

```
#####
#####
##### Horseshoe Crabs #####
#####

#### Clear environment and load libraries
rm(list = ls())
library(coda)
library(rsaq)
library(mvtnorm)

#### Data
data("hcrabs")
dim(hcrabs)
head(hcrabs)
#note that
#color: 1 = light; 2 = medium light; 3 = medium; 4 = medium dark; 5 = dark (not all of these colors appear)
#spine condition: 1 = both good, 2 = one worn or broken, 3 = both worn or broken
#carapace width: cm
#weight: kg

#### Now to the sampler
#Data summaries
Y <- hcrabs$num.satellites
X <- model.matrix(~color+spine+width+weight,data=hcrabs)
n <- nrow(X)
p <- ncol(X)

#Hyperparameters for the prior
beta_0 <- matrix(0,nrow=p)
Sigma_0 <- diag(1,p)

#Set parameters for proposal density
c <- 0.5
delta <- 0.1 #use it to tune acceptance ratio
var_prop <- delta*var(log(Y+c))*solve(t(X)%*%X)

#Initial values for sampler
beta <- beta_0

#First set number of iterations and burn-in, then set seed
n_iter <- 10000
burn_in <- 0.3*n_iter
thin <- 1
set.seed(1234)

#Set counter for acceptances
accept_counter <- 0

#Set null matrices to save samples
BETA <- matrix(0,nrow=n_iter,ncol=p)

#Now, to the Gibbs sampler
for(s in 1:(n_iter+burn_in)){

  #generate proposal
  beta_star <- t(rmvnorm(1,beta,var_prop))

  #compute acceptance ratio/probability
  #do so on log scale because r can be numerically unstable
  log_r <- sum(dpois(Y,exp(X%*%beta_star),log=T)) + dmvnorm(c(beta_star,beta_0,Sigma_0,log=T) -
    sum(dpois(Y,exp(X%*%beta),log=T)) - dmvnorm(c(beta),beta_0,Sigma_0,log=T)

  if(log(runif(1)) < log_r){
    accept_counter <- accept_counter + 1
    beta <- beta_star
  }

  if(s > burn_in){
    BETA[s-burn_in,] <- beta
  }
}

#Check acceptance rate
accept_counter/(n_iter+burn_in)

#thinning
sample_thin <- seq(1,n_iter,by=thin)
BETA_thinned <- BETA[sample_thin,]
colnames(BETA_thinned) <- colnames(X)

plot(mcmc(BETA_thinned))
#trace plots look fine
autocorr.plot(mcmc(BETA_thinned))
#unlike Gibbs sampling, we do have some stickiness when doing Metropolis/W-H
#you also saw this in the lab
apply(BETA_thinned,2,effectiveSize)
#we can still:
#1. increase burn-in
#2. increase the number of samples
#3. thin some more
#I leave that as an exercise for you!!

#quick convergence diagnostics.
geweke.diag(mcmc(BETA_thinned))
#these are z-scores for the hypothesis that both parts of the chain come from the same distribution.
#so, Geweke's test is reassuring.

#posterior summaries
round(apply(BETA_thinned,2,mean),2)
round(apply(BETA_thinned,2,function(x) quantile(x,probs=c(0.025,0.975))),2)

#Just for reference, take a look at frequentist model
freq_model <- glm(num.satellites~color+spine+width+weight,family=poisson,data=hcrabs)
summary(freq_model)
#quite similar...

###We can actually do a better job with model selection but I leave that to you!!!
```