Stat4DS II - Homework#1

Iason Tsardanidis - 1846834

May 4, 2019

1) A-R algorithm

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:

Using the Accept/Reject Method we can construct an algorithm that simulates from the standard Normal Distribution using as an auxiliary function the standard Cauch Distribution. More specifically:

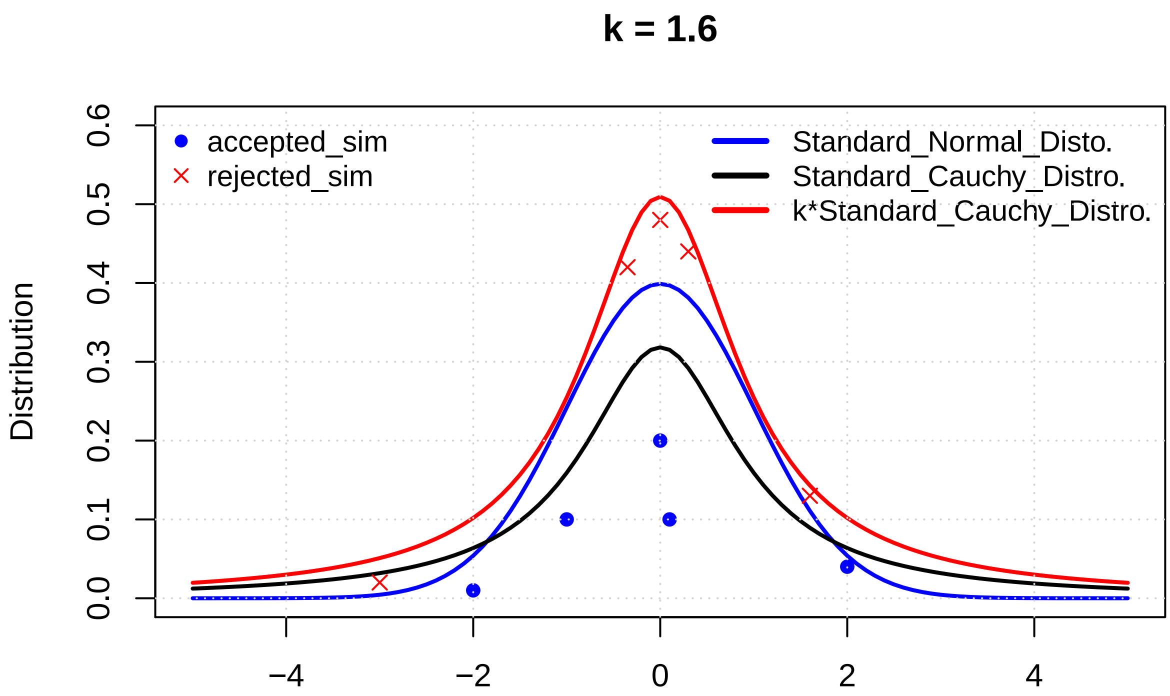
Defining as $f(x)$ the Normal Distribution we want to simulate for and $q(x)$ the Normal Cauchy Distribution we can extract pseudo-random numbers from the later inside a specific range $[-\alpha, \alpha]$ and *accept* them using the condition above

$$\frac{f(x)}{kq(x)} \in [0, 1]$$

where k is a properly positive finite number in order the auxiliary function dominates the target one in all the $[-\alpha, \alpha]$ range.

In other words we want to construct an auxiliary distro that covers/dominates our desired target distribution.

