Homework-1

Valerio Antonini 1611556

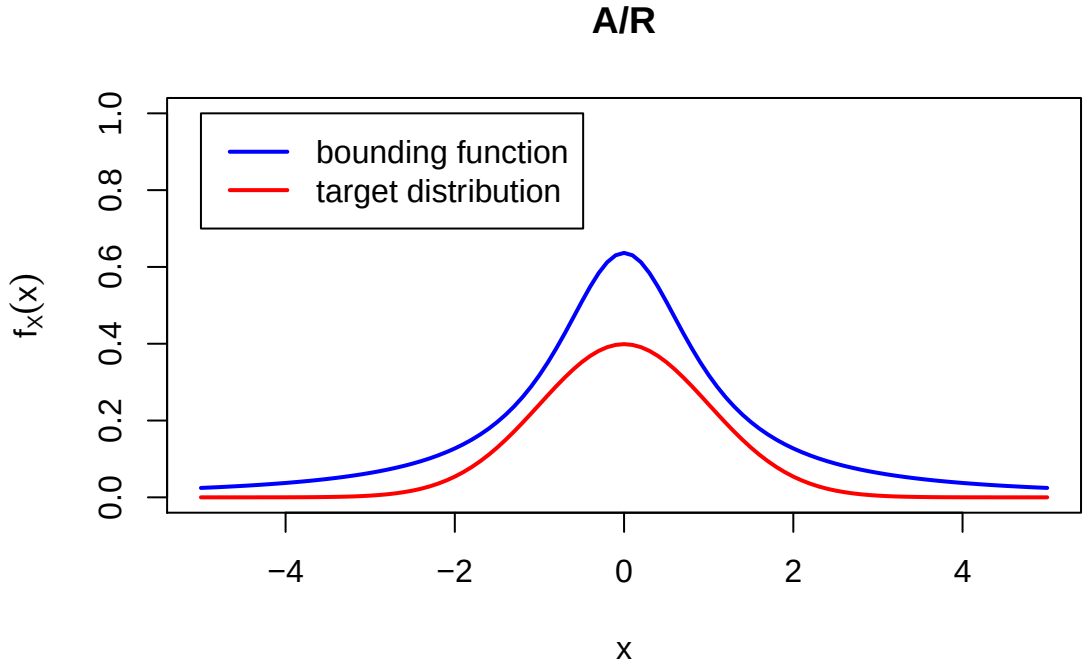
PART 1

a) show how it is possible to simulate from a standard Normal distribution using pseudo-random deviates from a standard Cauchy and the A-R algorithm.

Acceptance-Rejection method is a basic technique used to generate observations from a distribution. It is a type of exact simulation method. Rejection sampling is based on the observation that to sample a random variable in one dimension, and keep the samples in the region under the graph of its density function.

In order to simulate through A-R method we need: 1) A target distribution: $X = fx(x)$. In our case: $X = fx(x) \sim N(0,1)$. 2) A candidate distribution from which it's easy to simulate, $g(X) = Y$. In our case $g(X) = Y \sim Cauchy(0,1)$.

```
set.seed(123)

k = 2

# Let's set k=2 to see how dcauchy*k behaves


curve(dnorm(x), col = "red", lwd = 2, xlim = c(-5, 5), ylim = c(0,
    1), ylab = expression(f[X](x)))
curve(k * dcauchy(x, 0, 1), col = "blue", lwd = 2, add = TRUE,
    ylim = c(0, 0.7))
legend(x = -5, y = 1, legend = c("bounding function", "target distribution"),
    col = c("blue", "red"), lwd = 2)
title(main = "A/R")
```



The plot shows that $f_X(y) \leq k = kf_U(y)$, so we can implemtent the A-R method.

b) provide your R code for the implementation of the A-R;

