Assignment6\_B04704016林家毅

# Question 1

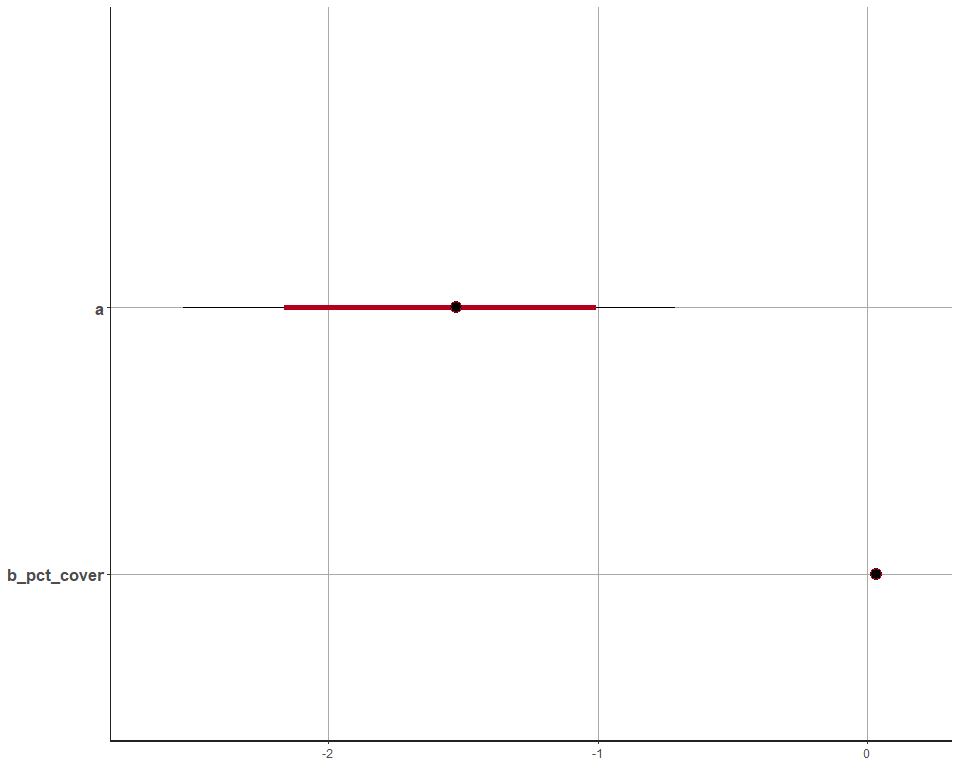
# Load data  
data( "salamanders" )  
data <- salamanders; rm( salamanders )  
data$FORESTAGE <- ( data$FORESTAGE - mean( data$FORESTAGE ))/sd( data$FORESTAGE )  
str( data )

## 'data.frame': 47 obs. of 4 variables:  
## $ SITE : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ SALAMAN : int 13 11 11 9 8 7 6 6 5 5 ...  
## $ PCTCOVER : int 85 86 90 88 89 83 83 91 88 90 ...  
## $ FORESTAGE: num 0.761 -0.418 1.96 -0.542 -0.65 ...

