HW2 - Tardella - part I: exercises 1, 2 and 3

Your Last+First Name: Rossini Valerio, Your Matricola: 1613638

Part 1

Dugongs - part I

1a answer)

We are doing a nonconjugate Bayesian analysis of a data set that cointains information about dugongs. The data are length (the dependent variable) and age (the independent variable) measurements for 27 (i.e. we have n = 27) captured dugongs, and they are modelled with the use of a nonlinear growth curve with no inflection point and an asymptote as X_i tends to infinity:

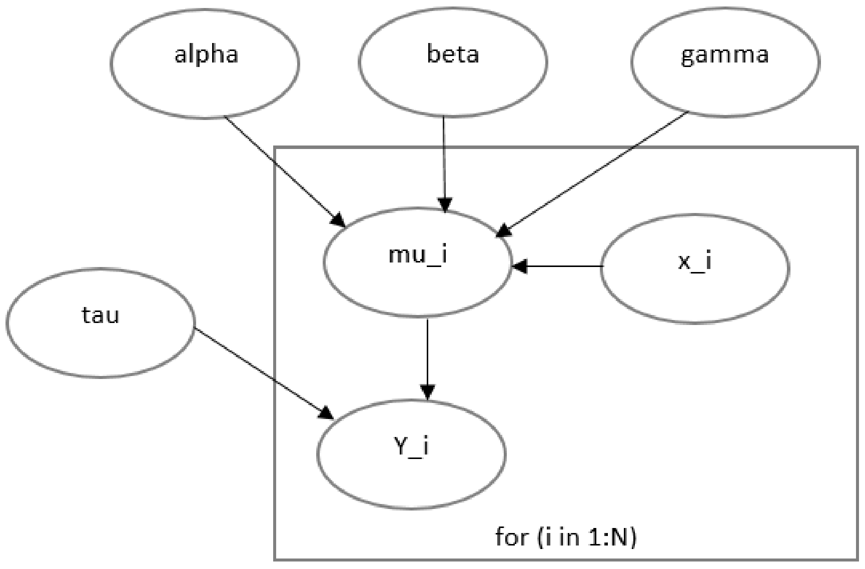
$$Y_i \sim Normal(\mu_i, \tau) \qquad with \quad i = 1, ..., 27$$

where $\mu_i = \mathbb{E}[Y_i|X_i = x_i] = f(x_i) = \alpha - \beta\gamma^{x_i}$

The model parameters are $\alpha \in (1, \infty)$, $\beta \in (1, \infty)$, $\gamma \in (0, 1)$ and $\tau^2 \in (0, \infty)$ and we consider the following prior distribution:

$$\alpha \sim N(0, \sigma_\alpha^2)$$
$$\beta \sim N(0, \sigma_\beta^2)$$
$$\gamma \sim Unif(0, 1)$$
$$\tau^2 \sim IG(a, b)$$

The graph is given below:



After that we load the data set, we can see graphically a simple relation between the two variables

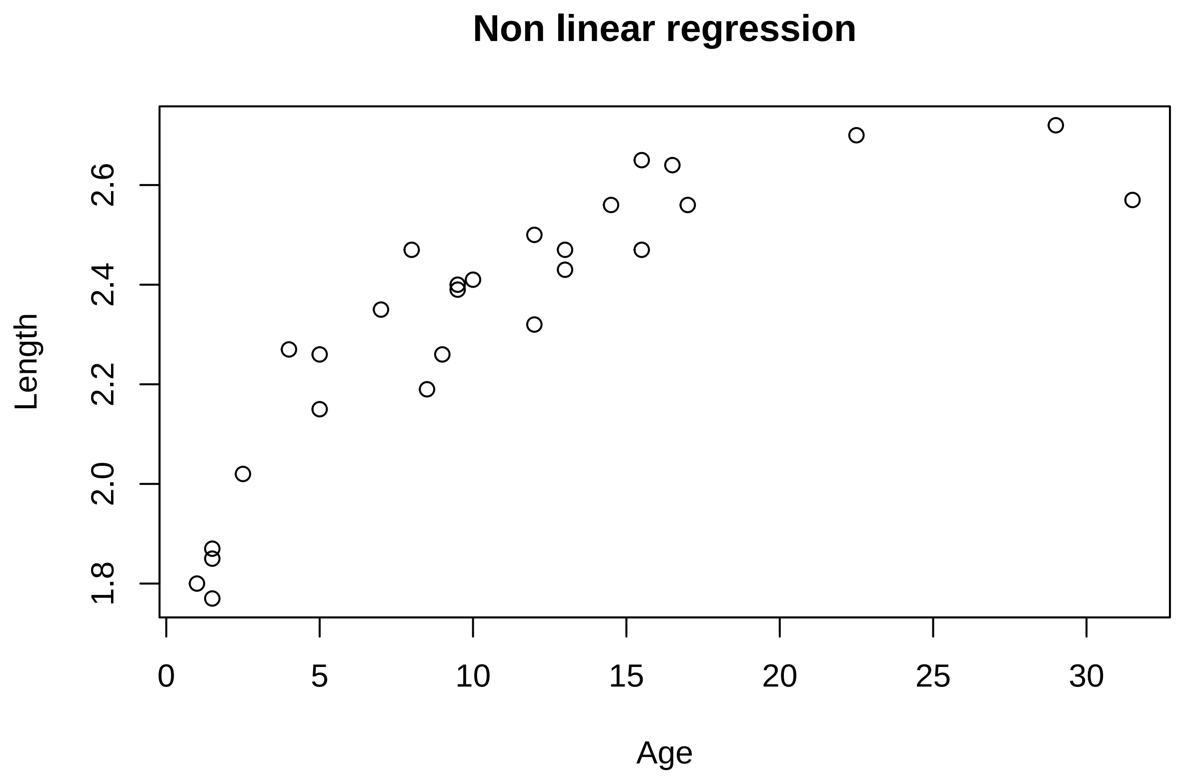
```r
df=read.csv("dugongs.txt",sep="\t")

# let's see the variables into the dataframe
names(df)
```

```
## [1] "Age"    "Length"
```

```
# number of dugongs present in the dataframe
n = dim(df)[1]
n
```

```
## [1] 27
```

```
# the dependent variable represents the length of the dugongs, while the dependent
# variable represents the age of the dugongs
x = df$Age
y = df$Length

## x = age
# x = c( 1.0,  1.5,  1.5,  1.5, 2.5,   4.0,  5.0,  5.0,  7.0,
#        8.0,  8.5,  9.0,  9.5, 9.5,  10.0, 12.0, 12.0, 13.0,
#        13.0, 14.5, 15.5, 15.5, 16.5, 17.0, 22.5, 29.0, 31.5)

## y = lenght
# y = c(1.80, 1.85, 1.87, 1.77, 2.02, 2.27, 2.15, 2.26, 2.47,
#       2.19, 2.26, 2.40, 2.39, 2.41, 2.50, 2.32, 2.32, 2.43,
#       2.47, 2.56, 2.65, 2.47, 2.64, 2.56, 2.70, 2.72, 2.57)


plot(x,y, xlab = "Age", ylab = "Length", main = "Non linear regression")
```



