

## Methods 4 – Portfolio Assignment 3

### Exercise 11H3

The data contained in `library(MASS); data(eagles)` are records of salmon pirating attempts by Bald Eagles in Washington State. See `?eagles` for details. While one eagle feeds, sometimes another will swoop in and try to steal the salmon from it. Call the feeding eagle the “victim” and the thief the “pirate.” Use the available data to build a binomial GLM of successful pirating attempts.

```
pacman::p_load(MASS, rethinking, tidyverse, tidybayes, tidybayes.rethinking,
               boot, knitr)
set_cmdstan_path('C:/cmdstan/cmdstan-2.31.0')
```

```
## CmdStan path set to: C:/cmdstan/cmdstan-2.31.0
```

```
data(eagles)
eagles <- eagles
```

#### (a) (MST)

Consider the following model:

$$y_i \sim \text{Binomial}(n_i, p_i)$$

$$\log \frac{p_i}{1 - p_i} = \alpha + \beta_P P_i + \beta_V V_i + \beta_A A_i$$

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

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

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

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

where  $y$  is the number of successful attempts,  $n$  is the total number of attempts,  $P$  is a dummy variable indicating whether or not the pirate had large body size,  $V$  is a dummy variable indicating whether or not the victim had large body size, and finally  $A$  is a dummy variable indicating whether or not the pirate was an adult. Fit the model above to the eagles data, using both `quap` and `ulam`. Is the quadratic approximation okay?

```
m_eagles_quap <- quap(
  alist(
    y ~ dbinom( n , p ) ,
    logit(p) <- a + Bp[P] + Bv[V] + Ba[A],
    a ~ dnorm(0,1.5),
    Bp[P] ~ dnorm(0,0.5),
    Bv[V] ~ dnorm(0,0.5),
    Ba[A] ~ dnorm(0,0.5)
```

```
) , data=eagles )
```

```
precis( m_eagles_quap , depth=2 )
```

```
##           mean          sd        5.5%        94.5%
## a          0.5632082 0.5979207 -0.3923846  1.5188010
## Bp[1]      1.0757669 0.3908947  0.4510417  1.7004921
## Bp[2]     -1.0131877 0.3917740 -1.6393182 -0.3870573
## Bv[1]     -1.0632542 0.3956189 -1.6955296 -0.4309788
## Bv[2]      1.1258329 0.3992853  0.4876979  1.7639679
## Ba[1]      0.4162567 0.3895095 -0.2062547  1.0387680
## Ba[2]     -0.3536776 0.3896532 -0.9764187  0.2690636
```

```
m_eagles_ulam <- ulam(
  alist(
    y ~ dbinom( n , p ) ,
    logit(p) <- a + Bp[P] + Bv[V] + Ba[A],
    a ~ dnorm(0,1.5),
    Bp[P] ~ dnorm(0,0.5),
    Bv[V] ~ dnorm(0,0.5),
    Ba[A] ~ dnorm(0,0.5)
  ) , data=eagles , chains=4 , log_lik=TRUE )
```

```
precis(m_eagles_ulam, depth = 2)
```

```
##           mean          sd        5.5%        94.5%      n_eff      Rhat4
## a          0.5876014 0.6191305 -0.3672652  1.5736589  758.1296  1.006401
## Bp[1]      1.0725600 0.3752147  0.4767108  1.6727354 1260.4115  1.002895
## Bp[2]     -1.0391475 0.3836097 -1.6500670 -0.4286703 1289.7846  1.006410
## Bv[1]     -1.0829893 0.3986640 -1.7162030 -0.4209534 1376.0372  1.002341
## Bv[2]      1.1649696 0.3968714  0.5402599  1.7964216 1444.4825  0.999414
## Ba[1]      0.4342904 0.3994161 -0.2063887  1.0864017 1067.3794  1.001992
## Ba[2]     -0.3508935 0.3901568 -0.9812906  0.2759091 1290.1473  1.001079
```

```
compare(m_eagles_quap, m_eagles_ulam)
```

```
## Warning in compare(m_eagles_quap, m_eagles_ulam): Not all model fits of same class.
## This is usually a bad idea, because it implies they were fit by different algorithms.
## Check yourself, before you wreck yourself.
```

```
##           WAIC          SE    dWAIC          dSE    pWAIC    weight
## m_eagles_ulam 45.68741 7.157177 0.000000          NA 6.386959 0.6495134
## m_eagles_quap 46.92122 7.406757 1.233802 0.5289268 6.629453 0.3504866
```

Comparing the ulam and quap model, we see that the ulam is slightly better. We can conclude, that the estimates are very similar and thus the quadratic approximation is okay.

