University of Edinburgh
School of Mathematics
Bayesian Data Analysis, 2018/2019, Semester 2
Jonathan Gair & Rubén Amorós Salvador

May 13, 2019

Assignment 2 - Solutions

- To be uploaded to Learn by 23:59, Sunday 21 April, 2019.
- This assignment is worth 50% of your final grade for the course.
- Assignments should be typed (LaTeX, Word, etc.) and should be no more than 10 pages using a type size no smaller than 11 point and with 1.5-2.0 line space. This includes figures but excludes the appended code. Document your code so that someone can read it without too much guesswork.
- Answers to questions should be in full sentences.
- Any output (e.g., graphs, tables) from `R/JAGS` that you use to answer questions must be included with the assignment. You will want to be judicious with what you include in the written report—not every figure and table you constructed needs to be included. Also, please append your `R/JAGS` code at the end of the assignment.
- The assignment is out of 100 marks.
- You are expected to work independently and not discuss the assignment with others.
- We recommend that you examine the advice given during lecture 6 about both writing assignments and carrying out Bayesian analysis.
- Briefly indicate the technical details of the MCMC analyses you perform (number of iterations, convergence checks, etc.).
- You should write the hierarchical expressions (in terms of probability distributions) of all your models.

1. *Modelling abundance of gulls* **(44 marks)**

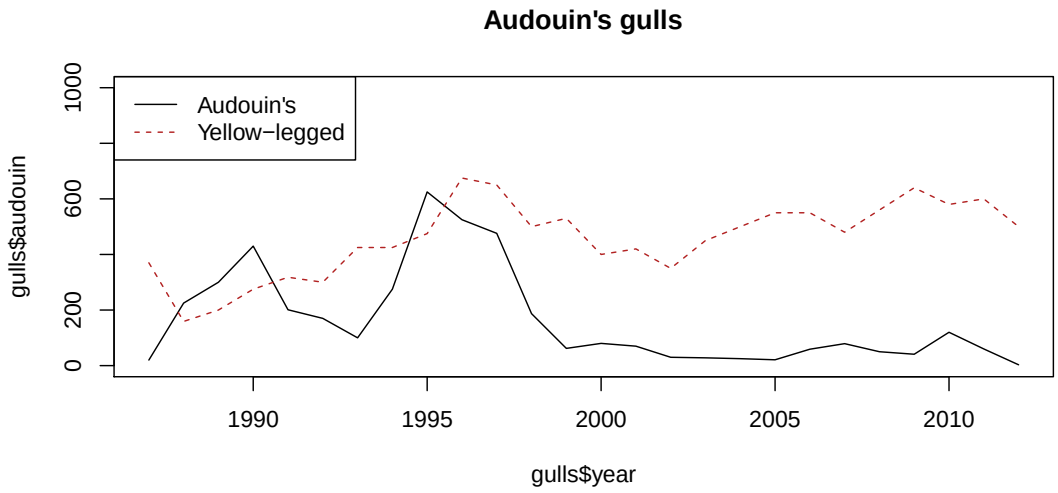
   The Columbretes Islands[1] is a group of uninhabited islets in the east coast of Spain. Two species of gulls (Audouin's[2] and Yellow-legged[3]) nest there every year. The cycle of reproduction of the Yellow-legged gulls starts earlier in the year so they start nesting in the islets earlier than the Audouin's gulls, and ecologists theorize that the Yellow-legged gulls are displacing the Audouin's gulls. The file `gulls_data.csv` contains the complete census counts of the nesting couples of both species of gulls during the years 1987-2012.

```
# Loading the data
gulls <- read.csv("gulls_data.csv")

# We can define a database with all the data for all the JAGS models
#    (we will get some warnings, though)
gulls.data <- list(n=nrow(gulls), audouin=gulls$audouin, year=gulls$year,
                   yellowlegged=gulls$yellowlegged)
```

[1] https://en.wikipedia.org/wiki/Columbretes_Islands
[2] https://en.wikipedia.org/wiki/Audouin%27s_gull
[3] https://en.wikipedia.org/wiki/Yellow-legged_gull

(a) **(3 marks)** Perform some exploratory data analysis including a graph of the temporal evolution of the abundance of Audouin's gulls and a graph showing the relation between the abundance of the two species of gulls. Briefly comment this analysis.

```
plot(gulls$year, gulls$audouin, ylim=c(0,1000), main="Audouin's gulls",
     type = "l")
lines(gulls$year, gulls$yellowlegged, ylim=c(0,1000), col="firebrick", lty=2)
legend("topleft", legend=c("Audouin's", "Yellow-legged"), lty = 1:2,
       col=c("black","firebrick"))
```




```
cor(gulls$yellowlegged, gulls$audouin)

## [1] -0.04226575
```

Solution: The abundance of Audouin's gulls has two peaks around 1990 and 1996 and stays low after year 2000. The abundance of Yellow-legged gulls seems to increase, mostly until 1997. The linear correlation between the two variables, though is close to zero.

(b) **(10 marks)** Fit a Bayesian Poisson model with a logarithm link function where the abundance of the Audouin's gull couples is considered to be dependent on the year. Summarize the estimated parameters and briefly interpret them.

```
# Block model with year covariate
gulls.model.year <-    "model {
  #Likelihood
  for(i in 1:n) {
    audouin[i] ~ dpois(mu[i])
    log(mu[i]) <- beta0 + beta.yr*(year[i] - mean(year[]))

    # Prediction
    audouin.rep[i] ~ dpois(mu[i])
  }
```

```
  # prior
  beta0 ~ dnorm(0,0.0001)
  beta.yr ~ dnorm(0,0.0001)
}"
```

```
# Initial values
gulls.year.inits <- function(){list(beta0=rnorm(1,0,10), beta.yr=rnorm(1,0,10))}

# Running JAGS
gulls.year.res.A <- jags.model(file=textConnection(gulls.model.year),
                               data=gulls.data,  n.chains=3,
                               inits = gulls.year.inits, quiet = TRUE)
update(gulls.year.res.A, n.iter=5000)
gulls.year.res.B <- coda.samples(gulls.year.res.A, variable.names=c("beta0",
                                  "beta.yr", "audouin.rep"), n.iter=100000)
# Getting DIC
gulls.year.DIC <- dic.samples(model=gulls.year.res.A, n.iter = 100000,
                              type = "pD")

# Joinning all the chains in one data.frame
gulls.year.output <- do.call(rbind.data.frame, gulls.year.res.B)

# Checking the chains
mcmcplots::mcmcplot(gulls.year.res.B, parms = c("beta0", "beta.yr"))

# Checking the effective sample size
effectiveSize(gulls.year.res.B)       # (all > 100 000)

# Checking the Gelman-Ruben-Brooks statistic
gelman.diag(gulls.year.res.B)         # (all = 1)

# Summary
gulls.year.summ <- summary(gulls.year.res.B)
```

```
# Summary of the results
gulls.year.summ$statistics[c("beta.yr", "beta0"),]
```

```
##                Mean         SD     Naive SE Time-series SE
## beta.yr -0.07557711 0.00223877 4.087417e-06   6.568452e-06
## beta0    4.94333153 0.01772280 3.235725e-05   5.269360e-05
```

```
gulls.year.summ$quantiles[c("beta.yr", "beta0"),]
```

```
##                2.5%         25%         50%         75%      97.5%
## beta.yr -0.07997417 -0.07708428 -0.07556956 -0.07406556 -0.0711998
## beta0    4.90845851  4.93139697  4.94336737  4.95530672  4.9779137
```

```
mean(exp(gulls.year.output$beta.yr))
```

```
## [1] 0.9272106
```

```r
mean(exp(gulls.year.output$beta0))

## [1] 140.2587
```

