hw6

GISH

2019/5/21

## Q1

#center the pctcover  
d$PCTCOVER\_c <- d$PCTCOVER - mean(d$PCTCOVER)  
  
  
model <- map(   
 alist(  
 SALAMAN ~ dpois( lambda ),  
 log(lambda) <- a + bpc\*PCTCOVER\_c,  
 a ~ dnorm(0,30),  
 bpc ~ dnorm(0,1)  
 ),  
 data=d)  
  
precis(model)

## mean sd 5.5% 94.5%  
## a 0.42972260 0.158293961 0.17673828 0.68270693  
## bpc 0.03240015 0.005389335 0.02378696 0.04101335

model.stan <- map2stan(model,chains=2)

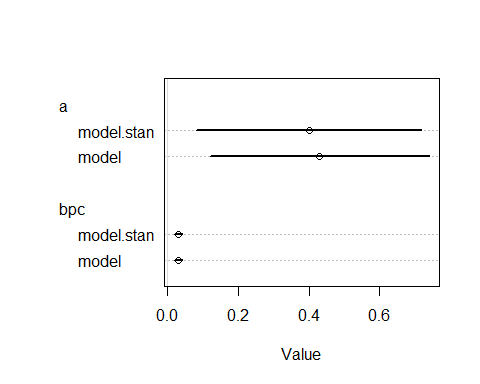
##   
## SAMPLING FOR MODEL 'fa314c2f34fe25c3627325799c4e5c84' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 1: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 1: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 1: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 1: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 1: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 1: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 1: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 1: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 1: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 1: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 0.269 seconds (Warm-up)  
## Chain 1: 0.082 seconds (Sampling)  
## Chain 1: 0.351 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'fa314c2f34fe25c3627325799c4e5c84' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 0 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 0.256 seconds (Warm-up)  
## Chain 2: 0.172 seconds (Sampling)  
## Chain 2: 0.428 seconds (Total)  
## Chain 2:

## Computing WAIC

coeftab( model, model.stan )

## model model.stan  
## a 0.43 0.40   
## bpc 0.03 0.03   
## nobs 47 47

plot( coeftab( model, model.stan))

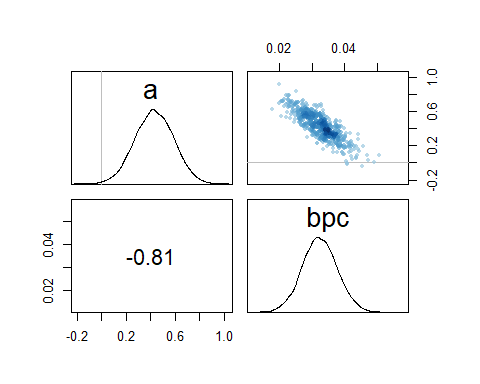


compare(model,model.stan)

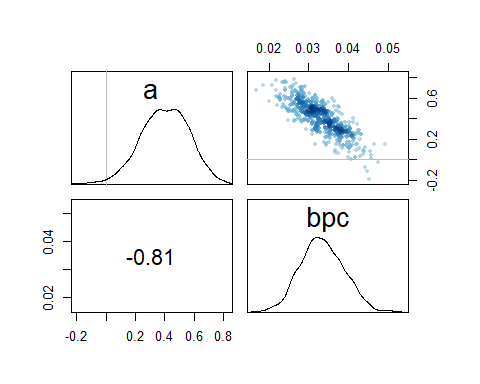
## Warning in compare(model, model.stan): Not all model fits of same class.  
## This is usually a bad idea, because it implies they were fit by different algorithms.  
## Check yourself, before you wreck yourself.

## WAIC pWAIC dWAIC weight SE dSE  
## model 213.1485 4.626559 0.0000000 0.538777 26.04645 NA  
## model.stan 213.4593 4.815233 0.3108401 0.461223 26.42267 0.6290469

pairs(model)



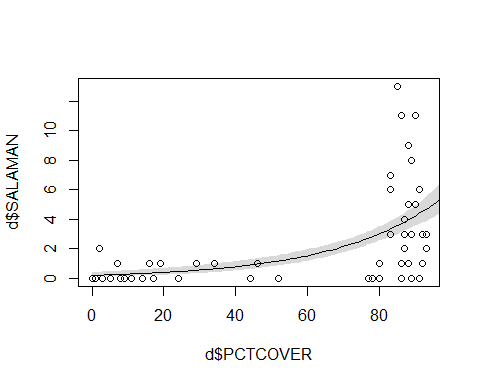
pairs(model.stan)



#probably stan is better  
  
  
# sequence of percent ground cover  
cover.seq <- seq(from=0, to=100, length.out=100)  
  
d.pred <- data.frame(  
 PCTCOVER = cover.seq,  
 PCTCOVER\_c = cover.seq - mean(d$PCTCOVER)  
)  
  
lambda.pred <- link(model.stan, data=d.pred)

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

lambda.med <- apply(lambda.pred, 2, median)  
lambda.PI <- apply(lambda.pred, 2, PI, prob=0.89)  
  
  
# plot data and plot prediction with lines  
plot( d$PCTCOVER, d$SALAMAN)  
  
lines( cover.seq, lambda.med)  
shade( lambda.PI, cover.seq)

