

# Poisson

*Chris Parrish*

*June 29, 2016*

## Contents

<b>Binomial and Poisson distributions</b>	<b>1</b>
<b>Island tools</b>	<b>1</b>
data . . . . .	1
map . . . . .	2
post . . . . .	3
several models . . . . .	3
comparisons . . . . .	4
plot . . . . .	5
stan . . . . .	8
centered predictor . . . . .	10

Poisson

reference: McElreath, Statistical Rethinking, chap 10, p.304

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

## Binomial and Poisson distributions

```
## R code 10.38
y <- rbinom(1e5,1000,1/1000)
c( mean(y) , var(y) )
```

```
## [1] 0.9938300 0.9935619
```

## Island tools

data

```
## R code 10.39
data(Kline)
d <- Kline
d
```

```
##      culture population contact total_tools mean_TU
## 1  Malekula      1100     low           13      3.2
## 2  Tikopia       1500     low           22      4.7
## 3  Santa Cruz    3600     low           24      4.0
## 4    Yap        4791    high           43      5.0
## 5  Lau Fiji     7400    high           33      5.0
```

```
## 6  Trobriand      8000   high      19      4.0
## 7    Chuuk       9200   high      40      3.8
## 8    Manus      13000   low       28      6.6
## 9    Tonga      17500   high      55      5.4
## 10   Hawaii     275000   low       71      6.6
```

```
## R code 10.40
d$log_pop <- log(d$population)
d$contact_high <- ifelse( d$contact=="high" , 1 , 0 )
```

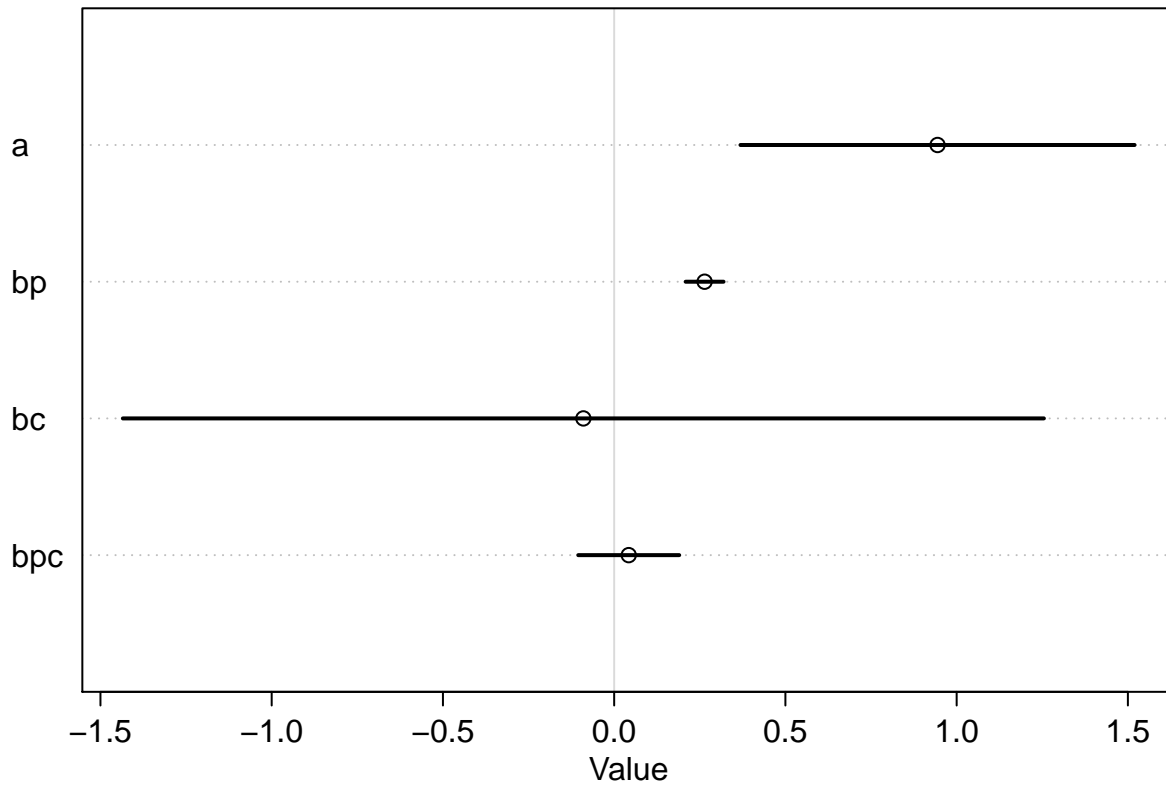
## map

```
## R code 10.41
m10.10 <- map(
  alist(
    total_tools ~ dpois( lambda ),
    log(lambda) <- a + bp*log_pop +
      bc*contact_high + bpc*contact_high*log_pop,
    a ~ dnorm(0,100),
    c(bp,bc,bpc) ~ dnorm(0,1)
  ),
  data=d )
```

```
## R code 10.42
precis(m10.10,corr=TRUE)
```

```
##      Mean StdDev 5.5% 94.5%      a      bp      bc      bpc
## a      0.94   0.36 0.37  1.52  1.00 -0.98 -0.13  0.07
## bp     0.26   0.03 0.21  0.32 -0.98  1.00  0.12 -0.08
## bc    -0.09   0.84 -1.43  1.25 -0.13  0.12  1.00 -0.99
## bpc    0.04   0.09 -0.10  0.19  0.07 -0.08 -0.99  1.00
```

```
plot(precis(m10.10))
```



