chimpanzees

Chris Parrish

June 30, 2016

Contents

chimpanzees

references:
- McElreath, Statistical Rethinking, chap 12, p.370

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

chimpanzees as actors

data

```
## R code 12.21
data(chimpanzees)
d <- chimpanzees
d$recipient <- NULL     # get rid of NAs
```

model

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

stan

```
m12.4 <- map2stan(
    alist(
        pulled_left ~ dbinom( 1 , p ) ,
        logit(p) <- a + a_actor[actor] + (bp + bpC*condition)*prosoc_left ,
        a_actor[actor] ~ dnorm( 0 , sigma_actor ),
        a ~ dnorm(0,10),
        bp ~ dnorm(0,10),
        bpC ~ dnorm(0,10),
        sigma_actor ~ dcauchy(0,1)
    ) ,
    data=d , warmup=1000 , iter=5000 , chains=4 , cores=3 )
```

```
##
## 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: 5e-06 seconds (Warm-up)
##                0.000228 seconds (Sampling)
##                0.000233 seconds (Total)
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
## [ 1600 / 16000 ]
[ 3200 / 16000 ]
[ 4800 / 16000 ]
[ 6400 / 16000 ]
[ 8000 / 16000 ]
[ 9600 / 16000 ]
[ 11200 / 16000 ]
[ 12800 / 16000 ]
[ 14400 / 16000 ]
[ 16000 / 16000 ]
```

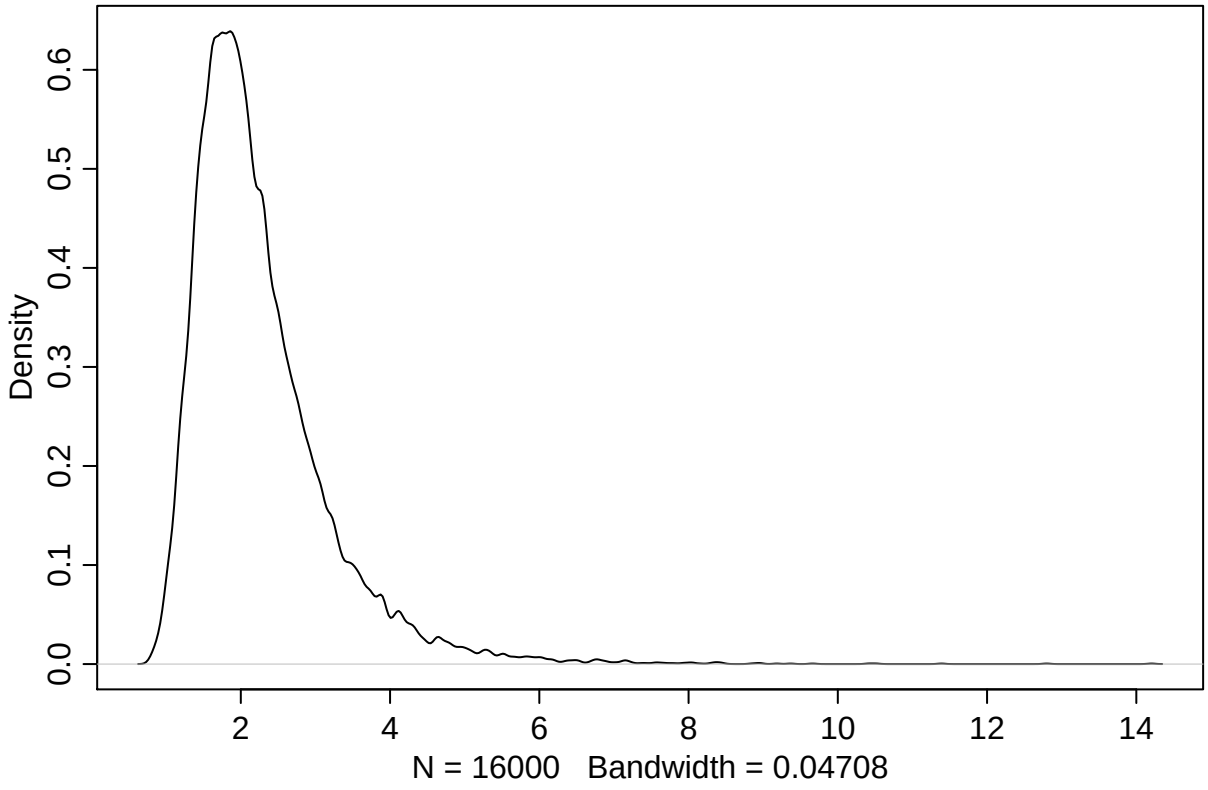
check

```
# plot(m12.4)     # trace plots
precis(m12.4)   # n_eff and Rhat
```

```
## 7 vector or matrix parameters omitted in display. Use depth=2 to show them.
```

```
##              Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a            0.48   0.94      -0.94       1.90  2886    1
## bp           0.82   0.25       0.42       1.23  8354    1
## bpC         -0.13   0.29      -0.60       0.33  7915    1
## sigma_actor  2.25   0.92       1.03       3.35  3658    1
```

```
post <- extract.samples(m12.4)
par(mfrow=c(1, 1))
dens(post$sigma_actor)
```