Solution: The parameter β_{yr} is estimated to be negative, which indicates that the abundance of Audouin's gulls decrease with time. The mean quantity of Audouin's gull couples is estimated to be multiplied by $E(e^{\beta_{yr}}) = 0.927$ each year, equivalent to a yearly decrease of 7.3%. The estimated number of Audouin's gull couples for the center of the period (between 1999 and 2000) is $E(e^{\beta_0}) = 140.3$.

(c) **(9 marks)** Expand the previous model by including an extra variance term in the regressor. Set a Gaussian prior for this extra variance term with zero mean and standard deviation with a uniform hyper-prior distribution between 0 and 10.

```r
# Block model with year covariate and extra variance
gulls.model.yextravar <-    "model {
  #Likelihood
  for(i in 1:n) {
    audouin[i] ~ dpois(mu[i])
    log(mu[i]) <- beta0 + beta.yr*(year[i] - mean(year[])) + epsilon[i]
    epsilon[i] ~ dnorm(0, tau.epsilon)

    # Prediction
    audouin.rep[i] ~ dpois(mu.rep[i])
    log(mu.rep[i]) <- beta0 + beta.yr*(year[i] - mean(year[])) + epsilon.rep[i]
    epsilon.rep[i] ~ dnorm(0, tau.epsilon)
  }

  # prior
  beta0 ~ dnorm(0,0.0025)
  beta.yr ~ dnorm(0,0.0025)
  tau.epsilon <- pow(sigma.epsilon, -2)
  sigma.epsilon ~ dunif(0,10)
}"
```

```r
# Initial values
gulls.yextravar.inits <- function(){list(beta0=rnorm(1,0,20),
                                         beta.yr=rnorm(1,0,20),
                                         sigma.epsilon=runif(1,0,10))}
# Running JAGS
gulls.yextravar.res.A <- jags.model(file=textConnection(gulls.model.yextravar),
                                    data=gulls.data,  n.chains=3,
                                    inits = gulls.yextravar.inits, quiet = TRUE)
update(gulls.yextravar.res.A, n.iter=100000)
gulls.yextravar.res.B <- coda.samples(gulls.yextravar.res.A,
                                      variable.names=c("beta0", "beta.yr",
                                                       "sigma.epsilon",
                                                       "audouin.rep"),
                                      n.iter=200000)
```

```r
# Getting DIC
gulls.yextravar.DIC <- dic.samples(model=gulls.yextravar.res.A, n.iter = 200000,
                                   type = "pD")

# Joinning all the chains in one data.frame
gulls.yextravar.output <- do.call(rbind.data.frame, gulls.yextravar.res.B)

# Checking the chains
mcmcplots::mcmcplot(gulls.yextravar.res.B, parms = c("beta0",
                                  "beta.yr", "sigma.epsilon"))

# Checking the effective sample size
effectiveSize(gulls.yextravar.res.B)       # (all > 1000)

# Checking the Gelman-Ruben-Brooks statistic
gelman.diag(gulls.yextravar.res.B)         # (all < 1.1)

# Summary
gulls.yextravar.summ <- summary(gulls.yextravar.res.B)
```

```r
# Summary of the results
gulls.yextravar.summ$statistics[c("beta.yr", "beta0", "sigma.epsilon"),]
```

```
##                      Mean        SD     Naive SE Time-series SE
## beta.yr       -0.09352985 0.0272516 3.518166e-05   0.0007493974
## beta0          4.51224699 0.2150777 2.776641e-04   0.0067922000
## sigma.epsilon  1.05592681 0.1711193 2.209141e-04   0.0008995572
```

```r
gulls.yextravar.summ$quantiles[c("beta.yr", "beta0", "sigma.epsilon"),]
```

```
##                     2.5%        25%         50%         75%       97.5%
## beta.yr       -0.1479417 -0.1114082 -0.09359568 -0.07570662 -0.04008188
## beta0          4.0813592  4.3722212  4.51575779  4.65422736  4.92676033
## sigma.epsilon  0.7802245  0.9350829  1.03571009  1.15431570  1.44808023
```

```r
mean(exp(gulls.yextravar.output$beta.yr))
```

```
## [1] 0.9110491
```

```r
mean(exp(gulls.yextravar.output$beta0))
```

```
## [1] 93.24166
```

Solution: The estimated values of the parameters noticeably change. The mean quantity of Audouin's gull couples is estimated to be multiplied by $E(e^{\beta_{yr}}) = 0.911$ each year, equivalent to a yearly decrease of 8.9%. The estimated number of Audouin's gull couples for the center of the period (between 1999 and 2000) is $E(e^{\beta_0}) = 93.2$. The extra variance term's standard deviation is estimated to be $E(\sigma_\epsilon) = 1.06$, a non neglegible quantity compared to the estimate for β_0.