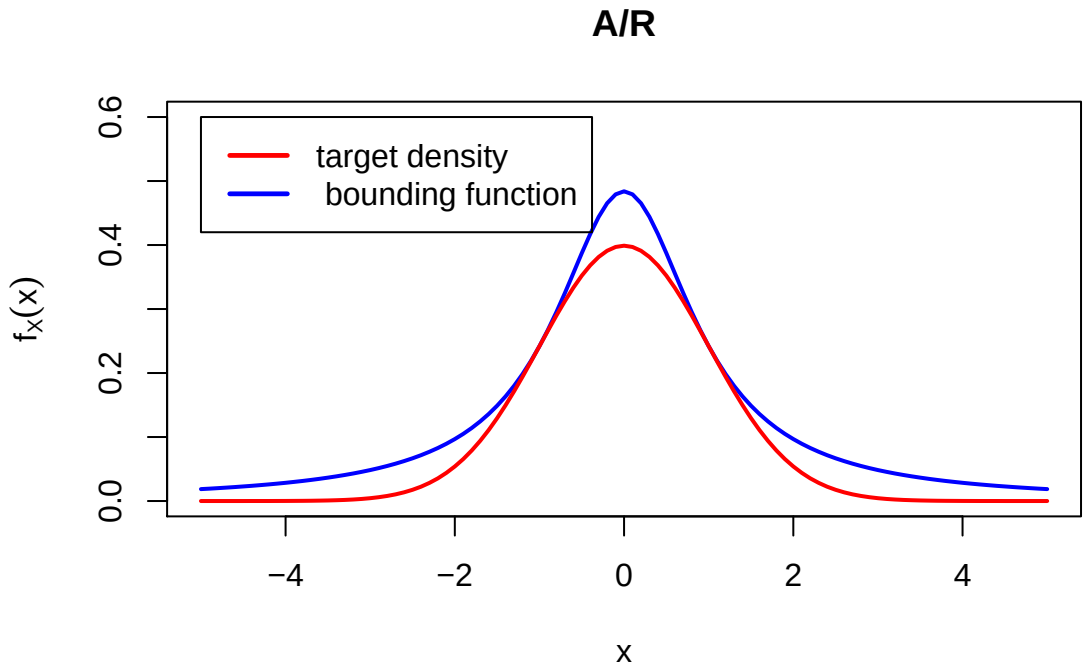
First step is find out a suitable k which allows the corresponding auxiliary density f_U generating candidates X_i to become f_U a density kf_U dominating the target f_X We can get a valid approximation for the optimal k from this formula:

$$\hat{k} = sup(f_X(x)/f_U(x))$$

```
x_grid = seq(0, 1, length = 1e+05)

ef = function(x) {
    dnorm(x)/dcauchy(x, 0, 1)
}

k_star <- max(ef(x_grid))
k = k_star


# value of the maximum of the ratio between the standard
# normal and the standard cauchy
k
```

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

```
curve(k * dcauchy(x, 0, 1), xlab = "x", ylab = expression(f[X](x)),
    lwd = 2, xlim = c(-5, 5), ylim = c(0, 0.6), col = "blue")
curve(dnorm(x), add = TRUE, col = "red", lwd = 2)
legend(x = -5, y = 0.6, lty = 1, lwd = 2.4, col = c("red", "blue"),
    inset = c(-0.1, 0), legend = c("target density  ", " bounding function"))
title(main = "A/R")
```



