Assignment5\_B04704016林家毅

# Question 1 (a)

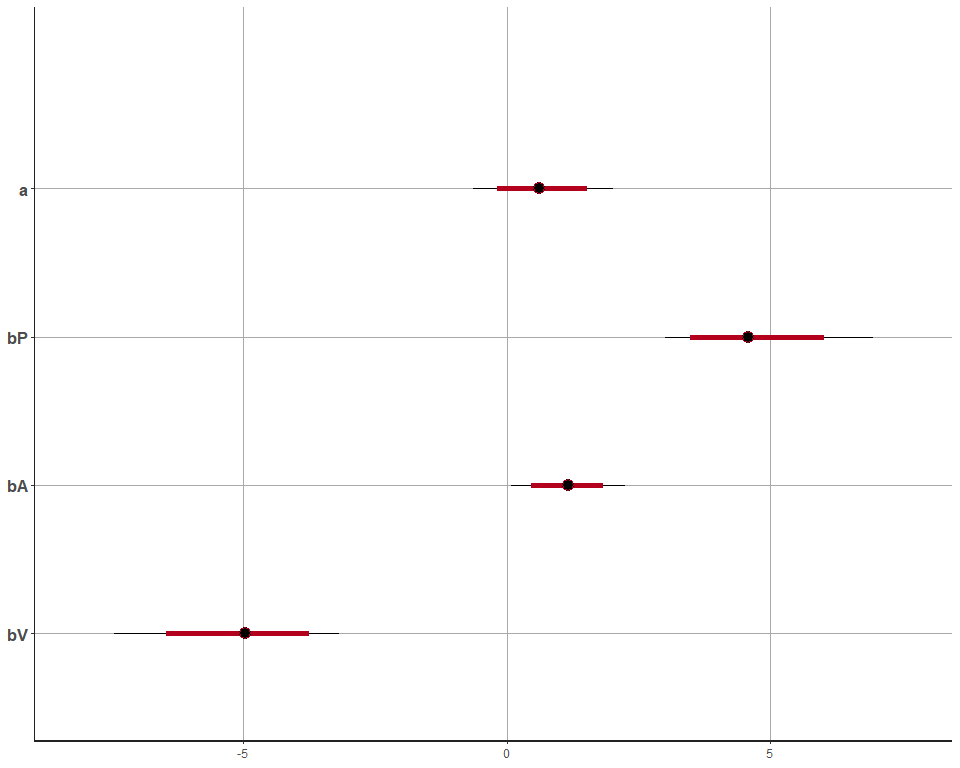
# Load data  
data( "eagles" )  
data <- eagles; rm( eagles )  
  
# Set dummy variables  
data$P <- ifelse( data$P=="L" , 1 , 0 )  
data$A <- ifelse( data$A=="A" , 1 , 0 )  
data$V <- ifelse( data$V=="L" , 1 , 0 )  
  
# Create data list for the stan model  
data.model <- list(  
 N = NROW( data ),  
 y = data$y,  
 n = data$n,  
 P = data$P,  
 A = data$A,  
 V = data$V  
)  
str( data.model )

## List of 6  
## $ N: int 8  
## $ y: int [1:8] 17 29 17 20 1 15 0 1  
## $ n: int [1:8] 24 29 27 20 12 16 28 4  
## $ P: num [1:8] 1 1 1 1 0 0 0 0  
## $ A: num [1:8] 1 1 0 0 1 1 0 0  
## $ V: num [1:8] 1 0 1 0 1 0 1 0