(d) **(5 marks)** Further expand the previous extra variance model by including the number of

Yellow-legged gull couples as an explanatory variable. Discuss the posterior estimation for the parameter of the Yellow-legged gull. Based on the exploratory analysis, is it what you expected to obtain? Consider in your discussion the variability of abundance of resources (food) among years.

```
# Block model with year and Yellow-legged covariates and extra variance
gulls.model.yevyl <-    "model {
  #Likelihood
  for(i in 1:n) {
    audouin[i] ~ dpois(mu[i])
    log(mu[i]) <- beta0 + beta.yr*(year[i] - mean(year[])) + epsilon[i] +
                      beta.yl*(yellowlegged[i] - mean(yellowlegged[]))
    epsilon[i] ~ dnorm(0, tau.epsilon)

    # Prediction
    audouin.rep[i] ~ dpois(mu.rep[i])
    log(mu.rep[i]) <- beta0 + beta.yr*(year[i] - mean(year[])) + epsilon.rep[i] +
                      beta.yl*(yellowlegged[i] - mean(yellowlegged[]))
    epsilon.rep[i] ~ dnorm(0, tau.epsilon)
  }

  # prior
  beta0 ~ dnorm(0,0.0025)
  beta.yr ~ dnorm(0,0.0025)
  beta.yl ~ dnorm(0,0.01)
  tau.epsilon <- pow(sigma.epsilon, -2)
  sigma.epsilon ~ dunif(0,10)
}"
```

```
# Initial values
gulls.yevyl.inits <- function(){list(beta0=rnorm(1,0,2), beta.yr=rnorm(1,0,2),
                                     beta.yl=rnorm(1,0,2),
                                     sigma.epsilon=runif(1,0,1))}

# Running JAGS
gulls.yevyl.res.A <- jags.model(file=textConnection(gulls.model.yevyl),
                                data=gulls.data,  n.chains=3,
                                inits = gulls.yevyl.inits, quiet = TRUE)
update(gulls.yevyl.res.A, n.iter=150000)
gulls.yevyl.res.B <- coda.samples(gulls.yevyl.res.A, variable.names=c("beta0",
                                    "beta.yr", "beta.yl", "sigma.epsilon",
                                    "audouin.rep"), n.iter=600000)
# I have increased n.iter to make the SD for beta.yl be around 20 times the
#     MCMC standard error for this parameter.
# Getting DIC
gulls.yevyl.DIC <- dic.samples(model=gulls.yevyl.res.A, n.iter = 600000,
                               type = "pD")

# Joinning all the chains in one data.frame
```

```r
gulls.yevyl.output <- do.call(rbind.data.frame, gulls.yevyl.res.B)

# Checking the chains
mcmcplots::mcmcplot(gulls.yevyl.res.B, parms = c("beta0", "beta.yr", "beta.yl",
                                                 "sigma.epsilon"))

# Checking the effective sample size
effectiveSize(gulls.yevyl.res.B)       # (all > 1200)

# Checking the Gelman-Ruben-Brooks statistic
gelman.diag(gulls.yevyl.res.B)         # (all < 1.1)

# Summary
gulls.yevyl.summ <- summary(gulls.yevyl.res.B)
```

```r
# Summary of the results
gulls.yevyl.summ$statistics[c("beta.yr", "beta0", "beta.yl", "sigma.epsilon"),]
```

```
##                       Mean          SD     Naive SE Time-series SE
## beta.yr       -0.138629539 0.037269442 2.777900e-05   7.897128e-04
## beta0          4.504993267 0.203310622 1.515388e-04   3.466767e-03
## beta.yl        0.003801643 0.002102239 1.566917e-06   5.989344e-05
## sigma.epsilon  1.000620095 0.167913770 1.251555e-04   8.337293e-04
```

```r
gulls.yevyl.summ$quantiles[c("beta.yr", "beta0", "beta.yl", "sigma.epsilon"),]
```

```
##                        2.5%          25%          50%          75%
## beta.yr       -0.2151874693 -0.162167816 -0.137833535 -0.114006547
## beta0          4.1002631585  4.372913000  4.506939914  4.640056936
## beta.yl       -0.0003193191  0.002420221  0.003793136  0.005165326
## sigma.epsilon  0.7323809402  0.882120117  0.980115035  1.096355416
##                      97.5%
## beta.yr       -0.067457438
## beta0          4.898898543
## beta.yl        0.008010023
## sigma.epsilon  1.385494738
```

```r
mean(exp(gulls.yevyl.output$beta.yr))
```

```
## [1] 0.8711541
```

```r
mean(exp(gulls.yevyl.output$beta0))
```

```
## [1] 92.34499
```

```r
mean(exp(gulls.yevyl.output$beta.yl))
```

```
## [1] 1.003811
```

```r
mean(exp(100*gulls.yevyl.output$beta.yl))
```

```
## [1] 1.495509
```

```
mean(gulls.yevyl.output$beta.yl>0)

## [1] 0.965375
```

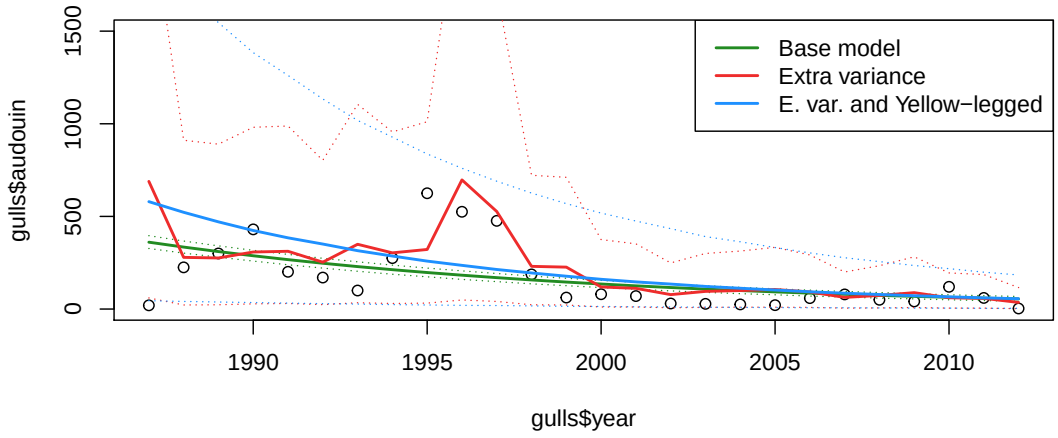
Solution: The mean quantity of Audouin's gull couples is estimated by this model to be multiplied by $E(e^{\beta_{yr}}) = 0.871$ each year, equivalent to a yearly decrease of 12.9%. The estimated number of Audouin's gull couples for the center of the period (between 1999 and 2000) almost doesn't change from the extra variance model, $E(e^{\beta_0}) = 92.3$. The mean number of Audouin's gull couples is estimated to be multiplied by $E(e^{100 \cdot \beta_{yl}}) = 1.496$ for each extra 100 couples of Yellow-legged gulls, equivalent to an increase of almost 50%. The posterior probability of β_{yl} being positive is estimated to be $Pr(\beta_{yl}|Y) = .965$, therefore there is notable evidence that suggests that the abundance of the two species for each year is positively related. This seems to contrat with the exploratory analysis which suggested no relation between the two abundances, but this model also takes into account the long term trend, so the covariate Yellow-legged most likely captures part of the variability around that decreasing mean. A possible explanation of this relation could be the abundance of resources each year, so that years with higher abundance of resources will result in a higher presence of both species.