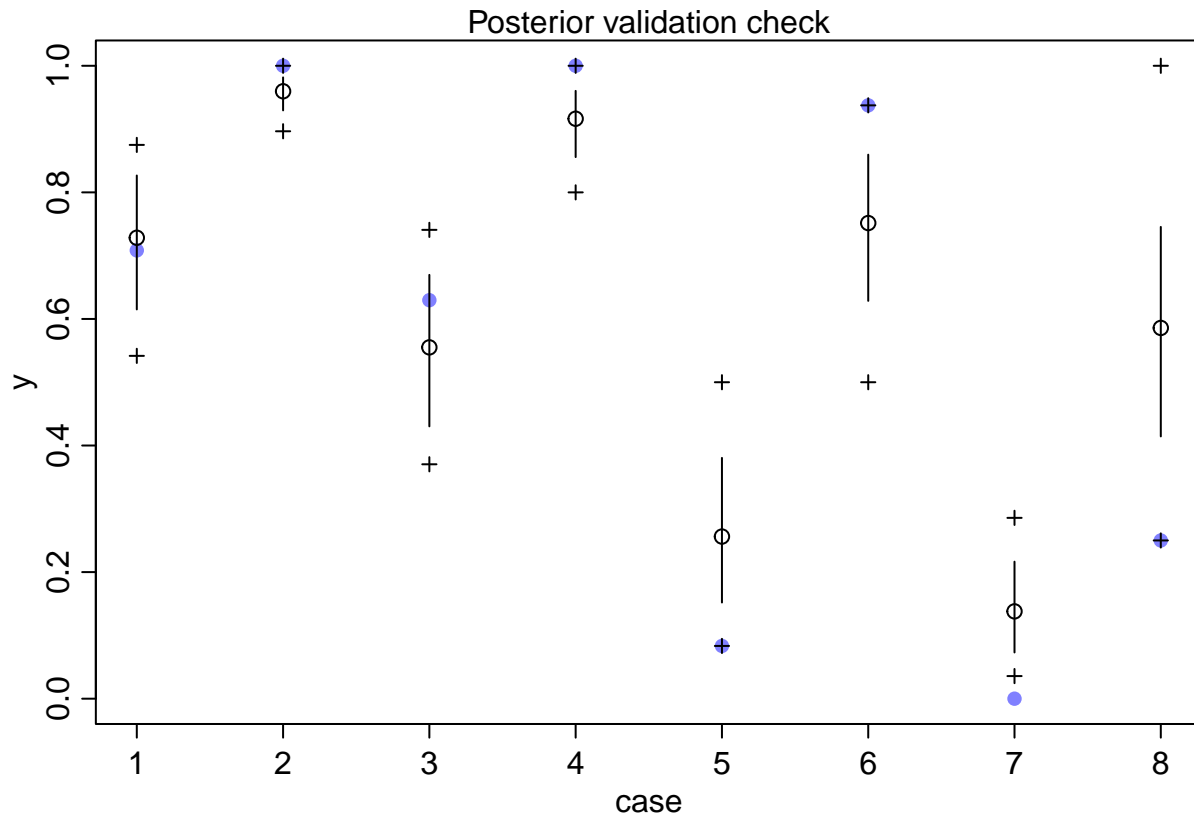
## (b) (LR)

Now interpret the estimates. If the quadratic approximation turned out okay, then it's okay to use the quap estimates. Otherwise stick to ulam estimates. Then plot the posterior predictions. Compute and display

both (1) the predicted probability of success and its 89% interval for each row (i) in the data, as well as (2) the predicted success count and its 89% interval. What different information does each type of posterior prediction provide?

