

Project SDS II

Surgical: Institutional ranking

Simone Chieppa1846140

21/07/2022

In this project I decided to analyze the analysis done from the authors in ‘Goudie, R. J. B., Turner, R. M., De Angelis, D., Thomas, A. (2020) MultiBUGS: A parallel implementation of the BUGS modelling framework for faster Bayesian inference. Journal of Statistical Software, 95(7).’ that considers mortality rates in 12 hospitals performing cardiac surgery in babies.

I will first replicate in a more detailed way the analysis done by the authors using the models implemented in the paper (a logit model and a beta model) and I will compare them with the results of a frequentist analysis. Then I will also implement probit model that is not present in the paper and I will analyze the results and compare this model with the others. But first let’s observe the data.

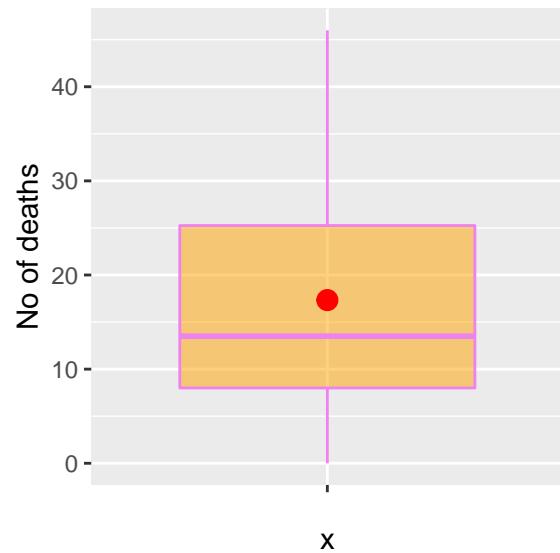
Hospital	No of ops	No of deaths
A	47	0
B	148	18
C	119	8
D	810	46
E	211	8
F	196	13
G	148	9
H	215	31
I	207	14
J	97	8
K	256	29
L	360	24

No of deaths

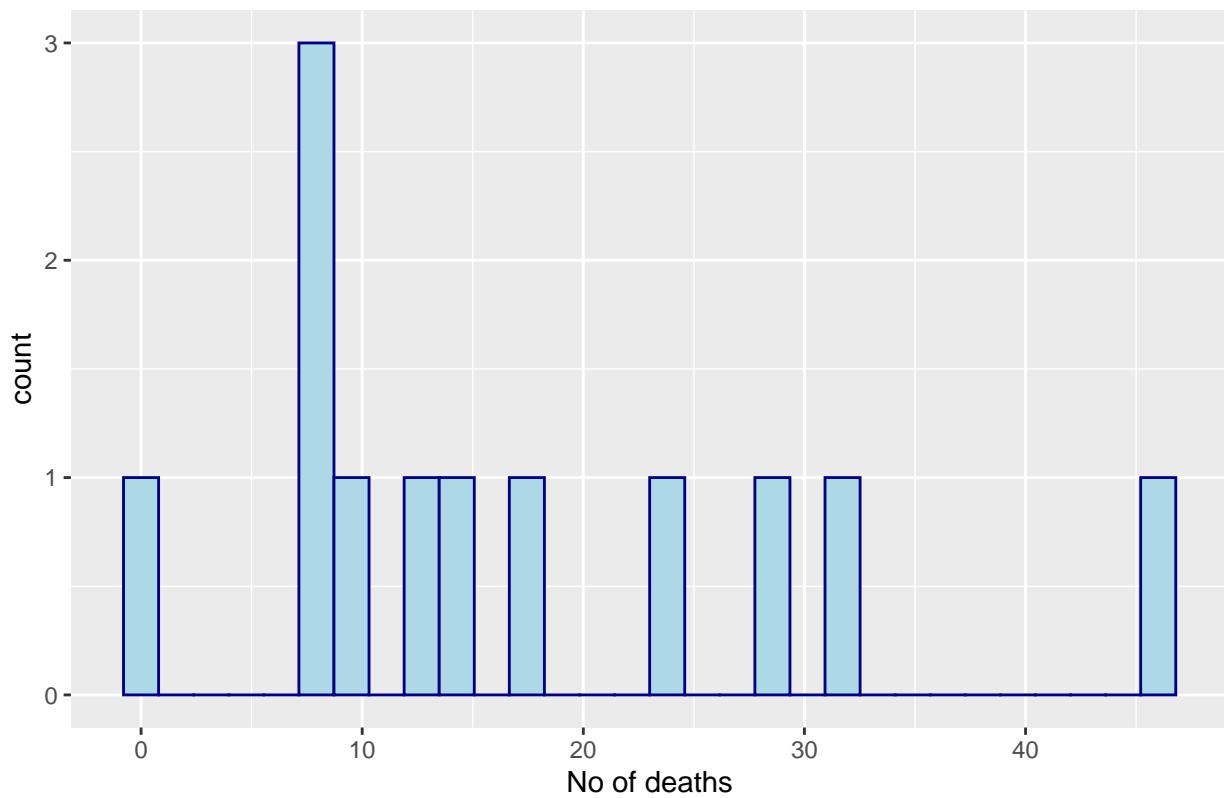
```
summary(data$`No of deaths`)
```

```
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   0.00   8.00  13.50  17.33  25.25  46.00
```

We can notice that the median and the mean of the number of deaths are really different, that means that the distribution is not symmetric. Let’s have a better visualization of these informations with a boxplot where the red point represent the mean.



Distribution of the Nuber of deaths



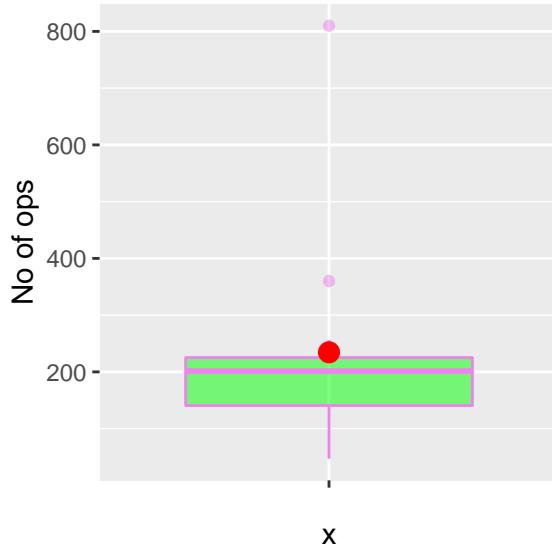
Also from the histogram we can observe that the distribution is not symmetric and that most hospitals had a number of death between 8 and 20. We can also notice that there are two outliers (0 death and 46 death) that are far away from the other values.

No of operations

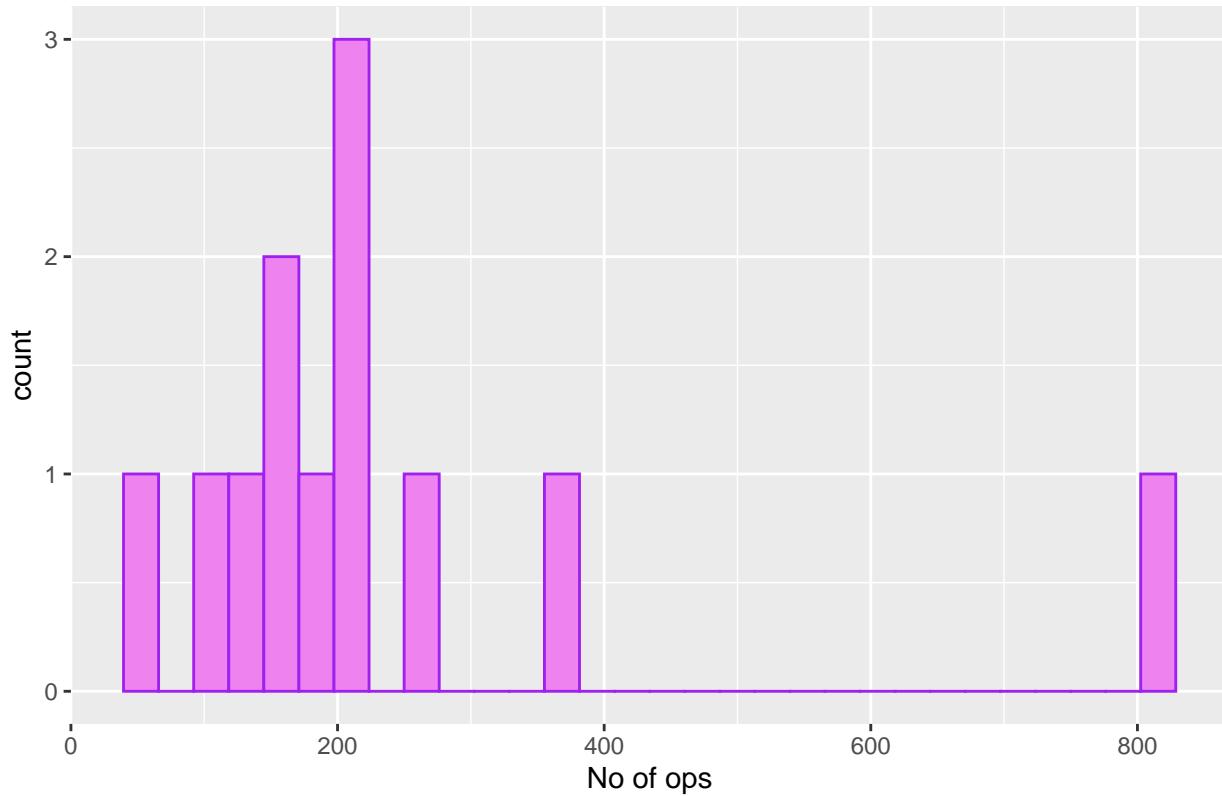
```
summary(data$`No of ops`)
```

```
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   47.0 140.8 201.5 234.5 225.2 810.0
```

It is interesting to notice that also in the number of operations the mean and the median are really different, in this case the mean is even bigger than the 3rd quartile as we can see also in the boxplot below



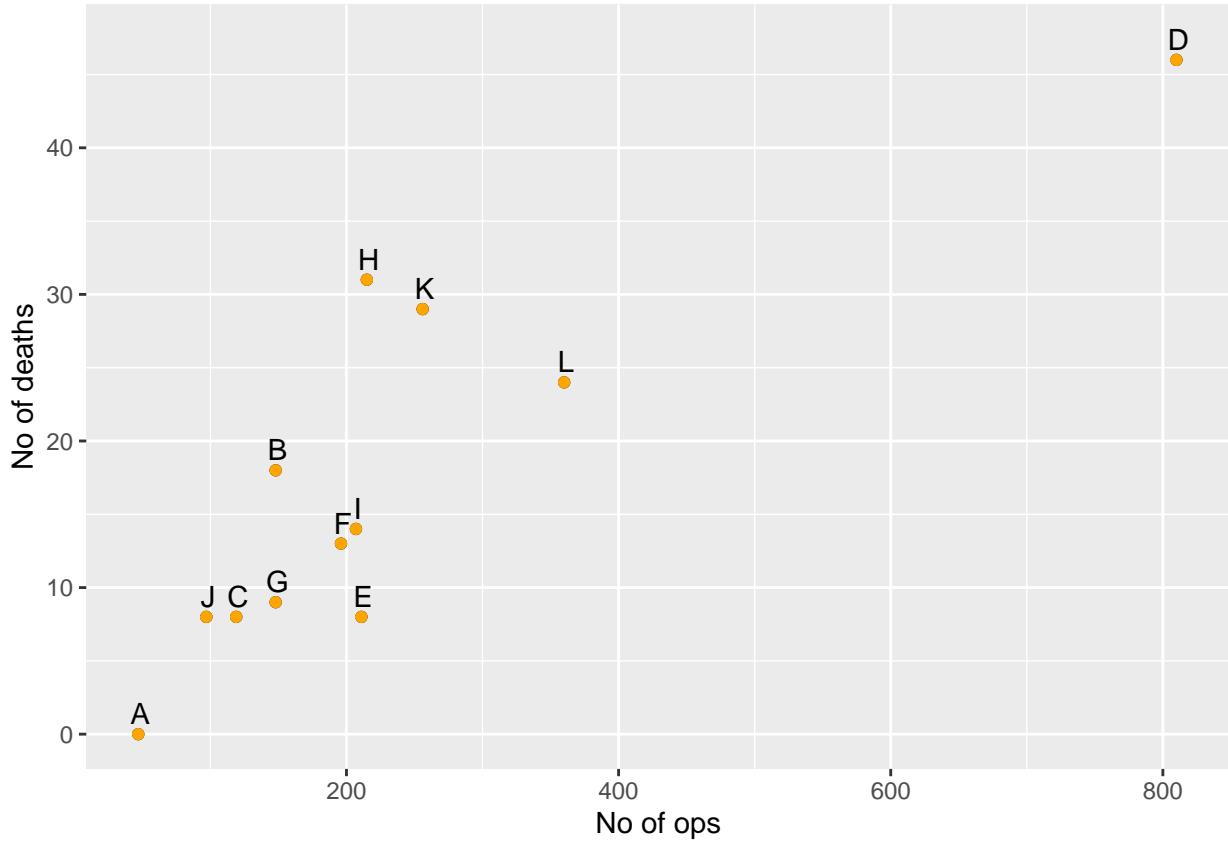
Distribution of the Nuber of operations



We can notice that there is an outlier that is way bigger than the other values, this is the reson why the mean is so large, in fact the mean is strongly affected by the outliers. Most of the values are concentrated around 100 and 200.

No of deaths vs No operations

Now lets see the data together:



We can observe that the hospital with more operation (hospital D with more than 800 operations) is also the one with more deaths (46) and the hospital with less operations (hospital A with less than 50 operations) is the one with less deaths (0).

Beta model

We first start with the most simple model assuming as the authors does in the paper that the true failure probabilities are independent for each hospital. We assume that the number of deaths r_i for hospital i are modelled as a binary response variable with failure probability p_i :

$$r_i \sim \text{Binomial}(p_i, n_i)$$

Since we assume that the failure probabilities are independent we can model p_i as:

$$p_i \sim \text{Beta}(1.0, 1.0)$$

That is equivalent to use an uniform distribution $U([0, 1])$. Here we can have a look at the very simple graphical model:

As the authors do I will run two different chains from two different lists of initial values to have more accurate results and I will run 15000 iterations. I decided to n.burnin parameter equal to 1000 to remove the initial noise.

```
parameters <- c('p')

inits <- list(list(p = c(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)), 
            list(p = c(0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)))
```

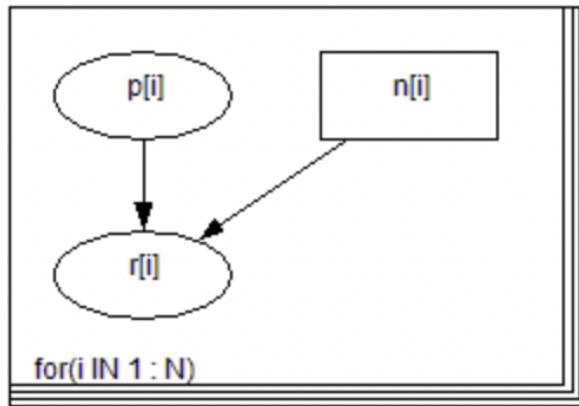


Figure 1: graphical model

```

initial.values <- list(inits)

surgical_beta <- jags(data=data_list,inits=inits,parameters.to.save=parameters,model.file="Model1.txt",
                       n.chains=2,n.thin = 1,n.burnin=1000,n.iter=15000)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 12
##   Unobserved stochastic nodes: 12
##   Total graph size: 38
##
## Initializing model

print(surgical_beta)

## Inference for Bugs model at "Model1.txt", fit using jags,
## 2 chains, each with 15000 iterations (first 1000 discarded)
## n.sims = 28000 iterations saved
##          mu.vect  sd.vect    2.5%     25%     50%     75%   97.5%   Rhat n.eff
## p[1]      0.020  0.020  0.001  0.006  0.014  0.028  0.073 1.001  9800
## p[2]      0.127  0.027  0.078  0.108  0.125  0.144  0.185 1.001 28000
## p[3]      0.074  0.024  0.035  0.057  0.072  0.089  0.127 1.001 28000
## p[4]      0.058  0.008  0.043  0.052  0.057  0.063  0.075 1.001 28000
## p[5]      0.042  0.014  0.019  0.033  0.041  0.051  0.073 1.001 25000
## p[6]      0.071  0.018  0.039  0.058  0.069  0.082  0.110 1.001 28000
## p[7]      0.067  0.020  0.033  0.052  0.065  0.079  0.111 1.001  5100
## p[8]      0.148  0.024  0.104  0.131  0.147  0.163  0.198 1.001 28000
## p[9]      0.072  0.018  0.041  0.059  0.071  0.083  0.111 1.001 28000
## p[10]     0.091  0.029  0.043  0.070  0.089  0.109  0.155 1.001 28000