post

```
## R code 10.43
post <- extract.samples(m10.10)
lambda_high <- exp( post$a + post$bc + (post$bp + post$bpc)*8 )
lambda_low <- exp( post$a + post$bp*8 )

## R code 10.44
diff <- lambda_high - lambda_low
sum(diff > 0)/length(diff)

## [1] 0.9557
```

several models

```
## R code 10.45
# no interaction
m10.11 <- map(
  alist(
    total_tools ~ dpois( lambda ),
    log(lambda) <- a + bp*log_pop + bc*contact_high,
    a ~ dnorm(0,100),
    c(bp,bc) ~ dnorm( 0 , 1 )
  ), data=d )

## R code 10.46
```

```

# no contact rate
m10.12 <- map(
  alist(
    total_tools ~ dpois( lambda ),
    log(lambda) <- a + bp*log_pop,
    a ~ dnorm(0,100),
    bp ~ dnorm( 0 , 1 )
  ), data=d )

# no log-population
m10.13 <- map(
  alist(
    total_tools ~ dpois( lambda ),
    log(lambda) <- a + bc*contact_high,
    a ~ dnorm(0,100),
    bc ~ dnorm( 0 , 1 )
  ), data=d )

## R code 10.47
# intercept only
m10.14 <- map(
  alist(
    total_tools ~ dpois( lambda ),
    log(lambda) <- a,
    a ~ dnorm(0,100)
  ), data=d )

