Statistical Methods For Data Science

Homework 1

Alessandro Gallo

Abstract

Due to some errors of Rmarkdown, the document has been written in Latex, uploading the codes and plots in the Rmarkdown style. The plots could be moved from Latex in its logical way in the respecting of the layout form

1 Bayesian Inference with a Bernoulli experiment

The Bernoulli experiment can be written as:

$$Ber(n, \theta) = \binom{n}{x} \theta^x (1-\theta)^{1-x} \tag{1}$$

1.1 The parametric space of interest

The parametric space of interest is:

$$\Theta = [0, 1] \tag{2}$$

1.2 Likelihood function

The Likelihood function is:

$$L_x(\theta) \propto \prod_{i=1}^{n} \left[\binom{n}{x_i} \theta^x (1-\theta)^{1-x} \right] \propto \theta^{\sum_{k=1}^{n} x_i} (1-\theta)^{n - \sum_{k=1}^{n} x_i} \tag{3}$$

And with 3 successes and 7 failures we have:

$$\propto \theta^3 (1-\theta)^7 \tag{4}$$

1.3 Prior distribution

We use a beta distribution for the prior:

$$Be(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{5}$$

That can be approximated as:

$$\pi(\theta) \propto \theta^{\alpha-1}(1-\alpha)^{\beta-1} \tag{6}$$

1.4 Posterior distribution

The posterior distribution is defined as:

$$\pi(\theta|x) \propto likelihood * prior \propto L_x \pi(\theta) \tag{7}$$

So in our case it becomes:

$$\pi(\theta|x) \propto \theta^{\sum_{k=1}^{n} x_i + \alpha - 1}(1-\theta)^{n - \sum_{k=1}^{n} x_i + \beta - 1} \tag{8}$$

That is a Beta distribution:

$$\propto B(\alpha', \beta') \propto \pi(\theta|x) \propto \theta^{\alpha'-1}(1-\theta)^{\beta'-1} \tag{9}$$

With:

$$\alpha' = \alpha + \sum_{i=1}^{n} x_i, \beta' = \beta + n - \sum_{i=1}^{n} x_i \tag{10}$$

1.5 How to make inference

We can make inference using summaries of $\pi(\theta|x)$ as point estimates, quantile-based intervals for credibility intervals or HPD (Highest Posterior Density) intervals and posterior probability statements for comparing two alternative hypothesis

2 The Dugongs

The Dugongs non-linear regression model is represented by:

$$Y_i \sim N(\mu_i, \tau^2) \tag{11}$$

2.1 Likelihood

The Likelihood of the function is:

$$L(\mu_i, \tau^2; y_n) = \prod_{k=1}^{n} (2\pi\tau^2)^{-1/2} exp[(-2\tau^2)^{-1} \sum_{k=1}^{n} (x_i - \mu)^2] \tag{12}$$

And assuming: $\mu_i = \alpha - \beta\gamma^{x_i}$ We can write it again:

$$L(\alpha, \beta, \gamma, \tau^2; y_n) = (2\pi\tau^2)^{-n/2} exp[(-2\tau^2)^{-1} \sum_{k=1}^{n} (x_i - \alpha + \beta\gamma^{x_i})^2] \tag{13}$$

Code in R:

```
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 = 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)
LL = function(theta){
  alpha = theta[1]
  beta = theta[2]
  gamma = theta[3]
  tau = theta[4]
  mu=alpha-beta*gamma^x
  lik = dnorm(y,mu, tau,log=T)
  return(sum(lik))
}
values=optim(c(1.5,2,0.3,1.2),LL,control=list(fnscale=-1))
res=values$par
```

Result:

```
[1] 2.65812211 0.96353596 0.87147283 0.08979761
```

We can plot so the likelihood function:

```
plot(x,y,col="blue",main="Dugongs likelihood")
lines(x,res[1]-res[2]*res[3]^x,type="l",col="red")
```

2.2 Maximum-a-Posteriori estimate

$$\theta_{MAP}(x) = argmax\pi(\theta|x) = L_x(\theta)\pi(\theta) = L(\alpha, \beta, \gamma, \tau^2; y_n)\pi(\alpha)\pi(\beta)\pi(\gamma)\pi(\tau^2) \tag{14}$$