First, we plot the predicted probability of successes and their 89% intervals.

```
postcheck(m_eagles_ulam, prob = 0.89, window = 20, n = 1000, col=rangi2)
```



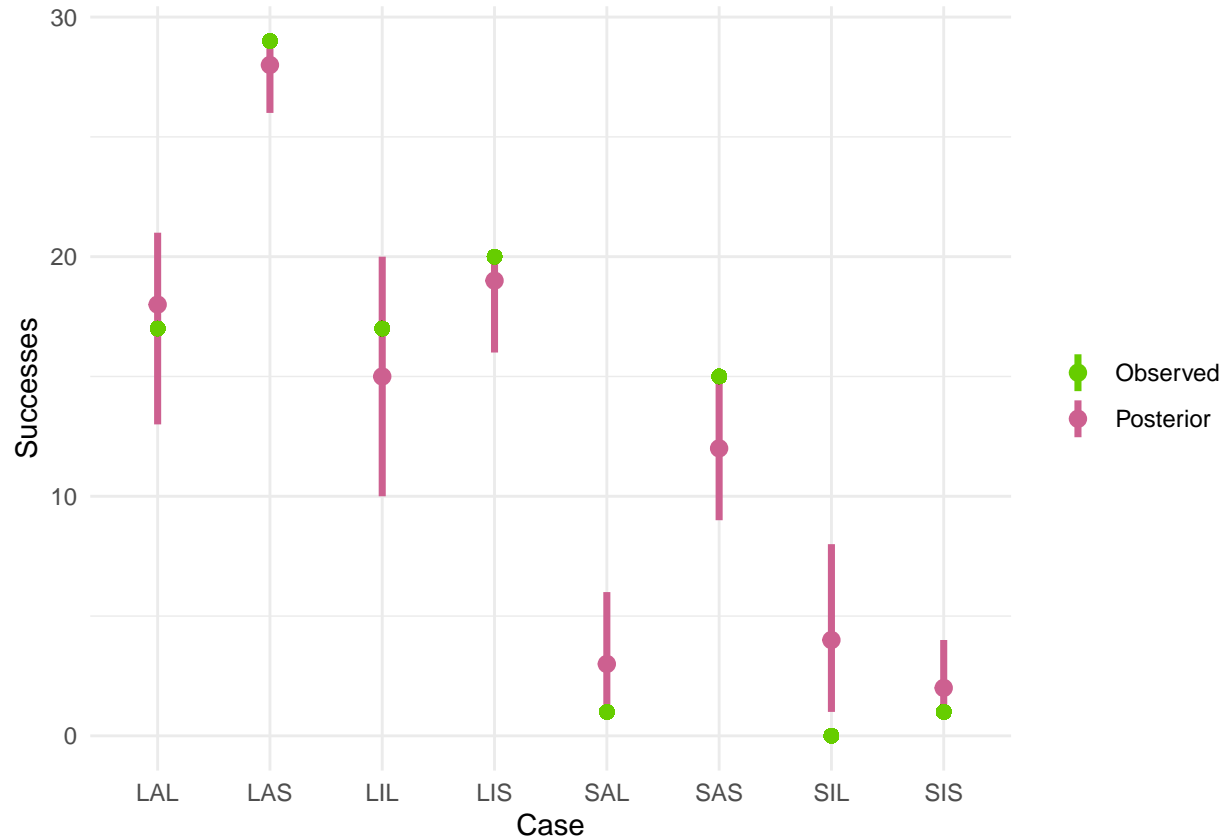
Blue dots = observed data White dots = posterior estimates Lines at white dots = 89 % intervals