```

```

## p[11]      0.116  0.020  0.080  0.103  0.115  0.129  0.159 1.001 28000
## p[12]      0.069  0.013  0.045  0.060  0.068  0.078  0.097 1.001 28000
## deviance   62.722  5.066 54.780 59.043 62.079 65.682 74.444 1.001 28000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 12.8 and DIC = 75.6
.....

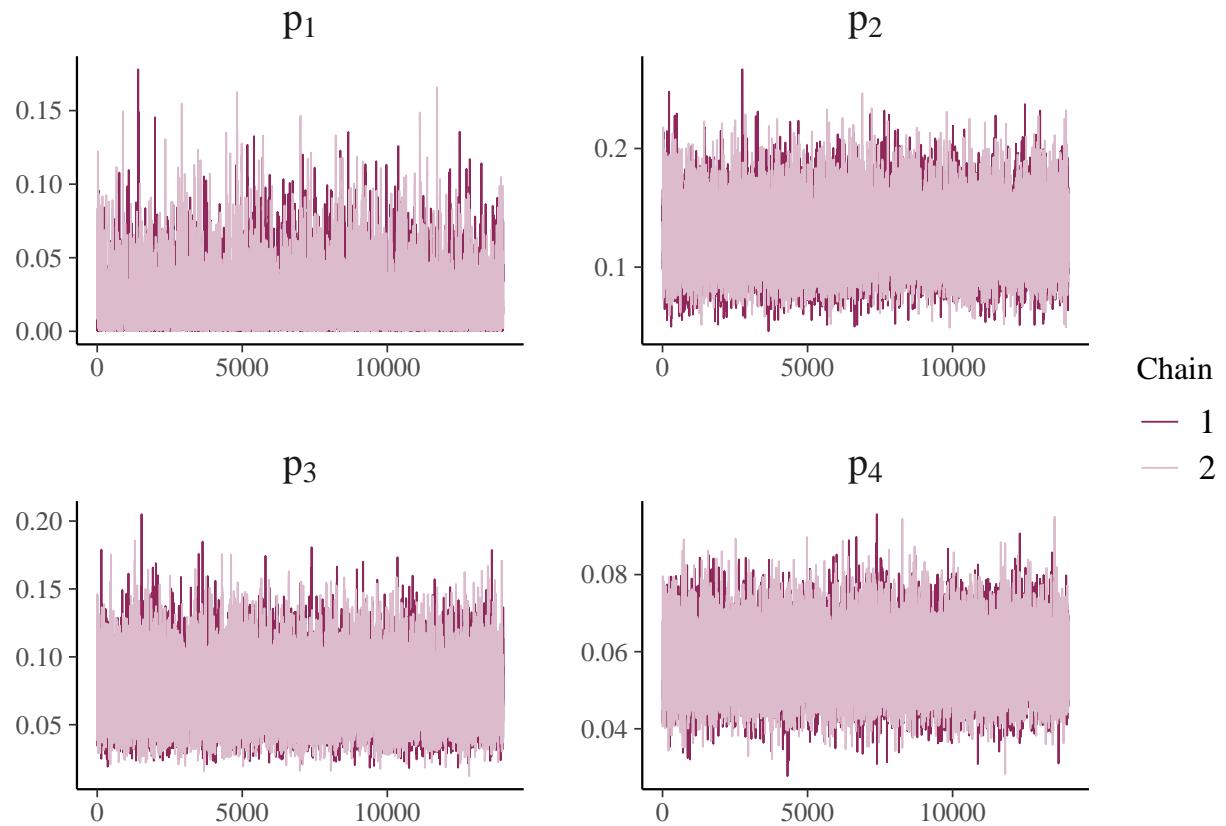
```

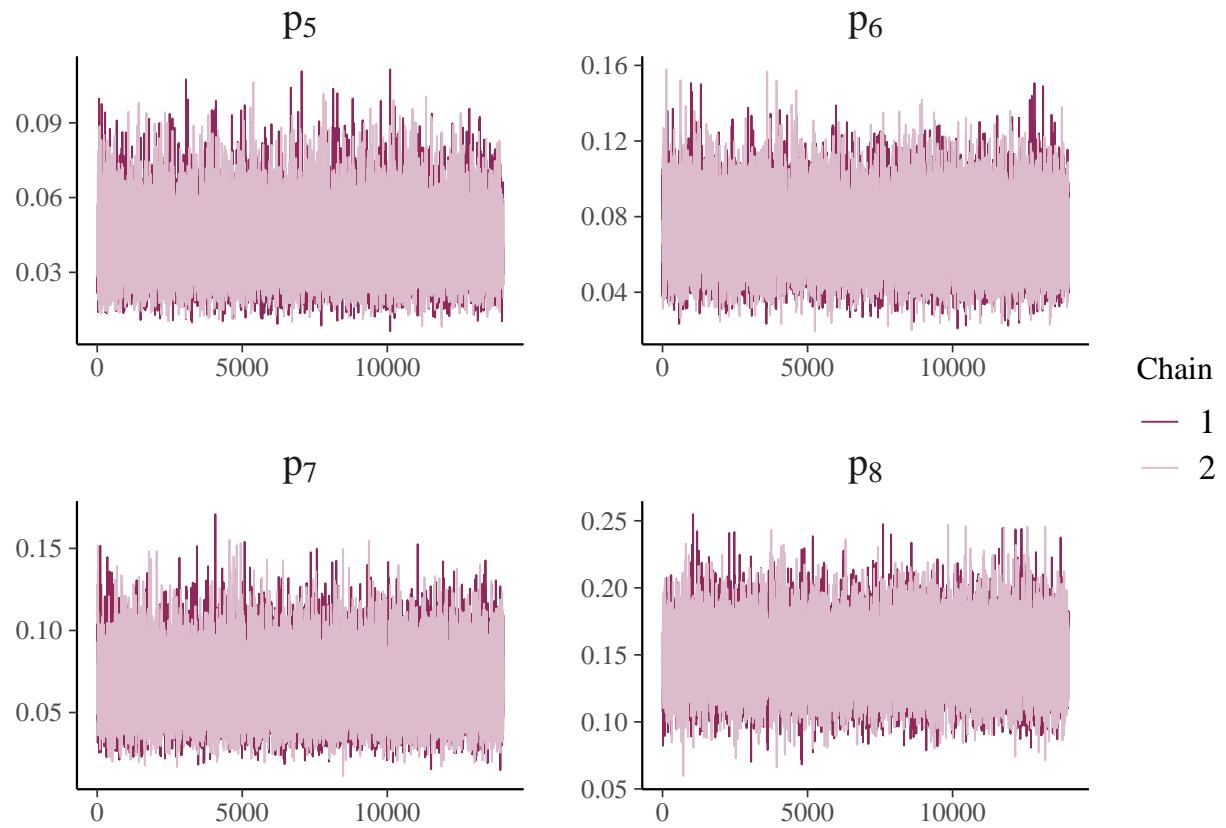
In this table we can see the point estimates values for p_i and we can also build the credible intervals using the quantiles. It is interesting to notice that all the credible intervals for the p_i do not contain the 0, that means that the variables are statistically significant. We can also notice from the $R_{\text{hat}} = 1$ that the series converges.

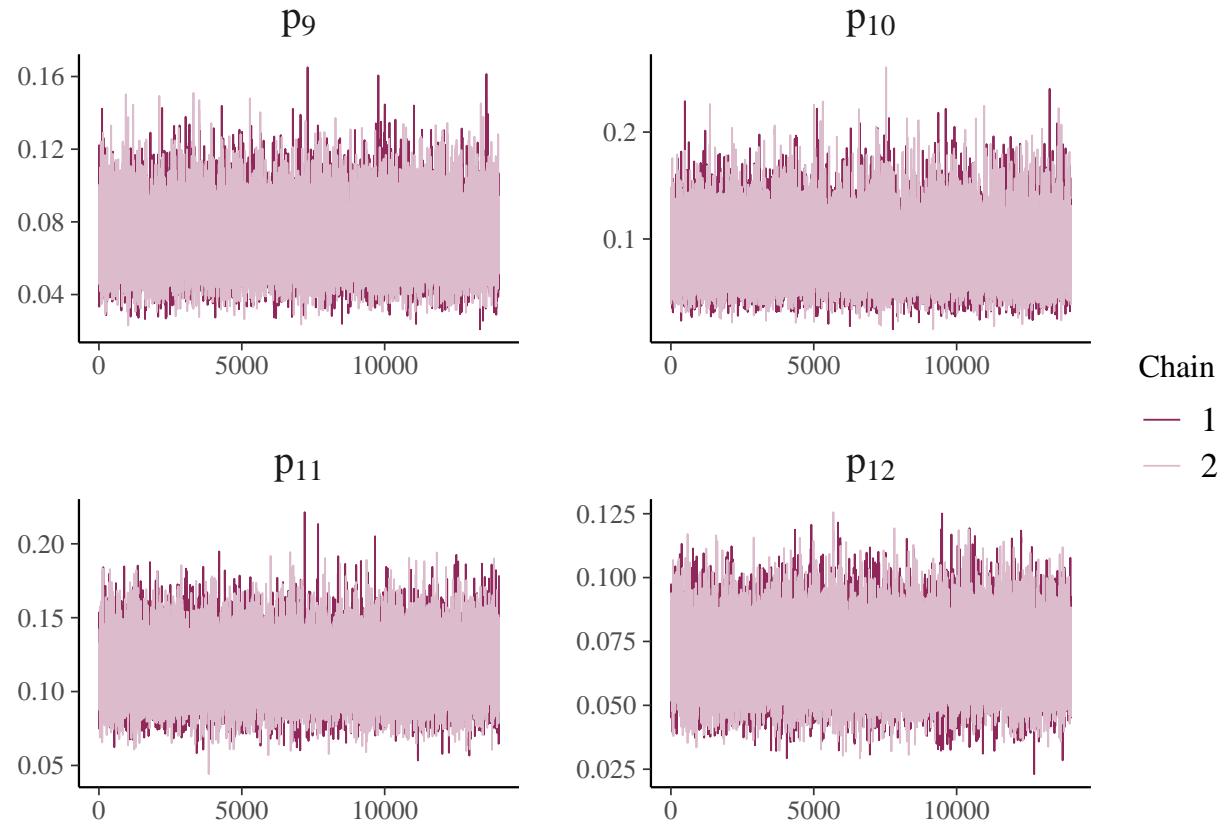
The hospital with the lowest probability of death is the first one (hospital A) and the hospital with the highest probability of death is the 8th (hospital H). We can also notice that the credible interval of $p[1]$ is very big and that is not a good sign Let's compare the results with the ones in the paper:

	mean	median	sd	MC_error	val2.5pc	val97.5pc	start	sample	ESS
p[1]	0.02059	0.01439	0.02002	1.338E-4	5.563E-4	0.07399	1001	20000	22367
p[2]	0.1265	0.1247	0.02712	1.805E-4	0.07787	0.1837	1001	20000	22576
p[3]	0.07435	0.07208	0.0237	1.648E-4	0.03474	0.1274	1001	20000	20679
p[4]	0.05782	0.0575	0.008108	5.243E-5	0.04304	0.07463	1001	20000	23916
p[5]	0.04224	0.04077	0.01382	9.507E-5	0.01957	0.07304	1001	20000	21129
p[6]	0.07053	0.06906	0.01813	1.309E-4	0.03945	0.1104	1001	20000	19192
p[7]	0.06663	0.06445	0.02024	1.427E-4	0.03287	0.1114	1001	20000	20120
p[8]	0.1478	0.1467	0.02426	1.638E-4	0.1034	0.1988	1001	20000	21929
p[9]	0.07173	0.07024	0.01781	1.276E-4	0.04118	0.1105	1001	20000	19504
p[10]	0.09062	0.08795	0.02854	1.989E-4	0.04316	0.1543	1001	20000	20592
p[11]	0.1162	0.1153	0.02004	1.473E-4	0.07966	0.1581	1001	20000	18511
p[12]	0.06897	0.06828	0.01324	9.687E-5	0.04535	0.09723	1001	20000	18680

The results are really similar and that is a good sign! Now let's see the trace-plots of the simulations of each parameter

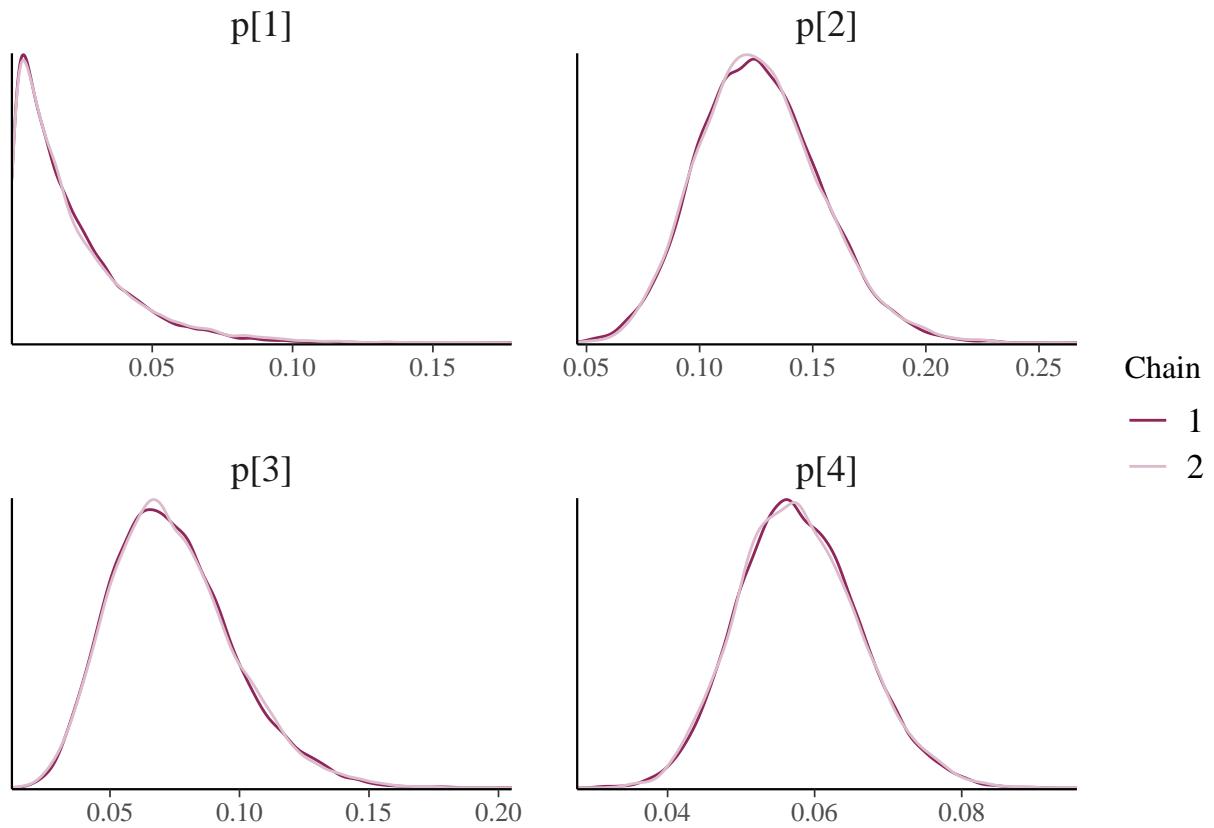


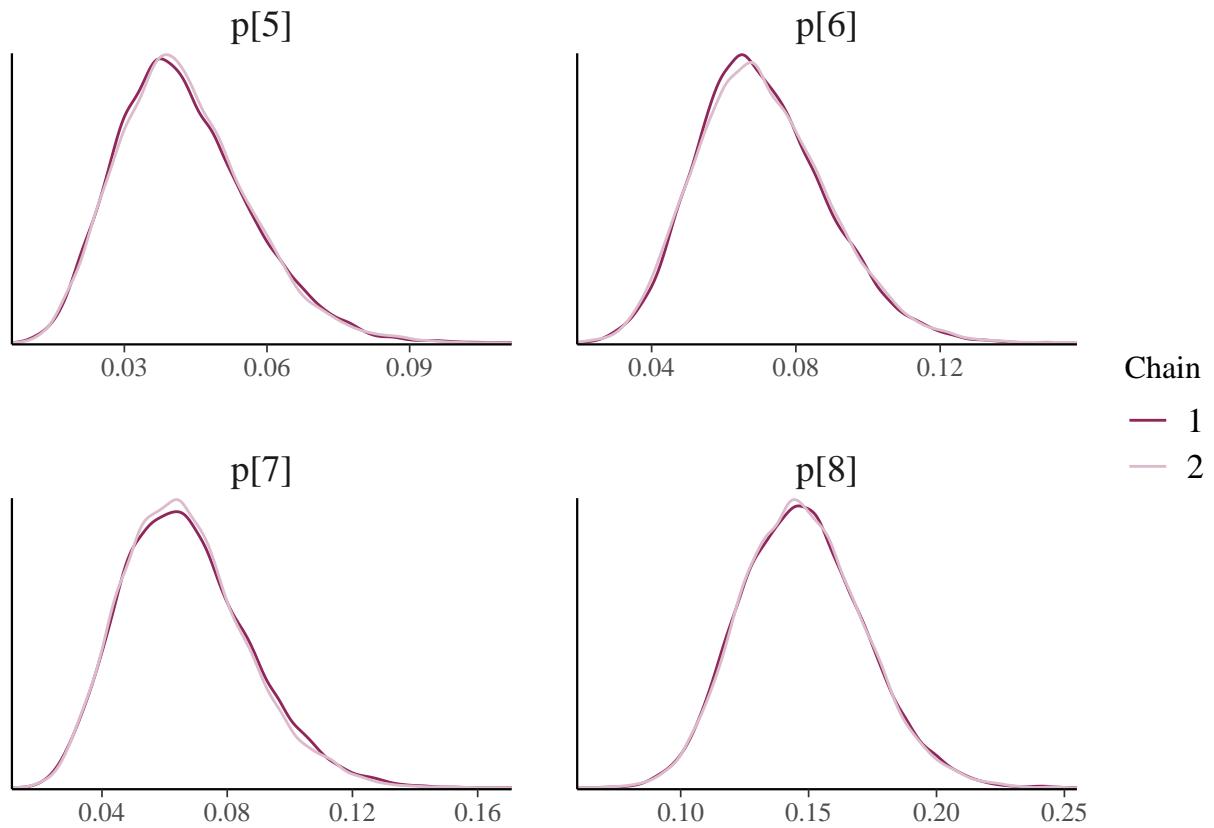


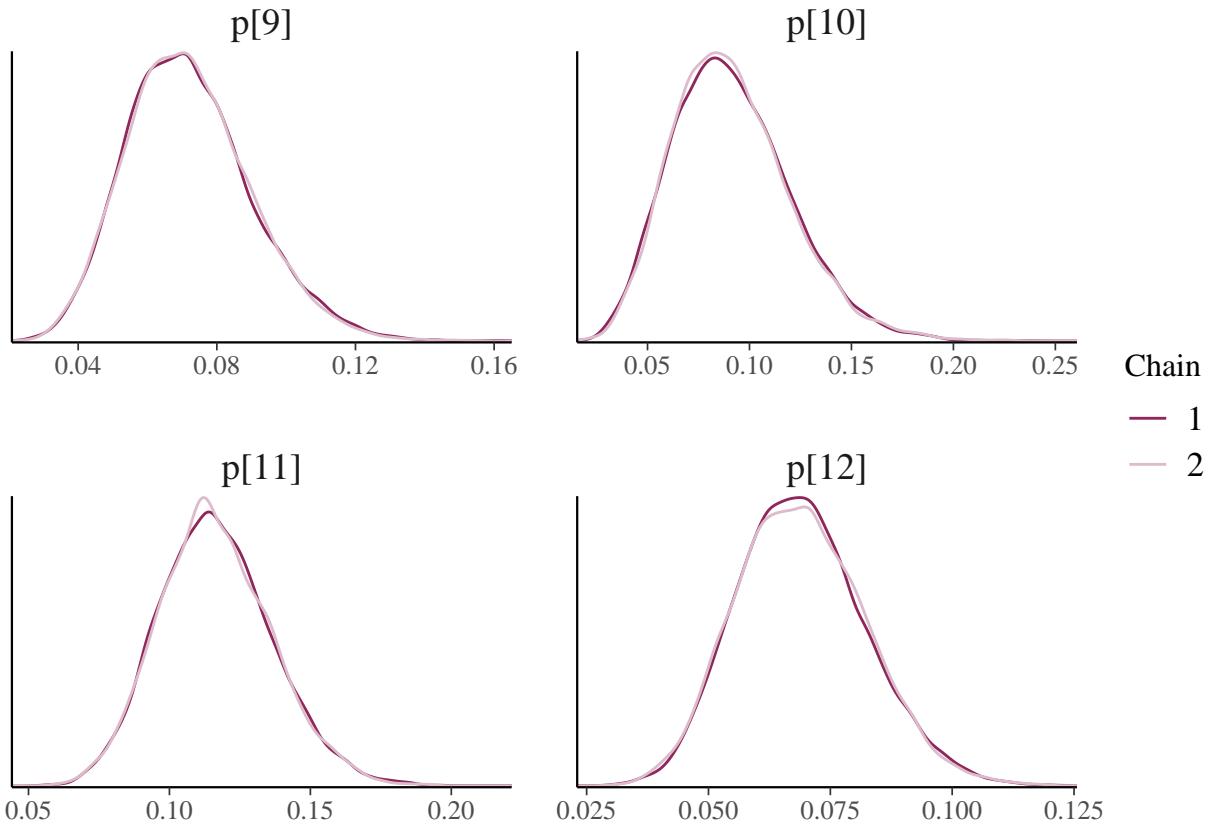


The traceplot that shows the sampled values per chain and node throughout iterations. Here we can see the two chains that both have a random behaviour around the same intervals of values.