1b answer)

The likelihood function can be derived in this way:

$$L(\alpha,\beta,\gamma,\tau|y,x) = \prod_{i=1}^{n} \frac{1}{(2\pi\tau^2)^{\frac{1}{2}}} \cdot exp\left\{-\frac{(y_i-\mu_i)^2}{2\tau^2}\right\} \mathbb{I}_{(1,\infty)}(\alpha)\mathbb{I}_{(1,\infty)}(\beta)\mathbb{I}_{(0,1)}(\gamma)\mathbb{I}_{(0,\infty)}(\tau^2)$$

$$L(\alpha,\beta,\gamma,\tau|y,x) = \frac{1}{(2\pi\tau^2)^{\frac{n}{2}}} \cdot exp\left\{-\frac{1}{2\tau^2}\sum_{i=1}^{n}(y_i-\mu_i)^2\right\} \mathbb{I}_{(1,\infty)}(\alpha)\mathbb{I}_{(1,\infty)}(\beta)\mathbb{I}_{(0,1)}(\gamma)\mathbb{I}_{(0,\infty)}(\tau^2)$$

$$= \frac{1}{(2\pi\tau^2)^{\frac{n}{2}}} \cdot exp\left\{-\frac{1}{2\tau^2}\sum_{i=1}^{n}(y_i-\alpha+\beta\gamma^{x_i})^2\right\} \mathbb{I}_{(1,\infty)}(\alpha)\mathbb{I}_{(1,\infty)}(\beta)\mathbb{I}_{(0,1)}(\gamma)\mathbb{I}_{(0,\infty)}(\tau^2)$$

1c answer)

Before writing the expression of the joint prior distribution of the parameters, we need to compute separately the prior of each parameter:

$$\pi(\alpha) = \frac{1}{\sqrt{2\pi\sigma_\alpha^2}} exp\left\{-\frac{\alpha^2}{2\sigma_\alpha^2}\right\}\mathbb{I}_{(1,\infty)}(\alpha) \propto exp\left\{-\frac{\alpha^2}{2\sigma_\alpha^2}\right\}\mathbb{I}_{(1,\infty)}(\alpha)$$

$$\pi(\beta) = \frac{1}{\sqrt{2\pi\sigma_\beta^2}} exp\left\{-\frac{\beta^2}{2\sigma_\beta^2}\right\}\mathbb{I}_{(1,\infty)}(\beta) \propto exp\left\{-\frac{\beta^2}{2\sigma_\beta^2}\right\}\mathbb{I}_{(1,\infty)}(\beta)$$

$$\pi(\gamma) = \mathbb{I}_{(0,1)}(\gamma)$$

$$\pi(\tau^2) = \frac{b^a}{\Gamma(a)}\tau^{2^{(-a-1)}} exp\left\{-\frac{b}{\tau^2}\right\}\mathbb{I}_{(0,\infty)}(\tau^2) \propto \tau^{2^{(-a-1)}} exp\left\{-\frac{b}{\tau^2}\right\}\mathbb{I}_{(0,\infty)}(\tau^2)$$

The joint prior can be obtained as:

$$\pi(\theta) = \pi(\alpha)\pi(\beta)\pi(\gamma)\pi(\tau^2) = \frac{1}{\sqrt{2\pi\sigma_\alpha^2}} exp\left\{-\frac{\alpha^2}{2\sigma_\alpha^2}\right\}\cdot\frac{1}{\sqrt{2\pi\sigma_\beta^2}} exp\left\{-\frac{\beta^2}{2\sigma_\beta^2}\right\}\cdot\mathbb{I}_{(0,1)}(\gamma)\cdot\frac{b^a}{\Gamma(a)}\tau^{2^{(-a-1)}} exp\left\{-\frac{b}{\tau^2}\right\}$$

$$\propto exp\left\{-\frac{\alpha^2}{2\sigma_\alpha^2}\right\}\cdot exp\left\{-\frac{\beta^2}{2\sigma_\beta^2}\right\}\cdot\mathbb{I}_{(0,1)}(\gamma)\cdot\tau^{2^{(-a-1)}} exp\left\{-\frac{b}{\tau^2}\right\}$$

About the values of σ_α^2, σ_β^2, a and b, I chose: $\sigma_\alpha^2 = 10000$, $\sigma_\beta^2 = 10000$, $a = 0.001$ and $b = 0.001$

$$\alpha \sim N(0,\sigma_\alpha^2) \Longrightarrow \alpha \sim N(0,10000)$$
$$\beta \sim N(0,\sigma_\beta^2) \Longrightarrow \beta \sim N(0,10000)$$
$$\gamma \sim Unif(0,1)$$
$$\tau^2 \sim IG(a,b) \Longrightarrow \tau^2 \sim IG(0.001,0.001)$$

1d answer)

We compute numerically the maximum likelihood estimate for the vector of parameters of interest $(\alpha,\beta,\gamma,\tau)$ and compare it with the Maximum-a-Posterori estimate in the following R code:

```r
# load tha packages
library(MCMCpack)
```

```
## Warning: package 'MCMCpack' was built under R version 3.3.3

## Loading required package: coda

## Warning: package 'coda' was built under R version 3.3.3

## Loading required package: MASS

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)

## ## Copyright (C) 2003-2017 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park

## ##
## ## Support provided by the U.S. National Science Foundation

## ## (Grants SES-0350646 and SES-0350613)
## ##
```

