

# prosocial

*Chris Parrish*

*June 9, 2016*

## Contents

<b>prosocial</b>	<b>1</b>
data	1
simplest model	2
estimate for $a$	2
second model	2
third model	3
compare the models	4
relative vs. absolute effect	4
predictions	5
stan	7
actors	10
check actor 2	13
check actor 1	14
aggregated binomial	15

prosocial

reference: McElreath, Statistical Rethinking, chap 10, p.291

```
library(rethinking)
library(ggplot2)
```

## prosocial

### data

```
## R code 10.1
data(chimpanzees)
?chimpanzees
d <- chimpanzees
str(d)

## 'data.frame':   504 obs. of  8 variables:
## $ actor      : int  1 1 1 1 1 1 1 1 1 1 ...
## $ recipient  : int  NA NA NA NA NA NA NA NA NA NA ...
## $ condition  : int  0 0 0 0 0 0 0 0 0 0 ...
## $ block      : int  1 1 1 1 1 1 2 2 2 2 ...
## $ trial      : int  2 4 6 8 10 12 14 16 18 20 ...
## $ prosoc_left : int  0 0 1 0 1 1 1 1 0 0 ...
## $ chose_prosoc: int  1 0 0 1 1 1 0 0 1 1 ...
## $ pulled_left : int  0 1 0 0 1 1 0 0 0 0 ...
```

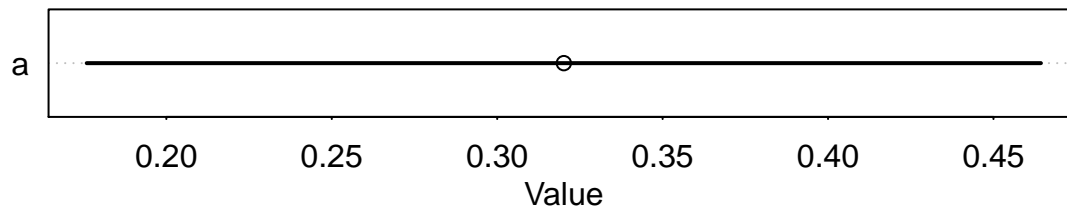
## simplest model

$$\begin{aligned}L_i &\sim \text{Binomial}(1, p_i) \\ \text{logit}(p_i) &= \alpha \\ \alpha &\sim \text{Normal}(0, 10)\end{aligned}$$

```
## R code 10.2
m10.1 <- map(
  alist(
    pulled_left ~ dbinom( 1 , p ) ,
    logit(p) <- a ,
    a ~ dnorm(0,10)
  ) ,
  data=d )
precis(m10.1)
```

```
## Mean StdDev 5.5% 94.5%
## a 0.32 0.09 0.18 0.46
```

```
par(mfrow=c(1, 1))
plot(precis(m10.1))
```



estimate for  $a$

```
logistic(0.32)
```

```
## [1] 0.5793243
```

```
## R code 10.3
```

```
logistic( c(0.18,0.46) )
```

```
## [1] 0.5448789 0.6130142
```

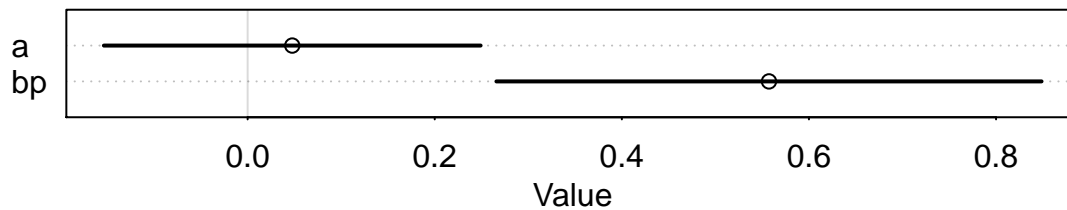
## second model

$$\begin{aligned}L_i &\sim \text{Binomial}(1, p_i) \\ \text{logit}(p_i) &= \alpha + \beta_P P_i \\ \alpha &\sim \text{Normal}(0, 10) \\ \beta_P &\sim \text{Normal}(0, 10)\end{aligned}$$

```
## R code 10.4
m10.2 <- map(
  alist(
    pulled_left ~ dbinom( 1 , p ) ,
    logit(p) <- a + bp*prosoc_left ,
    a ~ dnorm(0,10) ,
    bp ~ dnorm(0,10)
  ) ,
  data=d )
precis(m10.2)
```

```
##      Mean StdDev  5.5% 94.5%
## a   0.05  0.13 -0.15  0.25
## bp  0.56  0.18  0.27  0.85
```

```
plot(precis(m10.2))
```