compute intercept for each astor

```
## R code 12.22
post <- extract.samples(m12.4)
total_a_actor <- sapply( 1:7 , function(actor) post$a + post$a_actor[,actor] )
round( apply(total_a_actor,2,mean) , 2 )
```

```
## [1] -0.72  4.58 -1.02 -1.02 -0.71  0.22  1.77
```

chimpanzees in blocks

A block is a set of experiments taking place on the same day.

model

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

stan

```
## R code 12.23
# prep data
d$block_id <- d$block  # name 'block' is reserved by Stan

m12.5 <- map2stan(
    alist(
        pulled_left ~ dbinom( 1 , p ),
        logit(p) <- a + a_actor[actor] + a_block[block_id] +
                    (bp + bpc*condition)*prosoc_left,
        a_actor[actor] ~ dnorm( 0 , sigma_actor ),
        a_block[block_id] ~ dnorm( 0 , sigma_block ),
        c(a,bp,bpc) ~ dnorm(0,10),
        sigma_actor ~ dcauchy(0,1),
        sigma_block ~ dcauchy(0,1)
    ) ,
    data=d, warmup=1000 , iter=6000 , chains=4 , cores=3 )
```

```
## Warning: There were 11 divergent transitions after warmup. Increasing
## adapt_delta above 0.8 may help.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
##
## 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: 4e-06 seconds (Warm-up)
##                0.000254 seconds (Sampling)
##                0.000258 seconds (Total)
```

```
## Computing WAIC
```

```
## Constructing posterior predictions
```

```
## [ 2000 / 20000 ]
[ 4000 / 20000 ]
[ 6000 / 20000 ]
[ 8000 / 20000 ]
[ 10000 / 20000 ]
[ 12000 / 20000 ]
[ 14000 / 20000 ]
[ 16000 / 20000 ]
[ 18000 / 20000 ]
[ 20000 / 20000 ]
```