```

## comparisons

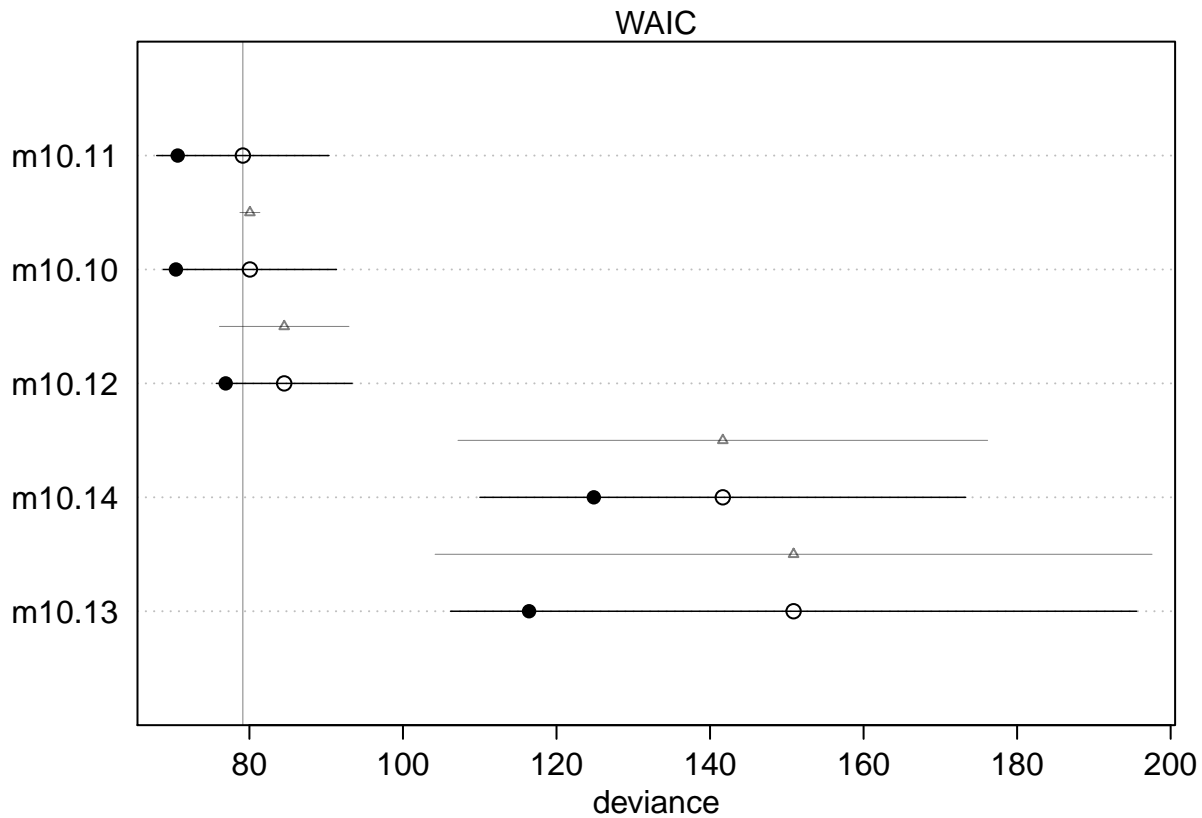
```

# compare all using WAIC
# adding n=1e4 for more stable WAIC estimates
# will also plot the comparison
( islands.compare <- compare(m10.10,m10.11,m10.12,m10.13,m10.14,n=1e4) )

##          WAIC pWAIC dWAIC weight    SE  dSE
## m10.11  79.1   4.2   0.0   0.59 11.22  NA
## m10.10  80.1   4.8   0.9   0.37 11.28  1.30
## m10.12  84.5   3.8   5.4   0.04  8.84  8.43
## m10.14 141.7   8.4  62.5   0.00 31.64 34.50
## m10.13 150.9  17.2  71.7   0.00 44.69 46.70

plot(islands.compare)

```



plot

```
## R code 10.48
# make plot of raw data to begin
# point character (pch) indicates contact rate
pch <- ifelse( d$contact_high==1 , 16 , 1 )
plot( d$log_pop , d$total_tools , col=rangi2 , pch=pch ,
      xlab="log-population" , ylab="total tools" )

# sequence of log-population sizes to compute over
log_pop.seq <- seq( from=6 , to=13 , length.out=30 )

# compute trend for high contact islands
d.pred <- data.frame(
  log_pop = log_pop.seq,
  contact_high = 1
)
lambda.pred.h <- ensemble( m10.10 , m10.11 , m10.12 , 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 ]

lambda.med <- apply( lambda.pred.h$link , 2 , median )
lambda.PI <- apply( lambda.pred.h$link , 2 , PI )

# plot predicted trend for high contact islands
lines( log_pop.seq , lambda.med , col=rangi2 )
shade( lambda.PI , log_pop.seq , col=col.alpha(rangi2,0.2) )

# compute trend for low contact islands
d.pred <- data.frame(
  log_pop = log_pop.seq,
  contact_high = 0
)
lambda.pred.l <- ensemble( m10.10 , m10.11 , m10.12 , 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 ]