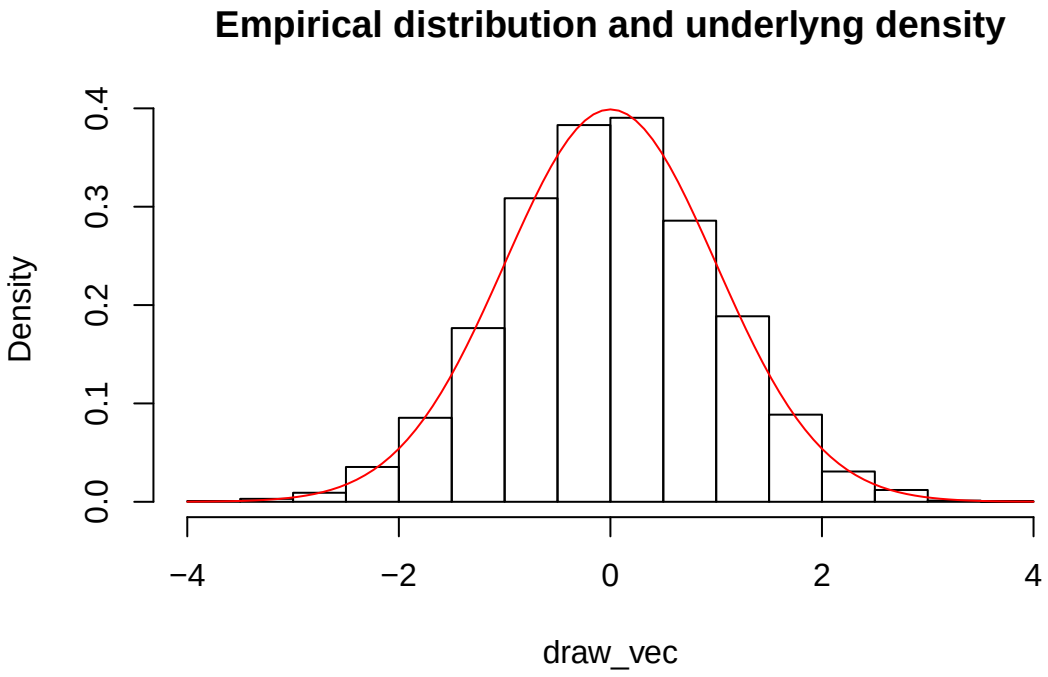
Implementation of AR algortithm

```
q = function(x) {
    dcauchy(x)
}
```

```r
draw_from_q = function(n) {
    rcauchy(n)
}

f = function(x) {
    dnorm(x)
}

AR = function(dtarget, dauxiliary, rauxiliary, k) {

    count = 0
    E = 0

    while (E == 0) {
        candidate = rauxiliary(1)
        acc_prob = dtarget(candidate)/(k * dauxiliary(candidate))
        E = sample(c(1, 0), prob = c(acc_prob, 1 - acc_prob),
            size = 1)
        count = count + 1
    }

    return(list(draw = candidate, computational_effort = count))

}

AR(dtarget = q, dauxiliary = q, rauxiliary = draw_from_q, k)
```

```
## $draw
## [1] -0.1892392
##
## $computational_effort
## [1] 3
```

```r
mcsize = 10000
draw_vec = rep(NA, mcsize)
effort_vec = rep(NA, mcsize)

for (i in 1:mcsize) {

    DD = AR(dtarget = f, dauxiliary = q, rauxiliary = draw_from_q,
        k = k_star)
    draw_vec[i] = DD$draw
    effort_vec[i] = DD$computational_effort

}

hist(draw_vec, freq = FALSE, main = "Empirical distribution and underlyng density")
curve(f(x), add = TRUE, col = "red", lwd = 1)
```



c) evaluate numerically (approximately by MC) the acceptance probability.

Empirical acceptance probability using MC:

```
acc_prob_emp <- mcsize/sum(effort_vec)
acc_prob_emp
```

```
## [1] 0.6609822
```

Let's compare with the theoretical acceptance probability:

```
acc_prob_theo <- 1/k
acc_prob_theo
```

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

Great! They are very similar!

d) write your theretical explanation about how you have conceived your Monte Carlo estimate of the acceptance probability.

The theoretical acceptance probability depends on the joint distribution of (Y,E). We can get this probability from this formula:

$$P(E=1) = \int_0^1 P(E=1|Y=y) f_U(y) dy = \int_0^1 \frac{f_X(y)}{k f_U(y)} f_U dy = \int_0^1 \frac{f_X(y)}{k} yx = \frac{1}{k} \int_0^1 f_X(y) dy = \frac{1}{k}$$

.

We can get an estimate of the acceptance probability using Monte Carlo simulation on our distribution. From the Monte Carlo simulation propriety's follows that :

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

From the slides: For unknown quantity of interest t which are functionals of some fixed known distribution, one can use consistent estimators based on i.i.d. simulations in order to get an approximation for the corresponding a.s. limit K.