Now let's see the parameters's density

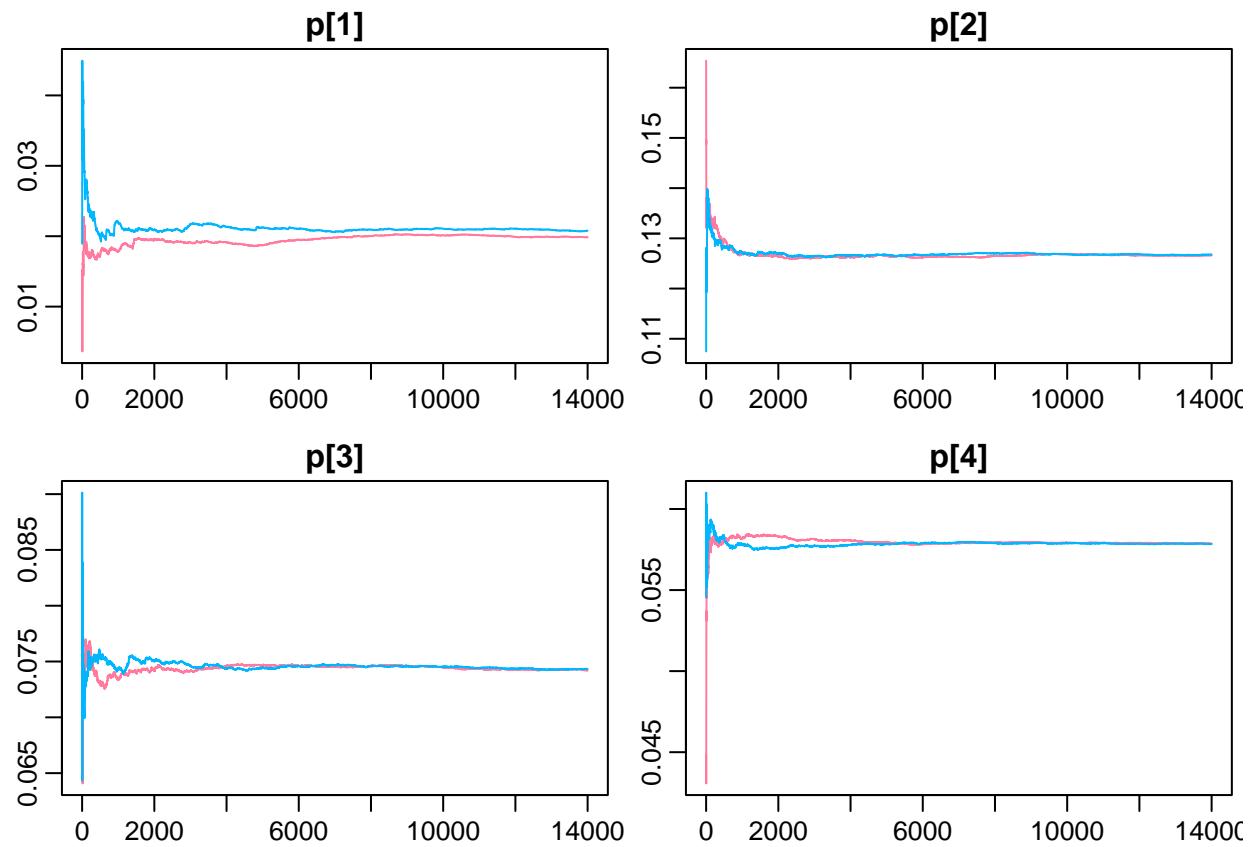


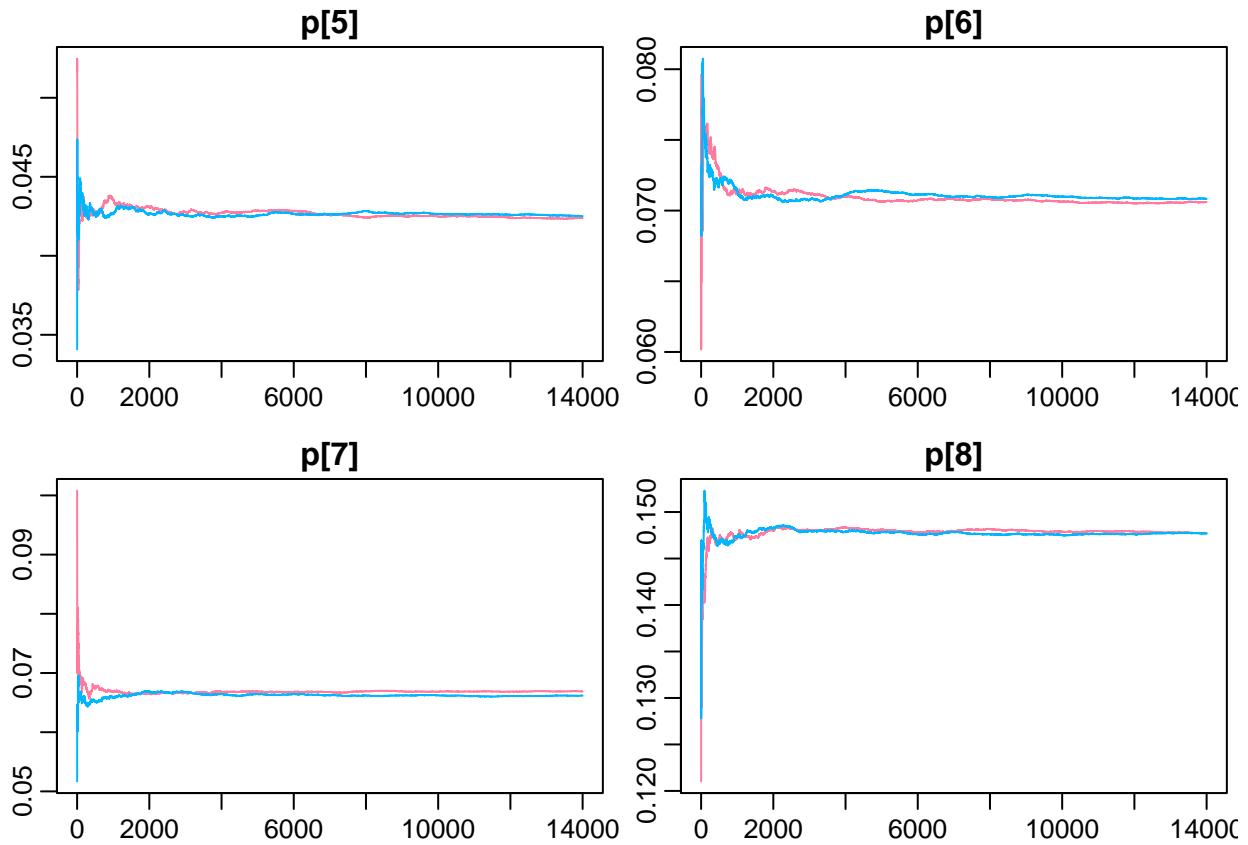


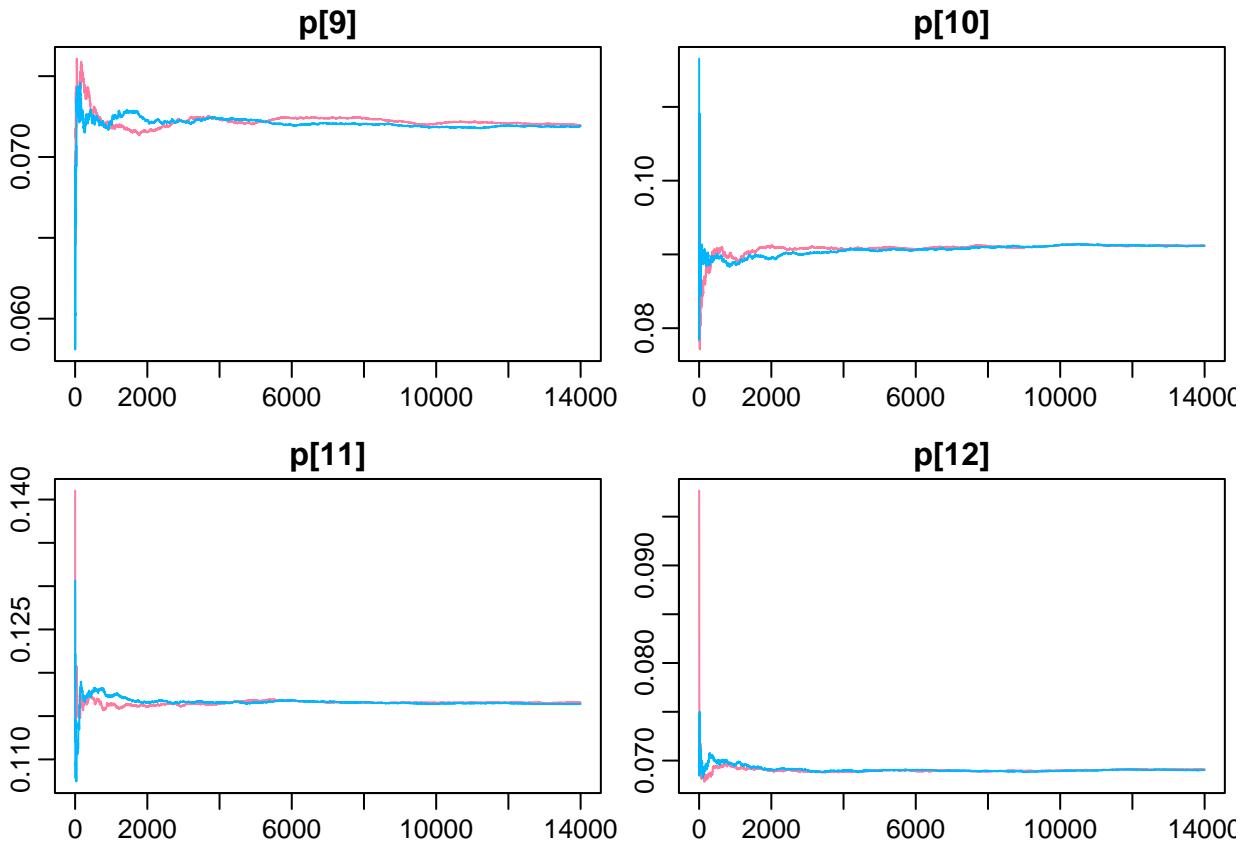


From these plots we can notice that the two chains are almost overlapped and that is a good sign, it means that we reached the stationarity.

To see when the parameters reached the stationarity we can see the plots of the running means that represents the empirical means that reaches their stationarity



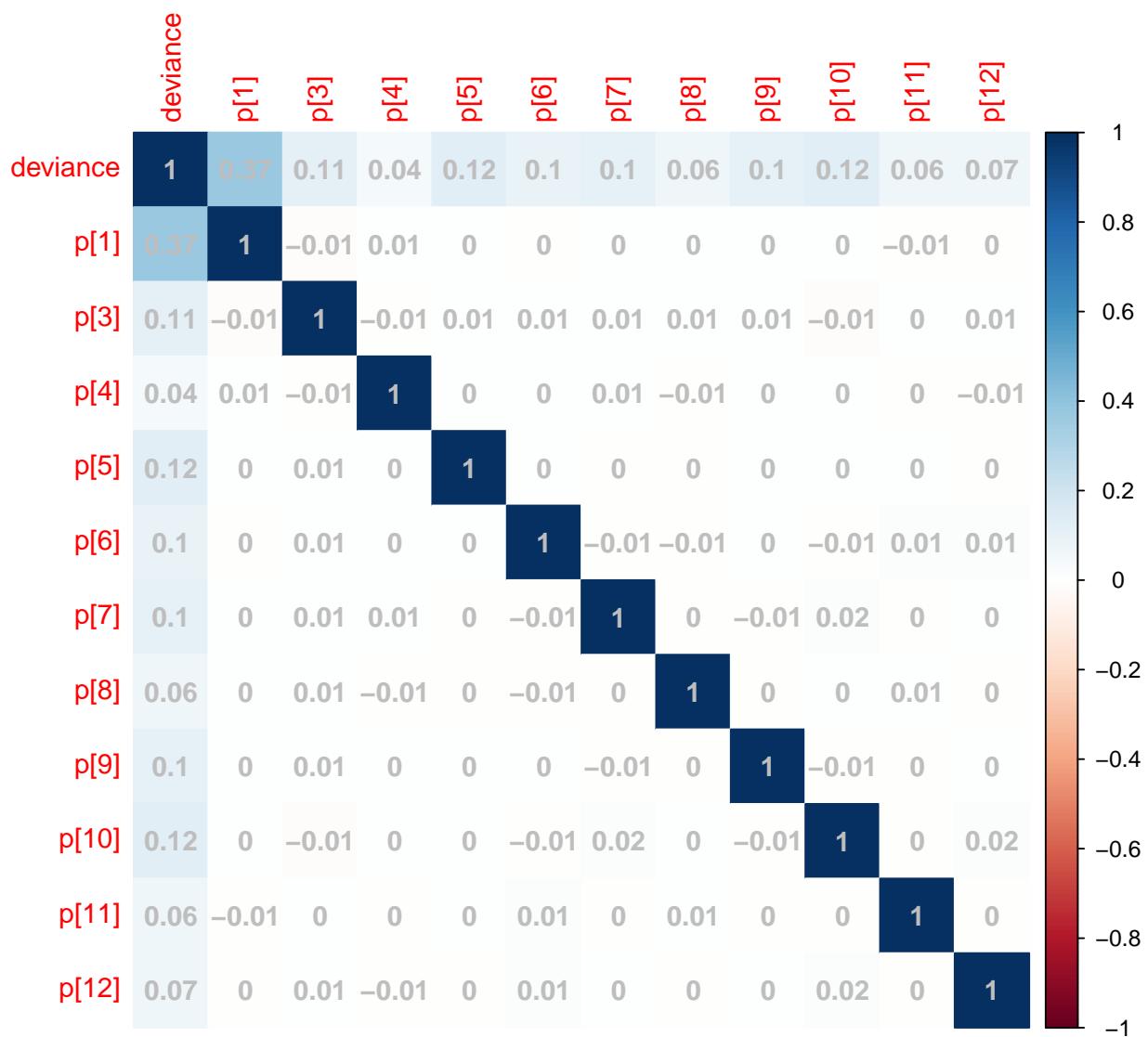




We can notice that the last parameter that reaches the stationarity is $p[1]$.

Now we can have a look at the correlation between parameters

```
corrplot::corrplot(cor(surgical_beta$BUGSoutput$sims.matrix)[-3,-3],
                   method = "color", addCoef.col="grey")
```



We can observe that the correlation between all the parameters is really low, almost 0 for between each parameter

Frequentist approach

Here we implement a frequentist approach and then I will compare the result with the bayesian analysis done before. In frequentist inference probabilities are interpreted in terms of the relative frequency of a certain observable event occurring, instead in Bayesian inference, since the parameters are unobservable, probabilities associated with unobservable parameters are most naturally interpreted as “degrees of belief” on a particular state of the unobservable parameters. Another difference is that in a frequentist approach procedures are random but parameters are fixed, in a bayesian approach instead everything is random.

Now let's see the implementation of a beta model (or beta regression) with a frequentist approach.

Small reminder of beta regression

The class of beta regression models is commonly used to model variables that assume values in the interval $(0, 1)$. It is based on the assumption that the dependent variable is beta-distributed and that its mean is related to a set of regressors through a linear predictor with unknown coefficients and a link function. But we are in the most simple case, in fact a $\text{beta}(1, 1)$ is equivalent to a $U([0, 1])$ so the p_i for the hospitals are just $\frac{\text{No of deaths}}{\text{No of ops}}$

Implementation

```
p = data$`No of deaths`/data$`No of ops`
p_estimates = data.frame(c('p[1]', 'p[2]', 'p[3]', 'p[4]', 'p[5]', 'p[6]', 'p[7]', 'p[8]',
                           'p[9]', 'p[10]', 'p[11]', 'p[12]'), p)
names(p_estimates) <- c('p', 'estimates')
kable(p_estimates)
```

p	estimates
p[1]	0.0000000
p[2]	0.1216216
p[3]	0.0672269
p[4]	0.0567901
p[5]	0.0379147
p[6]	0.0663265
p[7]	0.0608108
p[8]	0.1441860
p[9]	0.0676329
p[10]	0.0824742
p[11]	0.1132812
p[12]	0.0666667

We can observe that the estimates are similar to the estimates made with the bayesian model but there are some differences, for example $p[1]$ is equal to 0, this can represent one of the limit of the frequentist approach, in fact even if in our data the first hospital has no deaths it is impossible that the probability is 0, in fact in the bayesian approach $p[1]$ is near to 0 but not 0. The frequentist approach take in count only our data and not prior information or belief.