### third model

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

$$\text{logit}(p_i) = \alpha + (\beta_P + \beta_{PC}C_i)P_i$$

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

$$\beta_P \sim \text{Normal}(0, 10)$$

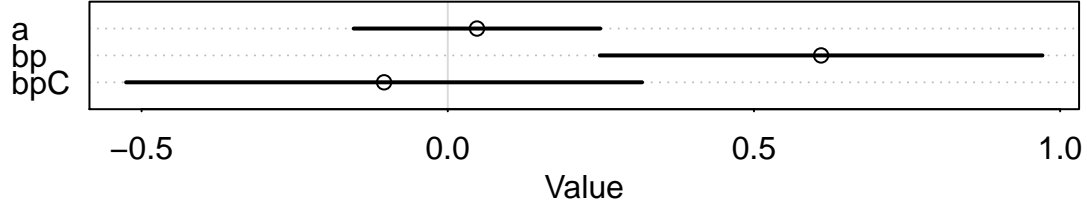
$$\beta_{PC} \sim \text{Normal}(0, 10)$$

```
m10.3 <- map(
  alist(
    pulled_left ~ dbinom( 1 , p ) ,
    logit(p) <- a + (bp + bpC*condition)*prosoc_left ,
    a ~ dnorm(0,10) ,
    bp ~ dnorm(0,10) ,
    bpC ~ dnorm(0,10)
  ) ,
  data=d )
```

```
## R code 10.6
precis(m10.3)
```

```
##      Mean StdDev  5.5% 94.5%
## a   0.05  0.13 -0.15  0.25
## bp  0.61  0.23  0.25  0.97
## bpC -0.10  0.26 -0.53  0.32
```

```
plot(precis(m10.3))
```

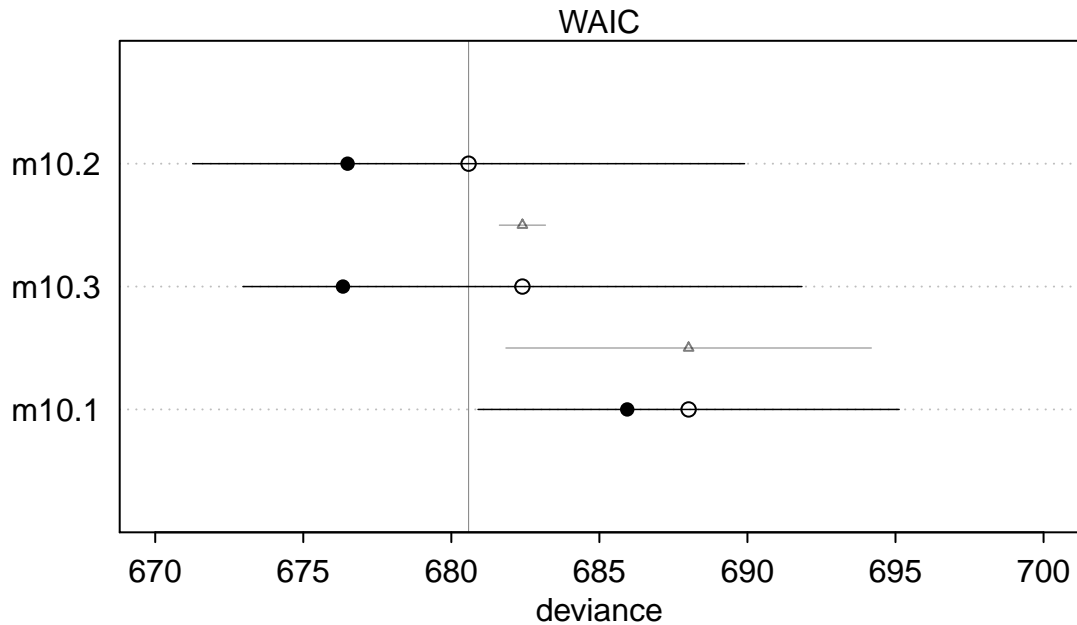


### compare the models

```
## R code 10.5  
compare( m10.1 , m10.2 , m10.3 )
```

```
##           WAIC pWAIC dWAIC weight   SE  dSE  
## m10.2 680.6    2.1   0.0   0.69 9.31  NA  
## m10.3 682.3    3.0   1.7   0.30 9.31 0.63  
## m10.1 687.9    1.0   7.3   0.02 7.23 6.16
```

```
plot(compare( m10.1 , m10.2 , m10.3 ),  
      xlim=c(670, 700))
```



### relative vs. absolute effect

```
## R code 10.7  
exp(0.61)
```

```
## [1] 1.840431
```

```
## R code 10.8  
logistic( 4 )
```

```
## [1] 0.9820138
```

```
## R code 10.9
logistic( 4 + 0.61 )
```

```
## [1] 0.9901462
```