```

```

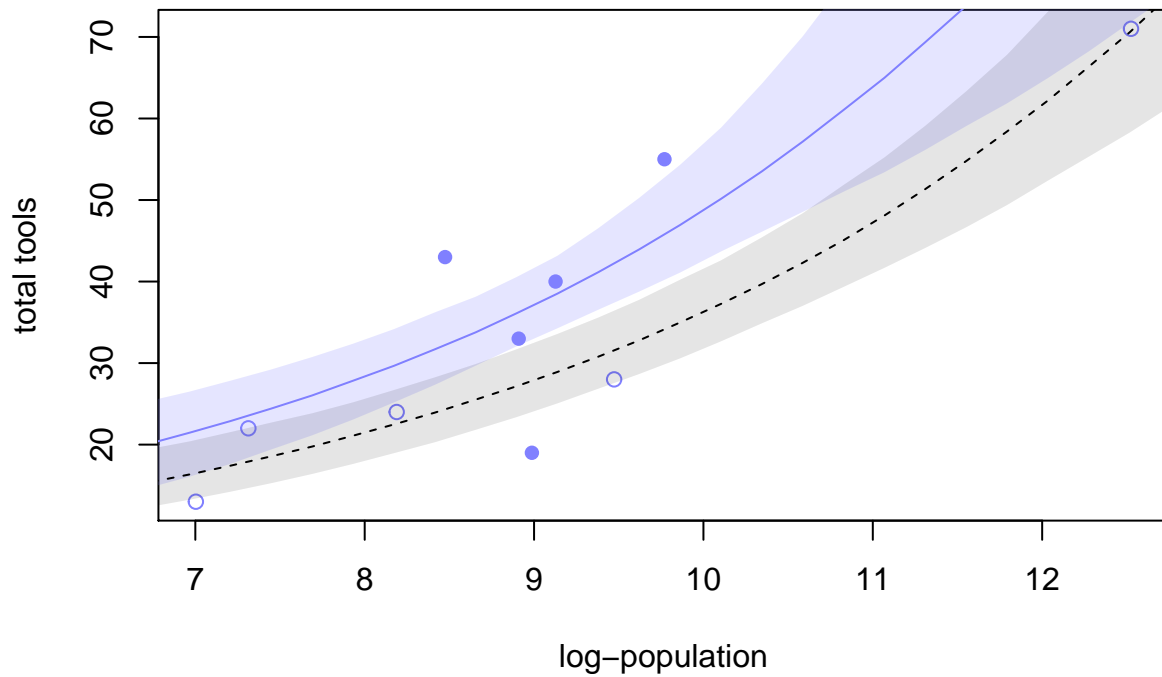
## Constructing posterior predictions
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]

## Constructing posterior predictions
## [ 100 / 1000 ]
[ 200 / 1000 ]
[ 300 / 1000 ]
[ 400 / 1000 ]
[ 500 / 1000 ]
[ 600 / 1000 ]
[ 700 / 1000 ]
[ 800 / 1000 ]
[ 900 / 1000 ]
[ 1000 / 1000 ]

lambda.med <- apply( lambda.pred.l$link , 2 , median )
lambda.PI <- apply( lambda.pred.l$link , 2 , PI )

# plot again
lines( log_pop.seq , lambda.med , lty=2 )
shade( lambda.PI , log_pop.seq , col=col.alpha("black",0.1) )

