

STAA 575 HW4

Jeremy Primus

Results Summary

A.

i)

	DIC	Effective No. of Parameters	Penalized DIC
Model 1	102.00	11.15	113.1
Model 2	102.40	11.69	114.1
Model 3	99.35	15.78	115.1
Model 4	113.90	4.08	118.0

ii) When we eliminate parameters from the model and see a parameter increase, it suggests that more unique random variables are being 'effective' to compensate for effects that were previously captured by former parameters. This is the case in model 3. We eliminate two parameters and actually see an increase in effective parameters. The model that is simplest in terms of effective parameters is the 4th, which eliminates random effects. This means that most of the behavior is captured by the physical parameters. However, this does have a much higher DIC, so eliminating random effects provides a worse overall fit. Generally, besides the model without random effects, the DICs are very similar.

iii) Model 1 has the lowest mean DIC and lowest DIC after assessing the parameter penalty. Model 2 is also very close, but the DIC and effective parameters give a slight edge to Model 1, so this model will be analyzed in greater detail.

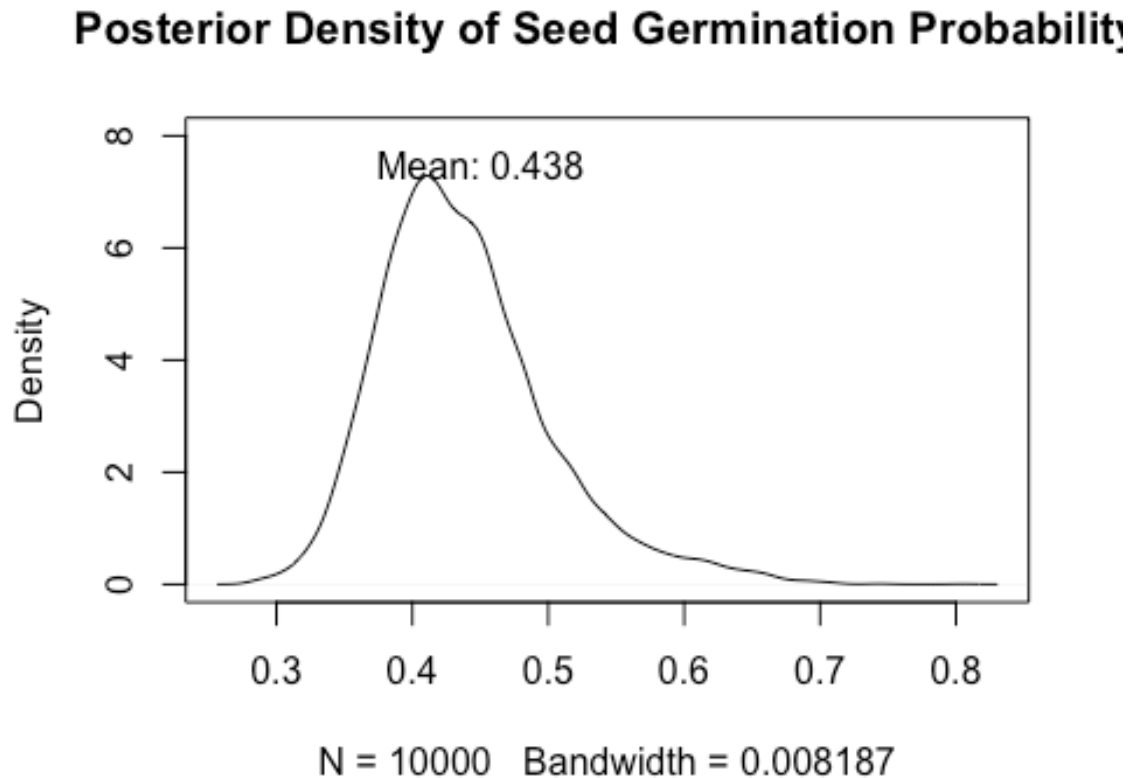
B. In this particular run, the sample path plots and ACF did not indicate any obvious problems with convergence. However, in some other runs I did notice that sharp spikes in the sample path plots were shared across the parameter set. The ACFs also have not died off completely. To be safe, I think this model should be run for double the iterations to ensure it converges.

