

Applied Statistical Modelling - Assignment 2

Chiara Torri (1066761)

2023-01-12

First Model - Conjugate Case

In this first model the prior is conjugate, which means that the posterior have the same structure of the prior, with updated parameters. For this reason, since the sampling model is a Bernoulli, the prior of this case is a Beta with parameters α and β .

To perform the Bayesian analysis, it is firstly necessary to set the data and the relative sufficient statistics.

```
# Data
y=c(rep(0,16), rep(1,11))
n=length(y)
M=sum(y)
cat("The sample size is",n,"and the students that sleep more than 8 hours are",M)
```

```
## The sample size is 27 and the students that sleep more than 8 hours are 11
```

Case 1

Prior hyperparameters choice

The choice of the hyperparameters is guided, when possible, by the available information about the specific field of study. In this case the prior information is based on the literature presented in the “Prior Information” section of the assignment.

In particular, I set the hyperparameters such that:

- i. $E(\theta) = \alpha / (\alpha + \beta) = 0.3$
- ii. The 85th quantile of the $\text{Beta}(\alpha, \beta)$ is approximately 0.5, or, equivalently $P(\theta < 0.5) = 0.85$

```
# Prior hyperparameters
alpha= 0.3*5.55
beta= 5.55-alpha
cat("alpha=",alpha,"beta=",beta)
```

```
## alpha= 1.665 beta= 3.885
```

```
qbeta(0.85, alpha, beta)
```

```
## [1] 0.5000048
```

```
pbeta(0.5,alpha,beta)
```

```
## [1] 0.8499951
```

At this point, the model can be written as:

$$Y_1, \dots, Y_n | \theta \stackrel{\text{iid}}{\sim} \text{Bernoulli}(\theta)$$

$$\theta \sim \text{Beta}(1.665, 3.885)$$

Bayesian analysis

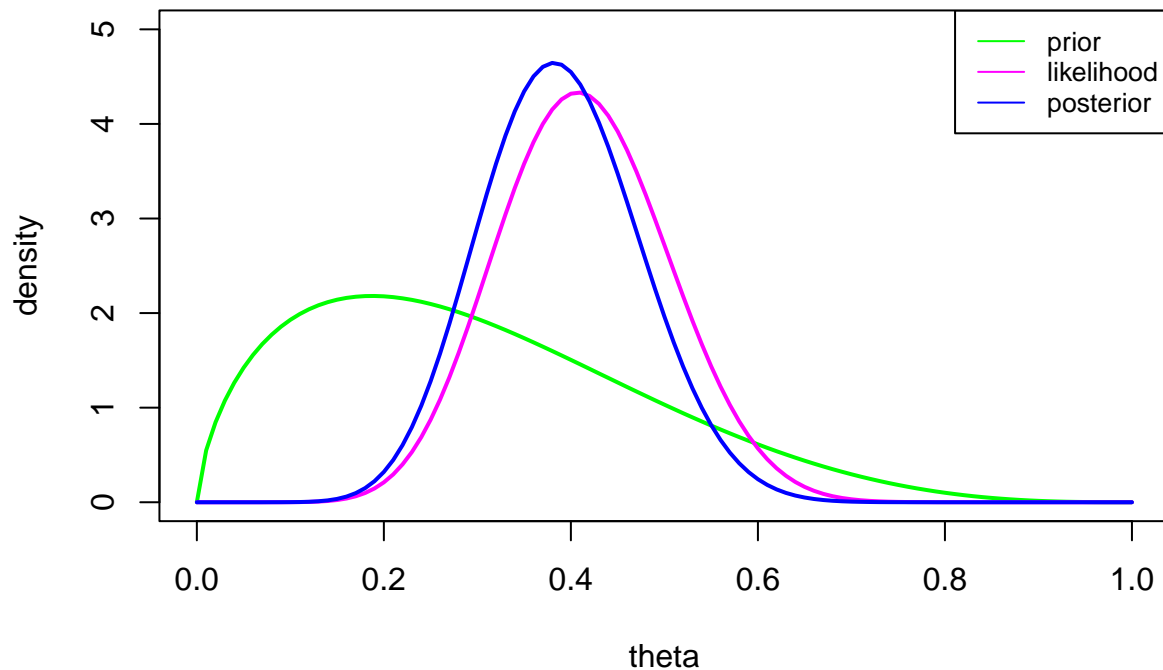
To perform a Bayesian analysis, I have to compute the posterior parameters which represent the way in which the prior hyperparameters are updated after the data are considered. For the Beta-Bernoulli model, the posterior parameters are the following.

```
# Posterior parameters
alpha_n= alpha+M
beta_n= beta+n-M
cat("alpha_n=",alpha_n,"beta_n=",beta_n)
```

```
## alpha_n= 12.665 beta_n= 19.885
```

It is now possible to graphically compare the prior, the likelihood and the posterior of this model.

- The prior is the density of a Beta with, as parameters, the hyperparameters.
- The likelihood looks like the kernel of a Beta(M+1,n-M+1), therefore plotting the density of a Beta with these parameters is equivalent to plot the normalized version of the likelihood.
- The posterior is the density of a Beta with updated parameters.



As expected, the posterior is between the prior and the likelihood.

The comparison between the prior and the posterior can also be conducted looking at the means, the variances and some relevant quantiles of the two distributions.

For what concerns the mean, it is possible to show that the posterior mean is between the prior mean and the sample average and, more precisely, it is a convex combination of the two:

$$E(\theta|y_{1:n}) = E(\theta) * \frac{\alpha}{\alpha+\beta} + \bar{y} * \frac{n}{\alpha+\beta+n}$$

```
# Mean
prior_mean=alpha/(alpha+beta)
post_mean= alpha_n/(alpha_n+beta_n)
sample_avg= mean(y)
cat("Prior mean:", prior_mean,"Posterior mean:",post_mean,"Sample average:",sample_avg)
```

```
## Prior mean: 0.3 Posterior mean: 0.3890937 Sample average: 0.4074074
```

The variances of the prior and the posterior are:

```
# Variance
prior_var= (alpha*beta)/((alpha+beta+1)*(alpha+beta)^2)
post_var=(alpha_n*beta_n)/((alpha_n+beta_n+1)*(alpha_n+beta_n)^2)
cat("Prior variance:", prior_var,"Posterior variance:",post_var)
```

```
## Prior variance: 0.03206107 Posterior variance: 0.007084942
```

The variance of the prior is higher than the variance of the posterior.

To further study the differences between the prior and the posterior, the 95% credible intervals are also relevant.

```
# Prior 95% credible interval
l0= qbeta(0.025, alpha, beta)
u0= qbeta(0.975, alpha, beta)
cat("The 95% credible interval of the prior is (",l0,",",u0,")")

## The 95% credible interval of the prior is ( 0.03444144 , 0.6970972 )
```

```
# Posterior 95% credible interval
l_n= qbeta(0.025, alpha_n, beta_n)
u_n= qbeta(0.975, alpha_n, beta_n)
cat("The 95% credible interval of the posterior is (",l_n,",",u_n,")")

## The 95% credible interval of the posterior is ( 0.2318016 , 0.5593008 )
```

As expected, since the variance is lower, the posterior credible interval is thinner.

Finally, I chose the prior hyperparameters basing on the assumption that with probability 0.85, θ is lower or equal than 0.5. For this reason, now I compute the 85th quantile of the posterior.

```
# 85th quantile
prior_85q=qbeta(0.85,alpha,beta)
post_85q=qbeta(0.85,alpha_n,beta_n)
cat("Prior 85th quantile:", prior_85q,", Posterior 85th quantile:",post_85q)
```

```
## Prior 85th quantile: 0.5000048 , Posterior 85th quantile: 0.4777214
```

The 85th quantile of the posterior is lower than the prior ones.

Prediction

The goal is to predict the number of students that get at least eight hours of sleep if a new sample of 20 students is taken.

The prediction of a new Y based on the previous observation can be seen as a posterior mean.

Since the data are modeled as dependent, indeed, the new y will depend on the previous $y_{1:n}$ and their dependence is based on θ . In particular, observing y_1, \dots, y_n gives information about θ , which in turn gives information about y_{n+1} .

For this reason, the predictive density $m(y_{n+1}|y_{1:n})$ can be seen as the posterior expectation of the sampling model $p(y_{n+1}|\theta)$, where θ (the posterior) is built basing on the data $y_{1:n}$.

The posterior predictive for a Bernoulli-Beta model is $\frac{(\alpha+M)}{(\alpha+\beta+n)}$, that is $Pr(Y_{n+1} = 1|y_{1:n}) = \frac{(\alpha+M)}{(\alpha+\beta+n)}$. Repeating this reasoning for all the 20 new students, we get the number of students of the new sample that

get at least 8 hours of sleep.

It is also possible to make an example of the new sample of 20 students.

```
# Posterior predictive
post_predictive=(alpha+M)/(alpha+beta+n)
cat("The posterior predictive is", post_predictive)

## The posterior predictive is 0.3890937

cat("The posterior predictive for the 20 observations of the new sample is", post_predictive*20)

## The posterior predictive for the 20 observations of the new sample is 7.781874

# Example of the new sample
set.seed(123)
y_new= rbinom(20,1,post_mean)
cat("The new sample is", y_new)

## The new sample is 0 1 0 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1

cat("Students of the sample that sleep at least 8 hours:",sum(y_new))

## Students of the sample that sleep at least 8 hours: 8
```

The posterior predictive is 0.3890937 and basing on this I can predict that the students that sleep at least 8 hours are $7.781874 \approx 8$ out of 20. The result is also confirmed in the example of new sample presented above, in which the number of students that get at least 8 hours of sleep is 8, as expected.

Hypothesis testing

In this next section the objective is to verify if the event “less than 50% of students get at least eight hours of sleep during a week day” is supported by the data or if its alternative event is.

Formally, the two events can be expressed as:

$$E_0 := \theta \leq 0.5 \text{ vs } E_1 := \theta > 0.5$$

To study which of the two events is supported by the data, I will consider two quantities:

- i. the posterior probability of E_0 ;
- ii. the Bayes Factor, which is the ratio between the posterior and prior odds of E_0 .

```
# Pr(E0|y_1:n)
post_prob0= pbeta(0.5,alpha_n,beta_n)
cat("The posterior probability of E0 is", post_prob0)
```

```
## The posterior probability of E0 is 0.901043
```

```

# Pr(E1|y_1:n)
post_prob1=1-pbeta(0.5,alpha_n,beta_n)

# Pr(E0)
prior_prob0=pbeta(0.5,alpha,beta)

# Pr(E1)
prior_prob1=1-pbeta(0.5,alpha,beta)

# Bayes factor
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
cat("The Bayes Factor is",BF)

```

```
## The Bayes Factor is 1.606898
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.901043$. This probability is higher than the prior probability, that was, by assumption, about 0.85.

The Bayes Factor is 1.606898, which is low but higher than 1. This means, following the table reported in the assignment, that the strength of data evidence in favor of E_0 is “not worth more than just a bare mention”. However, there is no evidence in favor of E_1 . The small value of the Bayes factor is probably due to the fact that, for what concern this probability, the prior and the posterior are quite similar and hence the data do not affect the choice of accepting or rejecting the hypothesis that much.

Case 2

Prior hyperparameters choice

In this case, the parameters have to reflect complete ignorance about the parameter θ . To this end, I choose the hyperparameters such that:

- i. The prior proportion of students that sleep more and less than 8 hours is the same, $E(\theta) = \alpha/(\alpha + \beta) = 0.5$;
- ii. The sum $\alpha + \beta$, which represents the certainty of the prior, is very low.

```

# Prior hyperparameters
alpha= 0.5*0.01
beta= 0.01-alpha
cat("alpha=",alpha,"beta=",beta)

```

```
## alpha= 0.005 beta= 0.005
```

```
qbeta(0.85, alpha, beta)
```

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

```
pbeta(0.5,alpha,beta)
```

```
## [1] 0.5
```

At this point, the model can be written as:

$$Y_1, \dots, Y_n | \theta \stackrel{\text{iid}}{\sim} \text{Bernoulli}(\theta)$$
$$\theta \sim \text{Beta}(0.005, 0.005)$$

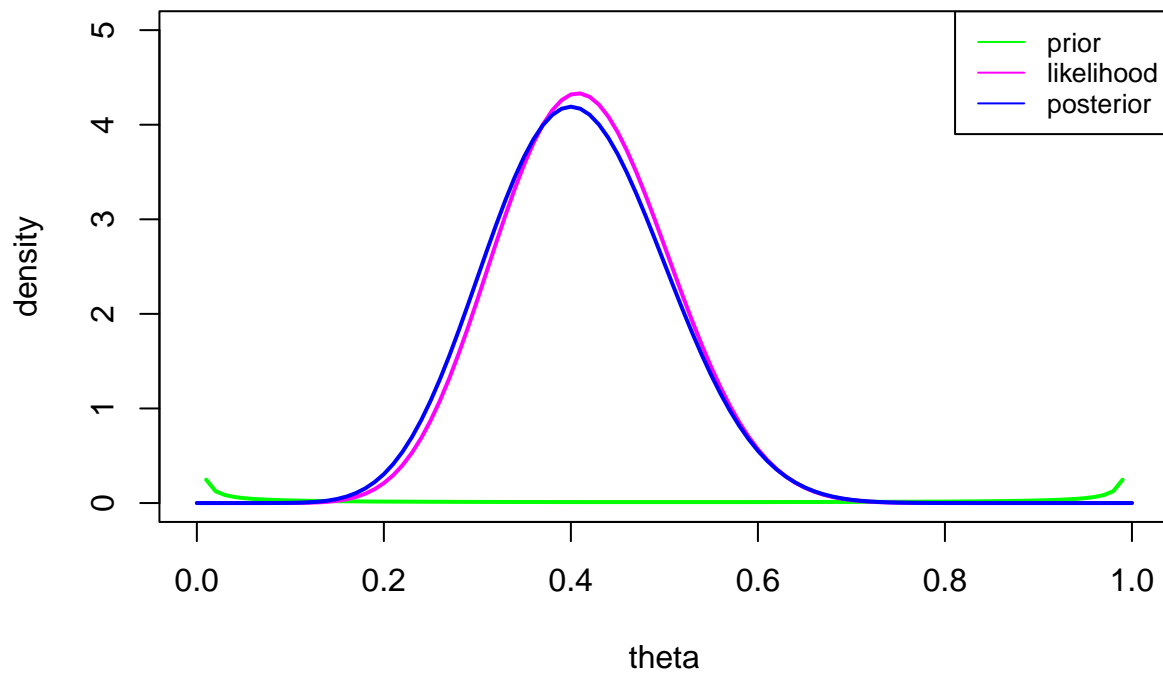
Bayesian analysis

Also in this case, I need to compute the posterior parameters, that are computed as before:

```
# Posterior parameters  
alpha_n= alpha+M  
beta_n= beta+n-M  
cat("alpha_n=",alpha_n,"beta_n=",beta_n)
```

```
## alpha_n= 11.005 beta_n= 16.005
```

It is now possible to plot the prior, the likelihood and the posterior, for a graphical comparison. The three curves are built as in the previous case.



Although it appears less clearly in the plot, the posterior is still between the prior and the likelihood. This time, however, it is much more similar to the likelihood, because in the choice of the hyperparameters, I set the prior as not very informative to reflect ignorance about θ .

Proceeding in the analysis, it is useful to consider some relevant quantities of the posterior and their comparison to the ones of the prior.

The posterior mean is, also in this case, a convex combination of the prior mean and the sample average:

$$E(\theta|y_{1:n}) = E(\theta) * \frac{\alpha}{\alpha+\beta} + \bar{y} * \frac{n}{\alpha+\beta+n}$$

```
# Mean
prior_mean=alpha/(alpha+beta)
post_mean= alpha_n/(alpha_n+beta_n)
sample_avg= mean(y)
cat("Prior mean:", prior_mean,"Posterior mean:",post_mean,"Sample average:",sample_avg)
```

```
## Prior mean: 0.5 Posterior mean: 0.4074417 Sample average: 0.4074074
```

The posterior mean is very similar to the sample average in this case, always for the reason that the prior has been chosen as not very informative.

The variances of the prior and the posterior are:

```
# Variance
prior_var= (alpha*beta)/((alpha+beta+1)*(alpha+beta)^2)
post_var=(alpha_n*beta_n)/((alpha_n+beta_n+1)*(alpha_n+beta_n)^2)
cat("Prior variance:", prior_var,"Posterior variance:",post_var)
```

```
## Prior variance: 0.2475248 Posterior variance: 0.008619527
```

Also here, the variance of the prior is higher than the variance of the posterior.

The 95% credible intervals of the prior and the posterior are the following.

```
# Prior 95% credible interval
l0= qbeta(0.025, alpha, beta)
u0= qbeta(0.975, alpha, beta)
cat("The 95% credible interval of the prior is (",l0,",",u0,")")
```

```
## The 95% credible interval of the prior is ( 6.172411e-261 , 1 )
```

```
# Posterior 95% credible interval
l_n= qbeta(0.025, alpha_n, beta_n)
u_n= qbeta(0.975, alpha_n, beta_n)
cat("The 95% credible interval of the posterior is (",l_n,",",u_n,")")
```

```
## The 95% credible interval of the posterior is ( 0.2335806 , 0.5942928 )
```


In this case the difference between the width of the two intervals is much more visible and this is again due to the choice of the prior.

Even if it was not taken into consideration when choosing the hyperparameters, also in this case I decided to include the 85th quantile, to allow a comparison between the three cases of the conjugate model.

```
# 85th quantile
prior_85q=qbeta(0.85,alpha,beta)
post_85q=qbeta(0.85,alpha_n,beta_n)
cat("Prior 85th quantile:", prior_85q," , Posterior 85th quantile:",post_85q)
```

```
## Prior 85th quantile: 1 , Posterior 85th quantile: 0.5054849
```

The 85th quantiles of the two distributions are very different and the difference is wider than in Case 1.

Prediction

The goal is still to predict the number of students that get at least eight hours of sleep if a new sample of 20 students is taken.

The reasoning and the procedure I used are the same of Case 1.

```
# Posterior predictive
post_predictive=(alpha+M)/(alpha+beta+n)
cat("The posterior predictive is", post_predictive)
```

```
## The posterior predictive is 0.4074417
```

```
cat("The posterior predictive for the 20 observations of the new sample is", post_predictive*20)
```

```
## The posterior predictive for the 20 observations of the new sample is 8.148834
```

```
# Example of the new sample
set.seed(123)
y_new= rbinom(20,1,post_mean)
cat("The new sample is", y_new)
```

```
## The new sample is 0 1 0 1 1 0 0 1 0 0 1 0 1 0 0 1 0 0 0 1
```

```
cat("Students of the sample that sleep at least 8 hours:",sum(y_new))
```

```
## Students of the sample that sleep at least 8 hours: 8
```

In this case the posterior predictive is 0.4074417, which is slightly higher than the previous one, due to the minor influence of the prior on the posterior. For this reason, the predicted number of students of the new sample of 20 that sleep at least 8 hours is slightly higher: 8.148834, but still approximately equal to 8.

The example of the new sample gives the same result.

Hypothesis testing

Also in this case, the objective is to verify which of this two alternative events is supported by the data:
 $E_0 := \theta \leq 0.5$ vs $E_1 := \theta > 0.5$

I will still consider the two quantities described above: the posterior probability and the Bayes Factor.

```
# Pr(E0|y_1:n)
post_prob0= pbeta(0.5,alpha_n,beta_n)
cat("The posterior probability of E0 is", post_prob0)
```

```
## The posterior probability of E0 is 0.8364835
```

```
# Pr(E1|y_1:n)
post_prob1=1-pbeta(0.5,alpha_n,beta_n)

# Pr(E0)
prior_prob0=pbeta(0.5,alpha,beta)

# Pr(E1)
prior_prob1=1-pbeta(0.5,alpha,beta)

# Bayes factor
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
cat("The Bayes Factor is",BF)
```

```
## The Bayes Factor is 5.115589
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.8364835$. This probability is lower than the posterior probability of Case 1, as a result of the different choice of the prior. Specifically, in the first case I set $E(\theta) = 0.3$, whereas in the second case I set $E(\theta) = 0.5$, and this affected the result of the test.

The Bayes Factor is 5.115589, which means that the strength of data evidence in favor of E_0 is now “substantial”. This increased evidence in favor of E_0 is a result of the increased difference between the prior and the posterior odds, meaning that the data support E_0 much more than the prior does. This is once again to impute to the “complete ignorance” hyperparameters choice.

Case 3

Prior hyperparameters choice

In this third case, the hyperparameters have to reflect different prior beliefs: the best guess of the value of θ is now 0.7 and, with probability 0.85, θ is bigger than 0.5.

Consequently, I set the hyperparameters such that:

- i. $E(\theta) = \alpha/(\alpha + \beta) = 0.7$
- ii. The 15th quantile of the $\text{Beta}(\alpha, \beta)$ is approximately 0.5, or, equivalently $P(\theta > 0.5) = 1 - P(\theta \leq 0.5) = 0.85$

```
# Prior hyperparameters
alpha= 0.7*5.55
beta= 5.55-alpha
cat("alpha=",alpha,"beta=",beta)
```

```
## alpha= 3.885 beta= 1.665
```

```
qbeta(0.15, alpha, beta)
```

```
## [1] 0.4999952
```

```
1-pbeta(0.5,alpha,beta)
```

```
## [1] 0.8499951
```

The parameters are inverted with respect to Case 1.

At this point, the model can be written as:

$$Y_1, \dots, Y_n | \theta \stackrel{\text{iid}}{\sim} \text{Bernoulli}(\theta)$$

$$\theta \sim \text{Beta}(3.885, 1.665)$$

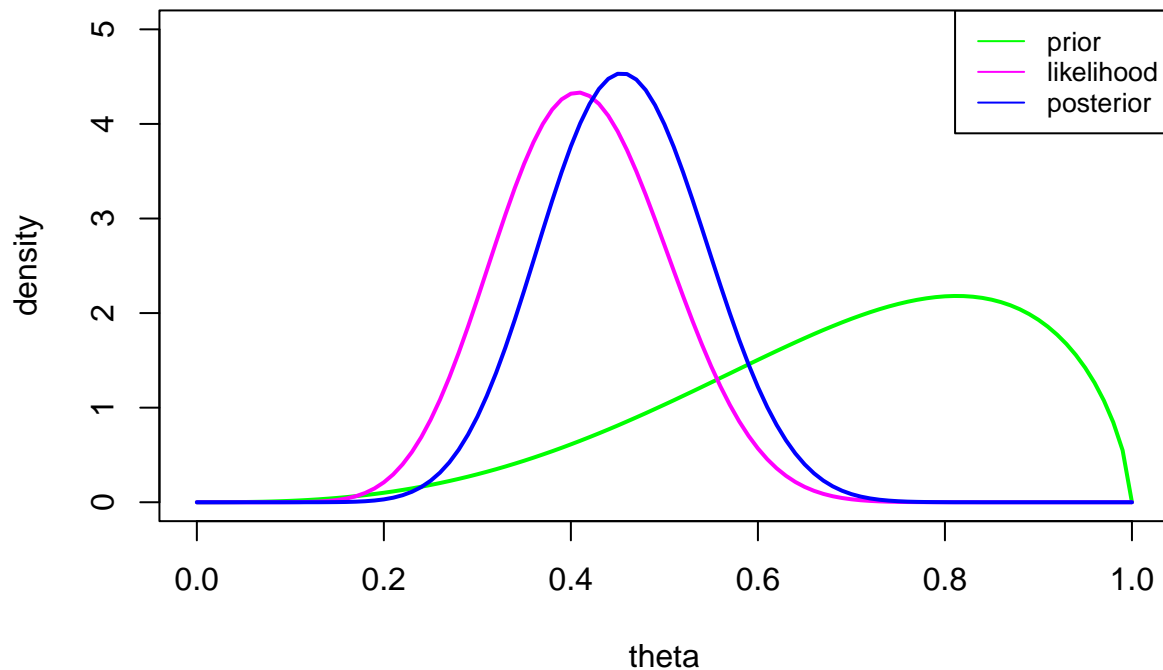
Bayesian analysis

Also in this case, I need to compute the posterior parameters, that are computed as before:

```
# Posterior parameters
alpha_n= alpha+M
beta_n= beta+n-M
cat("alpha_n=",alpha_n,"beta_n=",beta_n)
```

```
## alpha_n= 14.885 beta_n= 17.665
```

It is now possible to plot the prior, the likelihood and the posterior, for a graphical comparison. The three curves are built as in the previous case.



Also in Case 3, the posterior is between the prior and the likelihood.

In the next section, I am going to look at some relevant quantities of the prior and the posterior distribution, as previously done for Cases 1 and 2.

The posterior mean is, also in this case, a convex combination of the prior mean and the sample average:

$$E(\theta|y_{1:n}) = E(\theta) * \frac{\alpha}{\alpha+\beta} + \bar{y} * \frac{n}{\alpha+\beta+n}$$

```
# Mean
prior_mean=alpha/(alpha+beta)
post_mean= alpha_n/(alpha_n+beta_n)
sample_avg= mean(y)
cat("Prior mean:", prior_mean,"Posterior mean:",post_mean,"Sample average:",sample_avg)
```

```
## Prior mean: 0.7 Posterior mean: 0.4572965 Sample average: 0.4074074
```

The posterior mean is between the prior mean and the sample average and in this case is higher than in the previous two, as a result of the effect of the higher prior expected value.

The variances of the prior and the posterior are:

```
# Variance
prior_var= (alpha*beta)/((alpha+beta+1)*(alpha+beta)^2)
post_var=(alpha_n*beta_n)/((alpha_n+beta_n+1)*(alpha_n+beta_n)^2)
cat("Prior variance:", prior_var,"Posterior variance:",post_var)
```

```
## Prior variance: 0.03206107 Posterior variance: 0.00739721
```

Also here, the variance of the prior is higher than the variance of the posterior. I notice that the prior variance of this case is the same of the prior variance in Case 1, whereas the two posterior variances are different.

