Effect of bottom coverage to larval presence

Week6-ex1, problem statement

In this exercise, we continue the analysis of the white fish larval areas (week 2, exercise 3 and week4, exercise 2). This time we extend the model to include regression along two continues covariates in addition to the vegetation cover status. The additional covariates that we are interested in are the distance to sandy shore and the length of ice cover during winter. Many fishermen have observed that white fish are caught more easily from sandy shores than elsewhere during their spawning season. Moreover, white fish spawn their eggs during fall but the larvae hatch only in the spring. Hence, it has been suggested that longer ice cover period works as a shelter for the eggs. Hence, let’s take a look whether there is statistical signal to these covariates.

Let’s load the data and construct covariate matrix (a matrix where the ’th row contains the covariates for the ’th sampling site), vector of area indexes (areas in the code) and vector of white fish presence-absence observations .

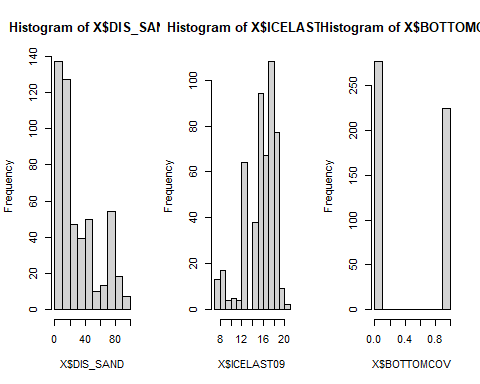
# Read the data  
data = read.csv("white\_fishes\_data.csv")  
  
# Let's then take the covariates to matrix X and standardize them  
X = data[,c("DIS\_SAND","ICELAST09","BOTTOMCOV")]  
  
# And for last let's take the presence-absence observations of white fish larvae into Y  
y = data$WHIBIN

Unlike in our previous analyses of this data we treat each sampling site as one observation and consider the triplets ( is the ’th row of ) exchangeable.

We will first build the following model to analyze the data. Hence, we assume that the prior expectation of the probability to observe white fish larvae () follows logit linear model where is the intercept and is a vector of (fixed) effects of covariates. Note that the matrix notation is the same as writing

Note also that the DIS\_SAND and ICELAST09 are continuous covariates whereas BOTTOMCOV is a categorical covariate getting value 1 if the bottom is covered by vegetation and 0 if the bottom is not covered by vegetation.

par(mfrow=c(1,3))  
hist(X$DIS\_SAND)  
hist(X$ICELAST09)  
hist(X$BOTTOMCOV)

 Hence, the parameter corresponds to the effect of vegetation to the observation probability of white fish larvae.

Before starting the analysis we standardize the continues covariates but not the categorical BOTTOMCOV covariate. If we standardized the categorical variable the interpretation of parameter would change.

mx = colMeans(X[,1:2])  
stdx = apply(X[,1:2],2,sd)  
X[,1:2] = (X[,1:2]-t(replicate(dim(X)[1],mx)))/t(replicate(dim(X)[1],stdx))

**Task 1:**

Implement the model in Stan and sample from the posterior for the parameters and . Check for convergence of the MCMC chain and examine the autocorrelation of the samples. Visualize the posterior for and and discuss the results.

First we define the model:

logit\_fish\_model = "  
data{  
 int<lower = 0> n;  
 real DIS\_SAND[n];  
 real ICELAST09[n];  
 int BOTTOMCOV[n];  
 int<lower=0,upper=1> y[n];  
}  
  
parameters{  
 real alpha;  
 real beta\_1;  
 real beta\_2;  
 real beta\_3;  
}  
  
model{  
 //priors:  
 alpha ~ normal(0, sqrt(10));  
 beta\_1 ~ normal(0, sqrt(10));  
 beta\_2 ~ normal(0, sqrt(10));  
 beta\_3 ~ normal(0, sqrt(10));  
 for (i in 1:n){  
 y[i] ~ bernoulli\_logit(alpha + beta\_1\*DIS\_SAND[i] + beta\_2\*ICELAST09[i] + beta\_3\*BOTTOMCOV[i]);  
 }  
}  
  
"