C. Table of fit parameters

##	means	sds	lower	upper
## alpha0	-0.54	0.20	-0.93	-0.12
## alpha1	0.08	0.33	-0.59	0.72
## alpha12	-0.84	0.44	-1.72	0.05

```
## alpha2    1.34 0.28  0.79  1.89
## sigma     0.29 0.14  0.05  0.56
```

D. Summary plot



Conclusions.

The posterior density plot shows a distribution of seed germination with a mean of about 44%. The 95% credible interval stretches from 32-57%. The model showed no apparent convergence issues, but the correlation has not quite dropped low enough for my comfort yet. As well, some sample path runs from Model 1 gave indications of possible parameter correlations. The model should probably be run for more iterations for a more definitive analysis.

STAA575_HW4

Jeremy Primus

2/11/2018

Model Statement

Model 1: original

```
model1.seeds.text="model {
  for (i in 1:N) {
    b[i] ~ dnorm(0.0,tau)
    logit(p[i]) <- alpha0 + alpha1*x1[i] + alpha2*x2[i] +
      alpha12*x1[i]*x2[i] + b[i]
    r[i] ~ dbin(p[i],n[i])
  }
  alpha0 ~ dnorm(0.0,1.0E-6) # intercept
  alpha1 ~ dnorm(0.0,1.0E-6) # seed coeff
  alpha2 ~ dnorm(0.0,1.0E-6) # extract coeff
  alpha12 ~ dnorm(0.0,1.0E-6)
  tau ~ dgamma(1.0E-3,1.0E-3) # 1/sigma^2
  sigma <- 1.0/sqrt(tau)
}"
```

Model 2: no interaction effect

```
model2.seeds.text="model {
  for (i in 1:N) {
    b[i] ~ dnorm(0.0,tau)
    logit(p[i]) <- alpha0 + alpha1*x1[i] + alpha2*x2[i] + b[i]
    r[i] ~ dbin(p[i],n[i])
  }
  alpha0 ~ dnorm(0.0,1.0E-6) # intercept
  alpha1 ~ dnorm(0.0,1.0E-6) # seed coeff
  alpha2 ~ dnorm(0.0,1.0E-6) # extract coeff
  tau ~ dgamma(1.0E-3,1.0E-3) # 1/sigma^2
  sigma <- 1.0/sqrt(tau)
}"
```

Model 3: No root effect no interaction

```
model3.seeds.text="model {
  for (i in 1:N) {
    b[i] ~ dnorm(0.0,tau)
    logit(p[i]) <- alpha0 + alpha1*x1[i] + b[i]
    r[i] ~ dbin(p[i],n[i])
  }
  alpha0 ~ dnorm(0.0,1.0E-6) # intercept
  alpha1 ~ dnorm(0.0,1.0E-6) # seed coeff
  tau ~ dgamma(1.0E-3,1.0E-3) # 1/sigma^2
}"
```

```

    sigma <- 1.0/sqrt(tau)
  }"

# Model 4: no random effect
model4.seeds.text="model {
  for (i in 1:N) {
    logit(p[i]) <- alpha0 + alpha1*x1[i] + alpha2*x2[i] +
      alpha12*x1[i]*x2[i]
    r[i] ~ dbin(p[i],n[i])
  }
  alpha0 ~ dnorm(0.0,1.0E-6) # intercept
  alpha1 ~ dnorm(0.0,1.0E-6) # seed coeff
  alpha2 ~ dnorm(0.0,1.0E-6) # extract coeff
  alpha12 ~ dnorm(0.0,1.0E-6)
  tau ~ dgamma(1.0E-3,1.0E-3) # 1/sigma^2
  sigma <- 1.0/sqrt(tau)
}"

model.seeds.list <- list(model1.seeds.text, model2.seeds.text,
model3.seeds.text, model4.seeds.text)

# Model parameters
model.1.param <- c("alpha0", "alpha1","alpha2","alpha12","sigma")
model.2.param <- c("alpha0", "alpha1","alpha2","sigma")
model.3.param <- c("alpha0", "alpha1","sigma")
model.4.param <- c("alpha0", "alpha1","alpha2","alpha12")
model.params.list <- list(model.1.param, model.2.param, model.3.param,
model.4.param)