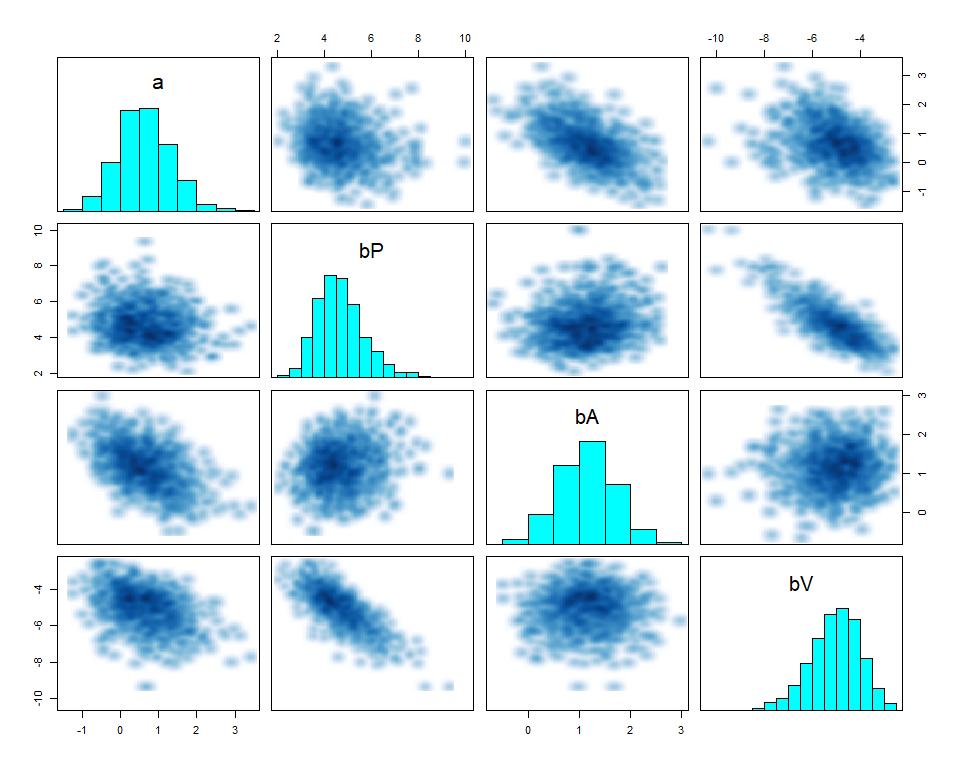
# Model code  
model.a.stan <- "  
data{  
 int<lower=1> N;  
 int y[N];  
 int n[N];  
 real P[N];  
 real A[N];  
 real V[N];  
}  
parameters{  
 real a;  
 real bP;  
 real bA;  
 real bV;  
}  
model{  
 vector[N] p;  
 bV ~ normal( 0 , 5 );  
 bA ~ normal( 0 , 5 );  
 bP ~ normal( 0 , 5 );  
 a ~ normal( 0 , 10 );  
 for ( i in 1:N ) {  
 p[i] = a + bP \* P[i] + bA \* A[i] + bV \* V[i];  
 p[i] = inv\_logit(p[i]);  
 }  
 y ~ binomial( n , p );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] p;  
 vector[N] c;  
 for ( i in 1:N ) {  
 p[i] = a + bP \* P[i] + bA \* A[i] + bV \* V[i];  
 p[i] = inv\_logit(p[i]);  
 c[i] = binomial\_rng( n[i] , p[i] );  
 log\_lik[i] = binomial\_lpmf( y[i] | n[i] , p[i] );  
 }  
}  
"  
  
# Fit stan model with model code and model data  
fit.a.stan <- stan( model\_code=model.a.stan , data=data.model ,   
 chains=4 , cores=4 , iter=1000 )  
  
# Show results  
print( fit.a.stan , probs=c(0.055,0.945) , pars=c("a","bP","bA","bV") )

## Inference for Stan model: 1d646e2f077e5c0304cfb44cd4f478e1.  
## 4 chains, each with iter=1000; warmup=500; thin=1;   
## post-warmup draws per chain=500, total post-warmup draws=2000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## a 0.63 0.02 0.69 -0.45 1.75 829 1.00  
## bP 4.67 0.04 1.02 3.28 6.40 779 1.00  
## bA 1.15 0.02 0.54 0.28 2.00 1206 1.00  
## bV -5.06 0.04 1.08 -6.89 -3.49 637 1.01  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 16 11:28:55 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

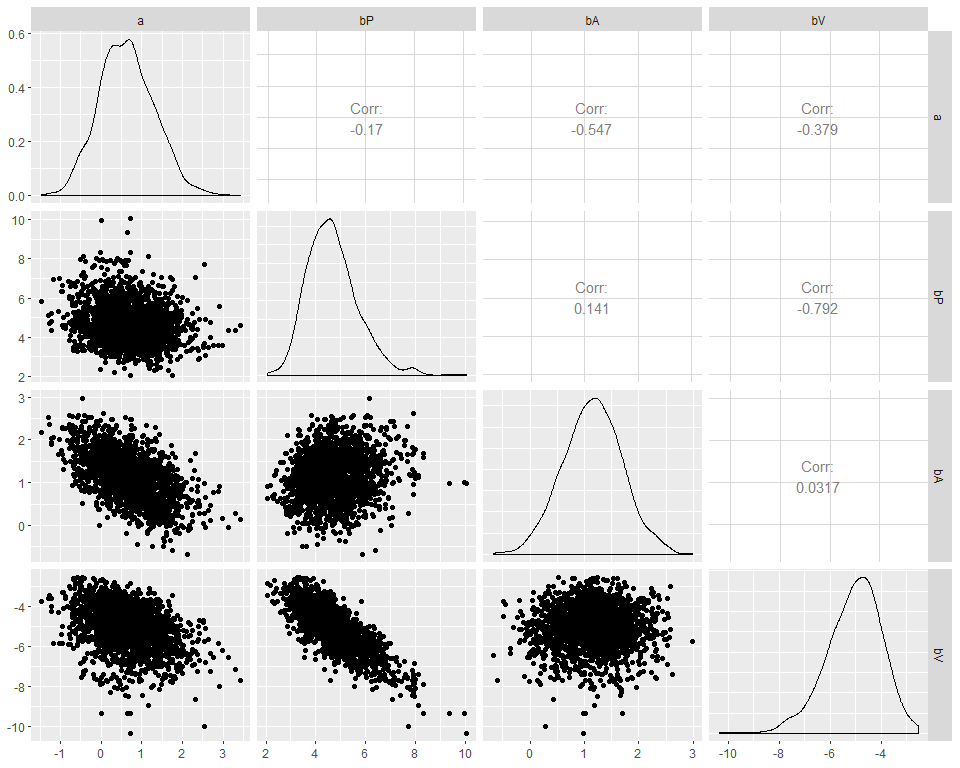
# Plot coeftab  
plot( fit.a.stan , pars=c("a","bP","bA","bV") )