Then prepare the data:

data <- list (n=length(y), DIS\_SAND=X$DIS\_SAND, ICELAST09=X$ICELAST09, BOTTOMCOV=X$BOTTOMCOV, y=y)

Then running the model:

##   
## rstan version 2.26.13 (Stan version 2.26.1)

## For execution on a local, multicore CPU with excess RAM we recommend calling  
## options(mc.cores = parallel::detectCores()).  
## To avoid recompilation of unchanged Stan programs, we recommend calling  
## rstan\_options(auto\_write = TRUE)  
## For within-chain threading using `reduce\_sum()` or `map\_rect()` Stan functions,  
## change `threads\_per\_chain` option:  
## rstan\_options(threads\_per\_chain = 1)

## Do not specify '-march=native' in 'LOCAL\_CPPFLAGS' or a Makevars file

## Warning: package 'coda' was built under R version 4.2.2

##   
## Attaching package: 'coda'

## The following object is masked from 'package:rstan':  
##   
## traceplot

post=stan(model\_code=logit\_fish\_model,data=data,warmup=500,iter=2000,chains=4,thin=1)

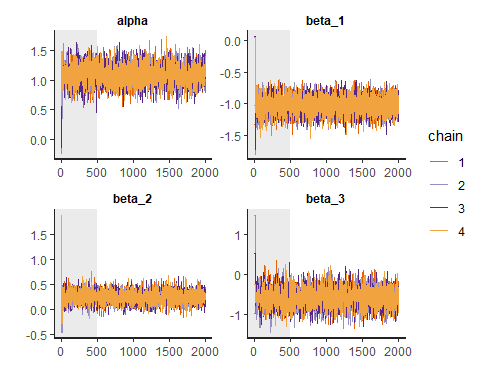
##   
## SAMPLING FOR MODEL 'anon\_model' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 0.000719 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 7.19 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 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: 501 / 2000 [ 25%] (Sampling)  
## Chain 1: Iteration: 700 / 2000 [ 35%] (Sampling)  
## Chain 1: Iteration: 900 / 2000 [ 45%] (Sampling)  
## Chain 1: Iteration: 1100 / 2000 [ 55%] (Sampling)  
## Chain 1: Iteration: 1300 / 2000 [ 65%] (Sampling)  
## Chain 1: Iteration: 1500 / 2000 [ 75%] (Sampling)  
## Chain 1: Iteration: 1700 / 2000 [ 85%] (Sampling)  
## Chain 1: Iteration: 1900 / 2000 [ 95%] (Sampling)  
## Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 1.536 seconds (Warm-up)  
## Chain 1: 5.666 seconds (Sampling)  
## Chain 1: 7.202 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'anon\_model' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 0.000514 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 5.14 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: 501 / 2000 [ 25%] (Sampling)  
## Chain 2: Iteration: 700 / 2000 [ 35%] (Sampling)  
## Chain 2: Iteration: 900 / 2000 [ 45%] (Sampling)  
## Chain 2: Iteration: 1100 / 2000 [ 55%] (Sampling)  
## Chain 2: Iteration: 1300 / 2000 [ 65%] (Sampling)  
## Chain 2: Iteration: 1500 / 2000 [ 75%] (Sampling)  
## Chain 2: Iteration: 1700 / 2000 [ 85%] (Sampling)  
## Chain 2: Iteration: 1900 / 2000 [ 95%] (Sampling)  
## Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 1.466 seconds (Warm-up)  
## Chain 2: 3.85 seconds (Sampling)  
## Chain 2: 5.316 seconds (Total)  
## Chain 2:   
##   
## SAMPLING FOR MODEL 'anon\_model' NOW (CHAIN 3).  
## Chain 3:   
## Chain 3: Gradient evaluation took 0.000536 seconds  
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 5.36 seconds.  
## Chain 3: Adjust your expectations accordingly!  
## Chain 3:   
## Chain 3:   
## Chain 3: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 3: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 3: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 3: Iteration: 501 / 2000 [ 25%] (Sampling)  
## Chain 3: Iteration: 700 / 2000 [ 35%] (Sampling)  
## Chain 3: Iteration: 900 / 2000 [ 45%] (Sampling)  
## Chain 3: Iteration: 1100 / 2000 [ 55%] (Sampling)  
## Chain 3: Iteration: 1300 / 2000 [ 65%] (Sampling)  
## Chain 3: Iteration: 1500 / 2000 [ 75%] (Sampling)  
## Chain 3: Iteration: 1700 / 2000 [ 85%] (Sampling)  
## Chain 3: Iteration: 1900 / 2000 [ 95%] (Sampling)  
## Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 3:   
## Chain 3: Elapsed Time: 1.765 seconds (Warm-up)  
## Chain 3: 3.935 seconds (Sampling)  
## Chain 3: 5.7 seconds (Total)  
## Chain 3:   
##   
## SAMPLING FOR MODEL 'anon\_model' NOW (CHAIN 4).  
## Chain 4:   
## Chain 4: Gradient evaluation took 0.000484 seconds  
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 4.84 seconds.  
## Chain 4: Adjust your expectations accordingly!  
## Chain 4:   
## Chain 4:   
## Chain 4: Iteration: 1 / 2000 [ 0%] (Warmup)  
## Chain 4: Iteration: 200 / 2000 [ 10%] (Warmup)  
## Chain 4: Iteration: 400 / 2000 [ 20%] (Warmup)  
## Chain 4: Iteration: 501 / 2000 [ 25%] (Sampling)  
## Chain 4: Iteration: 700 / 2000 [ 35%] (Sampling)  
## Chain 4: Iteration: 900 / 2000 [ 45%] (Sampling)  
## Chain 4: Iteration: 1100 / 2000 [ 55%] (Sampling)  
## Chain 4: Iteration: 1300 / 2000 [ 65%] (Sampling)  
## Chain 4: Iteration: 1500 / 2000 [ 75%] (Sampling)  
## Chain 4: Iteration: 1700 / 2000 [ 85%] (Sampling)  
## Chain 4: Iteration: 1900 / 2000 [ 95%] (Sampling)  
## Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)  
## Chain 4:   
## Chain 4: Elapsed Time: 1.439 seconds (Warm-up)  
## Chain 4: 3.746 seconds (Sampling)  
## Chain 4: 5.185 seconds (Total)  
## Chain 4:

