hw5

GISH

2019/5/14

# (a)Fit the model above to the eagles data, using rstan. Is the quadratic approximation okay?

d$pd = ifelse(d$P == "L",1,0) #pirate body dummy  
d$vd = ifelse(d$V == "L",1,0) # victim body dummy  
d$ad = ifelse(d$A == "A",1,0) #pirate adult dummy  
  
  
m <- map(  
 alist(  
 y ~ dbinom(n,p),  
 logit(p) <- a + bpd\*pd + bvd\*vd + bad\*ad,  
 a ~ dnorm(0,10),  
 c(bpd,bvd,bad) ~ dnorm( 0 , 5)  
 ), data=d )  
  
  
dstan <- d  
  
mstan <- map2stan( m , data=dstan , iter=1e4 , warmup=1000 )

##   
## SAMPLING FOR MODEL 'e0f5f033610516b3f64847fc94127ad3' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## 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)  
## Chain 1:   
## Chain 1: Elapsed Time: 0.127 seconds (Warm-up)  
## Chain 1: 1.306 seconds (Sampling)  
## Chain 1: 1.433 seconds (Total)  
## Chain 1:

## Computing WAIC

precis(m)

## mean sd 5.5% 94.5%  
## a 0.5915206 0.6622710 -0.4669164 1.649957  
## bpd 4.2418124 0.8960170 2.8098042 5.673821  
## bvd -4.5925888 0.9613914 -6.1290779 -3.056100  
## bad 1.0814123 0.5339197 0.2281055 1.934719

precis(mstan)

## mean sd 5.5% 94.5% n\_eff Rhat  
## a 0.6535672 0.6932935 -0.4320963 1.799726 4205.251 0.9999999  
## bpd 4.6368540 1.0126906 3.2081962 6.436570 3578.579 0.9998909  
## bvd -5.0421766 1.0683652 -6.8992201 -3.493879 3551.547 0.9998969  
## bad 1.1341734 0.5435467 0.2784386 2.025343 4475.230 1.0002512

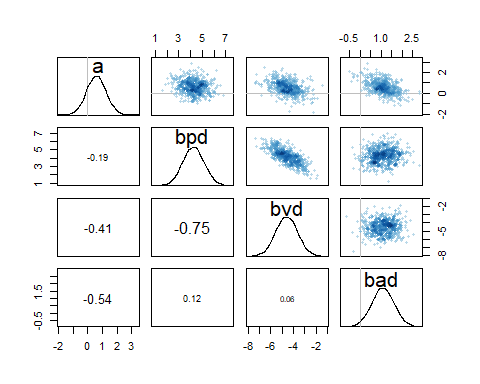
compare(mstan,m)

## Warning in compare(mstan, m): 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.

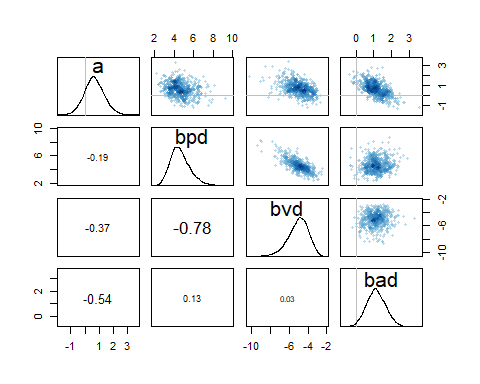
## WAIC pWAIC dWAIC weight SE dSE  
## m 98.58072 3.767801 0.00000000 0.5087223 12.49560 NA  
## mstan 98.65050 4.054796 0.06978582 0.4912777 13.33223 1.100571

警告訊息是因為新版rethinking package的關係，但事實上mstan與m用的模型是一樣的 從模型上可以看出兩個模型的參數估計值幾乎沒差多少 且從compare方法也可以看出，從資訊值來看兩者的資訊含量其實差不多，兩者所佔的weight幾乎接近0.5 所以二項式接近模型(m)在這邊是可以用的模型

pairs(m)



pairs(mstan)



但從圖形上可以看出其實stan做出來的估計，會呈現出較偏斜的分配(bpd,bvd)，可能相較於使用二項式分配，較能呈現真實分布情況

# (b) Now interpret the estimates.Then plot the posterior predictions. Compute and

# display both (1) the predicted probability of successand 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?

由mstan的模型來看 bpd 之mean 為4.64 表當pirate的身體為large時，log(p/(1-p))平均來說會增加4.64 =>即p會增加約0.9904% bvd -5.04 表當victim的身體為large時，log(p/(1-p))平均來說會減少5.04 => 以此類推 p 會增加約0.0064% bad 1.13 表當pirate的年齡為Adult時，log(p/(1-p))平均來說會增加1.13 => p會增加0.753%