The 95% credible intervals of the prior and the posterior are the following.

```
# Prior 95% credible interval
l0= qbeta(0.025, alpha, beta)
u0= qbeta(0.975, alpha, beta)
cat("The 95% credible interval of the prior is (",l0,",",u0,")")
```

```
## The 95% credible interval of the prior is ( 0.3029028 , 0.9655586 )
```

```
# Posterior 95% credible interval
l_n= qbeta(0.025, alpha_n, beta_n)
u_n= qbeta(0.975, alpha_n, beta_n)
cat("The 95% credible interval of the posterior is (",l_n,",",u_n,")")
```

```
## The 95% credible interval of the posterior is ( 0.2923892 , 0.6271787 )
```

As in Cases 1 and 2, the width of the prior credible interval is lower than the one of the posterior credible interval.

Similarly to the reasoning that I applied in Case 1, also here I am interested in comparing the prior and posterior quantiles on which I based the choice of prior hyperparameter. In this case, the relevant quantile is the 15th.

```
# 15th quantiles
prior_15q=qbeta(0.15,alpha,beta)
post_15q=qbeta(0.15,alpha_n,beta_n)
cat("Prior 15th quantile:", prior_15q," Posterior 15th quantile:",post_15q)
```

```
## Prior 15th quantile: 0.4999952 , Posterior 15th quantile: 0.3669565
```

The posterior 15th, as expected, is lower than the prior one, that was, by assumption, approximately equal to 0.5.

Prediction

The goal is still to predict the number of students that get at least eight hours of sleep if a new sample of 20 students is taken.

The reasoning and the procedure I used are the same of Case 1 and 2.

```
# Posterior predictive
post_predictive=(alpha+M)/(alpha+beta+n)
cat("The posterior predictive is", post_predictive)
```

```
## The posterior predictive is 0.4572965
```

```
cat("The posterior predictive for the 20 observations of the new sample is", post_predictive*20)
```

```
## The posterior predictive for the 20 observations of the new sample is 9.145929
```

```
# Example of the new sample
set.seed(123)
y_new= rbinom(20,1,post_mean)
cat("The new sample is", y_new)
```

```
## The new sample is 0 1 0 1 1 0 0 1 1 0 1 0 1 1 0 0 0 1
```

```
cat("Students of the sample that sleep at least 8 hours:",sum(y_new))
```

```
## Students of the sample that sleep at least 8 hours: 10
```

In this case the posterior predictive is 0.4572965, which is higher than the previous two, due to the influence of the higher prior mean. For this reason, the predicted number of students of the new sample of 20 that sleep at least 8 hours is higher: $9.145929 \approx 9$.

The example of the new sample gives, indeed, different results than in the first two Cases: 10 students out of 20 sleep at least 8 hours.

These examples show clearly how the prior beliefs affect the result of a Bayesian analysis.

Hypothesis testing

Also in this case, the objective is to verify which of this two alternative events is supported by the data: $E_0 := \theta \leq 0.5$ vs $E_1 := \theta > 0.5$

I will still consider the two quantities described above: the posterior probability and the Bayes Factor.

```
# Pr(E0|y_1:n)
post_prob0= pbeta(0.5,alpha_n,beta_n)
cat("The posterior probability of E0 is", post_prob0)
```

```
## The posterior probability of E0 is 0.6893061
```

```
# Pr(E1|y_1:n)
post_prob1=1-pbeta(0.5,alpha_n,beta_n)
```

```
# Pr(E0)
prior_prob0=pbeta(0.5,alpha,beta)
```

```
# Pr(E1)
prior_prob1=1-pbeta(0.5,alpha,beta)
```

```
# Bayes factor
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
cat("The Bayes Factor is",BF)
```

```
## The Bayes Factor is 12.57159
```

```
rm(list = ls())
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.6893061$. This probability is lower than the posterior probability of Case 1 and 2, as a result of the different choice of the prior mean. Specifically, in the first case I set $E(\theta) = 0.3$, in the second case I set $E(\theta) = 0.5$ and in the third case I set $E(\theta) = 0.7$ and this shifted the posterior distribution to the right.

The Bayes Factor is 12.57159, which means that the strength of data evidence in favor of E_0 is now “strong”. This increased evidence in favor of E_0 is, as in case 2, a result of the increased difference between the prior and the posterior odds.

In particular, in this third case, the data supported more E_0 (the sample average was $0.4074074 < 0.5$), whereas the prior supported more E_1 (the prior mean was $0.7 > 0.5$).

Second model - Discrete prior

In this second model we assume that θ is discrete and that it can assume the following values: $\theta \in \Theta = \{\tau_1, \dots, \tau_9\} = \{0.1, \dots, 0.9\}$. The probability mass function of θ is: $\pi_2(\tau_j) = p_j$ $j = 1, \dots, 9$.

To begin my analysis, I set the data and the relative sufficient statistics.

```
# Data
y=c(rep(0,16), rep(1,11))
n=length(y)
M=sum(y)
cat("The sample size is",n,"and the students that sleep more than 8 hours are",M)
```

```
## The sample size is 27 and the students that sleep more than 8 hours are 11
```

Case 1

Prior choice

In this first case the prior has to reflect the information presented in the “Prior Information” section. To this end, I set the values $\{p_j\}$ such that the expected value of θ is 0.3 (which corresponds to the best guess) and the probability that θ is lower or equal than 0.5 is 0.85.

To verify this last condition, I also set the cumulative distribution function of θ .

```
# Choice of the prior
theta=seq(from=0.1,to=0.9,by=0.1)
cat("Theta:", theta)
```

```
## Theta: 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

```
prior=c(0.33,0.20,0.16,0.09,0.07,0.06,rep(0.03,3))
cat("Prior:", prior)
```

```
## Prior: 0.33 0.2 0.16 0.09 0.07 0.06 0.03 0.03 0.03
```

```
cat("The sum of the prior probabilities is", sum(prior))
```

```
## The sum of the prior probabilities is 1
```

```
# Cdf of the prior
```

```
cump=c()
```

```
cdf=function(p){
```

```
  for (j in 1:9){
```

```
    cump[j]=sum(p[1:j])
```

```
  }
```

```
  return(cump)
```

```
}
```

```
cumprior=cdf(prior)
```

```
cat("Cumulative distribution function of the prior:", cumprior)
```

```
## Cumulative distribution function of the prior: 0.33 0.53 0.69 0.78 0.85 0.91 0.94 0.97 1
```

```
# P(theta<=0.5)=0.85 and E(theta)=0.3
```

```
cat("P(theta<=0.5)=", cumprior[5])
```

```
## P(theta<=0.5)= 0.85
```

```
prior_mean=weighted.mean(theta,prior)
```

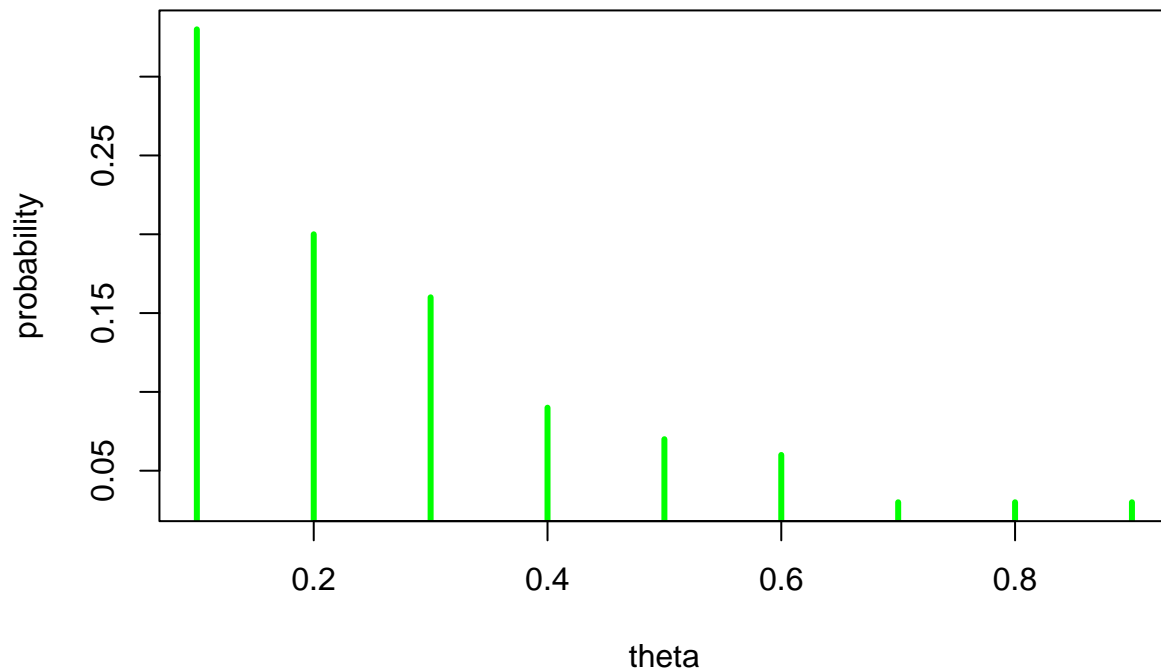
```
cat("E(theta)=", prior_mean)
```

```
## E(theta)= 0.3
```

```
# Plot of the prior
```

```
plot(theta, prior, main="Prior", type="h", xlab="theta", ylab="probability",  
      col="green", lwd=3)
```


Prior



Bayesian analysis

In order to perform my Bayesian analysis, I have now to compute the posterior. In particular, I know that the posterior is equal to $(likelihood * prior) / marginal$ and it is proportional to $likelihood * prior$.

Moreover, I know from the model that the likelihood coincides with the kernel of a $Beta(M+1, n-M+1)$.

Finally, the marginal is the integral of $likelihood * prior$, which in this discrete case coincides with the sum of this product.

```
# Posterior
prop_post=c()
like=c()

for (j in 1:9){
  like[j]=(theta[j])^M*(1-theta[j])^(n-M)
  prop_post[j]=((theta[j])^M*(1-theta[j])^(n-M)*prior[j])
}

sum(prop_post) # not equal to 1 because I still have to divide for the marginal
```

```
## [1] 2.740378e-09
```

```
post=prop_post/sum(prop_post)
cat("Posterior:", post)
```

```
## Posterior: 0.0002231432 0.04207162 0.3437256 0.3886081 0.190317 0.03411649 0.000931814 6.162835e-06
```

```
cat("The sum of the posterior probabilities is", sum(post)) # equal to 1!
```

```
## The sum of the posterior probabilities is 1
```

Together with the posterior, I also compute the likelihood and a normalized version of it, that will be useful for the upcoming plots.

```
# Normalized version of the likelihood
cat("Likelihood:", like)
```

```
## Likelihood: 1.85302e-12 5.764608e-10 5.887114e-09 1.183259e-08 7.450581e-09 1.558201e-09 8.511743e-10
```

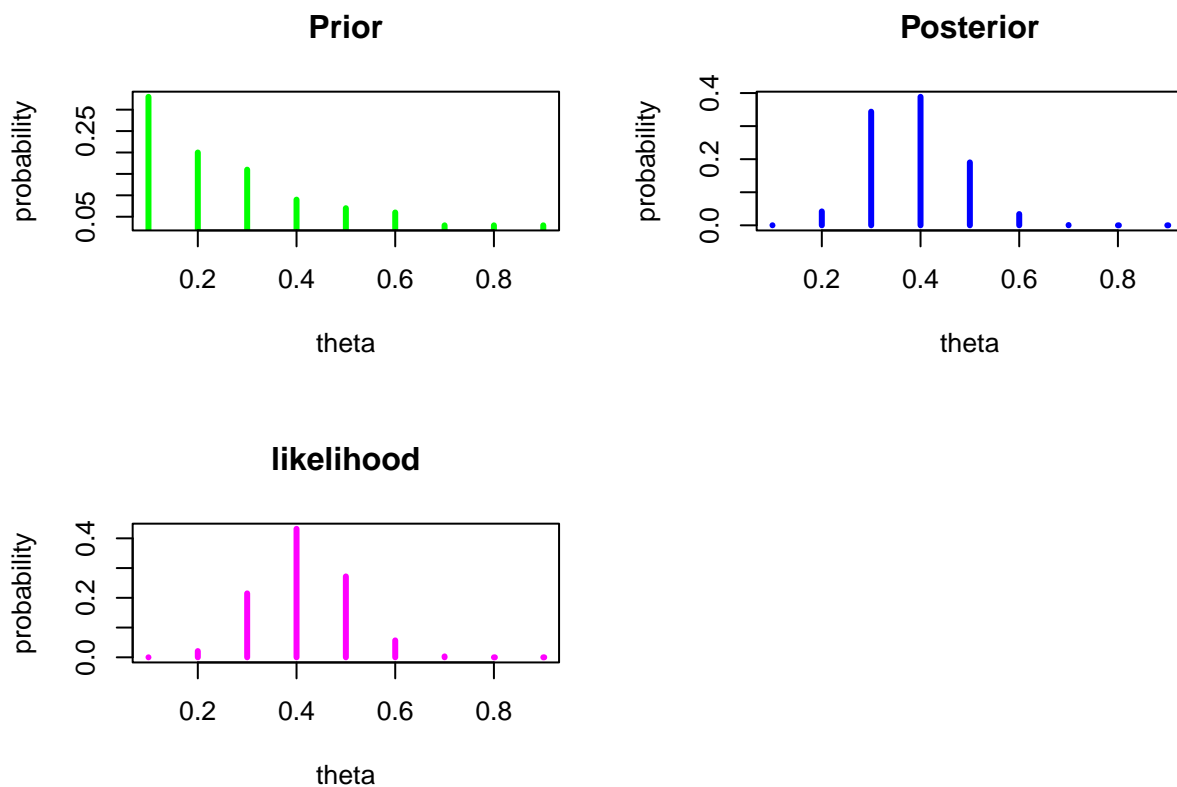
```
like2=like/sum(like)
cat("Normalized version of the likelihood:", like2)
```

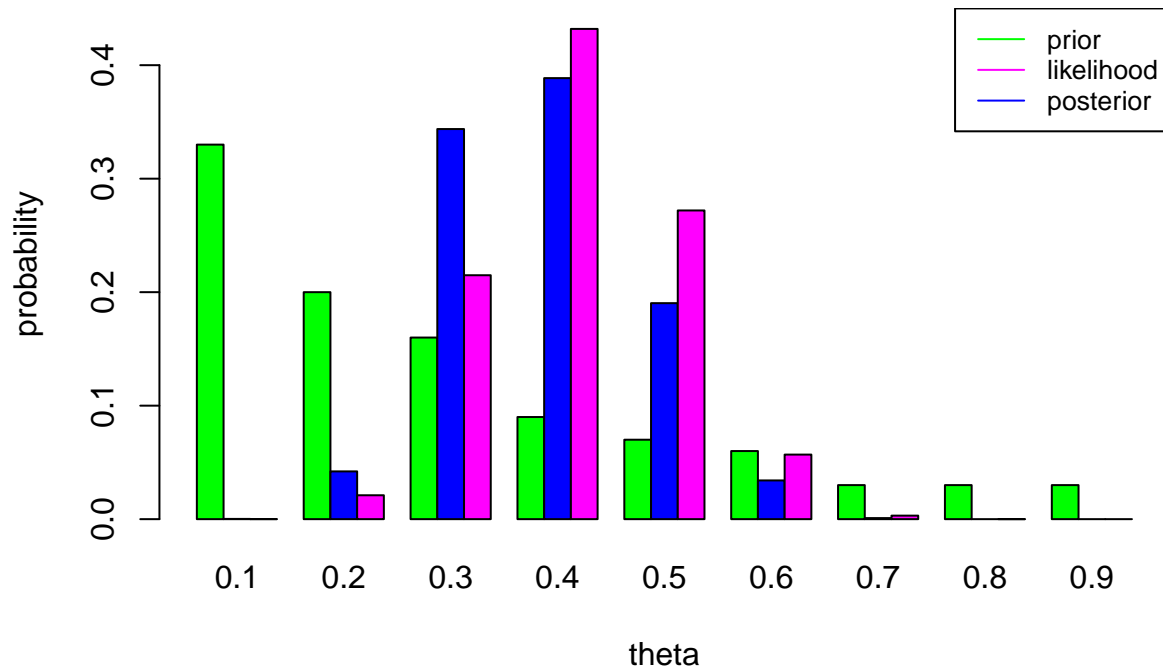
```
## Normalized version of the likelihood: 6.764703e-05 0.02104449 0.2149171 0.431965 0.2719936 0.0568842
```

```
cat("The sum of the normalized likelihood probabilities is", sum(like2)) # equal to 1!
```

```
## The sum of the normalized likelihood probabilities is 1
```

At this point, I am able to start my Bayesian analysis, by plotting the prior, the likelihood and the posterior distributions.





I proceed computing some relevant quantities of the posterior and comparing them with those of the prior.

For what concern the posterior mean, as expected, I find that it is between the prior mean and the sample average, as in the previous cases.

```
# Posterior mean
post_mean=weighted.mean(theta,post)
sample_avg=mean(y)
cat("Prior mean:", prior_mean, ", Posterior mean:", post_mean,
    ", Sample average:", sample_avg)
```

```
## Prior mean: 0.3 , Posterior mean: 0.3832832 , Sample average: 0.4074074
```

Regarding the variances, I notice that once again the prior variance is higher than the posterior variance.

```
# Posterior variance
prior_var=weighted.mean((theta-prior_mean)^2, prior)
post_var=weighted.mean((theta-post_mean)^2, post)
cat("Prior variance:", prior_var, ", Posterior variance:", post_var)
```

```
## Prior variance: 0.0474 , Posterior variance: 0.008213431
```

Moving on, I compute the 95% credible intervals for both the prior and the posterior. To this end, I first set the cumulative distribution function of the posterior and then I build a function that computes the quantiles in this discrete setting.

```

# Quantiles

# Posterior cdf
cumpost=cdf(post)
cat("Cumulative distribution function of the posterior:", cumpost)

## Cumulative distribution function of the posterior: 0.0002231432 0.04229476 0.3860204 0.7746285 0.964

# Discrete quantile function
my_quant=function(cfd, p){
  indx=which(cfd>=p)
  return(theta[indx[1]])
}

```

At this point I can compute the two credible intervals and I find that the posterior credible interval is narrower compared to the prior one.

```

# Prior 95% credible interval
l0=my_quant(cumprior, 0.025)
u0=my_quant(cumprior, 0.975)
cat("The 95% credible interval of the prior is (",l0,",",u0,")")

```

```
## The 95% credible interval of the prior is ( 0.1 , 0.9 )
```

```

# Posterior 95% credible interval
l_n=my_quant(cumpost, 0.025)
u_n=my_quant(cumpost, 0.975)
cat("The 95% credible interval of the posterior is (",l_n,",",u_n,")")

```

```
## The 95% credible interval of the posterior is ( 0.2 , 0.6 )
```

To conclude, I compute the 85th quantile, since I based my prior choice also on this feature. I notice that the prior and the posterior 85th quantiles are equal.

```

# 85th quantile
prior_85q=my_quant(cumprior, 0.85)
post_85q=my_quant(cumpost, 0.85)
cat("Prior 85th quantile:", prior_85q,", Posterior 85th quantile:",post_85q)

```

```
## Prior 85th quantile: 0.5 , Posterior 85th quantile: 0.5
```

Prediction

The goal of this section is to estimate the number of students that sleep at least 8 hours when a new sample of 20 students is considered. Consequently, I have to compute the posterior predictive and, as in Model 1, I exploit the fact that the posterior predictive can be seen as the posterior mean.

In particular the predictive is $\int m(y_{new}, \theta | y_{1:n}) d\theta$. In this discrete case, it is possible to show that this coincides with:

$$\sum p(y_{new} | \theta) * p(\theta | y_{1:n}) = \sum \theta * p(\theta | y_{1:n}) = E(p(y_{new} | \theta) | y_{1:n}).$$

Therefore, the posterior predictive is equal to the posterior mean.

```
# Prediction
cat("The psterior predictive is:", post_mean)

## The psterior predictive is: 0.3832832

cat("The posterior predictive for the 20 observations of the new sample is", post_mean*20)
```

```
## The posterior predictive for the 20 observations of the new sample is 7.665664
```

The posterior predictive is 0.3832832 and the number of students in the new sample that sleep at least 8 hours is $7.665664 \approx 8$.

I now create an example of this new sample.

```
# Example of the new sample
set.seed(123)
y_new=rbinom(20,1,prob=post_mean)
y_new

## [1] 0 1 0 1 1 0 0 1 0 0 1 0 1 0 0 1 0 0 0 1

cat("Students of the sample that sleep at least 8 hours:", sum(y_new))
```

```
## Students of the sample that sleep at least 8 hours: 8
```

In this sample, the number of students that sleep at least 8 hours is 8, as in the previous result.

Hypothesis testing

In this next section the objective is to verify if the event “less than 50% of students get at least eight hours of sleep during a week day” is supported by the data or if its alternative event is.

Formally: $E_0 := \theta \leq 0.5$ vs $E_1 := \theta > 0.5$

```
# Hypothesis testing

# Pr(E0|y_1:n)
post_prob0= cumpost[5]
cat("The posterior probability of E0 is", post_prob0)

## The posterior probability of E0 is 0.9649455

# Pr(E1|y_1:n)
post_prob1= 1-post_prob0

# Pr(E0)
prior_prob0= cumprior[5]
```

```
# Pr(E1)
prior_prob1= 1-prior_prob0

# Bayes factor
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
cat("The Bayes Factor is",BF)
```

```
## The Bayes Factor is 4.857712
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.9649455$. This probability is higher than the prior probability, that was, by assumption, approximately 0.85.

The Bayes Factor is 4.857712, which indicates that the strength of data evidence in favor of E_0 is “substantial”.

Case 2

Prior choice

In this second case, the prior has to represent complete ignorance about the parameter θ . To this aim, I set the same probability $\{p_j\} = \frac{1}{9}$ for each τ_j .

```
# Choice of the prior
theta=seq(from=0.1,to=0.9,by=0.1)
cat("Theta:", theta)
```

```
## Theta: 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

```
prior=rep(1/9, 9)
cat("Prior:", prior)
```

```
## Prior: 0.1111111 0.1111111 0.1111111 0.1111111 0.1111111 0.1111111 0.1111111 0.1111111 0.1111111
```

```
cat("The sum of the prior probabilities is", sum(prior))
```

```
## The sum of the prior probabilities is 1
```

```
# Cdf of the prior
cumprior=cdf(prior)
cat("Cumulative distribution function of the prior:", cumprior)
```

```
## Cumulative distribution function of the prior: 0.1111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667 0.7777778 0.8888889 1.0000000
```

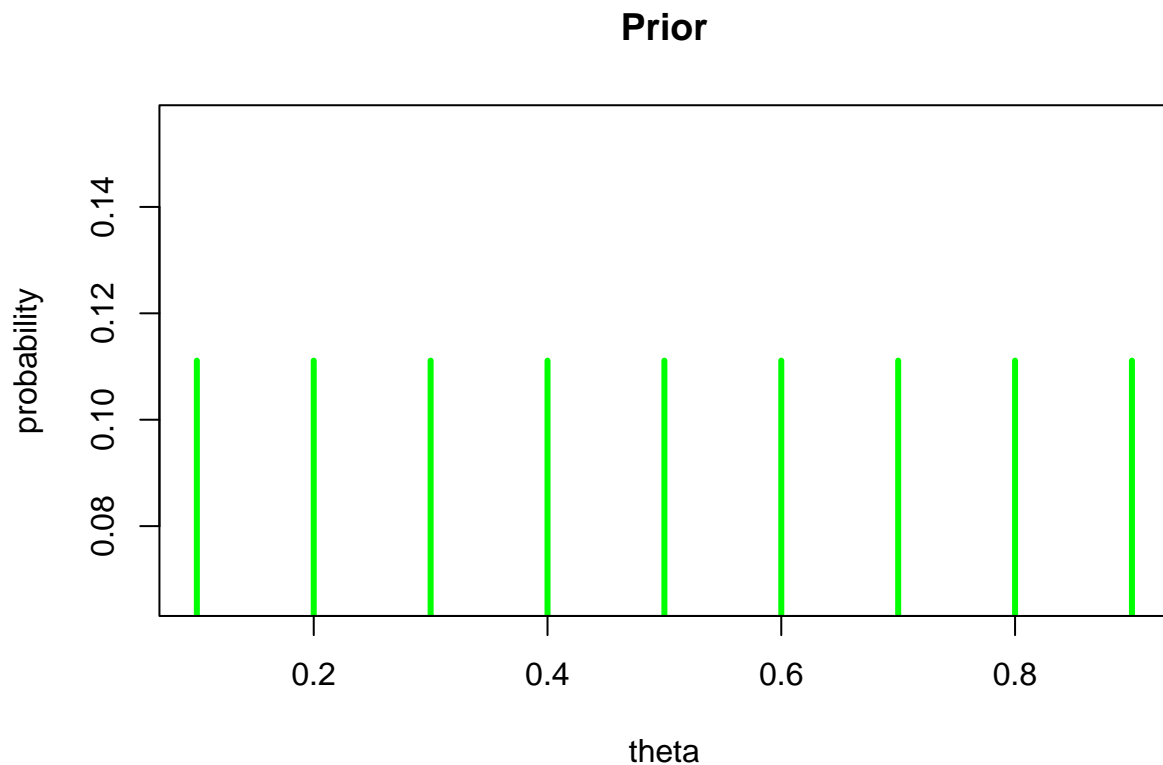
```
# P(theta<=0.5) and E(theta)=0.5
cat("P(theta<=0.5)=", cumprior[5])
```

```
## P(theta<=0.5)= 0.5555556
```

```
prior_mean=weighted.mean(theta,prior)
cat("E(theta)=", prior_mean)
```

```
## E(theta)= 0.5
```

```
# Plot of the prior
plot(theta, prior, main="Prior", type="h", xlab="theta", ylab="probability",
      col="green", lwd=3)
```



With this prior, the probability that θ is lower or equal to 0.5 is 0.5555556 and the expected value of θ is 0.5.