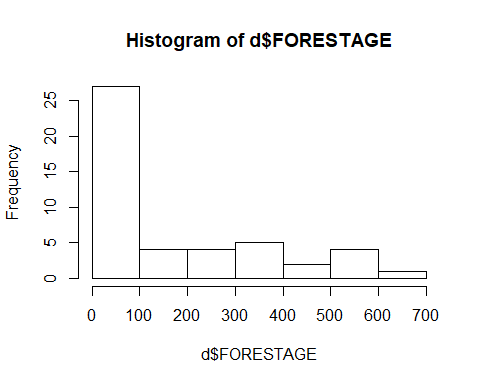


當percentage of cover低時，可以發現幾乎沒什麼salamanders，代表percentage of cover高，是salamanders生存的很大關鍵之一

在percentage of cover 高的時候 ，可以看到salamanders的數量變動很大 =>可能有其他重要的factor在影響

## Q2

#觀察forestage，發現多數不到100  
hist(d$FORESTAGE)



d$logFORESTAGE <- log(d$FORESTAGE + 1)  
  
#center forestage  
d$logFORESTAGE\_c <- d$logFORESTAGE - mean(d$logFORESTAGE)  
  
modelB <- map2stan(   
 alist(  
 SALAMAN ~ dpois( lambda ),  
 log(lambda) <- a + bpc\*PCTCOVER\_c + bfa\*logFORESTAGE\_c,  
 a ~ dnorm(0,30),  
 bpc ~ dnorm(0,1),  
 bfa ~ dnorm(0,1)  
 ),  
 data=d, chains = 2)

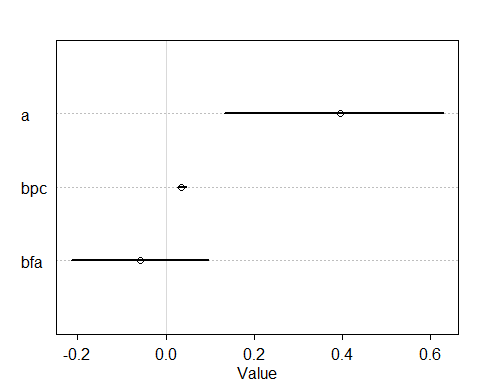
##   
## SAMPLING FOR MODEL 'cf74428672fd73a2c9524749ffda5270' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 1: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 1: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 1: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 1: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 1: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 1: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 1: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 1: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 1: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 1: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 0.785 seconds (Warm-up)  
## Chain 1: 0.303 seconds (Sampling)  
## Chain 1: 1.088 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'cf74428672fd73a2c9524749ffda5270' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 0 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)  
## Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)  
## Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)  
## Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)  
## Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)  
## Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)  
## Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)  
## Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)  
## Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 0.697 seconds (Warm-up)  
## Chain 2: 0.347 seconds (Sampling)  
## Chain 2: 1.044 seconds (Total)  
## Chain 2:

## Computing WAIC

precis(modelB)

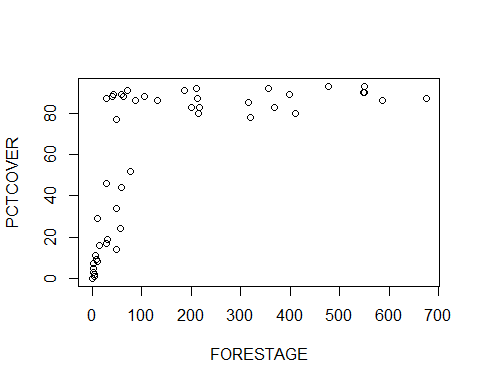
## mean sd 5.5% 94.5% n\_eff Rhat  
## a 0.39586614 0.155959589 0.13302923 0.63112625 886.1467 1.0001274  
## bpc 0.03553034 0.006508016 0.02529885 0.04582462 919.9872 0.9995350  
## bfa -0.05785092 0.096250144 -0.21443842 0.09662304 931.0668 0.9999668

plot(precis(modelB))



bfa標準差很大，區間-0.21~0.11 包含了0 看起來應該是對模型解釋沒有什麼幫助

plot(PCTCOVER ~ FORESTAGE, data=d)



#可以看到其實forestage的高低就蠻影響percentage of cover,與log forestage有高度正相關  
  
compare(model.stan,modelB)

## WAIC pWAIC dWAIC weight SE dSE  
## model.stan 213.4593 4.815233 0.000000 0.8456257 26.42267 NA  
## modelB 216.8607 7.295790 3.401393 0.1543743 27.64691 2.408648

compare(model.stan,modelB,func=LOO)

## Warning: Relative effective sample sizes ('r\_eff' argument) not specified.  
## For models fit with MCMC, the reported PSIS effective sample sizes and   
## MCSE estimates will be over-optimistic.  
  
## Warning: Relative effective sample sizes ('r\_eff' argument) not specified.  
## For models fit with MCMC, the reported PSIS effective sample sizes and   
## MCSE estimates will be over-optimistic.

## Warning: Some Pareto k diagnostic values are slightly high. See help('pareto-k-diagnostic') for details.

## LOO pLOO dLOO weight SE dSE  
## model.stan 213.5959 4.917499 0.000000 0.8114572 1.167017 NA  
## modelB 216.5149 7.061374 2.919013 0.1885428 1.647664 2.216986

#直接使用percentage of cover模型即可)

可以看到其實forestage的高低就蠻影響percentage of cover,與log forestage有高度正相關

直接使用percentage of cover模型即可)