Code in R:

```
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 = 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,
```

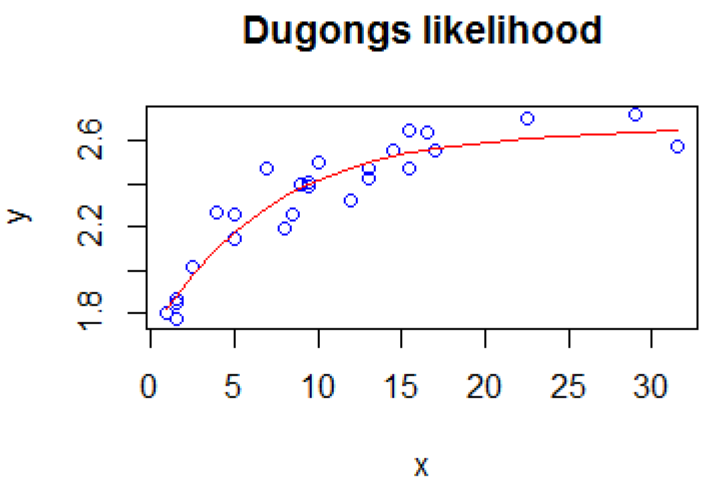

Figure 1: Dugongs likelihood of exercise 2.1

```
      2.47, 2.56, 2.65, 2.47, 2.64, 2.56, 2.70, 2.72, 2.57)
pialfa=function(alfa){
  val=dnorm(alfa,0,10000,log=T)
  return(val)
}
pibeta=function(beta){
  val=dnorm(beta,0,10000,log=T)
  return(val)
}
pigamma=function(gamma){
  val=dunif(gamma,0,1)
  return(log(val))
}
pitau=function(tau){
  val=densigamma(tau^2,0.001,0.001)
  return(log(val))
}
LLP=function(vec_param){
  alpha=vec_param[1]
  beta=vec_param[2]
  gamma=vec_param[3]
  tau=vec_param[4]
```

```
  mu=alpha-beta*gamma^x
  lik=sum(dnorm(y,mu,tau,log=T))

  return(lik+pialfa(alpha)+pitau(tau)+pibeta(beta)+pigamma(gamma))
}
thetas=optim(c(1.5,2,0.7,1),LLP, control=list(fnscale=-1))
res=thetas$par
```

Result:

```
[1] 2.65812211 0.96353596 0.87147283 0.08979761
```

Plot:

```
plot(x,y,col="blue")
lines(x,res[1]-res[2]*res[3]^x,type="l",col="green",add=TRUE)
```

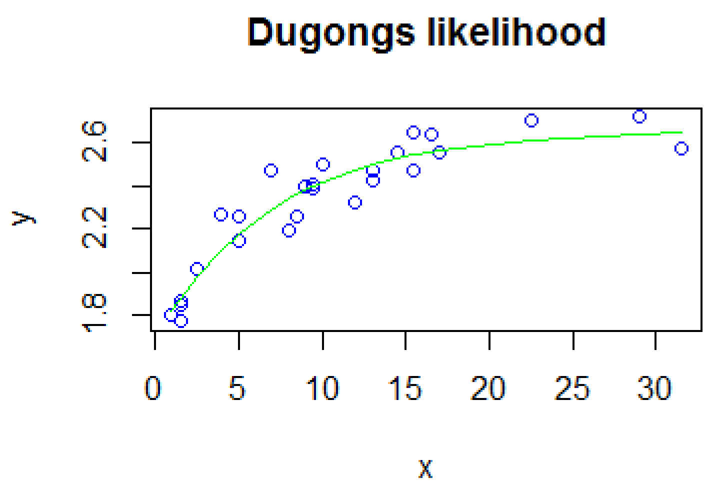

Figure 2: Plot of exercise 2.2

3 Normal distribution of an exponential function

$$F_y(y) = P(Y < y) = P(e^x < y) = P(x < ln(y)) \tag{15}$$

$$F_x(x) = F_x(ln(y)) \tag{16}$$

$$f_Y(y) = \frac{\partial F_y(y)}{\partial y} = \frac{\partial F_x(ln(y))}{\partial y} = f_x(ln(y))\frac{1}{y} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(\ln y - \mu)^2}{2\sigma^2} \right) \frac{1}{y} \tag{17}$$