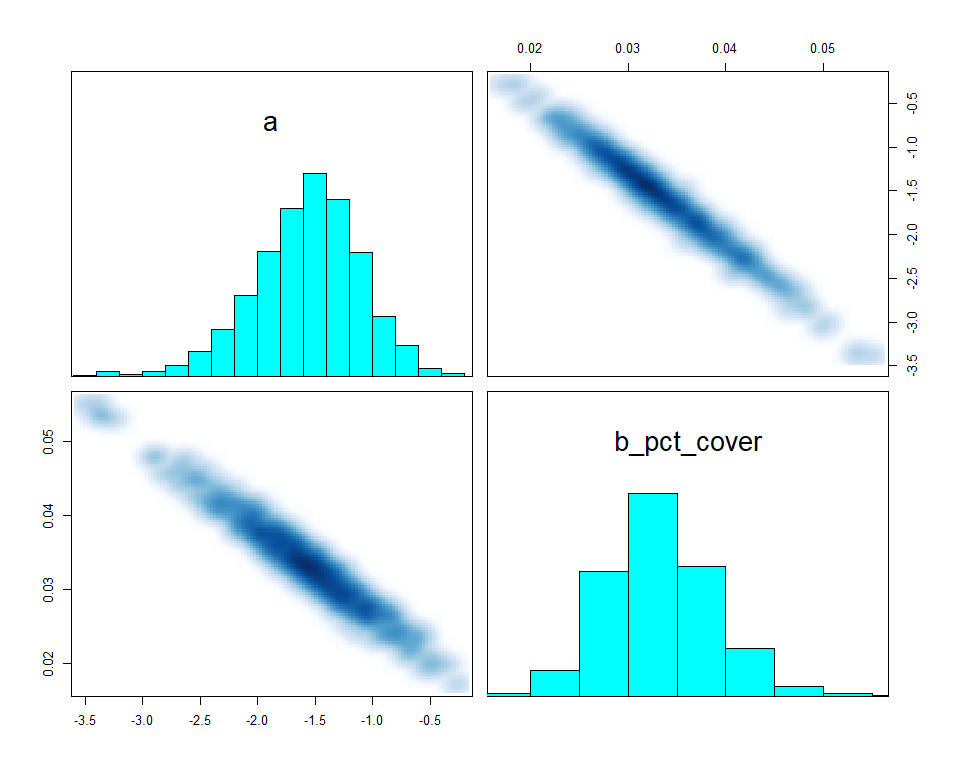
# Create data list for the stan model  
data.model <- list(  
 N = NROW( data ),  
 salaman = data$SALAMAN,  
 pct\_cover = data$PCTCOVER,  
 forest\_age = data$FORESTAGE  
)  
  
# Model code  
model.stan <- "  
data{  
 int<lower=1> N;  
 int salaman[N];  
 int pct\_cover[N];  
}  
parameters{  
 real a;  
 real b\_pct\_cover;  
}  
model{  
 vector[N] lamda;  
 b\_pct\_cover ~ uniform( -15 , 15 );  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_pct\_cover \* pct\_cover[i];  
 lamda[i] = exp( lamda[i] );  
 }  
 salaman ~ poisson( lamda );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_pct\_cover \* pct\_cover[i];  
 lamda[i] = exp( lamda[i] );  
 log\_lik[i] = poisson\_lpmf( salaman[i] | lamda[i] );  
 }  
}  
"  
  
# Fit stan model with model code and model data  
fit.stan.1 <- stan( model\_code=model.stan , data=data.model ,   
 chains=4 , cores=4 , iter=1000 )  
  
# Show results  
print( fit.stan.1 , probs=c(0.055,0.945) , pars=c("a","b\_pct\_cover") )

## Inference for Stan model: b60cb49efea2e41c4bee180b3f658845.  
## 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 -1.56 0.02 0.46 -2.33 -0.87 416 1.01  
## b\_pct\_cover 0.03 0.00 0.01 0.03 0.04 421 1.01  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 23 02:12:59 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).