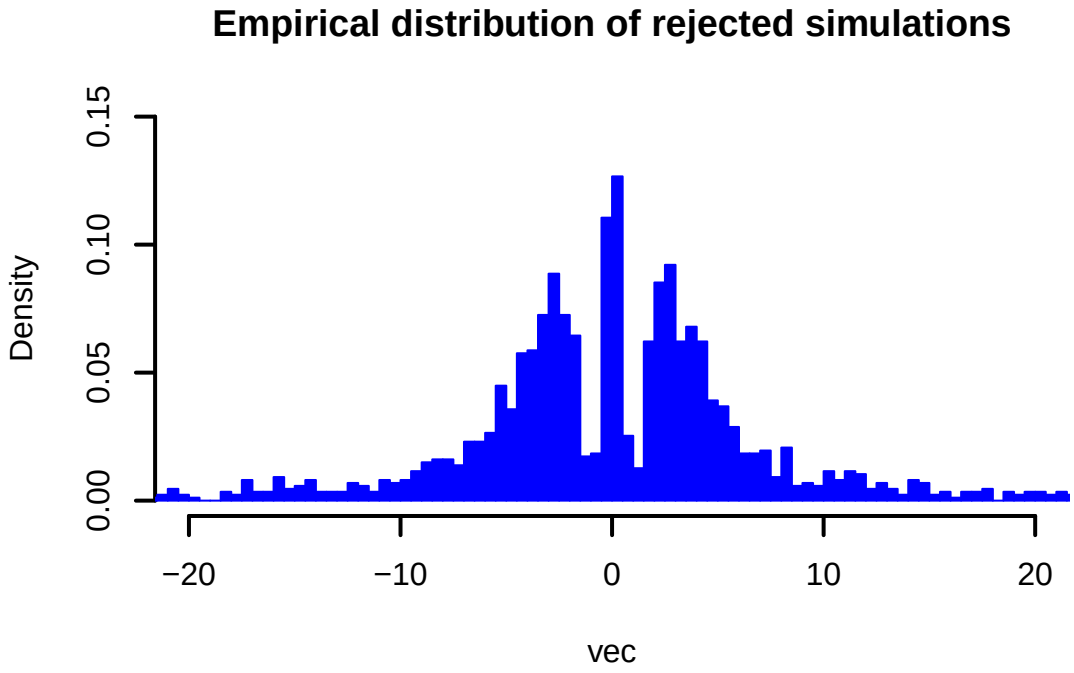
e) save the rejected simulations and provide a graphical representation of the empirical distribution. (histogram or density estimation)

```
count = 0
E = 0
reject <- NULL

AR_new = function(dtarget, dauxiliary, rauxiliary, k) {
    while (E == 0) {
        candidate = rauxiliary(1)
        acc_prob = dtarget(candidate)/(k * dauxiliary(candidate))
        E = sample(c(1, 0), prob = c(acc_prob, 1 - acc_prob),
            size = 1)
        if (E == 0) {
            reject[count] <- candidate
        }
        count = count + 1
    }

    return(list(draw = candidate, computational_effort = count,
        rej = reject))

}


AR_new(dtarget = q, dauxiliary = q, rauxiliary = draw_from_q,
    k)
```

```
## $draw
## [1] 0.5481551
##
## $computational_effort
## [1] 1
##
## $rej
## NULL
```