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

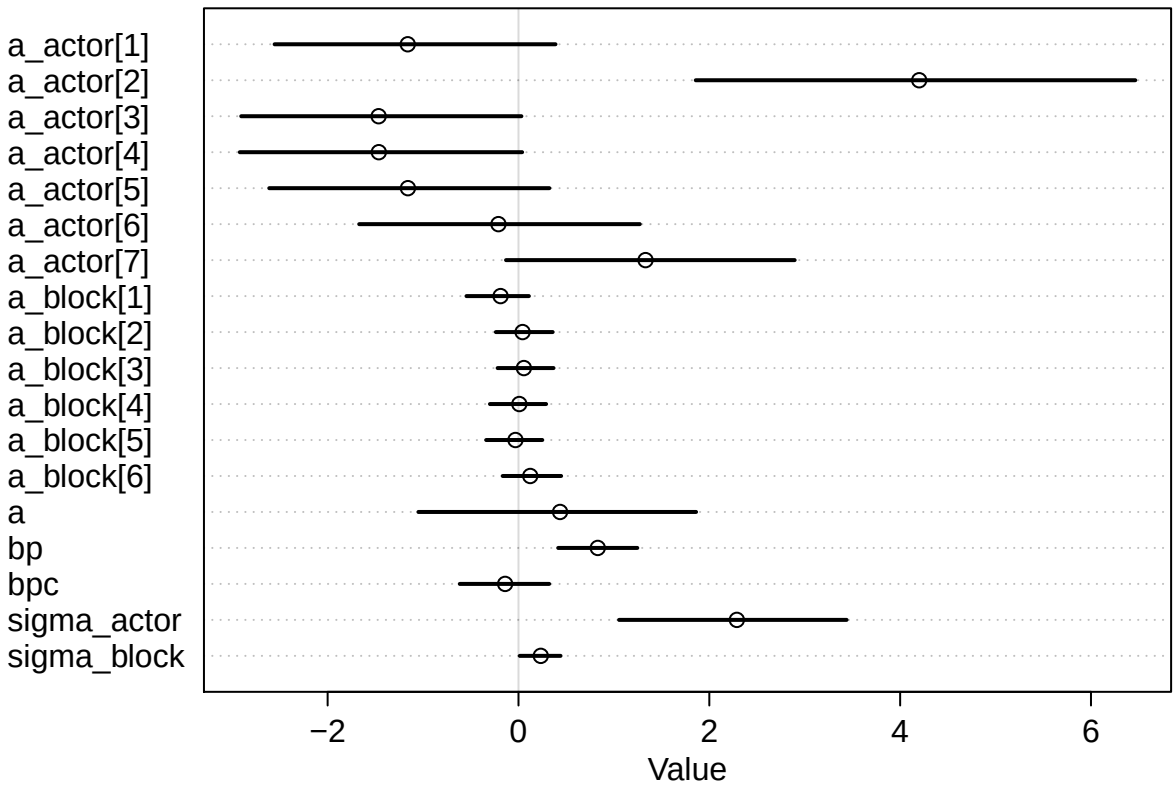
```
## R code 12.24
precis(m12.5,depth=2) # depth=2 displays varying effects
```

```
## Warning in precis(m12.5, depth = 2): There were 11 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_actor[1]  -1.16   0.99      -2.56       0.39  2693    1
## a_actor[2]   4.20   1.69       1.86       6.46  4807    1
## a_actor[3]  -1.47   0.99      -2.91       0.03  2687    1
## a_actor[4]  -1.46   0.99      -2.92       0.04  2690    1
## a_actor[5]  -1.16   0.99      -2.61       0.32  2704    1
## a_actor[6]  -0.21   0.99      -1.67       1.27  2701    1
## a_actor[7]   1.33   1.01      -0.13       2.89  2919    1
## a_block[1]  -0.19   0.23      -0.55       0.11  4931    1
## a_block[2]   0.04   0.19      -0.24       0.36  9676    1
## a_block[3]   0.06   0.19      -0.22       0.37  9485    1
## a_block[4]   0.01   0.19      -0.30       0.29  9821    1
## a_block[5]  -0.03   0.19      -0.34       0.25 10327    1
## a_block[6]   0.12   0.21      -0.17       0.45  6192    1
## a            0.44   0.98      -1.05       1.86  2620    1
## bp           0.83   0.26       0.42       1.25 13400    1
## bpc         -0.14   0.29      -0.62       0.32 13140    1
## sigma_actor  2.29   1.00       1.05       3.44  3810    1
## sigma_block  0.23   0.18       0.01       0.44  2774    1
```

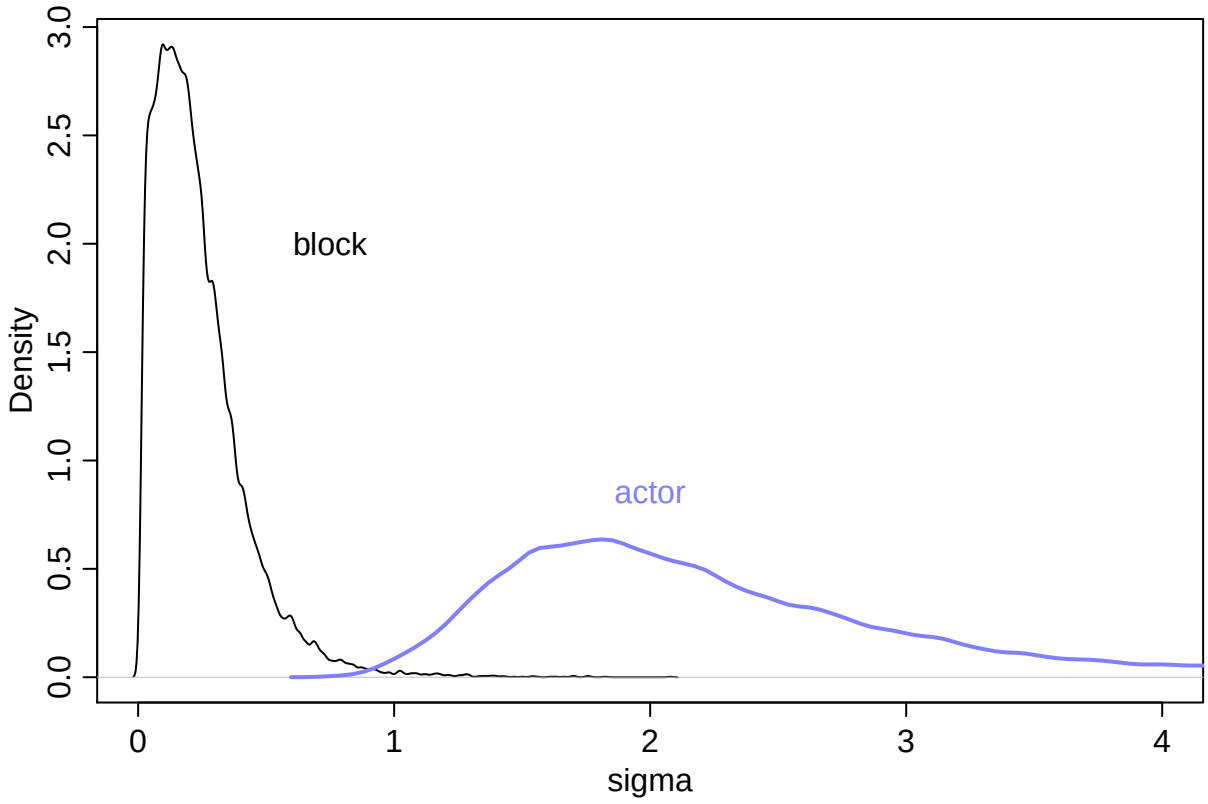
```
plot(precis(m12.5,depth=2)) # also plot
```

```
## Warning in precis(m12.5, depth = 2): There were 11 divergent iterations during sampling.
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.
```