```

Data input

```

#Data
N <- 21
r <-c(10, 23, 23, 26, 17, 5, 53, 55, 32, 46, 10, 8, 10, 8, 23, 0, 3, 22,
15, 32, 3)
n <-c(39, 62, 81, 51, 39, 6, 74, 72, 51, 79, 13, 16, 30, 28, 45, 4, 12, 41,
30, 51, 7)
x1 <-c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1) #
group 75 (0) or group 73 (1)
x2 <-c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1) #
bean (0) or cucumber (1)
seeds.data=list(N=N,r=r,n=n,x1=x1,x2=x2)

```

Initial Values

```

#Initial Values:
#A diverse set of values for the GR statistic
# model 1 initse
inits1=list(

```

```

    list(tau =1,alpha0=0,alpha1=10,alpha2=-10,alpha12=5),
    list(tau =5,alpha0=5,alpha1=0,alpha2=-5,alpha12=10),
    list(tau =.1,alpha0=10,alpha1=5,alpha2=0,alpha12=-10))
# model 2 inits
inits2=list(
  list(tau =1,alpha0=0,alpha1=10,alpha2=-10),
  list(tau =5,alpha0=5,alpha1=0,alpha2=-5),
  list(tau =.1,alpha0=10,alpha1=5,alpha2=0))
# model 3 inits
inits3=list(
  list(tau =1,alpha0=0,alpha1=10),
  list(tau =5,alpha0=5,alpha1=0),
  list(tau =.1,alpha0=10,alpha1=5))
# model 4 inits
inits4=list(
  list(tau =1,alpha0=0,alpha1=10,alpha2=-10,alpha12=5),
  list(tau =5,alpha0=5,alpha1=0,alpha2=-5,alpha12=10),
  list(tau =.1,alpha0=10,alpha1=5,alpha2=0,alpha12=-10))

initslist <- list(inits1, inits2, inits3, inits4)

```

Calls to model

```

source("runEvaluateModels.R")
noOfModels <- length(model.seeds.list)
DIC <- list()
plotstf = FALSE
for (ii in seq(1, noOfModels)){
  DIC[[ii]] <- runEvaluateModels(model.seeds.list[[ii]],
model.params.list[[ii]], initslist[[ii]], seeds.data, plotstf)
  print(paste0("Model", ii))
  print(DIC[[ii]])
}

## [1] "Model1"
## Mean deviance: 102
## penalty 11.15
## Penalized deviance: 113.1
## [1] "Model2"
## Mean deviance: 102.4
## penalty 11.69
## Penalized deviance: 114.1
## [1] "Model3"
## Mean deviance: 99.35
## penalty 15.78
## Penalized deviance: 115.1
## [1] "Model4"
## Mean deviance: 113.9
## penalty 4.076
## Penalized deviance: 118