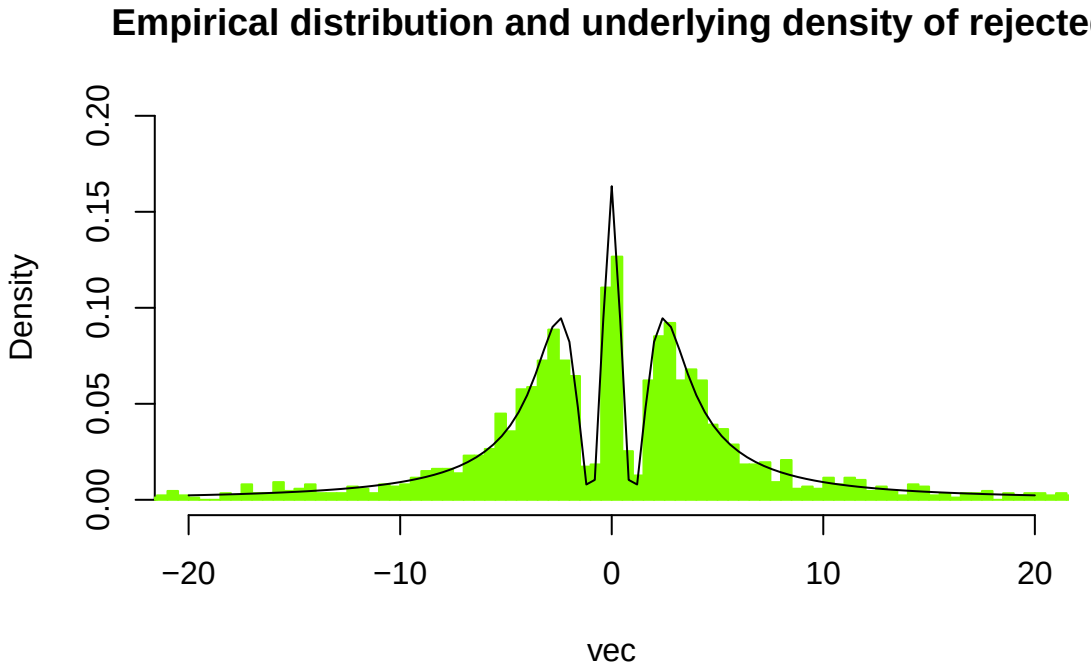
```
mcsize = 10000
draw_vec = rep(NA, mcsize)
effort_vec = rep(NA, mcsize)
vec = c()

for (i in 1:mcsize) {

    DD = AR_new(dtarget = f, dauxiliary = q, rauxiliary = draw_from_q,
        k = k_star)
    draw_vec[i] = DD$draw
    effort_vec[i] = DD$computational_effort
    vec <- c(vec, DD$rej)

}
```

```r
hist(vec, breaks = 10000, xlim = c(-20, 20), ylim = c(0, 0.15),
    freq = FALSE, lwd = 2, col = "blue1", border = "blue1", main = "Empirical distribution of rejected s
```



f) derive the underlying density corresponding to the rejected random variables and try to compare it with the empirical distribution

```r
hist(vec, breaks = 10000, freq = FALSE, xlim = c(-20, 20), ylim = c(0,
    0.2), col = "chartreuse", border = "chartreuse", main = "Empirical distribution and underlying densi
rejected_distr <- function(x) (dcauchy(x) * (1 - (dnorm(x))/(k *
    dcauchy(x)))/(1 - 1/k))
curve(rejected_distr(x), add = TRUE, lwd = 1, col = "black")
```



PART 2

Marginal likelihood evaluation for a Poisson data model. Simulate 10 observations from a known Poisson distribution with expected value 2. Use set.seed(123) before starting your simulation. Use a Gamma(1,1) prior distribution and compute the corresponding marginal likelihood in 3 differnt ways:

a) exact analytic computation

$$m(y) = \int_\Theta L(y|\theta)d\theta = \int_\Theta \prod_{i=1}^{n} f(y|\theta)\pi(\theta) = \int_0^\infty \prod_{i=1}^{n} (\frac{1}{y_i!}\theta^{\sum_{i=1}^{n} y_i} e^{n\theta}) \frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta} =$$

$$= \frac{b^a}{\Gamma(a)} \int_0^\infty (\frac{1}{\prod_{i=1}^{n} y_i!}\theta^{\sum_{i=1}^{n} y_i} e^{-n\theta})(\theta^{a-1}e^{-b\theta}) =$$

$$\frac{b^a}{\Gamma(a)\prod_{i=1}^{n} y_i!} \int_0^\infty (\theta)^{\sum i=1^n y_i} e^{-n\theta})(\theta^{a-1}e^{-b\theta}) = \frac{b^a}{\Gamma(a)\prod_{i=1}^{n} y_i!} \int_0^\infty \theta^{a+\sum_{i=1}^{n} y_i - 1} e^{(b+n)\theta}$$

.