Bayesian analysis

I now compute the posterior using the same reasoning I used in Case 1.

```
# Posterior
prop_post=c()
like=c()

for (j in 1:9){
  like[j]=(theta[j])^M*(1-theta[j])^(n-M)
  prop_post[j]=((theta[j])^M*(1-theta[j])^(n-M)*prior[j])
}
```

```
}
```

```
sum(prop_post) # not equal to 1 because I still have to divide for the marginal
```

```
## [1] 3.043609e-09
```

```
post=prop_post/sum(prop_post)
cat("Posterior:", post)
```

```
## Posterior: 6.764703e-05 0.02104449 0.2149171 0.431965 0.2719936 0.05688427 0.003107328 2.055126e-05
```

```
cat("The sum of the posterior probabilities is", sum(post)) # equal to 1!
```

```
## The sum of the posterior probabilities is 1
```

Also in this case, I compute a normalized version of the likelihood for the next plots.

```
# Normalized version of the likelihood
cat("Likelihood:", like)
```

```
## Likelihood: 1.85302e-12 5.764608e-10 5.887114e-09 1.183259e-08 7.450581e-09 1.558201e-09 8.511743e-10
```

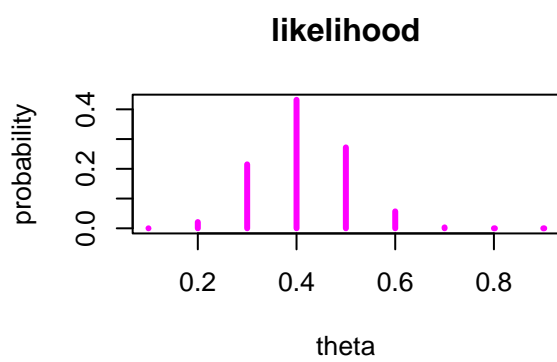
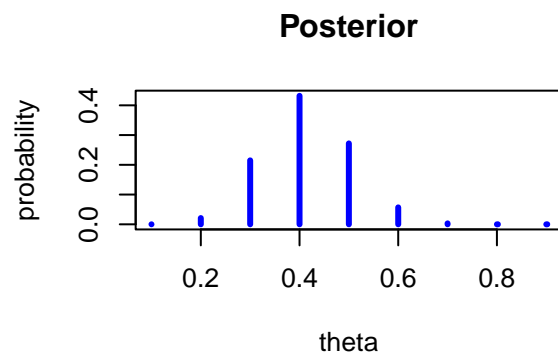
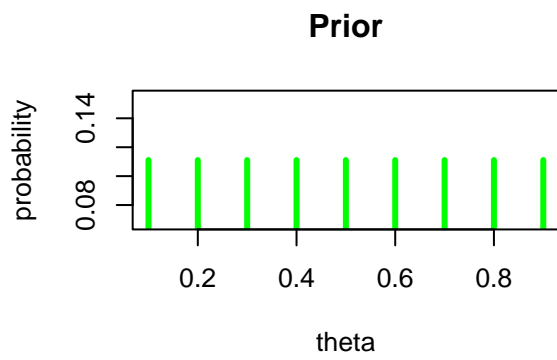
```
like2=like/sum(like)
cat("Normalized version of the likelihood:", like2)
```

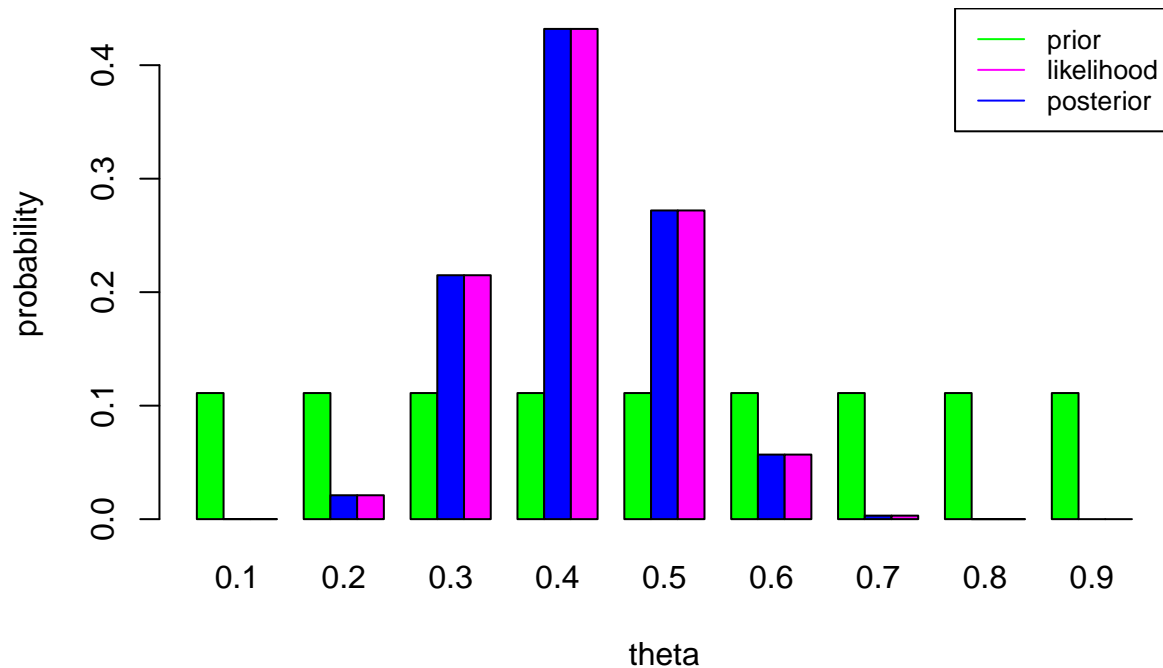
```
## Normalized version of the likelihood: 6.764703e-05 0.02104449 0.2149171 0.431965 0.2719936 0.05688427 0.003107328 2.055126e-05
```

```
cat("The sum of the normalized likelihood probabilities is", sum(like2)) # equal to 1!
```

```
## The sum of the normalized likelihood probabilities is 1
```

At this point, I have all the necessary elements to plot my prior, likelihood and posterior.





In this case, I notice that the posterior is very similar to the likelihood, due to the fact that the prior represent a situation of complete ignorance.

As usual, I next compute some relevant quantities about the posterior and I compare them with those of the prior.

Firstly, I compute the posterior mean and I observe that it is between the prior mean and the sample average.

```
# Posterior mean
post_mean=weighted.mean(theta,post)
sample_avg=mean(y)
cat("Prior mean:", prior_mean, ", Posterior mean:", post_mean,
    ", Sample average:", sample_avg)
```

```
## Prior mean: 0.5 , Posterior mean: 0.4137957 , Sample average: 0.4074074
```

Secondly, regarding the variances, I notice that the posterior variance is lower than the prior one, as in Case 1.

```
# Posterior variance
prior_var=weighted.mean((theta-prior_mean)^2, prior)
post_var=weighted.mean((theta-post_mean)^2, post)
cat("Prior variance:", prior_var, ", Posterior variance:", post_var)
```

```
## Prior variance: 0.06666667 , Posterior variance: 0.008084972
```

Furthermore, I compute the 95% credible intervals for both the prior and the posterior, using again the quantile function that I defined before.

It is possible to observe that the width of the posterior credible interval is smaller than the width of the prior credible interval.

```
# Quantiles
```

```
# Posterior cdf
```

```
cumpost=cdf(post)
```

```
cat("Cumulative distribution function of the posterior:", cumpost)
```

```
## Cumulative distribution function of the posterior: 6.764703e-05 0.02111213 0.2360293 0.6679942 0.939
```

```
# Prior 95% credible interval
```

```
l0=my_quant(cumprior, 0.025)
```

```
u0=my_quant(cumprior, 0.975)
```

```
cat("The 95% credible interval of the prior is (",l0,",",u0,")")
```

```
## The 95% credible interval of the prior is ( 0.1 , 0.9 )
```

```
# Posterior 95% credible interval
```

```
l_n=my_quant(cumpost, 0.025)
```

```
u_n=my_quant(cumpost, 0.975)
```

```
cat("The 95% credible interval of the posterior is (",l_n,",",u_n,")")
```

```
## The 95% credible interval of the posterior is ( 0.3 , 0.6 )
```

Lastly, I compute the 85th quantile to compare it with the results I had in Case 1. Obviously, since the prior placed relatively high probabilities also to the highest values of θ , the prior 85th quantile is higher than the posterior one.

```
# 85th quantile
```

```
prior_85q=my_quant(cumprior, 0.85)
```

```
post_85q=my_quant(cumpost, 0.85)
```

```
cat("Prior 85th quantile:", prior_85q,", Posterior 85th quantile:",post_85q)
```

```
## Prior 85th quantile: 0.8 , Posterior 85th quantile: 0.5
```

Prediction

For what concern the prediction of the number of students that sleep at least 8 hours in a new sample of 20, I exploit the same reasoning I used in case 1: the predictive can be seen as posterior mean.

Moreover, also in this case I create an example of the new sample of 20 students.

```
# Prediction
```

```
cat("The posterior predictive is:", post_mean)
```

```
## The posterior predictive is: 0.4137957
```

```
cat("The posterior predictive for the 20 observations of the new sample is", post_mean*20)
```

```
## The posterior predictive for the 20 observations of the new sample is 8.275915
```

```
# Example of the new sample
```

```
set.seed(123)
```

```
y_new=rbinom(20,1,prob=post_mean)
```

```
y_new
```

```
## [1] 0 1 0 1 1 0 0 1 0 0 1 0 1 0 0 1 0 0 0 1
```

```
cat("Students of the sample that sleep at least 8 hours:", sum(y_new))
```

```
## Students of the sample that sleep at least 8 hours: 8
```

The posterior predictive is 0.4137957 and it implies that, in the new sample of 20 students, $8.275915 \approx 8$ students sleep at least 8 hours. The example confirms this result.

Hypothesis testing

To conclude, I verify if the event “less than 50% of students get at least eight hours of sleep during a week day” is supported by the data or if its alternative event is.

```
# Hypothesis testing
```

```
# Pr(E0|y_1:n)
```

```
post_prob0= cumpost[5]
```

```
cat("The posterior probability of E0 is", post_prob0)
```

```
## The posterior probability of E0 is 0.9399878
```

```
# Pr(E1|y_1:n)
```

```
post_prob1= 1-post_prob0
```

```
# Pr(E0)
```

```
prior_prob0= cumprior[5]
```

```
# Pr(E1)
```

```
prior_prob1= 1-prior_prob0
```

```
# Bayes factor
```

```
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
```

```
cat("The Bayes Factor is",BF)
```

```
## The Bayes Factor is 12.53063
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.9399878$. This probability is obviously higher than the prior probability, that was approximately 0.5555556. Nevertheless, this probability is lower than the posterior probability in Case 1, due to the impact of the different prior.

The Bayes Factor is 12.53063, which indicates that the strength of data evidence in favor of E_0 is “strong”. The Bayes Factor increased with respect to Case 1, because in this case the data support E_0 much more than the prior.

Case 3

Prior choice

In this third case, the prior has to reflect different beliefs: the best guess of the value of θ is now 0.7 and, with probability 0.85, θ is bigger than 0.5. For this reason, I set the probabilities $\{p_j\}$ such that they represent this situation.

```
# Choice of the prior
theta=seq(from=0.1,to=0.9,by=0.1)
cat("Theta:", theta)

## Theta: 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

prior=c(0.01, 0.02, 0.03, 0.04, 0.05, 0.15, 0.25, 0.25, 0.20)
cat("Prior:", prior)

## Prior: 0.01 0.02 0.03 0.04 0.05 0.15 0.25 0.25 0.2

cat("The sum of the prior probabilities is", sum(prior))

## The sum of the prior probabilities is 1

# Cdf of the prior
cumprior=cdf(prior)
cat("Cumulative distribution function of the prior:", cumprior)

## Cumulative distribution function of the prior: 0.01 0.03 0.06 0.1 0.15 0.3 0.55 0.8 1

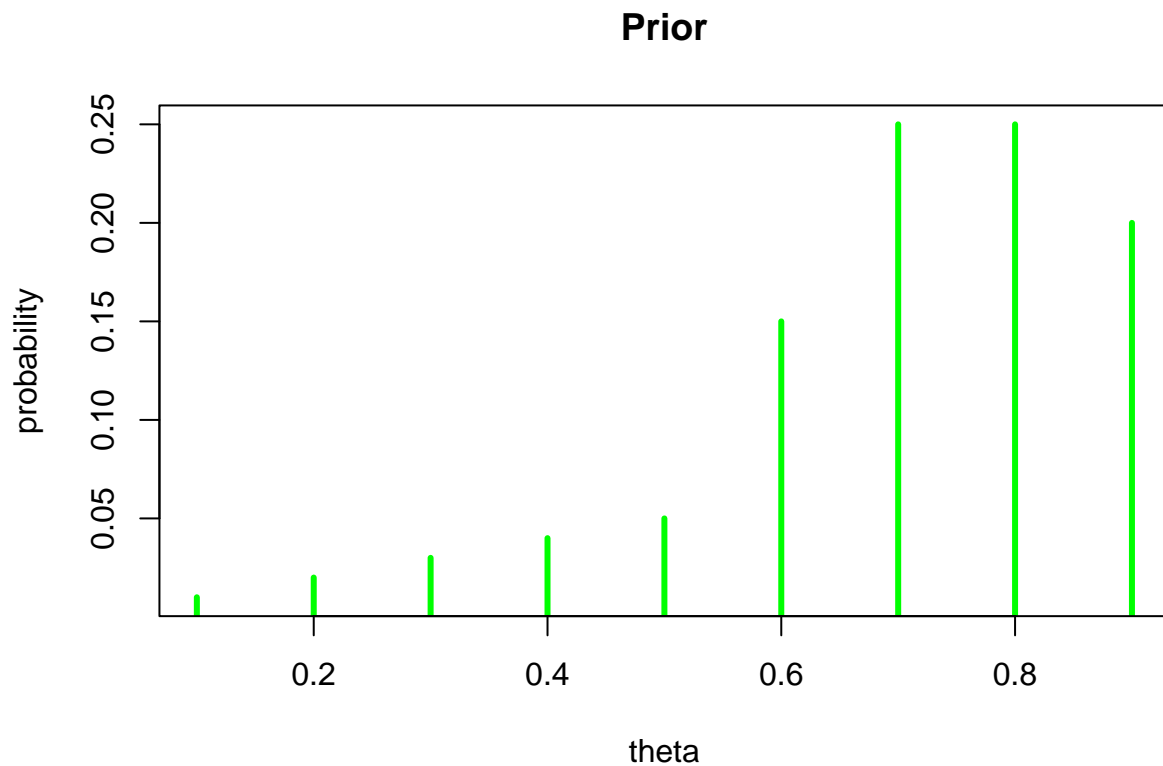
# P(theta<=0.5)=0.15 and E(theta)=0.7
cat("P(theta<=0.5)=", cumprior[5])

## P(theta<=0.5)= 0.15

prior_mean=weighted.mean(theta,prior)
cat("E(theta)=", prior_mean)

## E(theta)= 0.7
```

```
# Plot of the prior
plot(theta, prior, main="Prior", type="h", xlab="theta", ylab="probability",
      col="green", lwd=3)
```



Bayesian analysis

I now compute the posterior using the same reasoning I used in Case 1 and 2.

```
# Posterior
prop_post=c()
like=c()

for (j in 1:9){
  like[j]=(theta[j])^M*(1-theta[j])^(n-M)
  prop_post[j]=((theta[j])^M*(1-theta[j])^(n-M)*prior[j])
}

sum(prop_post) # not equal to 1 because I still have to divide for the marginal
```

```
## [1] 1.289144e-09
```

```
post=prop_post/sum(prop_post)
cat("Posterior:", post)
```

```
## Posterior: 1.437403e-05 0.008943309 0.1370005 0.3671457 0.2889739 0.1813065 0.01650658 0.0001091713 4
```

```
cat("The sum of the posterior probabilities is", sum(post)) # equal to 1!
```

```
## The sum of the posterior probabilities is 1
```

Also in this case, I compute a normalized version of the likelihood for the next plots.

```
# Normalized version of the likelihood
cat("Likelihood:", like)
```

```
## Likelihood: 1.85302e-12 5.764608e-10 5.887114e-09 1.183259e-08 7.450581e-09 1.558201e-09 8.511743e-1
```

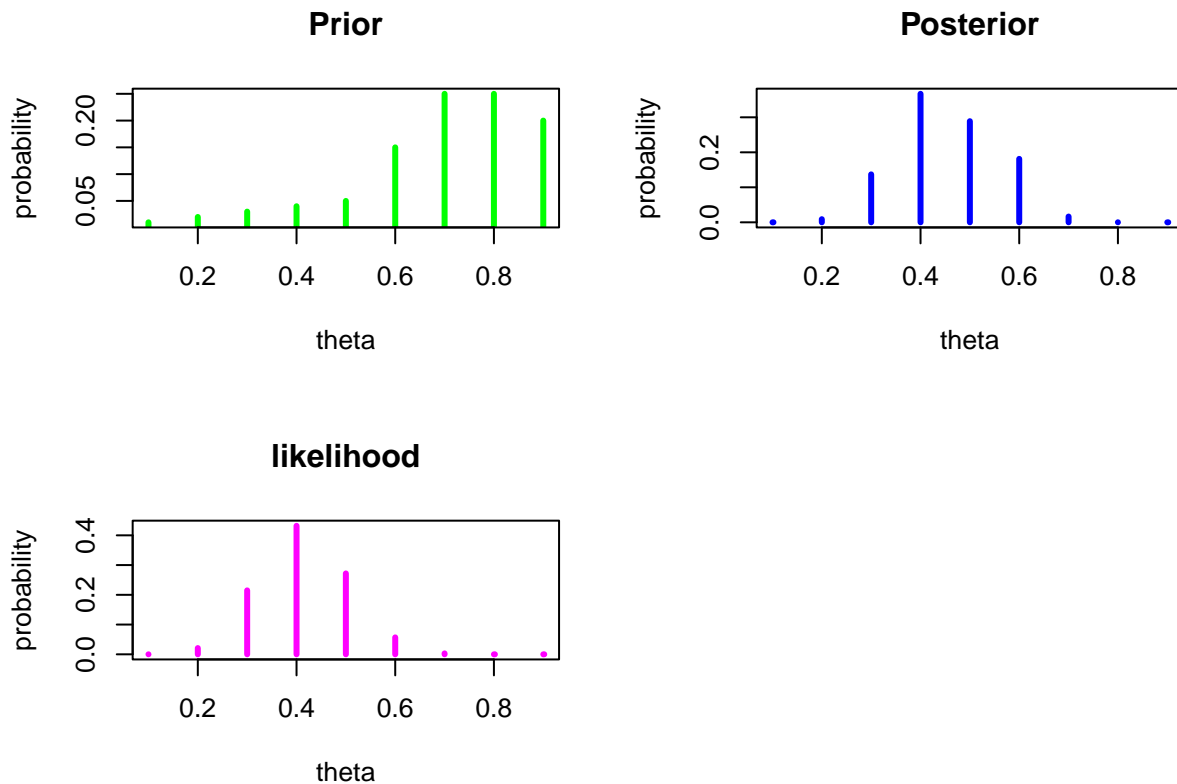
```
like2=like/sum(like)
cat("Normalized version of the likelihood:", like2)
```

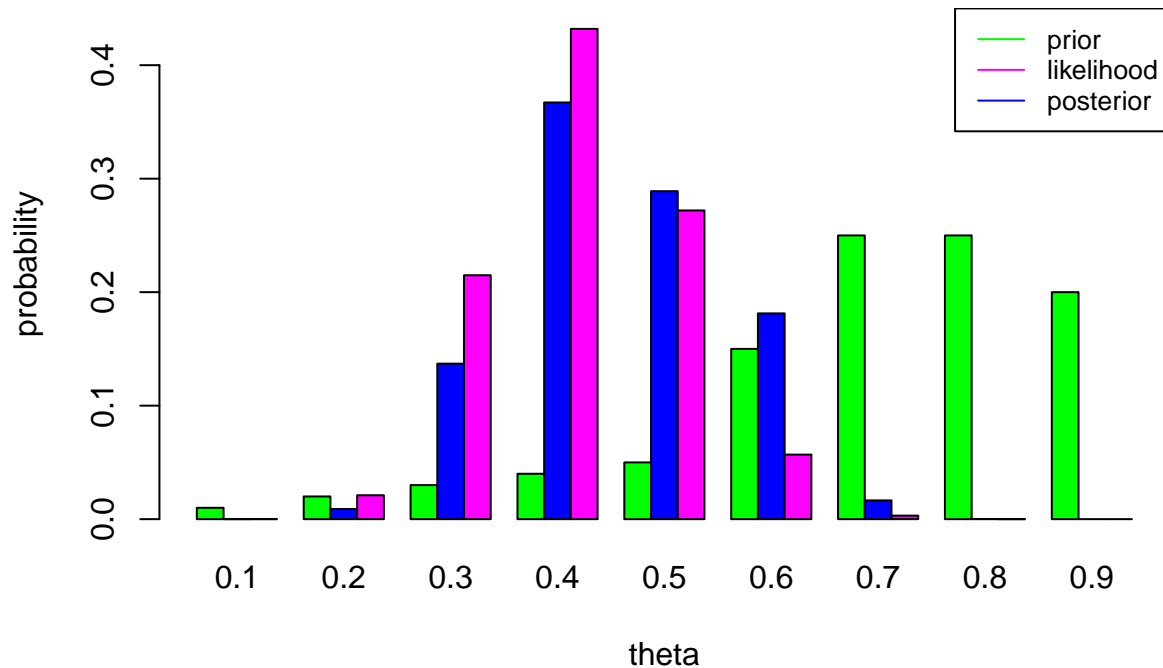
```
## Normalized version of the likelihood: 6.764703e-05 0.02104449 0.2149171 0.431965 0.2719936 0.0568842
```

```
cat("The sum of the normalized likelihood probabilities is", sum(like2)) # equal to 1!
```

```
## The sum of the normalized likelihood probabilities is 1
```

I am now able to plot my prior, likelihood and posterior.





As usual, I compute some relevant quantities with respect to the posterior and I compare them with those of the prior.

Firstly, I compute the posterior mean and I observe that it is between the prior mean and the sample average.

```
# Posterior mean
post_mean=weighted.mean(theta,post)
sample_avg=mean(y)
cat("Prior mean:", prior_mean, ", Posterior mean:", post_mean,
    ", Sample average:", sample_avg)
```

```
## Prior mean: 0.7 , Posterior mean: 0.4546613 , Sample average: 0.4074074
```

Secondly, the posterior variance is lower than the prior one, as in Case 1 and 2, but it is slightly higher than in the first two cases.

```
# Posterior variance
prior_var=weighted.mean((theta-prior_mean)^2, prior)
post_var=weighted.mean((theta-post_mean)^2, post)
cat("Prior variance:", prior_var, ", Posterior variance:", post_var)
```

```
## Prior variance: 0.031 , Posterior variance: 0.01038623
```

In addition, I compute the 95% credible intervals for both the prior and the posterior, using again the quantile function that I defined before.

It is again possible to observe that the width of the posterior credible interval is smaller than the width of the prior credible interval.

```
# Quantiles
```

```
# Posterior cdf
```

```
cumpost=cdf(post)
```

```
cat("Cumulative distribution function of the posterior:", cumpost)
```

```
## Cumulative distribution function of the posterior: 1.437403e-05 0.008957683 0.1459582 0.5131039 0.80
```

```
# Prior 95% credible interval
```

```
l0=my_quant(cumprior, 0.025)
```

```
u0=my_quant(cumprior, 0.975)
```

```
cat("The 95% credible interval of the prior is (",l0,",",u0,")")
```

```
## The 95% credible interval of the prior is ( 0.2 , 0.9 )
```

```
# Posterior 95% credible interval
```

```
l_n=my_quant(cumpost, 0.025)
```

```
u_n=my_quant(cumpost, 0.975)
```

```
cat("The 95% credible interval of the posterior is (",l_n,",",u_n,")")
```

```
## The 95% credible interval of the posterior is ( 0.3 , 0.6 )
```

Lastly, I compute the 15th quantile to compare it with the prior one that was, by assumption, 0.5. As in Case 2, since the prior placed high probabilities to the highest values of θ , the prior 15th quantile is higher than the posterior one.

```
# 15th quantile
```

```
prior_15q=my_quant(cumprior, 0.15)
```

```
post_15q=my_quant(cumpost, 0.15)
```

```
cat("Prior 15th quantile:", prior_15q, ", Posterior 15th quantile:", post_15q)
```

```
## Prior 15th quantile: 0.5 , Posterior 15th quantile: 0.4
```

Prediction

For what concern the prediction of the number of students that sleep at least 8 hours in a new sample of 20, I exploit the same reasoning I used in Case 1 and 2: the predictive can be seen as posterior mean.

Moreover, also in this case I create an example of the new sample of 20 students.

```
# Prediction
```

```
cat("The posterior predictive is:", post_mean)
```

```
## The posterior predictive is: 0.4546613
```

```
cat("The posterior predictive for the 20 observations of the new sample is", post_mean*20)
```

```
## The posterior predictive for the 20 observations of the new sample is 9.093226
```

```
# Example of the new sample
set.seed(123)
y_new=rbinom(20,1,prob=post_mean)
y_new
```

```
## [1] 0 1 0 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 0 1
```

```
cat("Students of the sample that sleep at least 8 hours:", sum(y_new))
```

```
## Students of the sample that sleep at least 8 hours: 10
```

The posterior predictive is 0.4546613 and it implies that, in the new sample of 20 students, $9.093226 \approx 9$ students sleep at least 8 hours. In the example sample 10 students out of 20 sleep at least 8 hours. The difference with the result of the predictive is due to the small numerosity of the sample.

Hypothesis testing

To conclude, as in other cases, I verify if the event “less than 50% of students get at least eight hours of sleep during a week day” is supported by the data or if its alternative event is.

```
# Hypothesis testing

# Pr(E0|y_1:n)
post_prob0= cumpost[5]
cat("The posterior probability of E0 is", post_prob0)
```

```
## The posterior probability of E0 is 0.8020777
```

```
# Pr(E1|y_1:n)
post_prob1= 1-post_prob0

# Pr(E0)
prior_prob0= cumprior[5]

# Pr(E1)
prior_prob1= 1-prior_prob0

# Bayes factor
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
cat("The Bayes Factor is",BF)
```

```
## The Bayes Factor is 22.9641
```

```
rm(list = ls())
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.8020777$. This probability is obviously higher than the prior probability, that was by assumption 0.15. Nevertheless, this probability is lower than the posterior probabilities in Case 1 and 2, due to the impact of the prior, that places higher probabilities on the highest values of θ .

The Bayes Factor is 22.9641, which indicates that the strength of data evidence in favor of E_0 is “strong”, as in Case 2. The Bayes Factor is, however, higher than in Case 2, because in this case the data support E_0 more, whereas the prior supports more E_1 more.

