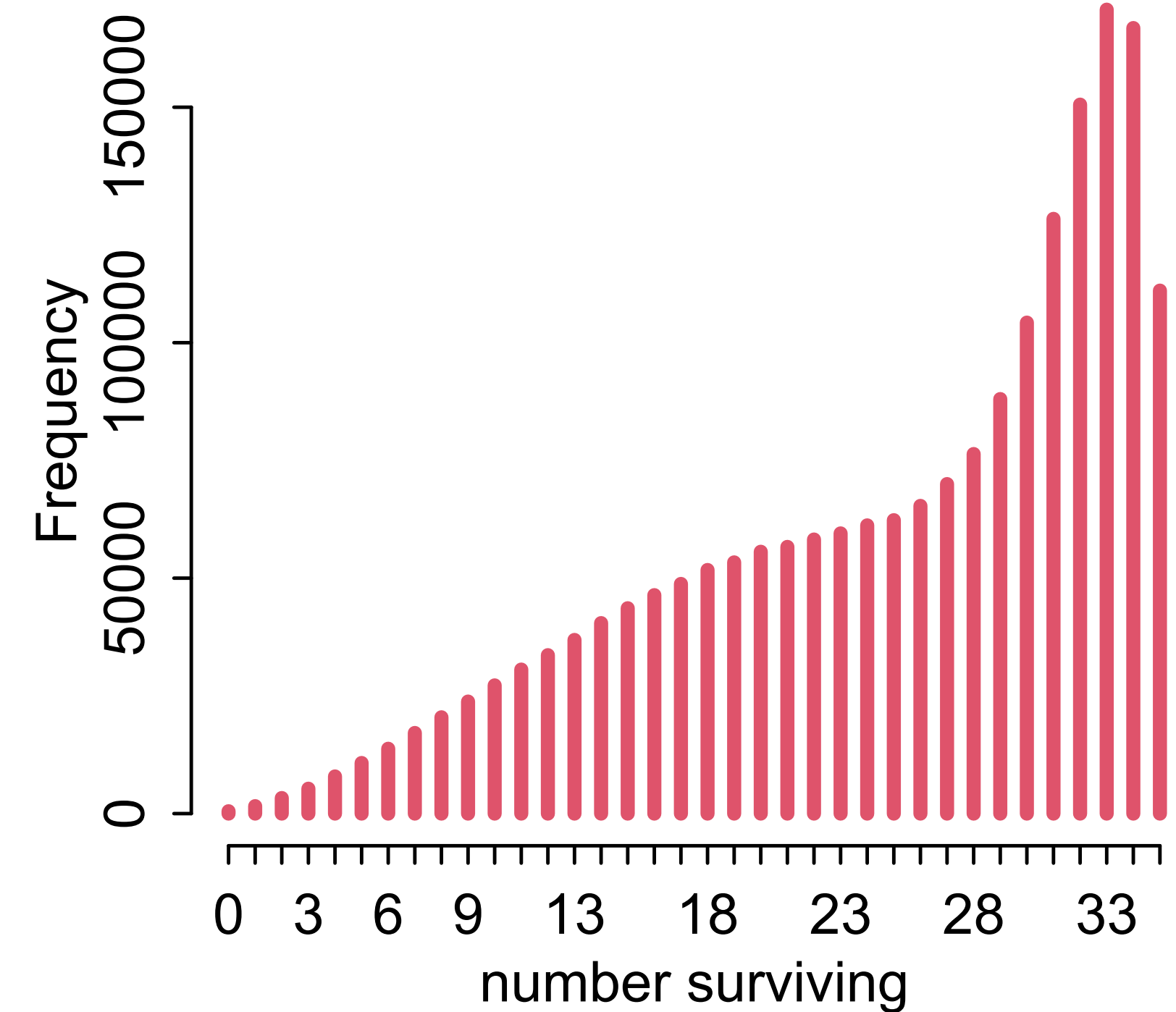
```
post <- extract.samples(mSPG)

# sim under status quo
n_groups <- 1000
n_samples <- 2000
S1 <- matrix(0, nrow=n_samples, ncol=n_groups)
for ( s in 1:n_groups ) {
  # sim a tank from posterior population
  aT <- rnorm(n_samples, 0, post$sigma)
  # sample P and G for this group
  P <- sample( 1:2 , size=1 , prob=c(0.5,0.5) ) # 50% pred
  G <- sample( 1:2 , size=1 , prob=c(0.75,0.25) ) # 25% large
  # sim survival
  p <- inv_logit( aT + post$b[,P,G] )
  S1[,s] <- rbinom(2000, 35, p)
}
```

Reedfrog status quo

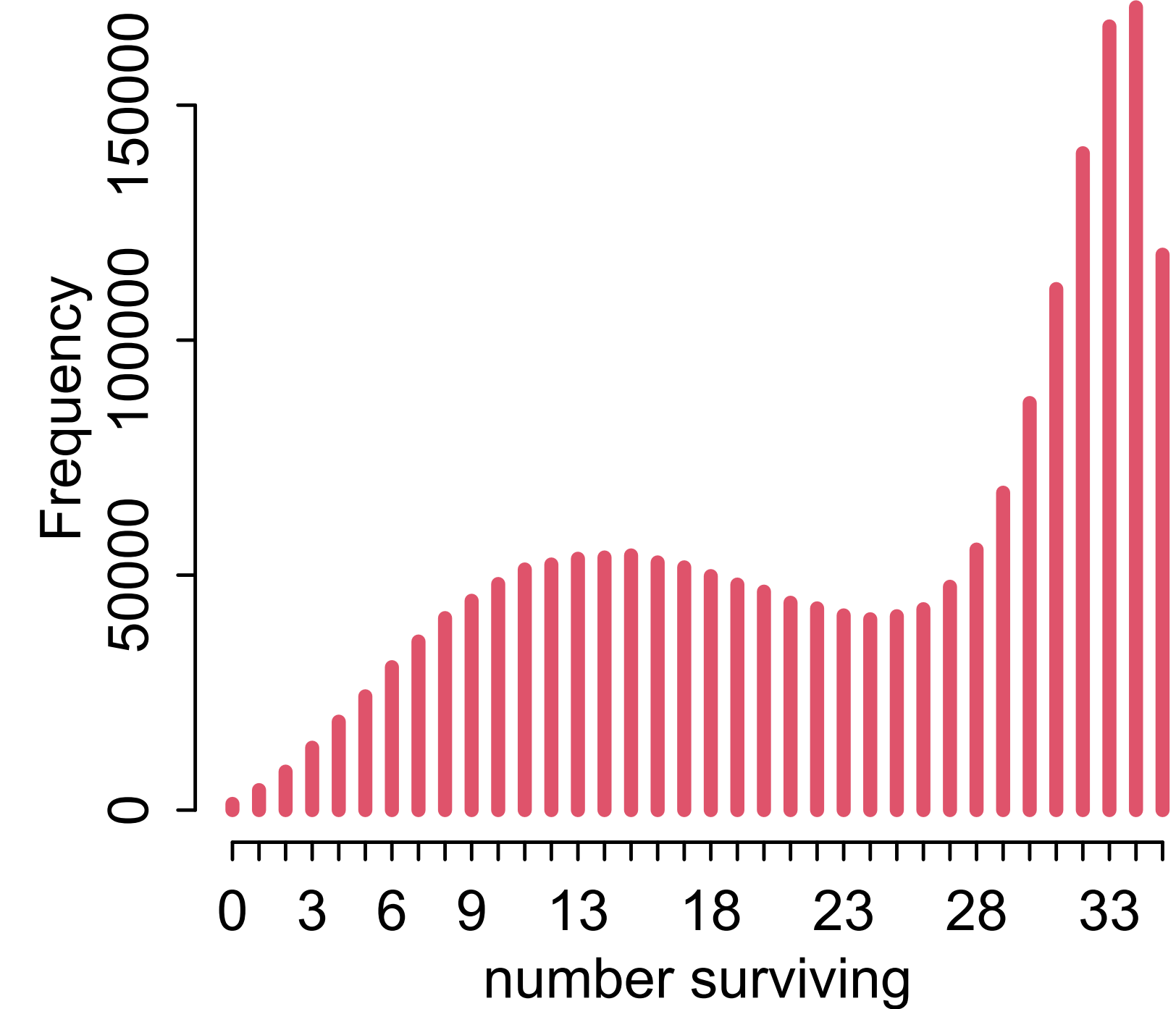
```
post <- extract.samples(mSPG)

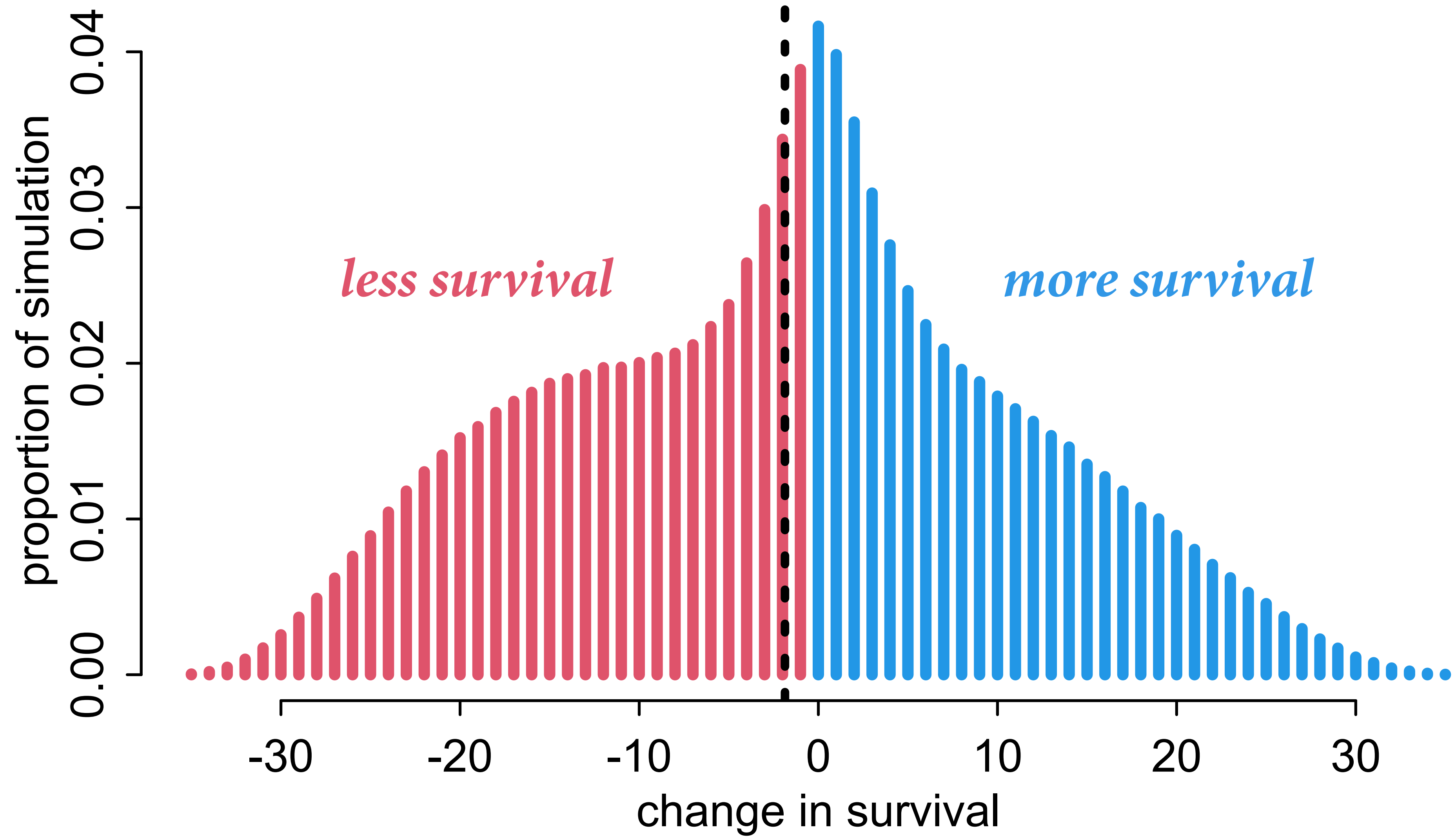
# sim under status quo
n_groups <- 1000
n_samples <- 2000
S1 <- matrix(0,nrow=n_samples,ncol=n_groups)
for ( s in 1:n_groups ) {
  # sim a tank from posterior population
  aT <- rnorm(n_samples,0,post$sigma)
  # sample P and G for this group
  P <- sample( 1:2 , size=1 , prob=c(0.5,0.5) ) # 50% pred
  G <- sample( 1:2 , size=1 , prob=c(0.75,0.25) ) # 25% large
  # sim survival
  p <- inv_logit( aT + post$b[,P,G] )
  S1[,s] <- rbinom(2000,35,p)
}
```