Logit model

If we assume that the failure rates across the hospitals are similar in some way we can use a more realistic model. We can specify a logit model for the true failure probabilities p_i as follows:

$$\begin{aligned} \text{logit}(p_i) &= b_i \\ b_i &\sim N(\mu, \tau) \end{aligned}$$

Where μ is the population mean and τ is the precision also known as the reciprocal of the variance ($\tau = \frac{1}{\sigma^2}$). Here we can see the graphical model of this bayesian analysis.

Also for this model we will run two different chains with 15000 iterations:

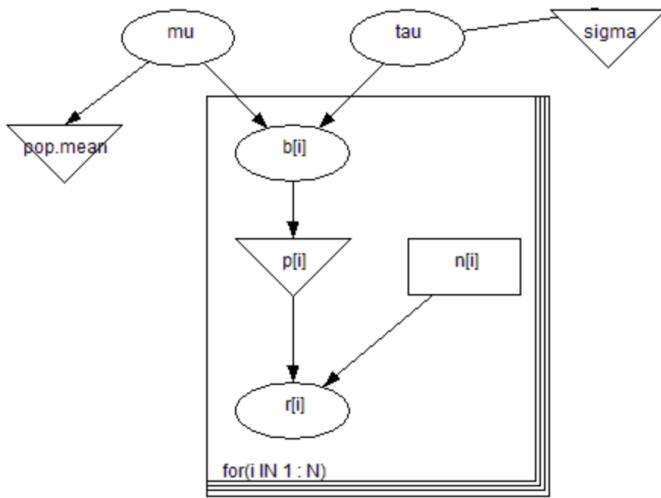


Figure 2: graphical model

```

parameters <- c('p')

inits <- list(list(b = c( 0.1,    0.1,    0.1,    0.1,    0.1,    0.1,    0.1,    0.1,
                      0.1,    0.1,    0.1,    0.1),tau = 1, mu = 0),
             list(b = c( 0.5,    0.5,    0.5,    0.5,    0.5,    0.5,    0.5,
                      0.5,    0.5,    0.5,    0.5, 0.5),tau = 0.1, mu = 1.0))

initial.values <- list(inits)

surgical_logit <- jags(data=data_list,inits=inits,parameters.to.save=parameters,
                         model.file="Model2.txt",n.chains=2,n.thin = 1,n.burnin=1000,n.iter=15000)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 12
##   Unobserved stochastic nodes: 14
##   Total graph size: 60
##
## Initializing model

print(surgical_logit)

## Inference for Bugs model at "Model2.txt", fit using jags,
## 2 chains, each with 15000 iterations (first 1000 discarded)
## n.sims = 28000 iterations saved
##          mu.vect sd.vect 2.5%   25%   50%   75% 97.5% Rhat n.eff
## p[1]      0.053  0.020  0.018  0.039  0.052  0.066  0.094 1.001 28000
## p[2]      0.103  0.022  0.067  0.087  0.101  0.117  0.151 1.001 24000

```

```

## p[3]      0.071  0.018  0.040  0.058  0.070  0.082  0.109 1.001 28000
## p[4]      0.059  0.008  0.044  0.054  0.059  0.064  0.076 1.001 28000
## p[5]      0.052  0.013  0.028  0.042  0.051  0.061  0.079 1.001 28000
## p[6]      0.069  0.015  0.043  0.059  0.068  0.078  0.100 1.001 7900
## p[7]      0.067  0.016  0.038  0.056  0.066  0.077  0.101 1.001 11000
## p[8]      0.123  0.022  0.084  0.107  0.122  0.138  0.171 1.001 6200
## p[9]      0.070  0.015  0.044  0.060  0.069  0.079  0.102 1.001 28000
## p[10]     0.079  0.020  0.045  0.065  0.077  0.090  0.124 1.001 28000
## p[11]     0.102  0.018  0.072  0.090  0.101  0.114  0.141 1.001 25000
## p[12]     0.068  0.012  0.047  0.060  0.068  0.076  0.093 1.001 28000
## deviance  65.998  5.508 57.350 62.051 65.281 69.122 78.893 1.001 27000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 15.2 and DIC = 81.2
.....

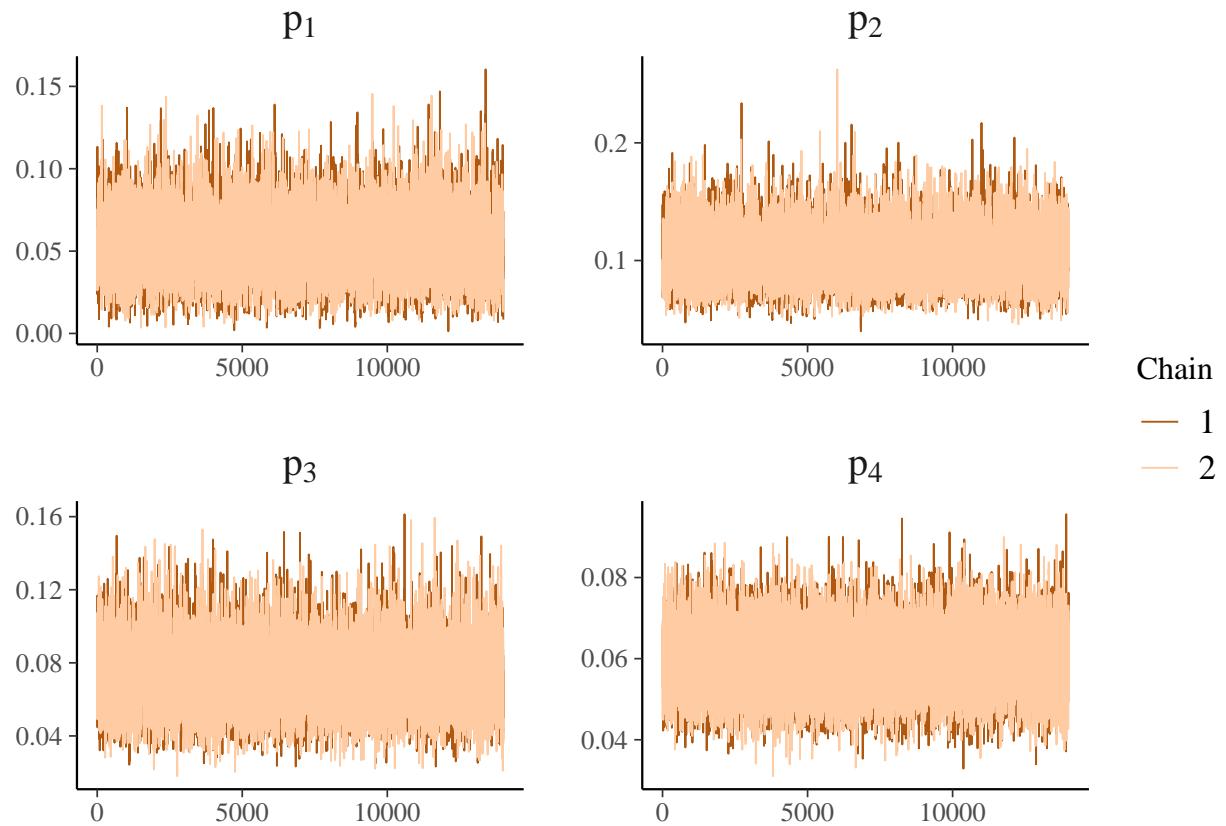
```

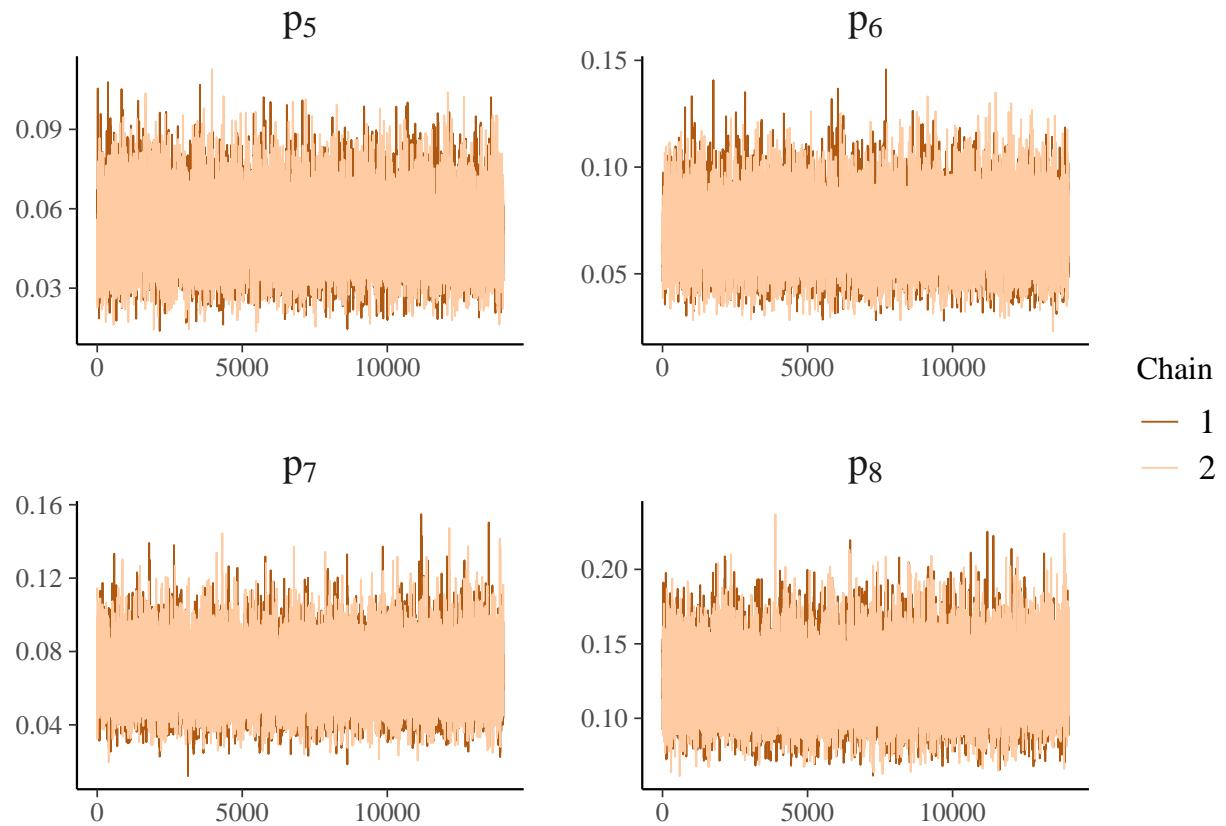
We can notice that the results are really different from the ones of the model used before, this is because we used a more complex model and we assumed that the failure probabilities are not independent but similar in some way. also in this model all the credible intervals for the p_i do not contain the 0, so the variables are statistically significant. The $R_{\text{hat}} = 1$ so the series converges. We can observe that in contrast with the model used before, here the hospital with the lowest probability of death is 5th (hospital E) with the beta model it was hospital A. The hospital with the highest probability of death is still hospital H.

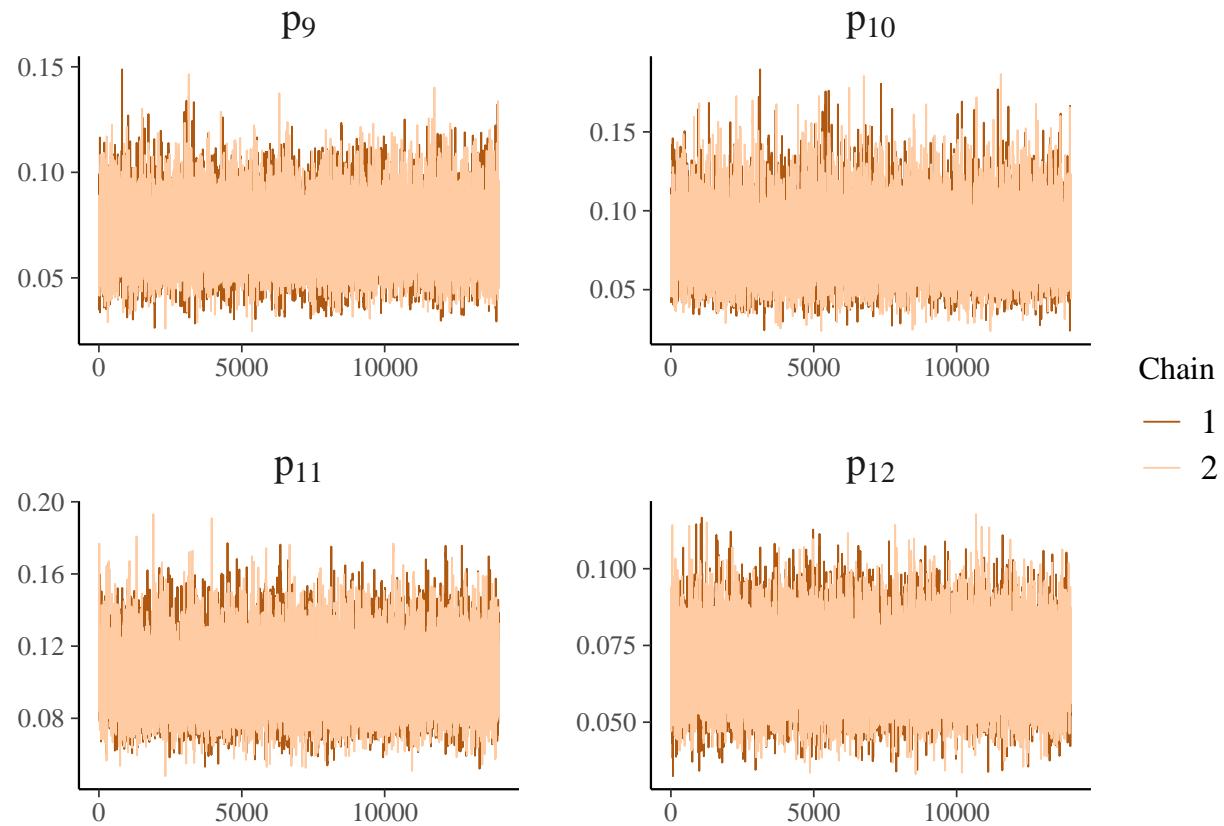
Now we compare the results with the paper's results:

	mean	median	sd	MC_error	val2.5pc	val97.5pc	start	sample	ESS
p[1]	0.05298	0.05224	0.01956	2.329E-4	0.01785	0.09385	1001	20000	7053
p[2]	0.1031	0.1007	0.02177	2.08E-4	0.06719	0.1511	1001	20000	10953
p[3]	0.07048	0.06921	0.01747	1.34E-4	0.03968	0.109	1001	20000	16997
p[4]	0.05918	0.05892	0.007908	6.766E-5	0.04444	0.07532	1001	20000	13659
p[5]	0.05152	0.05083	0.01327	1.558E-4	0.02742	0.07889	1001	20000	7252
p[6]	0.06895	0.06804	0.01471	1.077E-4	0.04248	0.1004	1001	20000	18639
p[7]	0.0668	0.0657	0.01597	1.271E-4	0.0386	0.1017	1001	20000	15788
p[8]	0.1231	0.1218	0.0223	2.659E-4	0.08342	0.1703	1001	20000	7032
p[9]	0.0698	0.06891	0.01451	1.181E-4	0.04407	0.1007	1001	20000	15080
p[10]	0.0784	0.07658	0.01988	1.453E-4	0.04463	0.1225	1001	20000	18726
p[11]	0.1023	0.101	0.01762	1.752E-4	0.07186	0.1403	1001	20000	10107
p[12]	0.06838	0.06786	0.01174	9.719E-5	0.04695	0.09286	1001	20000	14595

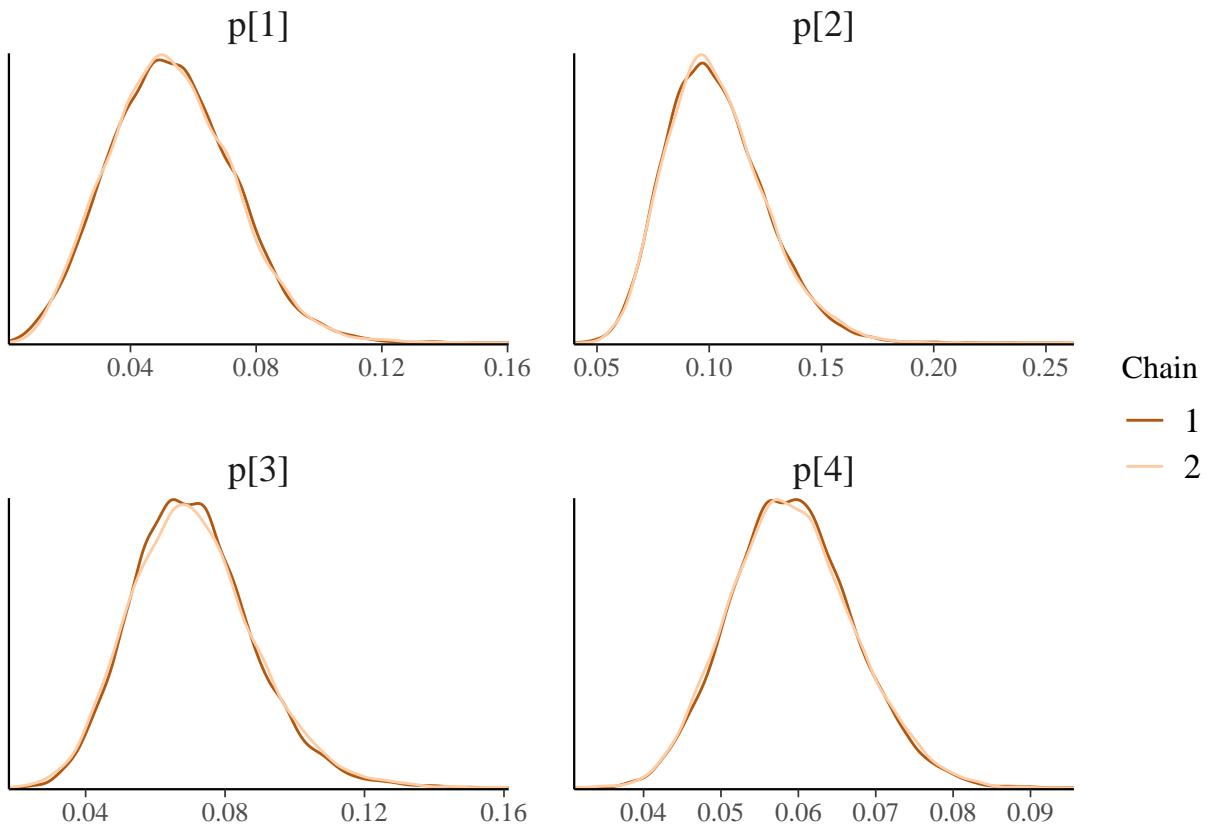
The results are really similar, the most different estimates is the one for p[2] that is 0.104 in my results and 0.1031 in the paper. Also for this model we are going to see the traceplot