## predictions

```
## R code 10.10
# dummy data for predictions across treatments
d.pred <- data.frame(
  prosoc_left = c(0,1,0,1), # right/left/right/left
  condition = c(0,0,1,1) # control/control/partner/partner
)

# build prediction ensemble
chimp.ensemble <- ensemble( m10.1 , m10.2 , m10.3 , data=d.pred )
```

```
## Constructing posterior predictions
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
## Constructing posterior predictions
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
## Constructing posterior predictions
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```

str(chimp.ensemble)

## List of 2
## $ link: num [1:1000, 1:4] 0.578 0.556 0.635 0.6 0.593 ...
## $ sim : int [1:1000, 1:4] 1 1 0 1 0 1 1 0 1 1 ...
## - attr(*, "weights")= Named num [1:3] 0.0157 0.7086 0.2757
## ..- attr(*, "names")= chr [1:3] "m10.1" "m10.2" "m10.3"
## - attr(*, "indices")= num [1:3, 1:2] 1 17 726 16 725 1000
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:3] "m10.1" "m10.2" "m10.3"
## .. ..$ : chr [1:2] "idx_start" "idx_end"
## - attr(*, "ictab")=Formal class 'compareIC' [package "rethinking"] with 2 slots
## .. ..@ output:'data.frame': 3 obs. of 3 variables:
## .. .. ..$ IC : num [1:3] 688 680 682
## .. .. ..$ dIC : num [1:3] 7.62 0 1.89
## .. .. ..$ weight: num [1:3] 0.0157 0.7086 0.2757
## .. ..@ dSE : logi [1:3, 1:3] NA NA NA NA NA NA ...

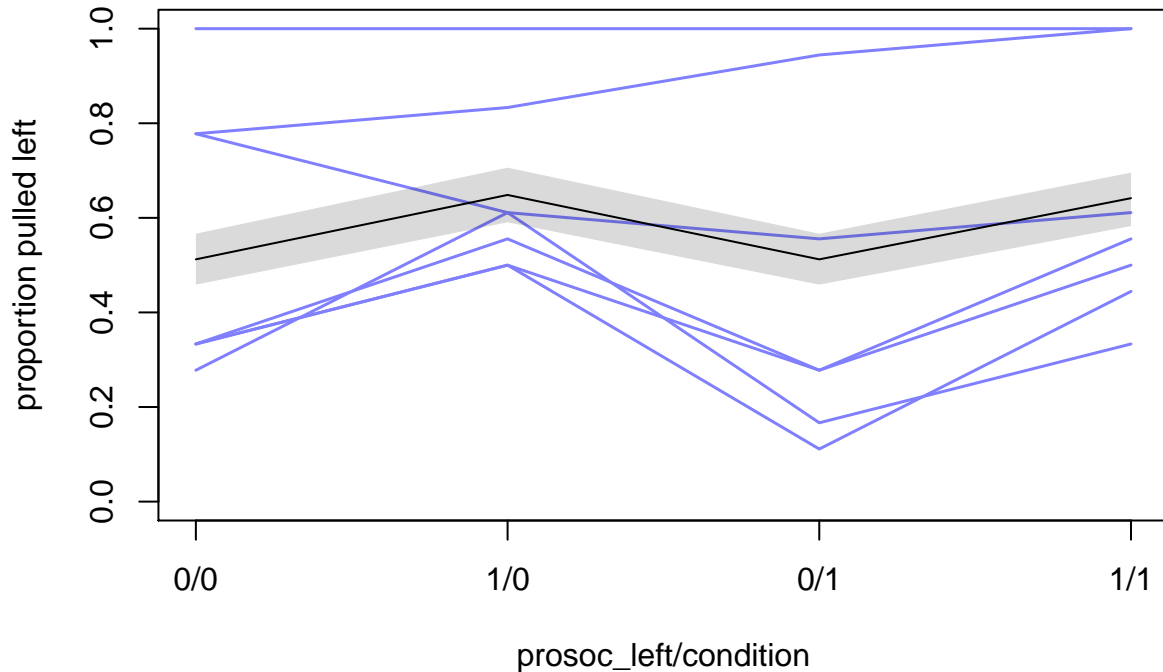
# summarize
pred.p <- apply( chimp.ensemble$link , 2 , mean )
pred.p.PI <- apply( chimp.ensemble$link , 2 , PI )

## R code 10.11
# empty plot frame with good axes
plot( 0 , 0 , type="n" , xlab="prosoc_left/condition" ,
      ylab="proportion pulled left" , ylim=c(0,1) , xaxt="n" ,
      xlim=c(1,4) )
axis( 1 , at=1:4 , labels=c("0/0","1/0","0/1","1/1") )

# plot raw data, one trend for each of 7 individual chimpanzees
# will use by() here; see Overthinking box for explanation
p <- by( d$pulled_left ,
        list(d$prosoc_left,d$condition,d$factor) , mean )
for ( chimp in 1:7 )
  lines( 1:4 , as.vector(p[,chimp]) , col=rangi2 , lwd=1.5 )

# now superimpose posterior predictions
lines( 1:4 , pred.p )
shade( pred.p.PI , 1:4 )