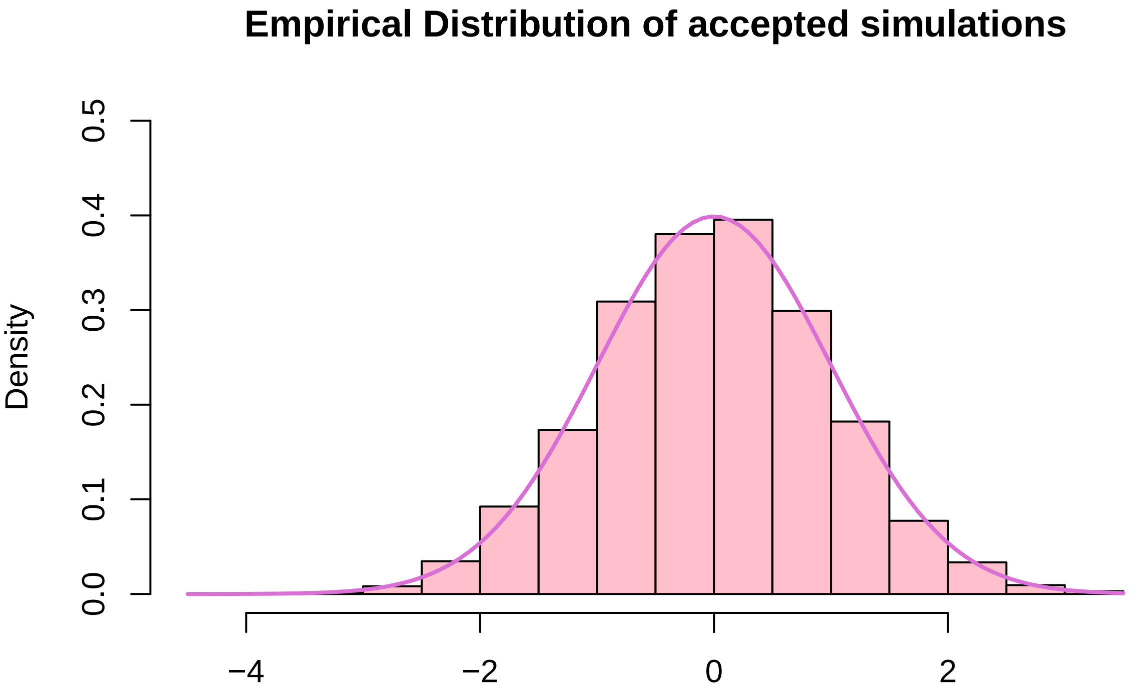
```r
k <- 1.6 # suitable k multiplier
curve(dnorm(x),col="blue",lwd=2,from = -5,to = 5,ylim = c(0,0.6),
      ylab='Distribution',xlab='')
curve(dcauchy(x),col="black",lwd=2,from = -5,to = 5,add=TRUE)
curve(dcauchy(x)*k,col="red",lwd=2,from = -5,to = 5,add=TRUE)
title(main = 'k = 1.6')
legend('topright',
      c("Standard_Normal_Disto.", "Standard_Cauchy_Distro.", "k*Standard_Cauchy_Distro."),
      col = c('blue','black','red'), lwd = 3, bty = "n",cex = 0.9)
legend('topleft',c('accepted_sim','rejected_sim'),
       pch=c(16,4),col=c('blue','red'),bty='n',cex=0.9)
points(0,0.2, pch=16,col='blue')
points(2,0.04, pch=16,col='blue')
points(0.1,0.1, pch=16,col='blue')
points(-1,0.1, pch=16,col='blue')
points(-2,0.01, pch=16,col='blue')
points(0,0.48, pch=4,col='red')
points(0.3,0.44, pch=4,col='red')
points(-3,0.02, pch=4,col='red')
points(1.6,0.13, pch=4,col='red')
points(-0.35,0.42, pch=4,col='red')
grid()
```