4 Mean and Variance

The mean is:

$$E[Y] = E[e^x] = \int e^x f_x \, dx = \int (2\pi\sigma^2)^{-1/2} e^{-\frac{(x-y)^2}{2\sigma^2} + x} = e^{\mu + \frac{\sigma^2}{2}} \tag{18}$$

And the variance:

$$Var[Y] = Var[e^x] = E[(e^x)^2] - (E[e^x])^2 = e^{2\mu + \frac{\sigma^2}{2}} e^{\sigma^2 - 1} \tag{19}$$

5 Monte Carlo method with a Pareto distribution

5.1 Generating a sequence of random deviates

The inverse function of the Pareto distribution is:

$$F(x) = \int f(x; \alpha, \beta) dx = \int \frac{\alpha\beta^\alpha}{x^{\alpha+1}} I_{\beta,\infty}(x) = \int_\beta^y \frac{\alpha\beta^\alpha}{x^{\alpha+1}} dx = 1 - (\frac{\beta}{x})^\alpha \tag{20}$$

$$F^{-1}(x) = \frac{\beta}{(1-y)^{\frac{1}{\alpha}}} \tag{21}$$

Code in R:

```
N=1000
alpha=2.5
beta=1
invfun=function(n){
  res=beta/((1-n)^(1/alpha))
  return(res)
}
un=runif(N)
pareto=invfun(un)
hist(pareto,breaks=100,freq=FALSE,xlim=c(0,20), main = "Distribution")
dens=function(x){
  alpha*beta^(alpha +1)/(x^(alpha+1))*(x>beta)
}
curve(dens(x),from=0,to=20,col="red",add=TRUE,n=1000)
```

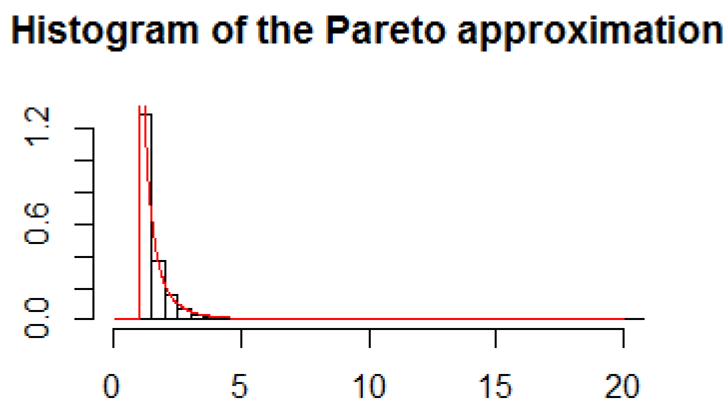

Figure 3: Pareto distribution of exercise 5.1

5.2 How to approximate the first moment

We can approximate the first moment to a precision of 10^{-2} with:

```
i=1
while(TRUE){
  if (2*sqrt(var/i)<0.01){
    print(i)
    break
  }
  i = i+1
}
```

Result:

```
[1] 88889
```

5.3 Approaching to I of the running means

We can see from the plot how the running mean approaches the target value I as the number of simulations grows.

Code in R:

```
curve(dens(x),from=0,to=20,col="red",add=TRUE,n=1000)
simul=pareto
I_hat=mean(pareto)
I_true=alpha/(alpha-1)
runningmeans=cumsum(simul)/(1:N)
runningmeans[N]
I_hat
runningmeans[N]==I_hat
plot(1:N,runningmeans,type="l",col="red",xlab="N",
main = "Approaching of the value of I")
abline(h=I_true,col="blue")
```