Convergence etc.:

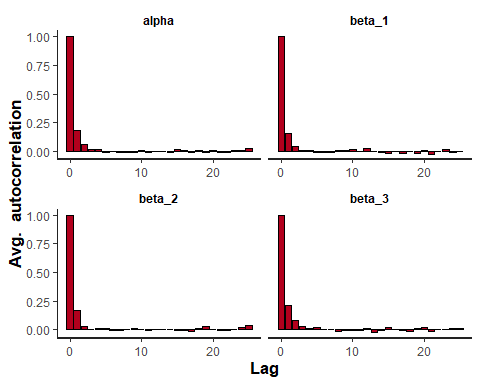
print(post)

## Inference for Stan model: anon\_model.  
## 4 chains, each with iter=2000; warmup=500; thin=1;   
## post-warmup draws per chain=1500, total post-warmup draws=6000.  
##   
## mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  
## alpha 1.09 0.00 0.16 0.79 0.98 1.09 1.20 1.42 3909 1  
## beta\_1 -1.02 0.00 0.13 -1.29 -1.11 -1.01 -0.93 -0.77 4283 1  
## beta\_2 0.24 0.00 0.12 0.02 0.16 0.24 0.32 0.46 4168 1  
## beta\_3 -0.59 0.00 0.23 -1.05 -0.74 -0.59 -0.43 -0.15 3477 1  
## lp\_\_ -250.69 0.03 1.44 -254.33 -251.37 -250.37 -249.63 -248.92 2830 1  
##   
## Samples were drawn using NUTS(diag\_e) at Mon Dec 12 11:40:38 2022.  
## 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).

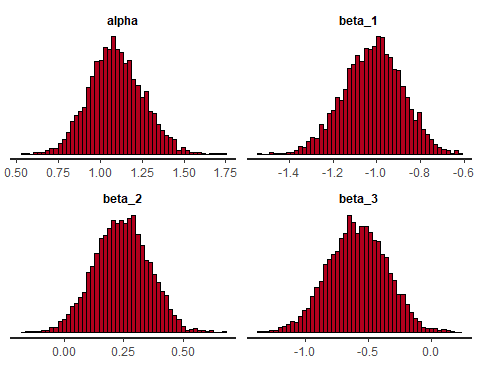
plot(post,plotfun= "trace", inc\_warmup = TRUE)



stan\_ac(post,inc\_warmup = FALSE, lags = 25)

 As we can conclude from above, the convergence and autocorrelation of the chains were in order. Rhats at 1, visually we can see that the chains have converged, and the autocorrelation is around zero at all higher lags.