Second, we plot the predicted success count and its 89% intervals.

```
predictions <- tidybayes::predicted_draws(
  m_eagles_ulam,
  eagles,
  value = ".prediction",
  ndraws = NULL,
  seed = NULL,
  re_formula = NULL
)
```

```
predictions %>%
  mutate(label = paste0(P, A, V)) %>%
  ggplot(aes(x = label, y = .prediction)) +
  stat_pointinterval(aes(color = "Posterior"), .width = 0.89, size = 5) +
  geom_point(data = predictions, size = 2,
    aes(x = paste0(P, A, V), y = y, color = "Observed")) +
```

```
scale_color_manual(values = c("Posterior" = "hotpink3",
                              "Observed" = "chartreuse3"),
                  name = NULL) +
labs(x = "Case", y = "Successes") + theme_minimal()
```



The two posterior prediction plots provide different information. The plot showing the predicted probability of success visualizes the relative values in terms of probability, where the plot for predicted success counts visualizes absolute values.

### (c) (SM & TI)

Now try to improve the model. Consider an interaction between the pirate's size and age (immature or adult). Compare this model to the previous one, using WAIC. Interpret.

```
set.seed(123)
eagles <- eagles %>%
  mutate(dP = ifelse(P == "L", 1, 2),
         dV = ifelse(V == "L", 1, 2),
         dA = ifelse(A == "A", 1, 2))

m2_eagles_ulam <- ulam(
  alist(
    y ~ dbinom( n , p ) ,
    logit(p) <- a[dP,dA] + Bv[dV] ,
```

```

      Bv[dV] ~ dnorm(0,0.5),
      matrix[dP,dA]:a ~ normal(0,1)
    ) , data=eagles , chains=4 , log_lik=TRUE )

m3_eagles_ulam <- ulam(
  alist(
    y ~ dbinom( n , p ) ,
    logit(p) <- a[dP, dV] + Ba[dA],
    Ba[dA] ~ dnorm(0,0.5),
    matrix[dP,dV]:a ~ normal(0,1)
  ) , data=eagles , chains=4 , log_lik=TRUE )