compare *sigma_actor* **and** *sigma_block*

```
## R code 12.25
post <- extract.samples(m12.5)
dens( post$sigma_block , xlab="sigma" , xlim=c(0,4) )
dens( post$sigma_actor , col=rangi2 , lwd=2 , add=TRUE )
text( 2 , 0.85 , "actor" , col=rangi2 )
text( 0.75 , 2 , "block" )
```



compare the two models

```
## R code 12.26
compare(m12.4,m12.5)
```

```
##        WAIC pWAIC dWAIC weight    SE  dSE
## m12.4 531.1   8.0   0.0   0.65 19.48   NA
## m12.5 532.3  10.3   1.2   0.35 19.68 1.85
```

multilevel posterior predictions

for same clusters

```
## R code 12.27
chimp <- 2
d.pred <- list(
    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.m12.4 <- link( m12.4 , data=d.pred )
```

```
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
```

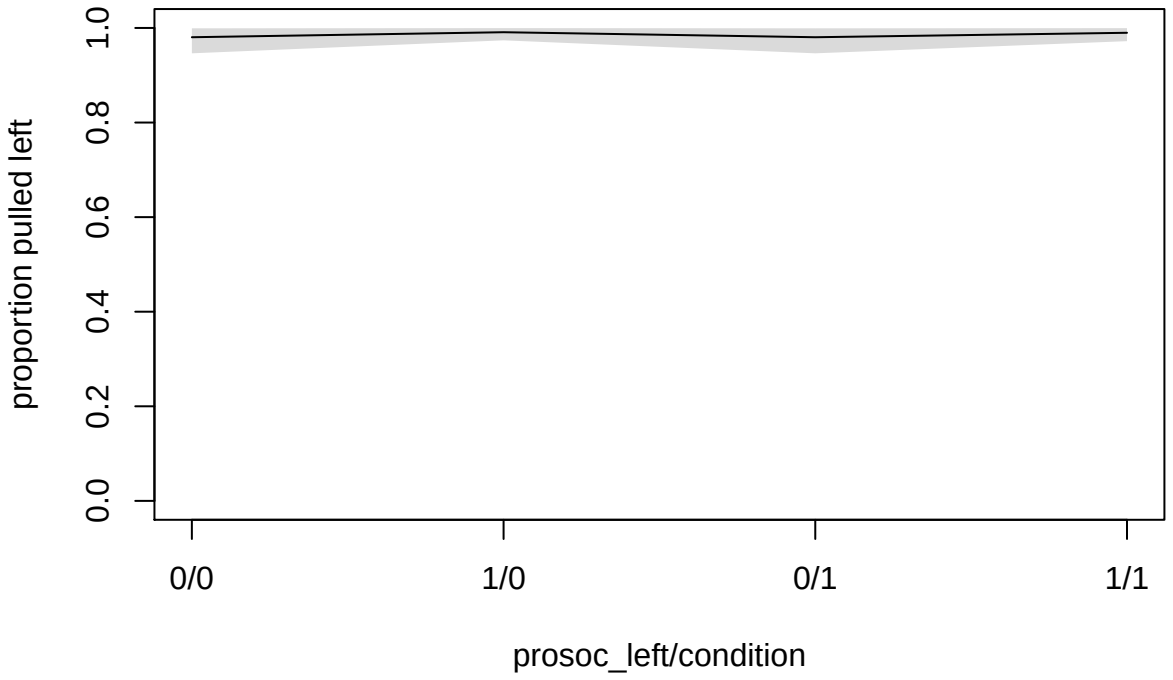
```
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]
```

```
pred.p <- apply( link.m12.4 , 2 , mean )
pred.p.PI <- apply( link.m12.4 , 2 , PI )
```