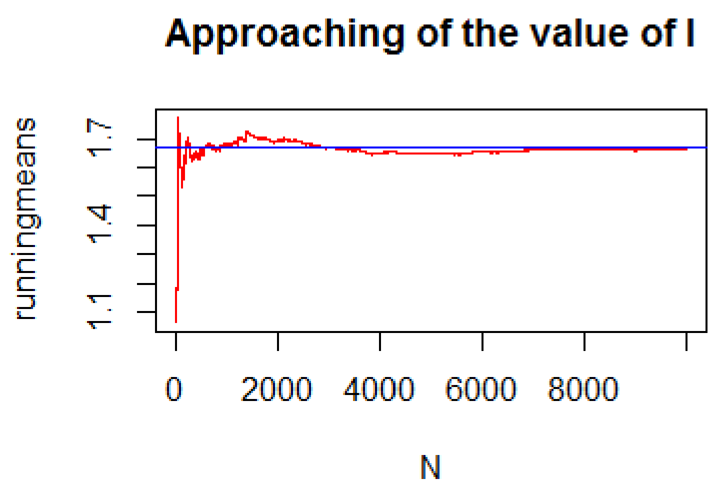

Figure 4: Approaching of the value of I

6 Simulate random deviates from a discrete distribution

The discrete distribution has a function:

$$F(x) = \begin{cases} 0 & x < 0 \\ 0.25 & 0 \leq x < 1 \\ 0.6 & 1 \leq x < 2 \\ 1 & 2 \leq x \end{cases}$$

We calculate the inverse:

$$F^{-1}(u) = \begin{cases} 0 & u \leq 0.25 \\ 1 & 0.25 < u \leq 0.6 \\ 2 & 0.6 < u \leq 1 \end{cases}$$

We can simulate 100 random deviates by putting a sample from a $U(0,1)$ in the inverse function, writing in R:

```
n=100
sequence=sample(c(0,1,2), size=n, replace=T, prob=c(0.25,0.35,0.4))
hist(sequence,breaks=100,probability=T)
un=runif(n)
invfun <- function(u){
 if (u<=0.25) {
   return(0)}
  else if (u<=0.6 && u>0.25){
    return(1)}
  else if (u<=1 && u>0.6){
    return(2)}
 }
vectfun=Vectorize(invfun)
vectfun(un)
```

Result:

```
[1] 1 0 1 1 2 1 2 2 1 0 2 2 2 2 0 2 2 2 ...
[45] 0 2 2 0 0 0 1 1 2 2 1 0 2 2 2 1 0 ...
[89] 2 2 0 0 2 0 1 0 0 1 2 2
```

7 Acceptance-Rejection with a Beta Distribution

We want to simulate from a Beta distribution:

$$f(x) = Beta(3,3) \tag{24}$$

using pseudo-random deviates from a uniform distribution.

$$g(x) = U(0,1) \tag{25}$$

We can generate x accepted values if is satisfied the condition:

$$U \leq \frac{f(x)}{Mg(x)} \tag{26}$$

```
N=100000
m = max(dbeta(x,3,3))
x = runif(N,0,1)
numb = c()
val= c()

for(i in 1:length(x)){
  U = runif(n = 1, min =  0, max = 1)
  if(dunif(x[i], 0, 1)*m*U <= dbeta(x[i], 3, 3)) {
    val[i]=x[i]
    #numb[i] = 1
  }
  else if(dunif(x[i],0,1)*m*U > dbeta(x[i], 3, 3)) {
    #numb[i] = 0
  }
}

val2=val[!is.na(val)]
numb=length(val2)
acc=abs((numb / N)-(1/m))
```

Result of acc:

```
  [1] 0.0007466191
```

We can see from the small result of acc how the frequency is very close to $\frac{1}{k}$ so we can accept $\frac{1}{k}$ as the acceptance probability