```

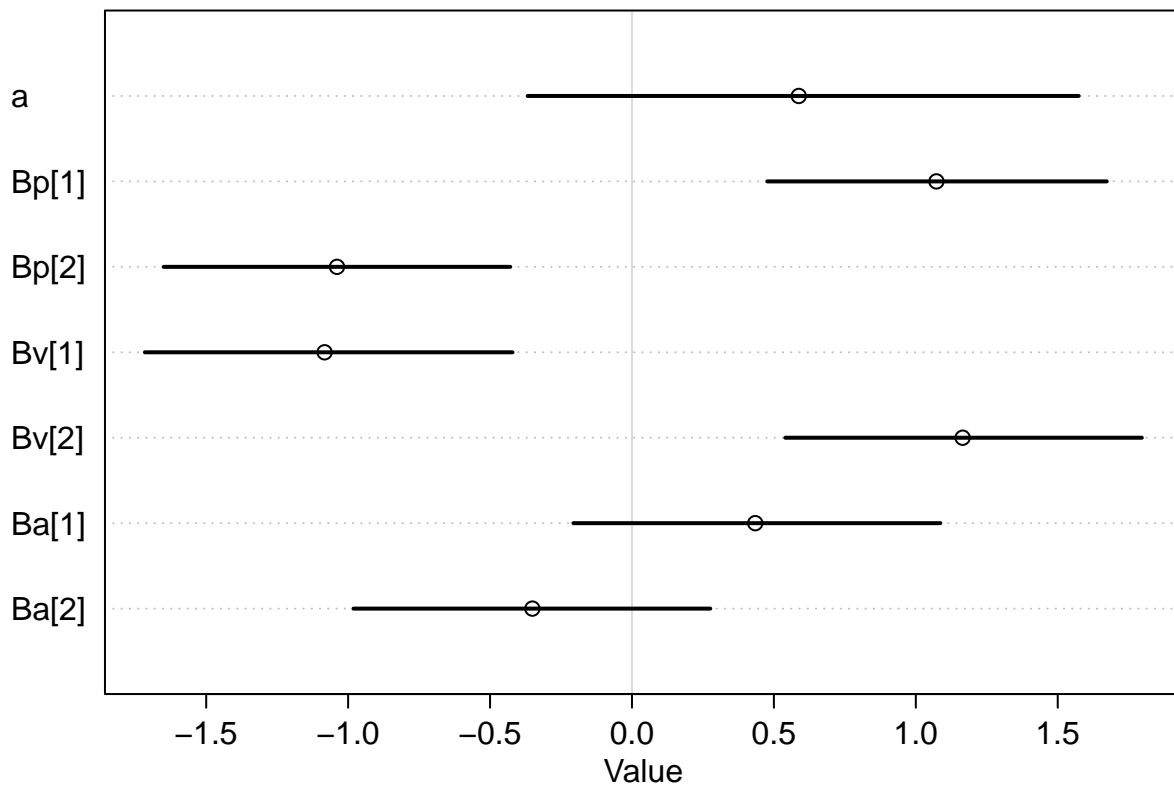
```
compare(m_eagles_ulam, m2_eagles_ulam, m3_eagles_ulam)
```

```

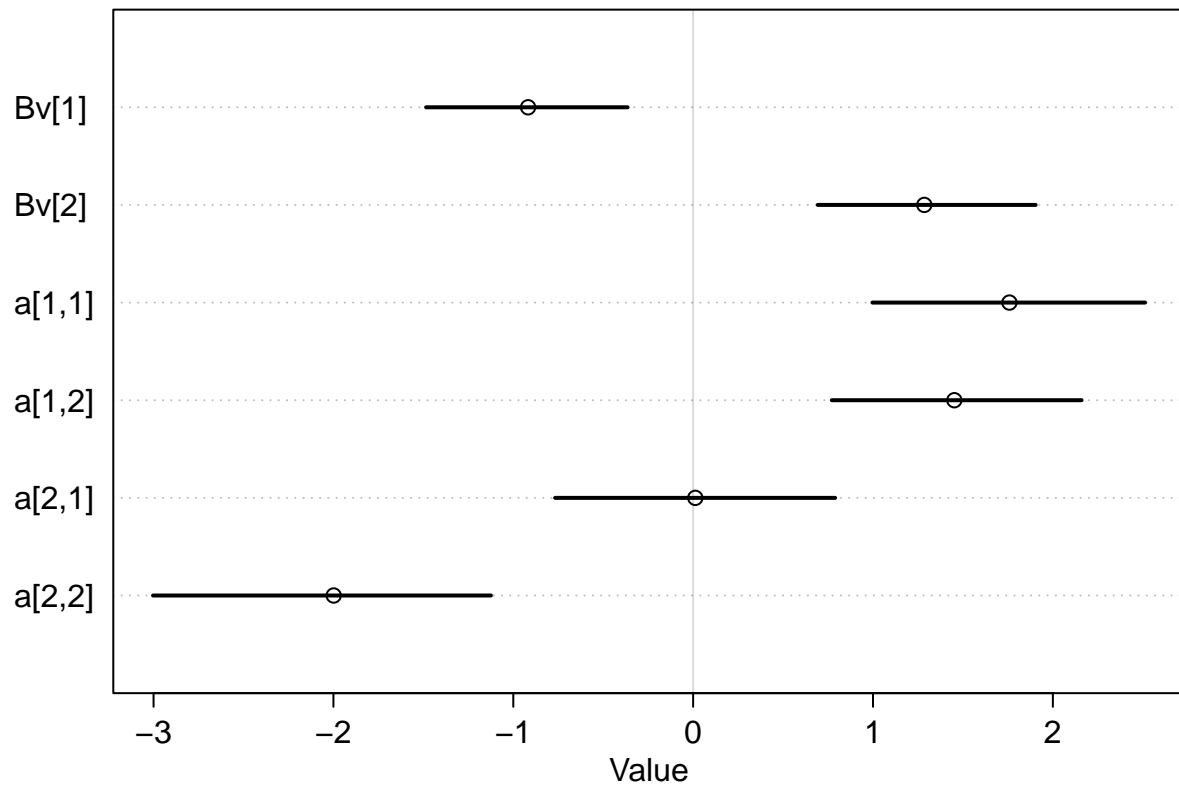
##              WAIC      SE    dWAIC      dSE    pWAIC    weight
## m3_eagles_ulam 37.90451 3.819052 0.0000000    NA 5.421063 0.57092080
## m2_eagles_ulam 38.53080 5.094141 0.6262984 6.602427 5.726650 0.41742349
## m_eagles_ulam  45.68741 7.157177 7.7829075 6.486811 6.386959 0.01165572