```



## stan

```
## R code 10.49
m10.10stan <- map2stan( m10.10 , iter=3000 , warmup=1000 , chains=4 )

## Warning in FUN(X[[i]], ...): data with name culture is not numeric and not
## used

## Warning in FUN(X[[i]], ...): data with name contact is not numeric and not
## used

##
## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' NOW (CHAIN 1).
##
## Chain 1, Iteration:    1 / 3000 [ 0%] (Warmup)
## Chain 1, Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 1, Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 1, Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 1, Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 1, Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 1, Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 1, Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 1, Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 1, Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 1, Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 1, Iteration:  3000 / 3000 [100%] (Sampling)
## Elapsed Time: 0.110749 seconds (Warm-up)
##                0.219252 seconds (Sampling)
##                0.330001 seconds (Total)
##
##
## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' NOW (CHAIN 2).
##
## Chain 2, Iteration:    1 / 3000 [ 0%] (Warmup)
## Chain 2, Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 2, Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 2, Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 2, Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 2, Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 2, Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 2, Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 2, Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 2, Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 2, Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 2, Iteration:  3000 / 3000 [100%] (Sampling)
## Elapsed Time: 0.138003 seconds (Warm-up)
##                0.181465 seconds (Sampling)
##                0.319468 seconds (Total)
##
##
## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' NOW (CHAIN 3).
##
## Chain 3, Iteration:    1 / 3000 [ 0%] (Warmup)
## Chain 3, Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 3, Iteration:   600 / 3000 [ 20%] (Warmup)
```



```

## Chain 3, Iteration: 900 / 3000 [ 30%] (Warmup)
## Chain 3, Iteration: 1001 / 3000 [ 33%] (Sampling)
## Chain 3, Iteration: 1300 / 3000 [ 43%] (Sampling)
## Chain 3, Iteration: 1600 / 3000 [ 53%] (Sampling)
## Chain 3, Iteration: 1900 / 3000 [ 63%] (Sampling)
## Chain 3, Iteration: 2200 / 3000 [ 73%] (Sampling)
## Chain 3, Iteration: 2500 / 3000 [ 83%] (Sampling)
## Chain 3, Iteration: 2800 / 3000 [ 93%] (Sampling)
## Chain 3, Iteration: 3000 / 3000 [100%] (Sampling)
## Elapsed Time: 0.162884 seconds (Warm-up)
##                 0.196407 seconds (Sampling)
##                 0.359291 seconds (Total)
##
##
## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' NOW (CHAIN 4).
##
## Chain 4, Iteration: 1 / 3000 [ 0%] (Warmup)
## Chain 4, Iteration: 300 / 3000 [ 10%] (Warmup)
## Chain 4, Iteration: 600 / 3000 [ 20%] (Warmup)
## Chain 4, Iteration: 900 / 3000 [ 30%] (Warmup)
## Chain 4, Iteration: 1001 / 3000 [ 33%] (Sampling)
## Chain 4, Iteration: 1300 / 3000 [ 43%] (Sampling)
## Chain 4, Iteration: 1600 / 3000 [ 53%] (Sampling)
## Chain 4, Iteration: 1900 / 3000 [ 63%] (Sampling)
## Chain 4, Iteration: 2200 / 3000 [ 73%] (Sampling)
## Chain 4, Iteration: 2500 / 3000 [ 83%] (Sampling)
## Chain 4, Iteration: 2800 / 3000 [ 93%] (Sampling)
## Chain 4, Iteration: 3000 / 3000 [100%] (Sampling)
## Elapsed Time: 0.399114 seconds (Warm-up)
##                 1.10181 seconds (Sampling)
##                 1.50092 seconds (Total)
##
## Warning in FUN(X[[i]], ...): data with name culture is not numeric and not
## used
##
## Warning in FUN(X[[i]], ...): data with name contact is not numeric and not
## used
##
## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' 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)
##                 2.3e-05 seconds (Sampling)
##                 2.6e-05 seconds (Total)
##
## Computing WAIC
## Constructing posterior predictions
## [ 800 / 8000 ]
## [ 1600 / 8000 ]
## [ 2400 / 8000 ]

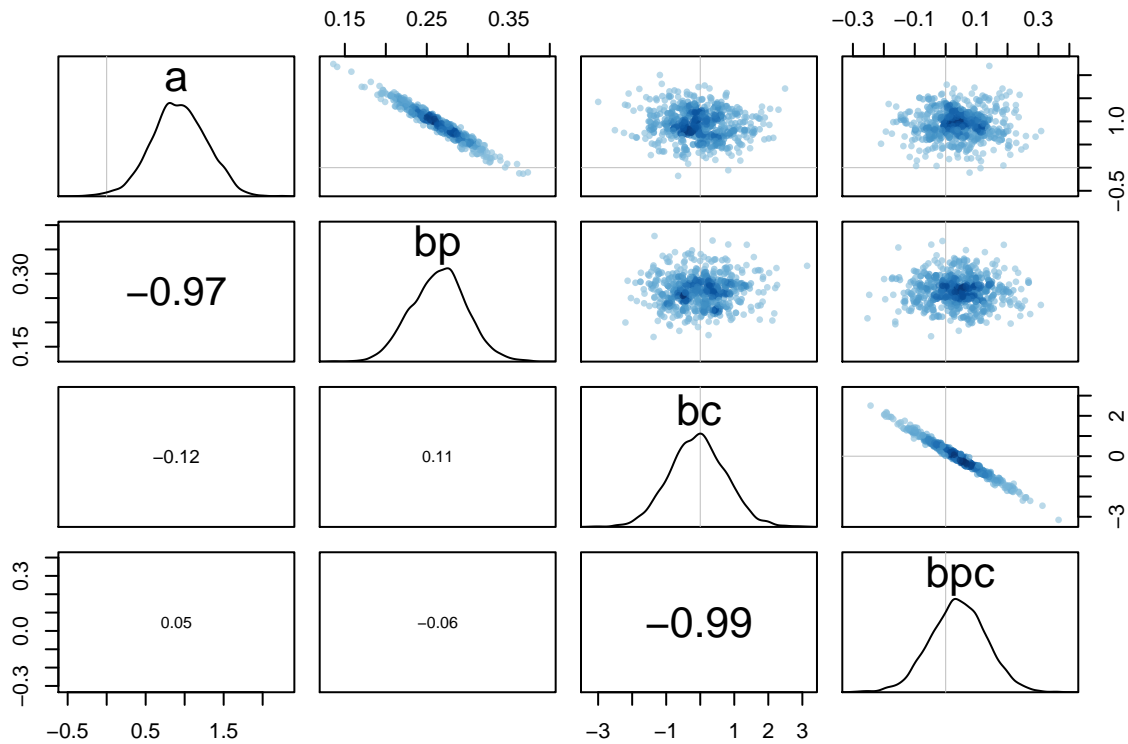
```

```
[ 3200 / 8000 ]
[ 4000 / 8000 ]
[ 4800 / 8000 ]
[ 5600 / 8000 ]
[ 6400 / 8000 ]
[ 7200 / 8000 ]
[ 8000 / 8000 ]
```

```
precis(m10.10stan)
```

```
##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a      0.93  0.35    0.38    1.50 2603    1
## bp     0.26  0.03    0.21    0.32 2629    1
## bc    -0.07  0.83   -1.37    1.25 1347    1
## bpc    0.04  0.09   -0.11    0.18 1351    1
```

```
pairs(m10.10stan)
```