Third model - non-conjugate analysis

Prior analysis

The objective in this first paragraph is to understand how the prior density $\pi_3(\theta)$ changes with the prior hyperparameters c and s . To this end, I used a Monte Carlo procedure to fill the two tables reported in the assignment.

Monte Carlo method is a numerical technique that approximates θ and the functionals of the relative distribution. The procedure consists in drawing a large iid sample from the distribution $(\theta_1, \dots, \theta_G \sim \pi_3(\theta))$ and in computing the approximation:

$$\hat{\eta}_G := \frac{1}{G} \sum_{j=1}^G h(\theta_j)$$

In particular, I am interested in approximating the mean $E(\theta)$ and the probability $P(\theta < 0.5)$.

To draw iid sample, I use the hint that suggests that $\theta = \frac{1}{1+e^{-X}}$ where $X \sim N(c, s^2)$ and I proceed as follows.

```
# Set the vectors of hyperparameters
c=c(-1,0,1)
s=c(4,1,0.1)

# Set up the two tables
m_theta=matrix(nrow=3,ncol=3)
p0.5_theta=matrix(nrow=3, ncol=3)

# For loop to draw the iid samples and fill the tables
for (i in 1:length(c)){
  for (j in 1:length(s)){
    set.seed(123)
    # sample X iid from the N(c,s^2)
    x=rnorm(10000,mean=c[i],sd=s[j])
    # transform the X sample into the relative theta
    # Since the X are iid, also the theta are iid
    theta=1/(1+exp(-x))

    m_theta[j,i]=mean(theta)
    p0.5_theta[j,i]=mean(theta<0.5)
  }
}
```

```
# E(theta) with different hyperparameters
m_theta
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.4083261 0.4990392 0.5901093
## [2,] 0.3027046 0.4994800 0.6963821
## [3,] 0.2693463 0.4999407 0.7305605
```

```
#P(theta<0.5) with different hyperparameters
p0.5_theta
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.6009 0.5041 0.3994
## [2,] 0.8424 0.5041 0.1612
## [3,] 1.0000 0.5041 0.0000
```

To verify my approximation, I also compute the same quantities drawing a sample from a Uniform(0,1) and assigning to each unit of the sample a probability equal to $\pi_3(\theta)$. The results are similar to what I obtained above.

```
c=c(-1,0,1)
s=c(4,1,0.1)

m_theta1=matrix(nrow=3,ncol=3)
p0.5_theta1=matrix(nrow=3, ncol=3)

for (i in 1:length(c)){
  for (j in 1:length(s)){
    set.seed(123)
    th=runif(10000)
    p3=1/(s[j]*sqrt(2*3.14))*exp((-1/(2*s[j]^2))*(log(th/(1-th))-c[i])^2)*1/(th*(1-th))
    theta=(sample(th,10000,replace=T,prob=p3))
    m_theta1[j,i]= mean(theta)
    p0.5_theta1[j,i]=mean(theta<0.5)
  }
}

m_theta1
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.4099258 0.4983206 0.5848241
## [2,] 0.3058240 0.4962844 0.6916300
## [3,] 0.2696746 0.4989374 0.7311644
```

```
p0.5_theta1
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.6014 0.5039 0.4081
## [2,] 0.8409 0.5108 0.1672
## [3,] 1.0000 0.5247 0.0000
```

```
rm(list = ls())
```

Case 1

Prior hyperparameters choice

In this first case, the hyperparameters have to reflect the information included in the “Prior Information” section.

I choose the hyperparameters $c=-1$ and $s=1$, because using them, the prior distribution of θ has approximately mean 0.3, which is the best guess described in the “Prior information” paragraph. Moreover, $P(\theta < 0.5) = 0.8424$, which is similar to the probability that theta is smaller than 0.5 described in the “Prior information” section (that was 85%)

Together with the hyperparameters I also define the prior function.

```
# Fix the hyperparameters
c=-1
s=1

# Prior function
model_prior=function(th, n, M, c, s){
  if (th<=0|th>=1) {
    return(-Inf)
  }
  out=1/(s*sqrt(2*3.14))*exp((-1/(2*s^2))*(log(th/(1-th))-c)^2)*1/(th*(1-th))
  return(out)
}
```

Posterior

To proceed in the Bayesian analysis, I also need the Data and the relative sufficient statistics.

```
# Data and sufficient statistics
y=c(rep(0,16), rep(1,11))
n=length(y)
M=sum(y)
cat("The sample size is",n,"and the students that sleep more than 8 hours are",M)
```

```
## The sample size is 27 and the students that sleep more than 8 hours are 11
```

Basing on the prior and on the data, I am now able to define the posterior function, which will correspond to the “target” of the Metropolis-Hastings procedure that I will implement.

A graphical representation of the posterior is also plotted.

```
# Posterior= target of the MH
model_posterior=function(th, n, M, c, s){
  if (th<=0|th>=1) {
    return(-Inf)
  }
  out= (th)^M*(1-th)^(n-M) * 1/(s*sqrt(2*3.14))*exp((-1/(2*s^2))*(log(th/(1-th))-c)^2)*1/(th*(1-th))
}
```

```

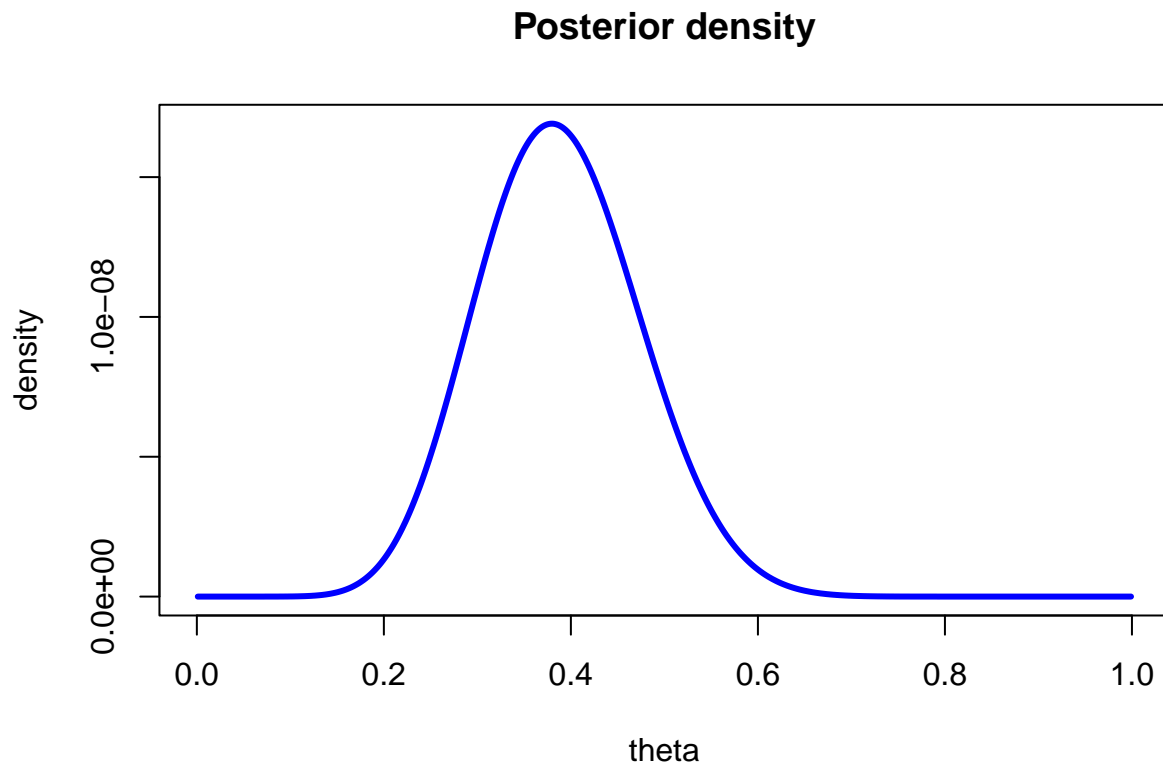
    return(out)
}

# Plot
gr_th=seq(0,1,0.001)
mp=c()

for (i in 1:length(gr_th)){
  mp[i]=model_posterior(gr_th[i], n, M, c, s)
}

plot(gr_th, mp, lwd=3, type="l", col="blue", main="Posterior density",
      xlab="theta", ylab="density")

```



Metropolis-Hastings procedure

At this stage, I have to implement a Metropolis-Hastings with the aim of drawing, from the posterior, a dependent identically distributed sample that I will use in my Bayesian analysis.

Firstly, I define a function that includes all the passages of the Metropolis Hastings.

I choose, as proposal, a Normal distribution with mean θ_{g-1} and standard deviation (η) to be defined. I am therefore applying a random walk Metropolis-Hastings and this allows me to compute the acceptance rejection ratio as the ratio between the target computed at the proposed value and the target computed at the given value.

Comparing this value with a randomly drawn value from a Uniform(0,1), I am able to decide if the move is accepted or rejected and, therefore, at the end, to compute the probability of acceptance.

I also considered the possibility to fix a burn-in, to improve the convergence of the Markov Chain to the stationary distribution, and a thinning, to improve the situation of dependence between elements of the Markov Chain.

```
# MH function
MH_post3=function(G, burnin, thin, n, M, c, s, eta,th0=0.5){

  #Set the number of iterations
  iterations= burnin+thin*G
  g= 1

  # Define the output vector
  theta= vector("numeric", G)
  current_state= th0 #the current state of the chain is th_0

  acc=0; #count how many time I accept a transition

  for (iter in 1:iterations) {
    prop_state = rnorm(1,mean=current_state,sd=eta) #My proposal are drawn from a Normal

    if(prop_state<=0|prop_state>=1){# Values I cannot accept
      alpha= -Inf
    }else{
      alpha= min(1, (model_posterior(prop_state, n, M, c, s))/
                  (model_posterior(current_state, n, M, c, s)))
    }

    u= runif(1)
    if (u < alpha){
      #Accept the move
      current_state= prop_state
      acc= acc+1
    } # the else is not needed because nothing changes

    if( (iter>burnin) & (iter%%thin==0) ){
      # if iter is larger than burn
      # and iter is multiple of thin
      # Save the current_state
      theta[g]= current_state
      g= g+1
    }
  }
  cat("I accepted the ", acc/iterations*100,"% of the proposed transition\n")
  return(theta)
}
```

I am now able to draw a sample from the posterior. In this context, I need to define an optimal variance for the proposal, such that the frequency of accepted transitions is about 0.234. After some attempts, I found that the optimal standard deviation (η) in this case is approximately 0.44.

```
set.seed(123)
th_sample1= MH_post3(G=5000, burnin = 1000, thin = 10, n, M, c, s, eta=0.44, th0=0.5)
```

```
## I accepted the 23.37647 % of the proposed transition
```

It is possible to show how the variance of the proposal affects the Metropolis-Hastings sample, by setting a non optimal variance in the sampling phase. I am going to show two cases: one with a variance that is higher than the optimal and one with a variance that is lower than the optimal.

```
set.seed(123)
wrong_th_sample1= MH_post3(G=5000, burnin = 1000, thin = 10, n, M, c, s, eta=4, th0=0.5)
```

```
## I accepted the 2.594118 % of the proposed transition
```

```
set.seed(123)
wrong_th_sample2= MH_post3(G=5000, burnin = 1000, thin = 10, n, M, c, s, eta=0.01, th0=0.5)
```

```
## I accepted the 96.47843 % of the proposed transition
```

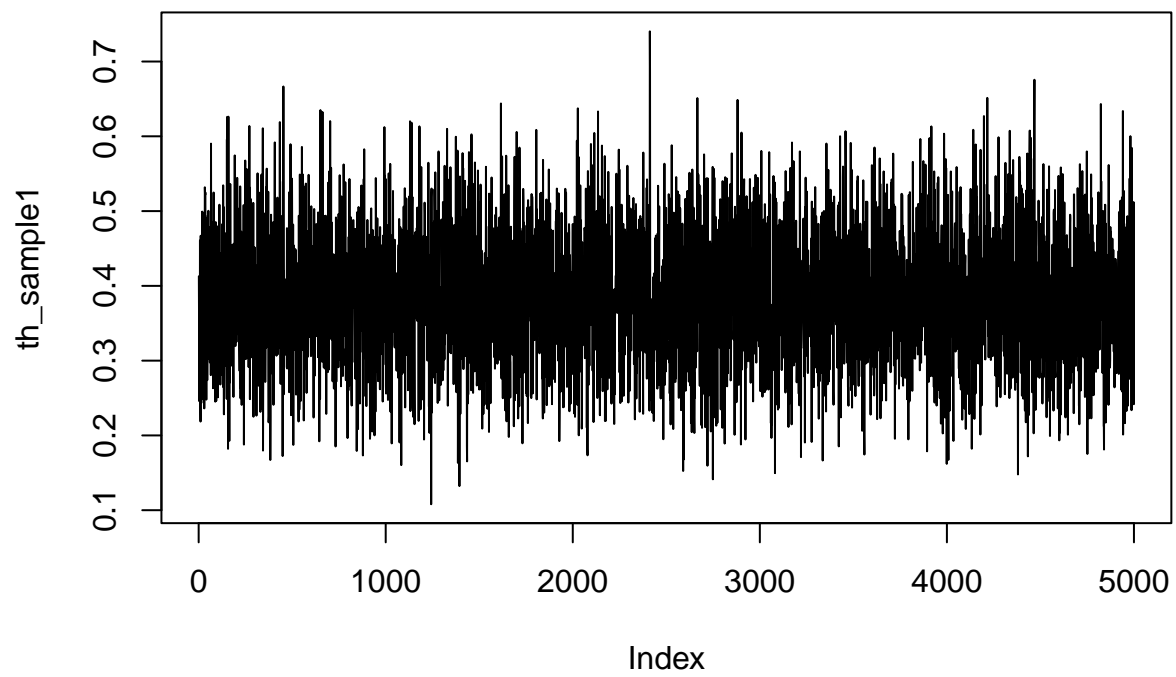
It is possible to notice that in the first case, the frequency of accepted transitions is too low, whereas in the second case, it is too high.

Convergence check

Since the Metropolis-Hasting Monte Carlo is a numerical technique and hence it is subject to approximation error, a proper analysis of the convergence is always appropriate.

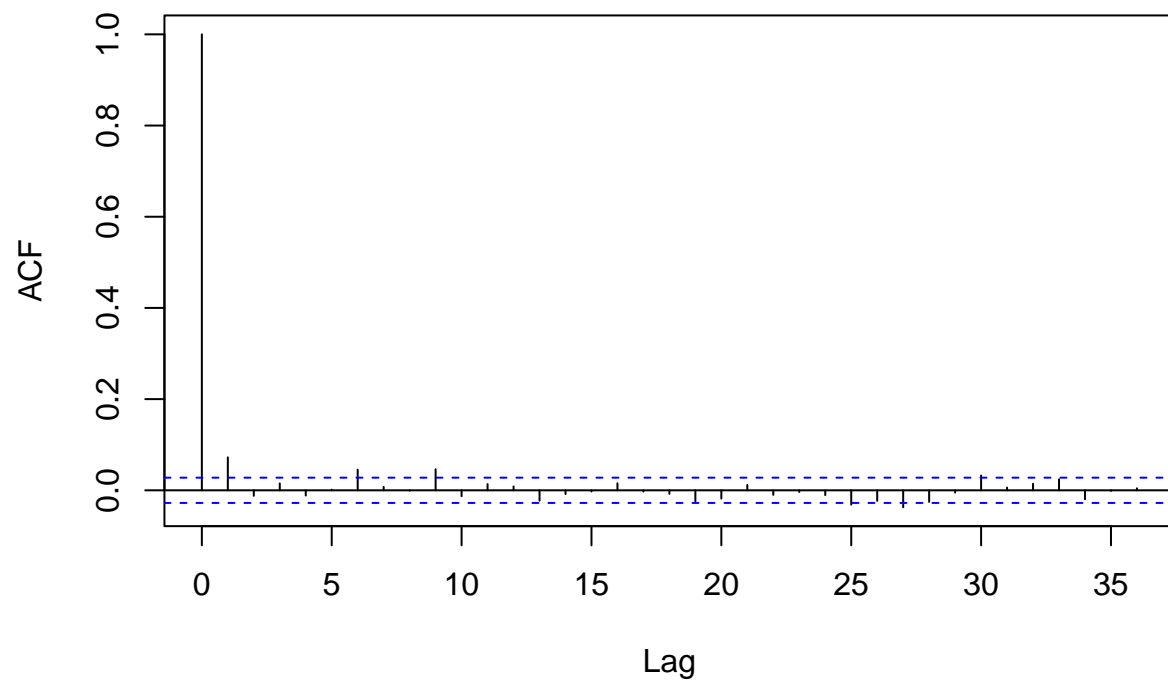
Firstly, a graphical analysis of the convergence and of the autocorrelation can be done, using the traceplot and the autocorrelation plot. The cumulative mean plot is also reported.

```
# Convergence and autocorrelation check for sample 1
# trace plot
plot(th_sample1,type="l")
```

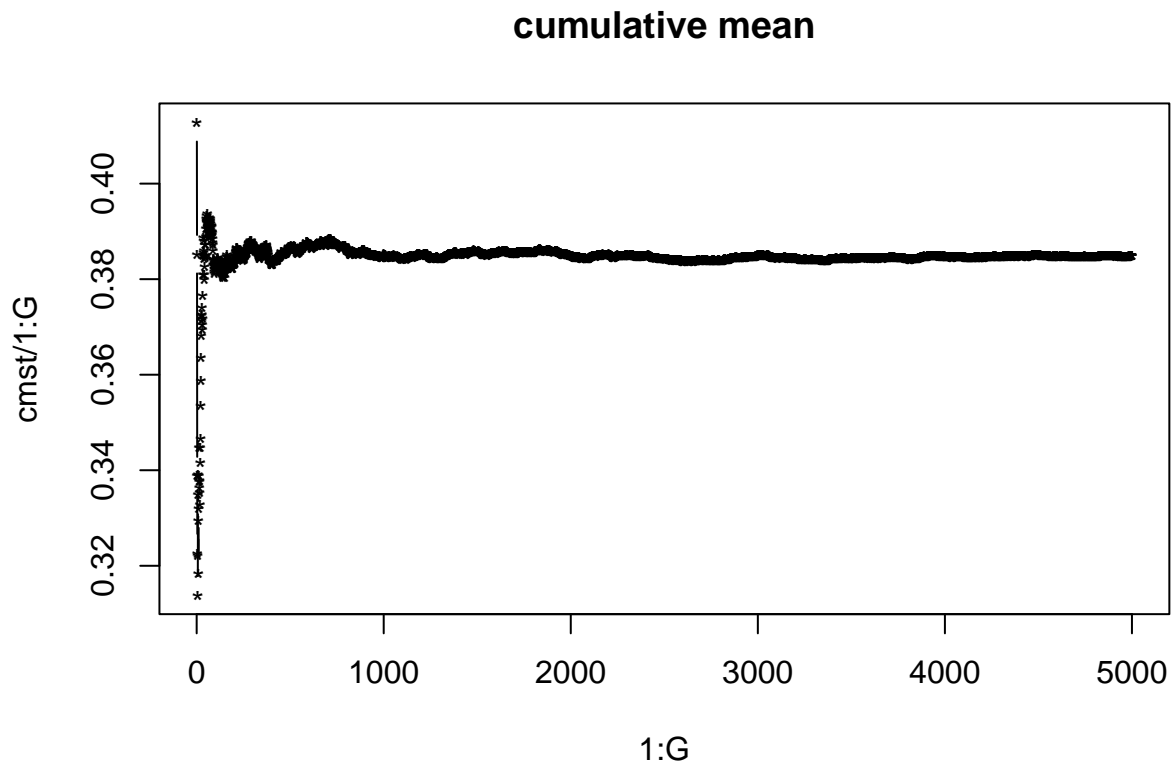



```
# auto correlation function  
acf(th_sample1)
```

Series th_sample1



```
# Cumulative mean
cmst <- cumsum(th_sample1)
G <- 5000
plot(1:G, cmst/1:G, type="b", main="cumulative mean", pch="*")
```

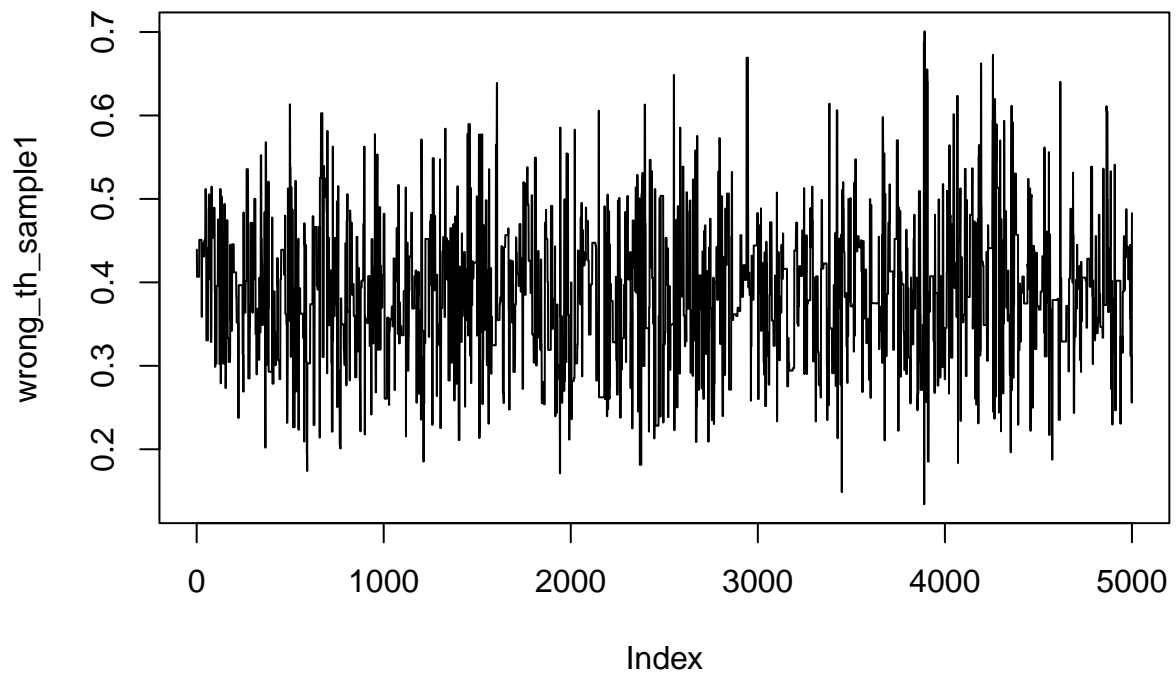


The traceplot shows a good mixing and also the autocorrelation drops after the first lag. In addition, the running mean stabilizes as g increases.

These graphs also show the effect of a non optimal choice of the variance of the proposal.