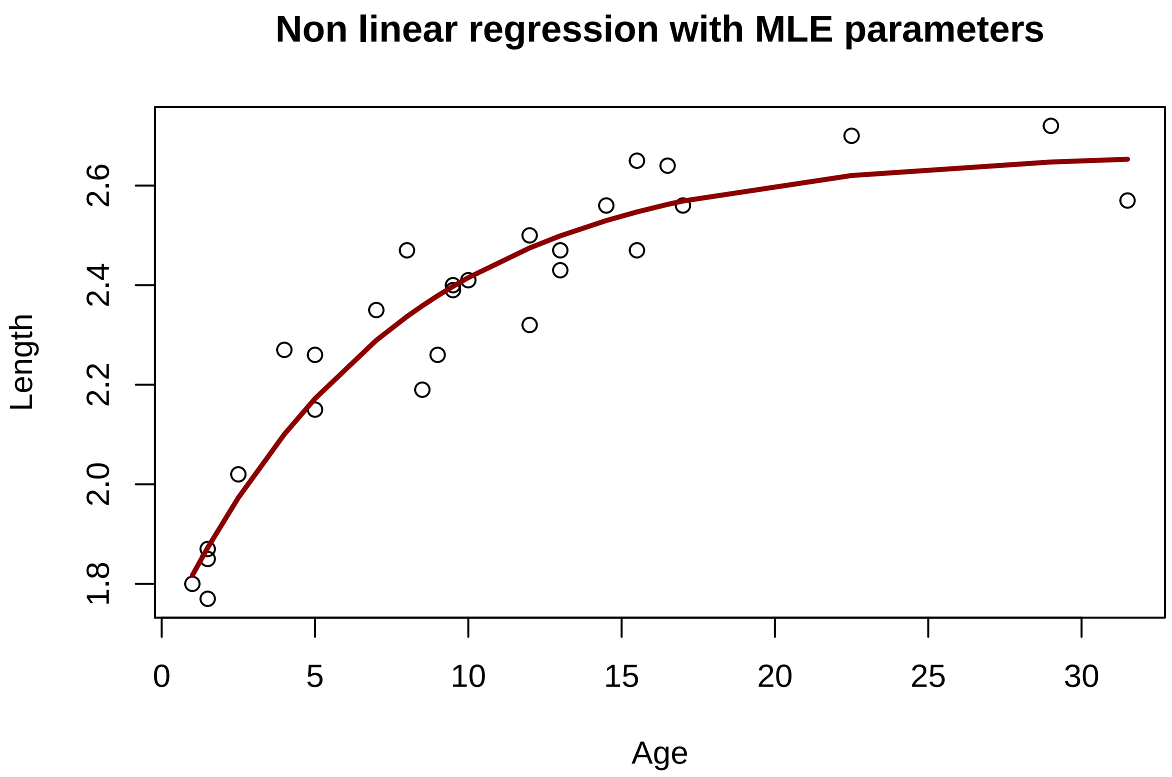
```r
# initial parameters of alpha, beta, gamma and tau (let's chose them randomly)
init_parameters = c(1.5, 1.5, 0.5, 1)

# compute the log-likelihood function
log_L = function(pars) {
  alpha = pars[1]
  beta = pars[2]
  gamma = pars[3]
  tau = pars[4]
  mu = alpha - beta*(gamma^(x))
  #I = *(alpha>1)*(beta>1)*(gamma>0 & gamma<1)*(tau>0)
  out = sum(dnorm(y, mu, tau, log=T)) #*I
  return(out)
}

# find maximum likelihood estimates for the vector of parameters of interest
opt_parameters = optim(par = init_parameters, log_L, control = list(fnscale=-1))$par
```

```
## Warning in dnorm(y, mu, tau, log = T): Si è prodotto un NaN
```

```r
# the maximum likelihood estimates for the vector of
# parameters of interest are respectively:
opt_parameters
```

```
## [1] 2.66662195 0.97255678 0.87349199 0.08316385
```

```r
# plot the data and let's see graphically the result of the non linear regression if we
# chose the parameters alpha, beta, gamma and tau with the command "optim"
plot(x,y, xlab = "Age", ylab = "Length",
     main = "Non linear regression with MLE parameters")
lines(x, opt_parameters[1]-opt_parameters[2]*opt_parameters[3]^x, type = "l",
      col = "dark red", lwd = 2.5)
```



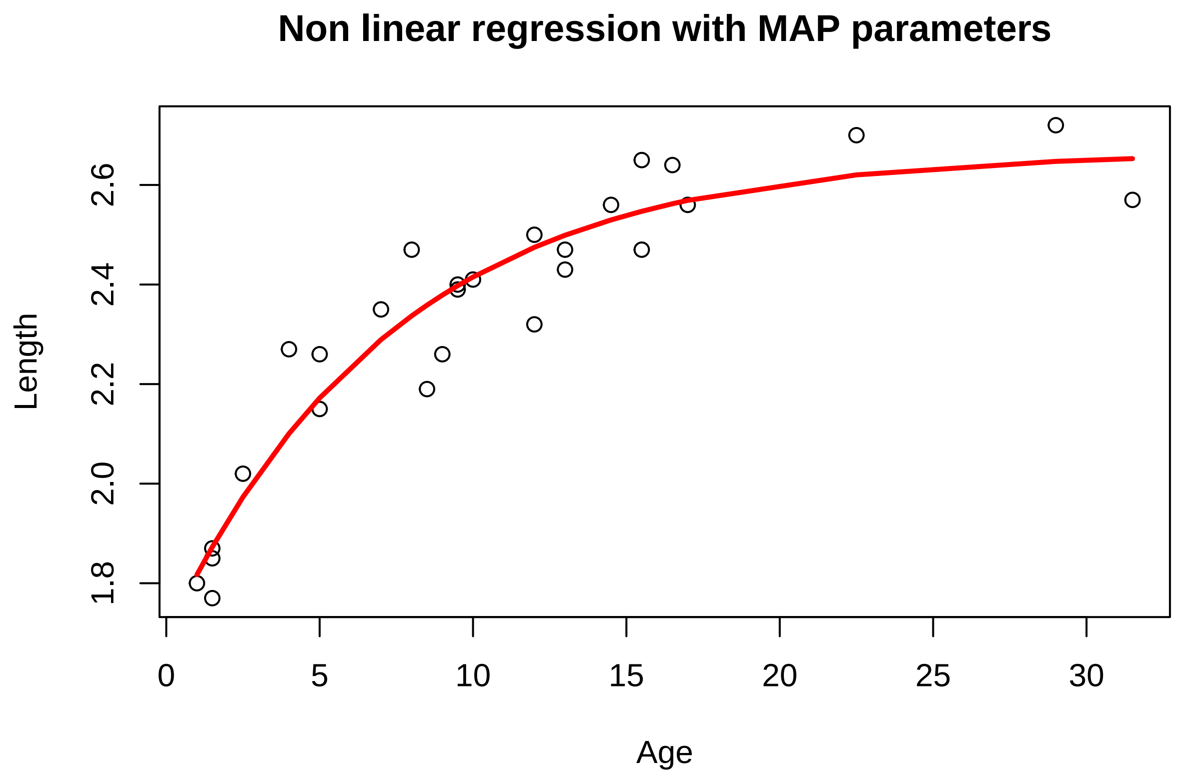
```r
# compute, for every parameter, the corresponding prior
prior_alpha = function(alpha) return(dnorm(alpha, 0, 10000))
prior_beta = function(beta) return(dnorm(beta, 0, 10000))
prior_gamma = function(gamma) return (dunif(gamma))
prior_tau = function(tau,a,b) return(dinvgamma(tau, 0.001, 0.001))

# the joint prior distribution can be computed as product of the prior of alpha,
# beta, gamma and tau
joint_prior = function(pars){
  alpha = pars[1]
  beta = pars[2]
  gamma = pars[3]
  tau = pars[4]
  out = prior_alpha(alpha)*prior_beta(beta)*prior_gamma(gamma)*prior_tau(tau,a,b)
  return(out)
}

# compute the log-posterior
log_posterior = function(pars){
  alpha = pars[1]
  beta = pars[2]
  gamma = pars[3]
  tau = pars[4]
  out = log_L(pars) + log(prior_alpha(alpha)) + log(prior_beta(beta)) +
    log(prior_gamma(gamma)) + log(prior_tau(tau))
  return(out)
}

# find Maximum-a-Posterori estimate for the vector of parameters of interest
opt_parameters_posterior = optim(par = init_parameters, log_posterior,
                                 control = list(fnscale=-1))$par
```

```
## Warning in dnorm(y, mu, tau, log = T): Si è prodotto un NaN

## Warning in log(x): Si è prodotto un NaN

## Warning in dnorm(y, mu, tau, log = T): Si è prodotto un NaN

## Warning in log(x): Si è prodotto un NaN

## Warning in dnorm(y, mu, tau, log = T): Si è prodotto un NaN

## Warning in log(x): Si è prodotto un NaN
```