Reedfrog intervention

```
# intervention - 50% large
S2 <- matrix(0,nrow=n_samples,ncol=n_groups)
for ( s in 1:n_groups ) {
  # sim a tank from posterior population
  aT <- rnorm(n_samples,0,post$sigma)
  # sample P and G for this group
  P <- sample( 1:2 , size=1 , prob=c(0.5,0.5) ) # 50% pred
  G <- sample( 1:2 , size=1 , prob=c(0.25,0.75) ) # 75% large
  # sim survival
  p <- inv_logit( aT + post$b[,P,G] )
  S2[,s] <- rbinom(n_samples,35,p)
}
```



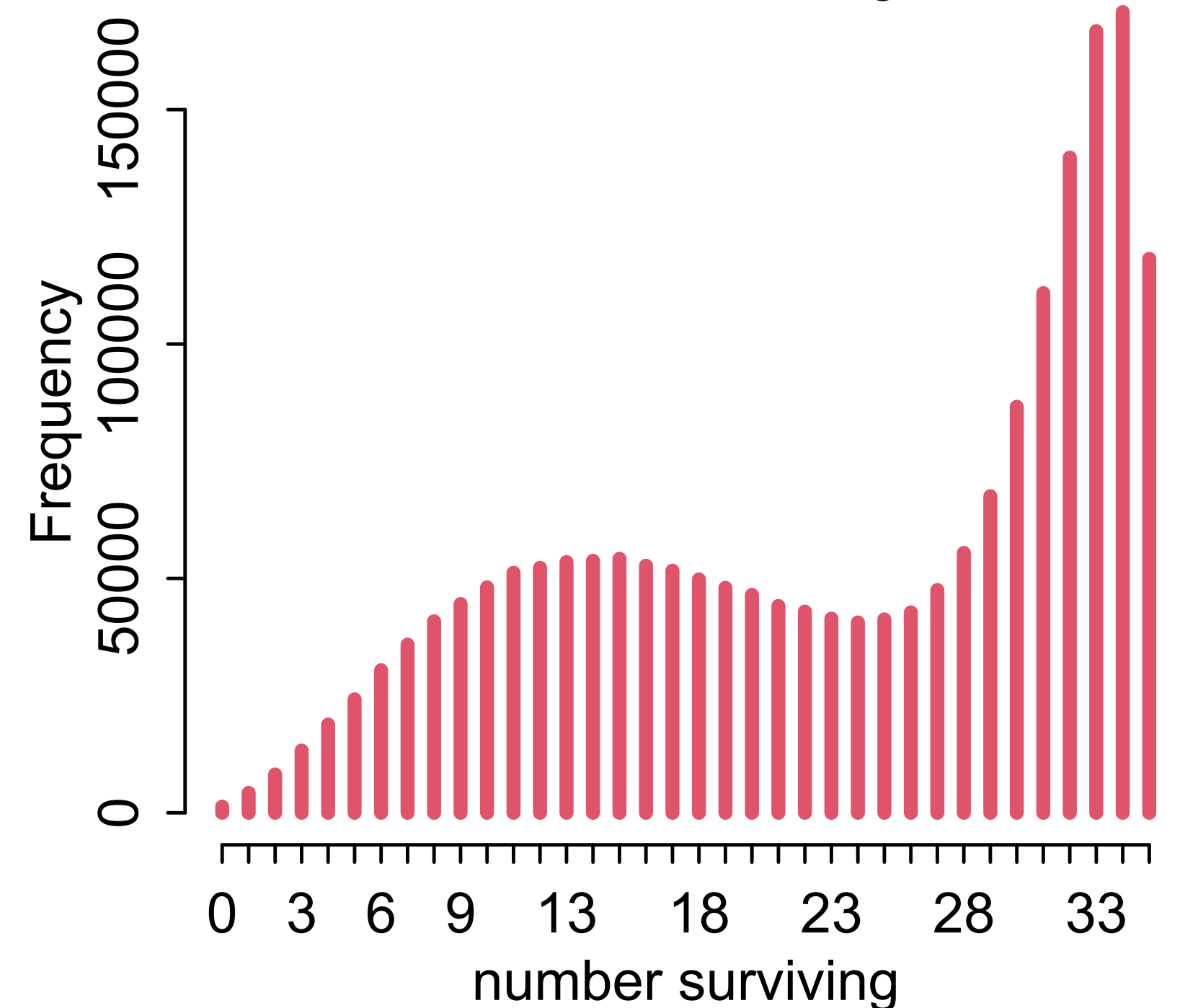
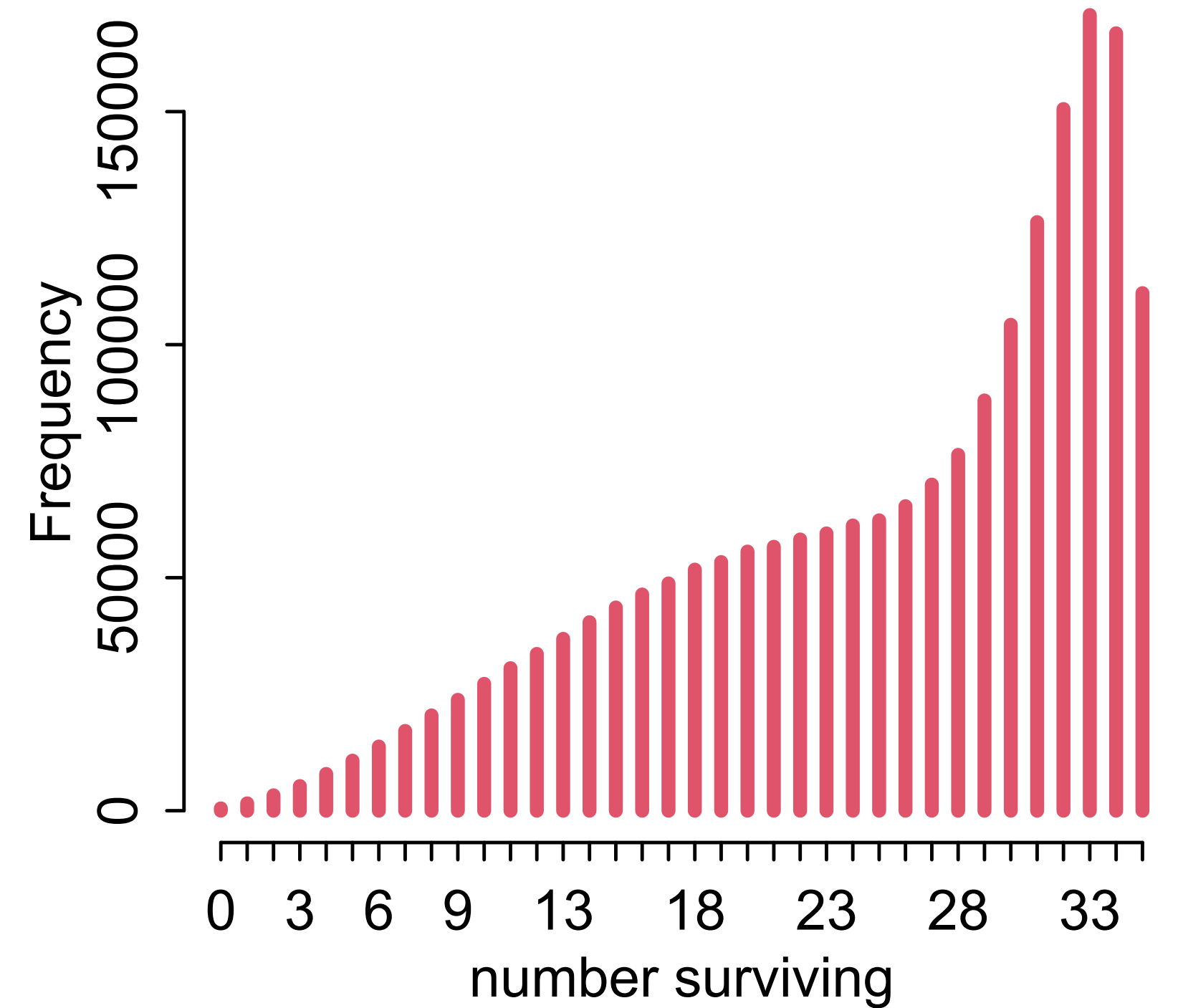


Multilevel predictions

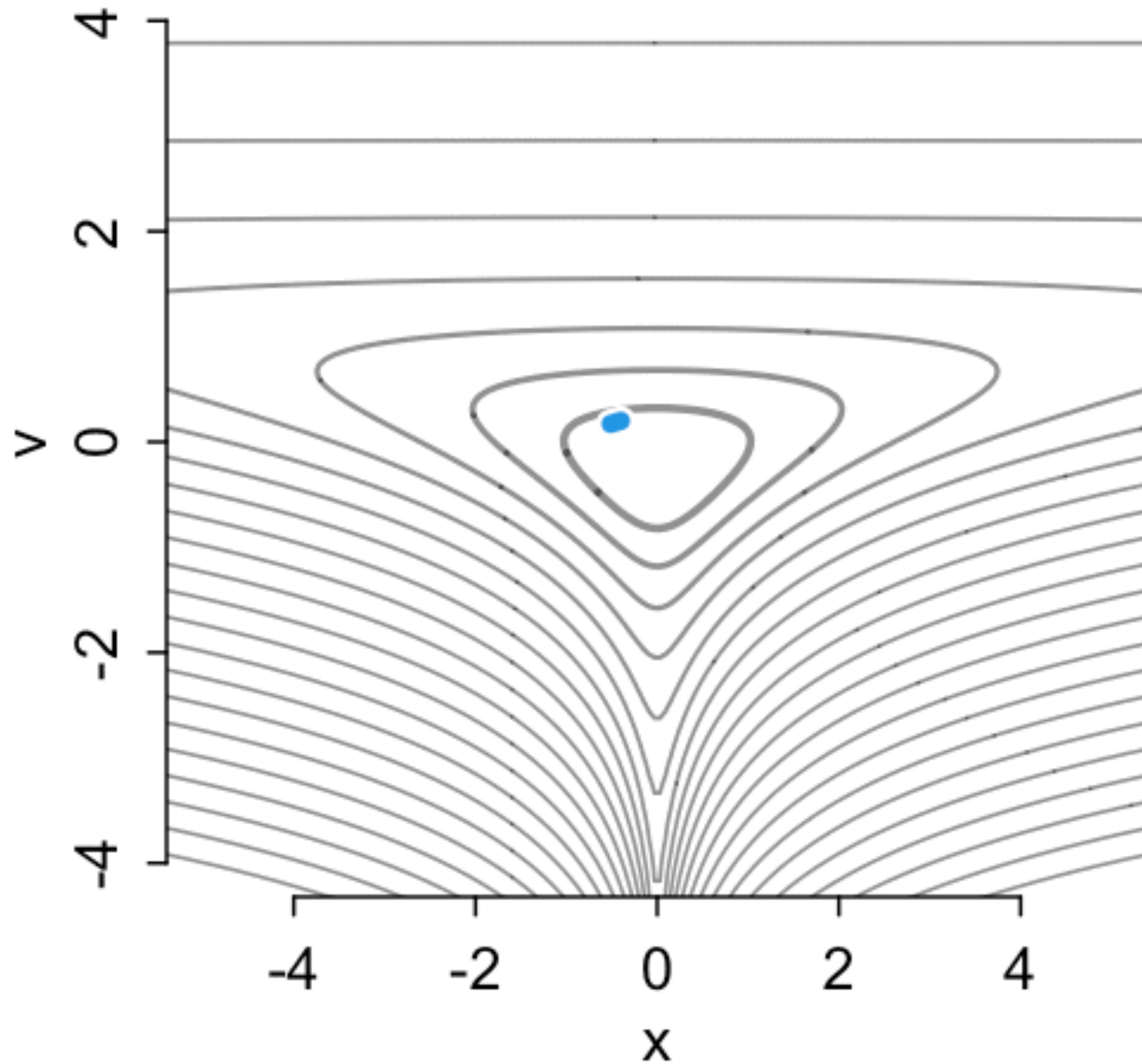
Group variation **moderates** causal effects