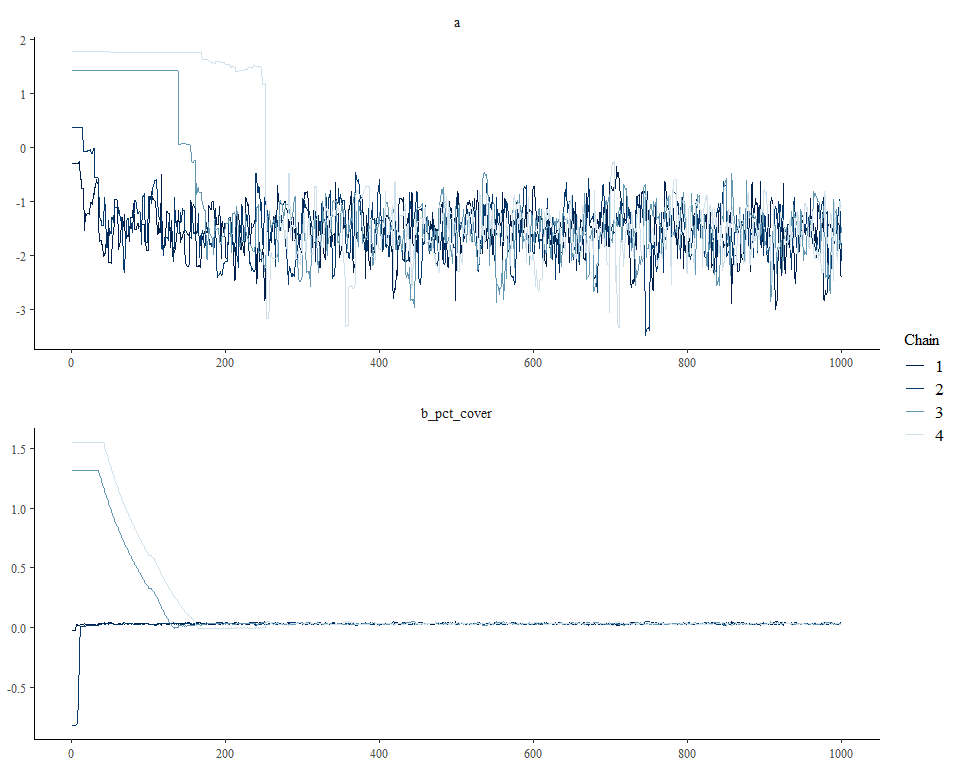
# Plot coeftab  
plot( fit.stan.1 , pars=c("a","b\_pct\_cover") )



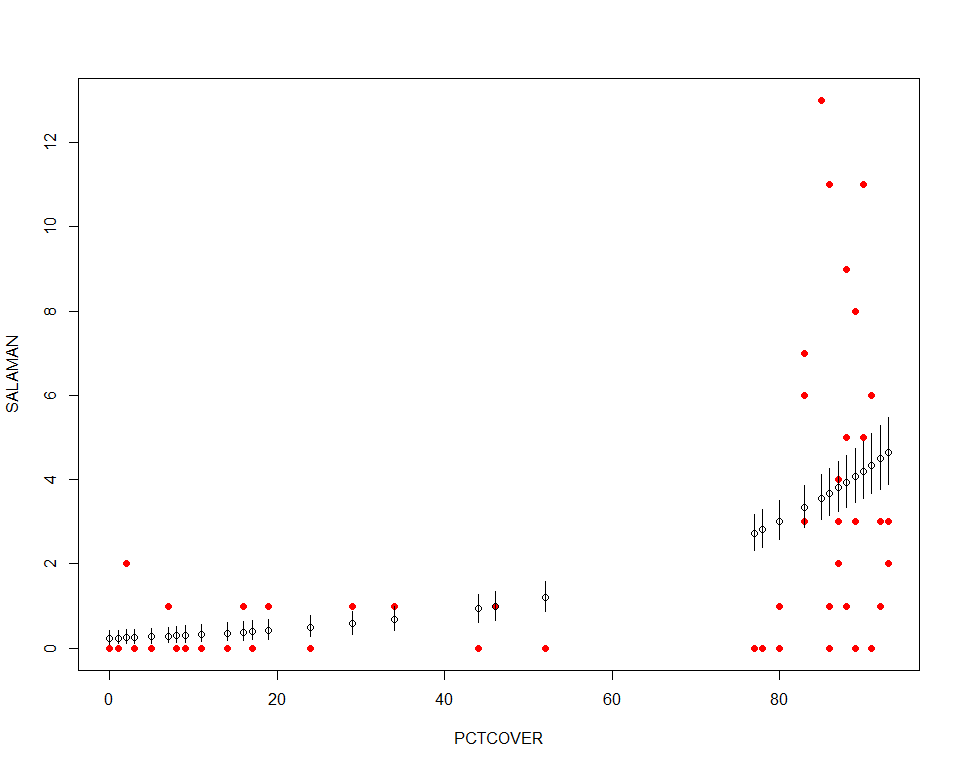
# Paits plot  
pairs( fit.stan.1 , pars=c("a","b\_pct\_cover") )