From the gamma distribution property:

$$\int_0^\infty \frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta}d\theta = 1$$

.

This means that:

$$\int_0^\infty \theta^{a-1}e^{-b\theta}d\theta = \frac{\Gamma(a)}{b^a}$$

Now let's change

$$a + \sum_{i=1}^{n} y_i$$

for a and $b+n$ for b and we get:

$$\int_0^\infty \theta^{a+\sum_{i=1}^n y_i - 1} e^{-(b+n)\theta} = \frac{\Gamma(a+\sum_{i=1}^n y_i)}{(b+1)^{(a+\sum_{i=1}^n y_i)}}$$

.

Let's update the original integrate:

$$\frac{b^a}{\Gamma(a)\prod_{i=1}^n y_i!}\frac{\Gamma(a+\sum_{i=1}^n y_i)}{(b+n)^{(a+\sum_{1=1}^n y_i)}} = \frac{\Gamma(a+\sum_{i=1}^n y_i)}{\Gamma(a)\prod_{i=1}^n y_i!}(\frac{b}{b+n})^a + (\frac{1}{b+n})^{\sum_{i=1}^n y_i}$$

.

From the previous info we know that a=1 and b=1, so:

$$\frac{\Gamma(1+\sum_{i=1}^n y_i)}{\prod_{i=1}^n y_i!} + (\frac{1}{11})^{1+\sum_{i=1}^n y_i}$$

```
set.seed(123)

comp_marginal <- function(y_obs) {
    (gamma(1 + sum(y_obs))/(prod(factorial(y_obs)))) * (1/(1 +
        length(y_obs)))^(1 + sum(y_obs))
}

y_obs <- rpois(10, 2)
marg <- comp_marginal(y_obs)
marg
```

```
## [1] 4.315028e-09
```

b) by Monte Carlo approximation using a sample form the posterior distribution and the harmonic mean approach. Try to evaluate random behaviour by repeating/iterating the approximation $\hat{I}$ a sufficiently large number of times and show that the approximation tends to be (positively) biased. Use these simulations to evaluate approximately the corresponding variance and mean square error

$$\varepsilon = m(x) = \int_\Theta L_x(\theta)\pi(\theta)d\theta$$

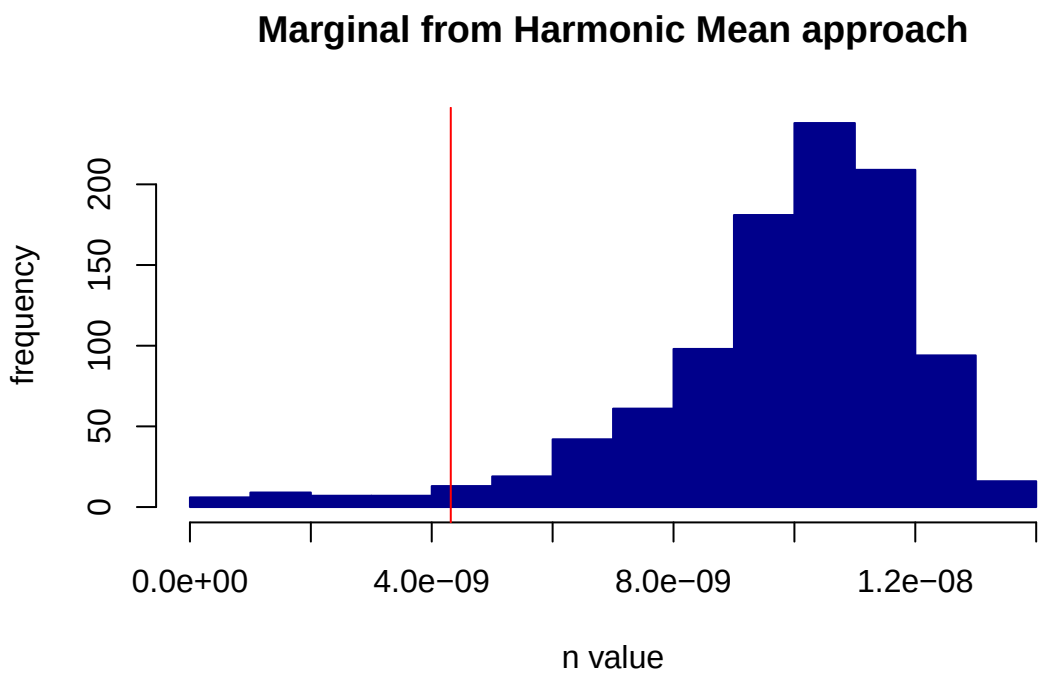
It's possible get an estimate of ε through 'Harmonic mean' approach: Simulate (MCMC sample) $\theta_1, ..., \theta_t$ from the posterior:. $\pi(\theta|x)$ ->