```



stan

```
## R code 10.12
# clean NAs from the data
d2 <- d
d2$recipient <- NULL

# re-use map fit to get the formula
m10.3stan <- map2stan( m10.3 , data=d2 , iter=1e4 , warmup=1000 )
```

```
##
## SAMPLING FOR MODEL 'pulled_left ~ dbinom(1, p)' NOW (CHAIN 1).
##
## Chain 1, Iteration: 1 / 10000 [ 0%] (Warmup)
## Chain 1, Iteration: 1000 / 10000 [ 10%] (Warmup)
## Chain 1, Iteration: 1001 / 10000 [ 10%] (Sampling)
## Chain 1, Iteration: 2000 / 10000 [ 20%] (Sampling)
## Chain 1, Iteration: 3000 / 10000 [ 30%] (Sampling)
## Chain 1, Iteration: 4000 / 10000 [ 40%] (Sampling)
## Chain 1, Iteration: 5000 / 10000 [ 50%] (Sampling)
## Chain 1, Iteration: 6000 / 10000 [ 60%] (Sampling)
## Chain 1, Iteration: 7000 / 10000 [ 70%] (Sampling)
## Chain 1, Iteration: 8000 / 10000 [ 80%] (Sampling)
## Chain 1, Iteration: 9000 / 10000 [ 90%] (Sampling)
## Chain 1, Iteration: 10000 / 10000 [100%] (Sampling)
## Elapsed Time: 0.47835 seconds (Warm-up)
## 4.1275 seconds (Sampling)
## 4.60585 seconds (Total)
##
##
## SAMPLING FOR MODEL 'pulled_left ~ dbinom(1, p)' NOW (CHAIN 1).
```

```

## WARNING: No variance estimation is
##           performed for num_warmup < 20
##
##
## Chain 1, Iteration: 1 / 1 [100%] (Sampling)
## Elapsed Time: 3e-06 seconds (Warm-up)
##           0.000201 seconds (Sampling)
##           0.000204 seconds (Total)

```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```

## [ 900 / 9000 ]
## [ 1800 / 9000 ]
## [ 2700 / 9000 ]
## [ 3600 / 9000 ]
## [ 4500 / 9000 ]
## [ 5400 / 9000 ]
## [ 6300 / 9000 ]
## [ 7200 / 9000 ]
## [ 8100 / 9000 ]
## [ 9000 / 9000 ]

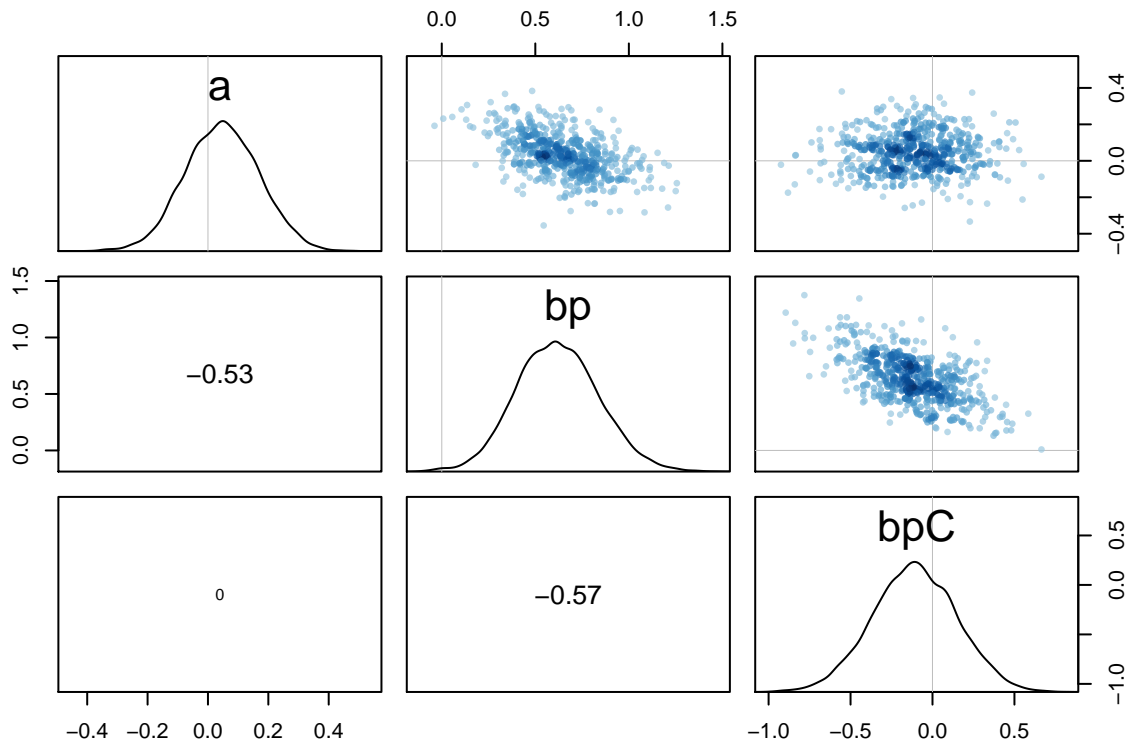
```

```
precis(m10.3stan)
```

	Mean	StdDev	lower 0.89	upper 0.89	n_eff	Rhat
## a	0.05	0.12	-0.14	0.25	4287	1
## bp	0.62	0.22	0.27	0.97	3882	1
## bpC	-0.11	0.26	-0.50	0.32	4097	1