(e) **(12 marks)** Compute random samples of the predictive distributions for replicates of your database (same years and abundance of Yellow-legged gull as covariates) for the three models. Plot the abundance of Audouin's gulls with the posterior predictive expected values and 90% credible interval bands of the replicates for the three models.

NOTE: Take into account that you should NOT compute the posterior predictive distribution of the replicates conditional on the extra variance term for each particular observation ($Pr(Y_i^{rep}|\boldsymbol{Y}, \varepsilon_i)$). This extra variance term should instead be integrated out in the predictive distribution (you should compute $Pr(Y_i^{rep}|\boldsymbol{Y})$).

```
# Getting the mean, 5th and 95th percentile for the predictions
EPred.year <- apply(gulls.year.output, 2, mean)[1:26]
EPred5.year <- apply(gulls.year.output, 2, quantile, probs=0.05)[1:26]
EPred95.year <- apply(gulls.year.output, 2, quantile, probs=0.95)[1:26]

EPred.yextravar <- apply(gulls.yextravar.output, 2, mean)[1:26]
EPred5.yextravar <- apply(gulls.yextravar.output, 2, quantile, probs=0.05)[1:26]
EPred95.yextravar <- apply(gulls.yextravar.output, 2, quantile,probs=0.95)[1:26]

EPred.yevyl <- apply(gulls.yevyl.output, 2, mean)[1:26]
EPred5.yevyl <- apply(gulls.yevyl.output, 2, quantile, probs=0.05)[1:26]
EPred95.yevyl <- apply(gulls.yevyl.output, 2, quantile, probs=0.95)[1:26]

# Plotting the observations
plot(gulls$year, gulls$audouin, ylim=c(0,1500))
# Adding the prediction for the replicates
# Model with year
lines(gulls$year, EPred.year, col="forestgreen", lwd=2)
lines(gulls$year, EPred5.year, col="forestgreen", lty=3)
```

```
lines(gulls$year, EPred95.year, col="forestgreen", lty=3)
# Model with year and extra variance
lines(gulls$year, EPred.yevyl, col="firebrick2", lwd=2)
lines(gulls$year, EPred5.yevyl, col="firebrick2", lty=3)
lines(gulls$year, EPred95.yevyl, col="firebrick2", lty=3)
# Model with year, Yellow-legged gulls and extra variance
lines(gulls$year, EPred.yextravar, col="dodgerblue", lwd=2)
lines(gulls$year, EPred5.yextravar, col="dodgerblue", lty=3)
lines(gulls$year, EPred95.yextravar, col="dodgerblue", lty=3)

legend("topright",
       legend = c("Base model", "Extra variance", "E. var. and Yellow-legged"),
       lwd = 2, col=c("forestgreen", "firebrick2", "dodgerblue"))
```



(f) **(5 marks)** Compute the Deviance Information Criterion (DIC) of the three models and compare the performance of the three models (according to DIC and all the previous analyses). In particular you should discuss the relevance of the extra variance term and of the covariate term. Which model or models do you consider adequate for modelling this data?

```
gulls.year.DIC

## Mean deviance:  2984
## penalty 2.008
## Penalized deviance: 2986

gulls.yextravar.DIC

## Mean deviance:  191
## penalty 25.62
## Penalized deviance: 216.6

gulls.yevyl.DIC
```

```
## Mean deviance:  191
## penalty 25.56
## Penalized deviance: 216.6
```

Solution: The first model is not well defined for this problem, as the significantly higher DIC score suggests, because the assumption of equal mean and variance of the Poisson model can not capture the high variability of the data. The second and third models offer similar trade-off between goodness-of-fit and complexity according to DIC. The effect of the abundance of Yellow-legged gulls appears to help better model the high number of Audouin's gulls in years 1996 and 1997 and the suddent drop on 1998, but it fails to capture other variations, so we are not sure if this relation can be extrapolated to future observations. It is also worth mentioning that there is a reduction on the posterior expected value of σ_ϵ from the second to the third model, but this reduction is not big (around 5%), so the explanatory variable seems to explain only a small proportion of the variability.

2. *Modelling presence of parasites in cows* **(44 marks)**

Hairworms are a common parasite that can infect the gastrointestinal tract of cows. A study was performed to assess risk factors related to the presence of this parasite in cattle. The file `cows.csv` contains data on 279 cows (one per rows) from 18 different farms with the following variables:

cowID Identification number of the cow.

farmID Identification number of the farm of the cow.

temp Average daily maximum temperature (Celsius) measured at the nearest weather station to the farm.

rain Average yearly rain (millimetres) measured at the nearest weather station to the farm.

permeab Permeability of the soil around the farm (1-low permeability, water can stagnate, 2-high permeability, water is absorbed by the soil easily)

height Height of the location of the farm (meters).

slope Average slope of the fields around the farm (percentage).

age Age of the cow (years)

parasite Presence (1) or absence (0) of the parasite in the gastrointestinal tract of the cow.