Nsamp=as.matrix(post)  
plot(post, plotfun = "hist", pars = list('alpha', 'beta\_1', 'beta\_2', 'beta\_3') ,bins=50)

 As we can see from the histograms above, alpha, i.e., the intercept is around 1.2, while distance to sandy shores have a negative impact on the prevalence of white fish larvae, a longer icecover has a positive effect on white fish larvae, and (as we have seen in earler exercises) bottom cover vegetation has a negative effect of white fish larvae. The distribution of the parameters has retained a gaussian spread.

**Task 2.**

Calculate the posterior correlation between and . How does this differ from the prior correlation and why?

corr <- cor(Nsamp[,'alpha'], Nsamp[,'beta\_3'])  
corr

## [1] -0.7100711

The prior did not still include any information on the target values y, and that is why the only thing that affected the values were the prior, uninformed and identical distribution. After training the model on the target data, the distribution of and have changed to adapt to their role in the linearly logistic model. The prior correlation, as both and was drawn from a N(0,10) distribution, should have been close to 0. In the posterior, the correlation is -0.716, which shows, that with a higher intercept or , needs to be lower, in order to reach the same result for .

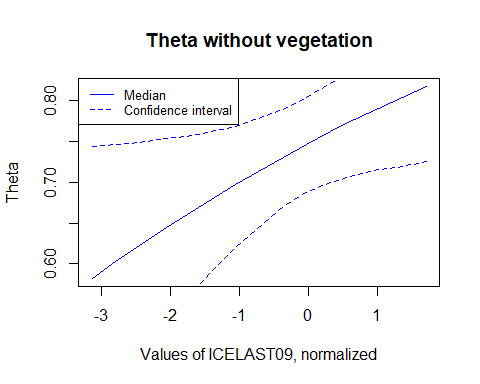
\*\*Task 3.\* Visualize the posterior of as a function of ICELAST09 when DISSAND is set to its mean value and in both cases when BOTTOMCOV=0 and BOTTOMCOV=1. That is, draw the median and 95% credible interval of the prediction function within the range from minimum to maximum value of ICELAST09 in the data.

mean\_sand = mean(X$DIS\_SAND)  
sigmoid = function(x) {  
 1 / (1 + exp(-x))  
}

#beta\_2s = Nsamp[, 'beta\_2']\*X$ICELAST09#calculate this individually for all values of icelast and plot median and confidence interval of result.  
thetas\_cov0 = array(dim=c(length(X$ICELAST09),3))  
icelast <- X$ICELAST09  
icelast <- sort(icelast)

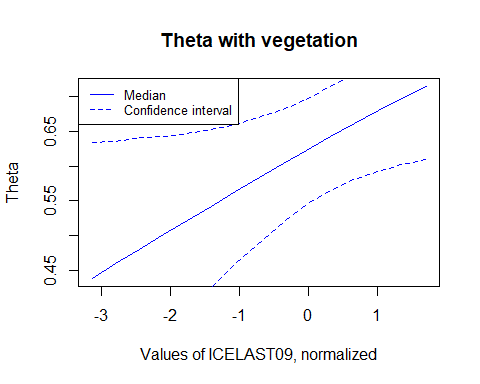
for (i in 1:502){  
 bx <- Nsamp[, 'alpha'] + Nsamp[,'beta\_1']\*mean\_sand +Nsamp[,'beta\_2']\*icelast[i] + Nsamp[,'beta\_3'] \* 0;  
 theta <- sigmoid(bx)  
 thetas\_cov0[i,1] <- median(theta)  
 thetas\_cov0[i,2] <- quantile(theta,probs=c(0.025))  
 thetas\_cov0[i,3] <- quantile(theta,probs=c(0.975))  
}