# Traceplot  
extract\_post <- rstan::extract( fit.stan.1 , inc\_warmup=TRUE , permuted=FALSE)  
post\_traceplot <- mcmc\_trace( extract\_post , n\_warmup=0 , pars=c("a","b\_pct\_cover") ,  
 facet\_args=list( nrow=2 , labeller=label\_parsed ) )  
post\_traceplot



# The predicted results of each row of data and the 89% interval are as follows:  
p <- as.data.frame(fit.stan.1)[,50:96]  
p.mean <- apply( p , 2 , mean )  
p.PI <- apply( p, 2, PI , prob=0.89 )  
plot(SALAMAN ~ PCTCOVER, data=data, col=50, pch=16)  
points(data$PCTCOVER , p.mean)  
for ( i in 1:47 ) lines( c(data$PCTCOVER[i], data$PCTCOVER[i]), p.PI[,i] )



## Conclusion

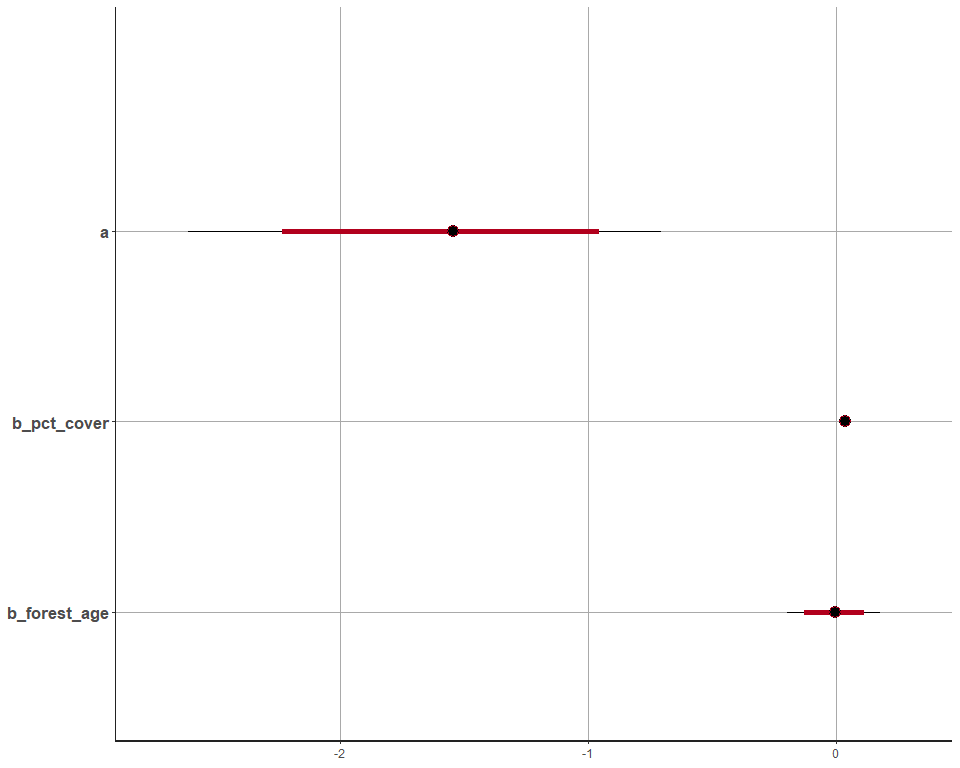
From all the plots and results above, we could see that the model did not predict so well, maybe some more variables should be added to the model.