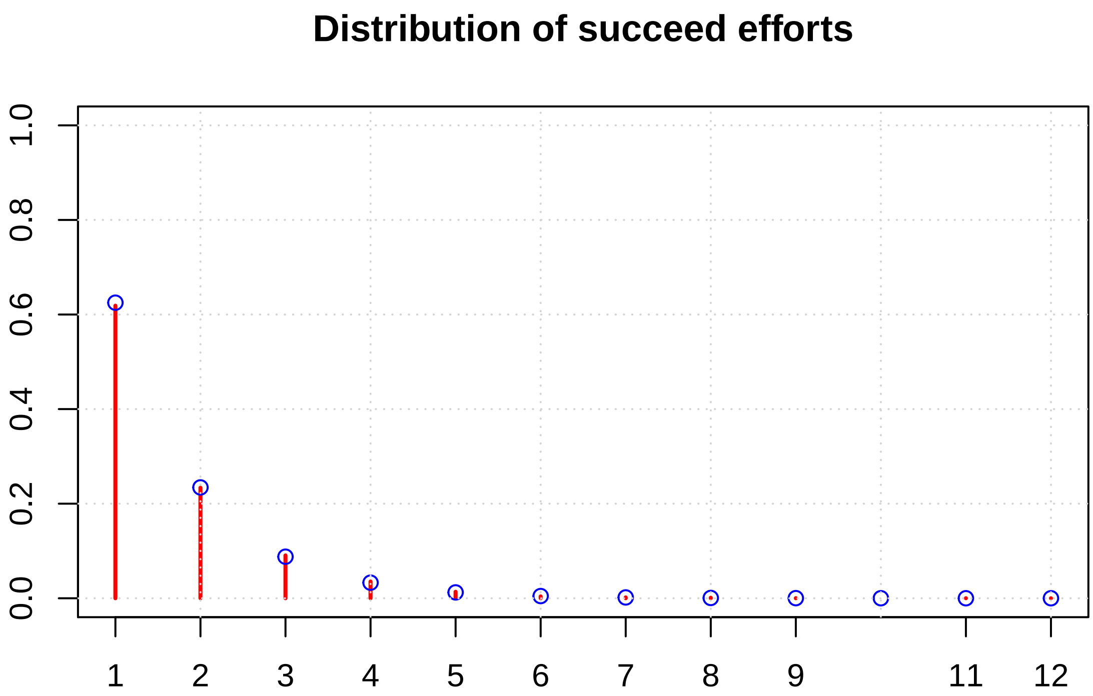
b) provide your R code for the implementation of the A-R:

```
#auxiliary function
q=function(x){
  dcauchy(x)
}
#random draws from the auxiliary distribution
draw_from_q=function(n){
  rcauchy(n)
}

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

#Accept-Reject algorithm
AR=function(dtarget,dauxiliary,rauxiliary,k){

  count=0
  E=0
  rejected = c()

  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) {
      rejected = c(rejected,candidate)
    }
    count=count+1
  }
```

```r
  return(list(draw=candidate,computational_effort=count,rejected_sim = rejected))

}


mcsize=10000  #|<--------- number of simulations
draw_vec=rep(NA,mcsize)
effort_vec=rep(NA,mcsize)
neg_effort_vec=rep(NA,mcsize)
rejected = c()

for(i in 1:mcsize){

  DD=AR(dtarget=f,dauxiliary=q,rauxiliary=draw_from_q,k=k)
  draw_vec[i] = DD$draw
  effort_vec[i] = DD$computational_effort
  neg_effort_vec[i] = effort_vec[i]-1
  rejected = c(rejected,DD$rejected_sim)

}

hist(draw_vec,freq=FALSE,col='pink',xlab='',ylim=c(0,0.5),
     main='Empirical Distribution of accepted simulations')
curve(f(x),add=TRUE,col='orchid',lwd=2)
```



```r
plot(prop.table(table(effort_vec)),ylim=c(0,1),pch=16,col="red",xlab='',ylab='',
     main='Distribution of succeed efforts')
points(1:20,dgeom(0:19,prob=1/k),col='blue')
grid()
```



From the plots above we can see that in the first one, the derived empirical distribution of the accepted samples and the theoretical of standard normal are in very good accordance. Also the second plot shows the distribution of number of efforts until we success. We expect to follow a geometrical distribution with probability of success p $= \frac{1}{k} = 0.625$, something that is very clear in the plot.