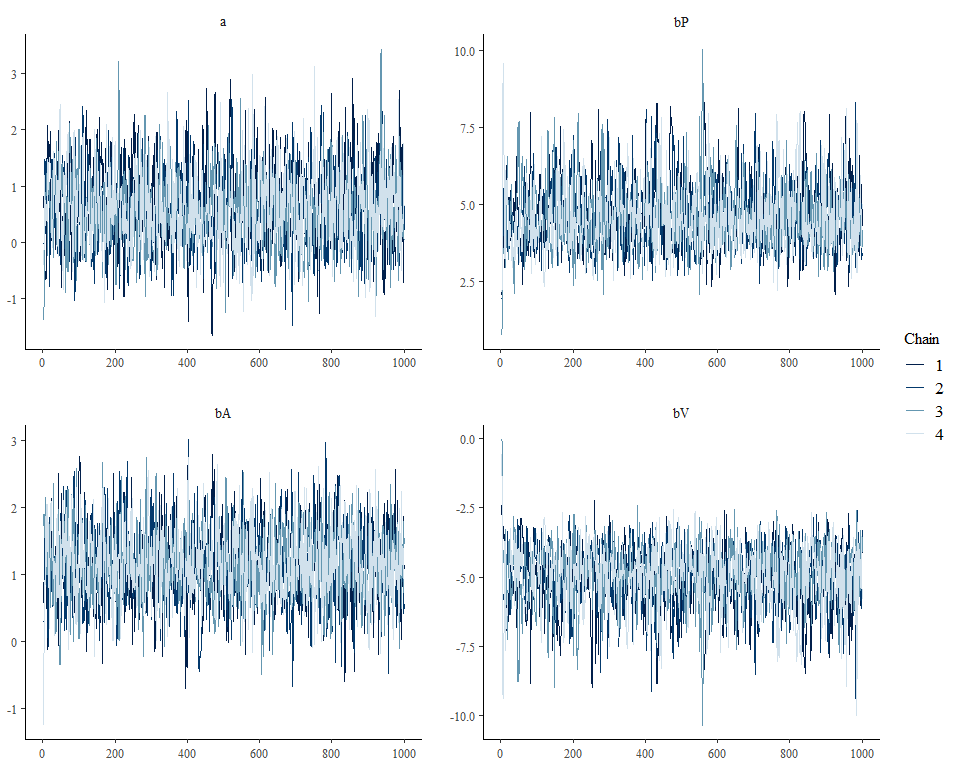
# Pairs plot  
pairs( fit.a.stan , pars=c("a","bP","bA","bV") )



ggpairs( data=as.data.frame( fit.a.stan ), columns=c(1:4))



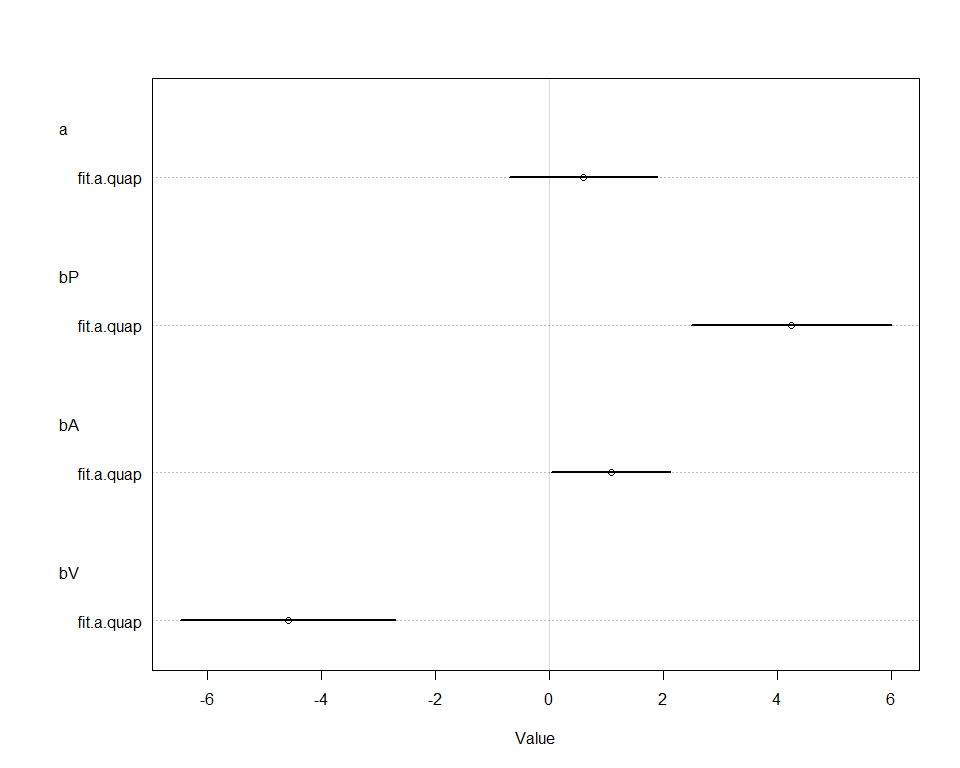
# Traceplot  
extract\_post <- rstan::extract( fit.a.stan , inc\_warmup=TRUE , permuted=FALSE)  
post\_traceplot <- mcmc\_trace( extract\_post , n\_warmup=0 , pars=c("a","bP","bA","bV"),  
 facet\_args=list( nrow=2 , labeller=label\_parsed ) )  
post\_traceplot