Averaging over group variation means **simulating groups** (or using estimates from observed groups as appropriate)

If you have a **generative model**, you can simulate interventions for new targets



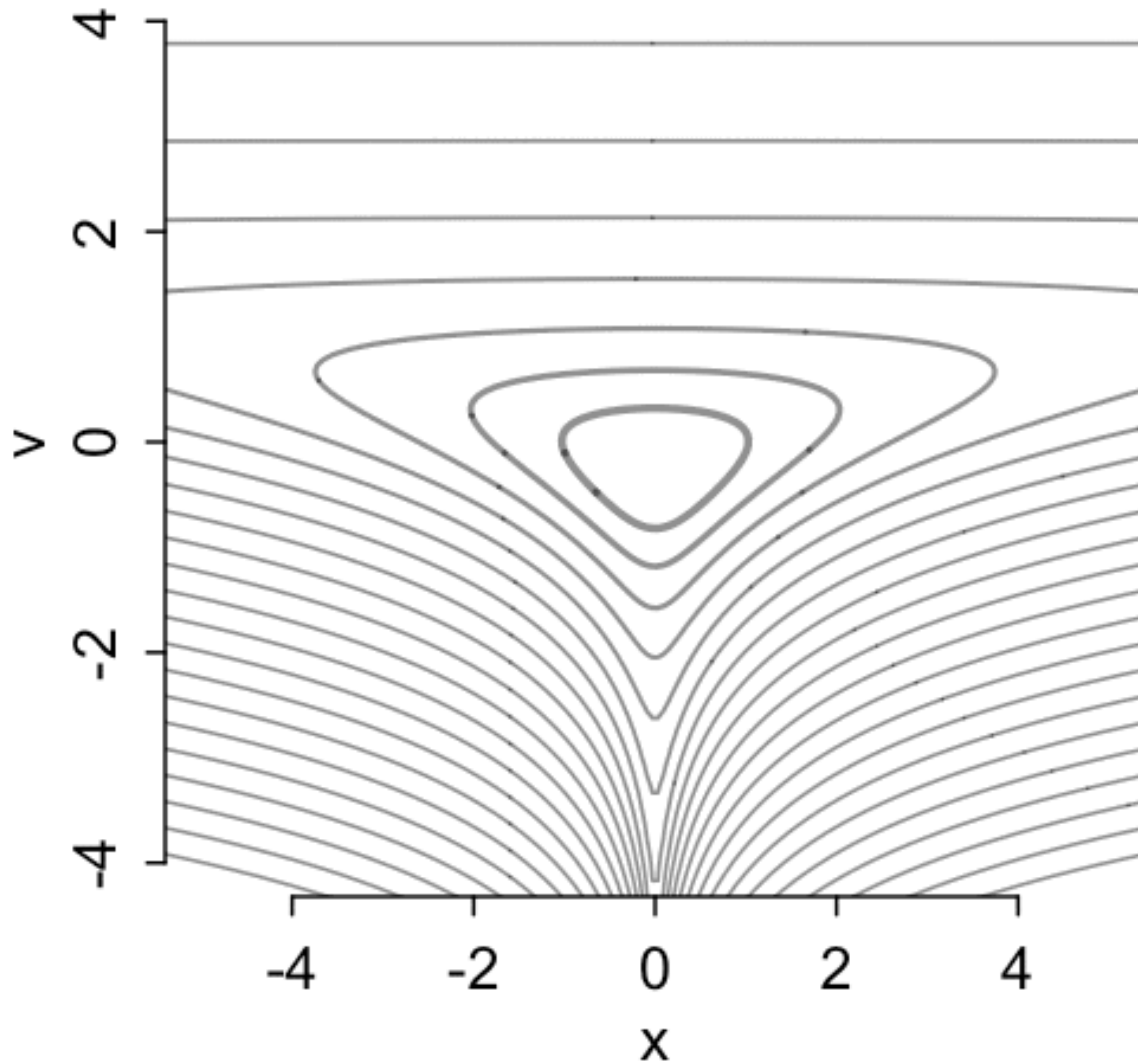
PAUSE



$$v \sim \text{Normal}(0, 0.5)$$

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

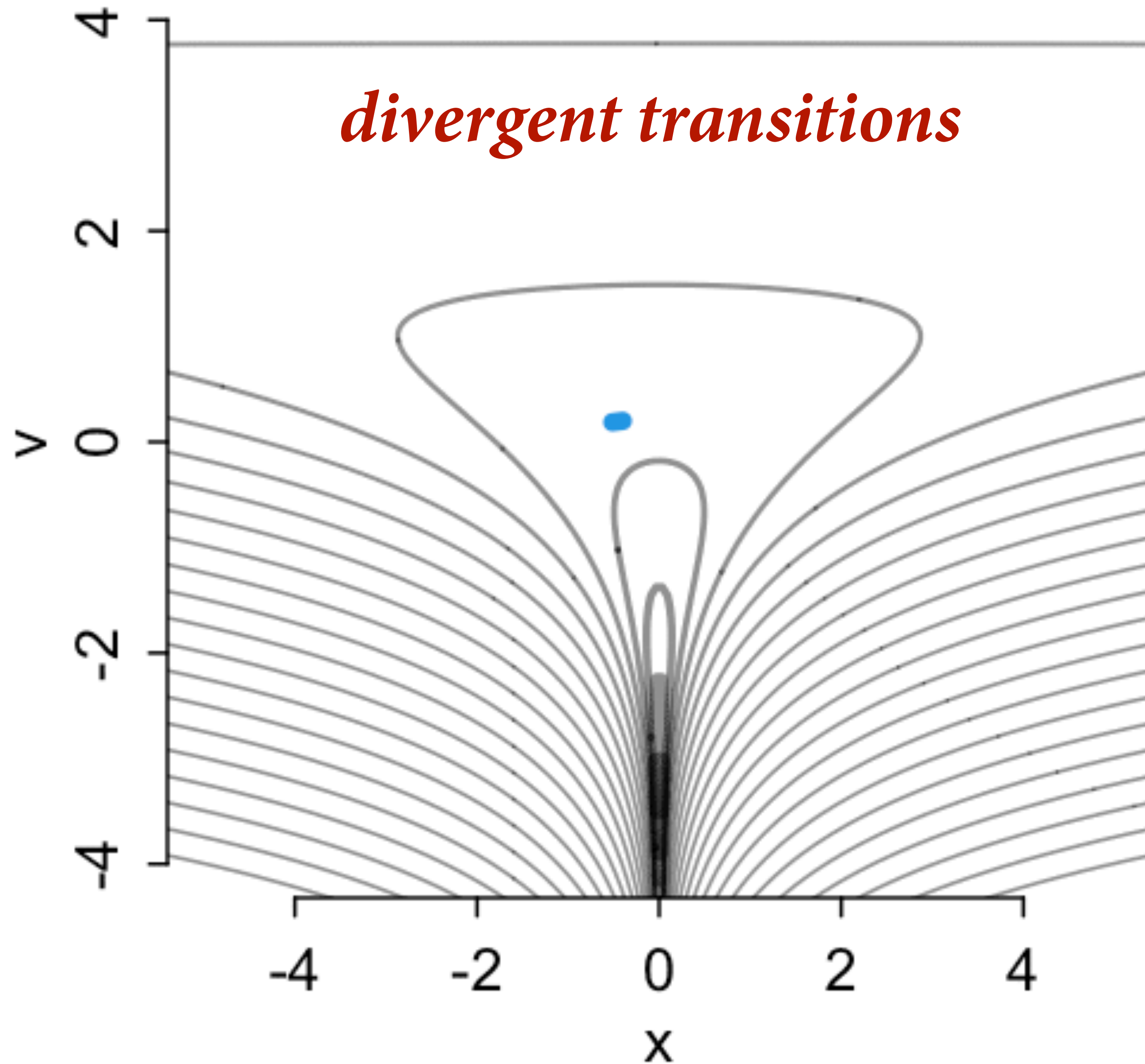
$v \sim \text{normal}(0, 0.5)$



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

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

divergent transitions



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

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



Divergent transitions

Why? Same step size not optimal everywhere

High curvature = simulation cannot follow surface

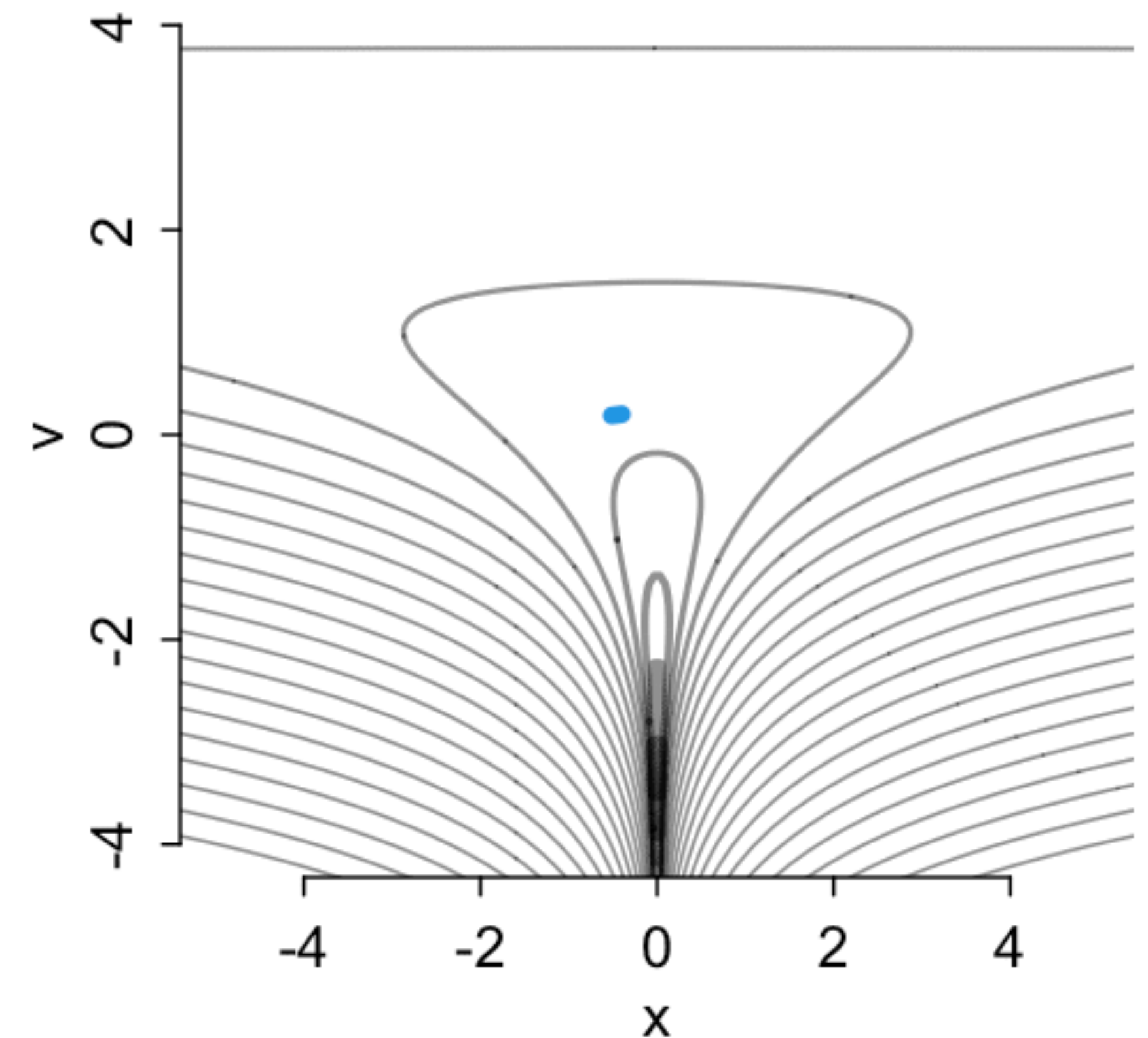
What can we do?