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

The expected acceptance probability of success is:

$$p = \frac{1}{k} = 0.625$$

We can approximately evaluate numerically this probability by dividing the MC simulation size by the total sum of the efforts until the success:

```r
mcsize/sum(effort_vec)
```

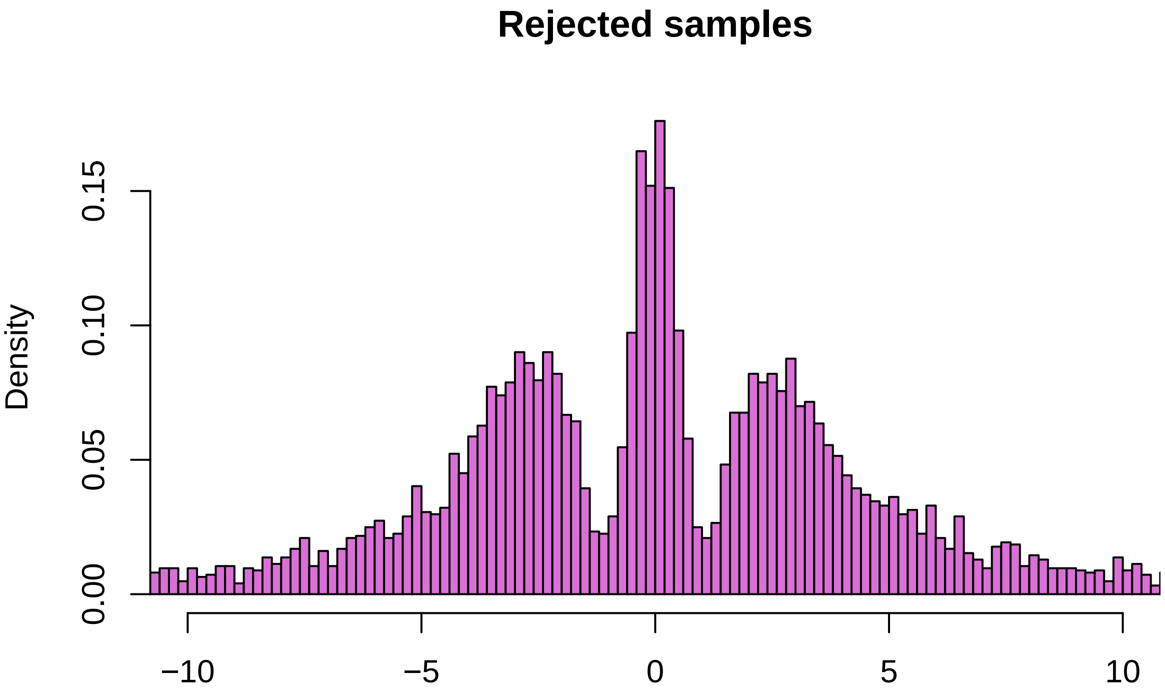
```
## [1] 0.6165608
```

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

The smaller k we are allowed to choose in order the auxiliary distribution overlaps the target one the better probabilty of success we have. Hence, we can approximate better with smaller MC-size/loops the probability p of success. The auxiliary function can generate pseudo-random numbers that have bigger probability belong into the target distribution and as a result smaller probability to get rejected since the non intersection area between the k-times multiplied auxiliary and the target distributions become smaller.

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

```r
hist(rejected,prob=TRUE,breaks =100000,xlim=c(-10,10),
     col='orchid',main = 'Rejected samples',xlab='')
```



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

From the question a) we can easily infer that the distribution of the rejected samples is equal with difference between the k*times - auxiliary distribution (in our case the standard cauchy distro.) and the target distribution (standard normal). So we can say that that the distribution of the rejected simulation is:

$$d(x) = kq(x) - f(x)$$

that comes from the conditional probability of cdf: $\mathbb{P}(x \leq \mathbb{X}|E = 0) = \frac{\mathbb{P}(x \leq \mathbb{X}, E=0)}{\mathbb{P}(E=0)} = \frac{\mathbb{P}(x \leq \mathbb{X}, E=0)}{1-\mathbb{P}(E=1)}$