## centered predictor

```
## R code 10.50
# construct centered predictor
d$log_pop_c <- d$log_pop - mean(d$log_pop)

# re-estimate
m10.10stan.c <- map2stan(
  alist(
    total_tools ~ dpois( lambda ),
    log(lambda) <- a + bp*log_pop_c + bc*contact_high +
      bpc*log_pop_c*contact_high ,
```

```

a ~ dnorm(0,10) ,
bp ~ dnorm(0,1) ,
bc ~ dnorm(0,1) ,
bcp ~ dnorm(0,1)
) ,
data=d , iter=3000 , warmup=1000 , chains=4 )

```

```

## Warning in FUN(X[[i]], ...): data with name culture is not numeric and not
## used

```

```

## Warning in FUN(X[[i]], ...): data with name contact is not numeric and not
## used

```

```

##
## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' NOW (CHAIN 1).
##

```

```

## Chain 1, Iteration:    1 / 3000 [ 0%] (Warmup)
## Chain 1, Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 1, Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 1, Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 1, Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 1, Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 1, Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 1, Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 1, Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 1, Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 1, Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 1, Iteration:  3000 / 3000 [100%] (Sampling)
## Elapsed Time: 0.022031 seconds (Warm-up)
##                0.044439 seconds (Sampling)
##                0.06647 seconds (Total)
##

```

```

## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' NOW (CHAIN 2).
##

```

```

## Chain 2, Iteration:    1 / 3000 [ 0%] (Warmup)
## Chain 2, Iteration:   300 / 3000 [ 10%] (Warmup)
## Chain 2, Iteration:   600 / 3000 [ 20%] (Warmup)
## Chain 2, Iteration:   900 / 3000 [ 30%] (Warmup)
## Chain 2, Iteration:  1001 / 3000 [ 33%] (Sampling)
## Chain 2, Iteration:  1300 / 3000 [ 43%] (Sampling)
## Chain 2, Iteration:  1600 / 3000 [ 53%] (Sampling)
## Chain 2, Iteration:  1900 / 3000 [ 63%] (Sampling)
## Chain 2, Iteration:  2200 / 3000 [ 73%] (Sampling)
## Chain 2, Iteration:  2500 / 3000 [ 83%] (Sampling)
## Chain 2, Iteration:  2800 / 3000 [ 93%] (Sampling)
## Chain 2, Iteration:  3000 / 3000 [100%] (Sampling)
## Elapsed Time: 0.024733 seconds (Warm-up)
##                0.043809 seconds (Sampling)
##                0.068542 seconds (Total)
##