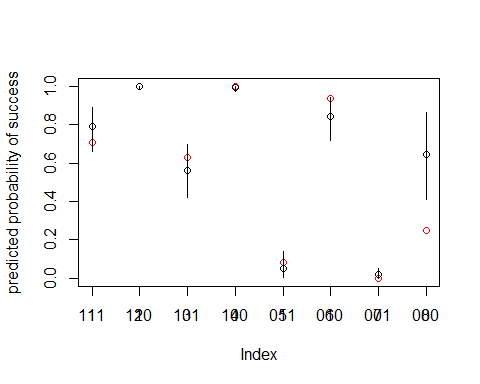
p <- link(mstan)

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

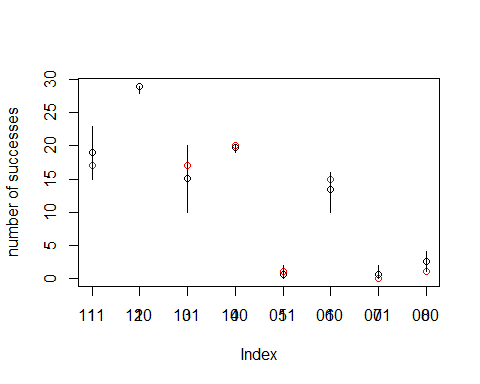
y <- sim(mstan)

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

p.mean <- apply(X = p, MARGIN = 2, FUN = mean)  
p.PI <- apply(X = p, MARGIN = 2, FUN = PI)  
y.mean <- apply(X = y, MARGIN = 2, FUN = mean)  
y.PI <- apply(X = y, MARGIN = 2, FUN = PI)  
  
d$proportion <- d$y / d$n  
plot(d$proportion, col='red', ylab="predicted probability of success") #紅為模擬出來的比例  
axis(1, at=1:8, labels=c( "111","110","101","100","011","010","001","000" ))  
points( 1:8 , p.mean ) #黑為預測出來的比例  
for ( i in 1:8 ) lines( c(i, i), p.PI[,i] )



plot(d$y, col='red', ylab="number of successes") #紅為模擬出來的比例  
axis(1, at=1:8, labels=c( "111","110","101","100","011","010","001","000" ))  
points( 1:8 , y.mean ) #黑為預測出來的比例  
for ( i in 1:8 ) lines( c(i, i), y.PI[,i] )

 差別就是一個是比例，一個是count成功的數量 所以count成功數量的模型，其y值的大小還會受到number of trails即n的影響 n越大，成功的次數當然越多

# (c)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 to interpret.

mstan.i <- map2stan(  
 alist(  
 y ~ dbinom(n, p),  
 logit(p) <- a + bpd\*pd + bvd\*vd + bad\*ad + bpa\*pd\*ad,  
 a ~ dnorm(0,10),  
 c(bpd,bvd,bad,bpa) ~ dnorm( 0 , 5)  
 ),  
 data = dstan, warmup = 1000, iter = 1e4  
)

##   
## SAMPLING FOR MODEL '1202891490222b2a261c99ae75b05692' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## 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)  
## Chain 1:   
## Chain 1: Elapsed Time: 0.184 seconds (Warm-up)  
## Chain 1: 1.606 seconds (Sampling)  
## Chain 1: 1.79 seconds (Total)  
## Chain 1:

## Computing WAIC

compare(mstan, mstan.i)

## WAIC pWAIC dWAIC weight SE dSE  
## mstan.i 93.87099 4.689315 0.000000 0.91604264 12.68414 NA  
## mstan 98.65050 4.054796 4.779508 0.08395736 13.33223 4.696601

precis(mstan.i)

## mean sd 5.5% 94.5% n\_eff Rhat  
## a -0.7903742 1.059759 -2.493431 0.842413 2653.372 1.000009  
## bpd 6.5493997 1.423491 4.433751 8.973470 2272.990 1.000622  
## bvd -5.2560466 1.150657 -7.218209 -3.582683 3254.339 1.000527  
## bad 3.4426961 1.251265 1.514551 5.491455 2312.457 1.000245  
## bpa -2.9654414 1.373968 -5.160384 -0.826887 2371.634 1.000394

precis(mstan)

## mean sd 5.5% 94.5% n\_eff Rhat  
## a 0.6535672 0.6932935 -0.4320963 1.799726 4205.251 0.9999999  
## bpd 4.6368540 1.0126906 3.2081962 6.436570 3578.579 0.9998909  
## bvd -5.0421766 1.0683652 -6.8992201 -3.493879 3551.547 0.9998969  
## bad 1.1341734 0.5435467 0.2784386 2.025343 4475.230 1.0002512

由compare可看出mstan.i的資訊含量較高，其weight佔了0.91 再由precis可看出，其實pirate的年齡大小(bad)在mstan.i的模型較大，代表其實pirate的年齡大小其實算一個影響因素，但是因為交互作用的影響，即pirate 的體積，影響了，年齡對於掠奪成功率的影響，導致年齡在沒有交互作用的模型下(mstan)其平均只有1.13(相較於交互作用的模型平均3.41)