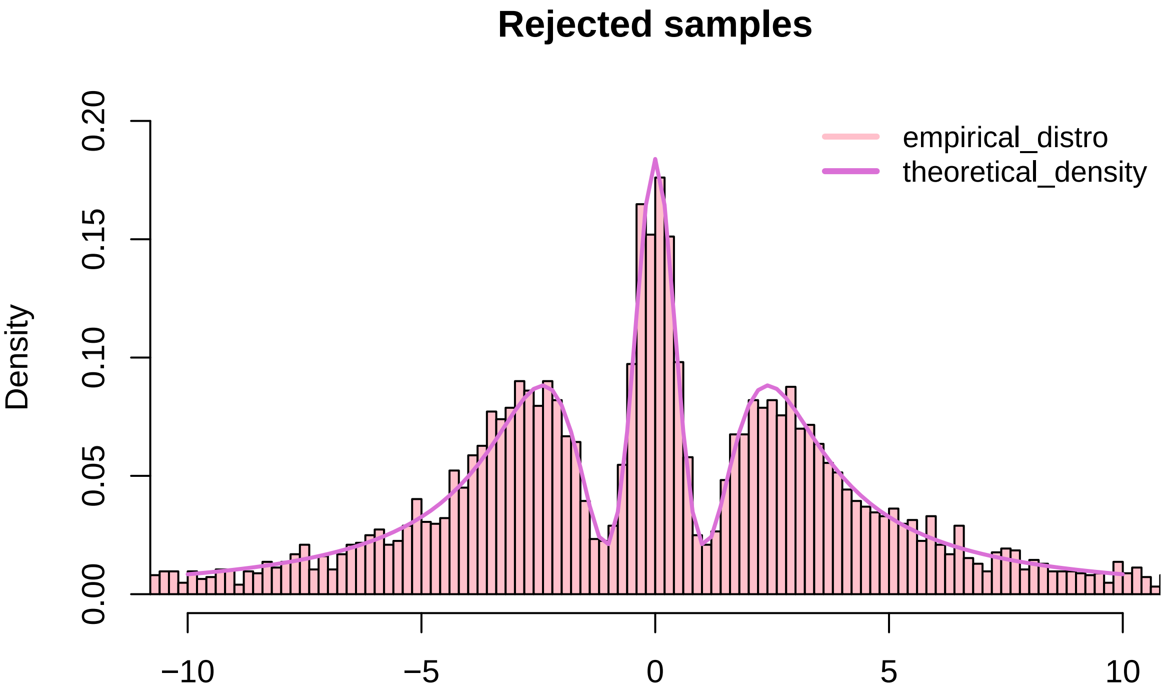
Since the cumulative distribution must sum up to one, we have to divide $d(x)$ with a normalizing constant c. That can be computed as follows:

$$c \int_{\mathbb{X}} kq(x) - f(x)dx = 1, x \in \mathbb{X} \Rightarrow c = \frac{1}{k-1}$$

So the normalized distribution function of the rejected Monte-Carlo samples at the end is:

$$d(x) = \frac{kq(x) - f(x)}{k-1}$$

```r
hist(rejected,prob=TRUE,breaks = 100000,xlim=c(-10,10),
     ylim=c(0,0.2),col='pink',main='Rejected samples',xlab='')
d=function(x){
  (k*q(x)-f(x))/(k-1)
}
curve(d(x),add=TRUE,col='orchid',lwd=2)
legend('topright',c('empirical_distro','theoretical_density'),
       col=c('pink','orchid'),bty='n',cex=0.9,lwd=3)
```



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:

The *random variable* X, distributed according to the *Poisson density*:

$$f_Y(y|\theta) = \frac{\theta^y e^{-\theta}}{y!}$$

and a given $Gamma(1,1)$ *prior distribution* for θ:

$$\pi(\theta) = Gamma(\theta|1,1) = \frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta} = e^{-\theta}$$