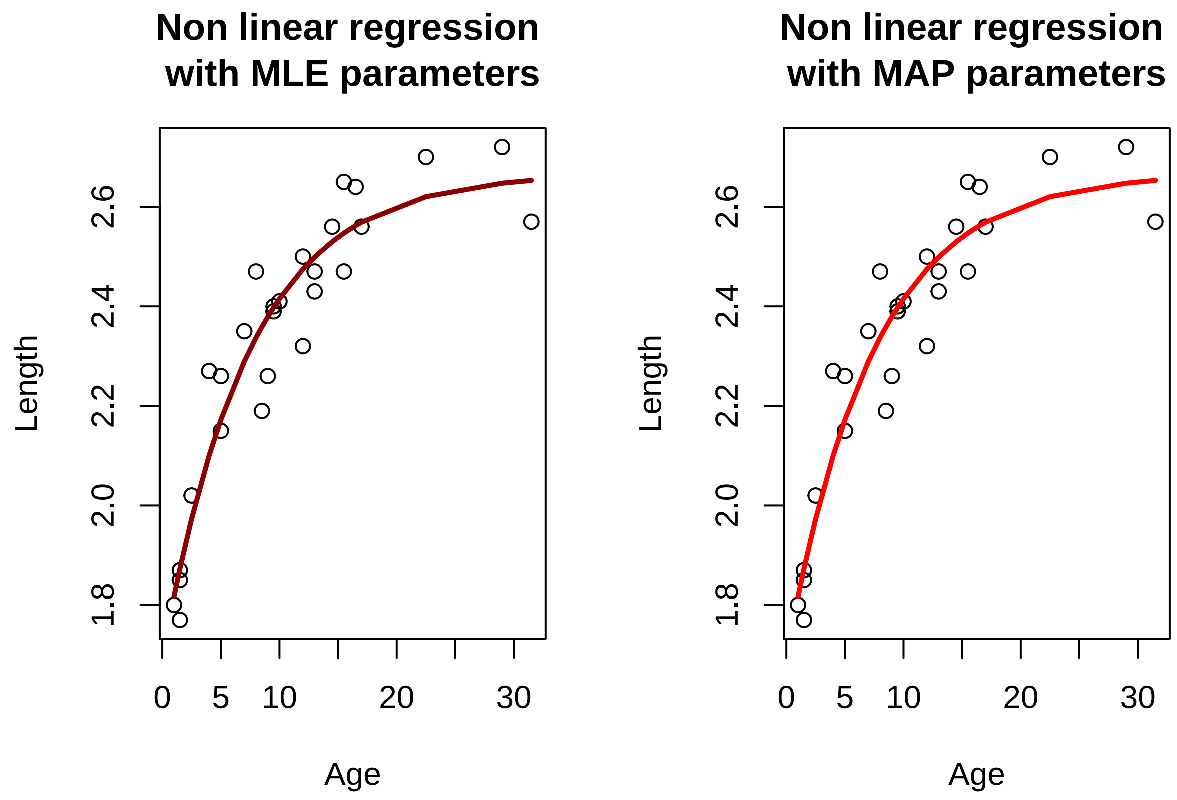
```
# the Maximum-a-Posterori estimate for the vector of parameters
# of interest are respectively:
opt_parameters_posterior
```

```
## [1] 2.66664140 0.97250181 0.87350365 0.08168616
```

```
# plot the data and let's see graphically the result of the non linear regression if we
# chose the parameters alpha, beta, gamma and tau with the command "optim"
plot(x,y, xlab = "Age", ylab = "Length",
     main = "Non linear regression with MAP parameters")
lines(x, opt_parameters_posterior[1]-opt_parameters_posterior[2]
      *opt_parameters_posterior[3]^x, type = "l", col = "red", lwd = 2.5)
```



```r
# let's compare the last plot with the initial plot
par(mfrow = c(1,2))
plot(x,y, xlab = "Age", ylab = "Length",
     main = "Non linear regression \nwith MLE parameters")
lines(x, opt_parameters[1]-opt_parameters[2]*opt_parameters[3]^x, type = "l",
       col = "dark red", lwd = 2.5)
plot(x,y, xlab = "Age", ylab = "Length",
     main = "Non linear regression \nwith MAP parameters")
lines(x, opt_parameters_posterior[1]-opt_parameters_posterior[2]
       *opt_parameters_posterior[3]^x, type = "l", col = "red", lwd = 2.5)
```



```r
par(mfrow = c(1,1))

# comparison between estimated MLE parameters and estimated MAP parameters
comparison = rbind("estimated MLE parameters:" = opt_parameters,
                   "estimated MAP parameters:" = opt_parameters_posterior)
colnames(comparison) = c("alpha", "beta", "gamma","tau")
comparison
```

```
##                              alpha      beta     gamma        tau
## estimated MLE parameters: 2.666622 0.9725568 0.8734920 0.08316385
## estimated MAP parameters: 2.666641 0.9725018 0.8735037 0.08168616
```

```r
# we can say that the maximum likelihood estimates andthe Maximum-a-Posterori estimate
# for the vector of parameters of interest are very similar
```

Part 2

Acceptance/Rejection method - part I

2a answer)

If we consider the A-R algorithm in the most general form, our target is known up to a moltiplicative costant

$$\tilde{f}_X(x) = c \cdot f_X(x),$$

then $f_X(x) = \frac{\tilde{f}_X(x)}{c}$.

We want a probability density q and a suitable costant k such that

$$f_X(x) \leq kq(x) \quad \forall x \in \mathcal{X}$$

About the Acceptance/Rejection method, to simulate we need:

1) target distribution written $f_X(x)$ from which we can compute it
2) a candidate distribution q that is easy to simulate, so we can simulate $Y_1, ..., Y_n$ from candidate distribution $Y \sim q$

The candidate must be able to dominate target distribution:

$$\exists k > 0 \quad s.t. \quad f_X(x) \leq kq(x) \quad \forall x \in \mathcal{Y}$$

For each $Y \sim f_U$ we use an auxiliary experiment

$$Y = \begin{cases} Y^A & \text{if } U \leq \frac{f_X(x)}{kq(y)} \\ Y^R & \text{otherwise} \end{cases}$$

About the acceptance probability, we have that:

1) $Y \sim q$ with support $\mathcal{Y}$ (suppose $\mathcal{Y} = \mathbb{R}$)
2) $U \sim Unif[0, 1]$
3) Y, U independent
4) If $U \leq \frac{\tilde{f}_X(y)}{ckq_Y(y)}$, we accept.