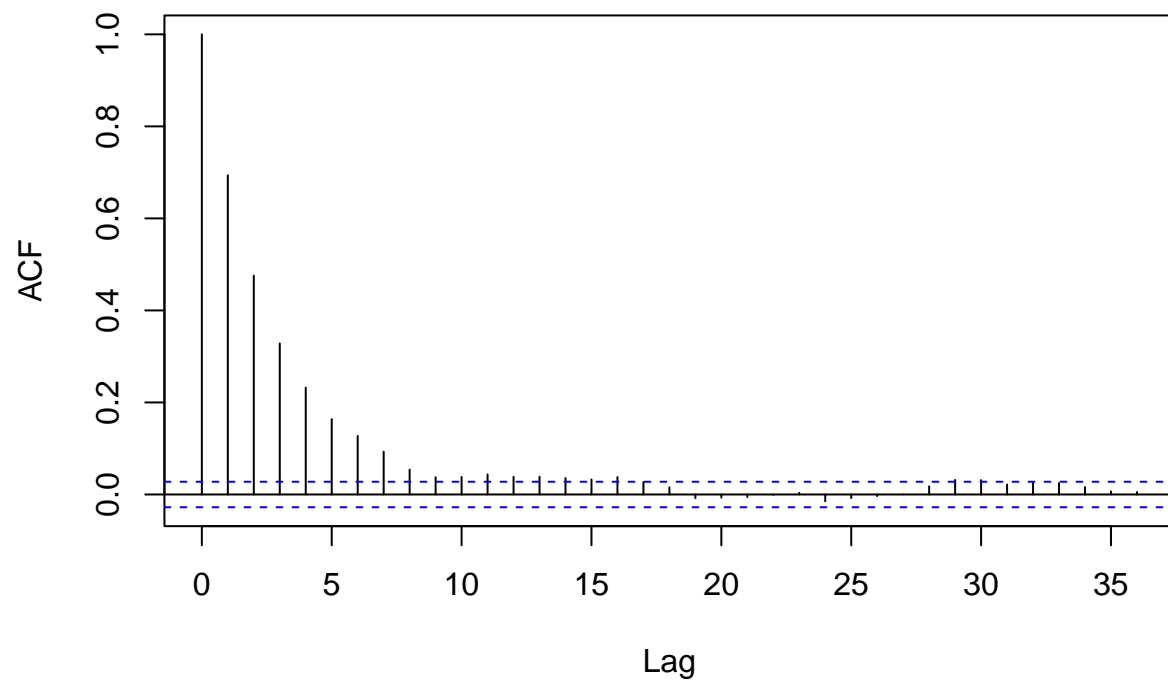
The first case is the one in which the variance is too high.

```
# Convergence and autocorrelation check for wrong sample 1  
# trace plot  
plot(wrong_th_sample1,type="l")
```



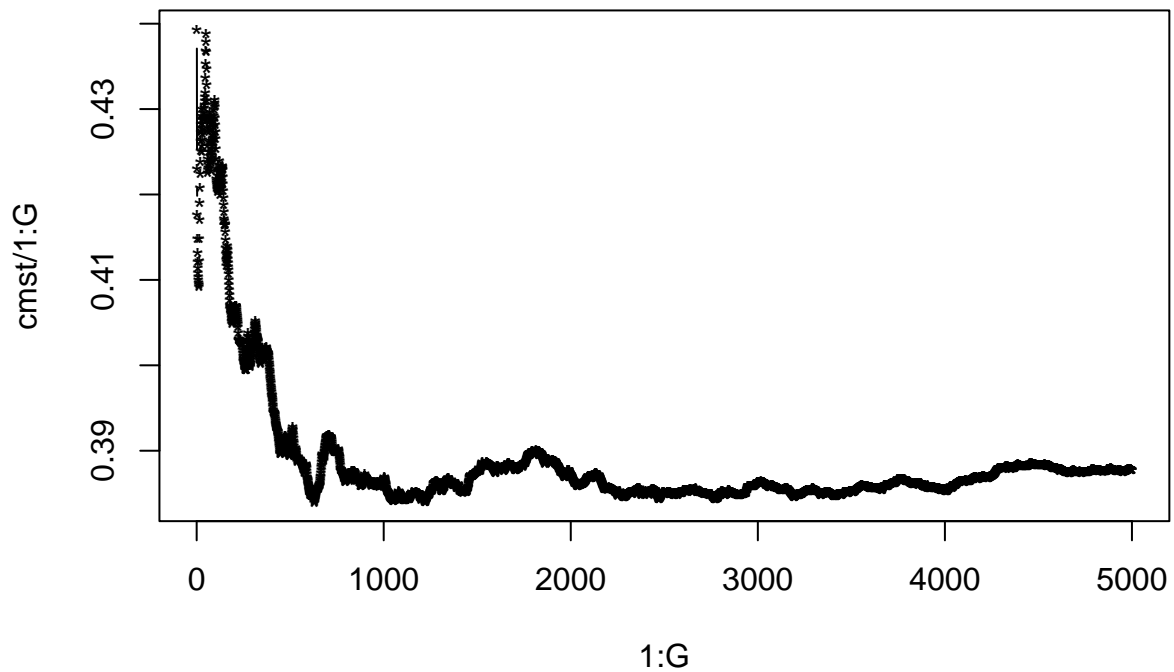
```
# auto correlation function  
acf(wrong_th_sample1)
```

Series wrong_th_sample1



```
# Cumulative mean
cmst <- cumsum(wrong_th_sample1)
G <- 5000
plot(1:G,cmst/1:G,type="b",main="cumulative mean",pch="*")
```

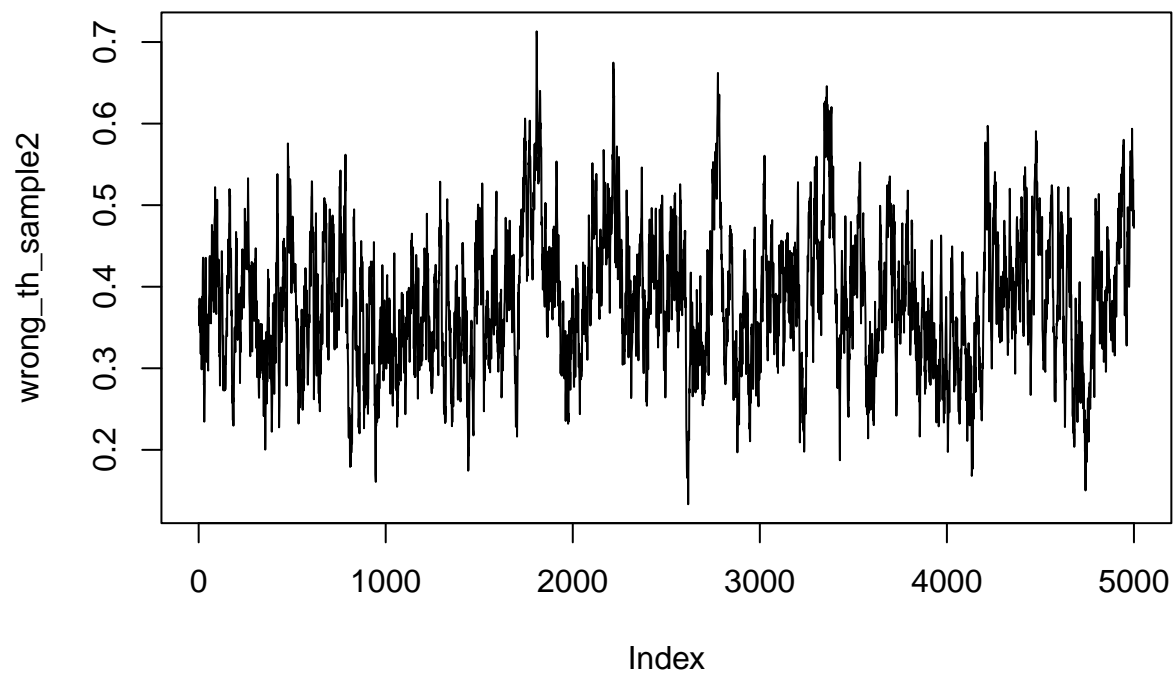
cumulative mean



It is possible to notice that, since the acceptance probability is too low, the traceplot has the typical form of a skyline. Furthermore, the autocorrelation decreases at a lower pace and the running mean is not stable, even when larger g are considered.

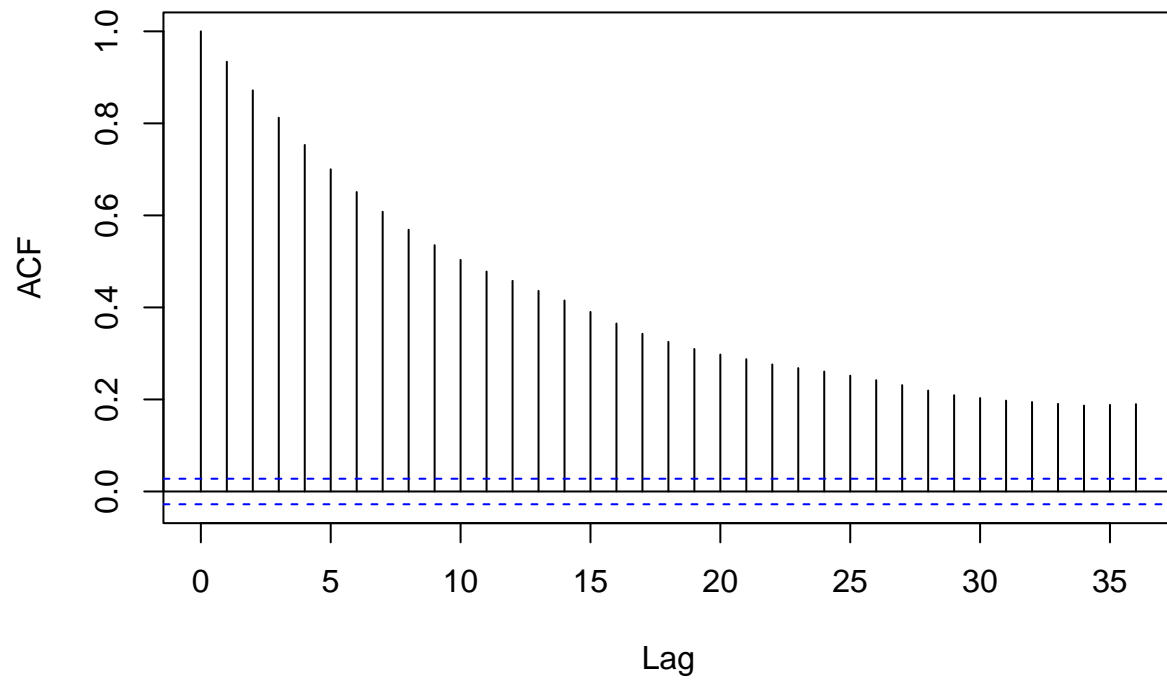
The second case is the one in which the variance is too low.

```
# Convergence and autocorrelation check for wrong sample 2
# trace plot
plot(wrong_th_sample2, type="l")
```

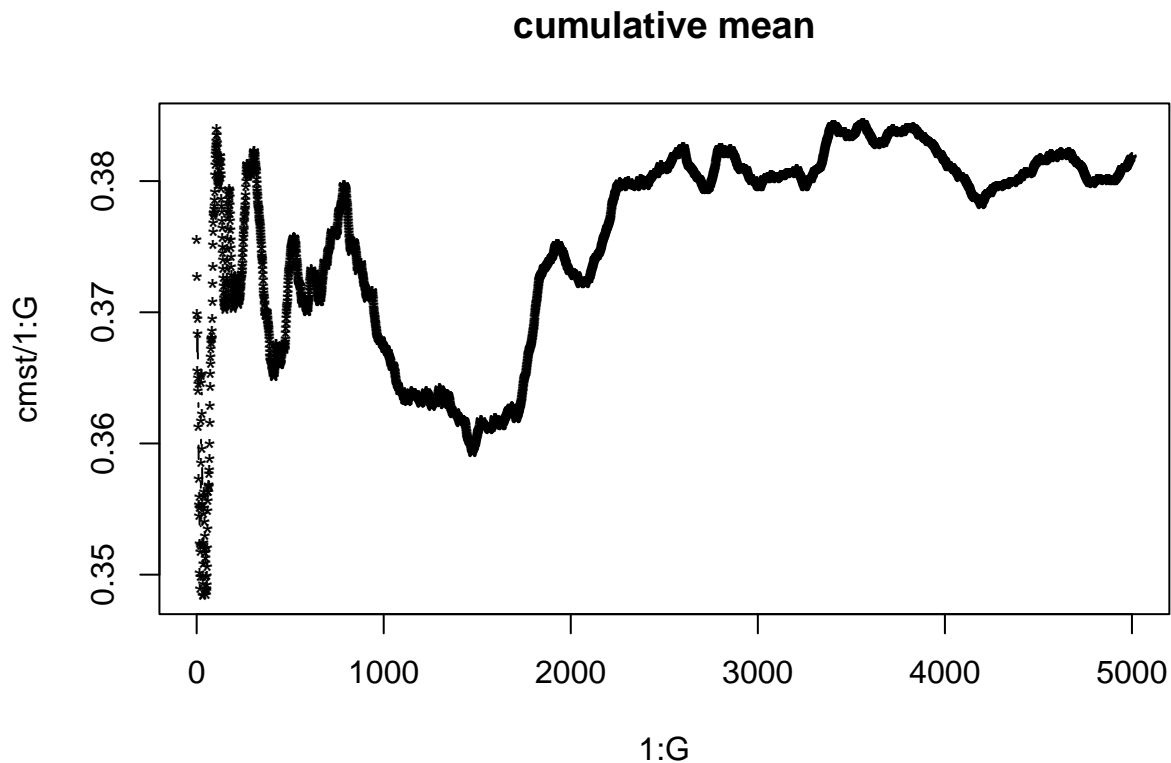


```
# auto correlation function  
acf(wrong_th_sample2)
```

Series wrong_th_sample2



```
# Cumulative mean
cmst <- cumsum(wrong_th_sample2)
G <- 5000
plot(1:G, cmst/1:G, type="b", main="cumulative mean", pch="*")
```

The traceplot follows a pattern that clearly indicates that the frequency of accepted transitions is too high. The autocorrelation plot shows a situation that is worse than the one before, with an autocorrelation that does not reach the zero even after 35 lags. Also in this case, the running mean keep varying for large g , even more than in the previous case.

To reach the purpose of checking the convergence, however, these plots are not enough: formal convergence diagnostics are needed. In particular, I am going to consider the Geweke diagnostic and the Gelman and Rubin diagnostic for the chain with optimal proposal variance. To this end, I will use the package “coda”, that provides these and other tools to check the convergence.

Firstly, I consider some summary statistics of my MH sample and I plot slightly different versions of the trace plot, the autocorrelation plot and the running mean plot, provided by this package. These alternative versions provide some additional information with respect to the one I plotted before, such as a plot of the density of the sample and a representation of the confidence intervals of the running mean.

```
# Summary and plots with coda
library(coda)
```

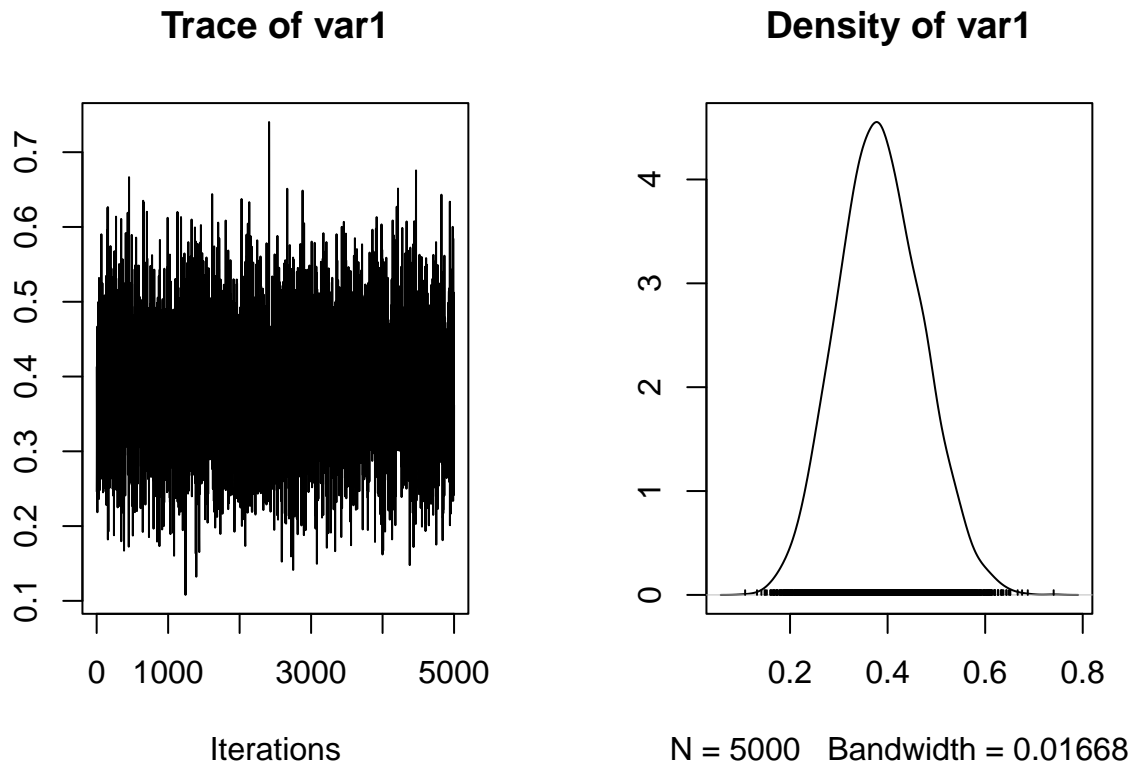
```
## Warning: il pacchetto 'coda' è stato creato con R versione 4.2.2
```

```
th.post.mc= mcmc(th_sample1)
summary(th.post.mc)
```

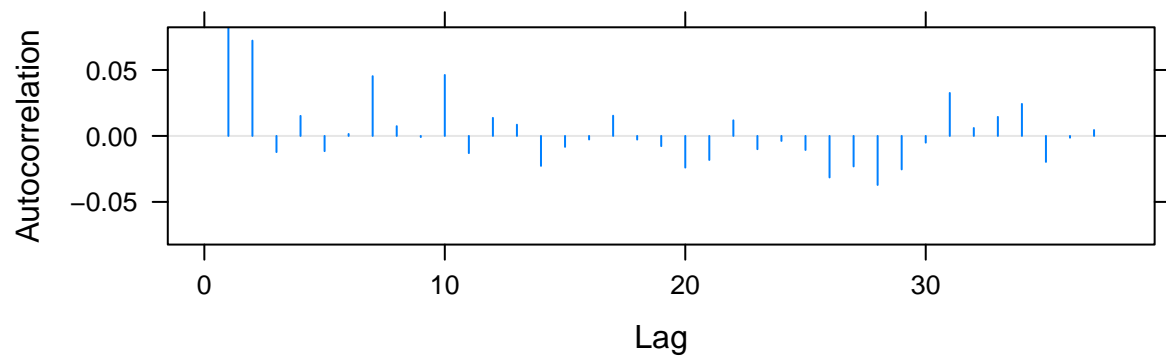
```
##
## Iterations = 1:5000
```

```
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean           SD      Naive SE Time-series SE
##      0.384855      0.086428      0.001222      0.001425
##
## 2. Quantiles for each variable:
##
##   2.5%   25%   50%   75%  97.5%
## 0.2229 0.3249 0.3818 0.4438 0.5579
```

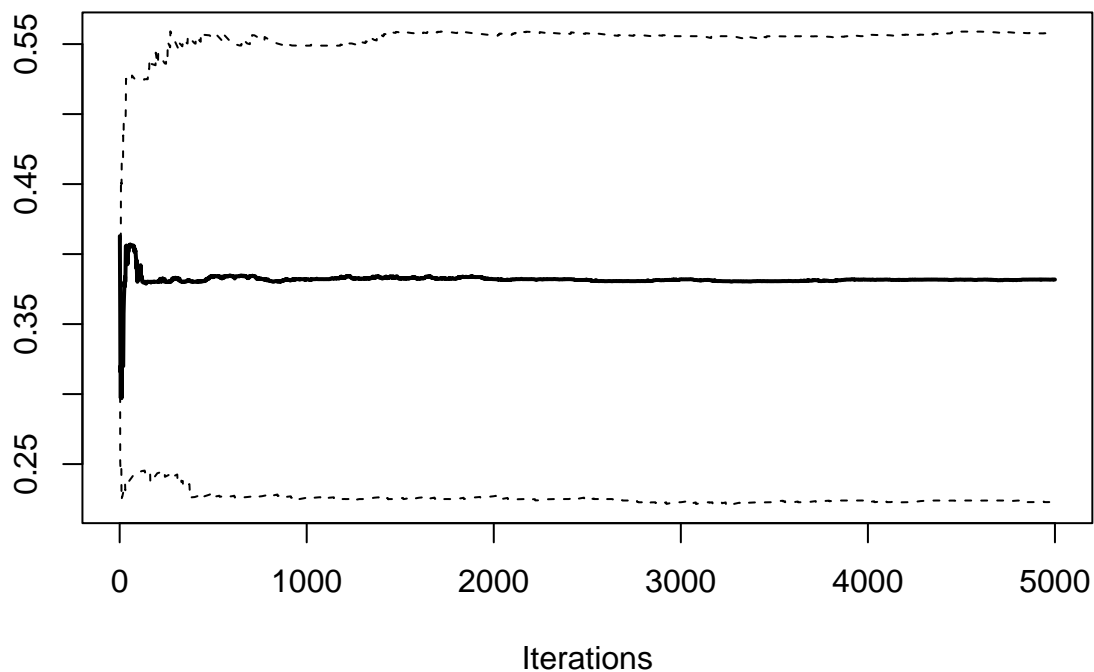
```
plot(th.post.mc)
```



```
acfplot(th.post.mc)
```



```
cumuplot(th.post.mc)
```



Now, I proceed with the Geweke diagnostic. This diagnostic is based on a test:

- The chain is divided into two parts, the initial 10
- A test of equal means in the two parts is performed
- If the hypothesis of equal means is accepted, the chain is stationary, otherwise it is not.

```
# Geweke diagnostic
geweke.diag(th.post.mc)
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1
## 0.3524
```

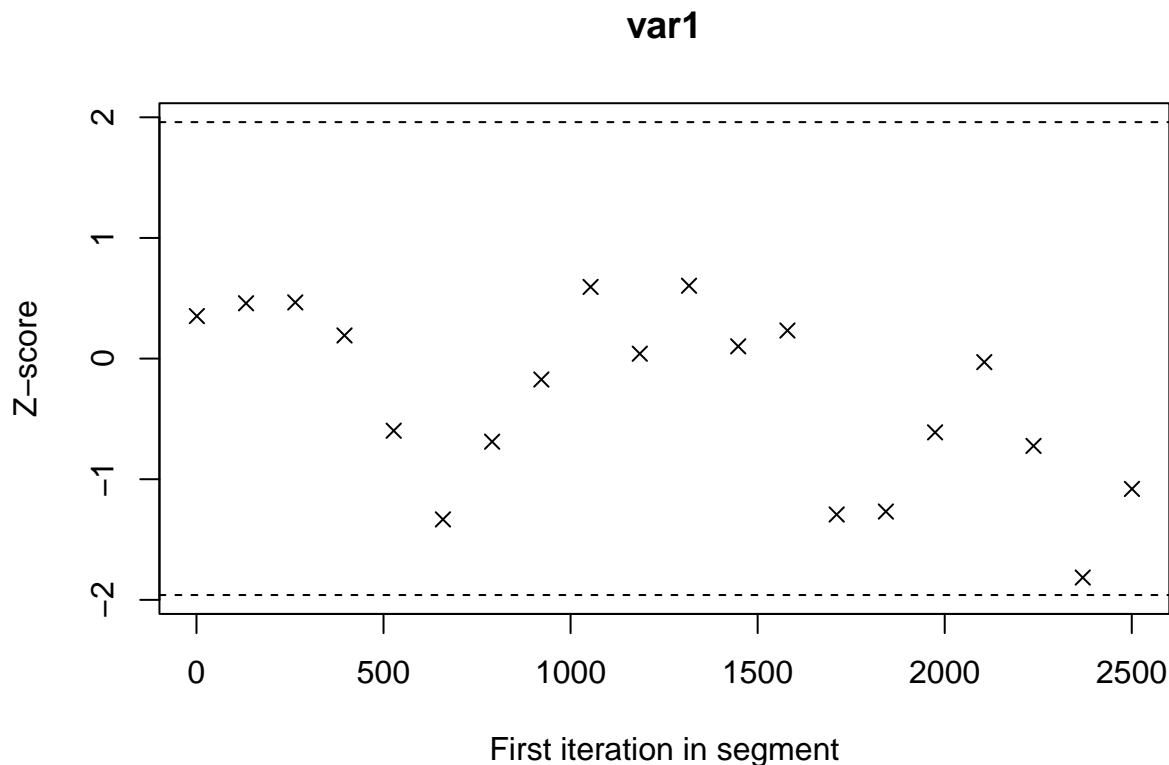
```
# P-value
2*min(pnorm(0.3524), 1-pnorm(0.3524))
```

```
## [1] 0.7245383
```

The p-value indicates that I can not reject the hypothesis of equal means. Therefore I conclude that the chain is stationary.

It is also possible to produce a plot that represents the z-scores of 20 tests of the hypothesis of equal means between the final 50% of the chain and each 1/20 fraction of the initial 50%.

```
geweke.plot(th.post.mc, frac1 = 0.1, frac2 = 0.5, nbins = 20 )
```



I notice that all the z-scores are between $+1.96$ and -1.96 and that there are no specific patterns. These are all signals of stationarity.

To further check the convergence, I will also consider the Gelman and Rubin diagnostic. This approach uses different chains with different starting values and verifies that these chains converge, by looking at the ratio between the within variance and the total variance.

```
# Gelman and Rubin
# Sample new chains and plot them
set.seed(123)
th.post2= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                        n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 24.12941 % of the proposed transition
```

```
th.post3= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                        n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 23.58627 % of the proposed transition
```

```
th.post4= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,  
n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 23.74902 % of the proposed transition
```

```
th.post5= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,  
n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 23.66275 % of the proposed transition
```

```
th.post6= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,  
n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 23.31373 % of the proposed transition
```

```
th.post7= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,  
n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 23.82157 % of the proposed transition
```

```
th.post8= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,  
n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 23.88431 % of the proposed transition
```

```
th.post9= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,  
n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

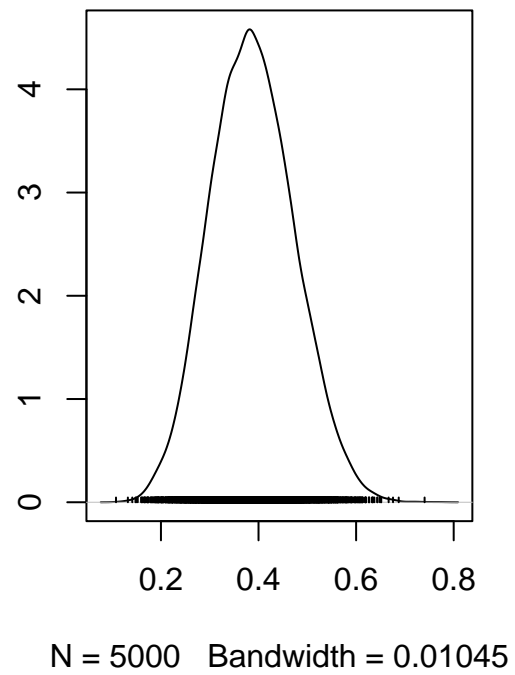
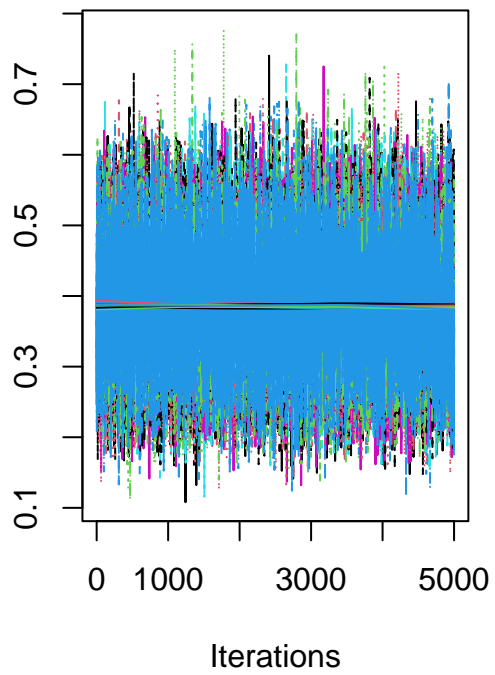
```
## I accepted the 23.93333 % of the proposed transition
```

```
th.post10= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,  
n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

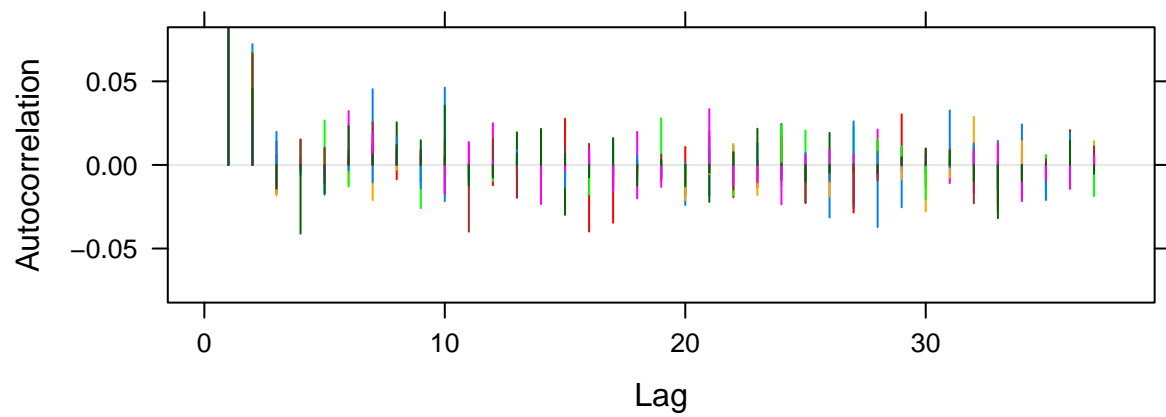
```
## I accepted the 23.58431 % of the proposed transition
```

```
ten.mh <-mcmc.list(th.post.mc,th.post2,th.post3,th.post4,th.post5,th.post6,  
th.post7,th.post8,th.post9,th.post10)
```

```
# Plots of the chains  
plot(ten.mh)
```



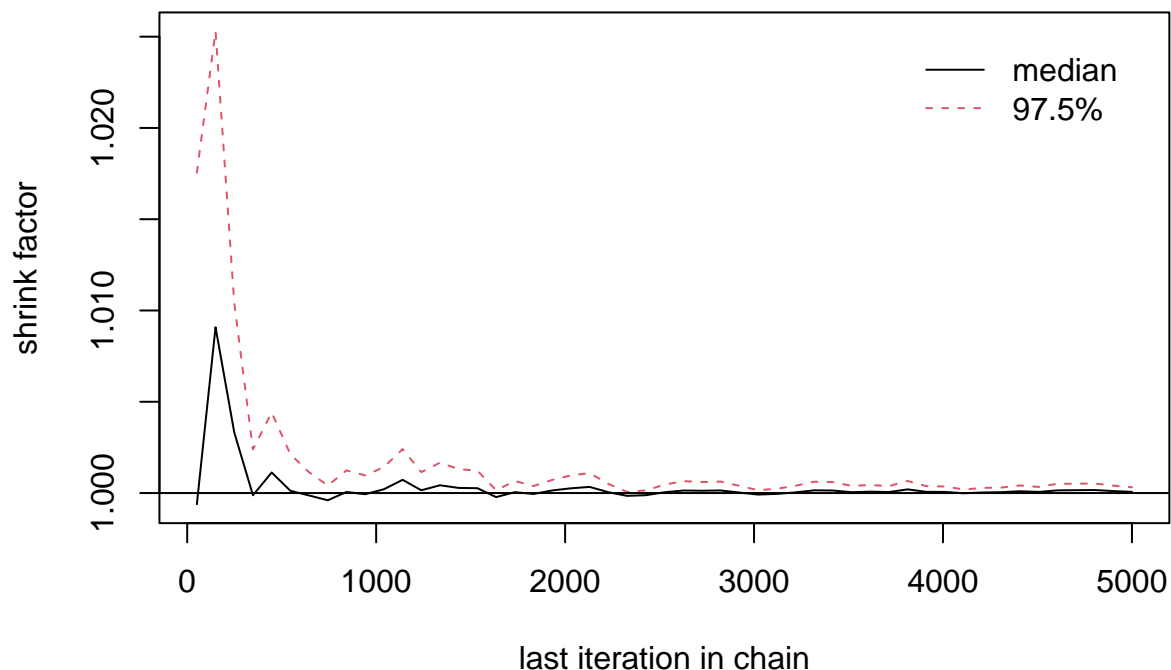
```
acfplot(ten.mh,type="h")
```



```
# Gelman and Rubin diagnostic
gelman.diag(ten.mh)
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
```

```
gelman.plot(ten.mh)
```

The Gelman and Rubin diagnostic exhibits a potential scale reduction factor, that is computed as the ratio between the total variance and the variance within, equal to 1. This indicates that the variance between is 0, which implies stationarity.