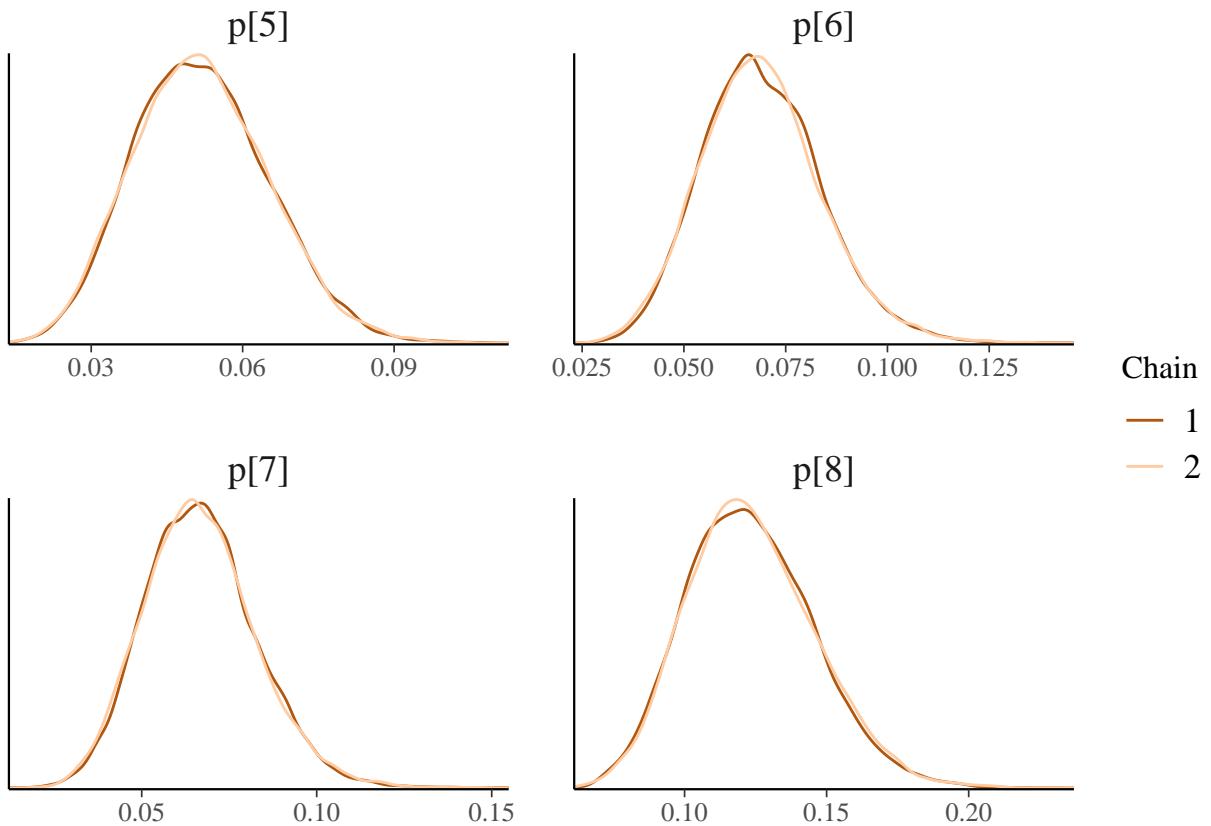


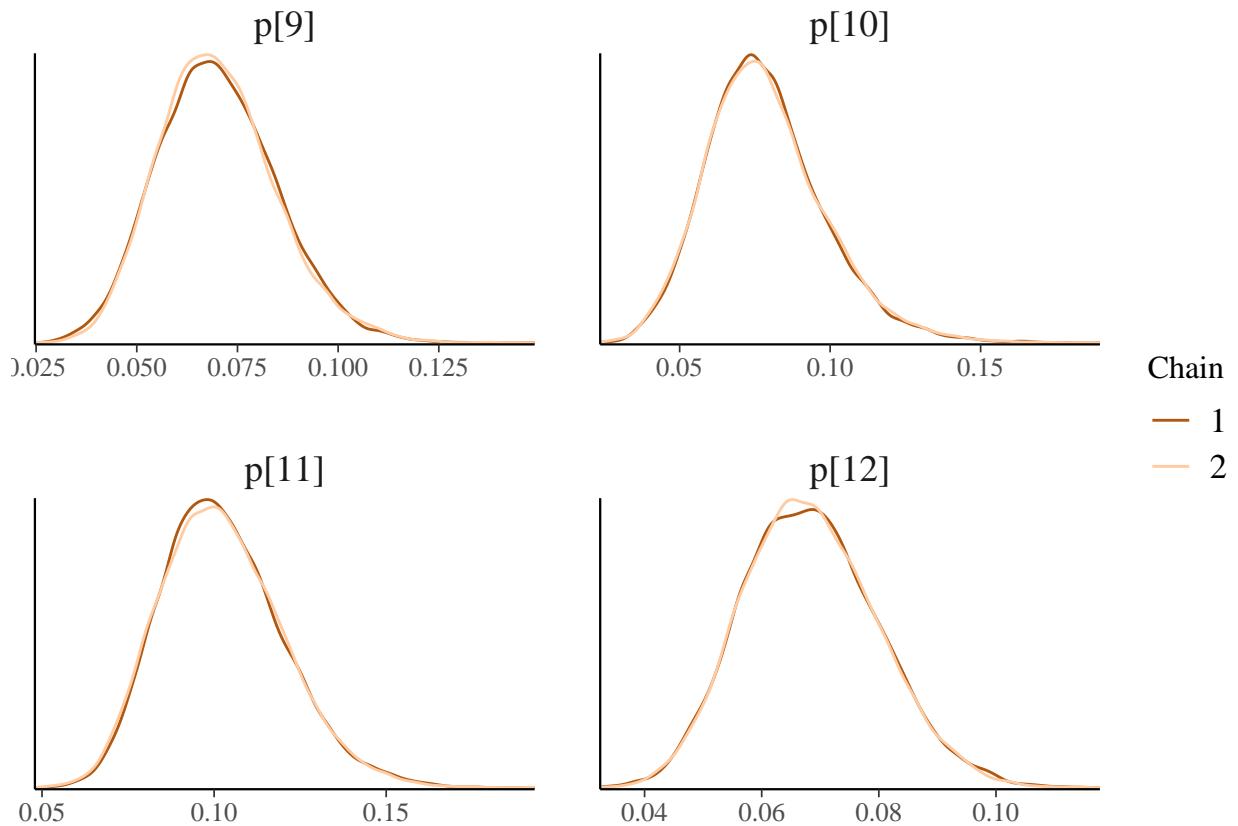


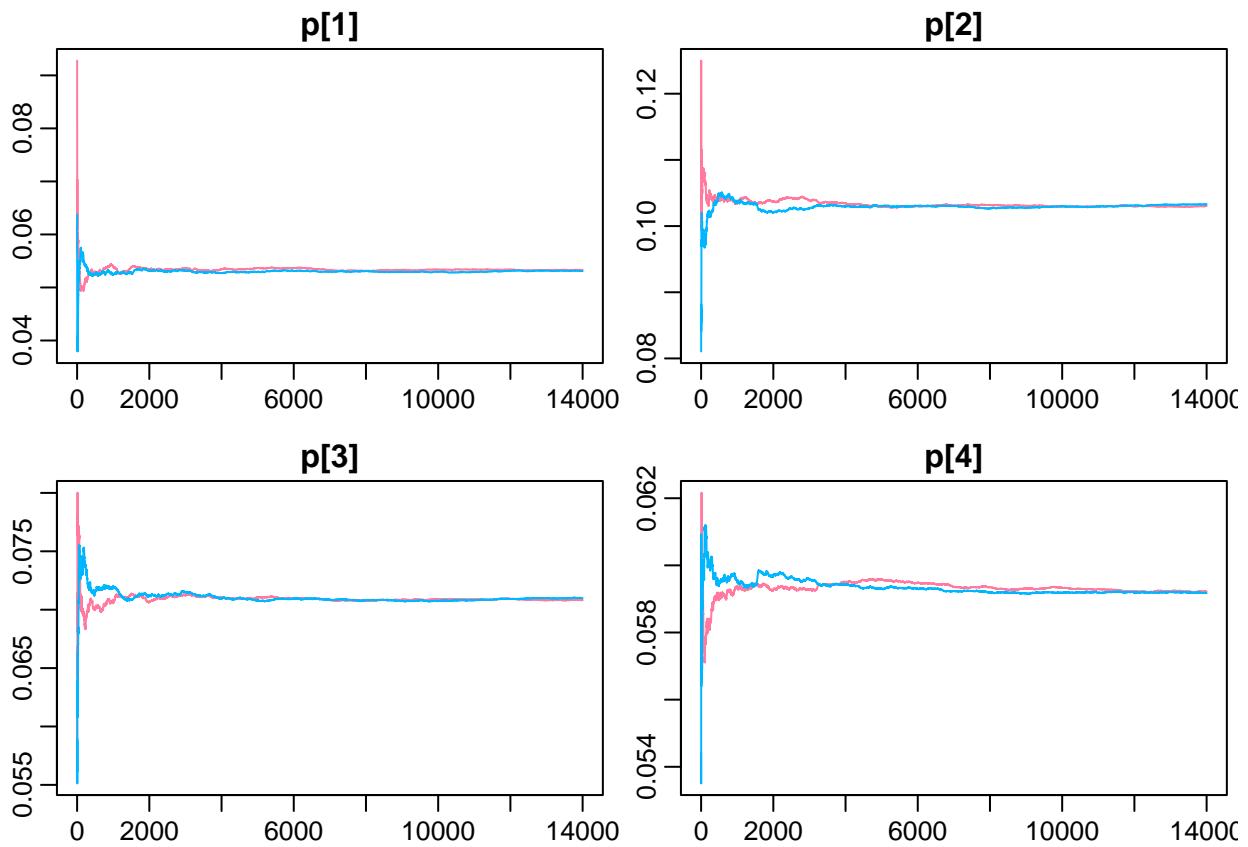


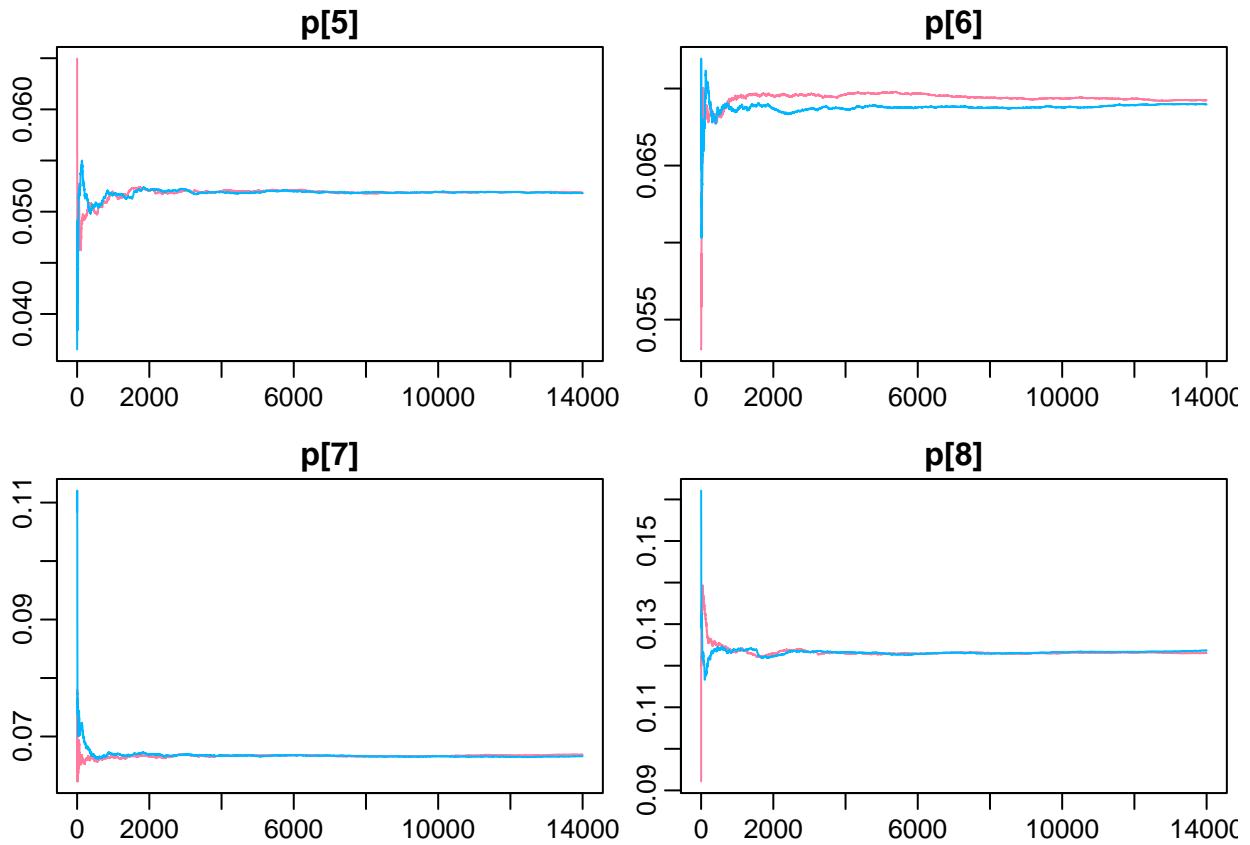
We can notice as in the model before that the chains have a random behaviour around the estimate