$$\hat{\varepsilon}^{HC} = \frac{1}{\frac{1}{t}\sum_{i=1}^t \frac{1}{L(\theta_i)}}$$

, . where $q(\theta)$ is a suitable probability density possibly close to the posterior.

```
h_m_function <- function(x) {
    1/((1/length(x)) * sum(1/x))
}


n1 = 10000
n2 = 1000
```

```
list_outcome = rep(NA, n2)

for (i in 1:n2) {
    theta_post = rgamma(n1, 1 + sum(y_obs), 1 + length(y_obs))
    outcome = rep(NA, n1)
    pos = 0

    for (j in theta_post) {
        pos = pos + 1
        Likelihood_ = prod(dpois(y_obs, j))
        outcome[pos] = Likelihood_
    }

    list_outcome[i] = h_m_function(outcome)
}


hist(list_outcome, main = "Marginal from Harmonic Mean approach",
    ylab = "frequency", xlab = "n value", col = "darkblue", border = "darkblue")
abline(v = marg, col = "red")
```




```
bias = mean(list_outcome) - marg
bias
```

```
## [1] 5.470616e-09
```

```
variance = var(list_outcome)
variance
```

```
## [1] 4.997171e-18
```

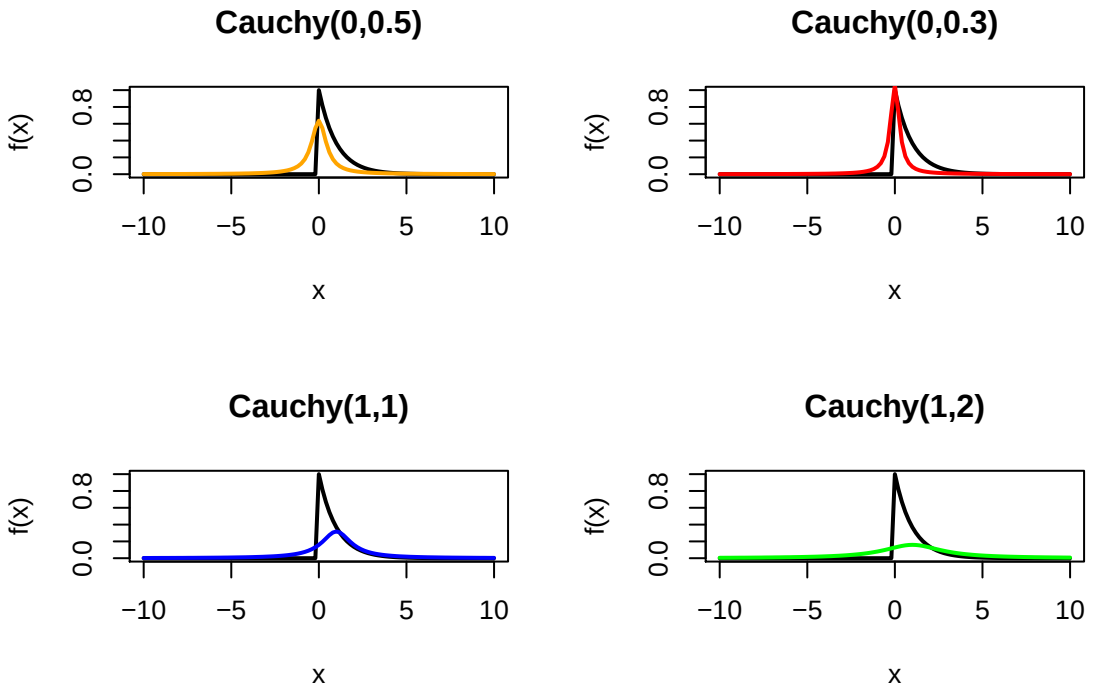
```
MSE = variance + bias^2
MSE
```

```
## [1] 3.492481e-17
```

