

Analysis and Diagnostics of Koram macaques

Jonathan Reeves

2022-04-13

Model Specification

```
model_spec <- alist(  
  U ~ dbinom(1, p),  
  logit(p) <- a[H] + h[H] * bC * C,  
  a[H] ~ dnorm(0,1.5),  
  h[H] ~ dnorm(0,1.5),  
  bC ~ dlnorm(0,.3)  
)
```

Application of model to the social learning condition

Diagnostic Run with 1 Chain

```
## Running MCMC with 1 chain, with 1 thread(s) per chain...  
##  
## Chain 1 Iteration:    1 / 2000 [  0%] (Warmup)  
## Chain 1 Iteration:   100 / 2000 [  5%] (Warmup)  
## Chain 1 Iteration:   200 / 2000 [ 10%] (Warmup)  
## Chain 1 Iteration:   300 / 2000 [ 15%] (Warmup)  
## Chain 1 Iteration:   400 / 2000 [ 20%] (Warmup)  
## Chain 1 Iteration:   500 / 2000 [ 25%] (Warmup)  
## Chain 1 Iteration:   600 / 2000 [ 30%] (Warmup)  
## Chain 1 Iteration:   700 / 2000 [ 35%] (Warmup)  
## Chain 1 Iteration:   800 / 2000 [ 40%] (Warmup)  
## Chain 1 Iteration:   900 / 2000 [ 45%] (Warmup)  
## Chain 1 Iteration:  1000 / 2000 [ 50%] (Warmup)  
## Chain 1 Iteration:  1001 / 2000 [ 50%] (Sampling)  
## Chain 1 Iteration:  1100 / 2000 [ 55%] (Sampling)  
## Chain 1 Iteration:  1200 / 2000 [ 60%] (Sampling)  
## Chain 1 Iteration:  1300 / 2000 [ 65%] (Sampling)  
## Chain 1 Iteration:  1400 / 2000 [ 70%] (Sampling)  
## Chain 1 Iteration:  1500 / 2000 [ 75%] (Sampling)  
## Chain 1 Iteration:  1600 / 2000 [ 80%] (Sampling)  
## Chain 1 Iteration:  1700 / 2000 [ 85%] (Sampling)  
## Chain 1 Iteration:  1800 / 2000 [ 90%] (Sampling)  
## Chain 1 Iteration:  1900 / 2000 [ 95%] (Sampling)  
## Chain 1 Iteration:  2000 / 2000 [100%] (Sampling)  
## Chain 1 finished in 0.3 seconds.
```

No errors or warnings returned.

Run again for diagnostics

The model is re-run using 3 chains to ensure that the posterior distributions is appropriately sampled.

```
## Running MCMC with 3 parallel chains, with 1 thread(s) per chain...
```

```
##
## Chain 1 Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 1 Iteration:   100 / 2000 [  5%] (Warmup)
## Chain 1 Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 1 Iteration:   300 / 2000 [ 15%] (Warmup)
## Chain 1 Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 1 Iteration:   500 / 2000 [ 25%] (Warmup)
## Chain 1 Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 1 Iteration:   700 / 2000 [ 35%] (Warmup)
## Chain 1 Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 2 Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 2 Iteration:   100 / 2000 [  5%] (Warmup)
## Chain 2 Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 2 Iteration:   300 / 2000 [ 15%] (Warmup)
## Chain 2 Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 2 Iteration:   500 / 2000 [ 25%] (Warmup)
## Chain 2 Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 2 Iteration:   700 / 2000 [ 35%] (Warmup)
## Chain 3 Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 3 Iteration:   100 / 2000 [  5%] (Warmup)
## Chain 3 Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 3 Iteration:   300 / 2000 [ 15%] (Warmup)
## Chain 3 Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 3 Iteration:   500 / 2000 [ 25%] (Warmup)
## Chain 3 Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 3 Iteration:   700 / 2000 [ 35%] (Warmup)
## Chain 1 Iteration:   900 / 2000 [ 45%] (Warmup)
## Chain 1 Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 1 Iteration:  1001 / 2000 [ 50%] (Sampling)
## Chain 1 Iteration:  1100 / 2000 [ 55%] (Sampling)
## Chain 1 Iteration:  1200 / 2000 [ 60%] (Sampling)
## Chain 1 Iteration:  1300 / 2000 [ 65%] (Sampling)
## Chain 1 Iteration:  1400 / 2000 [ 70%] (Sampling)
## Chain 1 Iteration:  1500 / 2000 [ 75%] (Sampling)
## Chain 1 Iteration:  1600 / 2000 [ 80%] (Sampling)
## Chain 1 Iteration:  1700 / 2000 [ 85%] (Sampling)
## Chain 1 Iteration:  1800 / 2000 [ 90%] (Sampling)
## Chain 2 Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 2 Iteration:   900 / 2000 [ 45%] (Warmup)
## Chain 2 Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 2 Iteration:  1001 / 2000 [ 50%] (Sampling)
## Chain 2 Iteration:  1100 / 2000 [ 55%] (Sampling)
## Chain 2 Iteration:  1200 / 2000 [ 60%] (Sampling)
## Chain 2 Iteration:  1300 / 2000 [ 65%] (Sampling)
## Chain 2 Iteration:  1400 / 2000 [ 70%] (Sampling)
## Chain 2 Iteration:  1500 / 2000 [ 75%] (Sampling)
## Chain 2 Iteration:  1600 / 2000 [ 80%] (Sampling)
## Chain 3 Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 3 Iteration:   900 / 2000 [ 45%] (Warmup)
## Chain 3 Iteration:  1000 / 2000 [ 50%] (Warmup)
```

```
## Chain 3 Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 3 Iteration: 1100 / 2000 [ 55%] (Sampling)
## Chain 3 Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 3 Iteration: 1300 / 2000 [ 65%] (Sampling)
## Chain 3 Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 3 Iteration: 1500 / 2000 [ 75%] (Sampling)
## Chain 1 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 1 Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 2 Iteration: 1700 / 2000 [ 85%] (Sampling)
## Chain 2 Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 2 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 2 Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 3 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 3 Iteration: 1700 / 2000 [ 85%] (Sampling)
## Chain 3 Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 3 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 3 Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 1 finished in 0.3 seconds.
## Chain 2 finished in 0.3 seconds.
## Chain 3 finished in 0.3 seconds.
##
## All 3 chains finished successfully.
## Mean chain execution time: 0.3 seconds.
## Total execution time: 0.5 seconds.
```