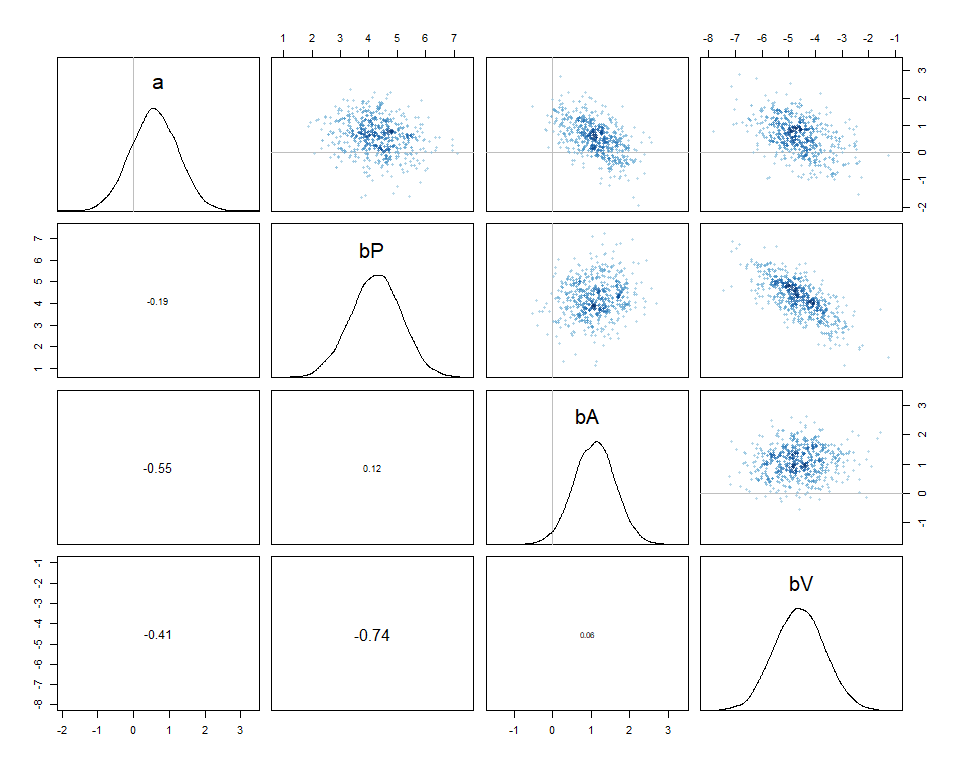
# Fit quadratic approximation model  
fit.a.quap <- map(  
 alist(  
 y ~ dbinom(n, p),  
 logit(p) <- a + bP\*P + bA\*A + bV\*V,  
 a ~ dnorm(0, 10),  
 bP ~ dnorm(0, 5),  
 bA ~ dnorm(0, 5),  
 bV ~ dnorm(0, 5)  
 ),  
 data=data  
)  
  
# Show results  
precis( fit.a.quap )

## mean sd 5.5% 94.5%  
## a 0.591542 0.6622736 -0.4668992 1.649983  
## bP 4.241822 0.8960180 2.8098126 5.673832  
## bA 1.081408 0.5339211 0.2280987 1.934717  
## bV -4.592604 0.9613943 -6.1290975 -3.056110