Finally, I also compute the effective sample size. This quantity indicates the size of an iid sample that have the same Monte Carlo Error as my MCMC of size “iterations”.

```
#Effective sample size
effectiveSize(th.post.mc) # On the first chain
```

```
##      var1
## 3676.235
```

```
effectiveSize(ten.mh) # On the ten chains
```

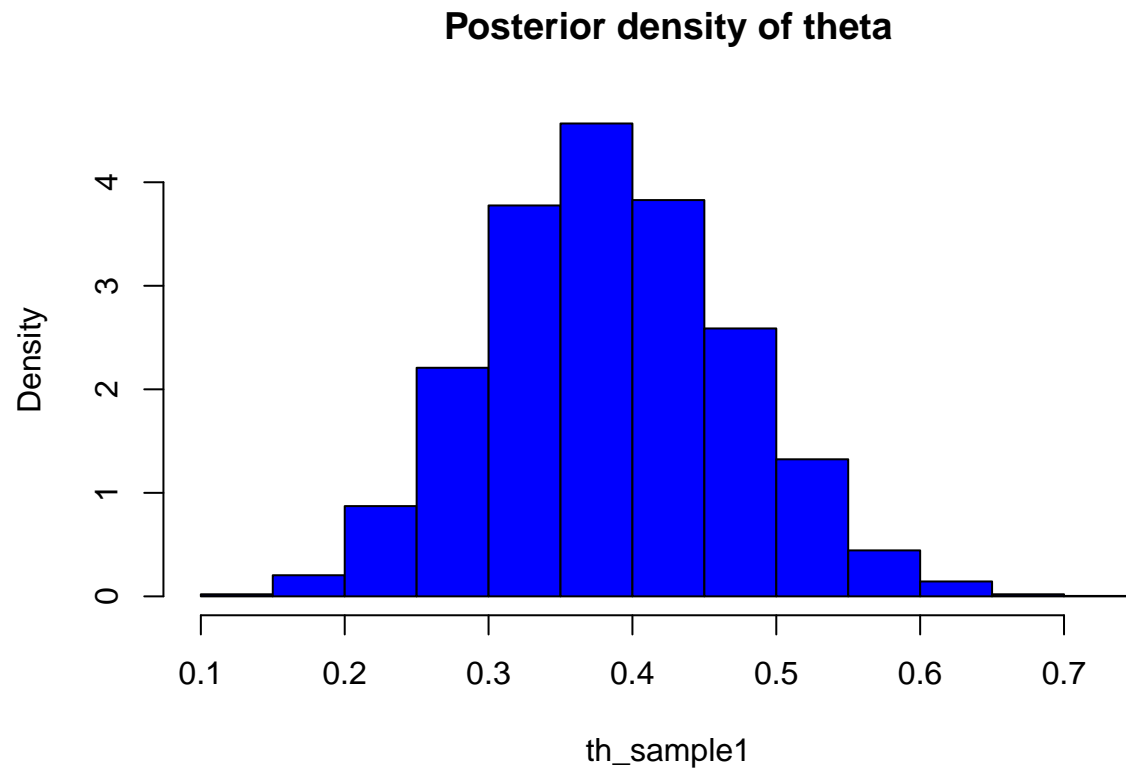
```
##      var1
## 45402.47
```

The effective sample size in both cases is, as expected, lower than the Markov Chain size. Nevertheless, the two sizes are not very different and this is due to the low autocorrelation when the number of lags increase.

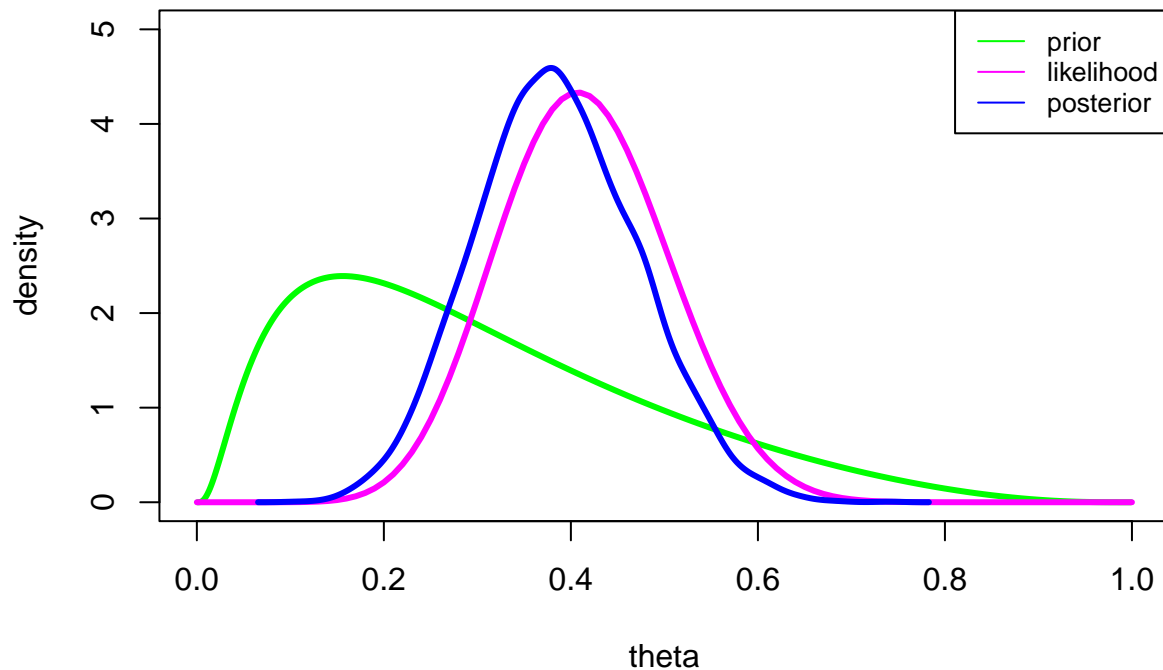
Bayesian analysis

To conduct my bayesian analysis, I first plot the sample that I am considering.

```
# Histogram  
hist(th_sample1,main="Posterior density of theta", freq=F, col="blue")
```



Moreover, I plot also a graph that represent the prior, the likelihood and the posterior of the model.



As expected, also here the posterior is between the likelihood and the prior.

For what concern the relevant quantities of the posterior distribution:

- i. The posterior mean is again between the prior mean and the sample average:

```
# Posterior mean
# I took the prior mean from the table at the beginning
post_mean=mean(th_sample1)
sample_avg= mean(y)
cat("Prior mean: 0.3027046, Posterior mean:",post_mean,"Sample average:",sample_avg)
```

```
## Prior mean: 0.3027046, Posterior mean: 0.3848551 Sample average: 0.4074074
```

- ii. The posterior variance is the following:

```
#Posterior variance
post_var=var(th_sample1)
cat("Posterior variance:",post_var)
```

```
## Posterior variance: 0.007469789
```

- iii. The posterior 95% credible interval is the following:

```
# Posterior 95% credible intervals
i_n=quantile(th_sample1, prob=c(0.025,0.975))
cat("The 95% credible interval of the posterior is:", i_n)
```

```
## The 95% credible interval of the posterior is: 0.2228625 0.5578949
```

iv. The posterior 85th quantile is slightly lower than in the prior case (in which 0.5 was the 84.24th quantile):

```
# Posterior 85th quantile
post_85q=quantile(th_sample1, prob=0.85)
cat("Posterior 85th quantile:",post_85q)
```

```
## Posterior 85th quantile: 0.476631
```

Prediction

A prediction of the number of students that get at least eight hours of sleep in a new sample of 20 students can be performed also using this non conjugate model. This is done by taking a random sample of size 20 from a Bernoulli with parameter equal to the first 20 elements of the chain.

```
# Prediction
G=20
y_star <- vector(length=G)

set.seed(123)
for(g in 1:G){
  y_star[g] <- rbinom(1,1, prob=th_sample1[g])
}

cat("Students of the sample that sleep at least 8 hours:",sum(y_star))
```

```
## Students of the sample that sleep at least 8 hours: 8
```

As in Case 1 of the conjugate model, the number of students of the sample that sleep at least 8 hours is 8.

Hypothesis testing

To conclude this case, I perform a test to verify the strength of data evidence in favor of the hypothesis that “less than 50% of students get at least eight hours of sleep during a week day”.

```
# Hypothesis testing

# Pr(E0|y_1:n)
post_prob0= mean(th_sample1<=0.5)
cat("The posterior probability of E0 is", post_prob0)
```

```
## The posterior probability of E0 is 0.9032
```

```

# Pr(E1|y_1:n)
post_prob1= mean(th_sample1>0.5)

# Pr(E0)
prior_prob0=0.8424 #from the prior table 2

# Pr(E1)
prior_prob1= 1-prior_prob0

# Bayes factor
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
cat("The Bayes Factor is",BF)

```

```
## The Bayes Factor is 1.745607
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.9032$. This probability is higher than the prior probability, that was 0.8424, as displayed in the Table 2.

The Bayes Factor is 1.745607, which is low but higher than 1. This means that the strength of data evidence in favor of E0 is “not worth more than just a bare mention”. However, again there is no evidence in favor of E1. The small value of the Bayes factor is probably due to the fact that the prior and the posterior odds are quite similar and hence the data do not affect the choice of accepting or rejecting the hypothesis that much.

Case 2

Prior hyperparameters choice

In this second case, the hyperparameters have to imply a weakly informative prior.

For this reason, I choose $c=0$, since it yields to $E(\theta) = 0.5$ and to $Pr(\theta < 0.5) = 0.5$ (which represent uncertainty), and $s=4$, such that the prior have the highest variance.

```

# Fix the hyperparameters
c=0
s=4

```

Posterior

Using the previously defined function, I am now able to plot the posterior of this case.

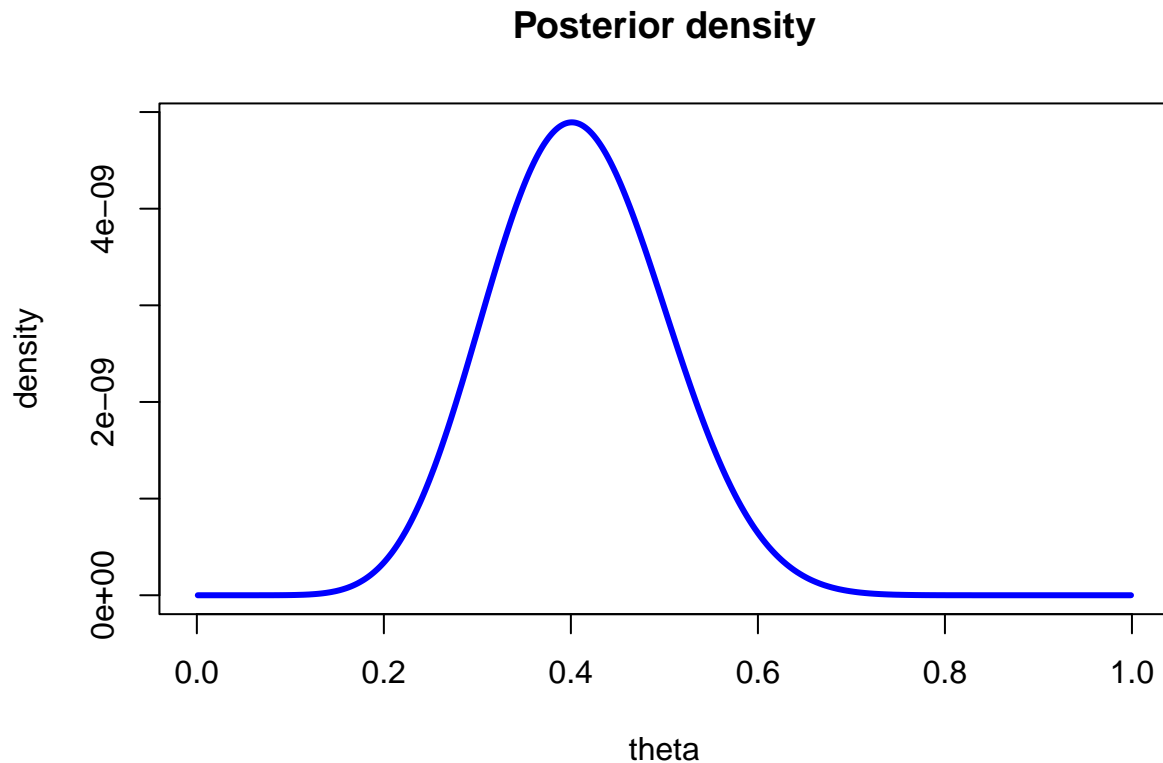
```

# Posterior plot
gr_th=seq(0,1,0.001)
mp=c()

for (i in 1:length(gr_th)){
  mp[i]=model_posterior(gr_th[i], n, M, c, s)
}

plot(gr_th, mp, lwd=3, type="l", col="blue", main="Posterior density",
      xlab="theta", ylab="density")

```



Metropolis-Hastings procedure

I now apply the function for the Metropolis-Hastings procedure that I defined in Case 1 to draw a sample from my posterior. The optimal standard deviation (η) in this case is 0.47, that yields to a frequency of accepted transitions that is near to 0.234.

```
set.seed(123)
th_sample1= MH_post3(G=5000, burnin = 1000, thin = 10, n, M, c, s, eta=0.47, th0=0.5)
```

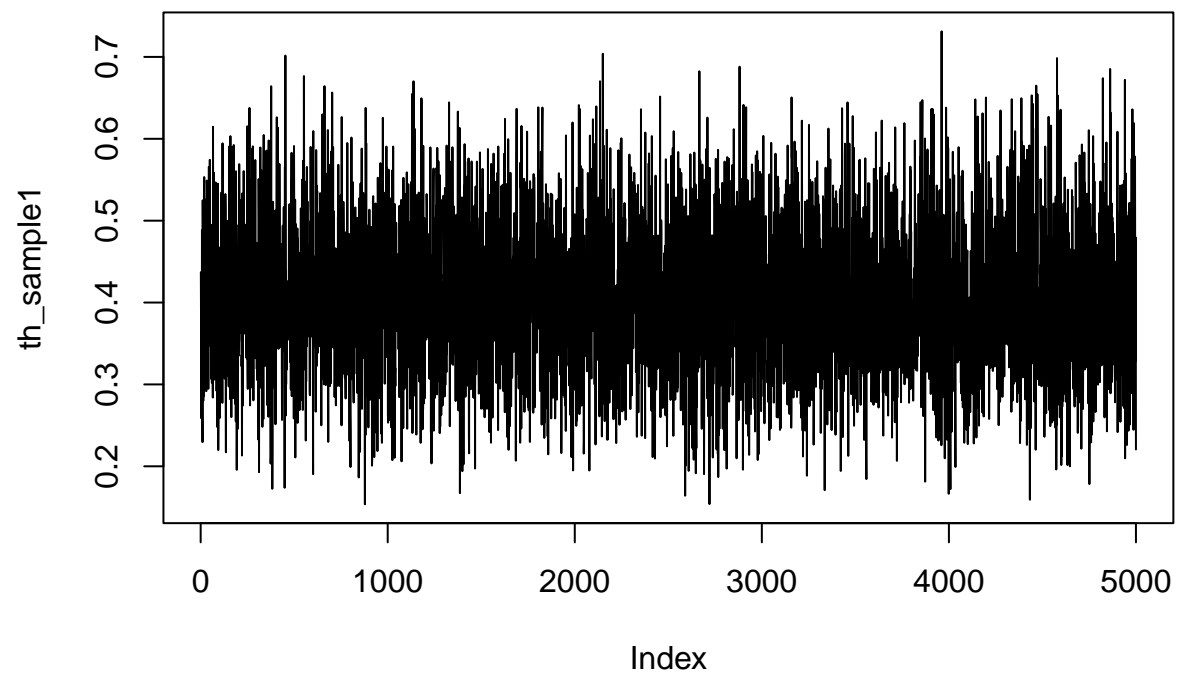
```
## I accepted the 23.63137 % of the proposed transition
```

Convergence check

Also in this case it is appropriate to check the convergence of the chain.

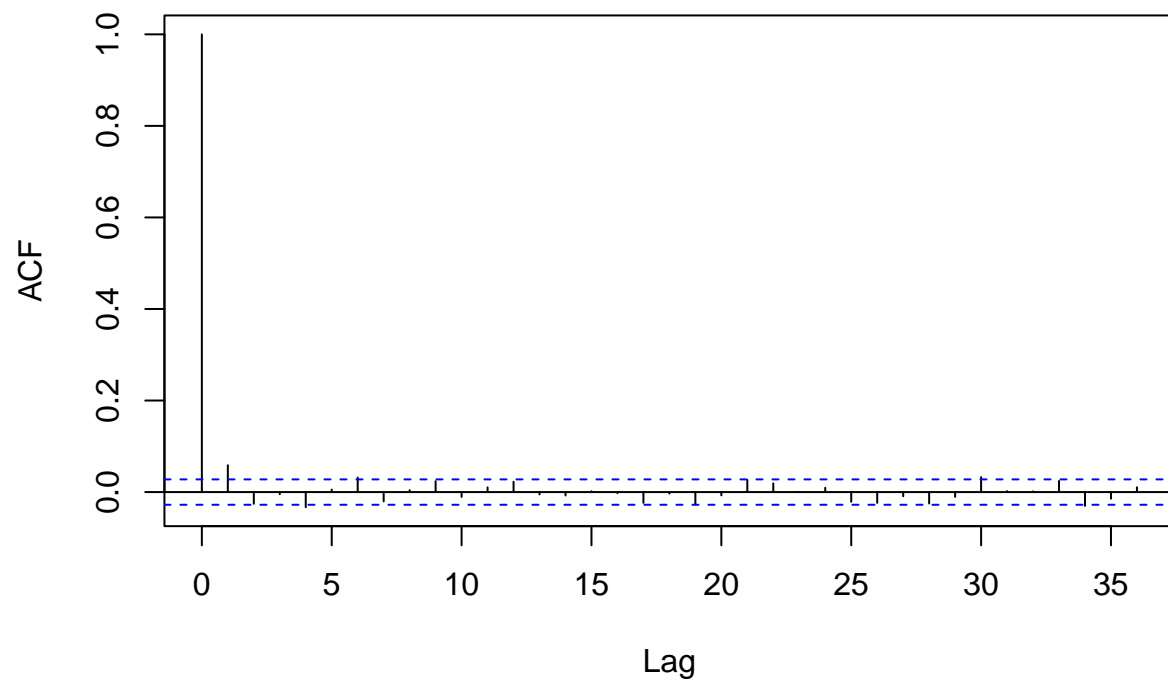
Firstly, I look at the usual plots.

```
# trace plot
plot(th_sample1,type="l")
```

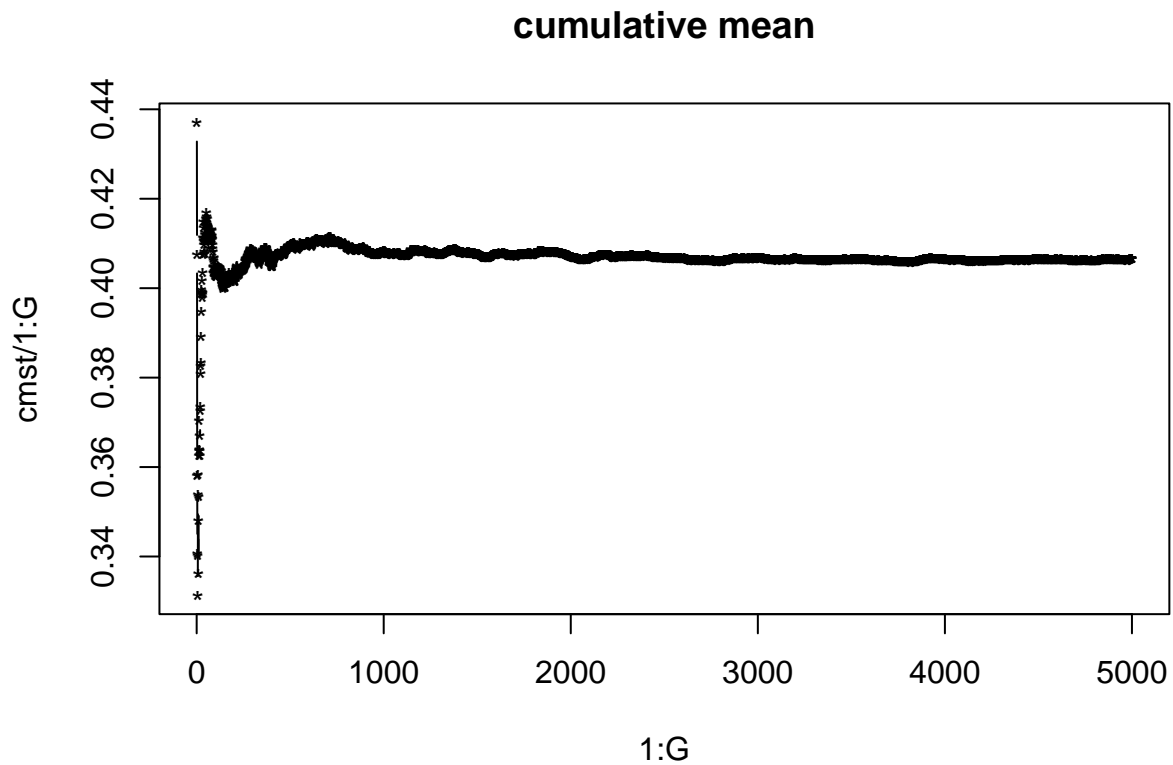


```
# auto correlation function  
acf(th_sample1)
```

Series th_sample1



```
# Cumulative mean  
cmst <- cumsum(th_sample1)  
G <- 5000  
plot(1:G, cmst/1:G, type="b", main="cumulative mean", pch="*")
```

The trace plot shows a good mixing and the autocorrelation plot shows that the autocorrelation drops after one lag and is approximately 0 after two.

The running mean plot shows that the mean does not vary a lot as g increases.

Also in this case, formal diagnostics are needed to verify the convergence of the chain and to perform them the “coda” packages is needed.