Also, we have to simulate 10 observations from a known a Poisson distribution with expected value 2:

```r
set.seed(123)
X_obs = rpois(10, 2)
```

The *marginal likelihood* function for the *single-variate* case can be estimated as:

$$m_Y(y) = \mathbb{P}(Y=y) = \int_\Theta f(Y=y|\theta)\pi(\theta)d\theta = \int_0^{+\infty}\theta^y\frac{e^{-\theta}}{y!}e^{-\theta}d\theta = \frac{1}{y!}\int_0^{+\infty}\theta^y e^{-2\theta}d\theta$$

which can be solved using the *normalization condition* of the known Gamma distribution:

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

Using the above we can dirive the *marginal likelihood*, which a *negative binomial* distribution:

$$NB(Y = y|r, p) = \binom{y + r - 1}{y}(1 - p)^r p^y$$

with $p = \frac{1}{2}$ and $r = 1$.

Now we can compute the *marginal likelihood* for the observations $(Y_1 = y_1, Y_2 = y_2, ...Y_{10} = y_10)$, and the *multi-variate* case with the same procedure:

$$m_{Y_1,Y_2,...Y_n}(y_1, y_2...y_n) = \int_\Theta \prod_{i=1}^{n} \theta^{y_i} \frac{e^{-\theta}}{y_i!} e^{-\theta} d\theta = \frac{1}{\displaystyle\prod_{i=1}^{n} y_i!} \int_\Theta \theta^{\sum y_i} e^{-(n+1)\theta} d\theta$$

and using the normalization condition of the Gamma distribution above:

with $a - 1 = \displaystyle\sum_{i=1}^{n} y_i$ and $b = (n + 1)$, we get:

$$m_{Y_1,Y_2,...Y_n}(y_1, y_2...y_n) = \frac{1}{\prod_{i=1}^{n} y_i!} \frac{\Gamma(\displaystyle\sum_{i=1}^{n} y_i + 1)}{(n + 1)^{\sum y_i + 1}}$$

We can see that the distribution above is a *negative multinomial* distribution:

$$NM_{Y_1,Y_2,...Y_n}(y_1, y_2...y_n|k_0, p_0, p_1, ...p_n) = \Gamma(\sum_{i=1}^{n} y_i + k_0) \frac{p_0^{k_0}}{\Gamma(k_0)} \prod_{i=1}^{n} \frac{p_i^{y_i}}{y_i!}$$

with $p_1 = p_2 = ... = p_n = \frac{1}{n+1}$,

$p_0 = 1 - \sum_{i=1}^{n} p_i = 1 - \frac{n}{n+1} = \frac{1}{n+1}$ and

$k_0 = 1$

Now we can compute analytical the integral and the marginal distribution for the given observation:

```
#analytical computation of the integral
f = function(X) prod(1/factorial(X))*gamma(sum(X)+1)/(length(X)+1)^(sum(X)+1)
I <- f(X_obs)
print(I)
```

```
## [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 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:

In order to compute the **estimator** of the marginal likelihood via the **harmonic mean** method:

$$\hat{I}_{HM} = \frac{1}{\displaystyle\frac{1}{t} \prod_{i=1}^{n} \frac{1}{L(\theta_i)}} = \frac{1}{\displaystyle\frac{1}{t} \prod_{i=1}^{n} \frac{1}{f(Y_1 = y_1, Y_2 = y_2, ...Y_n = y_n|\theta_i)}}$$

where the values $\theta_1, \theta_2, ...\theta_t$ are extracted from the posterior distribution that we can produce using our background in *conjugate analysis* and *Poisson/Gamma* model.

Since the prior is a $Gamma(\theta|\alpha = 1, \beta = 1)$, the posterior is:

$$\pi(\theta|y_1, y_2, ...y_n) = Gamma(\theta|\alpha + \sum_{i=1}^{n} y_i, \beta + n)$$

with known parameters $\alpha = 1$ and $\beta = 1$