c) by Monte Carlo Importance sampling choosing an appropriate Cauchy distribution as auxiliary distribution for the simulation. Compare its performance with respect to the previous harmonic mean approach.

First step is the choice of rigth parameters for the Cauchy distribution. Let's compare Gamma ~ (1,1) with Cauchy to varyng different values for a and b.

```
par(mfrow = c(2, 2))

curve(dgamma(x, 1, 1), xlim = c(-10, 10), lwd = 2, ylab = "f(x)",
    main = "Cauchy(0,0.5)")
curve(dcauchy(x, 0, 0.5), col = "orange", lwd = 2, xlim = c(-10,
    10), add = T)

curve(dgamma(x, 1, 1), xlim = c(-10, 10), lwd = 2, ylab = "f(x)",
    main = "Cauchy(0,0.3)")
curve(dcauchy(x, 0, 0.3), col = "red", lwd = 2, add = T, xlim = c(-10,
    10))

curve(dgamma(x, 1, 1), xlim = c(-10, 10), lwd = 2, ylab = "f(x)",
    main = "Cauchy(1,1)")
curve(dcauchy(x, 1, 1), col = "blue", lwd = 2, add = T, xlim = c(-10,
    10))

curve(dgamma(x, 1, 1), xlim = c(-10, 10), lwd = 2, ylab = "f(x)",
    main = "Cauchy(1,2)")
curve(dcauchy(x, 1, 2), col = "green", lwd = 2, add = T, xlim = c(-10,
    10))
```



The plots shows that the distribution more similar to Gamma ~ (1,1) is Cauchy ~ (0,0.30). I chose these parameters for the distribution. Then let's implement Importance Sampling:

```
a = 0
b = 0.3
```

```r
L1 <- function(y_obs, theta) {
    # Likelihood
    prod(dpois(y_obs, theta))
}

IS <- function(a, b, y_obs) {
    h_r_list <- c()
    theta_simul <- abs(rcauchy(1000, a, b))
    for (i in theta_simul) {
        h_r <- L1(y_obs, i) * dgamma(i, 1, 1)/dcauchy(i, a, b)
        h_r_list <- c(h_r_list, h_r)
    }
    I_hat <- cumsum(h_r_list)/(1:length(h_r_list))
}

mc_size = 1000

I_hat_lis <- c()

for (i in 1:mc_size) {
    I_hat <- IS(a, b, y_obs)
    I_hat_lis <- c(I_hat_lis, I_hat)
}

bias2 <- mean(I_hat_lis) - marg
bias2
```

```
## [1] 4.356259e-09
```

```r
variance2 <- var(I_hat_lis)
variance2
```

```
## [1] 8.366167e-18
```

```r
MSE2 = variance2 + bias2^2
MSE2
```

```
## [1] 2.734316e-17
```

```r
bias/bias2
```

```
## [1] 1.255806
```

```r
MSE/MSE2
```

```
## [1] 1.277278
```

The ratio between the two MSE,in the same way as the ratio between the two bias, is > 0. It means that IS method is more accurate than the Harmonic mean approach for estimating the marginal likelihood.