# Question 2

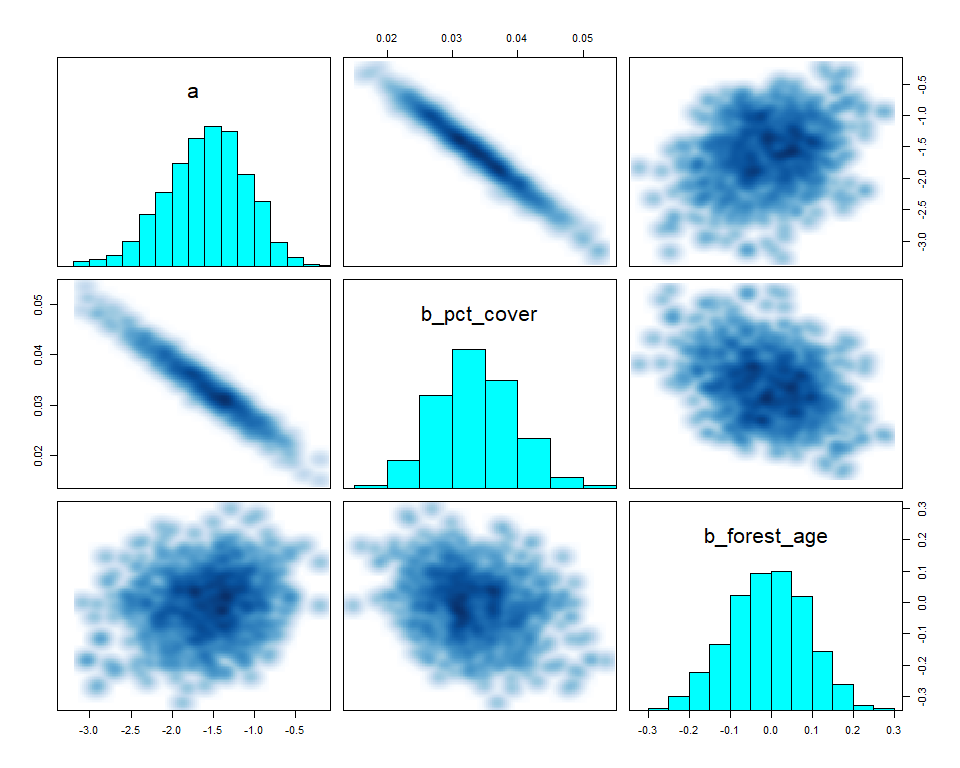
model.stan <- "  
data{  
 int<lower=1> N;  
 int salaman[N];  
 int pct\_cover[N];  
 real forest\_age[N];  
}  
parameters{  
 real a;  
 real b\_pct\_cover;  
 real b\_forest\_age;  
}  
model{  
 vector[N] lamda;  
 b\_forest\_age ~ uniform( -15 , 15 );  
 b\_pct\_cover ~ uniform( -15 , 15 );  
 a ~ uniform( -30, 30 );  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_pct\_cover \* pct\_cover[i] + b\_forest\_age \* forest\_age[i];  
 lamda[i] = exp( lamda[i] );  
 }  
 salaman ~ poisson( lamda );  
}  
generated quantities{  
 vector[N] log\_lik;  
 vector[N] lamda;  
 for ( i in 1:N ) {  
 lamda[i] = a + b\_pct\_cover \* pct\_cover[i] + b\_forest\_age \* forest\_age[i];  
 lamda[i] = exp( lamda[i] );  
 log\_lik[i] = poisson\_lpmf( salaman[i] | lamda[i] );  
 }  
}  
"  
  
# Fit stan model with model code and model data  
fit.stan.2 <- stan( model\_code=model.stan , data=data.model ,   
 chains=4 , cores=4 , iter=1000 )  
  
# Show results  
print( fit.stan.2 , probs=c(0.055,0.945) , pars=c("a","b\_pct\_cover","b\_forest\_age") )

## Inference for Stan model: 1a23cdb233bcb305b7be772ee9af1dd5.  
## 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 -1.58 0.02 0.49 -2.40 -0.85 535 1  
## b\_pct\_cover 0.03 0.00 0.01 0.02 0.04 537 1  
## b\_forest\_age -0.01 0.00 0.10 -0.17 0.14 684 1  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 23 02:13:51 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).