plot

```
## 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") )

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



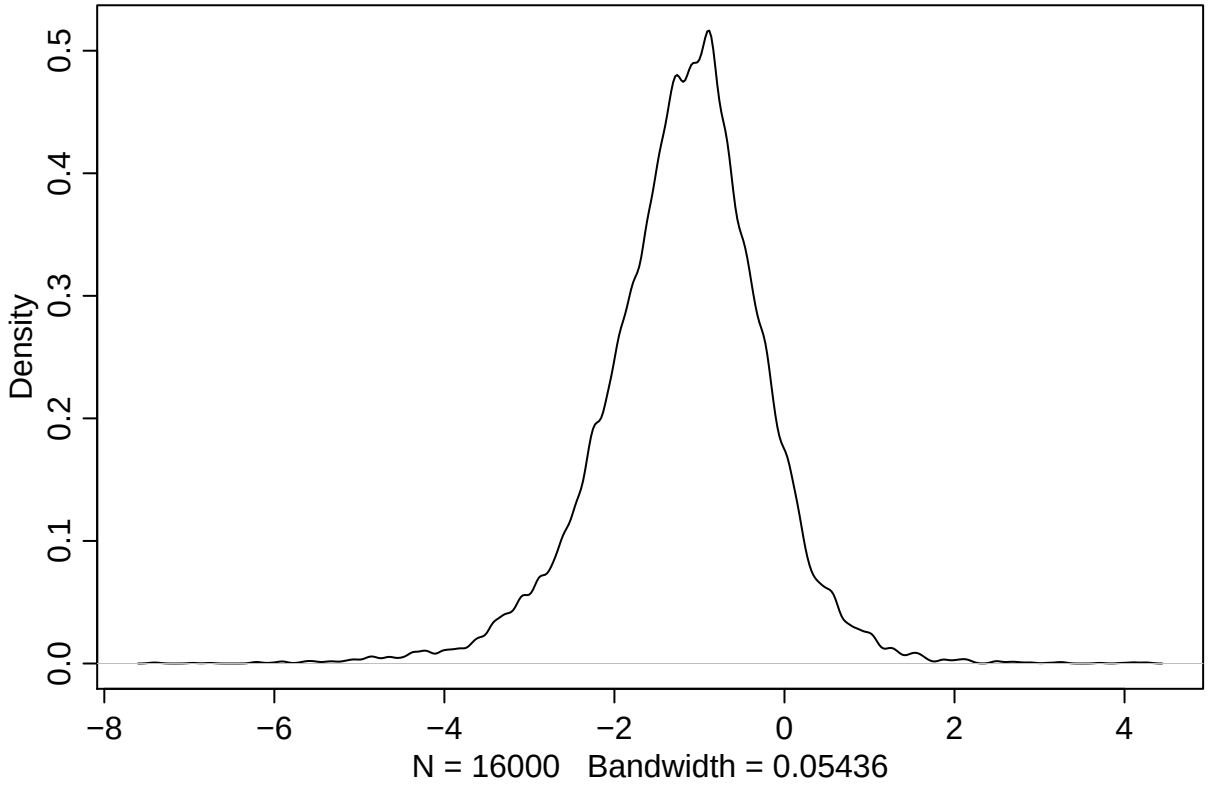
same calculations without `link`

```
## R code 12.28
post <- extract.samples(m12.4)
str(post)
```

```
## List of 5
##  $ a_actor    : num [1:16000, 1:7] -0.8578 -1.5153 -0.6761 -1.3479 -0.0161 ...
##  $ a          : num [1:16000(1d)] -0.0731 0.7379 -0.4881 0.8897 -0.6316 ...
##  $ bp         : num [1:16000(1d)] 0.828 0.888 1.193 1.084 1.05 ...
##  $ bpC        : num [1:16000(1d)] 0.00488 -0.17078 -0.18192 -0.39641 -0.1513 ...
##  $ sigma_actor: num [1:16000(1d)] 1.63 1.65 2.67 2.92 2.51 ...
```

Plot density for actor 5

```
## R code 12.29
dens( post$a_actor[,5] )
```



construct posterior predictions

Assemble the `link` function