```
## R code 10.13
```

```
pairs(m10.3stan)
```





```
show(m10.3stan)
```

```
## map2stan model fit
## 9000 samples from 1 chain
##
## Formula:
## pulled_left ~ dbinom(1, p)
## logit(p) <- a + (bp + bpC * condition) * prosoc_left
## a ~ dnorm(0, 10)
## bp ~ dnorm(0, 10)
## bpC ~ dnorm(0, 10)
##
## Log-likelihood at expected values: -338.17
## Deviance: 676.34
## DIC: 682.29
## Effective number of parameters (pD): 2.98
##
## WAIC (SE): 682.34 (9.5)
## pWAIC: 3
```

```
stancode(m10.3stan)
```

```
## data{
##   int<lower=1> N;
##   int pulled_left[N];
##   int condition[N];
##   int prosoc_left[N];
## }
## parameters{
##   real a;
##   real bp;
##   real bpC;
## }
## model{
##   vector[N] p;
##   bpC ~ normal( 0 , 10 );
##   bp ~ normal( 0 , 10 );
##   a ~ normal( 0 , 10 );
##   for ( i in 1:N ) {
##     p[i] = a + (bp + bpC * condition[i]) * prosoc_left[i];
##   }
##   pulled_left ~ binomial_logit( 1 , p );
## }
## generated quantities{
##   vector[N] p;
##   real dev;
##   dev = 0;
##   for ( i in 1:N ) {
##     p[i] = a + (bp + bpC * condition[i]) * prosoc_left[i];
##   }
##   dev = dev + (-2)*binomial_logit_lpmf( pulled_left | 1 , p );
## }
```

## actors

Model individual variation

$$\begin{aligned}L_i &\sim \text{Binomial}(1, p_i) \\ \text{logit}(p_i) &= \alpha_{\text{ACTOR}[i]} + (\beta_P + \beta_{PC}C_i)P_i \\ \alpha_{\text{ACTOR}[i]} &\sim \text{Normal}(0, 10) \\ \beta_P &\sim \text{Normal}(0, 10) \\ \beta_{PC} &\sim \text{Normal}(0, 10)\end{aligned}$$

```
## R code 10.14
m10.4 <- map2stan(
  alist(
    pulled_left ~ dbinom( 1 , p ) ,
    logit(p) <- a[actor] + (bp + bpC * condition) * prosoc_left ,
    a[actor] ~ dnorm(0,10),
    bp ~ dnorm(0,10),
    bpC ~ dnorm(0,10)
  ) ,
  data=d2 , chains=2 , iter=2500 , warmup=500 )
```

```
##
## SAMPLING FOR MODEL 'pulled_left ~ dbinom(1, p)' NOW (CHAIN 1).
##
## Chain 1, Iteration:    1 / 2500 [ 0%] (Warmup)
## Chain 1, Iteration:  250 / 2500 [ 10%] (Warmup)
## Chain 1, Iteration:  500 / 2500 [ 20%] (Warmup)
## Chain 1, Iteration:  501 / 2500 [ 20%] (Sampling)
## Chain 1, Iteration:  750 / 2500 [ 30%] (Sampling)
## Chain 1, Iteration: 1000 / 2500 [ 40%] (Sampling)
## Chain 1, Iteration: 1250 / 2500 [ 50%] (Sampling)
## Chain 1, Iteration: 1500 / 2500 [ 60%] (Sampling)
## Chain 1, Iteration: 1750 / 2500 [ 70%] (Sampling)
## Chain 1, Iteration: 2000 / 2500 [ 80%] (Sampling)
## Chain 1, Iteration: 2250 / 2500 [ 90%] (Sampling)
## Chain 1, Iteration: 2500 / 2500 [100%] (Sampling)
## Elapsed Time: 0.390412 seconds (Warm-up)
##                1.32024 seconds (Sampling)
##                1.71065 seconds (Total)
##
##
## SAMPLING FOR MODEL 'pulled_left ~ dbinom(1, p)' NOW (CHAIN 2).
##
## Chain 2, Iteration:    1 / 2500 [ 0%] (Warmup)
## Chain 2, Iteration:  250 / 2500 [ 10%] (Warmup)
## Chain 2, Iteration:  500 / 2500 [ 20%] (Warmup)
## Chain 2, Iteration:  501 / 2500 [ 20%] (Sampling)
## Chain 2, Iteration:  750 / 2500 [ 30%] (Sampling)
## Chain 2, Iteration: 1000 / 2500 [ 40%] (Sampling)
## Chain 2, Iteration: 1250 / 2500 [ 50%] (Sampling)
## Chain 2, Iteration: 1500 / 2500 [ 60%] (Sampling)
```

```

## Chain 2, Iteration: 1750 / 2500 [ 70%] (Sampling)
## Chain 2, Iteration: 2000 / 2500 [ 80%] (Sampling)
## Chain 2, Iteration: 2250 / 2500 [ 90%] (Sampling)
## Chain 2, Iteration: 2500 / 2500 [100%] (Sampling)
## Elapsed Time: 0.403269 seconds (Warm-up)
##           1.49396 seconds (Sampling)
##           1.89723 seconds (Total)
##
##
## SAMPLING FOR MODEL 'pulled_left ~ dbinom(1, p)' NOW (CHAIN 1).
## WARNING: No variance estimation is
##           performed for num_warmup < 20
##
##
## Chain 1, Iteration: 1 / 1 [100%] (Sampling)
## Elapsed Time: 6e-06 seconds (Warm-up)
##           0.000464 seconds (Sampling)
##           0.00047 seconds (Total)