# Plot coeftab  
plot( coeftab( fit.a.quap ) )



# Pairs plot  
pairs( fit.a.quap )



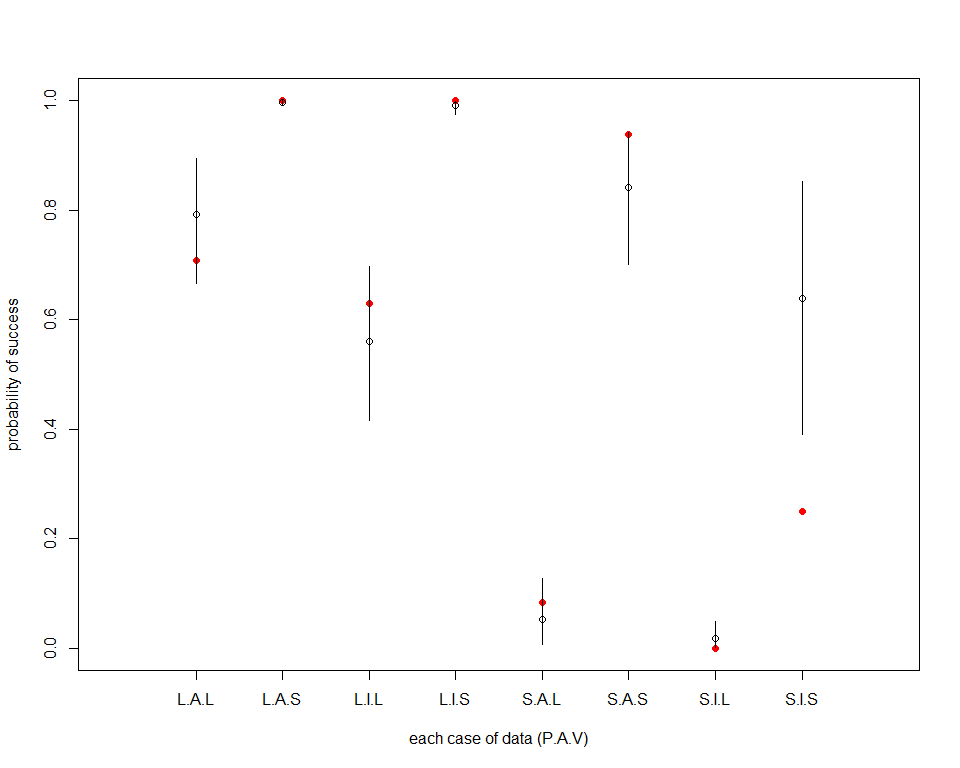
## Conclusion

From the pairs plot of the stan model, we can see the skewed distribution for both bP and bV. On the other hand, for the same parameters, the quadratic qpproximation model shows symmetric distribution. We would prefer the stan model as it is closer to what we expect.

# Question 1 (b)

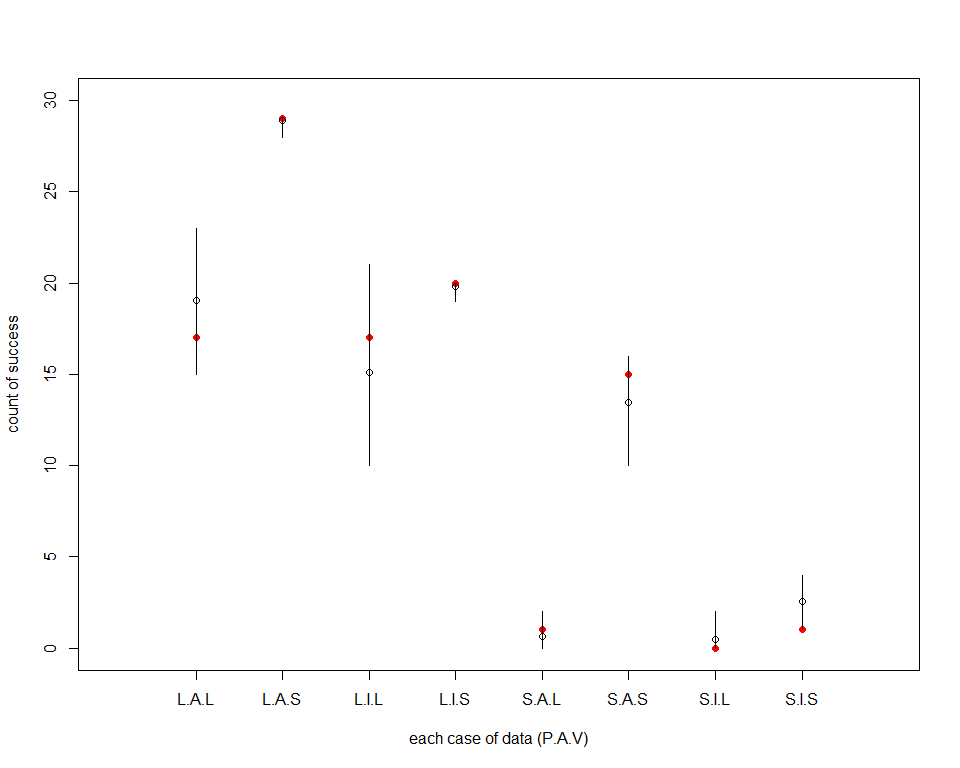
The predicted probability of success and the 89% interval for each row of data are as follows:

p <- as.data.frame(fit.a.stan)[,13:20]  
p.mean <- apply( p , 2 , mean )  
p.PI <- apply( p, 2, PI , prob=0.89 )  
data$prob\_success <- data$y / data$n  
plot( data$prob\_success , col=50 , ylab="probability of success" , xlab="each case of data (P.A.V)" , xlim=c(0,9) , ylim=c(0,1) , pch=16 , xaxt="n" )  
axis( 1 , at=1:8 , labels=c("L.A.L","L.A.S","L.I.L","L.I.S","S.A.L","S.A.S","S.I.L","S.I.S") )  
points( 1:8 , p.mean )  
for ( i in 1:8 ) lines( c(i, i), p.PI[,i] )