```
#Harmonic Mean Estimation
nsim <- 1000

I_hat_HM = function(X,theta){

  # Poisson likelihood function
  likelihood <- function(X,theta) prod(dpois(X,theta))

  S <- c()

  for (i in 1:length(theta)){
      S[i] <- likelihood(X_obs,theta[i])
  }
  return(1/mean(1/S))
}



I_hat_HM_values = c()

# simulate for nsim times
for (k in 1:nsim){
  #random draws of theta samples taken from the posterior Gamma distribution
  theta_sim = rgamma(1000, shape = 1 + sum(X_obs), rate = 1 + length(X_obs))
  I_hat_HM_values = c(I_hat_HM_values, I_hat_HM(X_obs,theta_sim))
}

#Estimator mean
mu_HM = mean(I_hat_HM_values)
#Bias
bias = mu_HM - I
sigma2_HM = var(I_hat_HM_values)
#MSE
MSE = bias^2 + sigma2_HM

c(Harmonic_Mean_Estimator=mu_HM,Bias=bias,MSE=MSE)
```

```
## Harmonic_Mean_Estimator                    Bias                     MSE
##            1.136818e-08            7.053154e-09            5.880438e-17
```

We can see that the bias is positevily defined.

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

We can compute an estimator for the marginal likelihood to the *importance sampling* method using:

$$m_{Y_1,Y_2,...Y_n}(y_1,y_2...y_n) = \int_\Theta \frac{L(\theta)\pi(\theta)}{q(\theta)} q(\theta)d\theta = \mathbb{E}_q\left[\frac{L(\theta)\pi(\theta)}{q(\theta)}\right]$$

that can be approximated as:

$$\hat{I}_{IS} = \frac{1}{t}\sum_{i=0}^{t} \frac{L(\theta_i)\pi(\theta_i)}{q(\theta_i)}$$

where $\theta_1, \theta_2, ...\theta_t$ are random draws from $q(\theta)$, which is an appropriate auxiliary distribution.

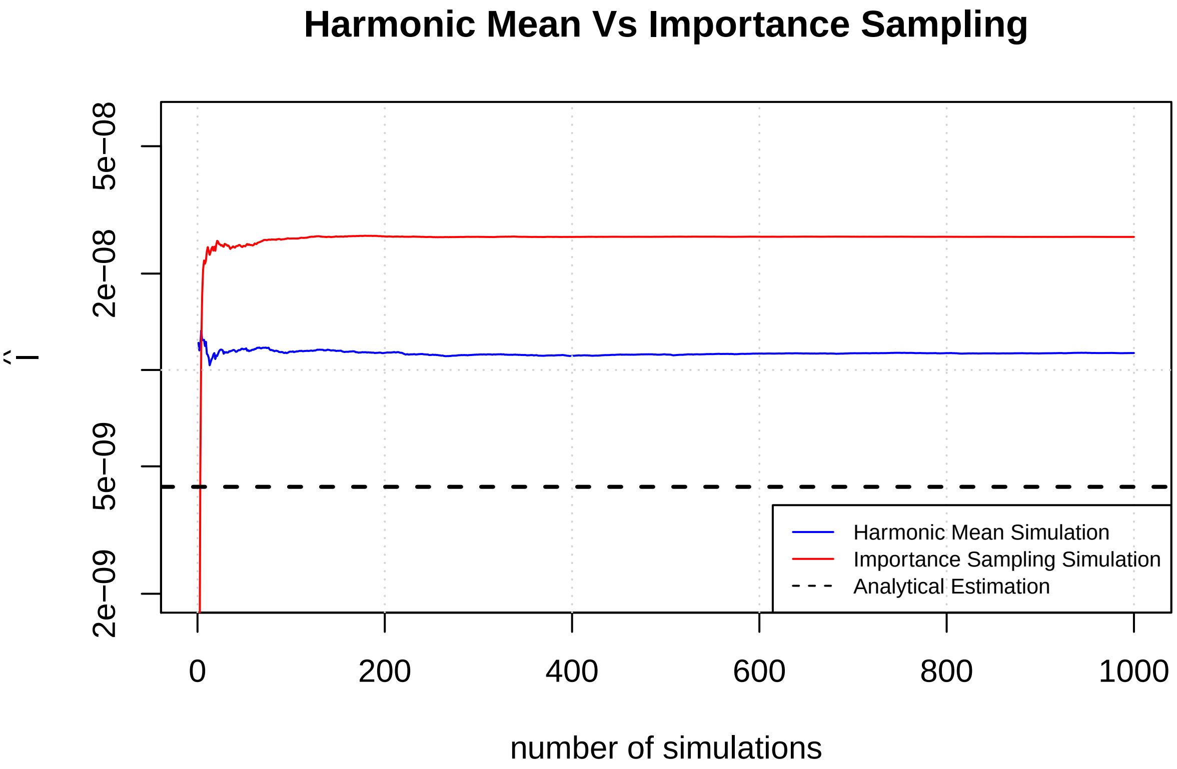
IWe will choose as a Cauchy distribution $q(\theta) = Cauchy(\theta|0,2)$:

```r
#Importance Sampling Estimation
nsim <- 1000
theta_sim = rcauchy(nsim,0,2) #random draws theta

#likelihood function
likelihood = function(X) prod(dpois(X, 2))

#prior distribution
prior = function(t) dgamma(t, shape = 1, rate = 1)

q = function(t) dcauchy(t, 0,2)


I_hat_IS = function(X, t){
  return( likelihood(X) * prior(t) / q(t) )
}


I_hat_IS_values = c()


for (i in 1:nsim){
  I_hat_IS_values = c(I_hat_IS_values, I_hat_IS(X_obs, theta_sim[i]))
}

#Estimator Mean
mu_IS = mean(I_hat_IS_values)
#Bias
bias = mu_IS - I
sigma2_IS = var(I_hat_IS_values)
#MSE
MSE = bias^2 + sigma2_IS

c(Importance_Sampling_Estimator=mu_IS,Bias=bias,MSE=MSE)
```

```
## Importance_Sampling_Estimator                          Bias
##                  2.365630e-08                  1.934127e-08
##                           MSE
##                  1.940931e-15
```

We can compare the performace of the two estimators in the plot below:

```r
#Monte-Carlo samples size
nsim = 1000

#Harmonic-Mean
r1 = c()
for (k in 1:nsim){
  theta_sim <- rgamma(1000, shape = 1 + sum(X_obs), rate = 1 + length(X_obs))
  r1 = c(r1,mean(I_hat_HM(X_obs,theta_sim)))
}

#Importance-Sampling
r2 = c()
for (k in 1:nsim){
  theta_sim <- rcauchy(k,0,2)
  r2 = c(r2,mean(I_hat_IS(X_obs,theta_sim)))
}

r1 = cumsum(r1)/(1:nsim)
r2 = cumsum(r2)/(1:nsim)
plot(1:nsim,r1,type="l",col="blue",
     xlab="number of simulations",ylab=expression(hat(I)),
     ylim=c(2e-9,6e-8),log="y")
lines(1:nsim,r2,col="red")
grid()
abline(h=I,col='black',lwd=2,lty=2)
title(main="Harmonic Mean Vs Importance Sampling")
legend(x="bottomright",
       legend=c("Harmonic Mean Simulation",
                "Importance Sampling Simulation","Analytical Estimation"),
       lty=c(1,1,2),col=c("blue","red","black"),cex=0.7)
```



We can see that the HM estimator is closer to the theoretical value than the IS as the number of simulations increases. Theoretically though, we except the IS estimator to give us better estimation for the most appropriate Cauchy distribution we put.