plot(icelast,thetas\_cov0[,1], col="blue", main="Theta without vegetation", type="l", xlab="Values of ICELAST09, normalized", ylab="Theta")  
lines(icelast,thetas\_cov0[,2], col="blue", lty=2, type='l')  
lines(icelast,thetas\_cov0[,3], col="blue", lty=2, type="l")  
legend(x="topleft", legend=c("Median", "Confidence interval"),  
 col=c("blue", "blue"), lty=1:2, cex=0.8)



thetas\_cov1 = array(dim=c(length(X$ICELAST09),3))  
for (i in 1:502){  
 bx2 <- Nsamp[, 'alpha'] + Nsamp[,'beta\_1']\*mean\_sand +Nsamp[,'beta\_2']\*icelast[i] + Nsamp[,'beta\_3'] \* 1;  
 #theta2 <- sigmoid(bx2)  
 thetas\_cov1[i,1] <- sigmoid(median(bx2))  
 thetas\_cov1[i,2] <- sigmoid(quantile(bx2,probs=c(0.025)))  
 thetas\_cov1[i,3] <- sigmoid(quantile(bx2,probs=c(0.975)))  
}

plot(icelast,thetas\_cov1[,1], col="blue", main="Theta with vegetation", type="l", xlab="Values of ICELAST09, normalized", ylab="Theta")  
lines(icelast,thetas\_cov1[,2], col="blue", lty=2, type='l')  
lines(icelast,thetas\_cov1[,3], col="blue", lty=2, type="l")  
legend(x="topleft", legend=c("Median", "Confidence interval"),  
 col=c("blue", "blue"), lty=1:2, cex=0.8)

 **Task 4.**

Visualize the posterior distribution of at location where DIS\_SAND is 60 and ICELAST is 18 for both vegetated and non-vegetated bottom types as well as their difference.

First normalizing the values given:

sand <- (60-mx['DIS\_SAND'])/stdx['DIS\_SAND']  
ice <- (18-mx['ICELAST09'])/stdx['ICELAST09']  
print(sand)

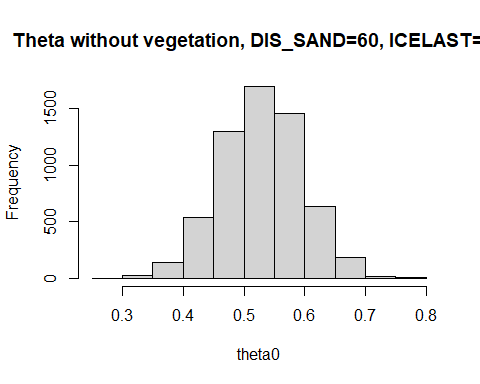
## DIS\_SAND   
## 1.11729

print(ice)

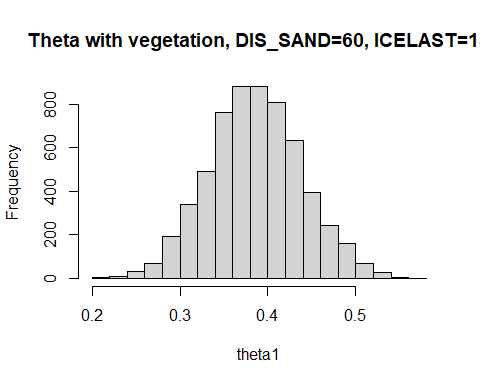
## ICELAST09   
## 0.6707792

Then calculate thetas:

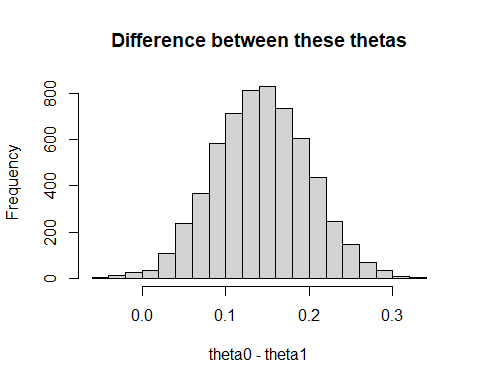
bx0 <- Nsamp[, 'alpha'] + Nsamp[,'beta\_1']\*sand +Nsamp[,'beta\_2']\*ice + Nsamp[,'beta\_3'] \* 0;  
theta0 <- sigmoid(bx0)  
  
bx1 <- Nsamp[, 'alpha'] + Nsamp[,'beta\_1']\*sand +Nsamp[,'beta\_2']\*ice + Nsamp[,'beta\_3'] \* 1;  
theta1 <- sigmoid(bx1)  
  
hist(theta0, main="Theta without vegetation, DIS\_SAND=60, ICELAST=18")



hist(theta1, main="Theta with vegetation, DIS\_SAND=60, ICELAST=18")



hist(theta0-theta1, main="Difference between these thetas")