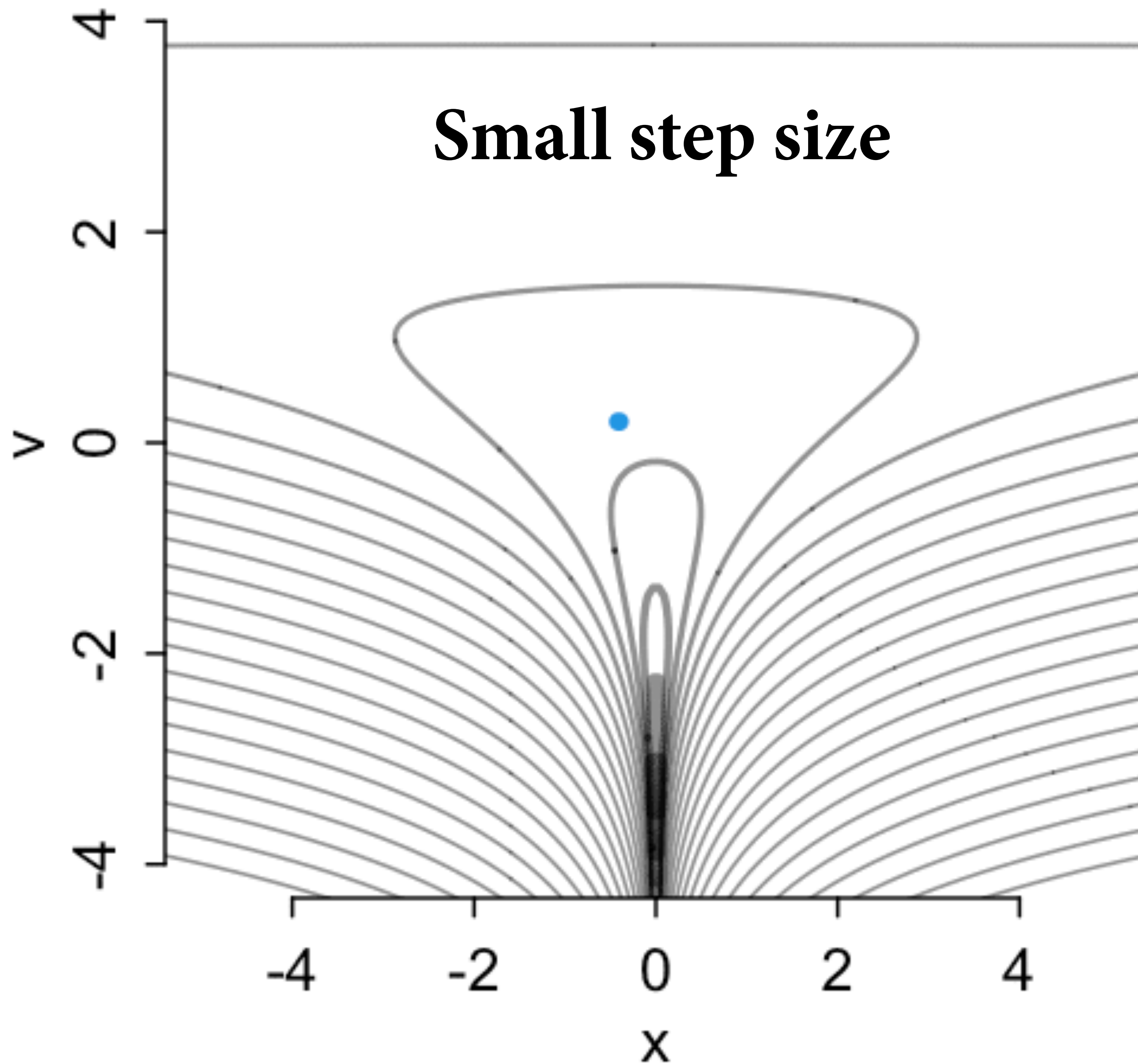
(1) use a smaller step size

(2) reparameterize!

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

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





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

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

Small step size helps, but
makes exploration slow

“Centered”

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

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

“Centered”

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

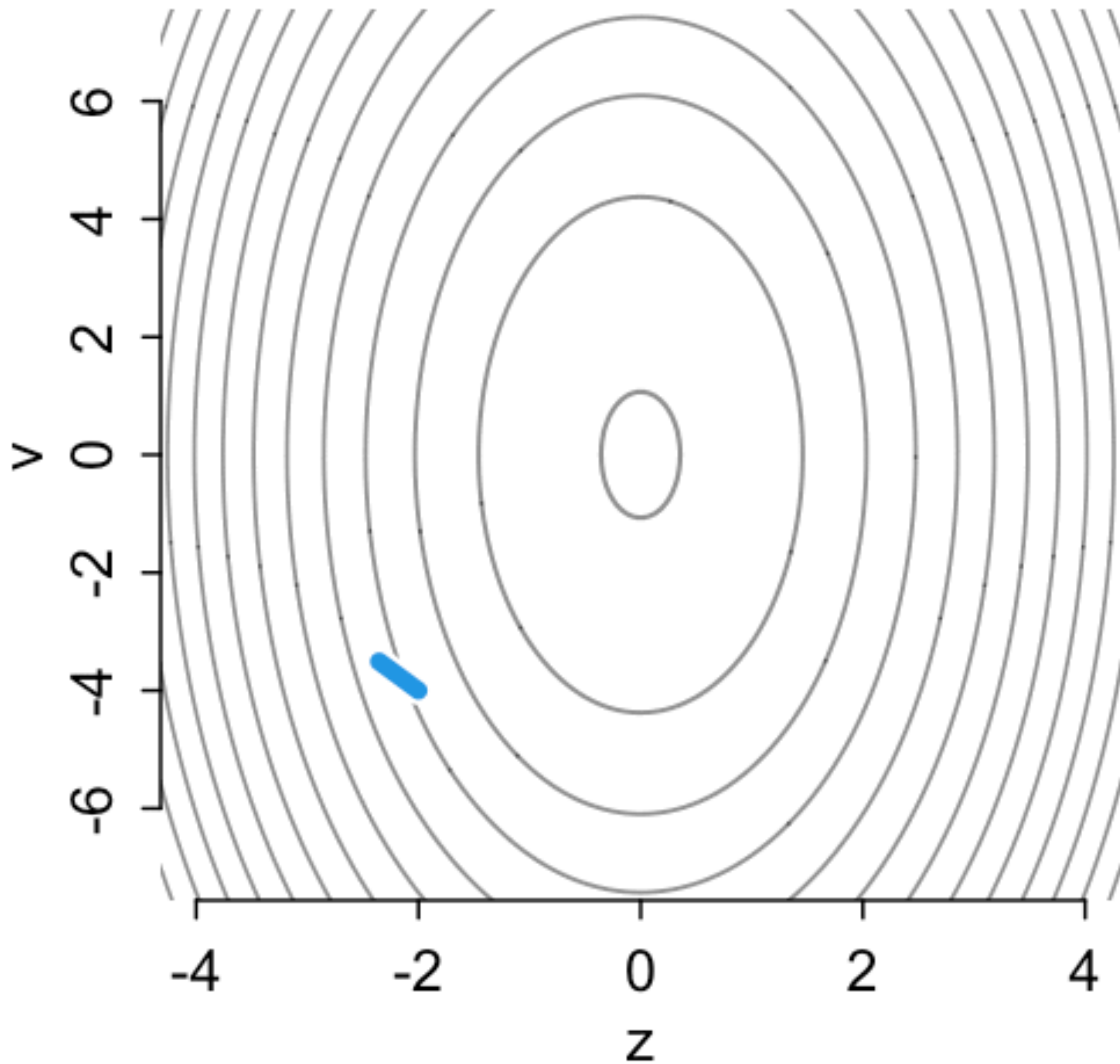
$$x \sim \text{Normal}(0, \exp(v))$$

“Non-centered”

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

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

$$x = z \exp(v)$$



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

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

$$x = z \exp(v)$$

```
m13.7 <- ulam(  
  alist(  
    v ~ normal(0,3),  
    x ~ normal(0,exp(v))  
  ), data=list(N=1) , chains=4 )
```

```
m13.7nc <- ulam(  
  alist(  
    v ~ normal(0,3),  
    z ~ normal(0,1),  
    gq> real[1]:x <<- z*exp(v)  
  ), data=list(N=1) , chains=4 )
```