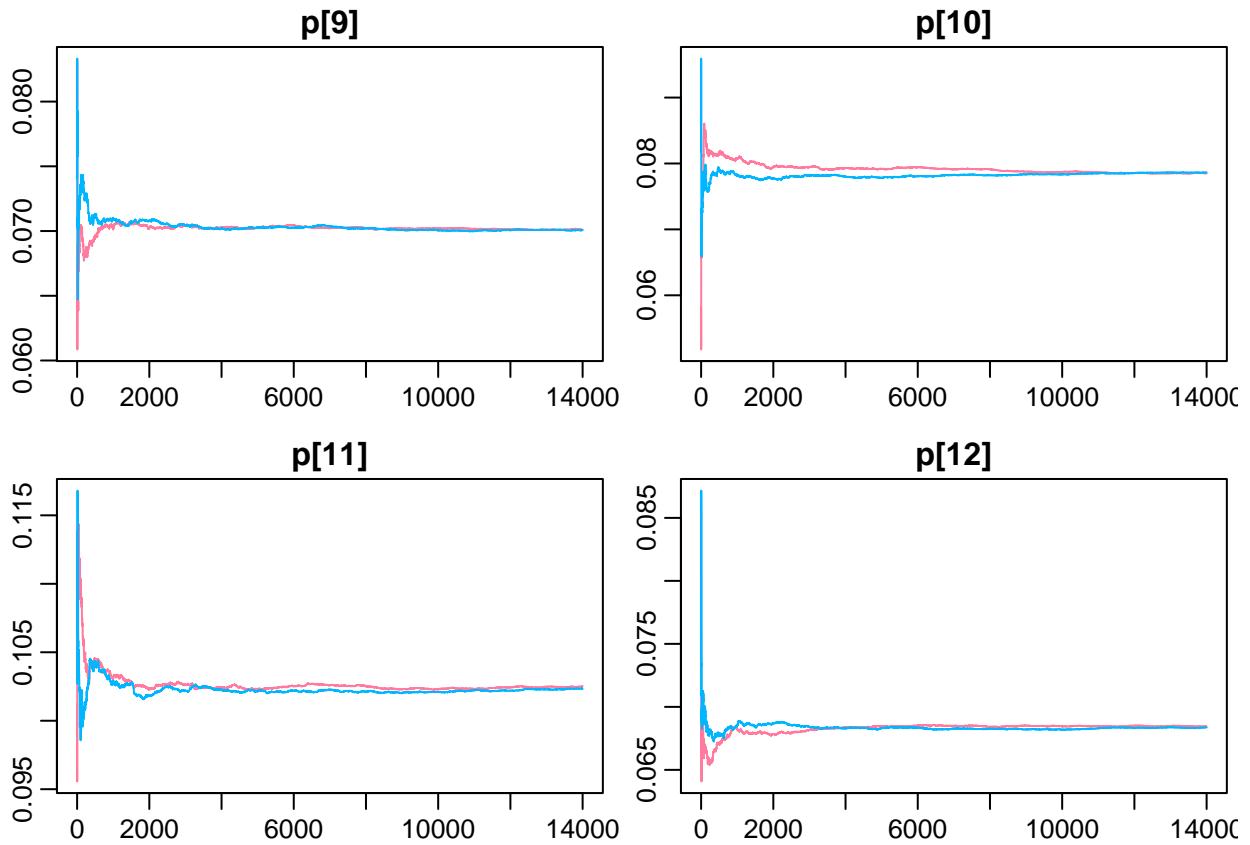






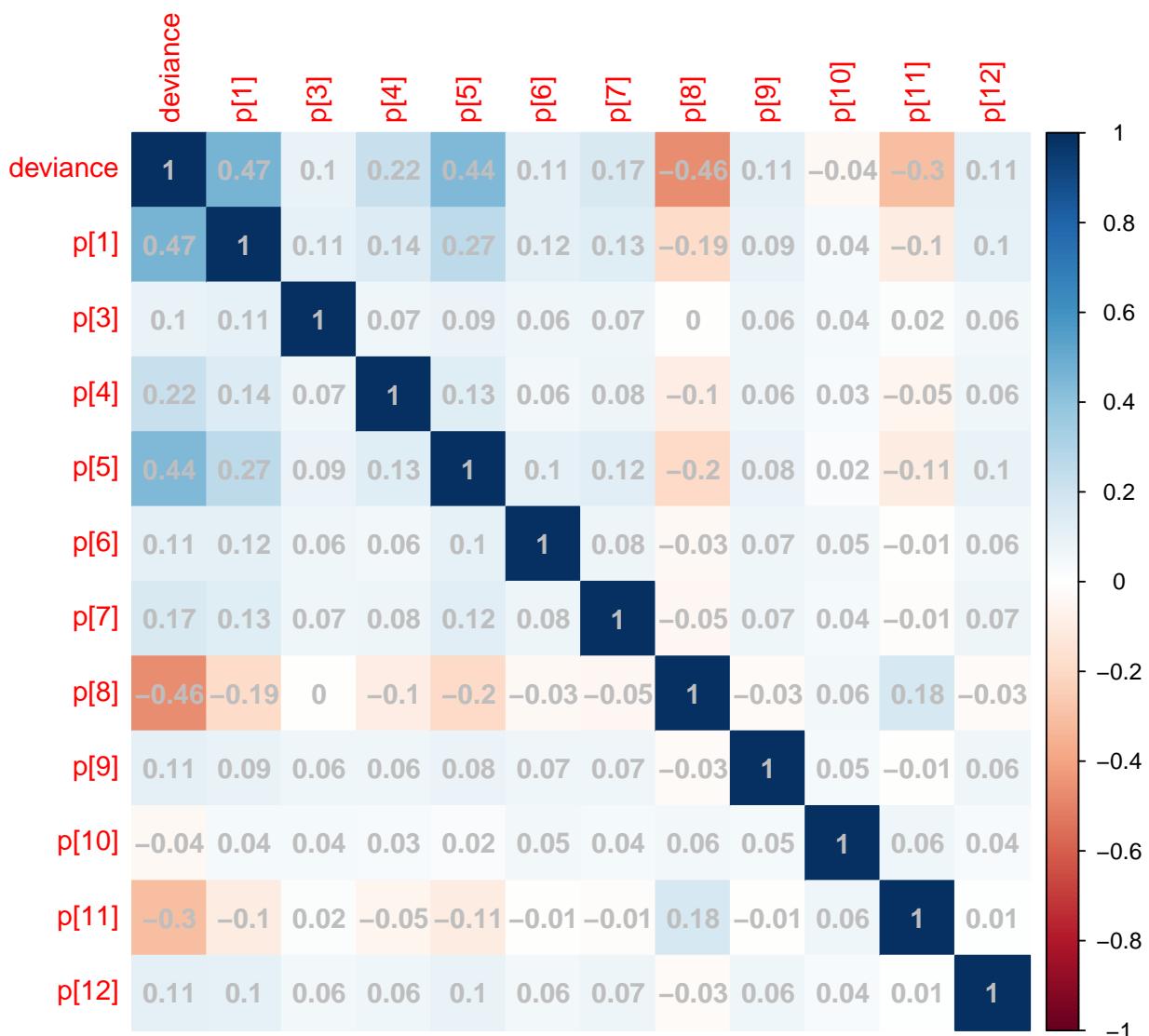






From the running means seems that in this model the parameters take a more time to converge (just little more time)

```
corrplot::corrplot(cor(surgical_logit$BUGSoutput$sims.matrix)[-3,-3],
                   method = "color", addCoef.col="grey")
```



There is not a big correlation between the parameters but it is still way bigger than the beta model seen before, for example between p[1] and p[5] the correlation is 0.26.

Frequentist approach

I will use the lme4 packet to implement a logistic regression with random effect and then I will compare the results, lme4 is used for generalized or linear model with random or fixed effects. To assume that in this model there are random effect means that part of the variability depend from the hospitals.

```
## $Hospital
##   (Intercept)
## A    -2.865888
## B    -2.174870
```

```

## C -2.586225
## D -2.769721
## E -2.911218
## F -2.605344
## G -2.645971
## H -1.964304
## I -2.593863
## J -2.479777
## K -2.175484
## L -2.614504
##
## attr(),"class")
## [1] "coef.mer"

```

These are the coefficient of the random effect in a frequentist approach but how I can interpret these results? for the interpretation I just need to extrapolate the probabilities from these coefficients so I can compare them with the other analysis. To transform these coefficients in probabilities I just need to do $\exp(\text{coef})/(1+\exp(\text{coef}))$

```

c = coef(logistic)$Hospital[,1]
p = rep(NA,12)
j = 1
for(i in c){

  p[j] = exp(i)/(1+exp(i))
  j=j+1
}
p_estimates = data.frame(c('p[1]', 'p[2]', 'p[3]', 'p[4]', 'p[5]', 'p[6]', 'p[7]', 'p[8]',
                           'p[9]', 'p[10]', 'p[11]', 'p[12]'), p)
names(p_estimates) <- c('p', 'estimates')
kable(p_estimates)

```

p	estimates
p[1]	0.0538658
p[2]	0.1020300
p[3]	0.0700303
p[4]	0.0589825
p[5]	0.0516018
p[6]	0.0687953
p[7]	0.0662378
p[8]	0.1230020
p[9]	0.0695345
p[10]	0.0772881
p[11]	0.1019737
p[12]	0.0682108

These estimates of the parameters are really similar to the estimates done with the bayesian model.

Probit model

The probit model is similar to the logit model, but it is based on probits instead logistic functions. The process for calculating probabilities in logit and probits differ from each other because logistic functions use

linear combinations while probit uses cumulative standard normal distribution function.

In Logit: $Pr(Y = 1|X) = (1 + e^{-X'\beta})^{-1}$

In Probit: $Pr(Y = 1|X) = \Phi(X'\beta)$

Also for this model we will run two different chains with 15000 iterations:

```

parameters <- c('p')

inits <- list(list(b = c( 0.1,    0.1,    0.1,    0.1,    0.1,    0.1,    0.1,    0.1,
                      0.1,    0.1,    0.1,    0.1),tau = 1, mu = 0),
              list(b = c( 0.5,    0.5,    0.5,    0.5,    0.5,    0.5,    0.5,
                        0.5,    0.5,    0.5,    0.5),tau = 0.1, mu = 1.0))

initial.values <- list(inits)

surgical_probit <- jags(data=data_list,inits=inits,parameters.to.save=parameters,
                           model.file="model3.txt",n.chains=2 , n.thin = 1,n.burnin=1000,n.iter=15000)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 12
##   Unobserved stochastic nodes: 14
##   Total graph size: 58
##
## Initializing model

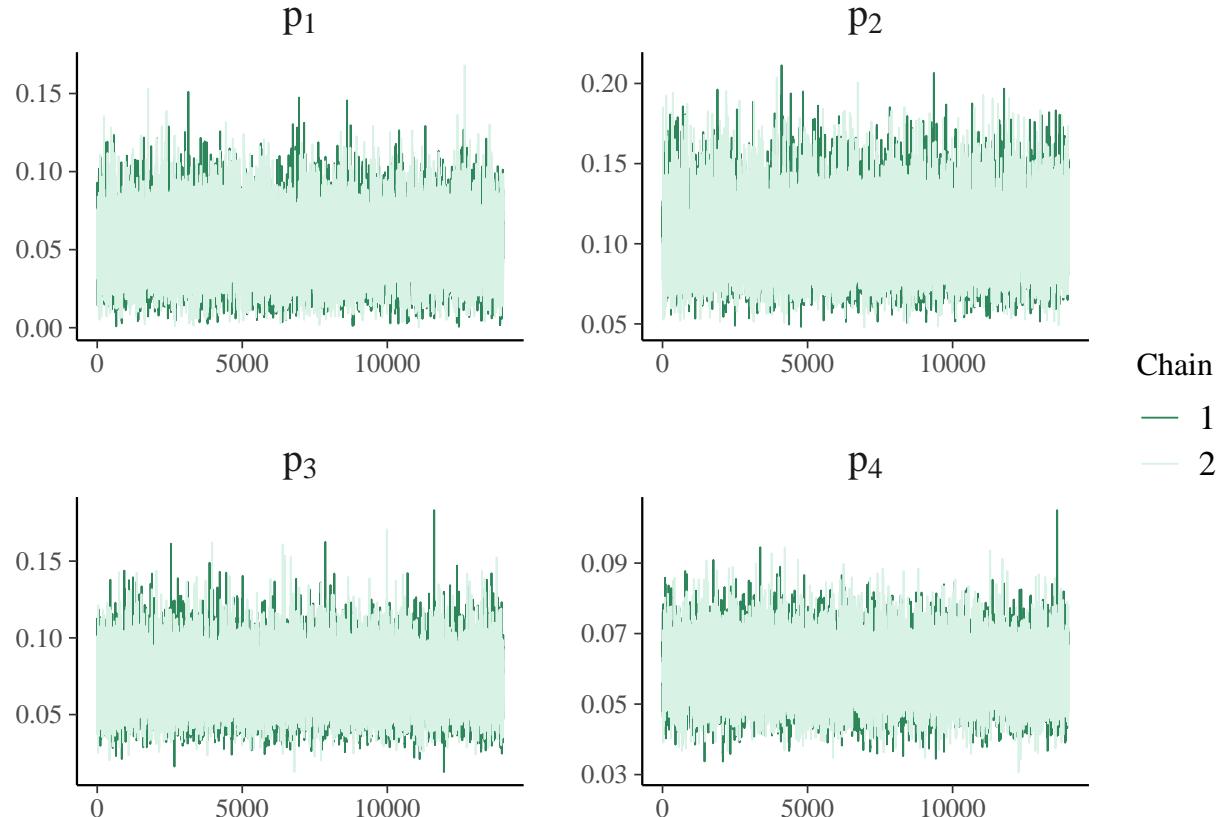
print(surgical_probit)

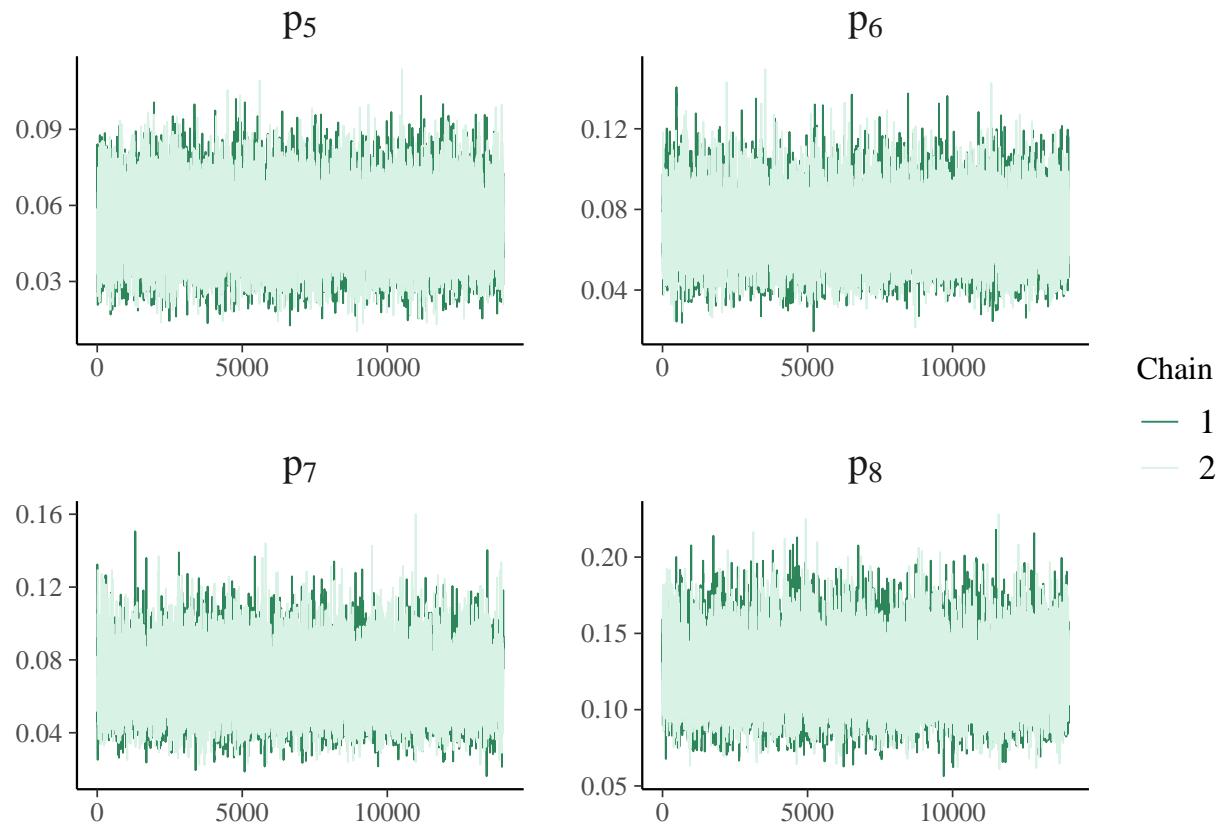
## Inference for Bugs model at "model3.txt", fit using jags,
## 2 chains, each with 15000 iterations (first 1000 discarded)
## n.sims = 28000 iterations saved
##          mu.vect sd.vect 2.5%   25%   50%   75% 97.5% Rhat n.eff
## p[1]      0.052  0.021  0.014  0.037  0.051  0.066  0.094 1.001 14000
## p[2]      0.104  0.022  0.067  0.088  0.101  0.117  0.152 1.002 2300
## p[3]      0.071  0.018  0.040  0.059  0.070  0.082  0.110 1.001 28000
## p[4]      0.059  0.008  0.045  0.054  0.059  0.064  0.076 1.001 28000
## p[5]      0.051  0.014  0.026  0.041  0.050  0.060  0.080 1.002 2700
## p[6]      0.070  0.015  0.043  0.059  0.069  0.079  0.102 1.001 11000
## p[7]      0.067  0.016  0.038  0.056  0.066  0.077  0.102 1.001 28000
## p[8]      0.123  0.022  0.084  0.107  0.121  0.137  0.170 1.001 28000
## p[9]      0.070  0.015  0.044  0.060  0.070  0.080  0.102 1.001 28000
## p[10]     0.079  0.020  0.045  0.066  0.077  0.091  0.123 1.001 13000
## p[11]     0.103  0.017  0.072  0.090  0.101  0.113  0.140 1.002 1600
## p[12]     0.069  0.012  0.047  0.060  0.068  0.076  0.093 1.001 28000
## deviance  65.849  5.444 57.062 61.951 65.227 69.055 78.230 1.002 2500
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
```

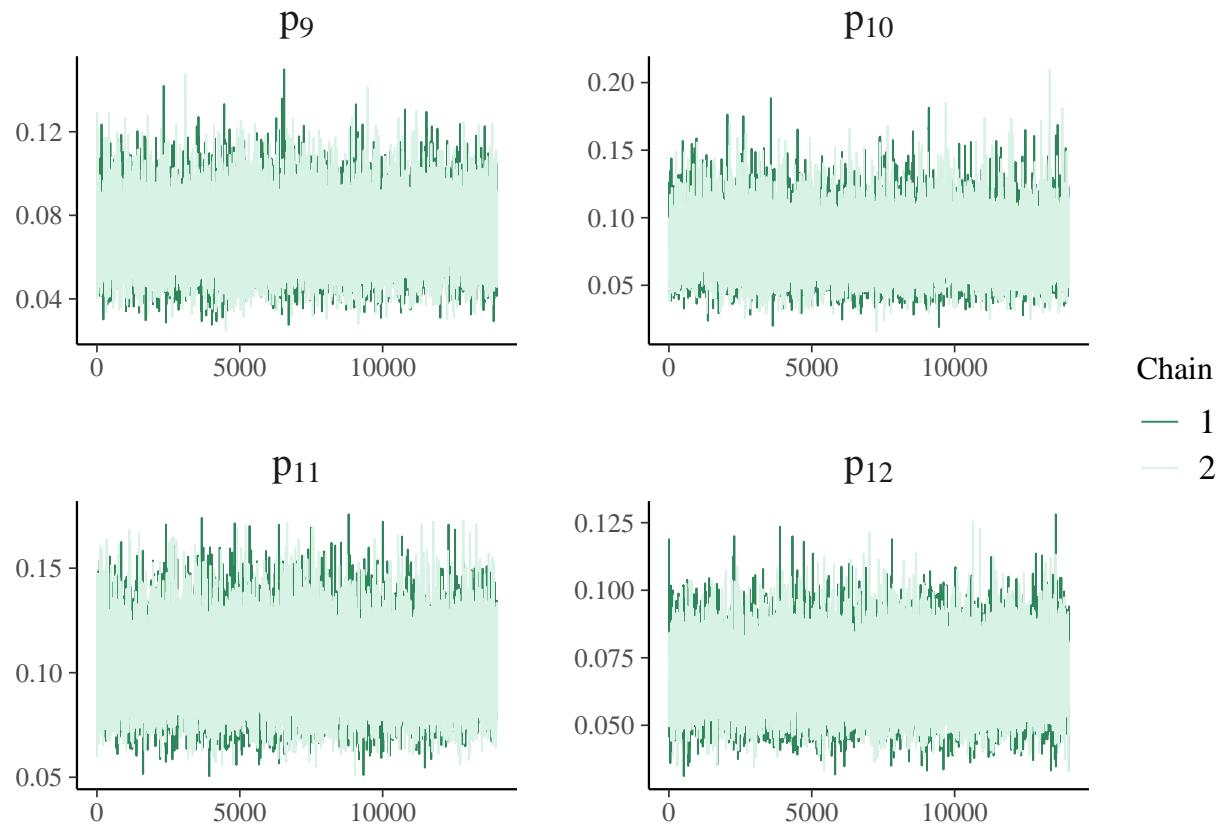
```
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 14.8 and DIC = 80.7
....
```