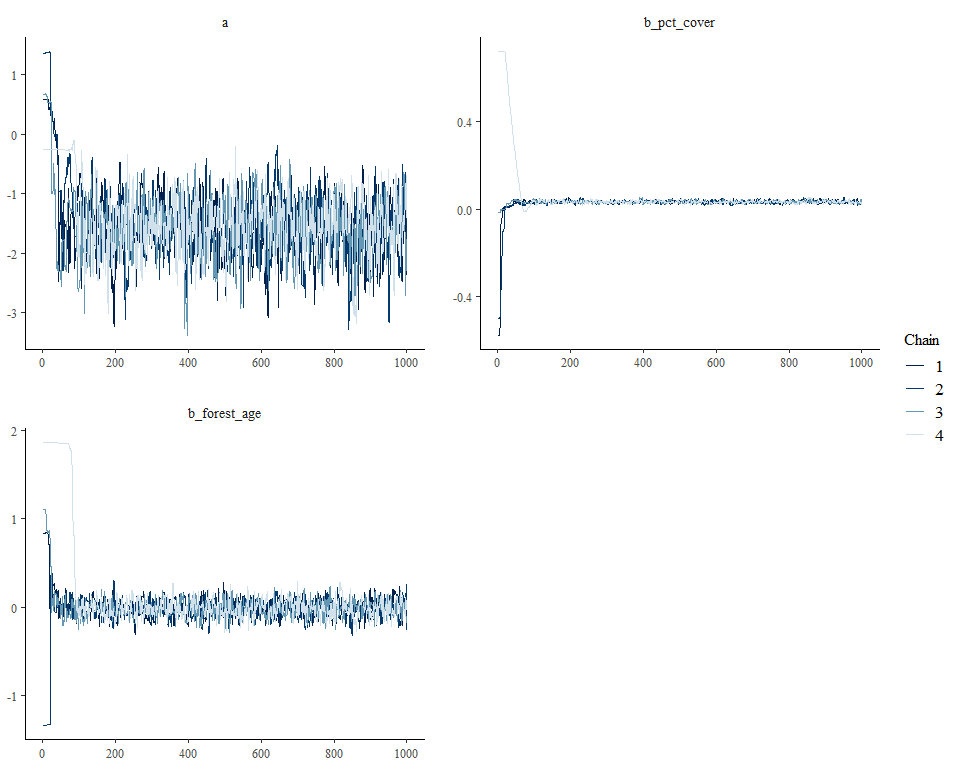
# Plot coeftab  
plot( fit.stan.2 , pars=c("a","b\_pct\_cover","b\_forest\_age") )



# Pairs plot  
pairs( fit.stan.2 , pars=c("a","b\_pct\_cover","b\_forest\_age") )



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



# Compute WAIC for both stan models  
ll.stan.1 <- extract\_log\_lik(fit.stan.1, merge\_chains=FALSE)  
reff.stan.1 <- relative\_eff(exp(ll.stan.1))  
waic.stan.1 <- waic(ll.stan.1, r\_eff=reff.a.stan, cores=4)

## Warning: 4 (8.5%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

ll.stan.2 <- extract\_log\_lik(fit.stan.2, merge\_chains=FALSE)  
reff.stan.2 <- relative\_eff(exp(ll.stan.2))  
waic.stan.2 <- waic(ll.stan.2, r\_eff=reff.c.stan, cores=4)

## Warning: 5 (10.6%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

# Compare WAICs  
loo::compare(waic.stan.1, waic.stan.2)

## elpd\_diff se   
## -2.4 0.7

## Conclusion

By comparing the second model to the first one, there isn’t much improvement since the elpd\_diff value is negative. The reason variable FORESTAGE did not help enhance prediction may be the fact that it is not relative to the outcome variable.