## Computing WAIC

## Constructing posterior predictions

## [ 400 / 4000 ]
[ 800 / 4000 ]
[ 1200 / 4000 ]
[ 1600 / 4000 ]
[ 2000 / 4000 ]
[ 2400 / 4000 ]
[ 2800 / 4000 ]
[ 3200 / 4000 ]
[ 3600 / 4000 ]
[ 4000 / 4000 ]

## Warning in map2stan(alist(pulled_left ~ dbinom(1, p), logit(p) <- a[actor] + : There were 25 divergent
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.

show(m10.4)

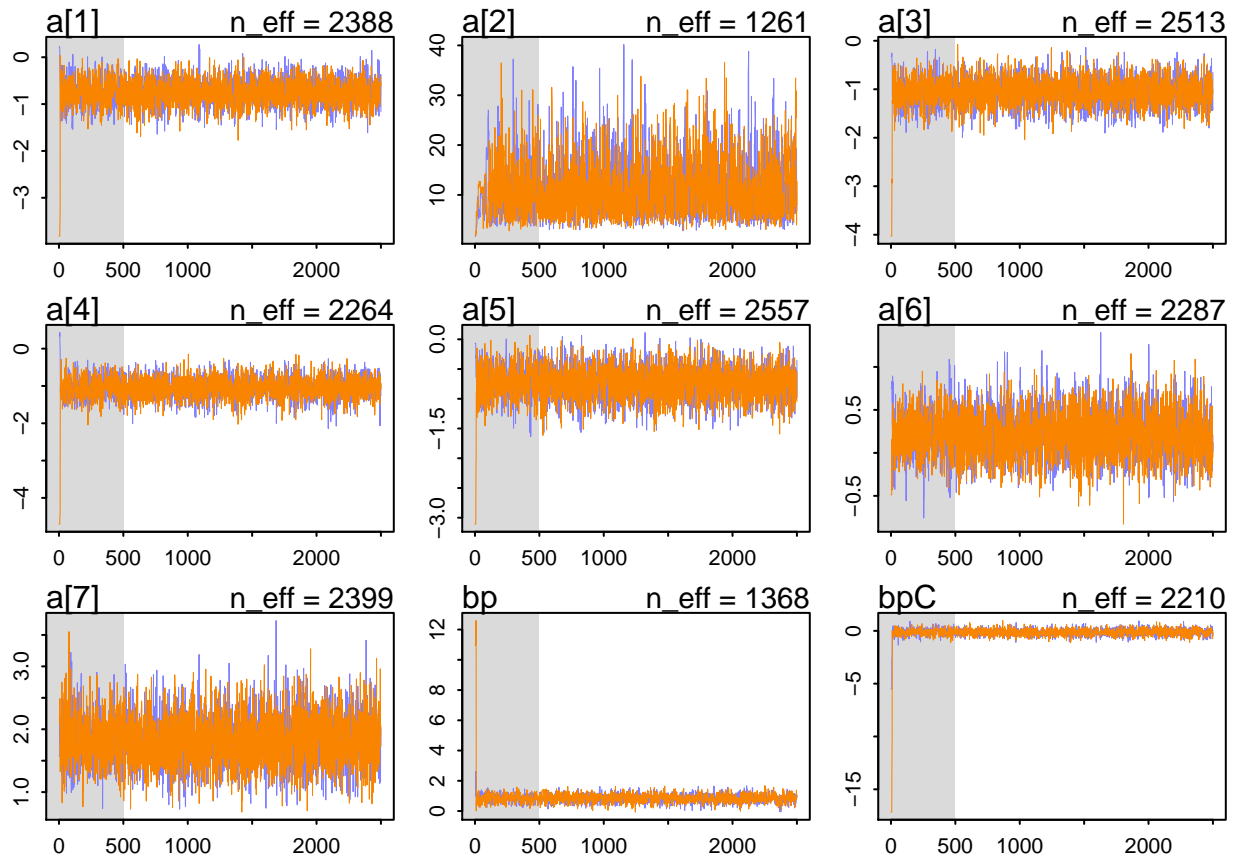
## map2stan model fit
## 4000 samples from 2 chains
##
## Formula:
## pulled_left ~ dbinom(1, p)
## logit(p) <- a[actor] + (bp + bpC * condition) * prosoc_left
## a[actor] ~ dnorm(0, 10)
## bp ~ dnorm(0, 10)
## bpC ~ dnorm(0, 10)
##
## Log-likelihood at expected values: -256.54
## Deviance: 513.08
## DIC: 529.38
## Effective number of parameters (pD): 8.15
##
## WAIC (SE): 529.66 (20)
## pWAIC: 8.24

```