```
## R code 12.30
p.link <- function( prosoc_left , condition , actor ) {
    logodds <- with( post ,
        a + a_actor[,actor] + (bp + bpC * condition) * prosoc_left
    )
    return( logistic(logodds) )
}
```

Compute predictions

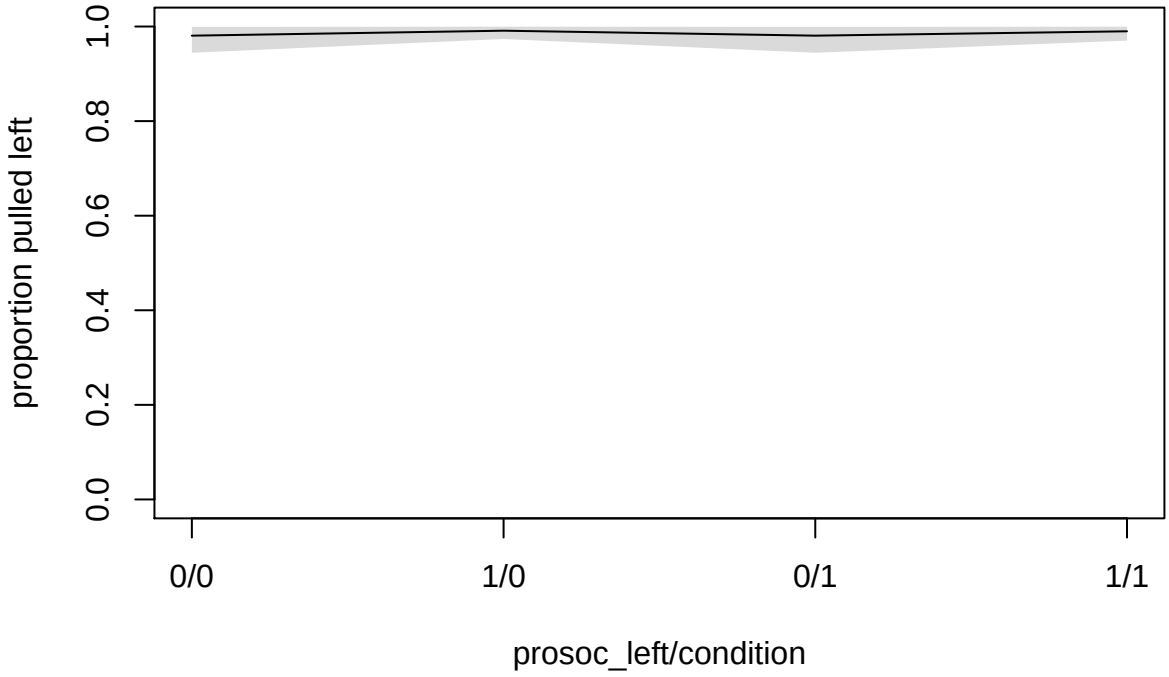
```
## R code 12.31
prosoc_left <- c(0,1,0,1)
condition <- c(0,0,1,1)
pred.raw <- sapply( 1:4 , function(i) p.link(prosoc_left[i],condition[i],2) )
```

```
pred.p <- apply( pred.raw , 2 , mean )
pred.p.PI <- apply( pred.raw , 2 , PI )
```

Plot

```
## 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") )

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



for new clusters

posterior prediction for a previouslu unobserved 'average' actor

```
## R code 12.32
d.pred <- list(
    prosoc_left = c(0,1,0,1),    # right/left/right/left
    condition = c(0,0,1,1),      # control/control/partner/partner
    actor = rep(2,4) )           # placeholder
```

```
## R code 12.33
# replace varying intercept samples with zeros
# 1000 samples by 7 actors
a_actor_zeros <- matrix(0,1000,7)
```