```

Create a table of DICs

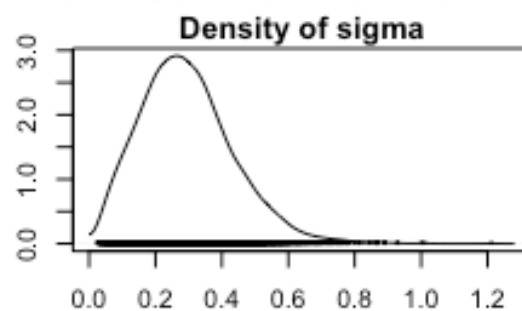
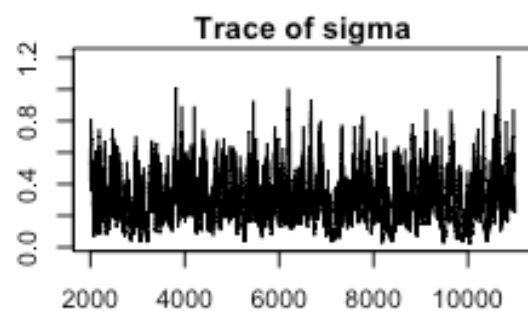
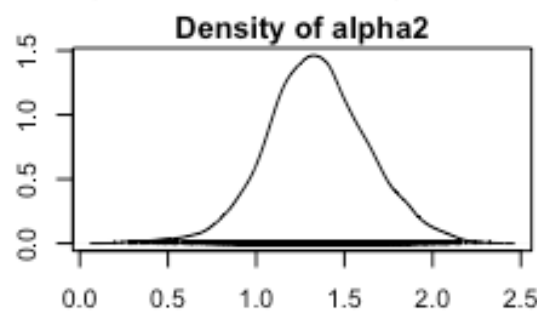
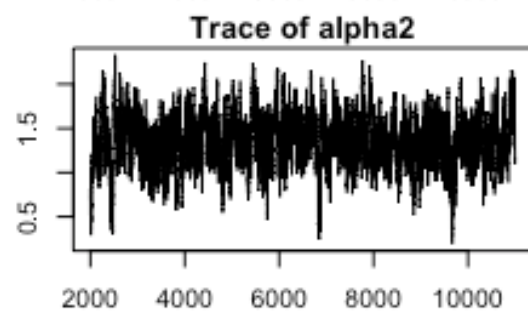
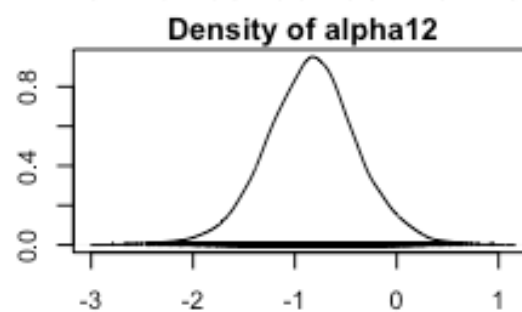
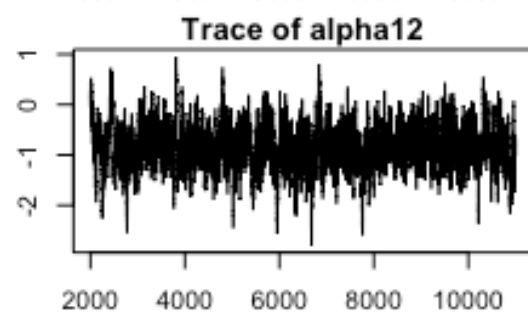
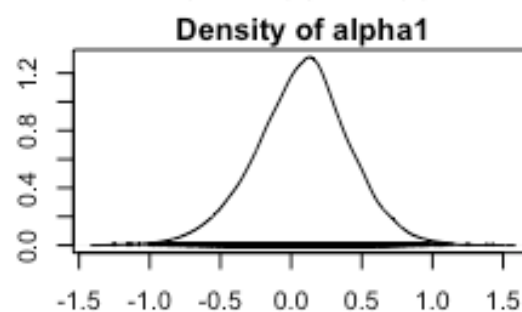
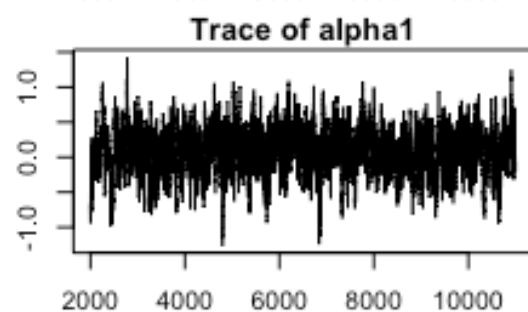
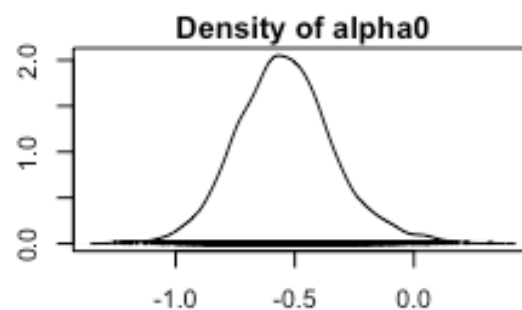
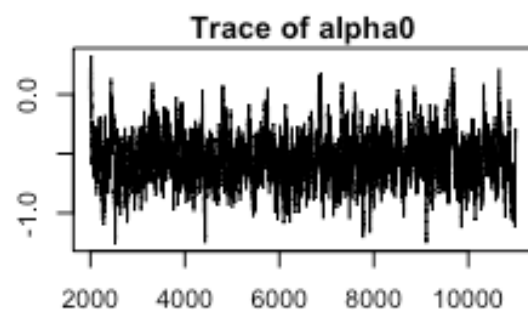
```
modelMeanDICs <- c(102.4, 102.3, 99.45, 113.8)
modelPenalty <- c(10.99, 11.77, 16.01, 3.935)
modelPenDev <- c(113.4, 114.1, 115.5, 117.8)
df <- data.frame(modelMeanDICs, modelPenalty, modelPenDev, row.names =
c("Model 1", "Model 2", "Model 3", "Model 4"))
colnames(df) <- c("DIC", "Effective No. of Parameters", "Penalized DIC")
kable(df)
```