```
stancode(m10.4)
```

```
## data{
##   int<lower=1> N;
##   int<lower=1> N_actor;
##   int pulled_left[N];
##   int actor[N];
##   int condition[N];
##   int prosoc_left[N];
## }
## parameters{
##   vector[N_actor] a;
##   real bp;
##   real bpC;
## }
## model{
##   vector[N] p;
##   bpC ~ normal( 0 , 10 );
##   bp ~ normal( 0 , 10 );
##   a ~ normal( 0 , 10 );
##   for ( i in 1:N ) {
##     p[i] = a[actor[i]] + (bp + bpC * condition[i]) * prosoc_left[i];
##   }
##   pulled_left ~ binomial_logit( 1 , p );
## }
## generated quantities{
##   vector[N] p;
##   real dev;
##   dev = 0;
##   for ( i in 1:N ) {
##     p[i] = a[actor[i]] + (bp + bpC * condition[i]) * prosoc_left[i];
##   }
##   dev = dev + (-2)*binomial_logit_lpmf( pulled_left | 1 , p );
## }
```

```
plot(m10.4)
```



```
## R code 10.15
unique( d$factor )
```

```
## [1] 1 2 3 4 5 6 7
```

```
## R code 10.16
precis( m10.4 , depth=2 )
```

```
## Warning in precis(m10.4, depth = 2): There were 25 divergent iterations during sampling.
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.
```

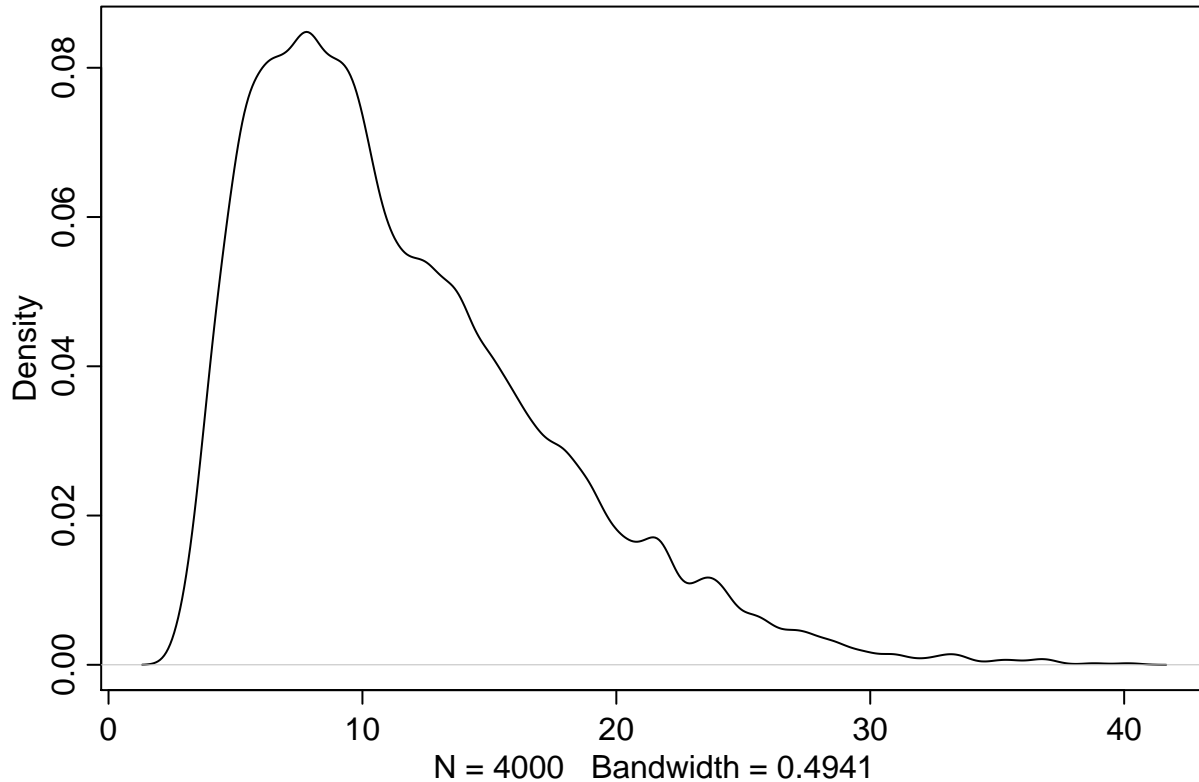
```
##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a[1] -0.75  0.27   -1.16   -0.32  2388   1
## a[2] 11.57  5.82    3.48   19.54  1261   1
## a[3] -1.05  0.28   -1.50   -0.60  2513   1
## a[4] -1.05  0.27   -1.45   -0.60  2264   1
## a[5] -0.74  0.27   -1.16   -0.29  2557   1
## a[6]  0.22  0.27   -0.22    0.65  2287   1
## a[7]  1.82  0.40    1.19    2.46  2399   1
## bp   0.84  0.26    0.43    1.27  1368   1
## bpC -0.13  0.30   -0.60    0.34  2210   1
```

check actor 2

```
## R code 10.17
post <- extract.samples( m10.4 )
str( post )
```

```
## List of 3
## $ a : num [1:4000, 1:7] -0.427 -0.692 -0.574 -0.754 -0.706 ...
## $ bp : num [1:4000(1d)] 1.233 0.988 0.615 0.895 0.779 ...
## $ bpC: num [1:4000(1d)] -0.511 -0.3332 -0.0621 -0.3553 -0.3694 ...
```