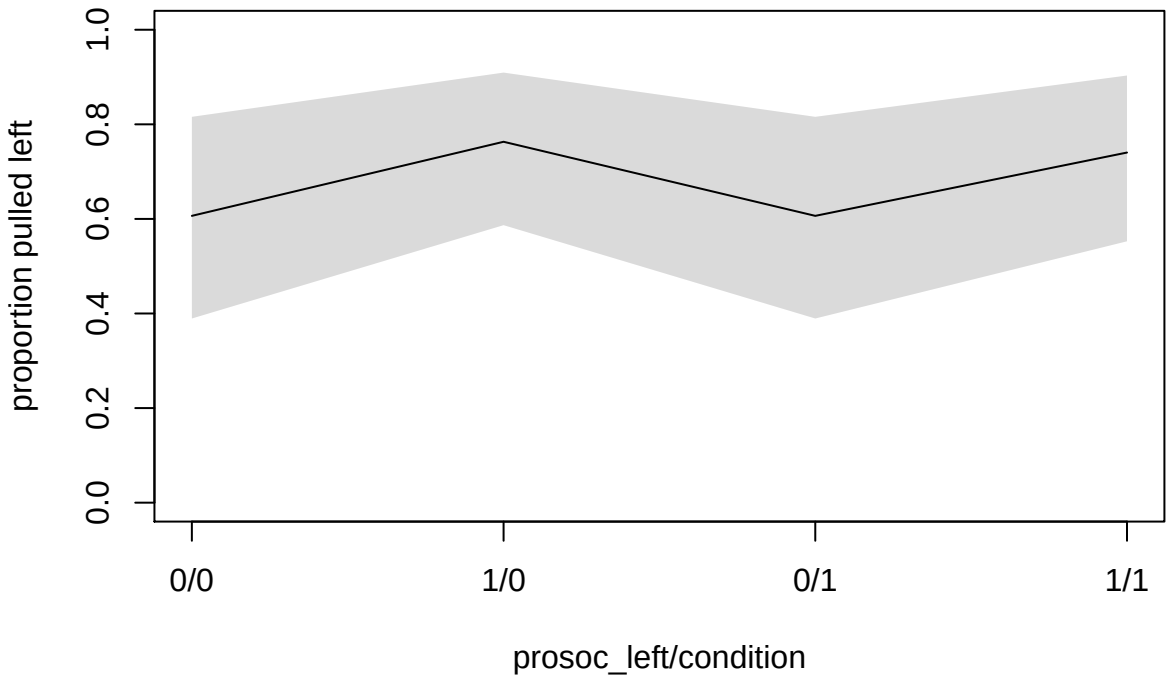
Pass this data to `link`

```
## R code 12.34
# fire up link
# note use of replace list
link.m12.4 <- link( m12.4 , n=1000 , data=d.pred ,
    replace=list(a_actor=a_actor_zeros) )
```

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

Plot 'average' actor

```
# summarize and plot
pred.p.mean <- apply( link.m12.4 , 2 , mean )
pred.p.PI <- apply( link.m12.4 , 2 , PI , prob=0.8 )
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") )
lines( 1:4 , pred.p.mean )
shade( pred.p.PI , 1:4 )
```



Show variation among actors

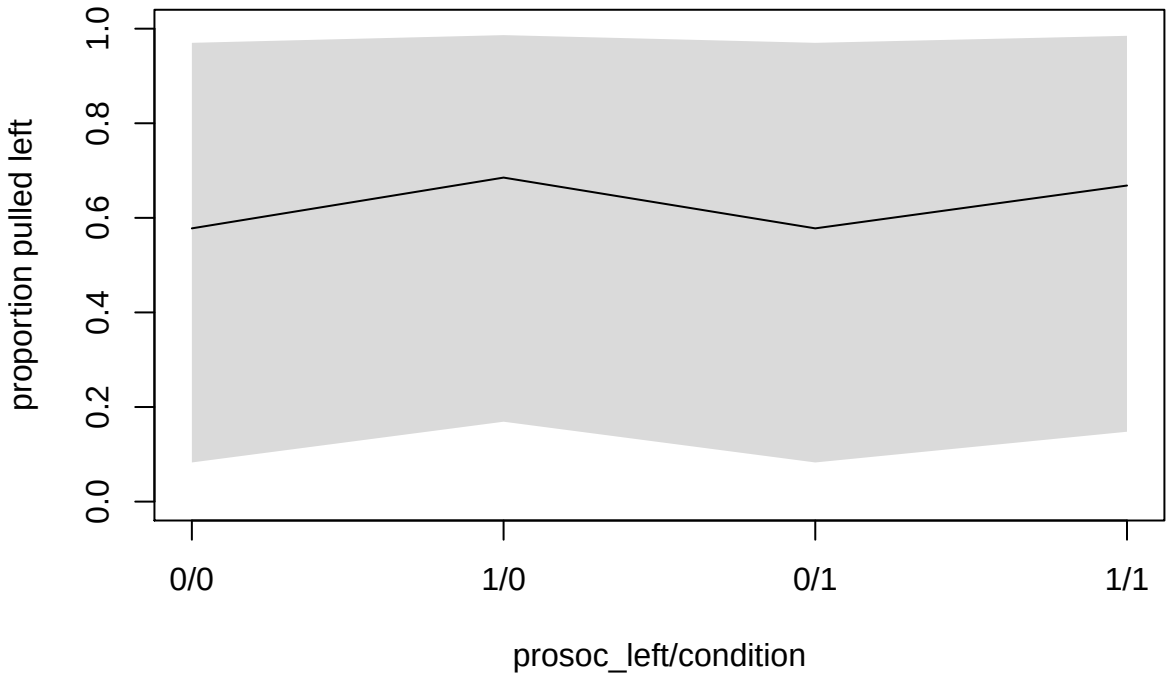
```
## R code 12.35
# replace varying intercept samples with simulations
post <- extract.samples(m12.4)
a_actor_sims <- rnorm(7000,0,post$sigma_actor)
a_actor_sims <- matrix(a_actor_sims,1000,7)
```

```
## R code 12.36
link.m12.4 <- link( m12.4 , n=1000 , data=d.pred ,
    replace=list(a_actor=a_actor_sims) )
```