The predicted success count and the 89% interval for each row of data are as follows:

c <- as.data.frame(fit.a.stan)[,21:28]  
c.mean <- apply( c , 2 , mean )  
c.PI <- apply( c, 2, PI , prob=0.89 )  
plot( data$y , col=50 , ylab="count of success" , xlab="each case of data (P.A.V)" , xlim=c(0,9) , ylim=c(0,30) , pch=16 , xaxt="n" )  
axis( 1 , at=1:8 , labels=c("L.A.L","L.A.S","L.I.L","L.I.S","S.A.L","S.A.S","S.I.L","S.I.S") )  
points( 1:8 , c.mean )  
for ( i in 1:8 ) lines( c(i, i), c.PI[,i] )



## Conclusion

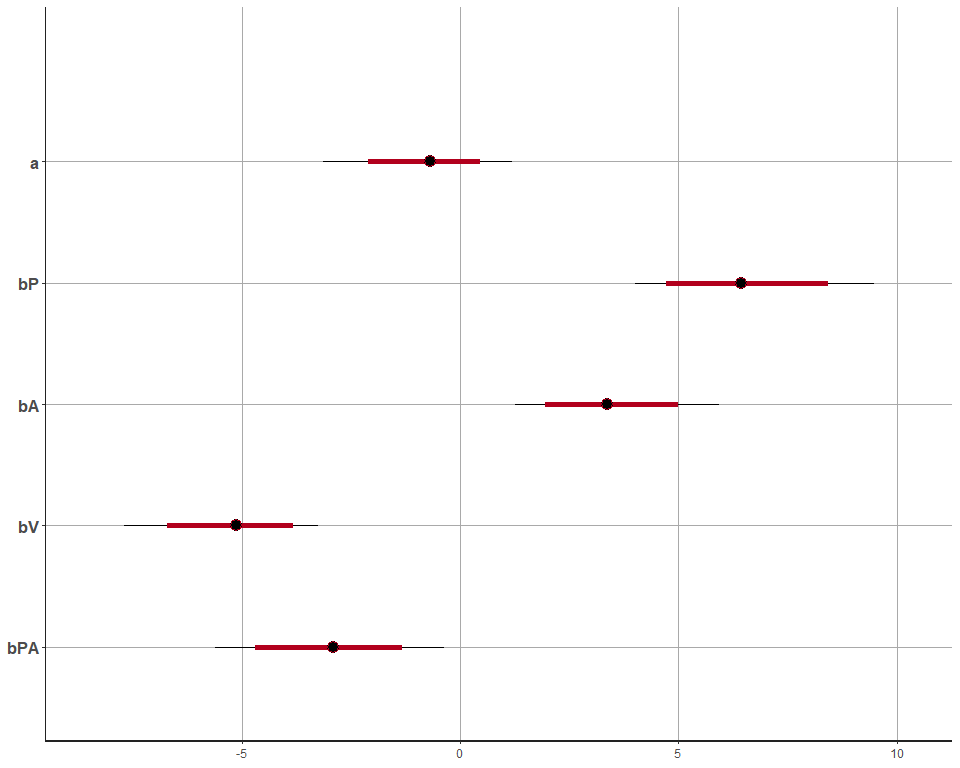
From the results, it shows that the probability of success could depend on pirate eagle size, victim eagle size and the age of pirate eagle. Besides, the success count is even more relative to the age of pirate eagle.