```
## R code 10.18
par(mfrow=c(1, 1))
dens( post$a[,2] )
```



check actor 1

```
## R code 10.19
chimp <- 1
d.pred <- list(
  pulled_left = rep( 0 , 4 ), # empty outcome
  prosoc_left = c(0,1,0,1), # right/left/right/left
  condition = c(0,0,1,1), # control/control/partner/partner
  actor = rep(chimp,4)
)
link.m10.4 <- link( m10.4 , data=d.pred )
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
```

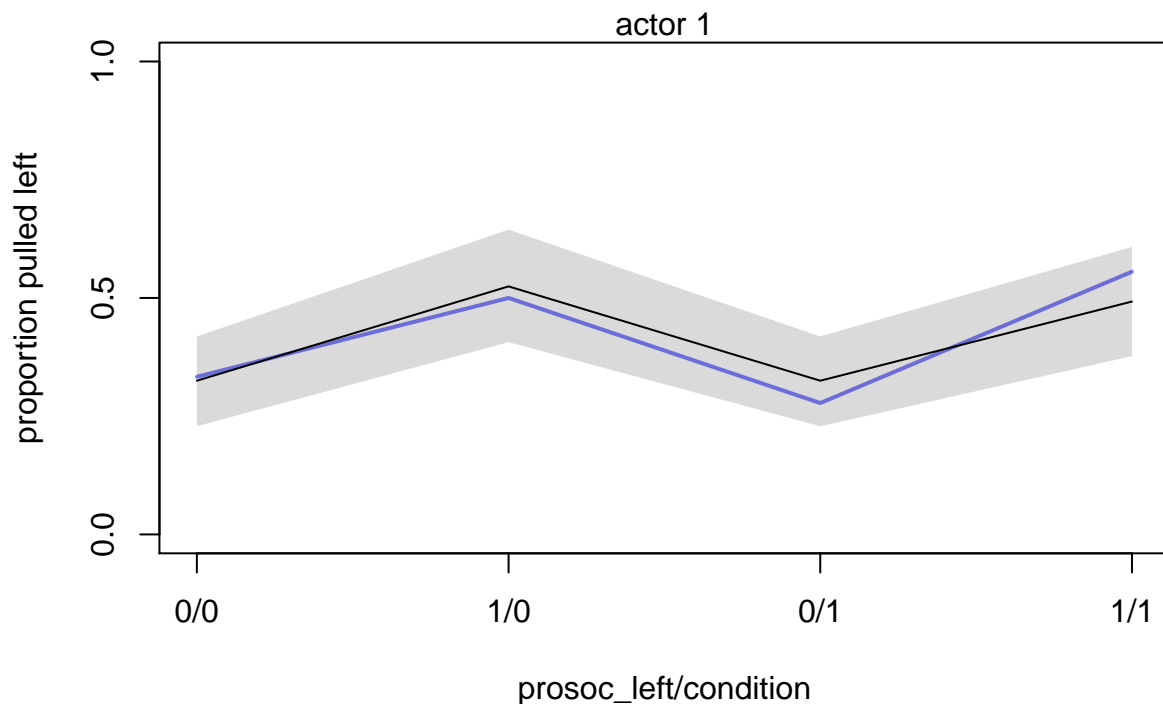
```
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
pred.p <- apply( link.m10.4 , 2 , mean )
pred.p.PI <- apply( link.m10.4 , 2 , PI )

plot( 0 , 0 , type="n" , xlab="prosoc_left/condition" ,
      ylab="proportion pulled left" , ylim=c(0,1) , xaxt="n" ,
      xlim=c(1,4) , yaxp=c(0,1,2) )
axis( 1 , at=1:4 , labels=c("0/0","1/0","0/1","1/1") )
mtext( paste( "actor" , chimp ) )

p <- by( d$pulled_left ,
        list(d$prosoc_left,d$condition,d$actor) , mean )
lines( 1:4 , as.vector(p[,chimp]) , col=rangi2 , lwd=2 )

lines( 1:4 , pred.p )
shade( pred.p.PI , 1:4 )
```



### aggregated binomial

```
## R code 10.20
data(chimpanzees)
d <- chimpanzees
d.aggregated <- aggregate( d$pulled_left ,
  list(prosoc_left=d$prosoc_left,condition=d$condition,actor=d$actor) ,
  sum )
```

```
## R code 10.21
m10.5 <- map(
```

```
alist(  
  x ~ dbinom( 18 , p ) ,  
  logit(p) <- a + (bp + bpC*condition)*prosoc_left ,  
  a ~ dnorm(0,10) ,  
  bp ~ dnorm(0,10) ,  
  bpC ~ dnorm(0,10)  
),  
data=d.aggregated )  
precis( m10.5 )
```

```
##      Mean StdDev 5.5% 94.5%  
## a      0.05  0.13 -0.15  0.25  
## bp     0.61  0.23  0.25  0.97  
## bpC   -0.10  0.26 -0.53  0.32
```