Summary of outcome

```
row_names <- c("Intercept: Common-Like, not-tool-using age",
               "Intercept: Hybrid-Like, not-tool-using age",
               "Intercept: Common-Like, tool-using age",
               "Intercept: Hybrid-Like, tool-using age",
               "Hair Pattern: Common-Like, not-tool-using age",
               "Hair Pattern: Hybrid-Like, not-tool-using age",
               "Hair Pattern: Common-Like, tool-using age",
               "Hair Pattern: Hybrid-Like, tool-using age",
               "Centrality")

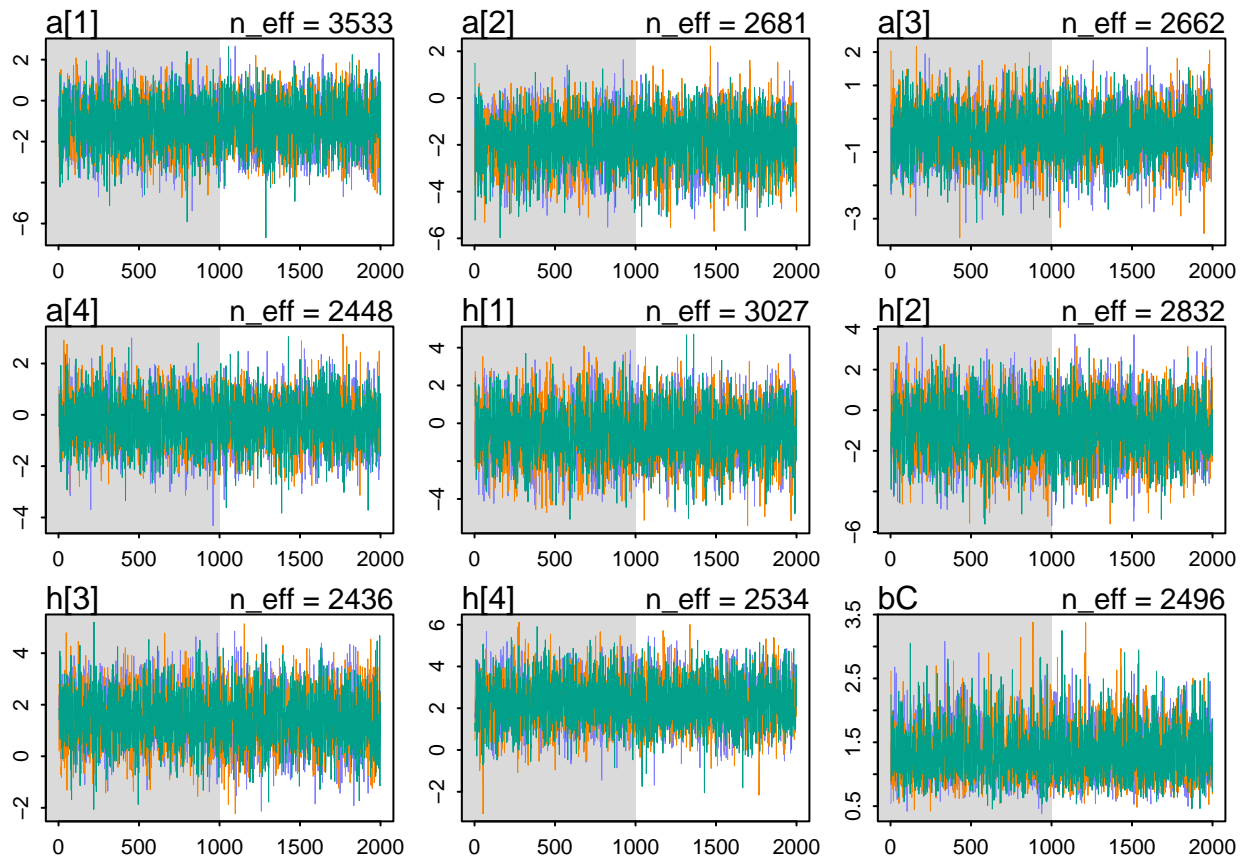
koram_model_sum <- precis(koram.m, depth = 2)
koram_model_link <- apply(koram_model_sum, 2, inv_logit)
koram_model_link <- round(koram_model_link, 3)
koram_model_link[,5] <- round(koram_model_sum$n_eff)
rownames(koram_model_link) <- row_names

write.csv(koram_model_link, "koram_ABM_Marginal_Distributions_2.csv")
```