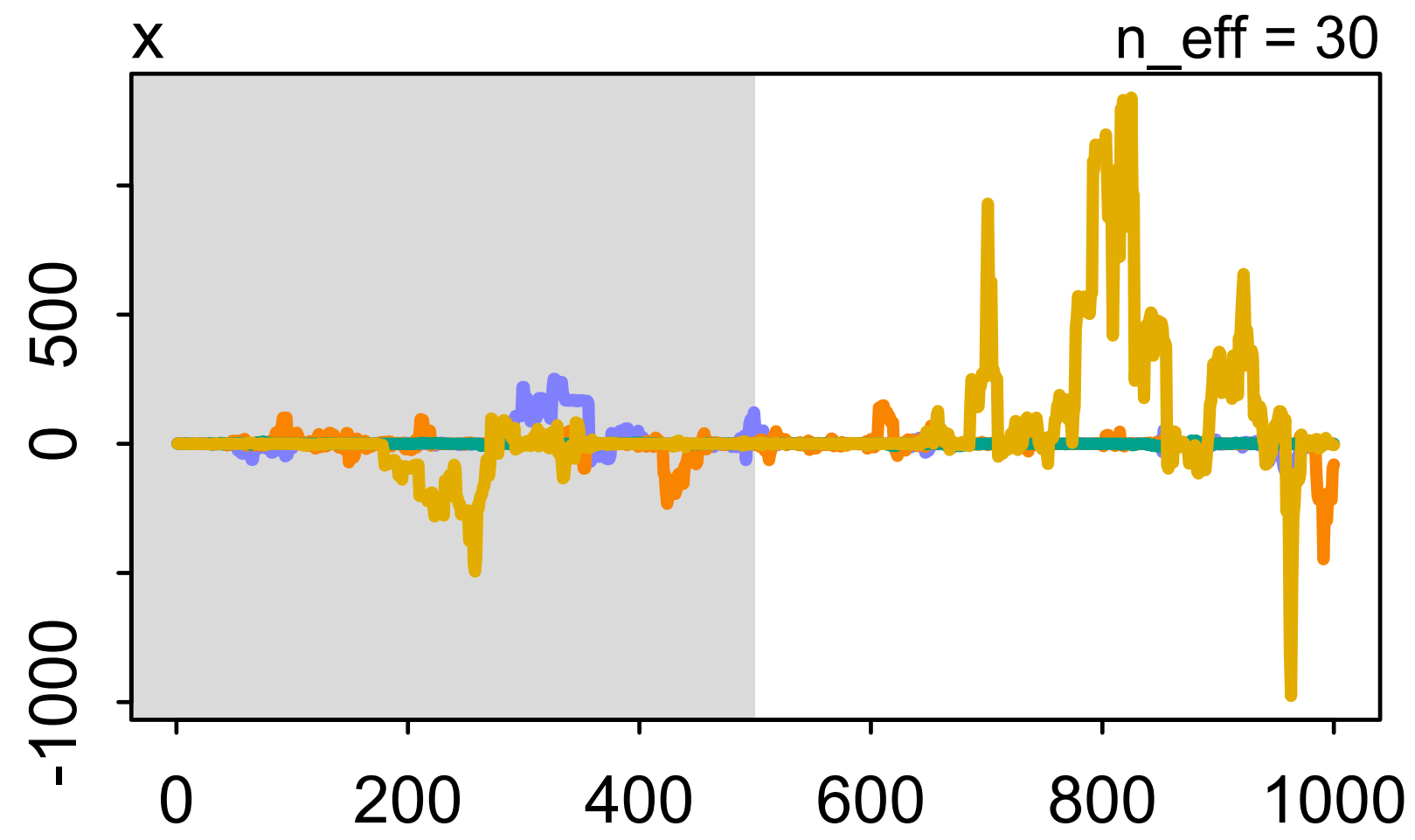
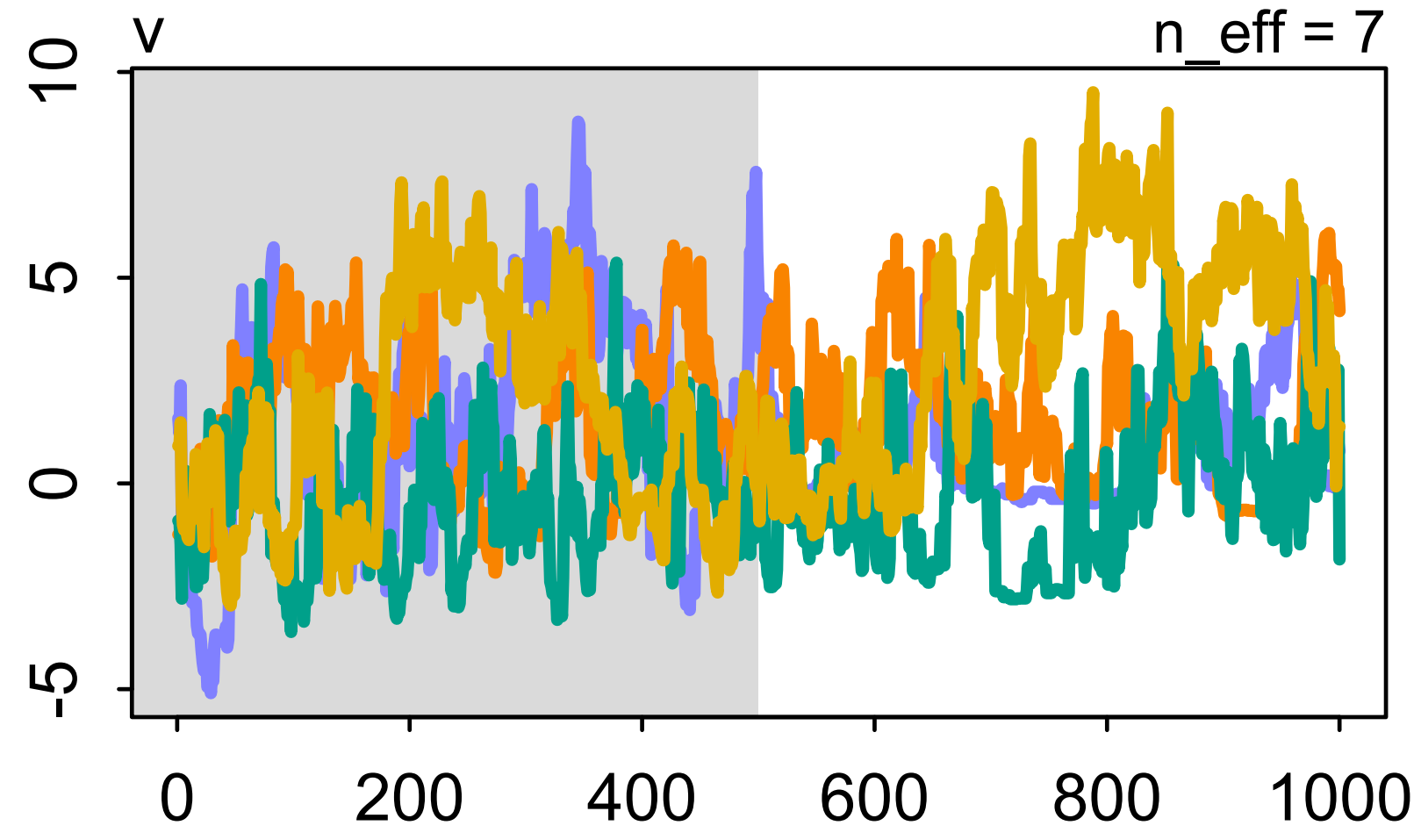
```
m13.7 <- ulam(  
  alist(  
    v ~ normal(0,3),  
    x ~ normal(0,exp(v))  
  ), data=list(N=1) , chains=4 )
```