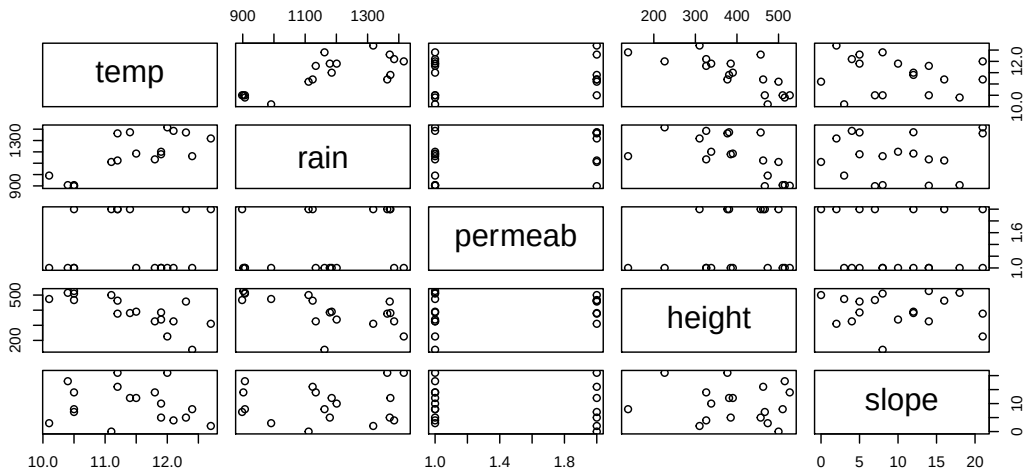
It should be noted that only a sub-sample of the cows of each farm was analysed.

```
# Loading the data
cows <- read.csv(file = "cows.csv")
# Correction of one variable name
names(cows)[6] <- "height"
# Auxiliary data.frame with only data for the farms (1 row per farm)
farms <- unique(cows[,-c(1,8,9)])

# We can define a database with all the data for all the JAGS models
```

```
#    (we will get some warnings, though)
cows.data <- list(n=dim(cows)[1], Nfarm=length(unique(cows$farmID)),
                  farmID=cows$farmID, temp=cows$temp, rain=cows$rain,
                  permeab=cows$permeab-1, height=cows$height, slope=cows$slope,
                  age=cows$age, parasite=cows$parasite)
# NOTE: We transform the variable permeab so that it is 0's and 1's

# Function inverse-logit
ilogit <- function(x) 1/(1+exp(-x))
```

(a) **(3 marks)** Perform some exploratory analysis of the database, including an analysis of the correlation between the variables of the environment of the farm (temperature, rain, permeability, height and slope). Briefly comment this analysis.

```
plot(farms[,-1])
```




```
cor(farms[,-1])          # Correlation among farm variables

##                temp        rain     permeab     height       slope
## temp     1.00000000  0.77483062  0.07279703 -0.7653097 -0.16035673
## rain     0.77483062  1.00000000  0.26857189 -0.6067080  0.06742564
## permeab  0.07279703  0.26857189  1.00000000  0.2100778 -0.12810357
## height  -0.76530965 -0.60670798  0.21007777  1.0000000 -0.10179237
## slope   -0.16035673  0.06742564 -0.12810357 -0.1017924  1.00000000

cor(cows[,-c(1,2,8,9)])  # Not so correct way of checking the correlation

##               temp       rain   permeab     height      slope
## temp     1.0000000  0.7443559 0.1056443 -0.5337633 -0.3007426
## rain     0.7443559  1.0000000 0.3485617 -0.3143744  0.1249165
## permeab  0.1056443  0.3485617 1.0000000  0.3393498  0.0235213
## height  -0.5337633 -0.3143744 0.3393498  1.0000000 -0.1121208
## slope   -0.3007426  0.1249165 0.0235213 -0.1121208  1.0000000
```

Solution: We observe considerably high correlations among several variables of the farms and, in particular, for the variables rain, height and temperature. They will probably compete to explain the presence of the parasite.

(b) **(12 marks)** Fit a Bayesian hierarchical Bernoulli logistic model where the probability for each cow of having hairworm parasites is explained by their own covariates (age), the covariates of the environment of the farm (temperature, rain, permeability, height and slope) and a random effect on the farms themselves. Discuss the estimates for the posterior distribution of the parameters.

```
# Block model
cows.model.all <-    "model {
  #Likelihood
  for(i in 1:n) {
    parasite[i] ~ dbern(p[i])
    logit(p[i]) <- beta0 + b.farm[farmID[i]]+
                   beta.temp*(temp[i] - mean(temp[])) +
                   beta.rain*(rain[i] - mean(rain[])) +
                   beta.permeab* permeab[i] +   #NOTE: factor, we do not centre
                   beta.height*(height[i] - mean(height[])) +
                   beta.slope*(slope[i] - mean(slope[])) +
                   beta.age*(age[i] - mean(age[]))
  }

  for(j in 1:Nfarm){
    b.farm[j] ~ dnorm(0, tau.farm)
  }

  # priors
  beta0 ~ dnorm(0,0.1)

  beta.temp ~ dnorm(0,0.1)
  beta.rain ~ dnorm(0,0.1)
  beta.permeab ~ dnorm(0,0.1)
  beta.height ~ dnorm(0,0.1)
  beta.slope ~ dnorm(0,0.1)
  beta.age ~ dnorm(0,0.1)

  tau.farm <- pow(sigma.farm, -2)
  sigma.farm ~ dunif(0,10)
}"
```

```
# Initial values
cows.all.inits <- function(){list(beta0=rnorm(1,0,0.1),
                                  beta.temp=rnorm(1,0,0.1),
                                  beta.rain=rnorm(1,0,0.1),
                                  beta.permeab=rnorm(1,0,0.1),
                                  beta.height=rnorm(1,0,0.1),
                                  beta.slope=rnorm(1,0,0.1),
```

```
                                  beta.age=rnorm(1,0,0.1),
                                  sigma.farm=runif(1,0,0.1))}


# Running JAGS
cows.all.res.A <- jags.model(file=textConnection(cows.model.all),
                             data=cows.data,  n.chains=3,
                             inits = cows.all.inits, quiet = TRUE)
update(cows.all.res.A, n.iter=5000)
cows.all.res.B <- coda.samples(cows.all.res.A, variable.names=c("beta0",
                                  "beta.temp", "beta.rain", "beta.permeab",
                                  "beta.height", "beta.slope", "beta.age",
                                  "sigma.farm", "b.farm"), n.iter=50000)
# Getting DIC
cows.all.DIC <- dic.samples(model=cows.all.res.A, n.iter = 50000,
                                type = "pD")

# Joinning all the chains in one data.frame
cows.all.output <- do.call(rbind.data.frame, cows.all.res.B)

# Checking the chains
mcmcplots::mcmcplot(cows.all.res.B, parms = c("beta0",
                                  "beta.temp", "beta.rain", "beta.permeab",
                                  "beta.height", "beta.slope", "beta.age",
                                  "sigma.farm"))

# Checking the effective sample size
effectiveSize(cows.all.res.B)       # (all > 1500)

# Checking the Gelman-Ruben-Brooks statistic
gelman.diag(cows.all.res.B)         # (all < 1.01)

# Summary
cows.all.summ <- summary(cows.all.res.B)
```

```
# Summary of the results
cows.all.summ
```

```
##
## Iterations = 6001:56000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 50000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                   Mean       SD  Naive SE Time-series SE
## beta.age      0.096028 0.064278 1.660e-04      2.427e-04
```

```
## beta.height  -0.005837 0.006438 1.662e-05      1.454e-04
## beta.permeab -0.612648 0.888915 2.295e-03      1.705e-02
## beta.rain     0.009160 0.003695 9.540e-06      6.845e-05
## beta.slope   -0.064795 0.073219 1.891e-04      1.344e-03
## beta.temp    -1.173983 0.967664 2.498e-03      2.034e-02
## beta0        -1.682549 0.659822 1.704e-03      1.397e-02
## sigma.farm    0.978678 0.650805 1.680e-03      1.704e-02
##
## 2. Quantiles for each variable:
##
##                   2.5%       25%       50%       75%     97.5%
## beta.age     -0.028844  0.052547  0.095652  0.139275  0.223067
## beta.height  -0.020646 -0.009335 -0.005081 -0.001599  0.004914
## beta.permeab -2.238379 -1.178735 -0.662757 -0.102491  1.317104
## beta.rain     0.002606  0.006800  0.008874  0.011225  0.017266
## beta.slope   -0.207075 -0.107464 -0.065184 -0.022635  0.075981
## beta.temp    -3.296276 -1.733020 -1.125536 -0.562062  0.631512
## beta0        -3.182705 -2.047197 -1.614954 -1.245284 -0.567180
## sigma.farm    0.070037  0.519040  0.879600  1.309656  2.538692
```

```
mean(ilogit(cows.all.output$beta0))
```

```
## [1] 0.1744193
```