If we consider the Acceptance-Rejection algorithm in the most general form and denote with $\theta = Y^A$ the random variable obtained with the algorithm, we can compute the unconditional acceptance probability and the conditional acceptance probability

The unconditional acceptance probability is given by:

$$Pr(Y = Y^A | Y = y) = Pr\left(U \leq \frac{\tilde{f}_\Theta(Y)}{ckq_Y(Y)} | Y = y\right) = Pr\left(U \leq \frac{\tilde{f}_\Theta(y)}{ckq_Y(y)}\right) = \frac{\tilde{f}_\Theta(y)}{ckq_Y(y)}$$

The conditional acceptance probability is given by:

$$Pr(Y = Y^A) = \int_{\mathcal{Y}} Pr(Y = Y^A | Y = y) q_Y(y) dy = \int_{\mathcal{Y}} \frac{\tilde{f}_\Theta(y)}{ckq_Y(y)} q_Y(y) dy = \int_{\mathcal{Y}} \frac{\tilde{f}_\Theta(y)}{ck} dy = \frac{1}{ck}$$

2b answer)

We can prove that θ has the desired target distribution

$$Pr(Y^A < \theta) = Pr(Y < \theta | Y = Y^A) = \frac{\int_{-\infty}^{\theta} Pr(Y = Y^A | Y = y) q_Y(y) dy}{\int_{-\infty}^{+\infty} Pr(Y = Y^A | Y = y) q_Y(y) dy}$$

$$= \frac{\int_{-\infty}^{\theta} \frac{\tilde{f}_\Theta(y)}{ckq_Y(y)} q_Y(y) dy}{\int_{-\infty}^{+\infty} \frac{\tilde{f}_\Theta(y)}{ckq_Y(y)} q_Y(y) dy} = \frac{\int_{-\infty}^{\theta} \frac{\tilde{f}_\Theta(y)}{ck} dy}{\int_{-\infty}^{+\infty} \frac{\tilde{f}_\Theta(y)}{ck} dy} = \frac{\frac{1}{ck} \int_{-\infty}^{\theta} \tilde{f}_\Theta(y) dy}{\frac{1}{ck} \int_{-\infty}^{+\infty} \tilde{f}_\Theta(y) dy} = \int_{-\infty}^{\theta} \tilde{f}_\Theta(y) dy = F_\Theta(\theta)$$

2c answer)

We can show how in Bayesian inference we could use simulations from the prior (auxiliary density) to get a random draw from the posterior (target distribution) without knowing the proportionality constant

Let's consider the case of Bayesian inference where our target distribution is:

$$\tilde{f}_X(x) = \pi(\theta | x) = c \cdot \pi(\theta) L_X(\theta) \Longrightarrow f_X(\theta) = \frac{\tilde{f}_X(\theta)}{c} = \frac{c \cdot \pi(\theta) L_X(\theta)}{c} = \pi(\theta) L_X(\theta)$$

and the candidate $\theta \sim q(\theta) = \pi(\theta)$

The acceptance rule is:

$$U \leq \frac{f_X(\theta)}{kq(\theta)} = \frac{\pi(\theta) L_X(\theta)}{k\pi(\theta)} = \frac{L_X(\theta)}{k}$$

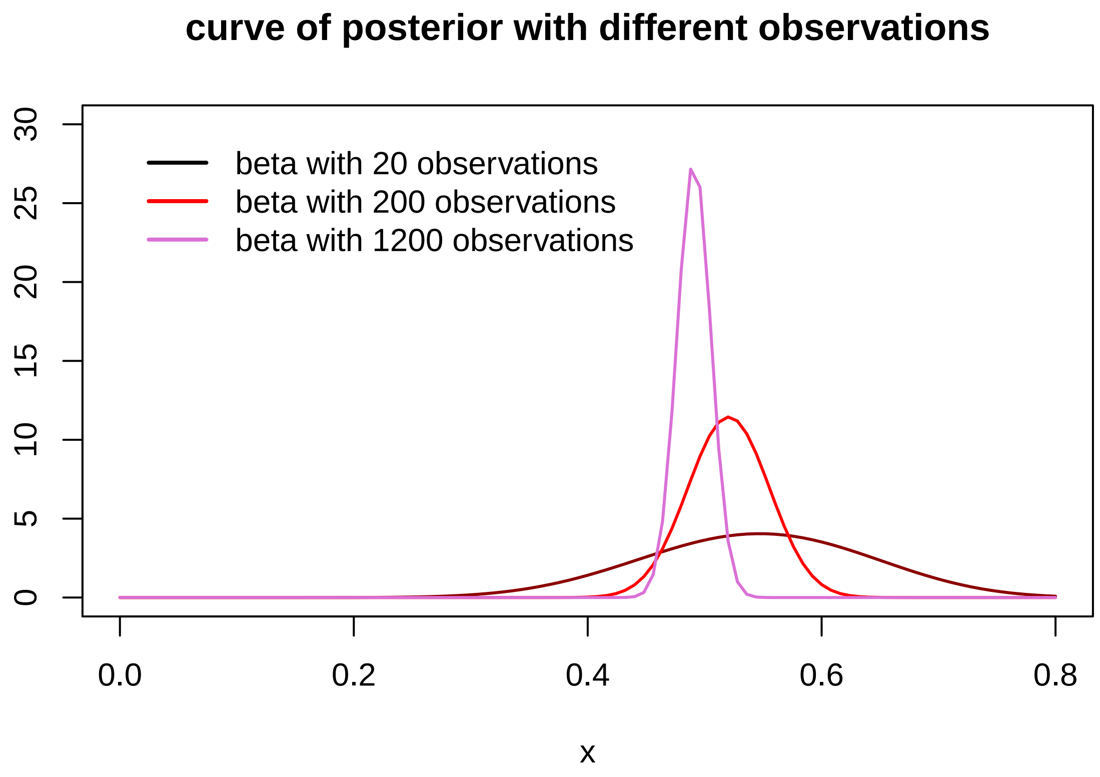
The optimal choice of k is obtained by maximizing the likelihood, then values are accepted if $U \leq \frac{L(\theta)}{\max_{\theta \in \Theta} L(\theta)}$

2d answer)

It's possible to find problem with A-R method, for example, when n (number of observations) is large: the likelihood is highly peaked (and so also the posterior, "precise measurement") relative to the prior, therefore the acceptance probability could be very small. In general, when the likelihood is concentrated around $\hat{\theta}_{MLE}$, it's difficult to have a good result, also in terms of MC approximation (much variability).