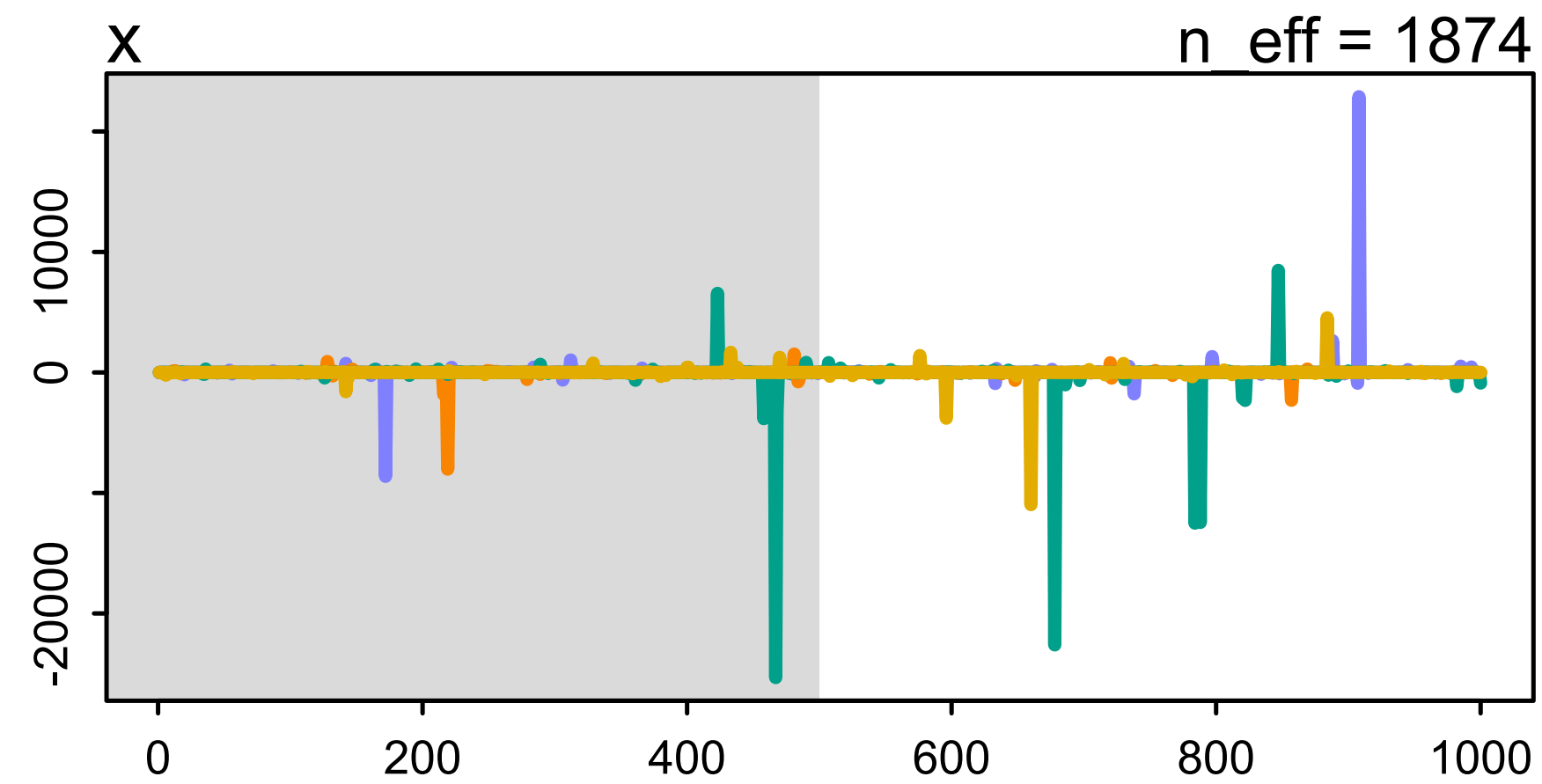
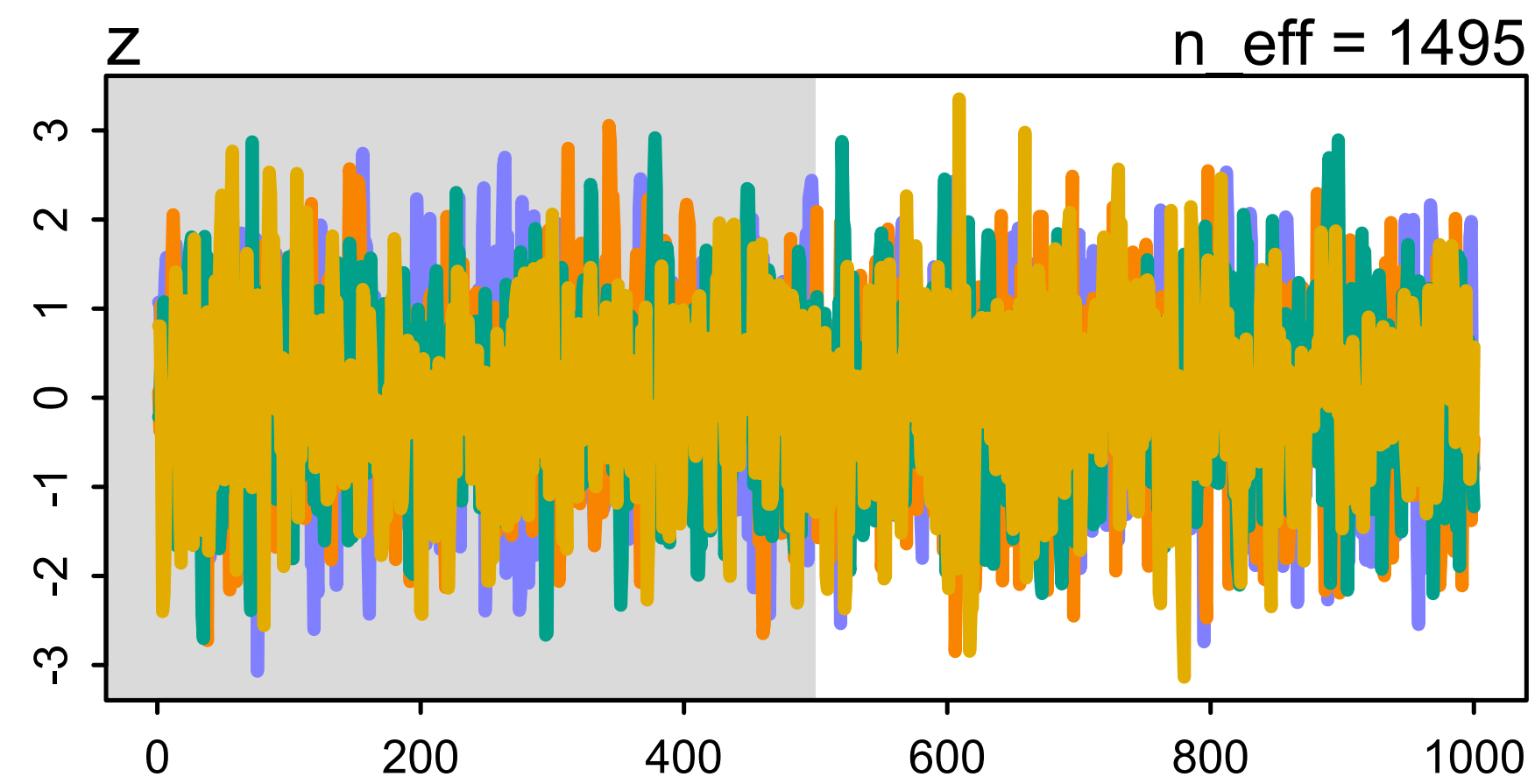
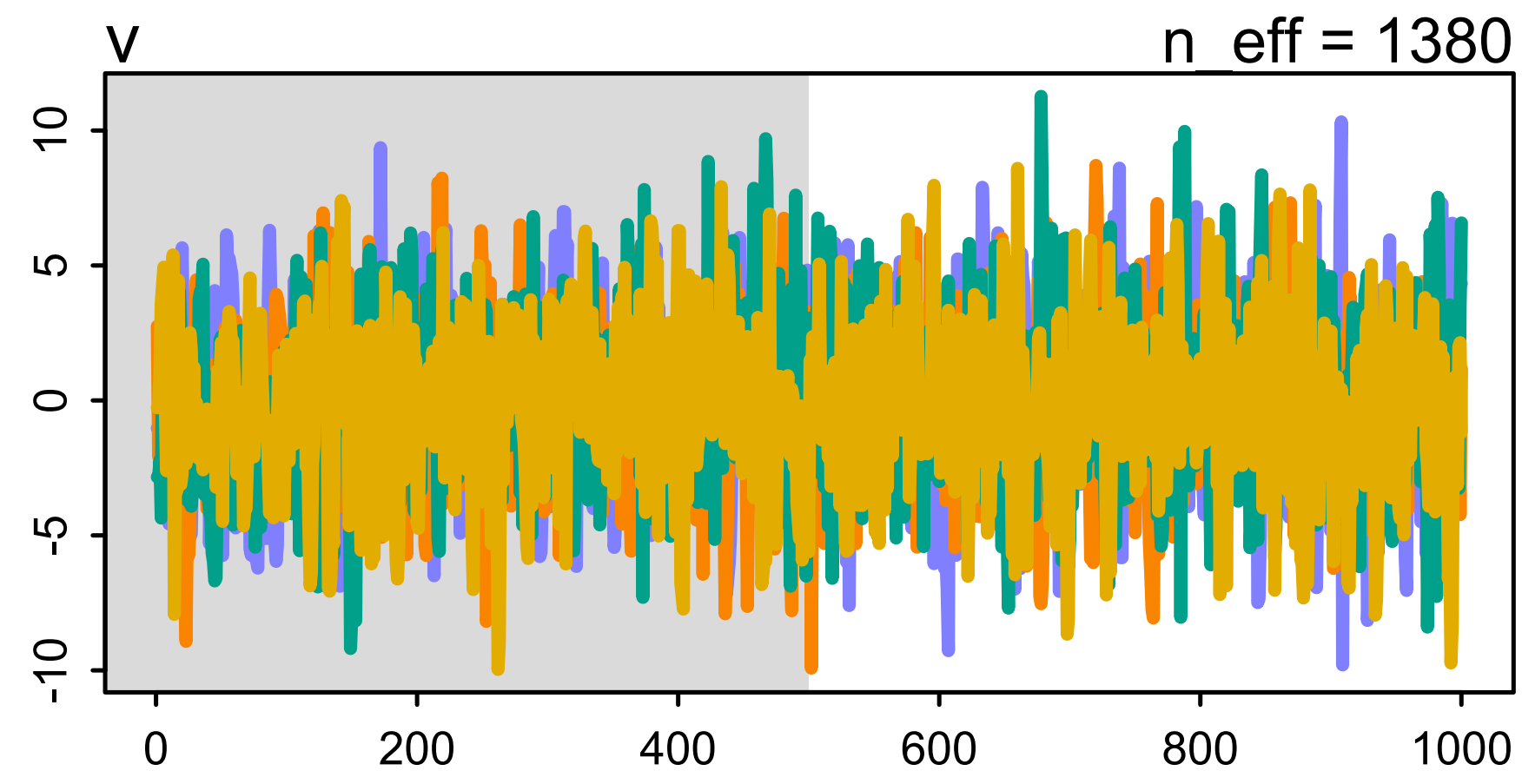
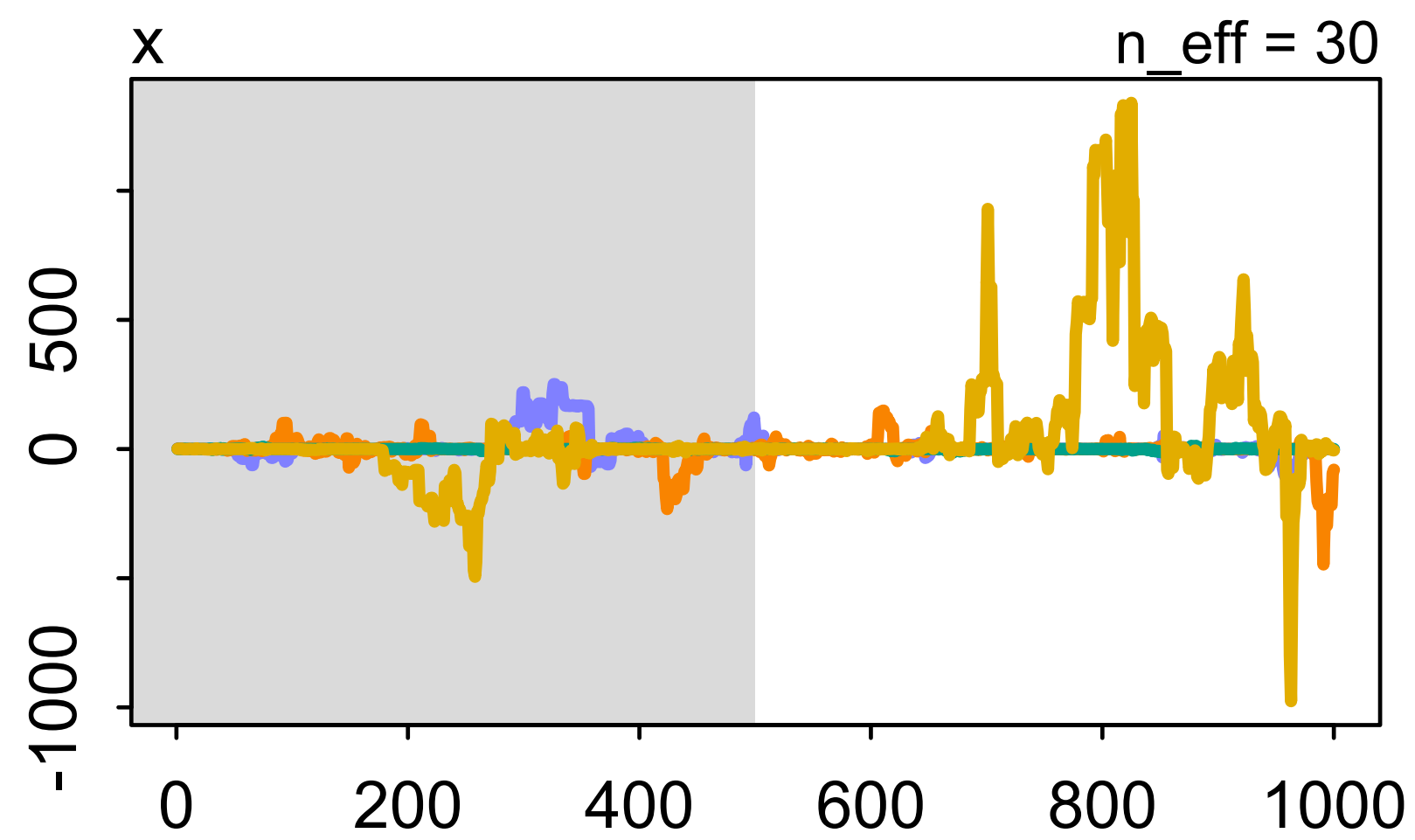
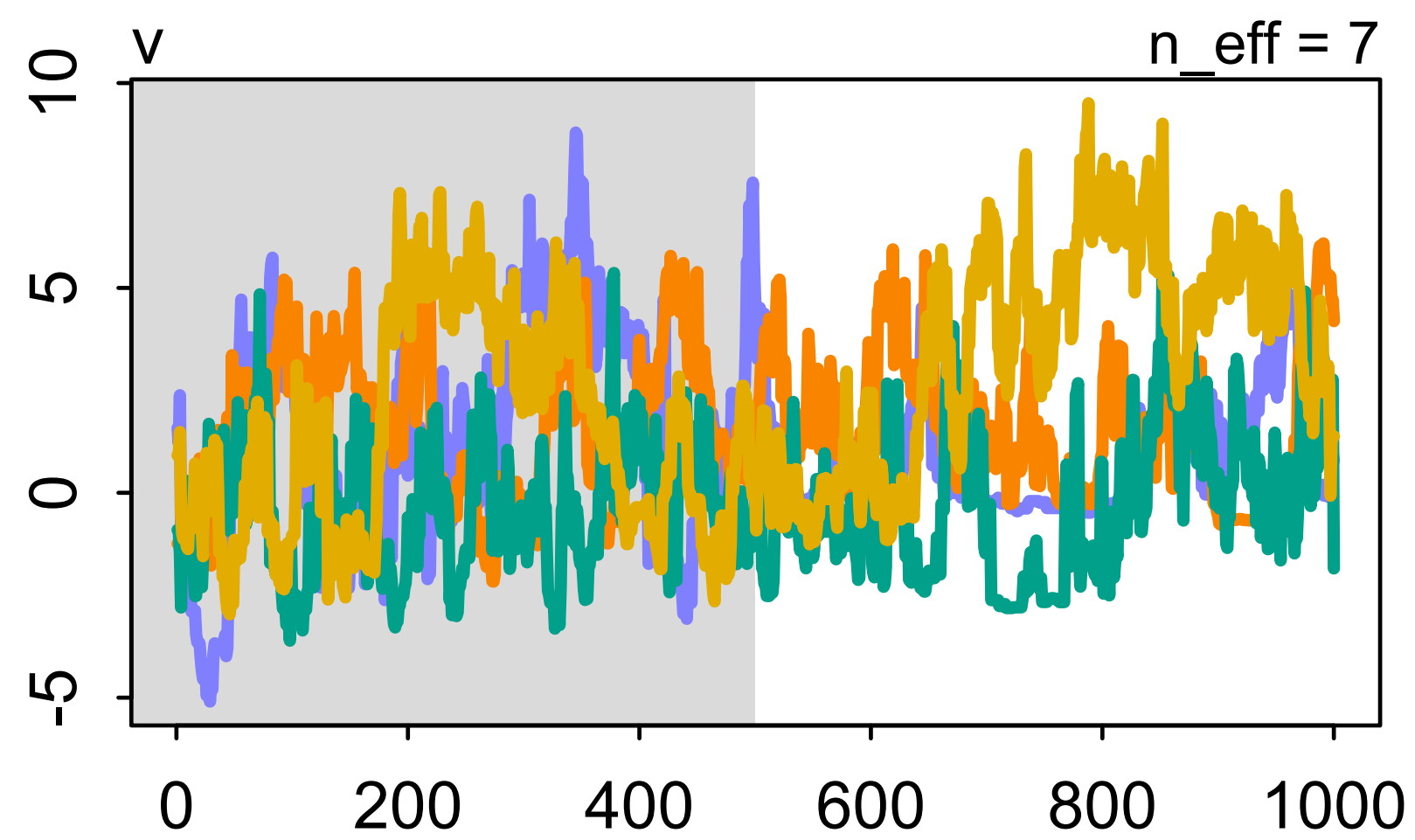
```
m13.7nc <- ulam(  
  alist(  
    v ~ normal(0,3),  
    z ~ normal(0,1),  
    gg> real[1]:x <<- z*exp(v)  
  ), data=list(N=1) , chains=4 )
```