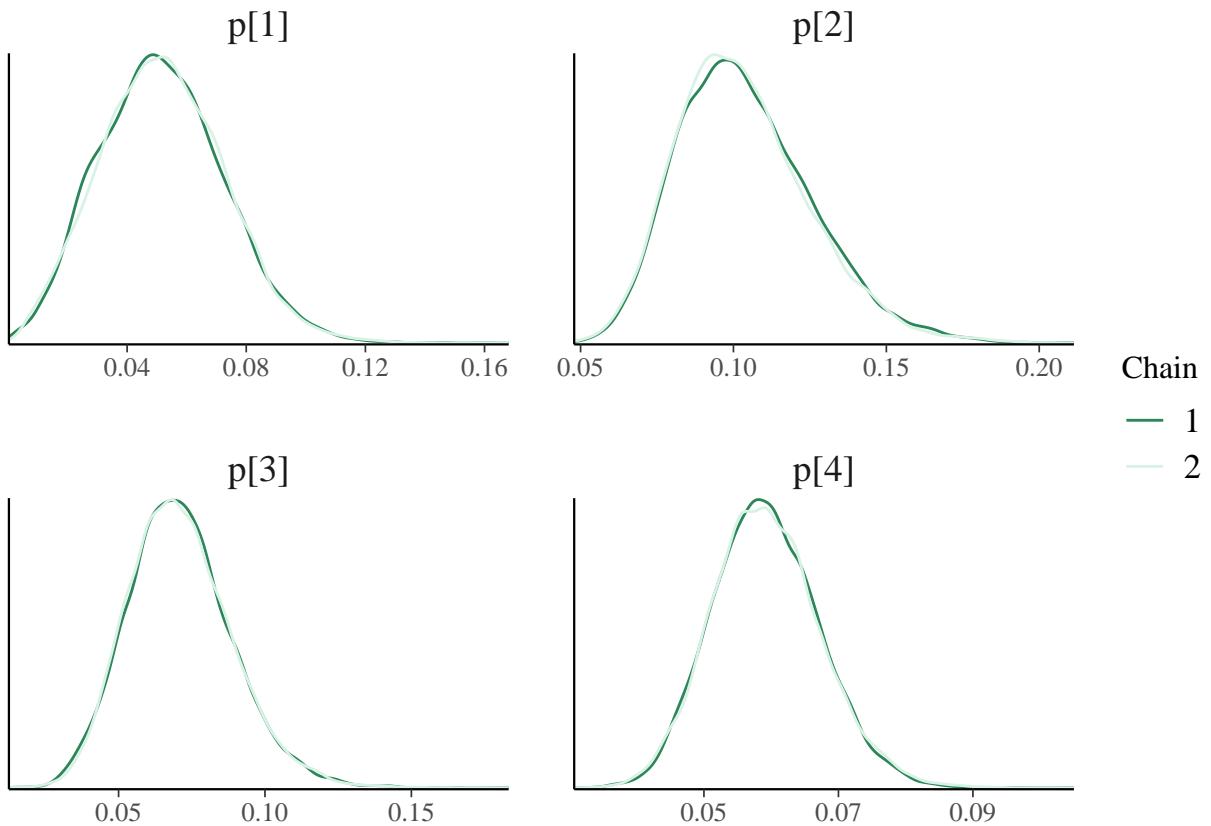
Observing these result we can notice that are really similar to the results obtained with the logit model, also here the estimates are significant and the chains converges. In this model the hospital A and E have the same probabilities and they are both the hospitals with the lowest probabilities of death, so it seems that this model is a ‘compromise’ between the other two models.

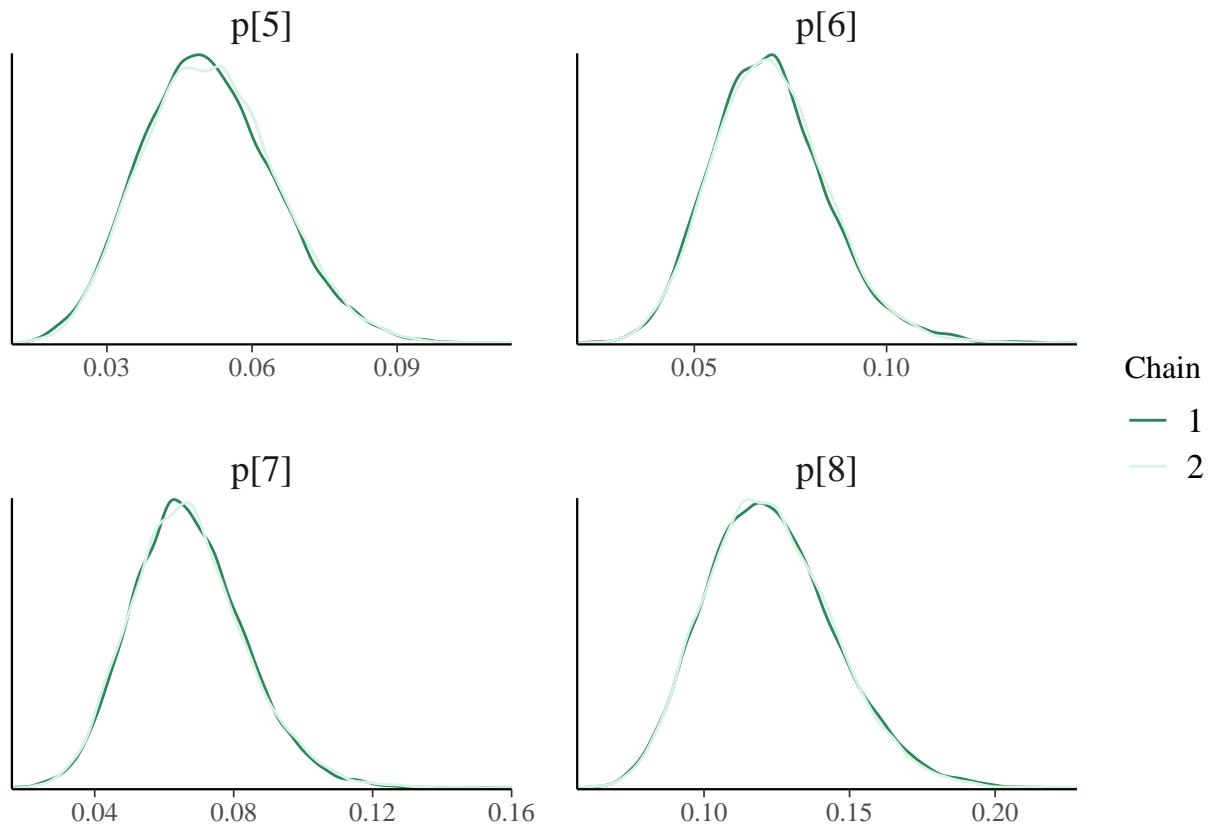
Now let's have a look to the traceplots, density plots and running means.