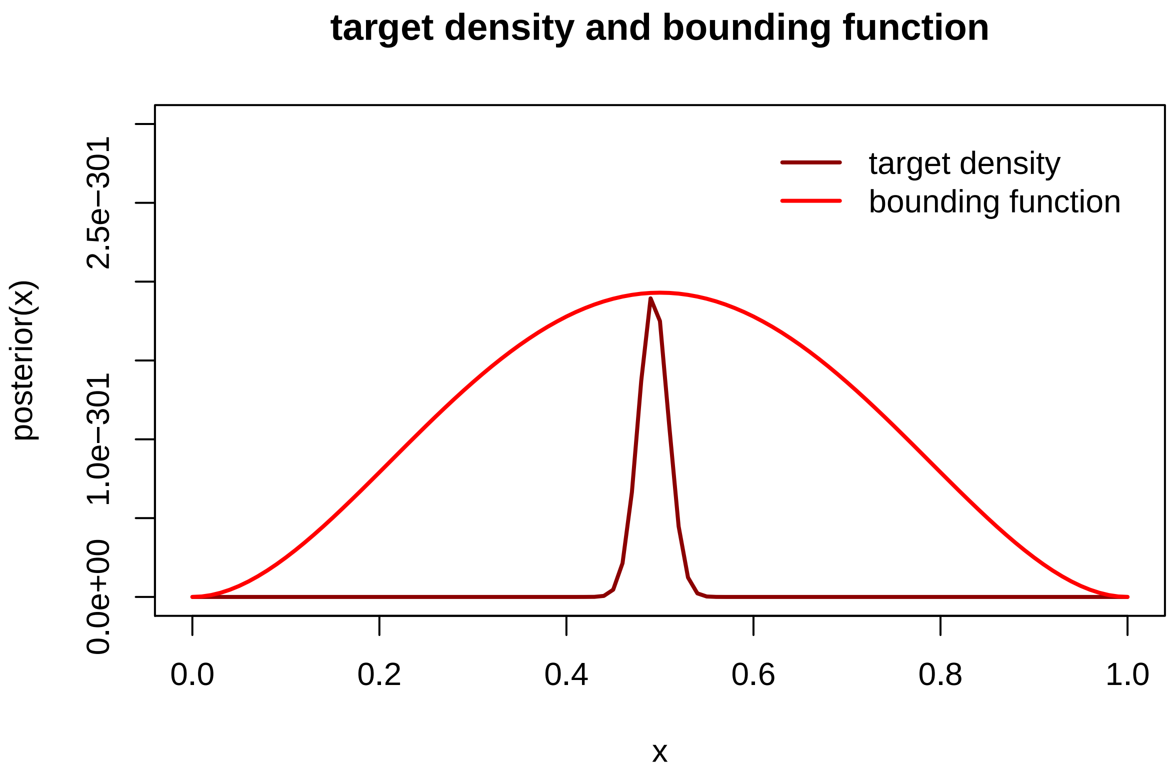
```r
# comparison between Beta posteriors with different number of obervations

alpha = 3
beta = 3

n_sim_small=20
# generation of sample from Beta with 20 observation
sample_small = rbeta(n_sim_small,alpha,beta)
# curve of posterior with 20 observations
a_20 = alpha + sum(sample_small)
b_20 = beta + length(sample_small)-sum(sample_small)
curve(dbeta(x, a_20, b_20), lwd=1.5, ylim=c(0,30),
      xlim=c(0,0.8), main="curve of posterior with different observations",
      ylab="", col="dark red")

n_sim_medium = 200
# generation of sample from Beta with 200 observation
sample_medium = rbeta(n_sim_medium,alpha,beta)
# curve of posterior with high observations
```

```r
a_200 = alpha + sum(sample_medium)
b_200 = beta + length(sample_medium)-sum(sample_medium)
curve(dbeta(x, a_200, b_200), add=T, col="red", lwd=1.5)

n_sim_big = 1200
# generation of sample from Beta with 1200 observation
sample_big = rbeta(n_sim_big,alpha,beta)
# curve of posterior with high observations
a_1200 = alpha + sum(sample_big)
b_1200 = beta + length(sample_big)-sum(sample_big)
curve(dbeta(x, a_1200, b_1200), add=T, col="orchid", lwd=1.5)

legend(x=0,y=30, c("beta with 20 observations", "beta with 200 observations",
                   "beta with 1200 observations"), lty=c(1,1), lwd=c(2,2),
                    col=c("black", "red","orchid"), bty="n")
```