# Question 1 (c)

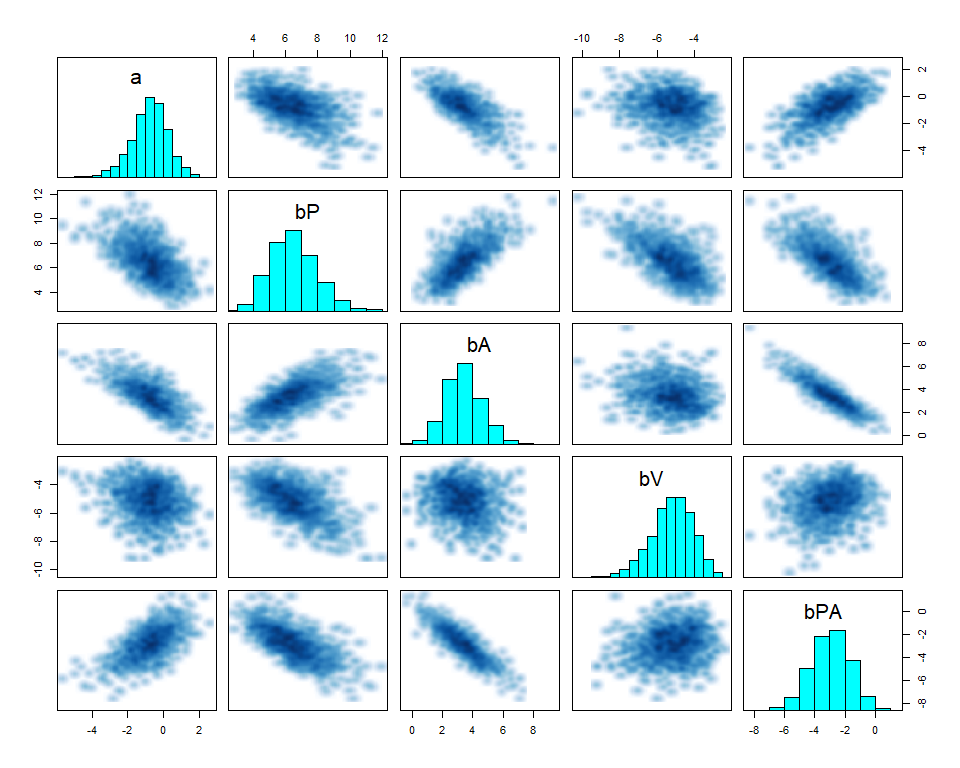
model.c.stan <- "  
data{  
 int<lower=1> N;  
 int y[N];  
 int n[N];  
 real P[N];  
 real A[N];  
 real V[N];  
}  
parameters{  
 real a;  
 real bP;  
 real bA;  
 real bV;  
 real bPA;  
}  
model{  
 vector[N] p;  
 bPA ~ normal( 0 , 5 );  
 bV ~ normal( 0 , 5 );  
 bA ~ normal( 0 , 5 );  
 bP ~ normal( 0 , 5 );  
 a ~ normal( 0 , 10 );  
 for ( i in 1:N ) {  
 p[i] = a + bP \* P[i] + bA \* A[i] + bV \* V[i] + bPA \* P[i] \* A[i];  
 p[i] = inv\_logit(p[i]);  
 }  
 y ~ binomial( n , p );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] p;  
 for ( i in 1:N ) {  
 p[i] = a + bP \* P[i] + bA \* A[i] + bV \* V[i] + bPA \* P[i] \* A[i];  
 p[i] = inv\_logit(p[i]);  
 log\_lik[i] = binomial\_lpmf( y[i] | n[i] , p[i] );  
 }  
}  
"  
  
fit.c.stan <- stan( model\_code=model.c.stan , data=data.model ,   
 chains=4 , cores=4 , iter=1000 )  
print( fit.c.stan , probs=c(0.055,0.945) , pars=c("a","bP","bA","bV","bPA") )

## Inference for Stan model: 025e21f51f81a03fd44433d4cd96f449.  
## 4 chains, each with iter=1000; warmup=500; thin=1;   
## post-warmup draws per chain=500, total post-warmup draws=2000.  
##   
## mean se\_mean sd 5.5% 94.5% n\_eff Rhat  
## a -0.77 0.04 1.06 -2.57 0.78 609 1  
## bP 6.50 0.06 1.42 4.39 8.90 550 1  
## bA 3.41 0.05 1.20 1.58 5.42 519 1  
## bV -5.21 0.04 1.14 -7.24 -3.56 810 1  
## bPA -2.95 0.06 1.33 -5.06 -0.94 537 1  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 16 11:29:58 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

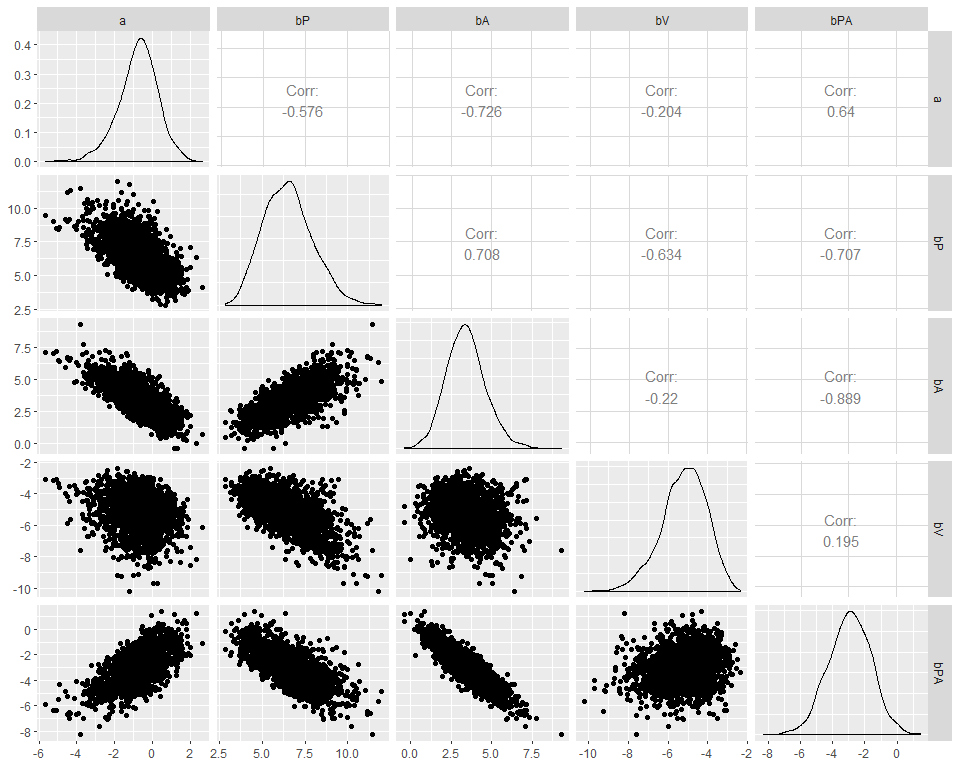
plot( fit.c.stan , pars=c("a","bP","bA","bV","bPA") )