	DIC	Effective No. of Parameters	Penalized DIC
Model 1	102.00	11.15	113.1
Model 2	102.40	11.69	114.1
Model 3	99.35	15.78	115.1
Model 4	113.90	4.08	118.0

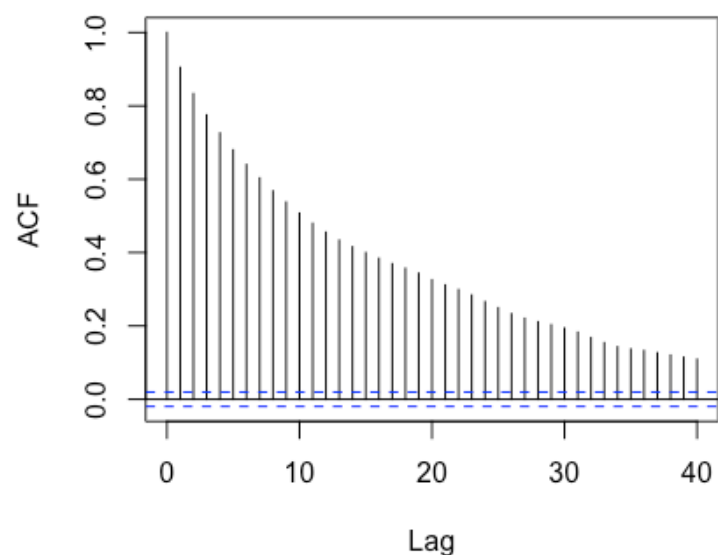
Interpret Results

Model 1 has the second lowest DIC, and after assessing a penalty for the effective number of parameters, Model 1 has the lowest DIC. Based upon this information, Model 1 will be selected for deeper analysis.

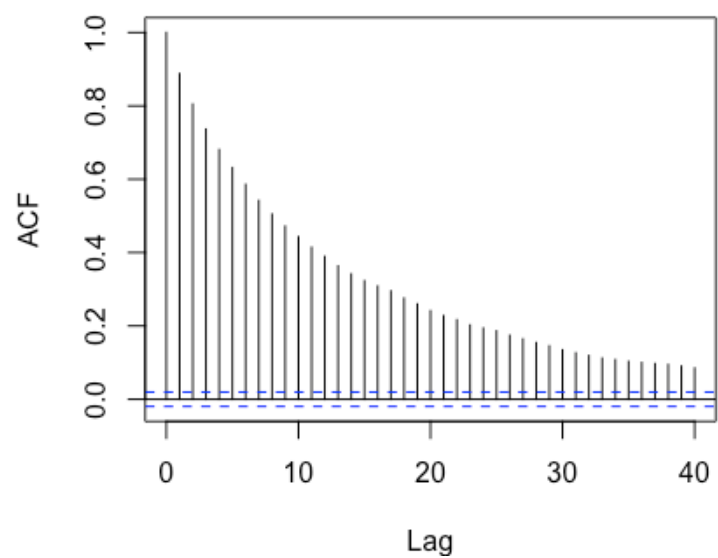
```
plotstf = TRUE
runEvaluateModels(model.seeds.list[[1]], model.params.list[[1]],
initslist[[1]], seeds.data, plotstf)
```