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

Plot marginal of actor

```
# summarize and plot
pred.p.mean <- apply( link.m12.4 , 2 , mean )
pred.p.PI <- apply( link.m12.4 , 2 , PI , prob=0.8 )
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") )
lines( 1:4 , pred.p.mean )
shade( pred.p.PI , 1:4 )
```



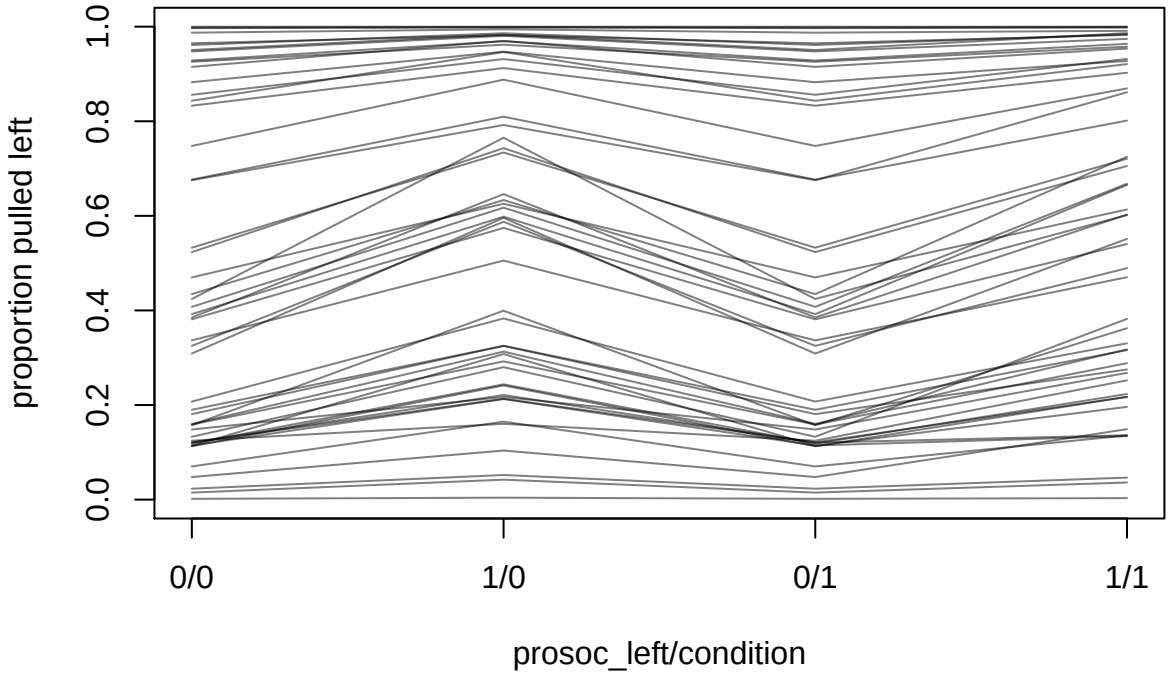
50 simulated actors

```
## R code 12.37
post <- extract.samples(m12.4)
sim.actor <- function(i) {
    sim_a_actor <- rnorm( 1 , 0 , post$sigma_actor[i] )
    P <- c(0,1,0,1)
    C <- c(0,0,1,1)
    p <- logistic(
        post$a[i] +
        sim_a_actor +
        (post$bp[i] + post$bpC[i]*C)*P
    )
    return(p)
}
```

Plot 50 simulated actors

```
## R code 12.38
# empty plot
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 50 simulated actors
for ( i in 1:50 ) lines( 1:4 , sim.actor(i) , col=col.alpha("black",0.5) )
```


proportion pulled left
0.0
0.2
0.4
0.6
0.8
1.0
0/0
1/0
0/1
1/1
prosoc_left/condition