Solution: The estimated mean probability of having a parasite for the 'typical' cow (average age and farm covariates) is 0.174. This probability is estimated to increase with age and rain and decrease with height, high permeability, slope and temperature. The standard deviation of the random effect on the farm is estimated to have a non-negligible magnitude (having in mind that the inverse-logit function covers almost all the range for values of the domain between -5 and 5). The multiplicative effect of each variable over the odds ratio can be calculated by the exponential of the associated parameter.

(c) **(7 marks)** Considering the results of the correlation between the variables of the environment of the farm, try some simplifications of the model and select one (you can use DIC as a criterion of selection). Have the parameters of the variables remaining in the model changed substantially? Briefly discuss this.

Solution: One possible approach is, considering the high correlation among rain, temperature and height, to select a model with only one of the three covariates. We take the decision of keeping the rest of the covariates as potential risk factors in the model.

```
# Block model
cows.model.tpsa <-    "model {
  #Likelihood
  for(i in 1:n) {
    parasite[i] ~ dbern(p[i])
    logit(p[i]) <- beta0 + b.farm[farmID[i]]+
                   beta.temp*(temp[i] - mean(temp[])) +
                   beta.permeab* permeab[i] +
```

```
                    beta.slope*(slope[i] - mean(slope[])) +
                    beta.age*(age[i] - mean(age[]))
  }

  for(j in 1:Nfarm){
    b.farm[j] ~ dnorm(0, tau.farm)
  }

  # priors
  beta0 ~ dnorm(0,0.1)

  beta.temp ~ dnorm(0,0.1)
  beta.permeab ~ dnorm(0,0.1)
  beta.slope ~ dnorm(0,0.1)
  beta.age ~ dnorm(0,0.1)

  tau.farm <- pow(sigma.farm, -2)
  sigma.farm ~ dunif(0,10)
}"
```

```
# Initial values
cows.tpsa.inits <- function(){list(beta0=rnorm(1,0,0.1),
                                   beta.temp=rnorm(1,0,0.1),
                                   beta.permeab=rnorm(1,0,0.1),
                                   beta.slope=rnorm(1,0,0.1),
                                   beta.age=rnorm(1,0,0.1),
                                   sigma.farm=runif(1,0,0.1))}


# Running JAGS
cows.tpsa.res.A <- jags.model(file=textConnection(cows.model.tpsa),
                              data=cows.data,  n.chains=3,
                              inits = cows.tpsa.inits, quiet = TRUE)
update(cows.tpsa.res.A, n.iter=5000)
cows.tpsa.res.B <- coda.samples(cows.tpsa.res.A, variable.names=c("beta0",
                                    "beta.temp", "beta.permeab", "beta.slope",
                                    "beta.age", "sigma.farm", "b.farm"),
                                  n.iter=50000)
# Getting DIC
cows.tpsa.DIC <- dic.samples(model=cows.tpsa.res.A, n.iter = 50000,
                                type = "pD")

# Joinning all the chains in one data.frame
cows.tpsa.output <- do.call(rbind.data.frame, cows.tpsa.res.B)

# Checking the chains
mcmcplots::mcmcplot(cows.tpsa.res.B, parms = c("beta0",
                                    "beta.temp", "beta.permeab",
```

```
                                    "beta.slope", "beta.age",
                                    "sigma.farm"))

# Checking the effective sample size
effectiveSize(cows.tpsa.res.B)       # (all > 2100)

# Checking the Gelman-Ruben-Brooks statistic
gelman.diag(cows.tpsa.res.B)         # (all < 1.01)

# Summary
cows.tpsa.summ <- summary(cows.tpsa.res.B)
```

```
# Block model
cows.model.rpsa <-    "model {
  #Likelihood
  for(i in 1:n) {
    parasite[i] ~ dbern(p[i])
    logit(p[i]) <- beta0 + b.farm[farmID[i]]+
                   beta.rain*(rain[i] - mean(rain[])) +
                   beta.permeab* permeab[i] +
                   beta.slope*(slope[i] - mean(slope[])) +
                   beta.age*(age[i] - mean(age[]))
  }

  for(j in 1:Nfarm){
    b.farm[j] ~ dnorm(0, tau.farm)
  }

  # priors
  beta0 ~ dnorm(0,0.1)

  beta.rain ~ dnorm(0,0.1)
  beta.permeab ~ dnorm(0,0.1)
  beta.slope ~ dnorm(0,0.1)
  beta.age ~ dnorm(0,0.1)

  tau.farm <- pow(sigma.farm, -2)
  sigma.farm ~ dunif(0,10)
}"
```

```
# Initial values
cows.rpsa.inits <- function(){list(beta0=rnorm(1,0,0.1),
                                   beta.rain=rnorm(1,0,0.1),
                                   beta.permeab=rnorm(1,0,0.1),
                                   beta.slope=rnorm(1,0,0.1),
                                   beta.age=rnorm(1,0,0.1),
                                   sigma.farm=runif(1,0,0.1))}
```

```r
# Running JAGS
cows.rpsa.res.A <- jags.model(file=textConnection(cows.model.rpsa),
                              data=cows.data,  n.chains=3,
                              inits = cows.rpsa.inits, quiet = TRUE)
update(cows.rpsa.res.A, n.iter=5000)
cows.rpsa.res.B <- coda.samples(cows.rpsa.res.A, variable.names=c("beta0",
                                    "beta.rain", "beta.permeab", "beta.slope",
                                    "beta.age", "sigma.farm", "b.farm"),
                                  n.iter=50000)
# Getting DIC
cows.rpsa.DIC <- dic.samples(model=cows.rpsa.res.A, n.iter = 50000,
                                  type = "pD")

# Joinning all the chains in one data.frame
cows.rpsa.output <- do.call(rbind.data.frame, cows.rpsa.res.B)

# Checking the chains
mcmcplots::mcmcplot(cows.rpsa.res.B, parms = c("beta0",
                                    "beta.rain", "beta.permeab",
                                    "beta.slope", "beta.age",
                                    "sigma.farm"))

# Checking the effective sample size
effectiveSize(cows.rpsa.res.B)       # (all > 1000)

# Checking the Gelman-Ruben-Brooks statistic
gelman.diag(cows.rpsa.res.B)         # (all < 1.01)

# Summary
cows.rpsa.summ <- summary(cows.rpsa.res.B)
```

```r
# Block model
cows.model.hpsa <-    "model {
  #Likelihood
  for(i in 1:n) {
    parasite[i] ~ dbern(p[i])
    logit(p[i]) <- beta0 + b.farm[farmID[i]]+
                   beta.height*(height[i] - mean(height[])) +
                   beta.permeab* permeab[i] +
                   beta.slope*(slope[i] - mean(slope[])) +
                   beta.age*(age[i] - mean(age[]))
  }

  for(j in 1:Nfarm){
    b.farm[j] ~ dnorm(0, tau.farm)
  }

  # priors
```

```
  beta0 ~ dnorm(0,0.1)

  beta.height ~ dnorm(0,0.1)
  beta.permeab ~ dnorm(0,0.1)
  beta.slope ~ dnorm(0,0.1)
  beta.age ~ dnorm(0,0.1)

  tau.farm <- pow(sigma.farm, -2)
  sigma.farm ~ dunif(0,10)
}"
```

```
# Initial values
cows.hpsa.inits <- function(){list(beta0=rnorm(1,0,0.1),
                                   beta.height=rnorm(1,0,0.1),
                                   beta.permeab=rnorm(1,0,0.1),
                                   beta.slope=rnorm(1,0,0.1),
                                   beta.age=rnorm(1,0,0.1),
                                   sigma.farm=runif(1,0,0.1))}


# Running JAGS
cows.hpsa.res.A <- jags.model(file=textConnection(cows.model.hpsa),
                              data=cows.data,  n.chains=3,
                              inits = cows.hpsa.inits, quiet = TRUE)
update(cows.hpsa.res.A, n.iter=5000)
cows.hpsa.res.B <- coda.samples(cows.hpsa.res.A, variable.names=c("beta0",
                                    "beta.height", "beta.permeab", "beta.slope",
                                    "beta.age", "sigma.farm", "b.farm"),
                                n.iter=50000)
# Getting DIC
cows.hpsa.DIC <- dic.samples(model=cows.hpsa.res.A, n.iter = 50000,
                                type = "pD")

# Joinning all the chains in one data.frame
cows.hpsa.output <- do.call(rbind.data.frame, cows.hpsa.res.B)

# Checking the chains
mcmcplots::mcmcplot(cows.hpsa.res.B, parms = c("beta0",
                                    "beta.height", "beta.permeab",
                                    "beta.slope", "beta.age",
                                    "sigma.farm"))

# Checking the effective sample size
effectiveSize(cows.hpsa.res.B)       # (all > 2500)

# Checking the Gelman-Ruben-Brooks statistic
gelman.diag(cows.hpsa.res.B)         # (all < 1.01)
```

```
# Summary
cows.hpsa.summ <- summary(cows.hpsa.res.B)
```

```
# Comparison of DIC scores
cows.all.DIC

## Mean deviance:  230.2
## penalty 12.17
## Penalized deviance: 242.4

cows.tpsa.DIC

## Mean deviance:  232.6
## penalty 13.66
## Penalized deviance: 246.2

cows.rpsa.DIC

## Mean deviance:  232.2
## penalty 9.851
## Penalized deviance: 242.1

cows.hpsa.DIC

## Mean deviance:  230.1
## penalty 14.02
## Penalized deviance: 244.1
```