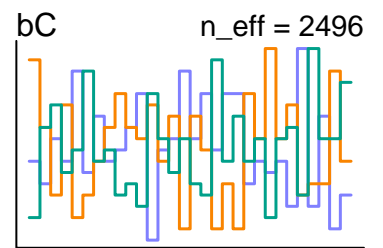
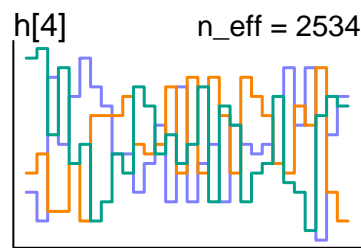
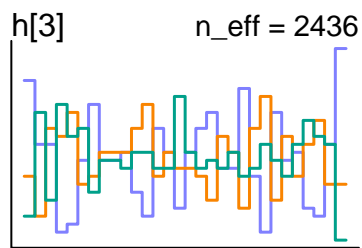
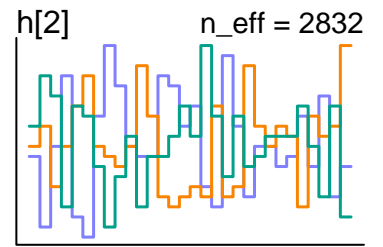
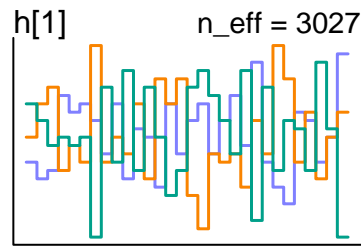
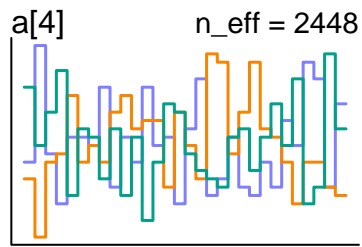
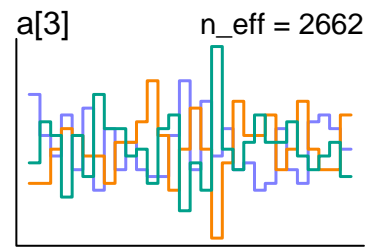
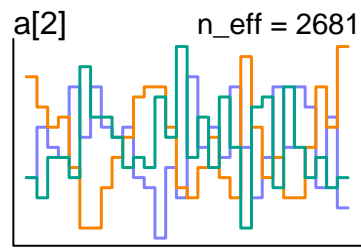
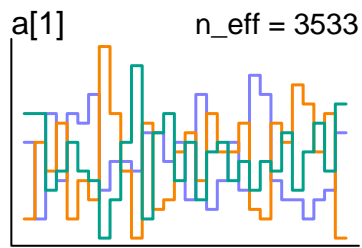
Markov Chain Performance

```
traceplot(koram.m, pars = c("a[1]",
                             "a[2]",
                             "a[3]",
                             "a[4]",
                             "h[1]",
                             "h[2]",
```

```
"h[3]",
"h[4]",
"bC"))
```



```
trankplot(koram.m, pars = c("a[1]",
                             "a[2]",
                             "a[3]",
                             "a[4]",
                             "h[1]",
                             "h[2]",
                             "h[3]",
                             "h[4]",
                             "bC"))
```



Trank Plot