Warning: 112 of 2000 (6.0%) transitions ended with a divergence.

```
> precis( m13.7 )  
      mean      sd  5.5%  94.5% n_eff Rhat4  
v  1.41  2.37 -1.84  5.93    7  1.46  
x 35.93 168.42 -21.15 258.86   30  1.19
```

```
> precis( m13.7nc )  
      mean      sd  5.5%  94.5% n_eff Rhat4  
v  -0.04  3.12 -5.17  4.84  1380    1  
z  -0.01  0.96 -1.60  1.51  1495    1  
x -19.34 899.98 -30.81 24.86  1874    1
```





Non-centered varying effects

“Centered”

$$S_i \sim \text{Binomial}(D_i, p_i)$$

$$\text{logit}(p_i) = \alpha_{T[i]}$$

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

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

$$\sigma \sim \text{Exponential}(1)$$



Non-centered varying effects

“Centered”

$$S_i \sim \text{Binomial}(D_i, p_i)$$

$$\text{logit}(p_i) = \alpha_{T[i]}$$

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

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

$$\sigma \sim \text{Exponential}(1)$$

“Non-centered”

$$S_i \sim \text{Binomial}(D_i, p_i)$$

$$\text{logit}(p_i) = \bar{\alpha} + z_{T[i]} \times \sigma$$

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

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

$$\sigma \sim \text{Exponential}(1)$$

Non-centered varying effects

“Centered”

$$S_i \sim \text{Binomial}(D_i, p_i)$$

$$\text{logit}(p_i) = \alpha_{T[i]}$$

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

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

$$\sigma \sim \text{Exponential}(1)$$

“Non-centered”

$$S_i \sim \text{Binomial}(D_i, p_i)$$

$$\text{logit}(p_i) = \bar{\alpha} + z_{T[i]} \times \sigma$$

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

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

$$\sigma \sim \text{Exponential}(1)$$

Non-centered chimpanzees

“Centered”

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

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

$$\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma_A)$$

$$\beta_{j,k} \sim \text{Normal}(0, \sigma_B)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

Non-centered chimpanzees

“Centered”

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

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

$$\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma_A)$$

$$\beta_{j,k} \sim \text{Normal}(0, \sigma_B)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

“Non-centered”

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

$$\text{logit}(p_i) = \bar{\alpha} + (z_{\alpha,A[i]})\sigma_A + (z_{\beta,T[i],B[i]})\sigma_B$$

$$z_{\alpha,j} \sim \text{Normal}(0, 1)$$

$$z_{\beta,j} \sim \text{Normal}(0, 1)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

Non-centered chimpanzees

“Centered”

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

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

$$\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma_A)$$

$$\beta_{j,k} \sim \text{Normal}(0, \sigma_B)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

“Non-centered”

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

$$\text{logit}(p_i) = \bar{\alpha} + (z_{\alpha,A[i]})\sigma_A + (z_{\beta,T[i],B[i]})\sigma_B$$

$$z_{\alpha,j} \sim \text{Normal}(0, 1)$$

$$z_{\beta,j} \sim \text{Normal}(0, 1)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

Non-centered chimpanzees

“Centered”

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

$$\text{logit}(p_i) = \beta_{T[i],B[i]} + \alpha_{A[i]}$$

$$\alpha_j \sim \text{Normal}(\bar{\alpha}, \sigma_A)$$

$$\beta_{j,k} \sim \text{Normal}(0, \sigma_B)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

“Non-centered”

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

$$\text{logit}(p_i) = \bar{\alpha} + (z_{\alpha,A[i]})\sigma_A + (z_{\beta,T[i],B[i]})\sigma_B$$

$$z_{\alpha,j} \sim \text{Normal}(0, 1)$$

$$z_{\beta,j} \sim \text{Normal}(0, 1)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0, 1.5)$$

```

mBT <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- b[T,B] + a[A],
    ## adaptive priors
    matrix[T,B]:b ~ dnorm( 0 , sigma_B ),
    a[A] ~ dnorm( a_bar , sigma_A ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1)
  ) , data=dat , chains=4 , cores=4 )

```

```

mBTnc <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- a_bar + z_a[A]*sigma_A + z_b[T,B]*sigma_B ,
    ## adaptive priors
    matrix[T,B]:z_b ~ dnorm( 0 , 1 ),
    z_a[A] ~ dnorm( 0 , 1 ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1),
    qq> vector[A]:a <<- a_bar + z_a*sigma_A,
    qq> matrix[T,B]:b <<- z_b*sigma_B
  ) , data=dat , chains=4 , cores=4 )

```

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

$$\text{logit}(p_i) = \bar{\alpha} + (z_{\alpha,A[i]})\sigma_A + (z_{\beta,T[i],B[i]})\sigma_B$$

$$z_{\alpha,j} \sim \text{Normal}(0,1)$$

$$z_{\beta,j} \sim \text{Normal}(0,1)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0,1.5)$$

```

mBT <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- b[T,B] + a[A],
    ## adaptive priors
    matrix[T,B]:b ~ dnorm( 0 , sigma_B ),
    a[A] ~ dnorm( a_bar , sigma_A ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1)
  ) , data=dat , chains=4 , cores=4 )

```

```

mBTnc <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- a_bar + z_a[A]*sigma_A + z_b[T,B]*sigma_B ,
    ## adaptive priors
    matrix[T,B]:z_b ~ dnorm( 0 , 1 ),
    z_a[A] ~ dnorm( 0 , 1 ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1),
    qq> vector[A]:a <<- a_bar + z_a*sigma_A,
    qq> matrix[T,B]:b <<- z_b*sigma_B
  ) , data=dat , chains=4 , cores=4 )

```

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

$$\text{logit}(p_i) = \bar{\alpha} + (z_{\alpha,A[i]})\sigma_A + (z_{\beta,T[i],B[i]})\sigma_B$$

$$z_{\alpha,j} \sim \text{Normal}(0,1)$$

$$z_{\beta,j} \sim \text{Normal}(0,1)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0,1.5)$$

```

mBT <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- b[T,B] + a[A],
    ## adaptive priors
    matrix[T,B]:b ~ dnorm( 0 , sigma_B ),
    a[A] ~ dnorm( a_bar , sigma_A ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1)
  ) , data=dat , chains=4 , cores=4 )

mBTnc <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- a_bar + z_a[A]*sigma_A + z_b[T,B]*sigma_B ,
    ## adaptive priors
    matrix[T,B]:z_b ~ dnorm( 0 , 1 ),
    z_a[A] ~ dnorm( 0 , 1 ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1),
    qq> vector[A]:a <<- a_bar + z_a*sigma_A,
    qq> matrix[T,B]:b <<- z_b*sigma_B
  ) , data=dat , chains=4 , cores=4 )

```

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

$$\text{logit}(p_i) = \bar{\alpha} + (z_{\alpha,A[i]})\sigma_A + (z_{\beta,T[i],B[i]})\sigma_B$$

$$z_{\alpha,j} \sim \text{Normal}(0,1)$$

$$z_{\beta,j} \sim \text{Normal}(0,1)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0,1.5)$$

```

mBT <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- b[T,B] + a[A],
    ## adaptive priors
    matrix[T,B]:b ~ dnorm( 0 , sigma_B ),
    a[A] ~ dnorm( a_bar , sigma_A ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1)
  ) , data=dat , chains=4 , cores=4 )

mBTnc <- ulam(
  alist(
    P ~ bernoulli( p ) ,
    logit(p) <- a_bar + z_a[A]*sigma_A + z_b[T,B]*sigma_B ,
    ## adaptive priors
    matrix[T,B]:z_b ~ dnorm( 0 , 1 ),
    z_a[A] ~ dnorm( 0 , 1 ),
    ## hyper-priors
    a_bar ~ dnorm( 0 , 1.5 ),
    sigma_A ~ dexp(1),
    sigma_B ~ dexp(1),
    gq> vector[A]:a <- a_bar + z_a*sigma_A,
    gq> matrix[T,B]:b <- z_b*sigma_B
  ) , data=dat , chains=4 , cores=4 )