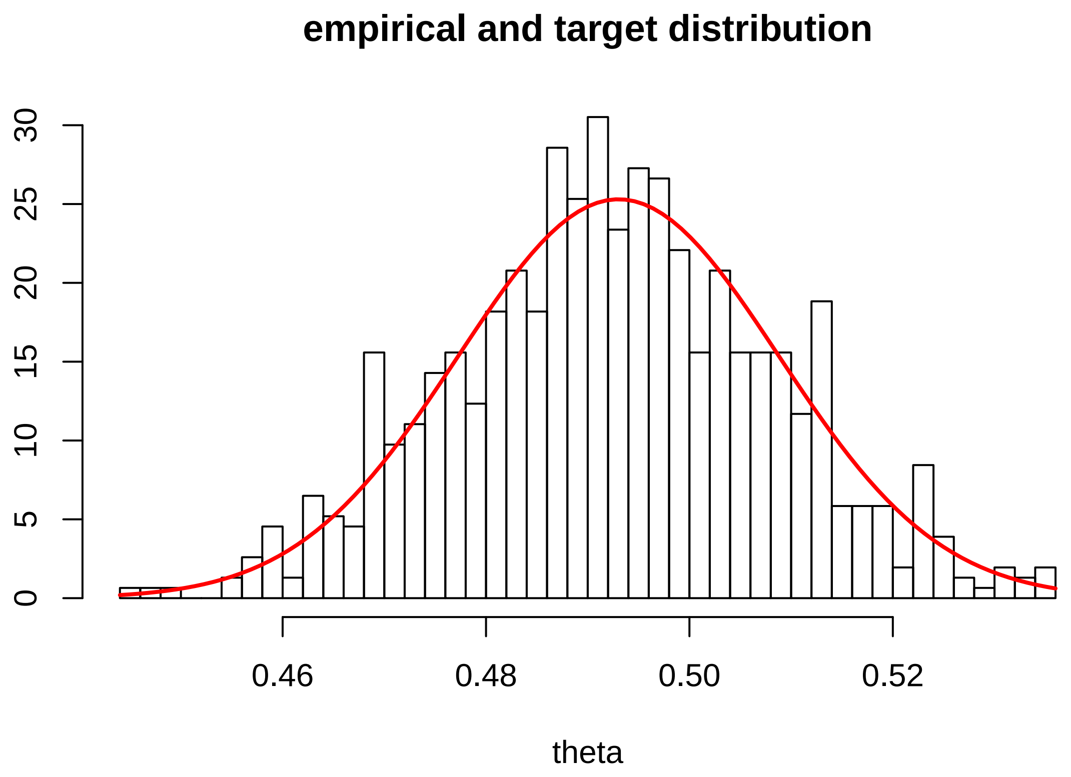


2e answer)

```r
set.seed(123)

# compute the likelihood function of a bernoulli
L = function(x, theta){
  n = length(x)
  out = theta^(sum(x))*(1-theta)^(n-sum(x))
  return(out)
}

# simulate from a bernoulli
theta= 0.5
sample = rbinom(1000, 1, theta)
n = length(sample)

# find the maximum value of the likehood function
k = L(sample, mean(sample))

# chose the parameters of the prior and compute it
alpha = 3
beta = 3
prior = function(x) return(dbeta(x, alpha, beta))

posterior = function(x) return(L(sample, x)*prior(x))

bounding_function = function(x) return(k*prior(x))

curve(posterior(x), col="dark red", lwd=2, ylim=c(0,3e-301),
      main = "target density and bounding function")
curve(bounding_function(x), lwd=2, add=T, col="red")
legend(x=0.60, y=3e-301, c("target density","bounding function"),
       lty=c(1,1), lwd=c(2,2), col=c("dark red", "red"), bty="n")
```



```r
# verify that theta has the target function as distribution

# generate Y_i where i=1,...10000 from q(y) ~ Beta(3,3)
nsim = 10000
y = rbeta(nsim,alpha, beta)
U = runif(nsim,0,1)

# acceptance condition
theta_final = y[U <= L(sample, y)/k]
num_accepted=length(theta_final)
accept_probability=num_accepted/nsim

hist(theta_final, prob=T, main="empirical and target distribution",
     xlab="theta", ylim=c(0,30), ylab="", breaks = 50)
a = alpha + sum(sample)
b = beta + n - sum(sample)
curve(dbeta(x, a, b), add=T, col="red", lwd=2)
```



Part 3

Acceptance/Rejection method - part II

3 answer)

We can simulate from a standard Normal distribution using pseudo-random deviates from a standard Cauchy and the A-R algorithm.

About the Acceptance/Rejection method, as seen before, to simulate we need:

1) target distribution written $f_X(x)$ from which we can compute it and in this case $f_X(x) \sim N(0,1)$
2) a candidate distribution q that is easy to simulate (in this case $q \sim Cauchy(0,1)$), so we can simulate $Y_1, ..., Y_n$ from candidate distribution $Y \sim q$

Our purpose corresponds to write the expression the corresponding acceptance probability of and evaluate it numerically by MC approximation.

About the choice of k, we know that the acceptance probability is high if the k value is smaller than possible, but at the same time k must satisfy the dominated condition, so the minimum value that k can assume is the maximum of function $\frac{f_X(x)}{q}$

The acceptance probability can be computed as $\frac{1}{k}$ or by Monte Carlo approximation that is egual to the ratio between the number of values accepted and the number of values genered:

$$\mathbb{E}(Y^A) \simeq \frac{1}{n}\sum_{i=1}^{n} Y^A$$

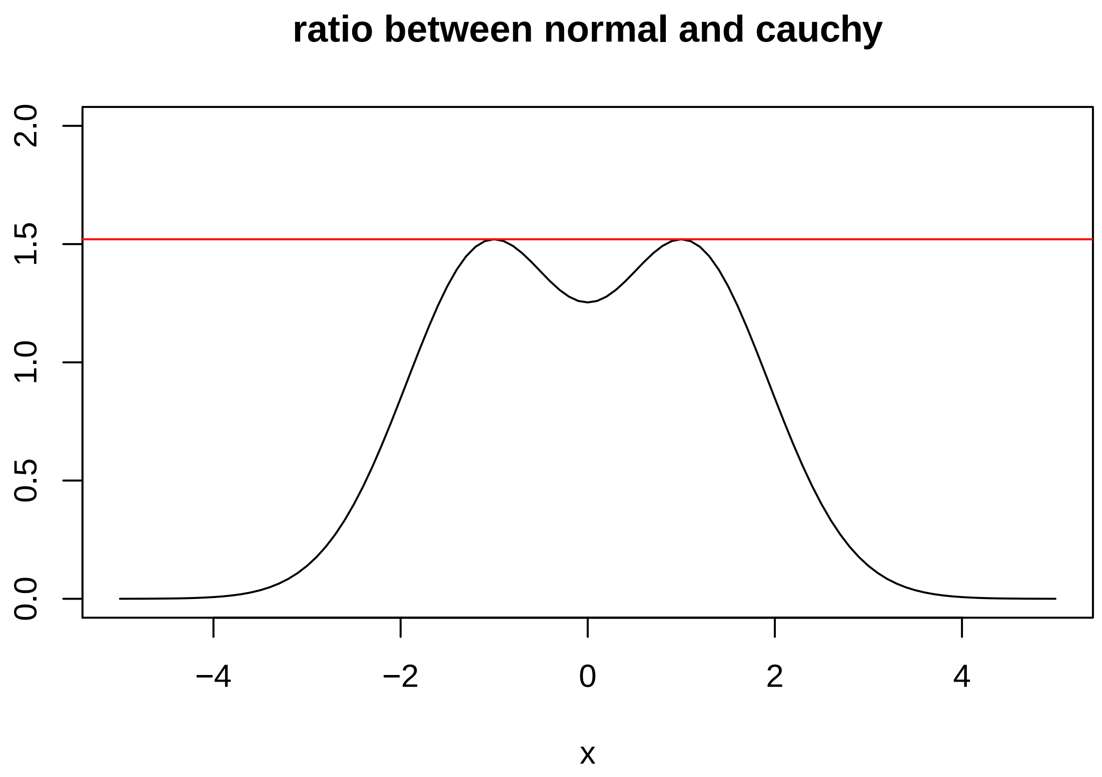
```r
normal = function(x) return (dnorm(x,0,1))
cauchy = function(x) return(dcauchy(x,0,1))

d = function(x) return(normal(x)/cauchy(x))
curve(d(x),xlim=c(-5,5), ylim=c(0,2), ylab="", main="ratio between normal and cauchy")
opt = optimize(d, interval=c(-5,5), maximum=TRUE)
# maximum of the ratio between a normal and cauchy
maximum=opt$maximum
maximum
```

```
## [1] -0.9999966
```