```

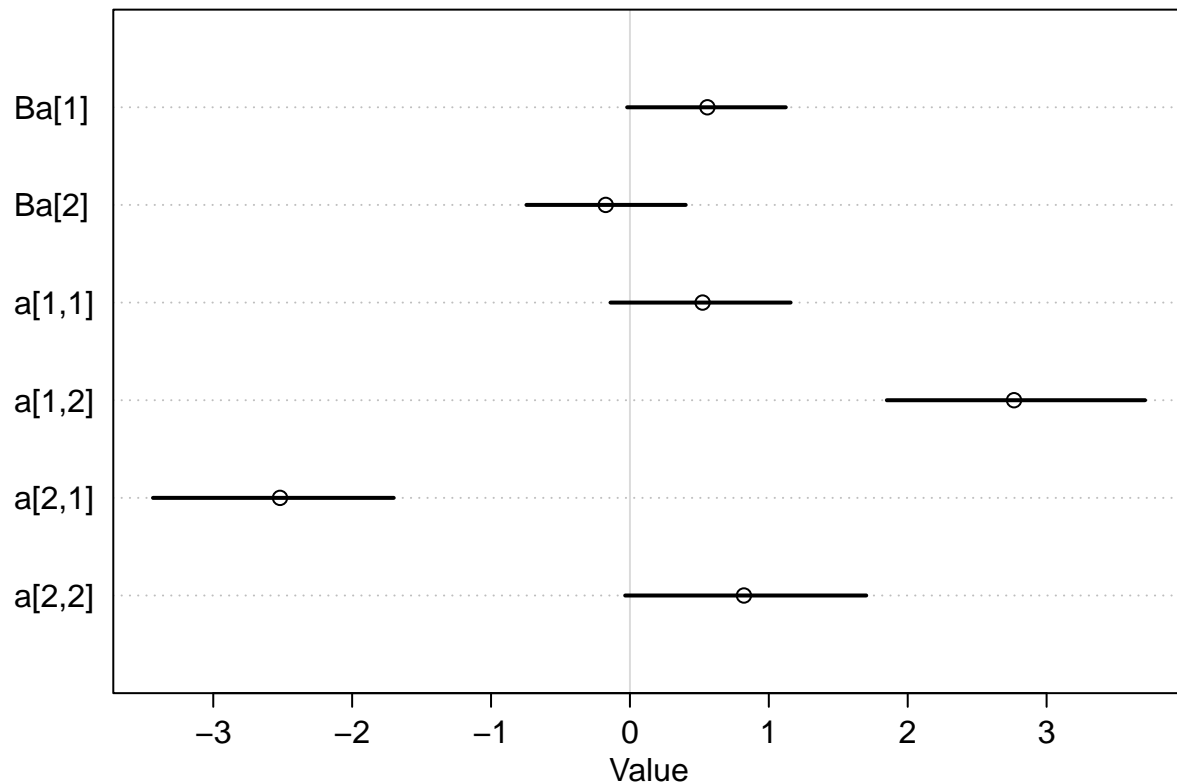
```
plot(precis(m_eagles_ulam, depth = 2))
```



```
plot(precis(m2_eagles_ulam, depth = 4))
```



```
plot(precis(m3_eagles_ulam, depth = 4))
```



From comparing the models using WAIC, we can conclude that model 2, which accounts for an interaction between age and body size of the pirate, has lowest WAIC and therefore the best performance. However, model 3, which accounts for an interaction between body size of the pirate and body size of the victim, performs almost as well. The difference between them is below 1 point on the WAIC scale and they have similar standard deviations.