```

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

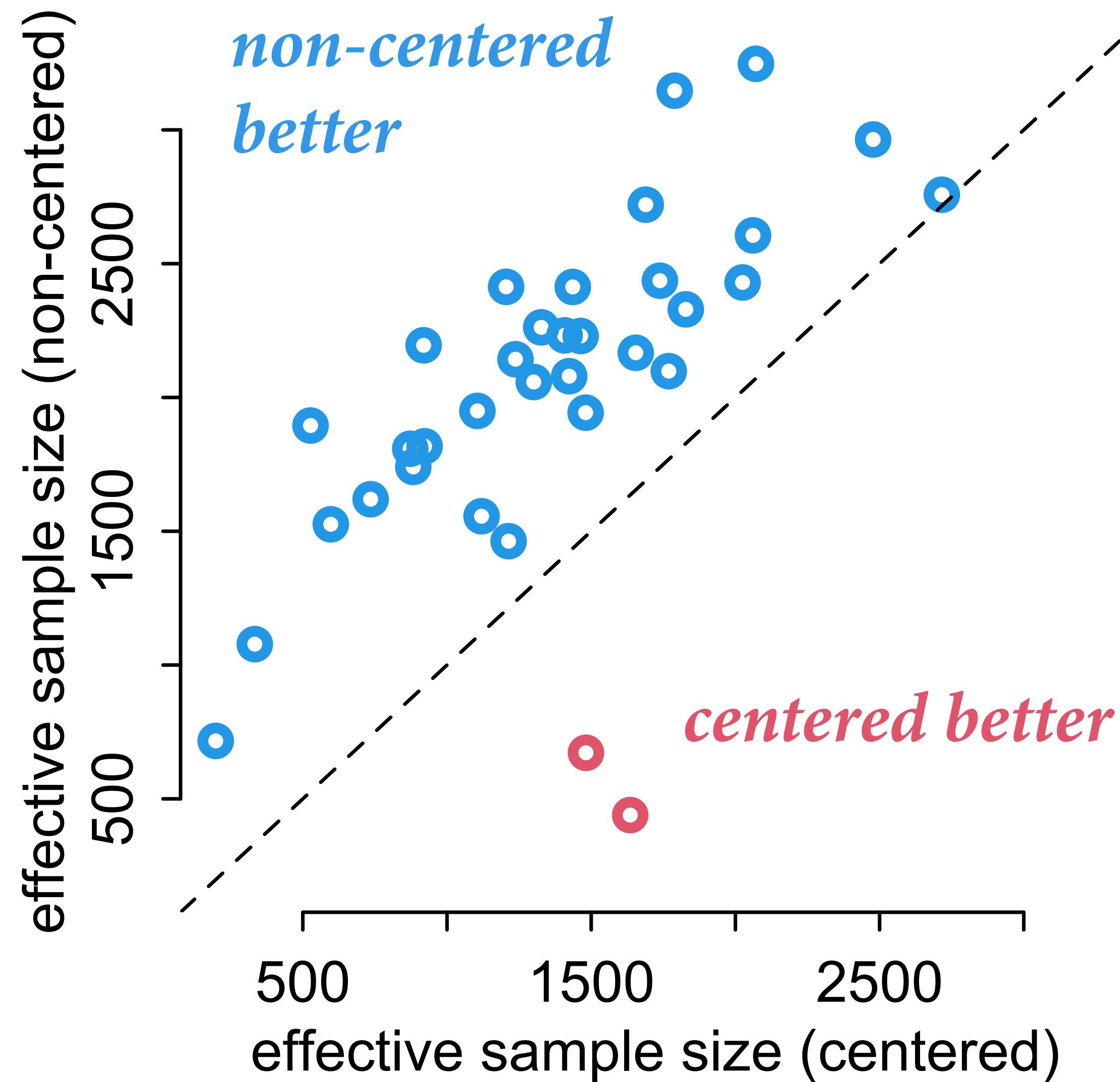
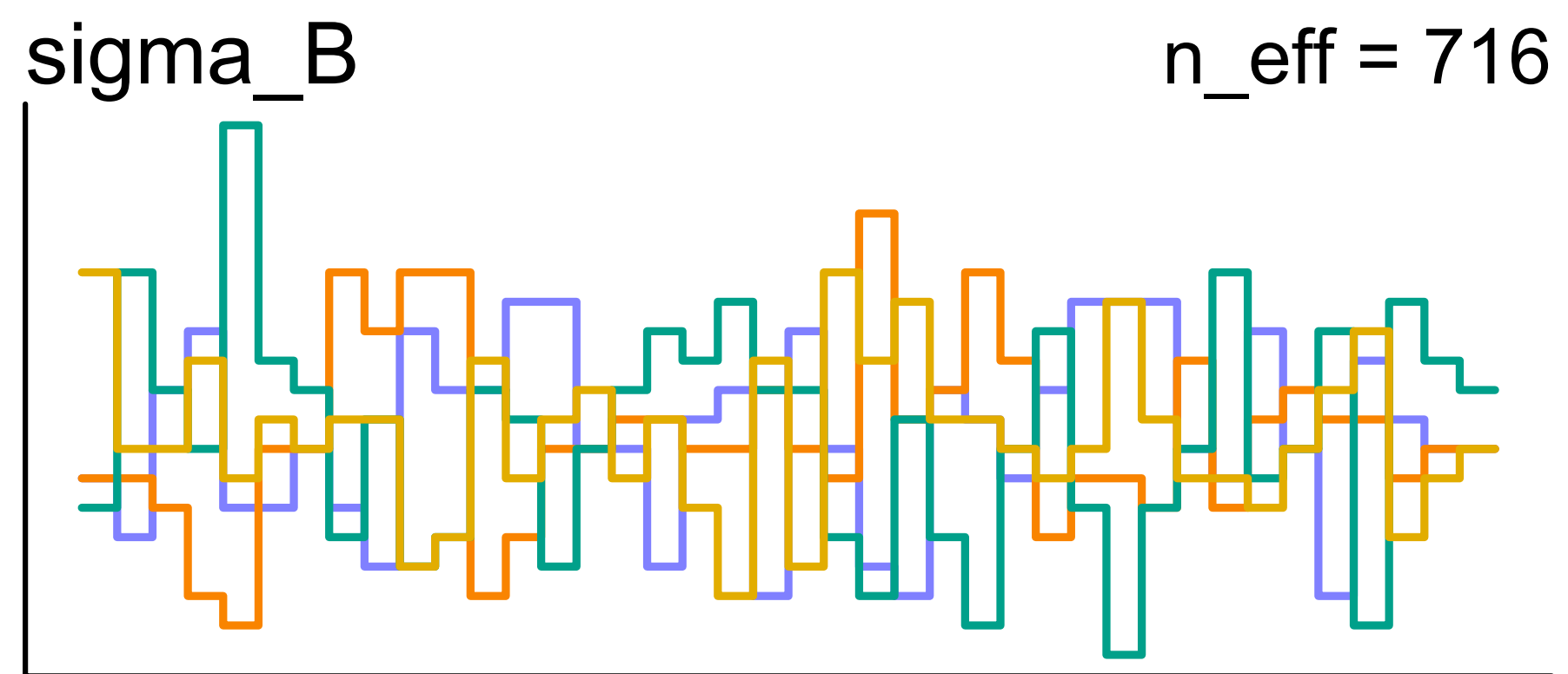
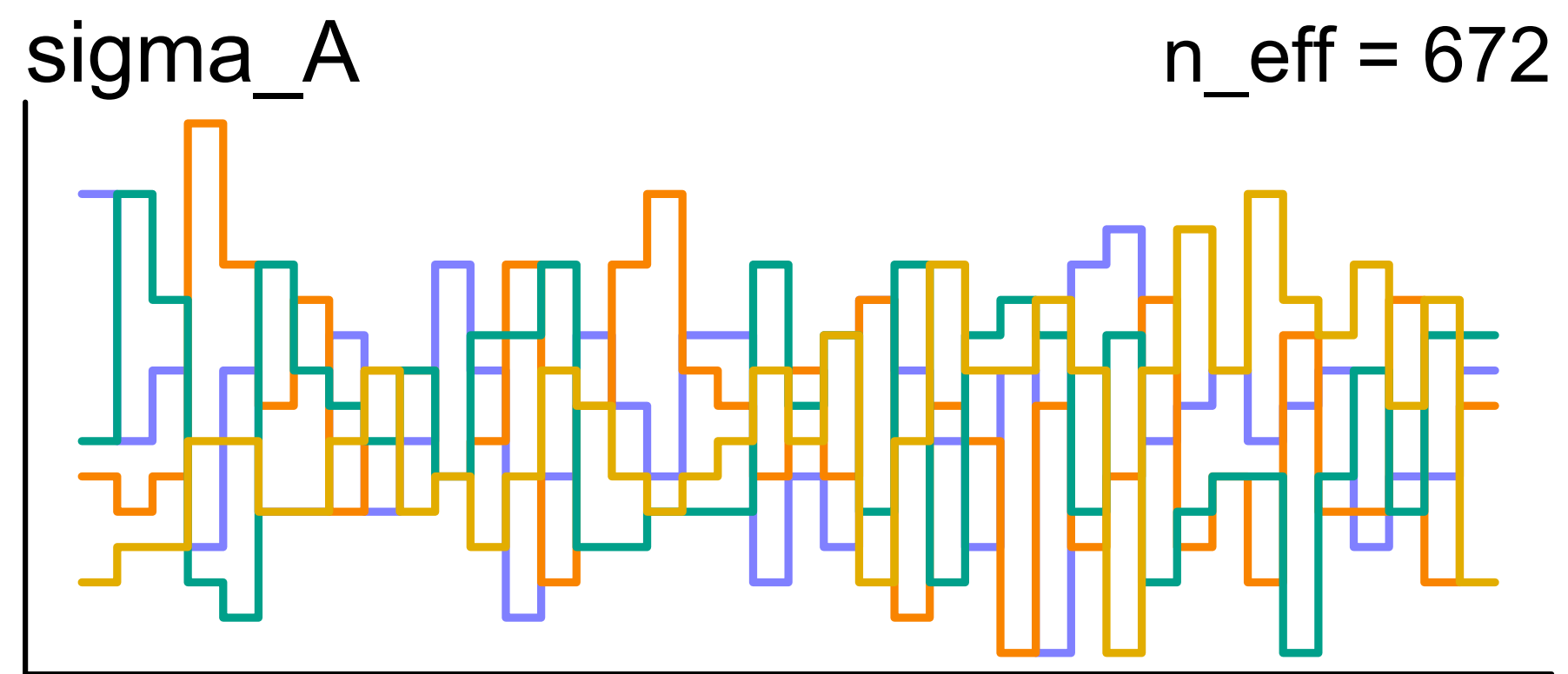
$$\text{logit}(p_i) = \bar{\alpha} + (z_{\alpha,A[i]})\sigma_A + (z_{\beta,T[i],B[i]})\sigma_B$$

$$z_{\alpha,j} \sim \text{Normal}(0,1)$$

$$z_{\beta,j} \sim \text{Normal}(0,1)$$

$$\sigma_A, \sigma_B \sim \text{Exponential}(1)$$

$$\bar{\alpha} \sim \text{Normal}(0,1.5)$$



Practical Solutions

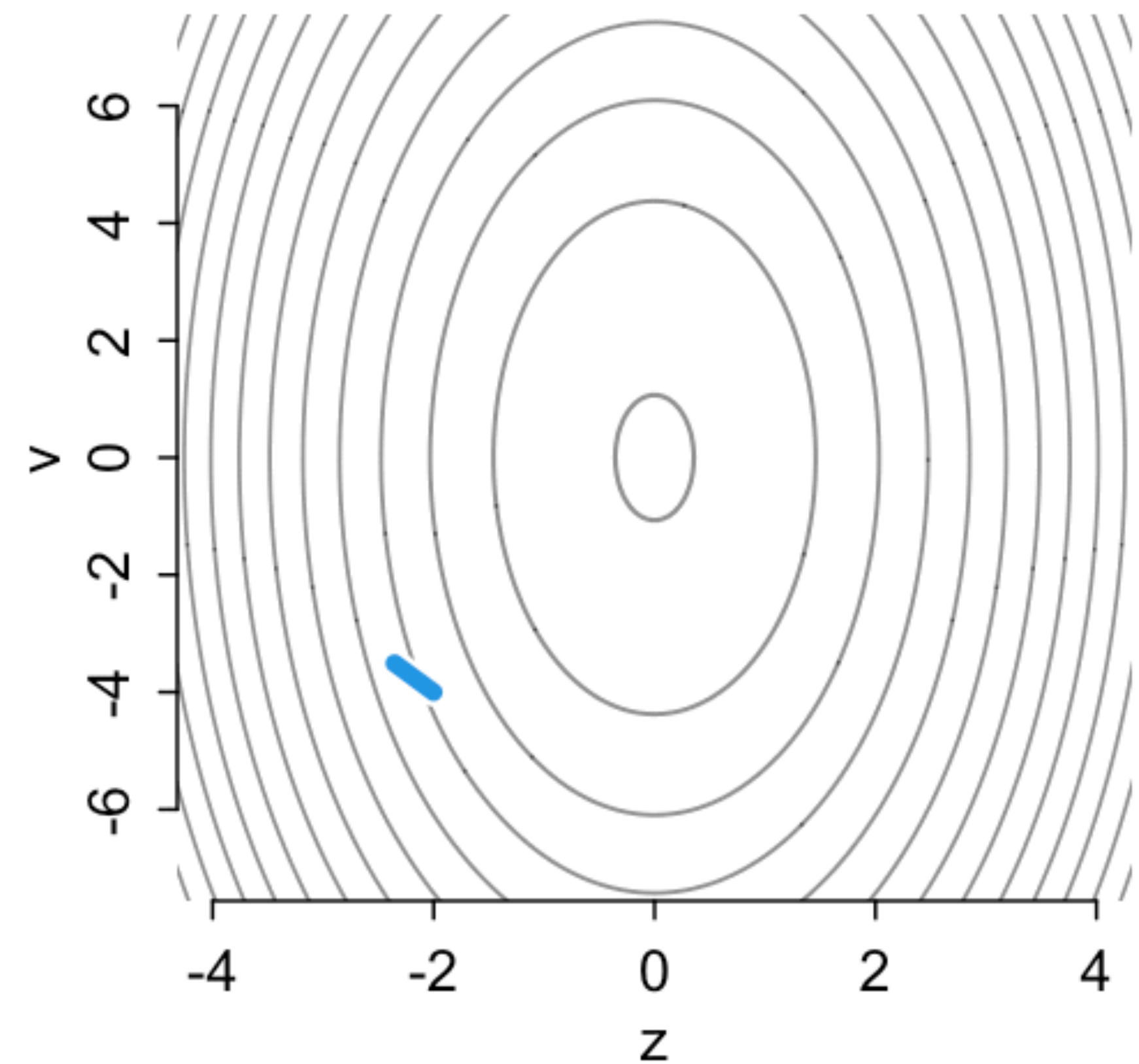
Research problems = technical problems

(1) Use **more than one** cluster type

(2) Calculate predictions

(3) Sample chains efficiently

Practice leads to mastery



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	Multilevel models & Gaussian processes	Chapter 14
Week 9	Measurement & Missingness	Chapter 15
Week 10	Generalized Linear Madness	Chapter 16

https://github.com/rmcelreath/stat_rethinking_2022