**Task 5.**

How does the difference in for vegetated and non-vegetated bottom differ from in exercise 3 of week 2 and in exercise 2 of week 4? Would you say that the result concerning the effect of vegetation is consistent in all these different analyses? Which analysis would you prefer?

Compared to the difference of in week 2, the difference found here is somewhat smaller. The calculation in week 2 was a somewhat more naive approach, that did not take into considerations the difference between different areas. The difference found in week 4, again, is closer to this difference. As the calculation in week 4 took into consideration the possible differences in the sites, the logic of that is somewhat closer to this one, albeit the difference calculated above is for one specific site only.

The result is still somewhat consistent, a site without vegetation tends to have more white fish larvae than sites with vegetation.

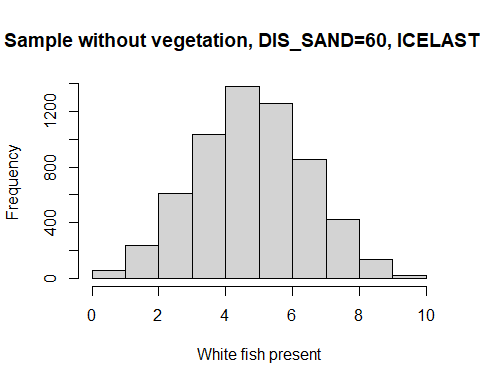
In my own opinion, this last one seems the most compelling model, as here we are also trying to understand the factors that affect the occurence of white fish larvae in different sites, and therefore it is also more informative. By building this model, we can actually make some kind of inference for a new site without calculation, just by looking at those parameters that we have found to be important.

**Task 6.**

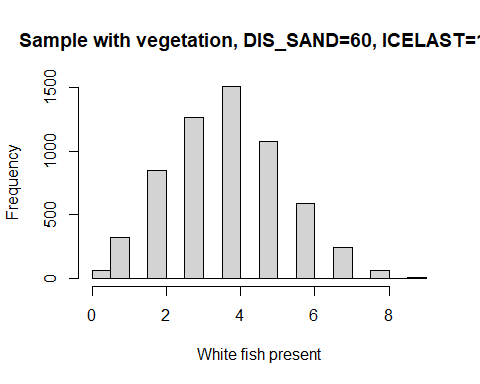
Visualize the posterior distribution of corresponding to the number sampling occasions where white fish is present out of a total 10 repeated sampling occasions at location where DIS\_SAND is 60 and ICELAST is 18 for both vegetated and non-vegetated bottom types.

Drawing random samples:

sample\_0 <- rbinom(theta0, 10, theta0)  
sample\_1 <- rbinom(theta1, 10, theta1)  
hist(sample\_0, main="Sample without vegetation, DIS\_SAND=60, ICELAST=18", xlab="White fish present")



hist(sample\_1, main="Sample with vegetation, DIS\_SAND=60, ICELAST=18", xlab="White fish present")



# Grading

**Total 20 points** Steps 1, 3 and 4 give 4 points each, steps 5 and 6 give 2 points each and step 2 gives 1 point if correctly solved. In other steps except 2 you may give half of the points if the step is solved half correctly. This could mean that some of the tasks have not been done (e.g. discussion is missing), there is only small typo that makes the final answer wrong or discussion is clearly not relevant or appropriate.

# References

Lari Veneranta, Richard Hudd and Jarno Vanhatalo (2013). Reproduction areas of sea-spawning Coregonids reflect the environment in shallow coastal waters. Marine Ecology Progress Series, 477:231-250. <http://www.int-res.com/abstracts/meps/v477/p231-250/>