```

```

## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' NOW (CHAIN 3).
##

```

```

## Chain 3, Iteration:    1 / 3000 [ 0%] (Warmup)

```

```

## Chain 3, Iteration: 300 / 3000 [ 10%] (Warmup)
## Chain 3, Iteration: 600 / 3000 [ 20%] (Warmup)
## Chain 3, Iteration: 900 / 3000 [ 30%] (Warmup)
## Chain 3, Iteration: 1001 / 3000 [ 33%] (Sampling)
## Chain 3, Iteration: 1300 / 3000 [ 43%] (Sampling)
## Chain 3, Iteration: 1600 / 3000 [ 53%] (Sampling)
## Chain 3, Iteration: 1900 / 3000 [ 63%] (Sampling)
## Chain 3, Iteration: 2200 / 3000 [ 73%] (Sampling)
## Chain 3, Iteration: 2500 / 3000 [ 83%] (Sampling)
## Chain 3, Iteration: 2800 / 3000 [ 93%] (Sampling)
## Chain 3, Iteration: 3000 / 3000 [100%] (Sampling)
## Elapsed Time: 0.026234 seconds (Warm-up)
##                 0.043312 seconds (Sampling)
##                 0.069546 seconds (Total)
##
##
## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' NOW (CHAIN 4).
##
## Chain 4, Iteration: 1 / 3000 [ 0%] (Warmup)
## Chain 4, Iteration: 300 / 3000 [ 10%] (Warmup)
## Chain 4, Iteration: 600 / 3000 [ 20%] (Warmup)
## Chain 4, Iteration: 900 / 3000 [ 30%] (Warmup)
## Chain 4, Iteration: 1001 / 3000 [ 33%] (Sampling)
## Chain 4, Iteration: 1300 / 3000 [ 43%] (Sampling)
## Chain 4, Iteration: 1600 / 3000 [ 53%] (Sampling)
## Chain 4, Iteration: 1900 / 3000 [ 63%] (Sampling)
## Chain 4, Iteration: 2200 / 3000 [ 73%] (Sampling)
## Chain 4, Iteration: 2500 / 3000 [ 83%] (Sampling)
## Chain 4, Iteration: 2800 / 3000 [ 93%] (Sampling)
## Chain 4, Iteration: 3000 / 3000 [100%] (Sampling)
## Elapsed Time: 0.022302 seconds (Warm-up)
##                 0.042647 seconds (Sampling)
##                 0.064949 seconds (Total)
##
## Warning in FUN(X[[i]], ...): data with name culture is not numeric and not
## used
##
## Warning in FUN(X[[i]], ...): data with name contact is not numeric and not
## used
##
## SAMPLING FOR MODEL 'total_tools ~ dpois(lambda)' 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)
##                 2.2e-05 seconds (Sampling)
##                 2.5e-05 seconds (Total)
##
## Computing WAIC
## Constructing posterior predictions
## [ 800 / 8000 ]

```

```

[ 1600 / 8000 ]
[ 2400 / 8000 ]
[ 3200 / 8000 ]
[ 4000 / 8000 ]
[ 4800 / 8000 ]
[ 5600 / 8000 ]
[ 6400 / 8000 ]
[ 7200 / 8000 ]
[ 8000 / 8000 ]

```

```
precis(m10.10stan.c)
```

```

##      Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a    3.31   0.09   3.17   3.44  4275    1
## bp   0.26   0.03   0.21   0.32  5044    1
## bc   0.29   0.11   0.10   0.47  3728    1
## bcp  0.06   0.16  -0.19   0.32  5023    1

```

```
pairs(m10.10stan.c)
```