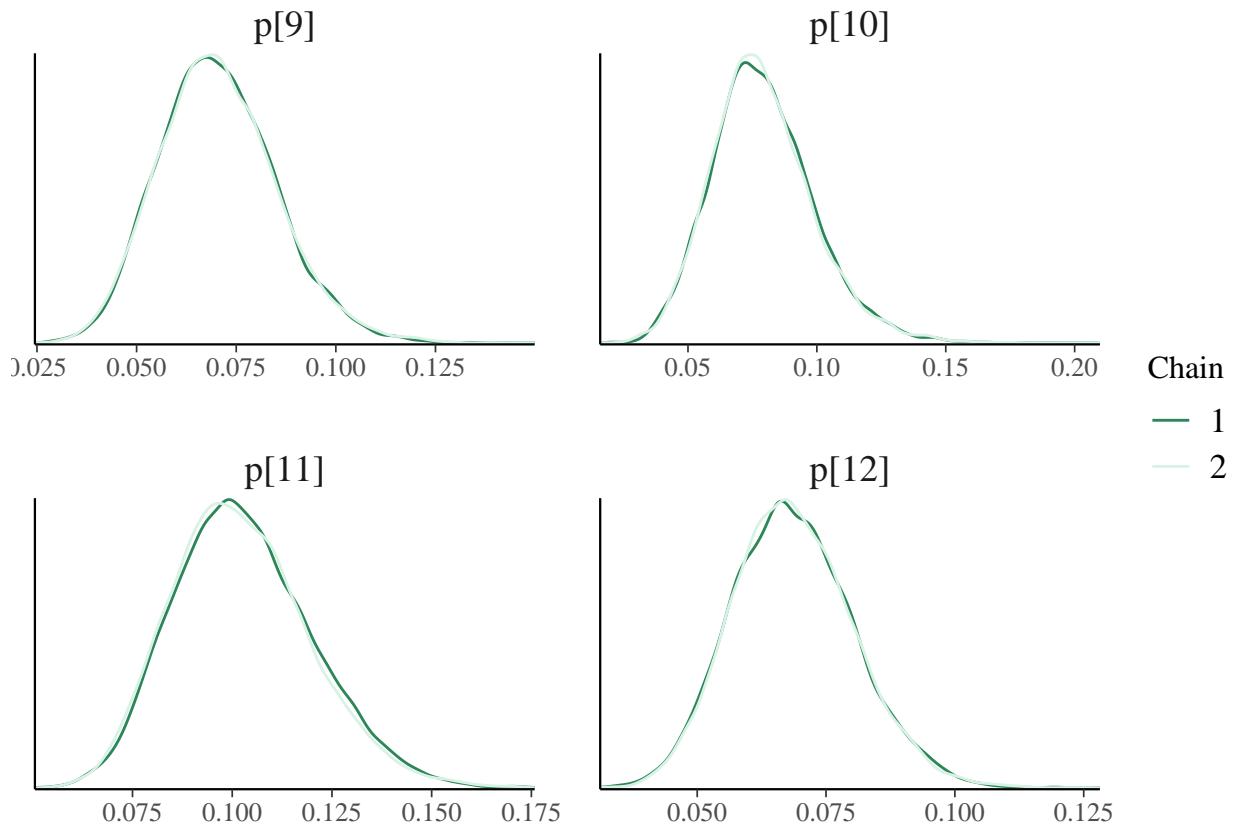


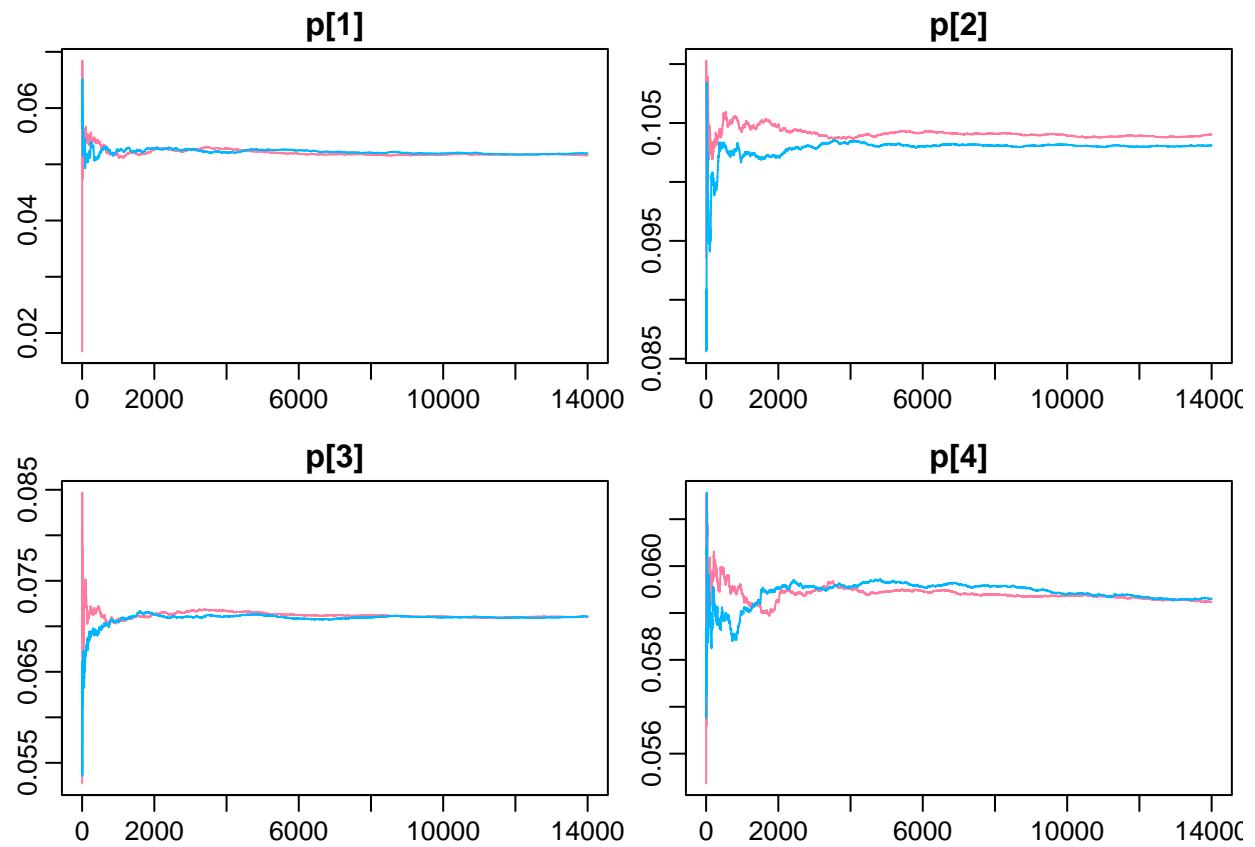


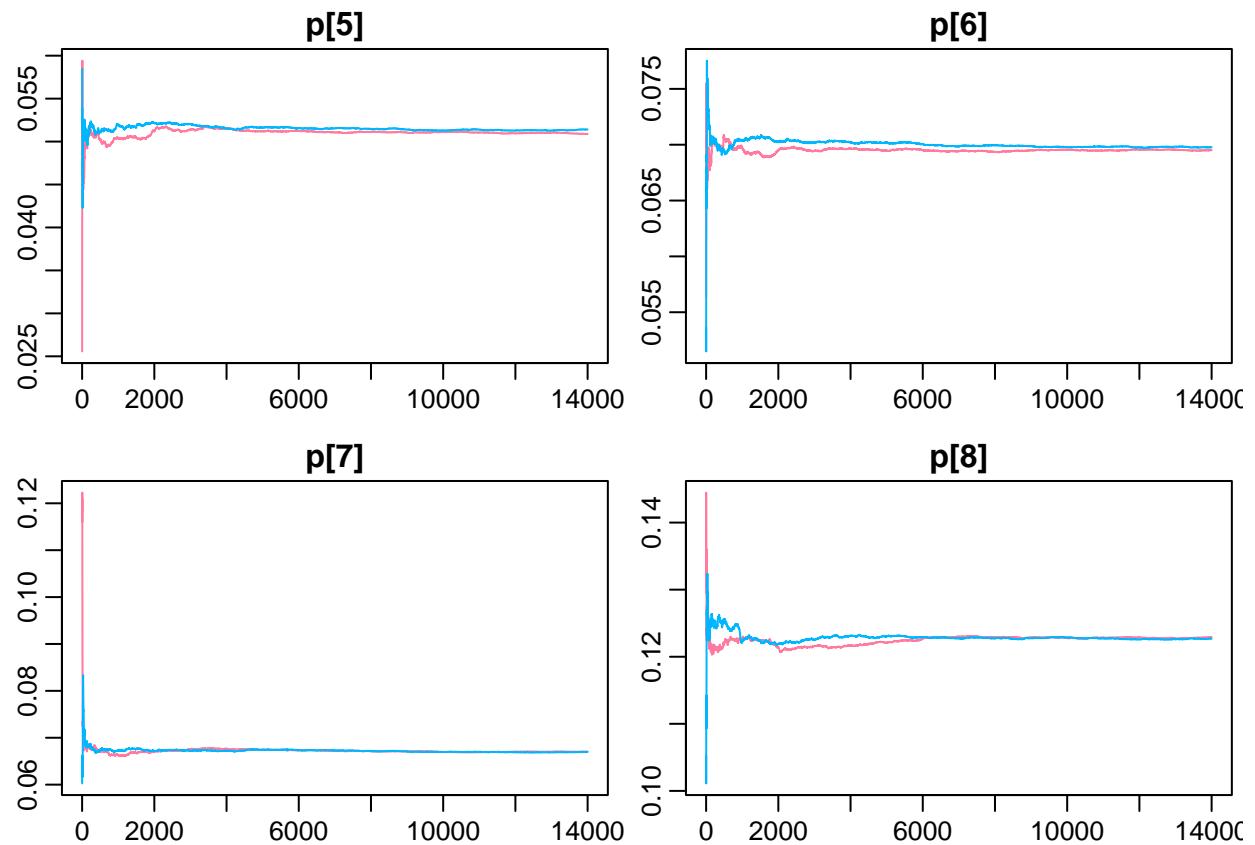


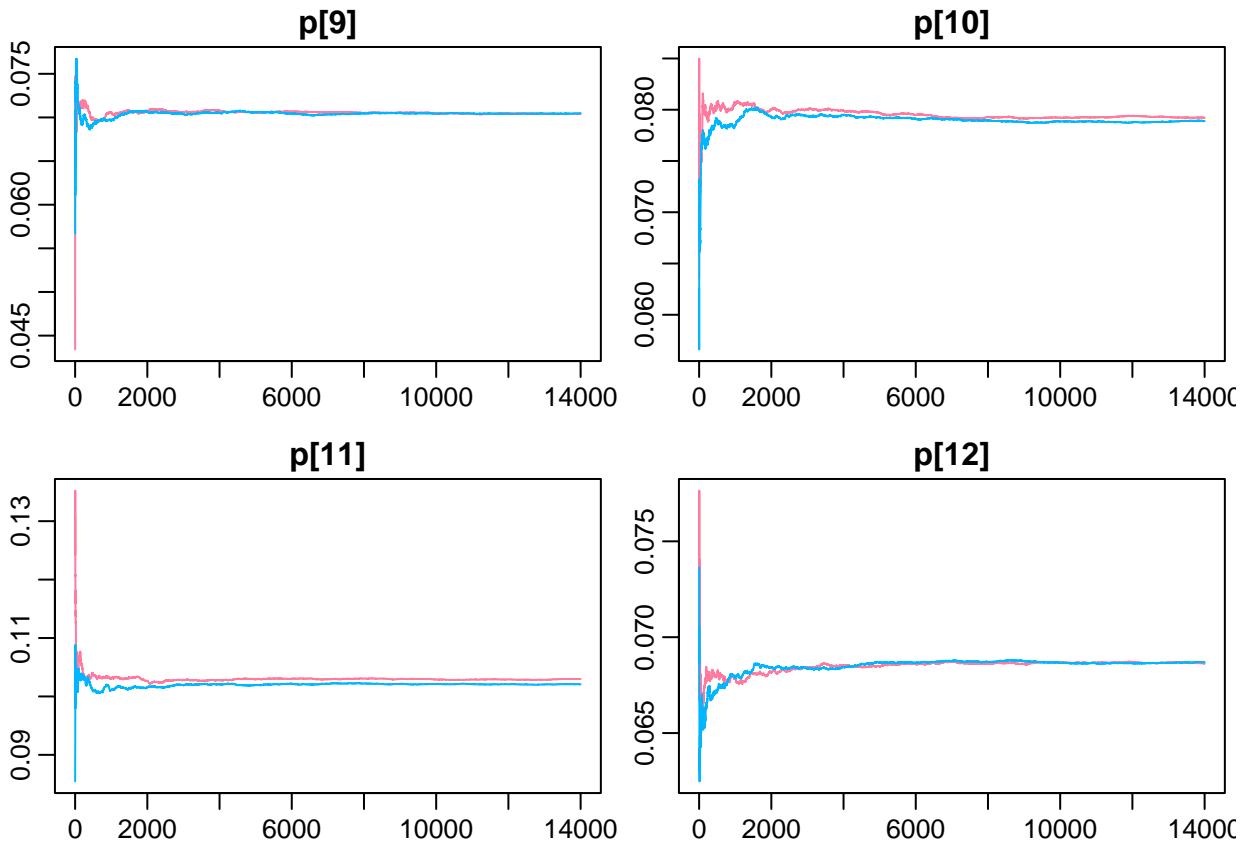






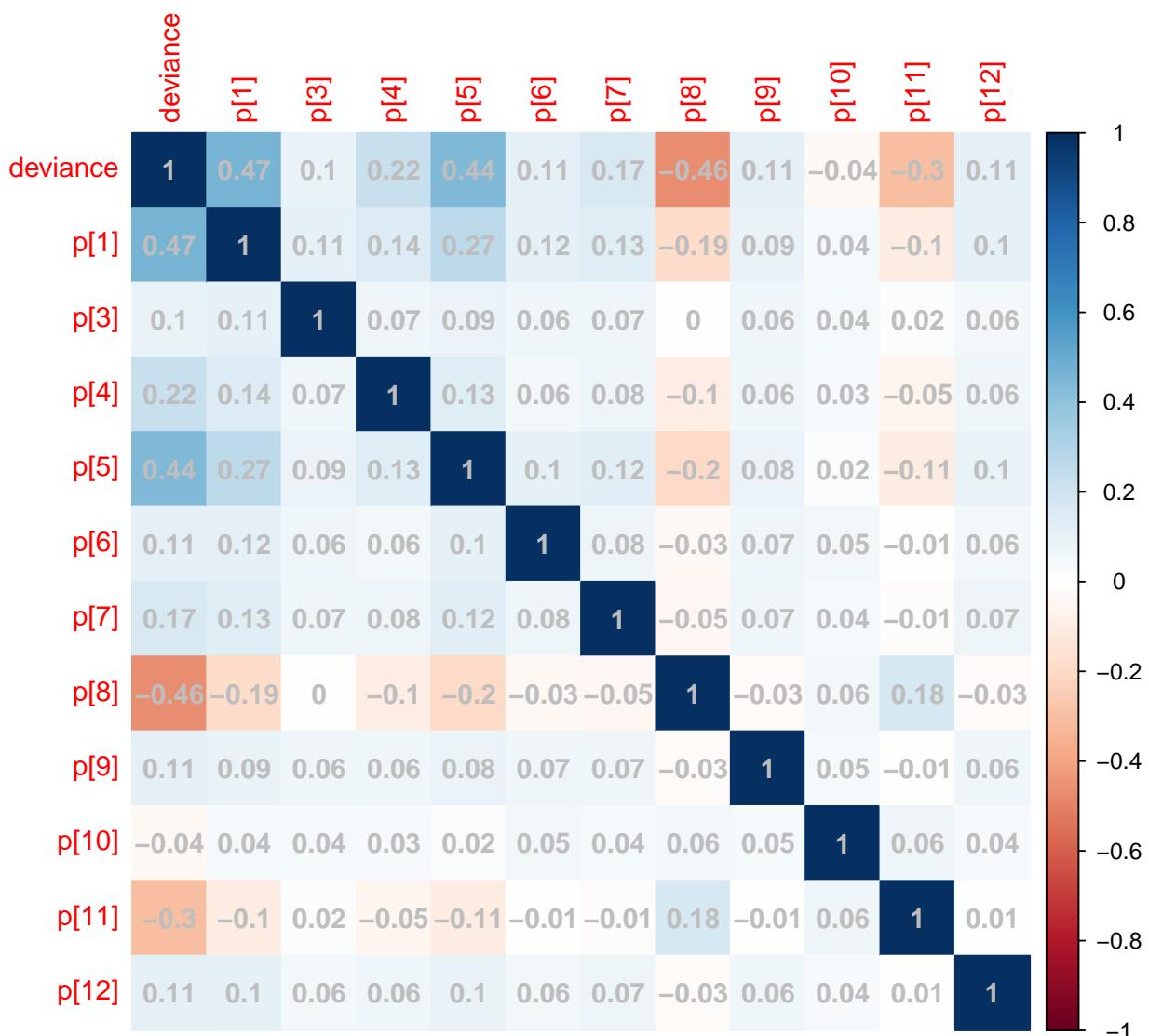






It is interesting to notice from the running means that the probit model takes way more time to converge than the beta and the logit model.

```
corrplot::corrplot(cor(surgical_logit$BUGSoutput$sims.matrix)[-3,-3],
                   method = "color", addCoef.col="grey")
```



The corrplots of the logit and the probit models are almost identical.

Frequentist approach

I will use the same approach of the logistic model I will just change the function from logit to probit

```
probit_freq<-glmer(cbind(`No of deaths`, `No of ops`~`No of deaths`) ~ (1| Hospital),
                     data=data, family = binomial(link = 'probit'))
coef(probit_freq)
```

```
## $Hospital
##   (Intercept)
## A -1.619275
```

```
## B -1.272602
## C -1.474891
## D -1.564302
## E -1.636910
## F -1.484500
## G -1.504623
## H -1.166786
## I -1.478844
## J -1.422119
## K -1.272134
## L -1.489104
##
## attr(),"class")
## [1] "coef.mer"
```

To interpret the these coefficients and transform them in probabilities I just need to compute the CDF on the standard normal distribution on the coefficients.

```
c = coef(probit_freq)$Hospital[,1]
p = rep(NA,12)
j = 1
for(i in c){

  p[j] = pnorm(i)
  j=j+1
}
p_estimates = data.frame(c('p[1]', 'p[2]', 'p[3]', 'p[4]', 'p[5]', 'p[6]', 'p[7]', 'p[8]',
                           'p[9]', 'p[10]', 'p[11]', 'p[12]'), p)
names(p_estimates) <- c('p', 'estimates')
kable(p_estimates)
```

p	estimates
p[1]	0.0526941
p[2]	0.1015796
p[3]	0.0701210
p[4]	0.0588733
p[5]	0.0508246
p[6]	0.0688381
p[7]	0.0662105
p[8]	0.1216483
p[9]	0.0695910
p[10]	0.0774959
p[11]	0.1016628
p[12]	0.0682300

The estimations are really similar to the estimations made with a bayesian approach they are just a little bit smaller, for example p[3] is 0.701 instead of 0.702.

Comparison of the three models

To compare the three models we can compare the DIC value (Deviance Information Criterion). The DIC value is based on the deviance and is calculated in the following way:

$$DIC = 2p_D + D(\bar{\theta})$$

where $D(\bar{\theta}) = -2\log(L(\bar{\theta}))$ that is the deviance of the model.

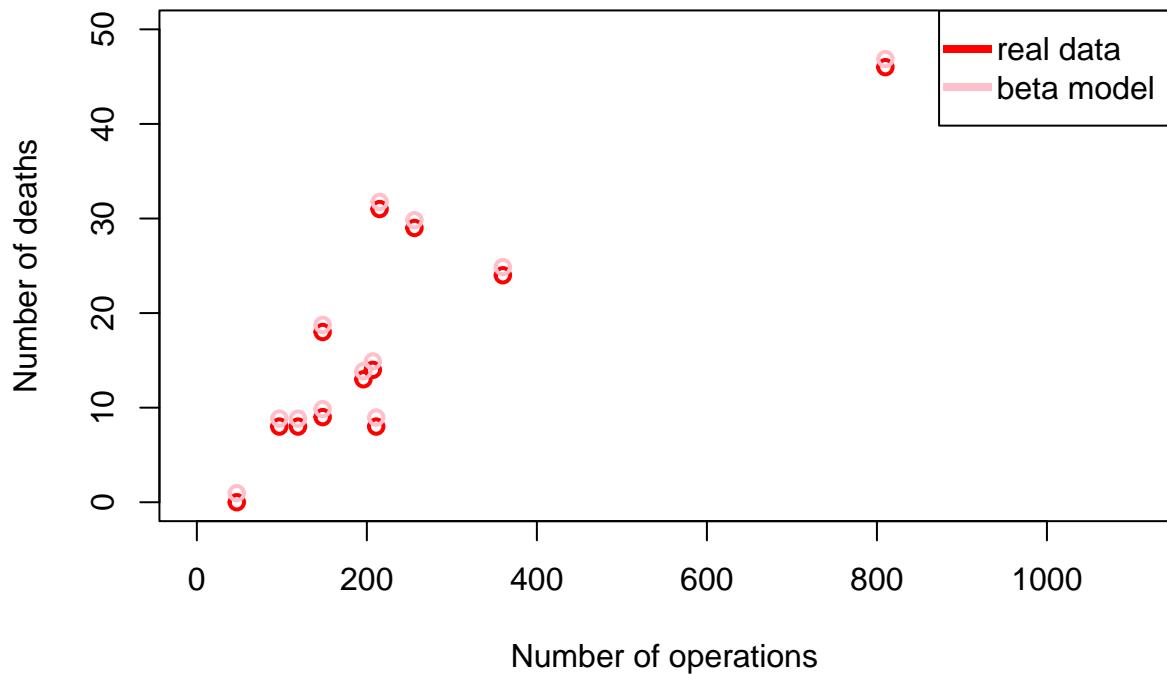
Now we can compute it for the three models with Jags and see which model has the lowest value

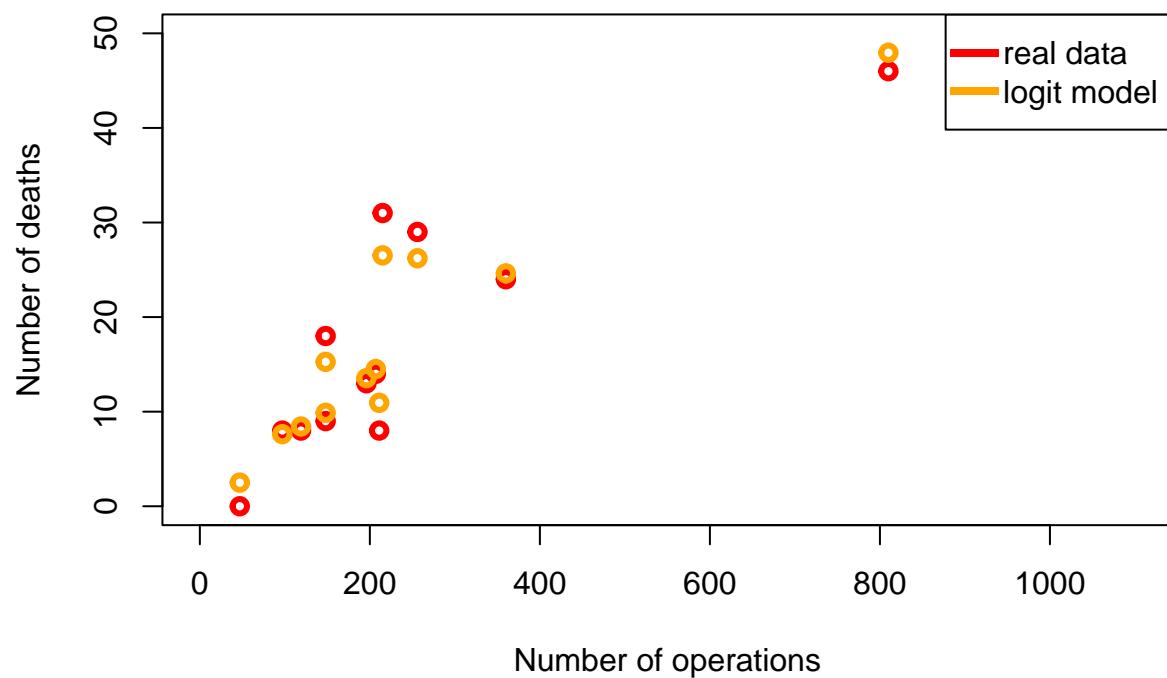
```
##   Beta model Logit model Probit model
##    75.55252     81.16845     80.66256
```

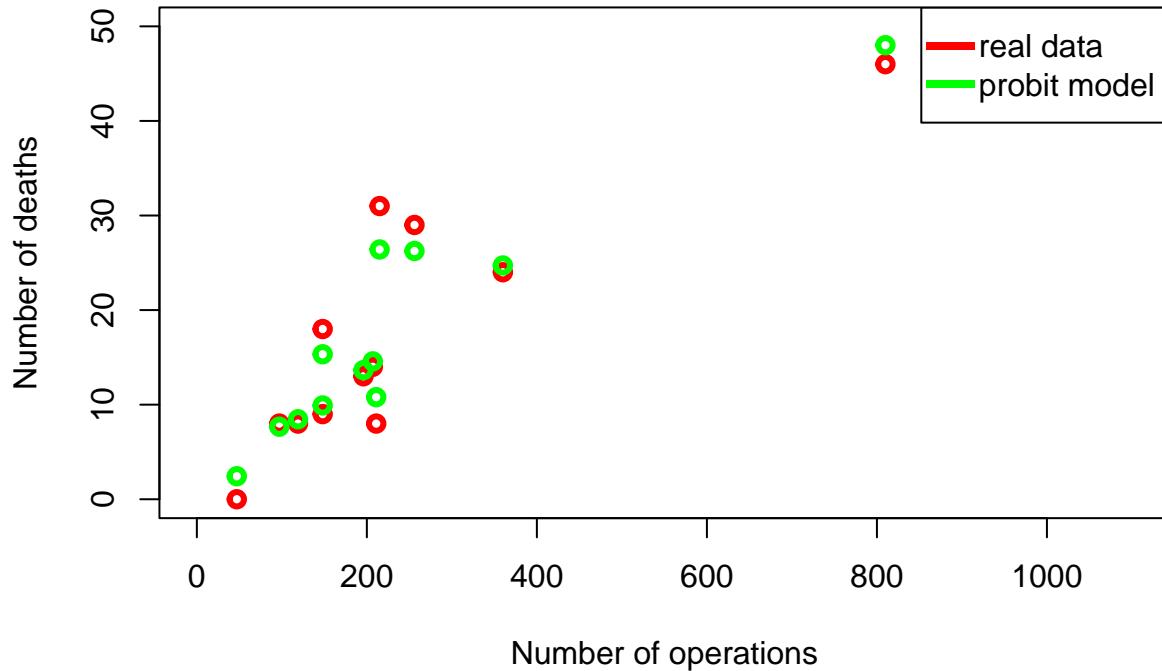
It is clear that from this value the best model seems to be the Beta model, but we should also consider that the Beta model is the less realistic because in the real world we should consider some random effects. It is interesting to notice that from the two most realistic models (logit and probit) the probit model seems to be better.

To have another comparison between the models we can compare the estimates that we got with the real data. Since we estimated the probabilities of death we can compare the real number of death in the hospitals with the one computed with the probabilities of our model.

```
n = data_list$n
r = data_list$r
p_beta = surgical_beta$BUGSoutput$summary[2:13,1]
p_logit = surgical_logit$BUGSoutput$summary[2:13,1]
p_probit = surgical_probit$BUGSoutput$summary[2:13,1]
r_beta = p_beta*n
r_logit = p_logit*n
r_probit = p_probit*n
```







As it was expected the r estimates more similar to the real data are the one made with the Beta model but us I said before this model fits well with this data but this is the less realistic model in the real world. Another interesting but expected thing is that the logit and probit models made really similar estimates.

Predictions

We can use the models that we studied also to predict the number of death in one of the hospital given the number of operations. We consider the probit model because it turned out to be a little bit better than the logit and it is more realistic than the beta model. If we assume that each hospital does 500 new operations we can predict the number of death.

```
deaths = rep(NA, 12)

for(j in 1:12){
  deaths[j] = rbinom(1, 500, p_probit[j])
}

n_deaths = data.frame(c('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L'), deaths)
names(n_deaths) <- c('Hospital', 'n_deaths')
kable(n_deaths)
```

Hospital	n_deaths
A	23
B	56
C	29

Hospital	n_deaths
D	30
E	25
F	51
G	31
H	62
I	29
J	42
K	58
L	33

The worst hospital is definitely the hospital H, the best one is the hospital D.

Conclusion

I analyzed the different models, it is clear that the best model seems to be the beta model but the assumptions are not realistic, the logit and probit model that assume that the failure rates across hospitals are similar in some way are more realistic and reliable. Between the logit and probit models the second one is a little bit preferable for this case. I also found out that in this case the best hospital where a baby should go to have cardiac surgery is the hospital, Nobody should ever take a baby to the hospital H!

In conclusion with this project I understood the importance that the random effects can have in a model and that they can be fundamental also in a frequentist approach.