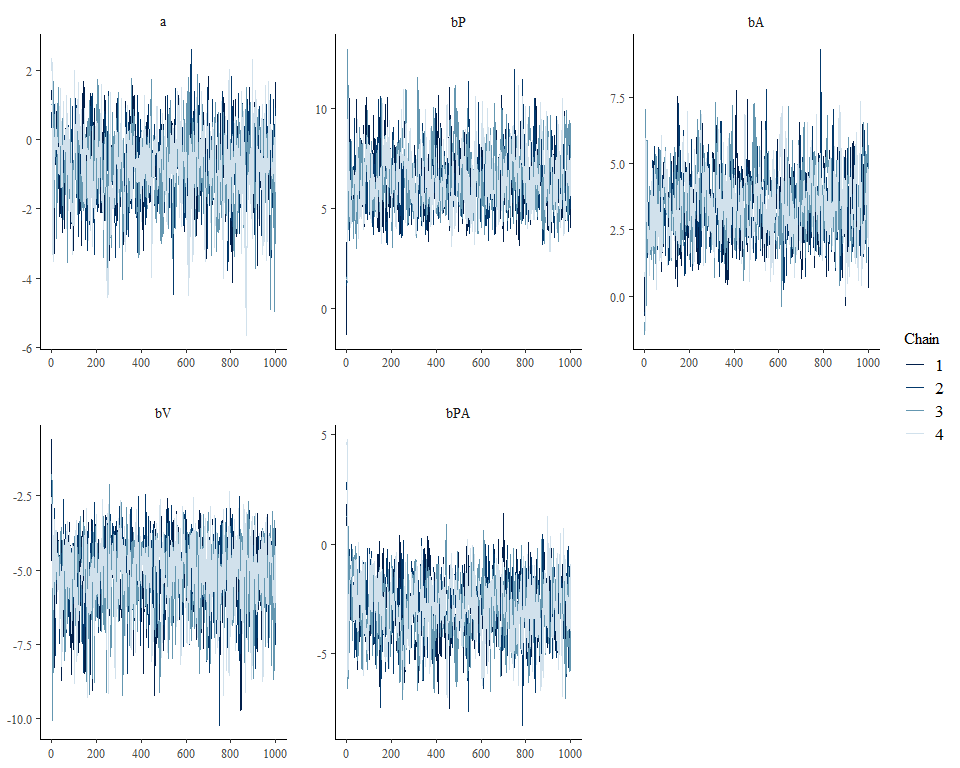
pairs( fit.c.stan , pars=c("a","bP","bA","bV","bPA") )



ggpairs( data=as.data.frame( fit.c.stan ), columns=c(1:5))



extract\_post <- rstan::extract( fit.c.stan , inc\_warmup=TRUE , permuted=FALSE)  
post\_traceplot <- mcmc\_trace( extract\_post , n\_warmup=0 , pars=c("a","bP","bA","bV","bPA"),  
 facet\_args=list( nrow=2 , labeller=label\_parsed ) )  
post\_traceplot



# Compute WAIC for both stan models  
ll.a.stan <- extract\_log\_lik(fit.a.stan, merge\_chains=FALSE)  
reff.a.stan <- relative\_eff(exp(ll.a.stan))  
waic.a.stan <- waic(ll.a.stan, r\_eff=reff.a.stan, cores=4)

## Warning: 4 (50.0%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

ll.c.stan <- extract\_log\_lik(fit.c.stan, merge\_chains=FALSE)  
reff.c.stan <- relative\_eff(exp(ll.c.stan))  
waic.c.stan <- waic(ll.c.stan, r\_eff=reff.c.stan, cores=4)

## Warning: 3 (37.5%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

# Compare WAICs  
loo::compare(waic.a.stan, waic.c.stan)

## elpd\_diff se   
## 4.7 1.6

## Conclusion

Since the elpd\_diff and se values are both positive, it implies that the latter model, which considers the interaction betwen variables P and A, predicts better than the other model.