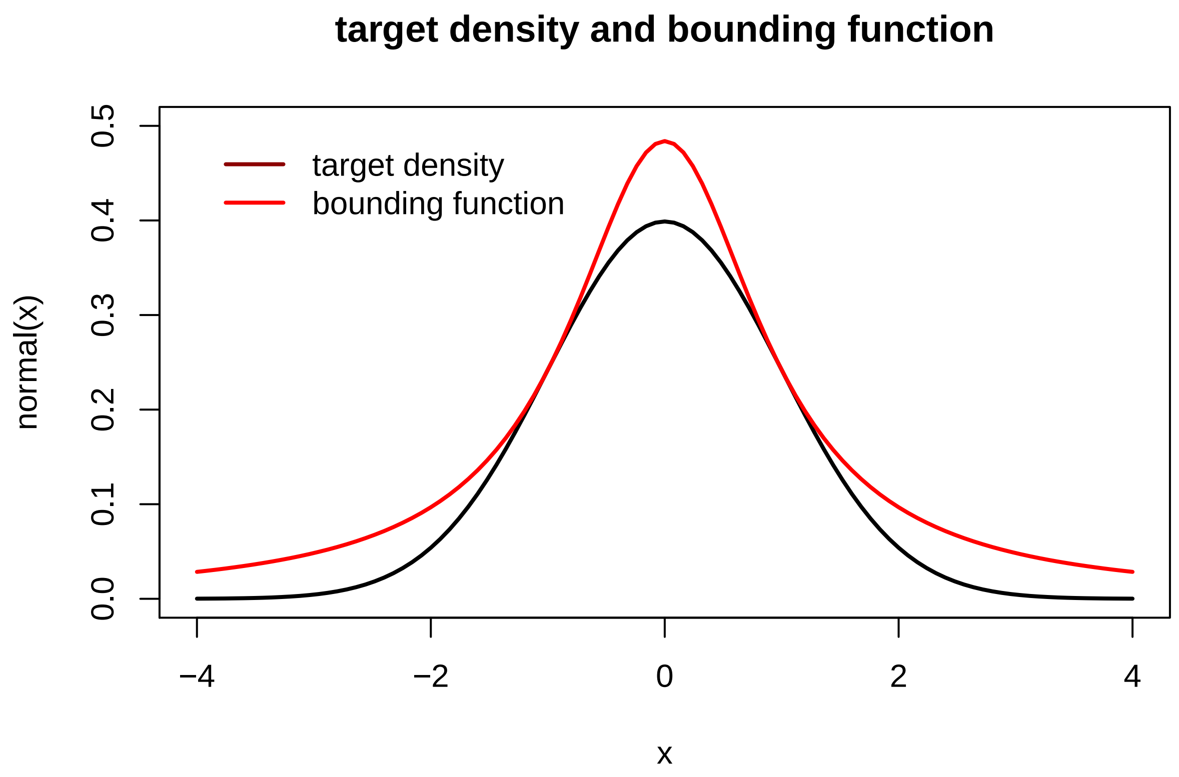
```r
# value of the maximum of the ratio between a normal and cauchy
k2=opt$objective
k2
```

```
## [1] 1.520347
```

```r
#curve(d(x), xlim=c(-5,5), ylim=c(0,2), ylab="ratio between normal and cauchy")
abline(h=k2, col='red')
```



```
bounding_function2 = function(x) return(k2*cauchy(x))
curve(normal(x), xlim = c(-4,4), ylim=c(0,0.5), lwd=2,
      main = "target density and bounding function")
curve(bounding_function2, add=T, col="red", lwd=2)
legend(x=-4, y=0.5, c("target density","bounding function"),
       lwd=c(2,2), col=c("dark red", "red"), bty = "n")
```



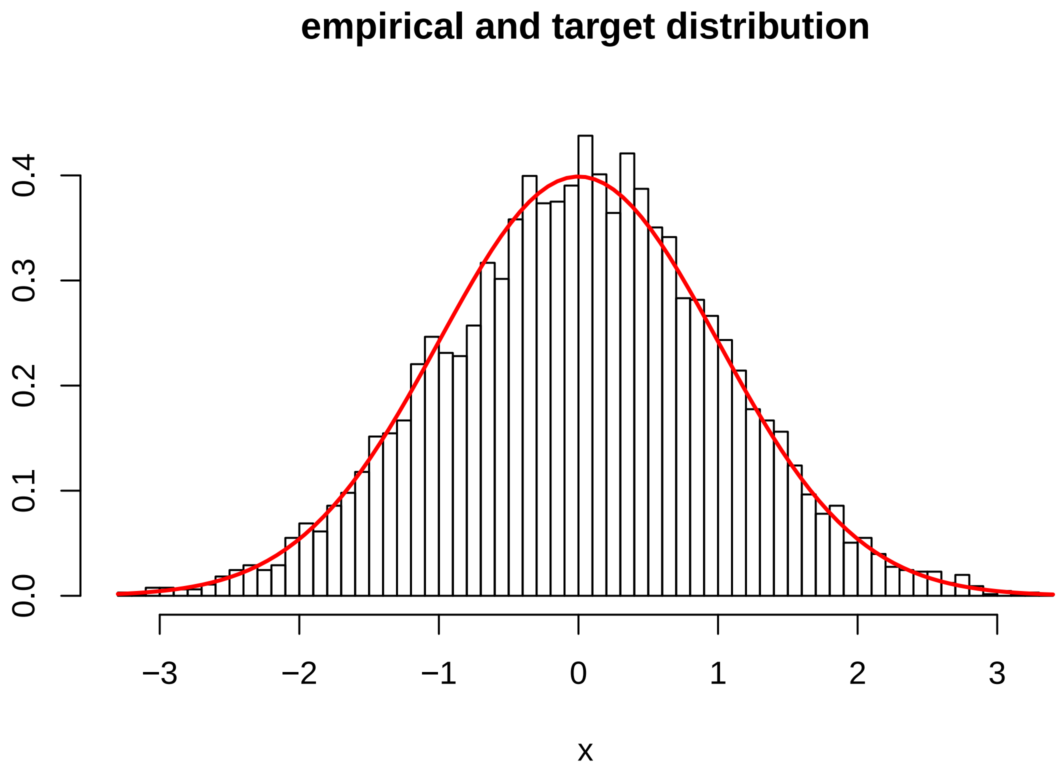
```
# generate Y_i where i=1,...10000 from q(y) ~ cauchy(0,1)
nsim=10000
y2 = rcauchy(nsim,0,1)
U2 = runif(nsim,0,1)

x = y2[U2<=dnorm(y2,0,1)/(k2*dcauchy(y2,0,1))]
num_accepted2 = length(x)

# empirical acceptance probability
accept_probability2 = num_accepted2/nsim
accept_probability2
```

```
## [1] 0.6533
```

```
#theoretical acceptance probability (1/k2)
th_accept_probability2 = 1/k2
th_accept_probability2
```

```
## [1] 0.6577446
```

```
# comparison between empirical distribution and theoretical distribution
hist(x, prob=T, ylim=c(0,0.45), breaks = 50, ylab="",
     main="empirical and target distribution")
curve(normal(x), add=T, col="red", lwd=2)
```


empirical and target distribution
0.0
0.1
0.2
0.3
0.4
−3
−2
−1
0
1
2
3
x