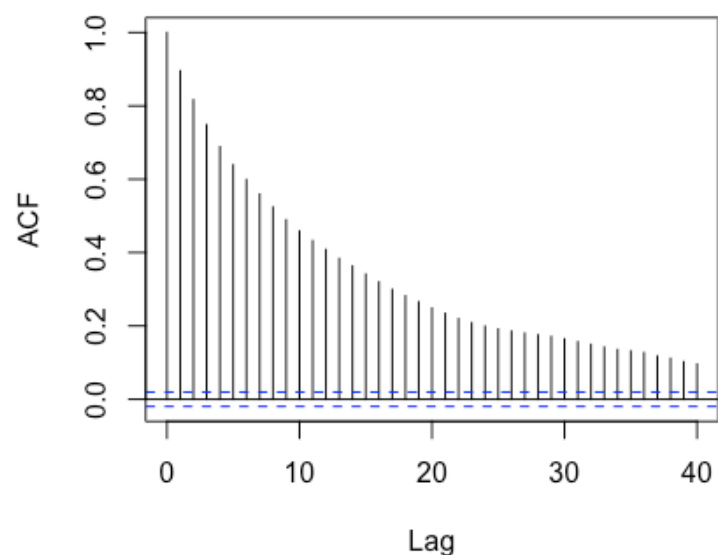
alpha0



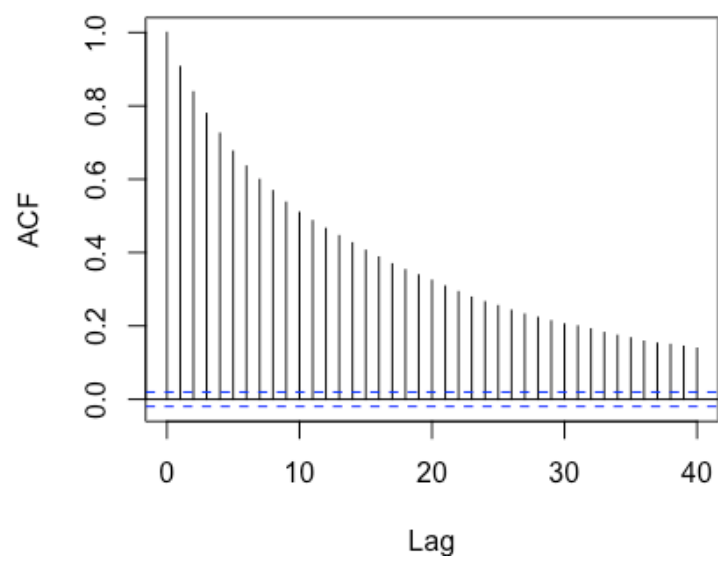
alpha1



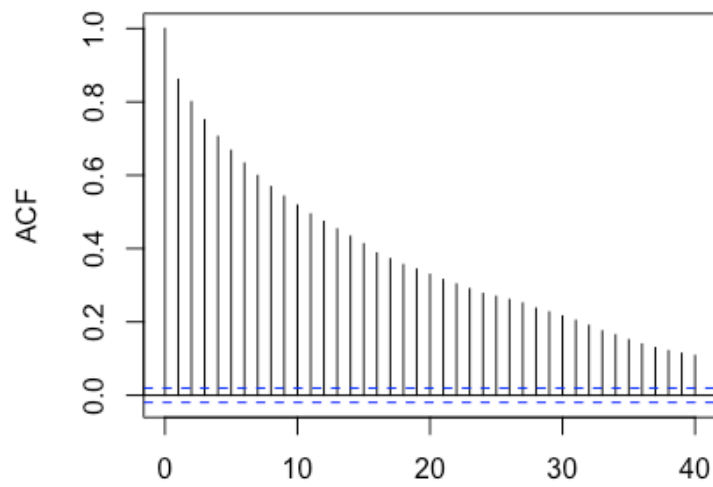
alpha2



alpha12

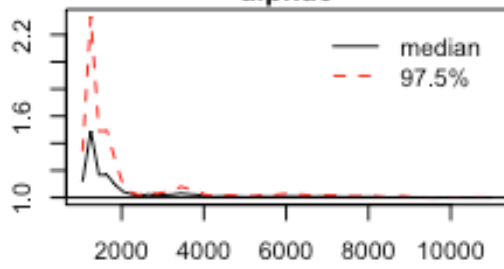


sigma

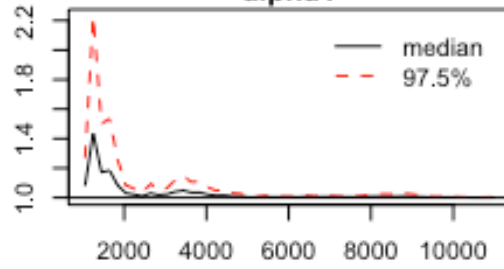


Lag

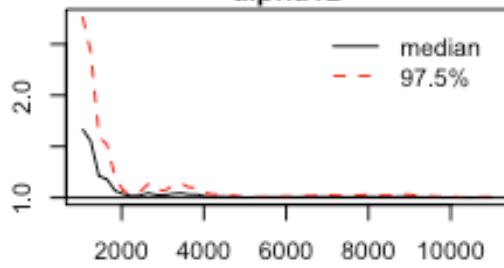
alpha0



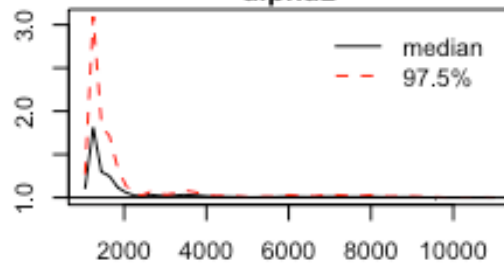
alpha1



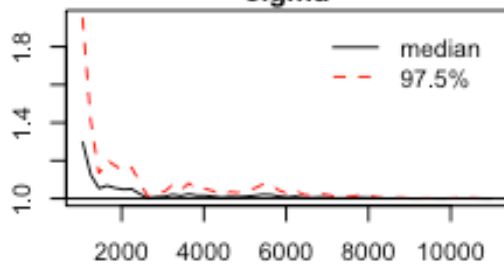
alpha12



alpha2

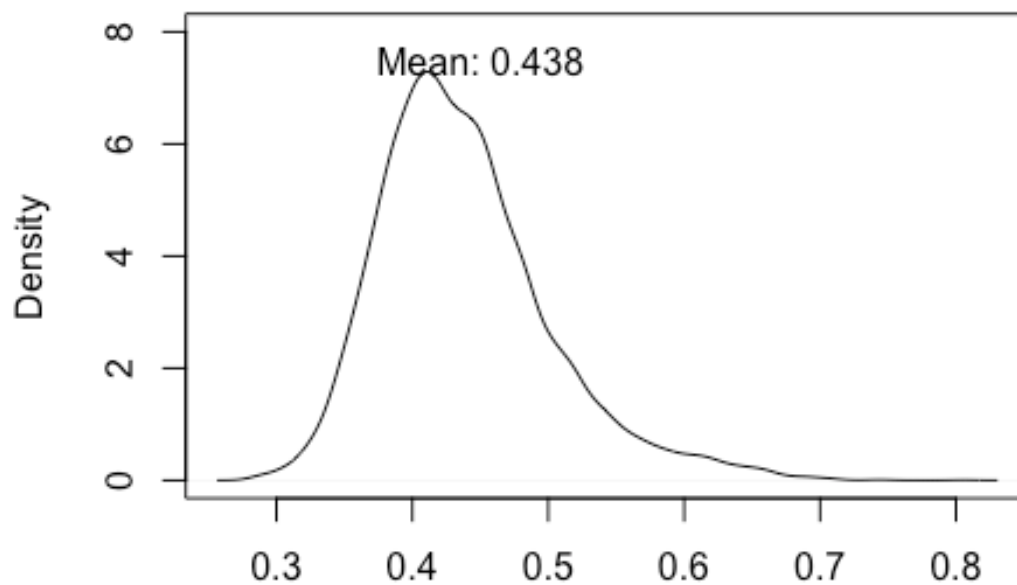


sigma



```
##      means  sds lower upper
## alpha0 -0.54 0.20 -0.93 -0.12
## alpha1  0.08 0.33 -0.59  0.72
## alpha12 -0.84 0.44 -1.72  0.05
## alpha2  1.34 0.28  0.79  1.89
## sigma   0.29 0.14  0.05  0.56
```

Posterior Density of Seed Germination Probability



N = 10000 Bandwidth = 0.008187

```
##      lower      upper
## var1 0.3217128 0.5697666
## attr("Probability")
## [1] 0.95

## Mean deviance: 102.1
## penalty 11.04
## Penalized deviance: 113.1
```

Analysis (b-d)

The sample path and ACF plots indicated no obvious convergence issues with Model 1. The posterior density suggests germination rates range from 32-57%, with a most probable germination rate of about 44%.