Solution: DIC does not offer a strongly superior model, though the best DIC scores are observed for the complete model and the model without temperature or height. Looking for external data (https://agrilifeextension.tamu.edu/library/ranching/common-cattle-parasite/) we find that rain is involved in the cicle of reproduction of hairworm. Therefore, for simplicity, we will keep on working with the model that considers rain and discards temperature and height as explanatory variables. This is not necessarily the best choice. For example, we could decide to be conservative and keep all the variables. In general, consultation with experts is recommended when uncertain.

(d) **(13 marks)** Based on the model you have selected, estimate the posterior expected value and 95% credible interval of the proportion of cows in farm 1 that have parasites (assume age equal to the mean age in the study). A farm is declared in epidemic state if the proportion of cows with the parasite in that farm is larger than 20%. What is the probability of farm 1 to be in the epidemic state? Perform the same analysis for farm 6.

```
mrain <- mean(cows$rain)
mslope <- mean(cows$slope)

# Posterior proportion of cows in farms 1 and 6 with parasites
p1.rpsa <- ilogit( cows.rpsa.output$beta0 +
                     cows.rpsa.output$`b.farm[1]` +
                     cows.rpsa.output$beta.rain*(farms$rain[1]-mrain) +
```

```
                    cows.rpsa.output$beta.permeab*(farms$permeab[1]-1) +
                    cows.rpsa.output$beta.slope*(farms$slope[1]-mslope) )

p6.rpsa <- ilogit( cows.rpsa.output$beta0 +
                    cows.rpsa.output$`b.farm[6]` +
                    cows.rpsa.output$beta.rain*(farms$rain[6]-mrain) +
                    cows.rpsa.output$beta.permeab*(farms$permeab[6]-1) +
                    cows.rpsa.output$beta.slope*(farms$slope[6]-mslope) )
```

```
# These posterior distributions can also be obtained including the following
#   lines in the model (note that cow 142 belongs to farm 6):
"
...
logit(p1.new) <- beta0 + b.farm[1]+
                  beta.rain*(rain[1] - mean(rain[])) +
                  beta.permeab* permeab[1] +
                  beta.slope*(slope[1] - mean(slope[]))
logit(p6.new) <- beta0 + b.farm[6]+
                  beta.rain*(rain[142] - mean(rain[])) +
                  beta.permeab* permeab[142] +
                  beta.slope*(slope[142] - mean(slope[]))
...
"
```

```
# Posterior expected mean and 95% CI
mean(p1.rpsa)

## [1] 0.02492309

quantile(p1.rpsa, prob=c(0.025,0.975))

##        2.5%       97.5%
## 0.002792211 0.075183689

mean(p6.rpsa)

## [1] 0.5861282

quantile(p6.rpsa, prob=c(0.025,0.975))

##      2.5%     97.5%
## 0.2386011 0.9558912

# Posterior probability of farms 1 and 6 to be in epidemic state
mean(p1.rpsa>0.2)

## [1] 6e-05

mean(p6.rpsa>0.2)

## [1] 0.98976
```