```
library(coda)
th.post.mc = mcmc(th_sample1)
```

First I perform a Geweke diagnostic.

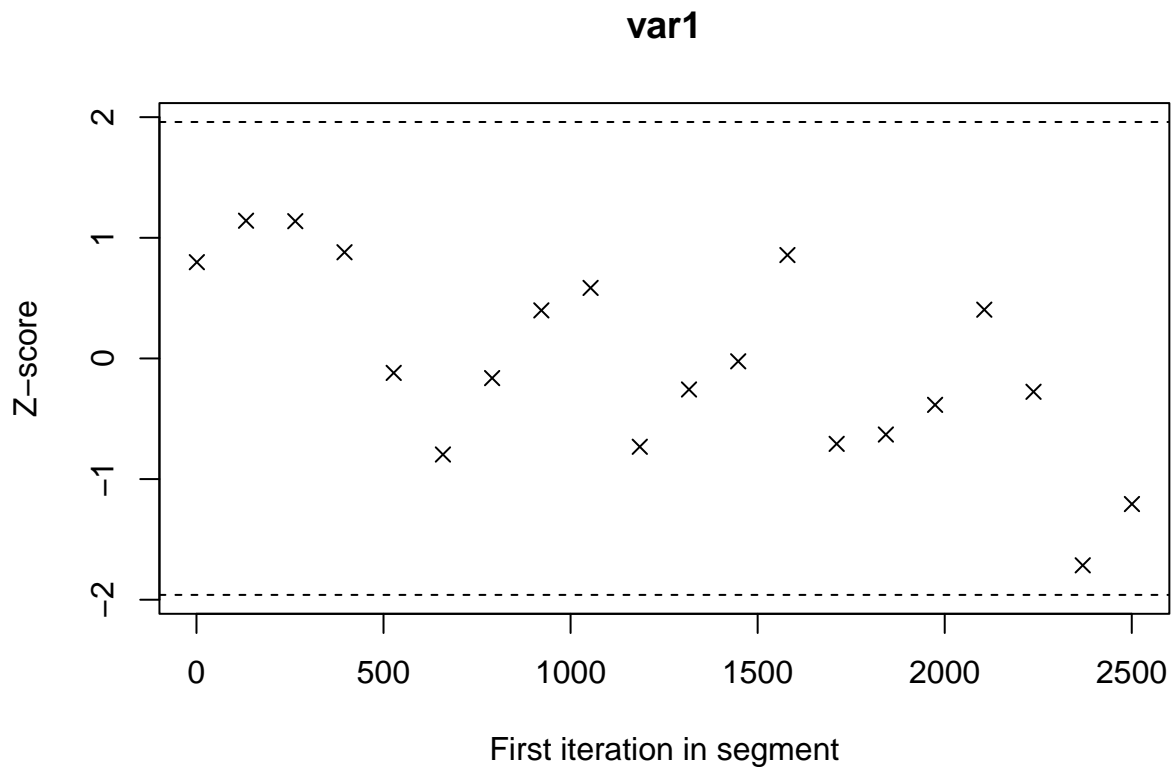
```
# Geweke diagnostic
geweke.diag(th.post.mc)
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## var1
## 0.798
```

```
# P-value
2*min(pnorm(0.798), 1-pnorm(0.798))
```

```
## [1] 0.4248705
```

```
# Geweke plot
geweke.plot(th.post.mc, frac1 = 0.1, frac2 = 0.5, nbins = 20)
```



Also in this case the p-value indicates that it is not possible to reject the hypothesis of equal means and the z-scores in the plot are all between +1.96 and -1.96. This are all signals of stationarity.

Secondly, I perform a Gelman and Rubin diagnostic.

```
# Gelman and Rubin
# Sample new chains and plot them
set.seed(123)
th.post2= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                        n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 25.80784 % of the proposed transition
```

```
th.post3= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                        n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 25.40196 % of the proposed transition

th.post4= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 25.45882 % of the proposed transition

th.post5= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 25.56275 % of the proposed transition

th.post6= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 25.2902 % of the proposed transition

th.post7= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 25.75686 % of the proposed transition

th.post8= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 25.69804 % of the proposed transition

th.post9= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

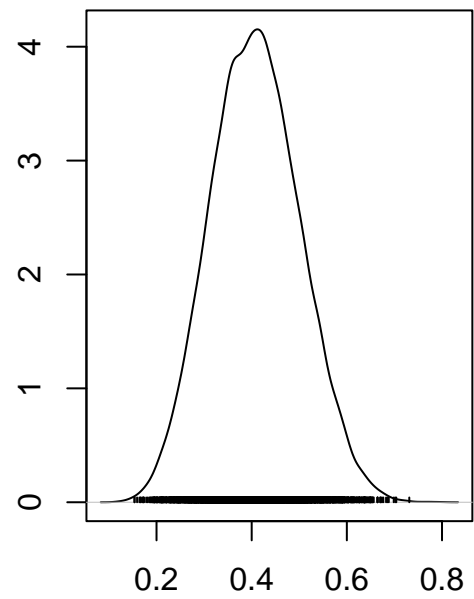
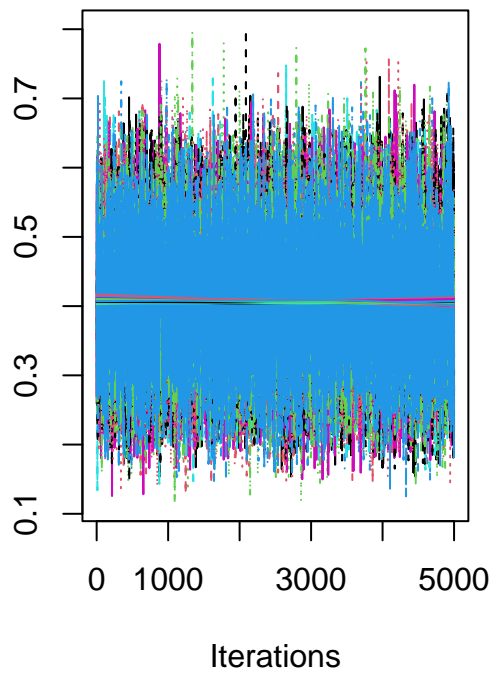
## I accepted the 25.81765 % of the proposed transition

th.post10= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 25.5902 % of the proposed transition

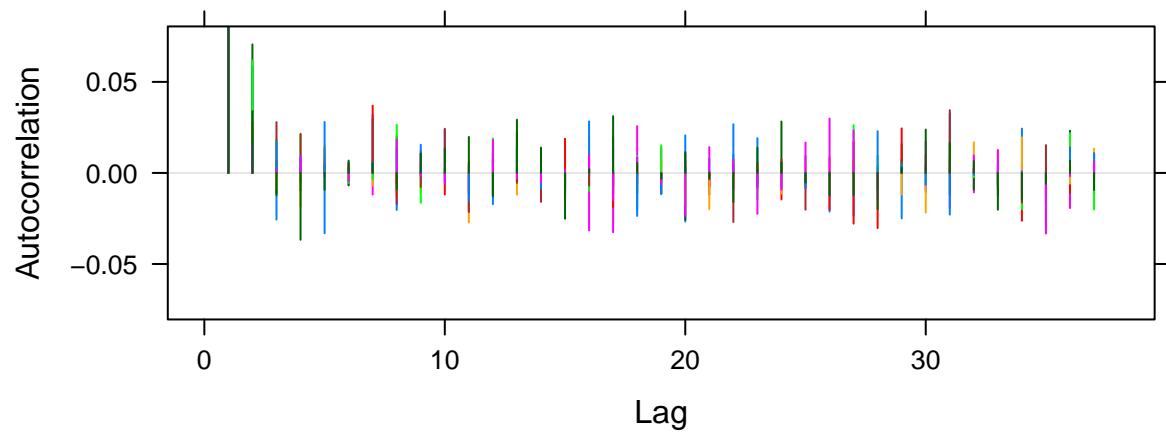
ten.mh <-mcmc.list(th.post.mc,th.post2,th.post3,th.post4,th.post5,th.post6,
                 th.post7,th.post8,th.post9,th.post10)

# Plots of the chains
plot(ten.mh)
```



N = 5000 Bandwidth = 0.01132

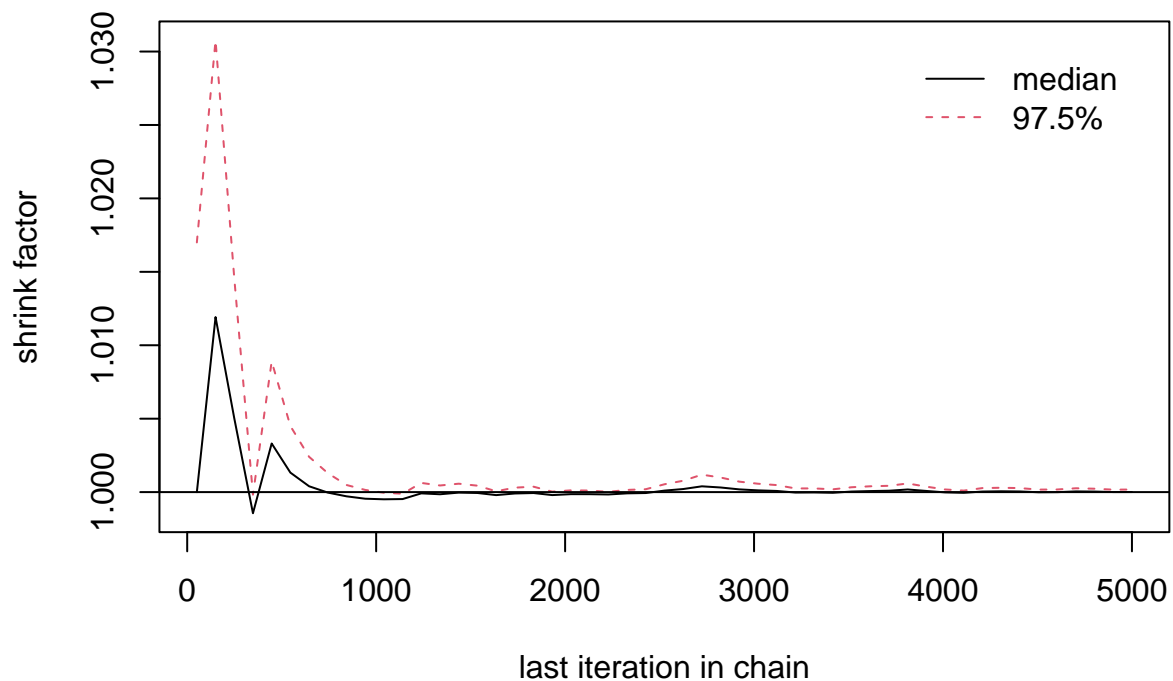
```
acfplot(ten.mh,type="h")
```



```
# Gelman and Rubin diagnostic
gelman.diag(ten.mh)
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
```

```
gelman.plot(ten.mh)
```



Also in this case the potential scale reduction factor is equal to 1 and, hence, I can conclude that the chains are stationary.

Finally, I compute the effective sample size for both the first chain and the sum of all the ten chains.

```
# Effective sample size
effectiveSize(th.post.mc) # On the first chain
```

```
##      var1
## 4902.484
```

```
effectiveSize(ten.mh) # On the ten chains
```

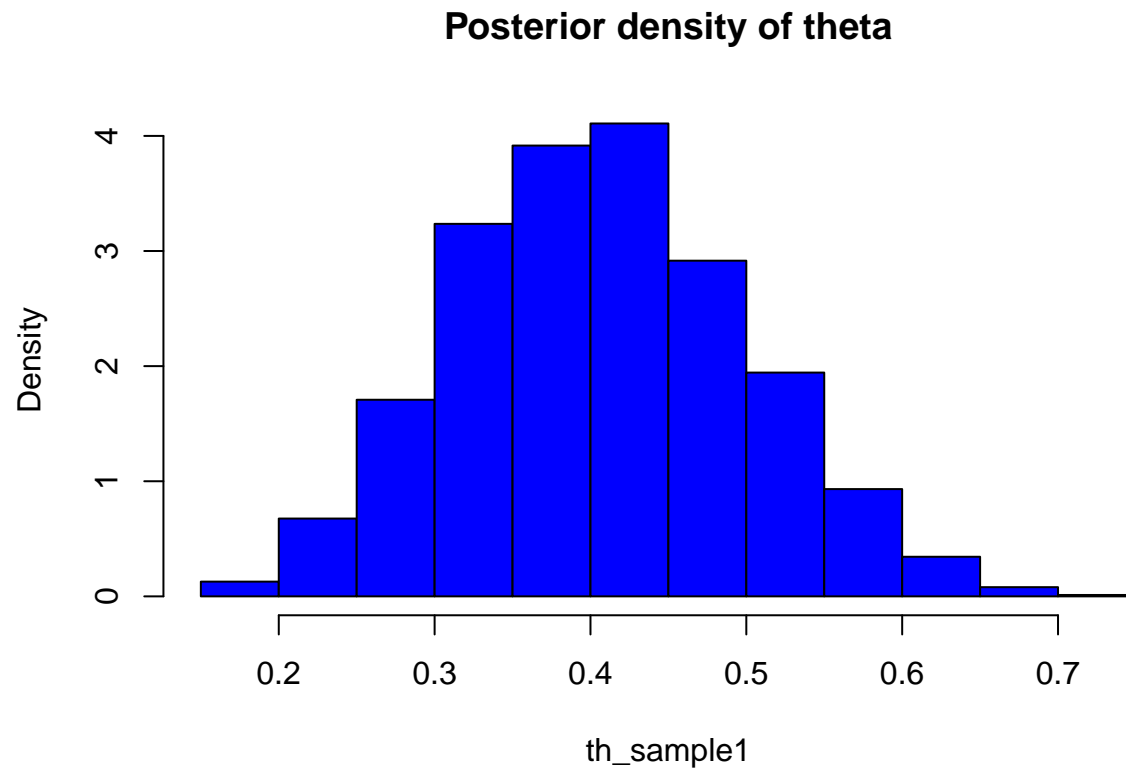
```
##      var1
## 48089.35
```

The effective sample size is very similar to the number of iterations of the Metropolis-Hastings, due to the low autocorrelation.

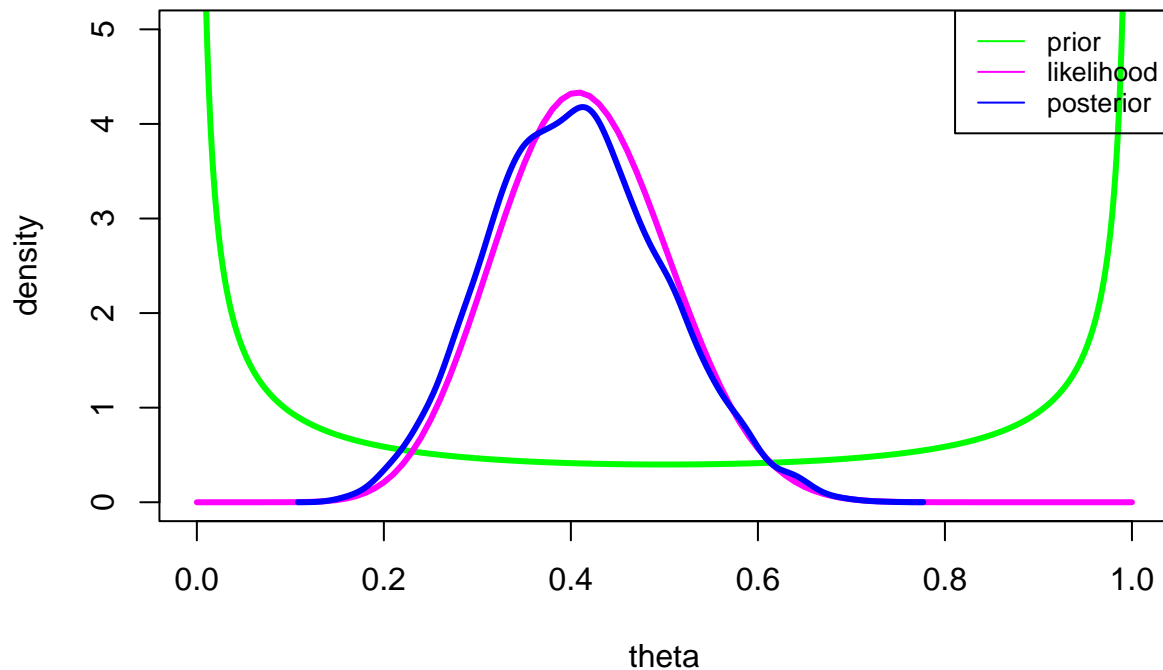
Bayesian analysis

To conduct my bayesian analysis, I first plot the sample that I am considering.

```
# Histogram  
hist(th_sample1,main="Posterior density of theta", freq=F, col="blue")
```



Moreover, I plot also a graph that represent the prior, the likelihood and the posterior of the model.



In this case, the fact that the posterior is between the likelihood and the prior is less visible.

For what concern the relevant quantities of the posterior distribution:

- i. The posterior mean is very similar to the sample average in this case, because the prior is not very informative.

```
# Posterior mean
# I took the prior mean from the table at the beginning
post_mean=mean(th_sample1)
sample_avg= mean(y)
cat("Prior mean: 0.4990392, Posterior mean:",post_mean,"Sample average:",sample_avg)
```

```
## Prior mean: 0.4990392, Posterior mean: 0.4065012 Sample average: 0.4074074
```

- ii. The posterior variance is the following:

```
#Posterior variance
post_var=var(th_sample1)
cat("Posterior variance:",post_var)
```

```
## Posterior variance: 0.008607036
```

- iii. The posterior 95% credible interval is the following:


```
# Posterior 95% credible intervals
i_n=quantile(th_sample1, prob=c(0.025,0.975))
cat("The 95% credible interval of the posterior is:", i_n)
```

```
## The 95% credible interval of the posterior is: 0.2357308 0.5934881
```

iv. The posterior 85th quantile is:

```
# Posterior 85th quantile
post_85q=quantile(th_sample1, prob=0.85)
cat("Posterior 85th quantile:",post_85q)
```

```
## Posterior 85th quantile: 0.5055941
```

Prediction

I now predict the number of students that get at least eight hours of sleep in a new sample of 20 students can be performed also for this case.

```
# Prediction
G=20
y_star <- vector(length=G)

set.seed(123)
for(g in 1:G){
  y_star[g] <- rbinom(1,1, prob=th_sample1[g])
}

cat("Students of the sample that sleep at least 8 hours:",sum(y_star))
```

```
## Students of the sample that sleep at least 8 hours: 9
```

In this case, the number of students of the sample that sleep at least 8 hours is 9.

Hypothesis testing

To conclude, I perform a test to verify the strength of data evidence in favor of the hypothesis that “less than 50% of students get at least eight hours of sleep during a week day”.

```
# Hypothesis testing

# Pr(E0|y_1:n)
post_prob0= mean(th_sample1<=0.5)
cat("The posterior probability of E0 is", post_prob0)
```

```
## The posterior probability of E0 is 0.8344
```

```

# Pr(E1|y_1:n)
post_prob1= mean(th_sample1>0.5)

# Pr(E0)
prior_prob0=0.5041 #from the prior table 2

# Pr(E1)
prior_prob1= 1-prior_prob0

# Bayes factor
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
cat("The Bayes Factor is",BF)

```

```
## The Bayes Factor is 4.956686
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.8344$. This probability is lower than the posterior probability of Case 1, as a result of the increased prior $E(\theta)$.

The Bayes Factor is 4.956686, which means that the strength of data evidence in favor of E_0 is now “substantial”. This increased evidence in favor of E_0 is a result of the increased difference between the prior and the posterior odds, meaning that the data support E_0 much more than the prior does. This is to impute to the non informative hyperparameters choice.

Case 3

Prior hyperparameters choice

In this third case, the hyperparameters have to imply a prior informatio that is opposite to the on discussed in the “Prior information” section.

For this reason, I choose $c=1$ and $s=1$, that yield to $E(\theta) = 0.6963821$ (opposite with respect to the 0.3 cosidered in the first case) and to $Pr(\theta < 0.5) = 0.1612$ (opposite to the 0.8424 considered in the first case).

```

# Fix the hyperparameters
c=1
s=1

```

Posterior

Using the previously defined function, I am now able to plot the posterior of this case.

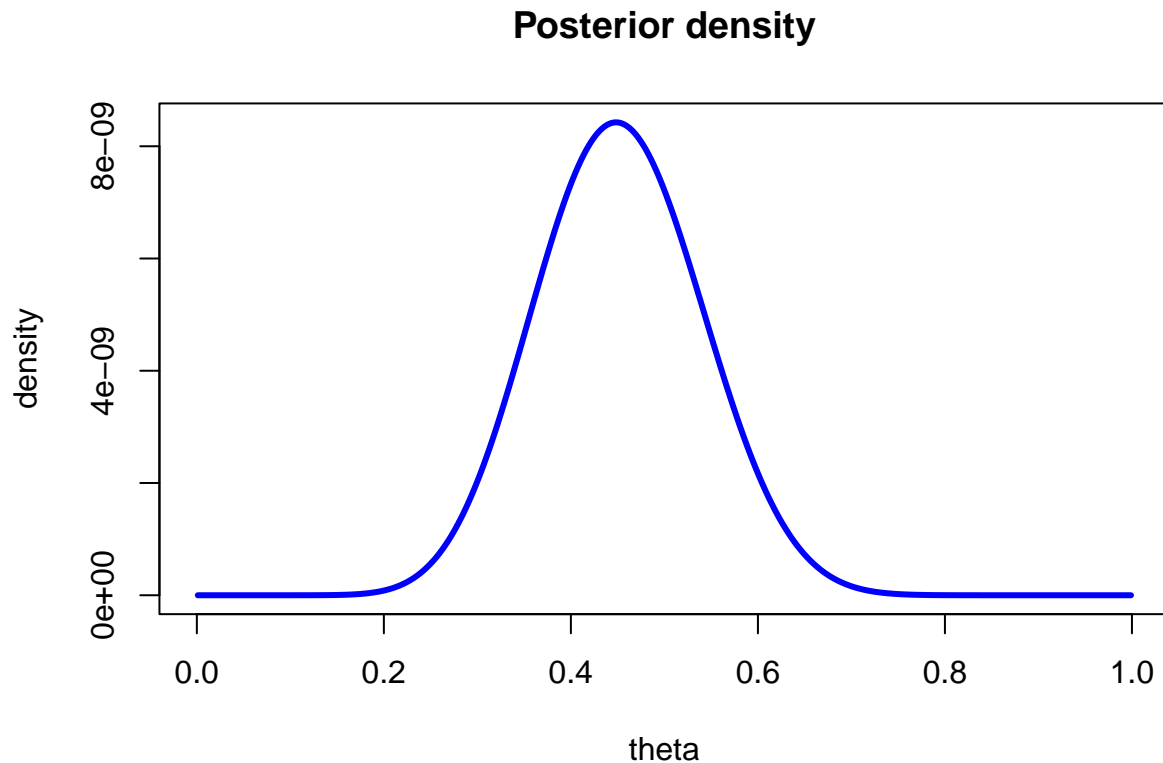
```

# Posterior plot
gr_th=seq(0,1,0.001)
mp=c()

for (i in 1:length(gr_th)){
  mp[i]=model_posterior(gr_th[i], n, M, c, s)
}

plot(gr_th, mp, lwd=3, type="l", col="blue", main="Posterior density",
     xlab="theta", ylab="density")

```



Metropolis-Hastings procedure

I now apply the function for the Metropolis-Hastings procedure that I defined in Case 1 to draw a sample from my posterior. The optimal standard deviation (η) in this case is 0.45, that yields to a frequency of accepted transitions that is near to 0.234.

```
set.seed(123)
th_sample1= MH_post3(G=5000, burnin = 1000, thin = 10, n, M, c, s, eta=0.45, th0=0.5)
```

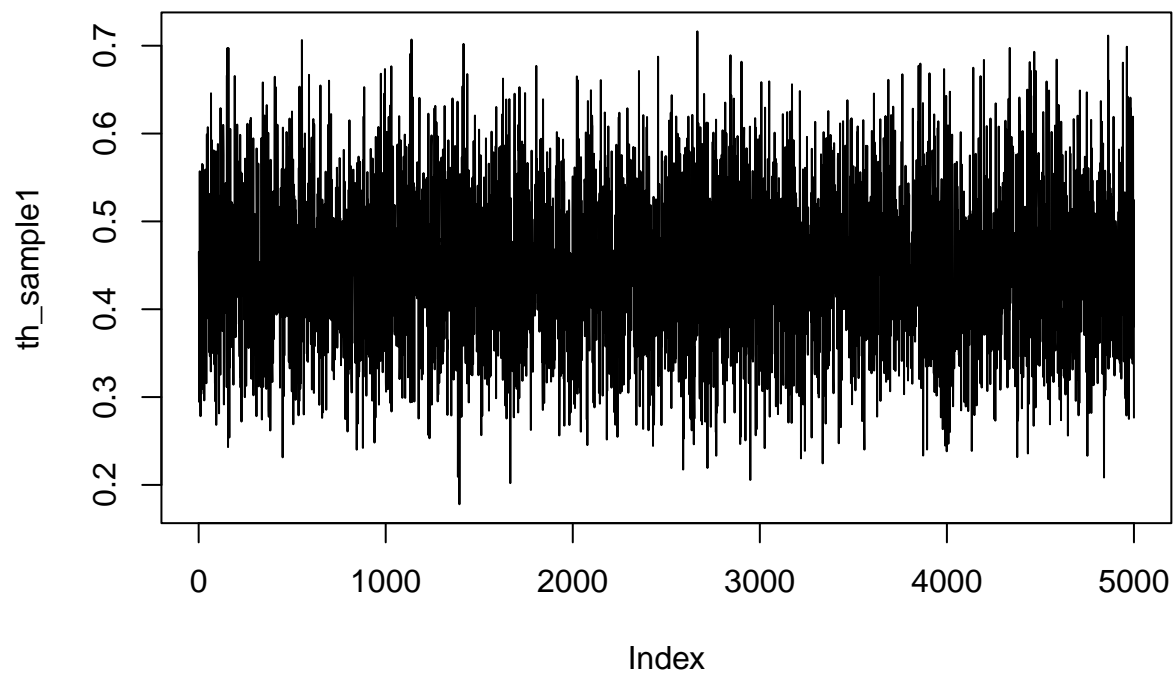
```
## I accepted the 23.39804 % of the proposed transition
```

Convergence check

Also in this case it is appropriate to check the convergence of the chain.