Additional code (functions)

```
runEvaluateModels <- function(model, model.params, inits, data.seeds,
plotstf){
  ## function to evaluate models and generate plots
  library(rjags)
  library(coda)
  library(knitr)
  library(gtools)

  ##Compile the model and run MCMC
  model.seeds <- textConnection(model)
  set.seed(1234)      # set seed for reproducibility
  seed.jags <- jags.model(model.seeds,
                           data=seeds.data,
                           inits=inits,
                           n.chains = 3,
                           n.adapt = 1000, quiet = TRUE)

  samps.coda <- (coda.samples(seed.jags, model.params,
                              n.iter=10000, thin=1))
  if(plotstf == TRUE){
    #an initial check of the output
    head(samps.coda[[1]])

    #Examine the summary output
    burn.in <- 2001
    #Examining one chain
    summary(window(samps.coda[[1]],start=burn.in))

    ### Some convergence diagnostics
    par(mar=c(2.0,2,1.5,2))
    plot(window(samps.coda[[1]],start=burn.in)) # plot sample path and
density

    # Auto-correlation function
    k <- length(model.params)
    par(mar=rep(4,4))
    for (i in seq(1,5)){
      acf(samps.coda[[1]][,i], main = model.params[i])
    }

    # Caculate GR Statistic
    par(mar=c(2.0,2,1.5,2))
    gelman.diag(samps.coda)
    gelman.plot(samps.coda)

    #Create a table of the results
    #Compute mean, sd, and 95% HPD interval:
    means=apply(samps.coda[[1]][2001:10000,],2,mean)
```

```

sds=apply(samps.coda[[1]][2001:10000,],2,sd)
hpds=HPDinterval(mcmc(samps.coda[[1]][2001:10000,]))
print(round(cbind(means,sds,hpds),2))
par(mar=c(5.1,4.1,4.1,2.1))
p <- samps.coda[[1]][,1] + samps.coda[[1]][,2]*x1[i] +
samps.coda[[1]][,3]*x2[i] +
samps.coda[[1]][,4]*x1[i]*x2[i] + samps.coda[[1]][,5]
plot(density(inv.logit(p)), main = 'Posterior Density of Seed
Germination Probability', ylim = c(0,8))
postMean <- round(mean(inv.logit(p)),3)
text(0.45, 7.5, paste0('Mean: ', postMean))
print(HPDinterval(mcmc(inv.logit(p))))
# hist(inv.logit(p), main = 'Posterior Density of Seed Germination
Probability', breaks = 50)

}
# Compute the DIC
DIC <- dic.samples(seed.jags, n.iter = 10000, type = "pD", burn.in=2000)
return(DIC)
}

```