(e) **(9 marks)** Modify the last model substituting the hierarchical random effect of the farm for a fixed effect of the farm. Estimate the posterior expected value and 95% credible interval of the proportion of cows in farm 1 and in farm 6 that have parasites, as well as the probability for each farm to be in the epidemic state. Discuss and compare the results for the fixed effects model and the random effects hierarchical model.

```
# Block model
cows.model.fixed <-    "model {
  #Likelihood
  for(i in 1:n) {
    parasite[i] ~ dbern(p[i])
    logit(p[i]) <- b.farm[farmID[i]]+       # We need to remove beta0
                   beta.rain*(rain[i] - mean(rain[])) +
                   beta.permeab* permeab[i] +
                   beta.slope*(slope[i] - mean(slope[])) +
                   beta.age*(age[i] - mean(age[]))
  }

  for(j in 1:Nfarm){
    b.farm[j] ~ dnorm(0, tau.farm)
  }

  # priors
  beta.rain ~ dnorm(0,0.1)
  beta.permeab ~ dnorm(0,0.1)
  beta.slope ~ dnorm(0,0.1)
  beta.age ~ dnorm(0,0.1)

  # CHANGES to transform the random effect into a fixed effect
  tau.farm <- 0.01
}"
```

```
# Initial values
cows.fixed.inits <- function(){list(beta.rain=rnorm(1,0,0.1),
                                    beta.permeab=rnorm(1,0,0.1),
                                    beta.slope=rnorm(1,0,0.1),
                                    beta.age=rnorm(1,0,0.1))}


# Running JAGS
cows.fixed.res.A <- jags.model(file=textConnection(cows.model.fixed),
                               data=cows.data,  n.chains=3,
                               inits = cows.fixed.inits, quiet = TRUE)
update(cows.fixed.res.A, n.iter=5000)
cows.fixed.res.B <- coda.samples(cows.fixed.res.A, variable.names=c(
                                    "beta.rain", "beta.permeab", "beta.slope",
                                    "beta.age", "b.farm"),
                                 n.iter=200000)
# Getting DIC
```

```r
cows.fixed.DIC <- dic.samples(model=cows.fixed.res.A, n.iter = 200000,
                              type = "pD")

# Joinning all the chains in one data.frame
cows.fixed.output <- do.call(rbind.data.frame, cows.fixed.res.B)

# Checking the chains
mcmcplots::mcmcplot(cows.fixed.res.B, parms = c("beta0",
                                  "beta.rain", "beta.permeab",
                                  "beta.slope", "beta.age"))

# Checking the effective sample size
effectiveSize(cows.fixed.res.B)       # (all > 700)

# Checking the Gelman-Ruben-Brooks statistic
gelman.diag(cows.fixed.res.B)         # (all < 1.01)

# Summary
cows.fixed.summ <- summary(cows.fixed.res.B)
```

```r
mrain <- mean(cows$rain)
mslope <- mean(cows$slope)

# Posterior proportion of cows in farms 1 and 6 with parasites
p1.fixed <- ilogit( cows.fixed.output$`b.farm[1]` +
                    cows.fixed.output$beta.rain*(farms$rain[1]-mrain) +
                    cows.fixed.output$beta.permeab*(farms$permeab[1]-1) +
                    cows.fixed.output$beta.slope*(farms$slope[1]-mslope) )

p6.fixed <- ilogit( cows.fixed.output$`b.farm[6]` +
                    cows.fixed.output$beta.rain*(farms$rain[6]-mrain) +
                    cows.fixed.output$beta.permeab*(farms$permeab[6]-1) +
                    cows.fixed.output$beta.slope*(farms$slope[6]-mslope) )

# Posterior expected mean and 95% CI
mean(p1.fixed)
```

```
## [1] 0.02840511
```

```r
quantile(p1.fixed, prob=c(0.025,0.975))
```

```
##         2.5%        97.5%
## 0.0006910121 0.1039682285
```

```r
mean(p6.fixed)
```

```
## [1] 0.9842384
```

```r
quantile(p6.fixed, prob=c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.8271983 1.0000000

# Posterior probability of farms 1 and 6 to be in epidemic state
mean(p1.fixed>0.2)

## [1] 0.000595

mean(p6.fixed>0.2)

## [1] 0.999965
```

Solution: We can observe how the posterior mean probability of parasite for both models is similar for farm 1 (slightly larger for teh fixed model) but the CI for the hierarchical model is narrower. The difference of estimates for farm 6 is greater, as is to be expected, as there are much fewer observations for that farm. The fixed model estimates a probability near to 1 with a quite narrow CI that suggests the probability is in any case larger than 80%. The hierarchical model, though, has a much lower mean posterior estimate, around 60%, with a much wider CI that suggests that almost any value over 24% is plausible. Usually, hierarchical models offer narrower posterior CI because the information of other groups is shared, but in this case, the hierarchical model offers a wider CI. This is because we are considering an inverse-logit transformation of the estimated regressor, which is a non-linear transformation and therefore the order of size of intervals is not necessarily conserved. What we can observe is that the CI estimated by the hierarchical model for farm 6 is pulled towards the common mean probability of having a parasite, even though all the observed cows in farm 6 have a parasite, because of the sharing of information among farms in the hierarchical model.

3. *Proposing a new model* **(12 marks)** Propose an extension, modification or an alternate model for either the gulls data or the cows data. Do the Bayesian inference for the proposed model and discuss the results comparing them to the models fitted in question 1 or 2.

   Some possible ideas are: using state space models, changing the likelihood distribution that models the data, changing the link function, considering interactions between variables or considering another variable as the response variable. Only one of those changes –or any other change in the same line– is enough for full marks (as long as it makes sense and the inference is adequately performed and discussed).