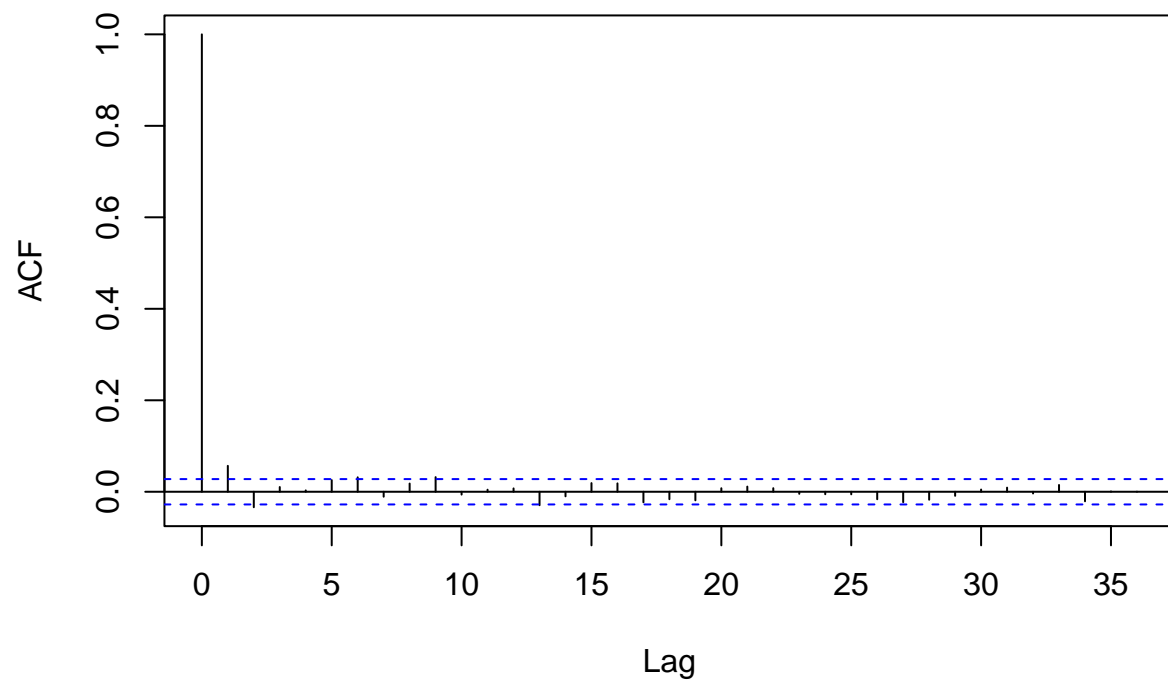
Firstly, I look at the usual plots.

```
# trace plot
plot(th_sample1,type="l")
```

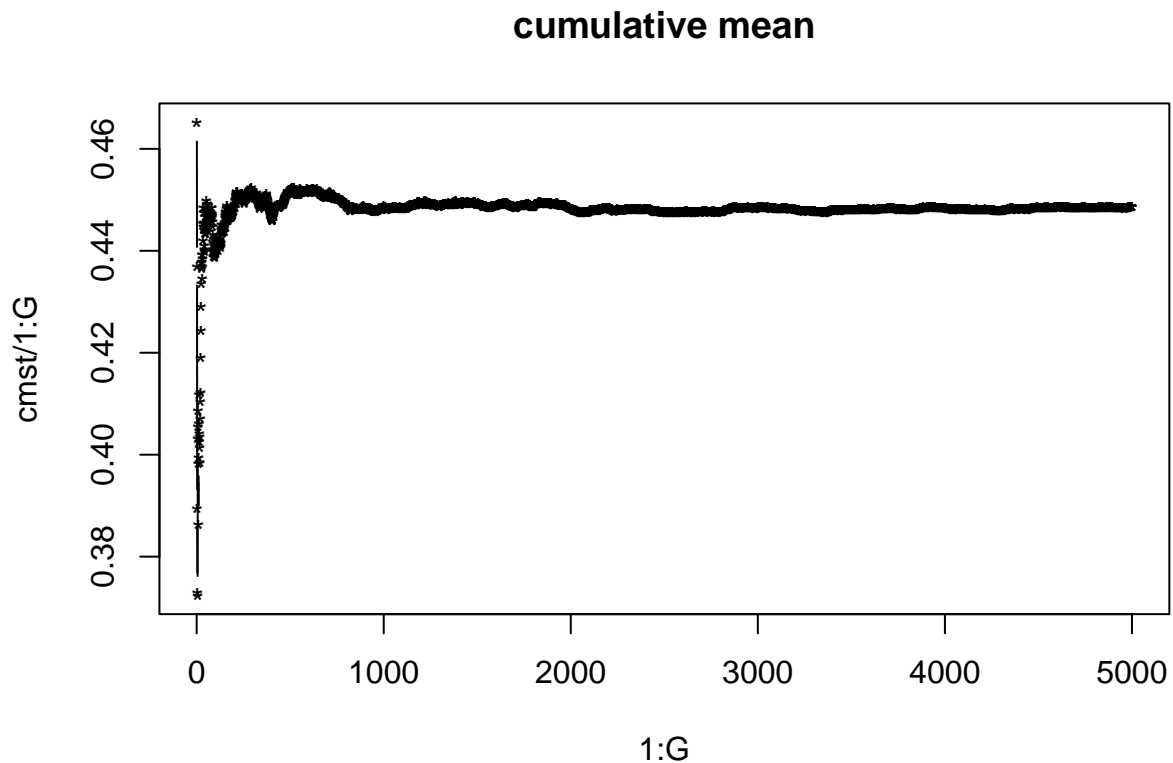


```
# auto correlation function  
acf(th_sample1)
```

Series th_sample1



```
# Cumulative mean
cmst <- cumsum(th_sample1)
G <- 5000
plot(1:G, cmst/1:G, type="b", main="cumulative mean", pch="*")
```



The trace plot shows a good mixing and the autocorrelation plot shows that the autocorrelation drops after one lag and is approximately 0 after two, as in the other cases.

The running mean plot shows that the running mean stabilizes as g increases.

Also in this case, formal diagnostics are needed to verify the convergence of the chain and to perform them the “coda” packages is needed.

```
library(coda)
th.post.mc= mcmc(th_sample1)
```

First I perform a Geweke diagnostic.

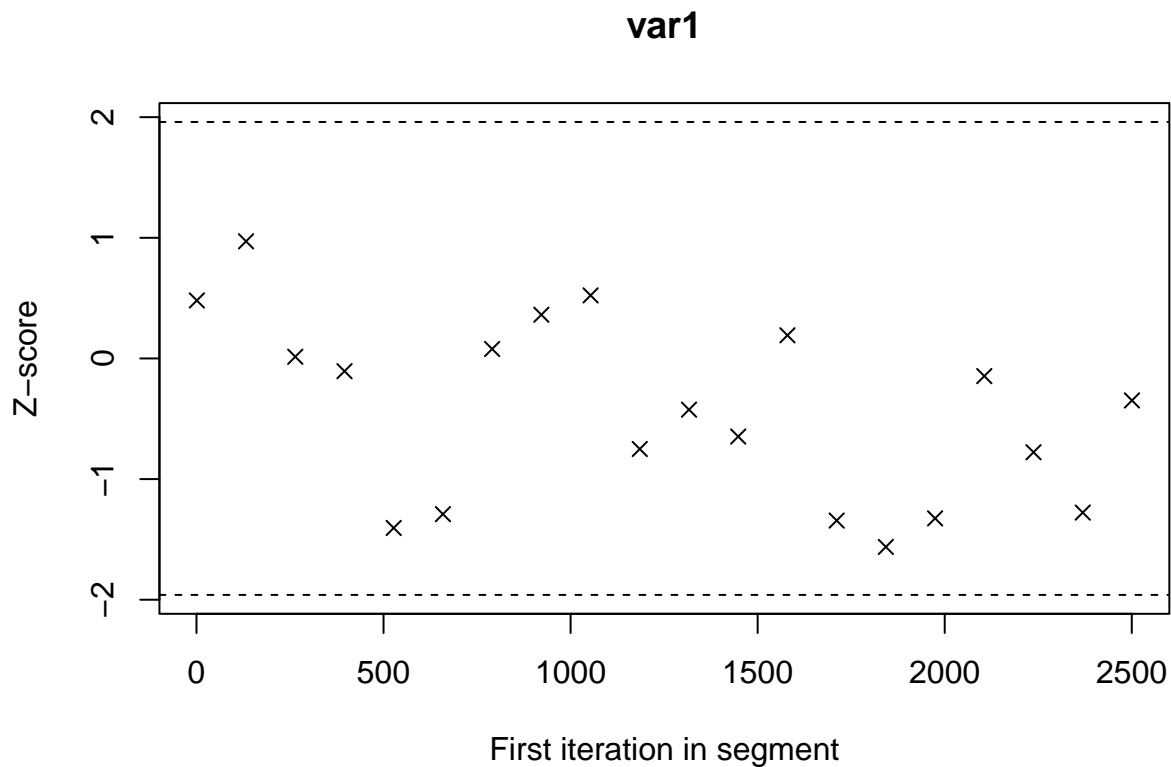
```
# Geweke diagnostic
geweke.diag(th.post.mc)
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1
## 0.4806
```

```
# P-value
2*min(pnorm(0.4806), 1-pnorm(0.4806))
```

```
## [1] 0.6308008
```

```
# Geweke plot
geweke.plot(th.post.mc, frac1 = 0.1, frac2 = 0.5, nbins = 20)
```



Also in this case the p-value indicates that it is not possible to reject the hypothesis of equal means and the z-scores in the plot are all between $+1.96$ and -1.96 . This are all signals of stationarity.

Secondly, I perform a Gelman and Rubin diagnostic.

```
# Gelman and Rubin
# Sample new chains and plot them
set.seed(123)
th.post2= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                        n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 24.67255 % of the proposed transition
```

```
th.post3= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                        n, M, c, s, eta=0.44, th0= runif(1,0,1)))
```

```
## I accepted the 24.14902 % of the proposed transition

th.post4= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 24.17451 % of the proposed transition

th.post5= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 24.36863 % of the proposed transition

th.post6= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 23.97255 % of the proposed transition

th.post7= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 24.51373 % of the proposed transition

th.post8= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 24.54314 % of the proposed transition

th.post9= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

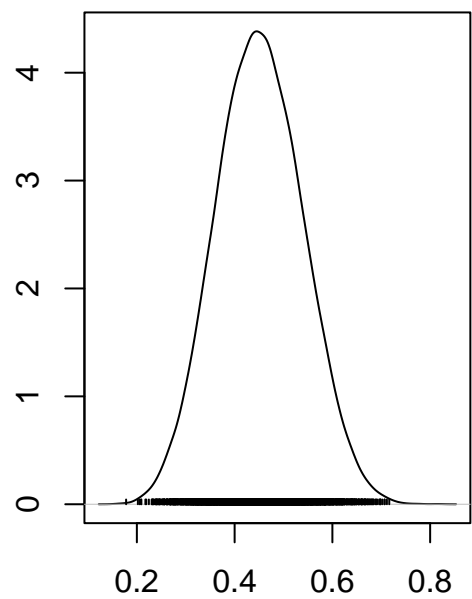
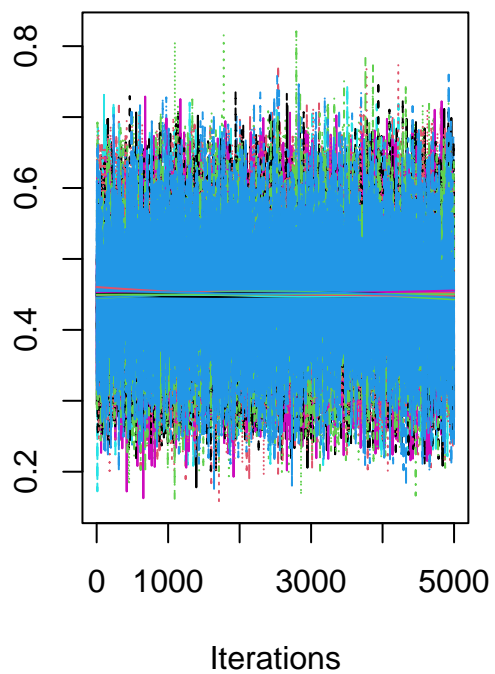
## I accepted the 24.62745 % of the proposed transition

th.post10= mcmc(MH_post3(G=5000, burnin = 1000, thin = 10,
                      n, M, c, s, eta=0.44, th0= runif(1,0,1)))

## I accepted the 24.42157 % of the proposed transition

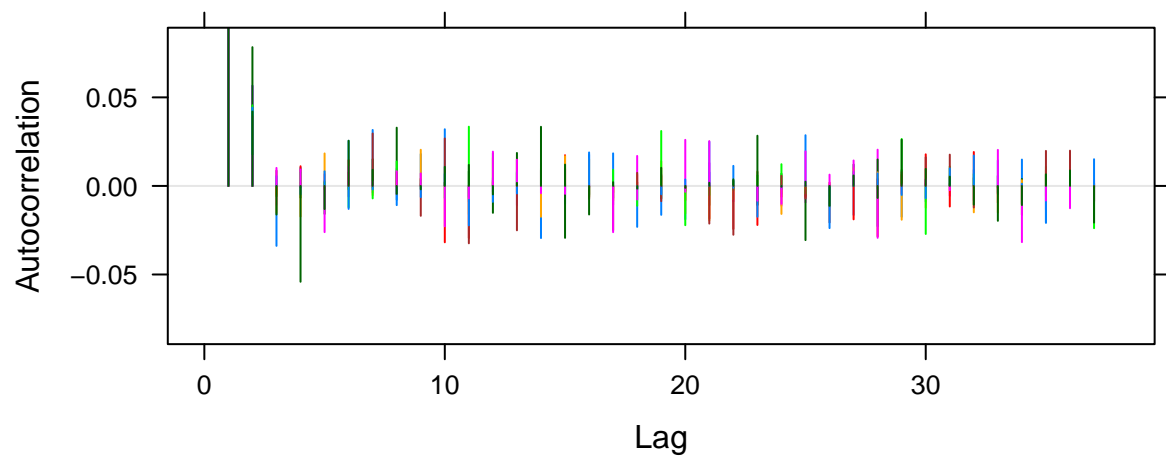
ten.mh <-mcmc.list(th.post.mc,th.post2,th.post3,th.post4,th.post5,th.post6,
                 th.post7,th.post8,th.post9,th.post10)

# Plots of the chains
plot(ten.mh)
```

N = 5000 Bandwidth = 0.01073

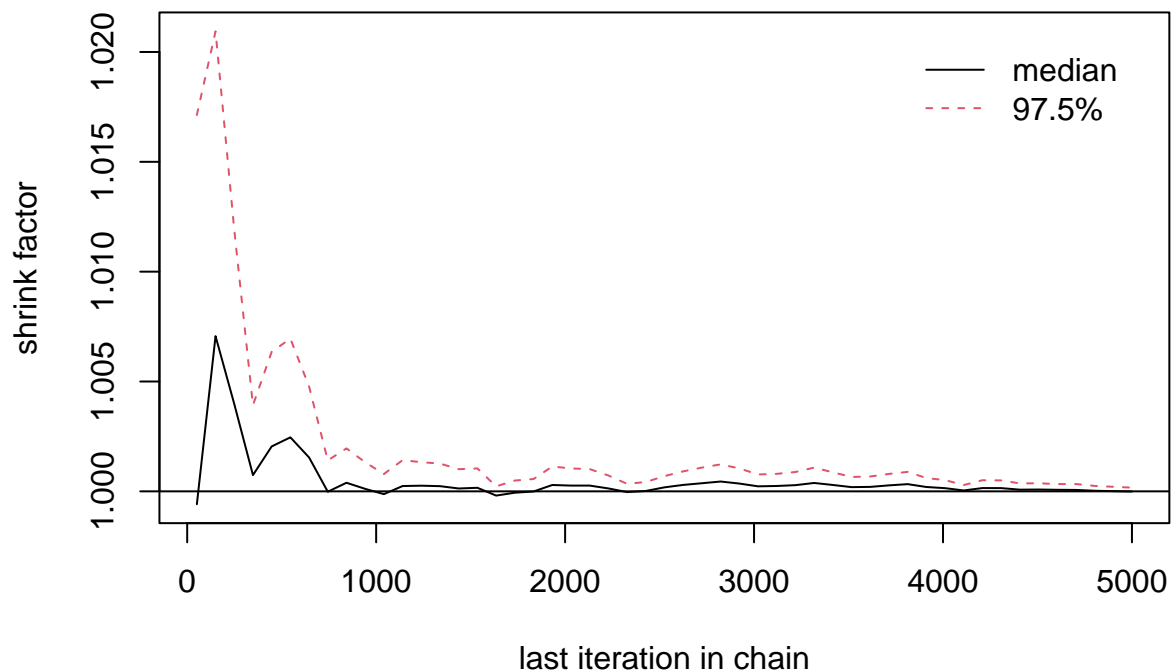
```
acfplot(ten.mh,type="h")
```



```
# Gelman and Rubin diagnostic
gelman.diag(ten.mh)
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
```

```
gelman.plot(ten.mh)
```



In this case, the Gelman plot shows that it takes more iterations to bring the potential scale reduction factor to 1, compared to the two previous cases. However, at the end, the potential scale reduction factor is equal to 1 and, hence, I can conclude that the chains are stationary.

Finally, I compute the effective sample size for both the first chain and the sum of all the ten chains.

```
# Effective sample size
effectiveSize(th.post.mc) # On the first chain
```

```
##      var1
## 3878.497
```

```
effectiveSize(ten.mh) # On the ten chains
```

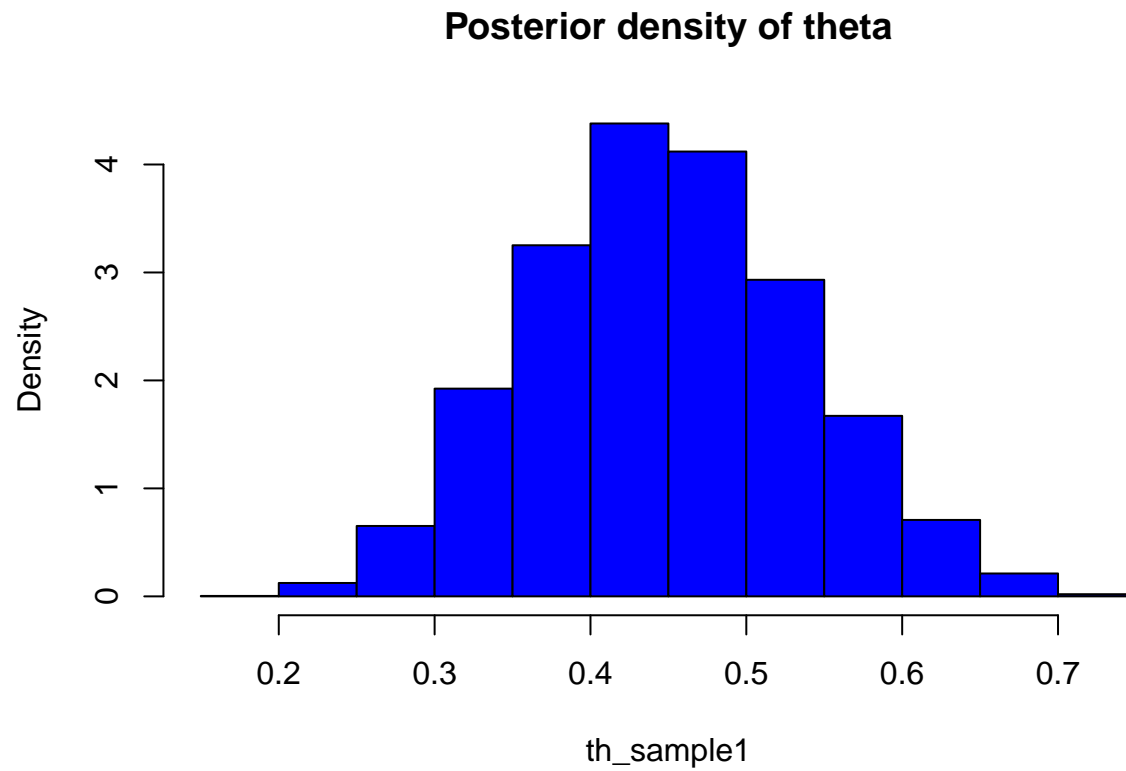
```
##      var1
## 46286.73
```

The effective sample size is very similar to the number of iterations of the Metropolis-Hastings, due to the low autocorrelation.

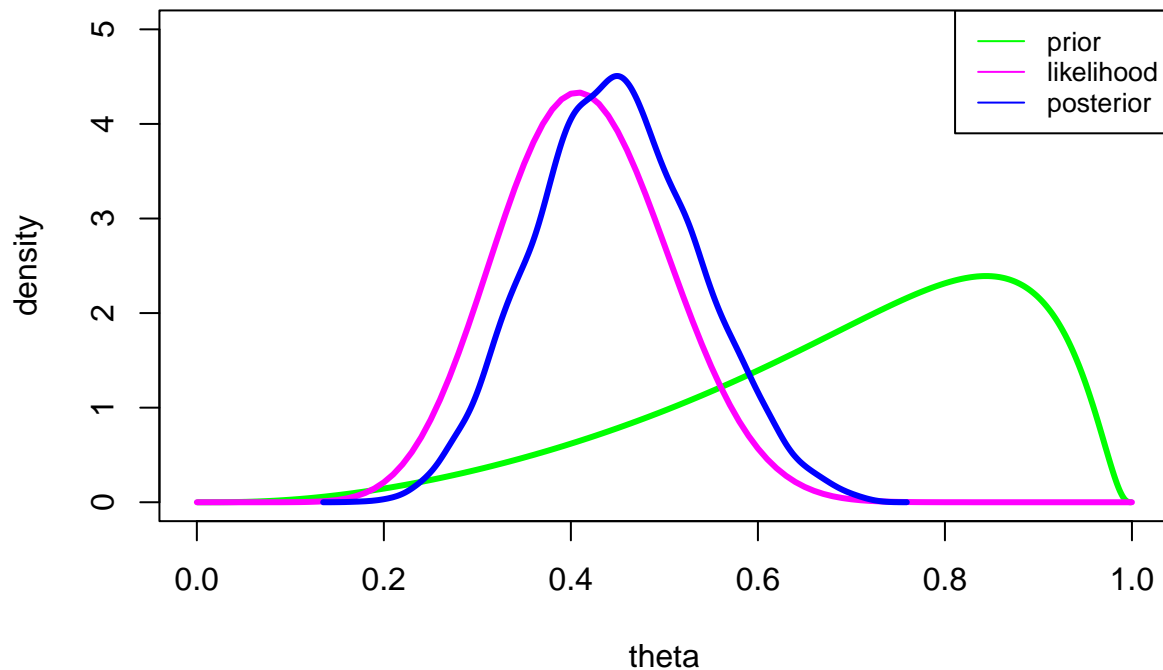
Bayesian analysis

To conduct my bayesian analysis, I first plot the sample that I am considering.

```
# Histogram  
hist(th_sample1,main="Posterior density of theta", freq=F, col="blue")
```



Moreover, I plot a graph that represent the prior, the likelihood and the posterior of the model.



As expected, the posterior is between the prior and the likelihood.

For what concern the relevant quantities of the posterior distribution:

- i. The posterior mean is between the prior mean and the sample average:

```
# Posterior mean
# I took the prior mean from the table at the beginning
post_mean=mean(th_sample1)
sample_avg= mean(y)
cat("Prior mean: 0.6963821, Posterior mean:",post_mean,"Sample average:",sample_avg)
```

```
## Prior mean: 0.6963821, Posterior mean: 0.4486973 Sample average: 0.4074074
```

- ii. The posterior variance is the following:

```
#Posterior variance
post_var=var(th_sample1)
cat("Posterior variance:",post_var)
```

```
## Posterior variance: 0.007605313
```

- iii. The posterior 95% credible interval is the following:

```
# Posterior 95% credible intervals
i_n=quantile(th_sample1, prob=c(0.025,0.975))
cat("The 95% credible interval of the posterior is:", i_n)
```

```
## The 95% credible interval of the posterior is: 0.282946 0.6216746
```

- iv. The posterior 16th quantile is significantly lower than in the prior case (in which it was approximately 0.5, as reported in Table 2):

```
# Posterior 16th quantile
post_16q=quantile(th_sample1, prob=0.16)
cat("Posterior 16th quantile:", post_16q)
```

```
## Posterior 16th quantile: 0.3599375
```

Prediction

I now predict the number of students that get at least eight hours of sleep in a new sample of 20 students can be performed also for this case.

```
# Prediction
G=20
y_star <- vector(length=G)

set.seed(123)
for(g in 1:G){
  y_star[g] <- rbinom(1,1, prob=th_sample1[g])
}

cat("Students of the sample that sleep at least 8 hours:", sum(y_star))
```

```
## Students of the sample that sleep at least 8 hours: 9
```

As in Case 2, the number of students of the sample that sleep at least 8 hours is 9.

Hypothesis testing

To conclude, I perform a test to verify the strength of data evidence in favor of the hypothesis that “less than 50% of students get at least eight hours of sleep during a week day”.

```
# Hypothesis testing

# Pr(E0|y_1:n)
post_prob0= mean(th_sample1<=0.5)
cat("The posterior probability of E0 is", post_prob0)
```

```
## The posterior probability of E0 is 0.7228
```

```

# Pr(E1|y_1:n)
post_prob1= mean(th_sample1>0.5)

# Pr(E0)
prior_prob0=0.1612 #from the prior table 2

# Pr(E1)
prior_prob1= 1-prior_prob0

# Bayes factor
BF= (post_prob0/post_prob1)/(prior_prob0/prior_prob1)
cat("The Bayes Factor is",BF)

```

```
## The Bayes Factor is 13.56808
```

The posterior probability that “less than 50% of students get at least eight hours of sleep during a week day” is: $Pr(\theta \leq 0.5|y_{1:n}) = 0.7228$. This probability is lower than the posterior probability of Cases 1 and 2, as a result of the increased prior $E(\theta)$.

The Bayes Factor is 13.56808, which means that the strength of data evidence in favor of E_0 is now “strong”. This increased evidence in favor of E_0 is, as in case 2, a result of the increased difference between the prior and the posterior odds.

In particular, in this third case, the data supported more E_0 (the sample average was $0.4074074 < 0.5$), whereas the prior supported more E_1 (the prior mean was $0.6963821 > 0.5$)