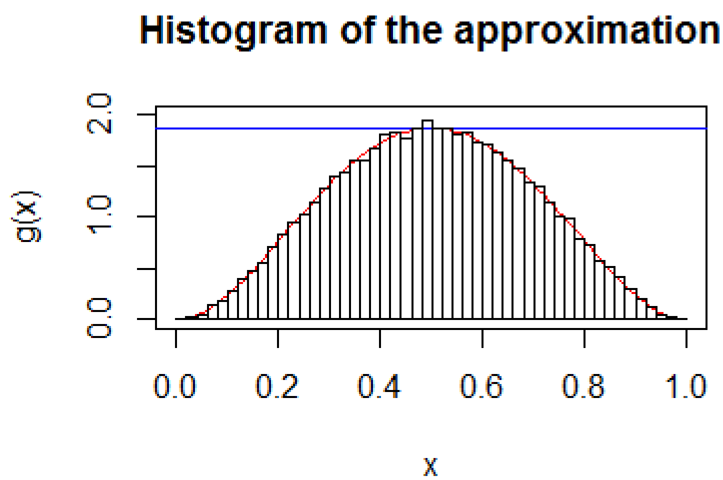

Figure 5: Histogram of exercise 7

8 Acceptance Rejection

8.1 Analytic expression of A-R

The algorithm of Acceptance Rejection is:

$$Y \rightarrow \begin{cases} U^A & U \leq \frac{f_x(y)}{kg(y)} \\ U^R & U > \frac{f_x(y)}{kg(y)} \end{cases}$$

The acceptance probability is:

$$Pr(Y = Y^A) = \int_{-\infty}^{+\infty} Pr(Y = Y^A | Y = y) g_y(y)(y)\, dy \tag{28}$$

where:

$$P(Y = Y^A | Y = y) = P(U \leq \frac{f_x(y)}{kg(y)} | Y = y) = P(U \leq \frac{f_x(y)}{kg(y)}) = \int_0^{\frac{f_x(y)}{kg(y)}} f_u(u)\, du = \frac{f_x(y)}{kg(y)} \tag{29}$$

So:

$$P(Y = Y^A) = \int_{-\infty}^{+\infty} \frac{f_x(y)}{kg(y)} g(y)\, dy = \frac{1}{kc} \tag{30}$$

8.2 Random draw from Bayesian Inference

We begin from the posterior distribution:

$$\pi(\theta|x) = \frac{\pi(x|\theta)\pi(\theta)}{\int_\Theta f(x|\theta)\pi(\theta)\,dx} = \frac{\pi(x|\theta)\pi(\theta)}{m(x)} \tag{31}$$

where m is the normalizing constant. Calling:

$$f(x) = \pi(\theta|x) \tag{32}$$

$$g(x) = \pi(\theta) \tag{33}$$

With the approximation:

$$\pi(\theta|x) \propto \pi(x|\theta)\pi(\theta) \tag{34}$$

We write:

$$\frac{f(x)}{q(x)} \leq k \tag{35}$$

As:

$$\pi(x|\theta) = L(\theta) \leq k \tag{36}$$

$$k = max(\frac{f(\theta)}{g(\theta)}) = L(\theta_{MLE}) \tag{37}$$

The acceptance rejection will be so:

$$ACC = \frac{1}{L(\theta_{MLE})m(x)} \tag{38}$$

8.3 Analytical difficulties with a simple conjugate model

The analytical difficulties can show up in the moment we are dealing with multiplicative constant m(x)=c in the bayesian formula that can't be ignored, in general, with the approximation of the terms.

8.4 Implementing A-R with a conjugate model

We start using the bernoulli distribution (already met in the 1 exercise):

$$Ber(n,\theta) = \binom{n}{x}\theta^x(1-\theta)^{1-x} \tag{39}$$

With likelihood:

$$L_x(\theta) \propto \prod_{i=1}^{n}\left[\binom{n}{x_i}\theta^x(1-\theta)^{1-x}\right] \propto \theta^{\sum_{k=1}^{n} x_i}(1-\theta)^{n-\sum_{k=1}^{n} x_i} \tag{40}$$

Finding the parameter k with θ_{MLE} :

```
LL=function(theta){
  lik=theta^(sum(x))*(1-theta)^(n-sum(x))
  return(lik)
}

LLV=Vectorize(LL)
N=200
n=10
x=rbinom(n=10,size=1,prob=0.3)
thetac=sum(x)/length(x)
k=LLV(thetac)
curve(LLV(x),from=0,to=1)
abline(h=k,col="red")
```

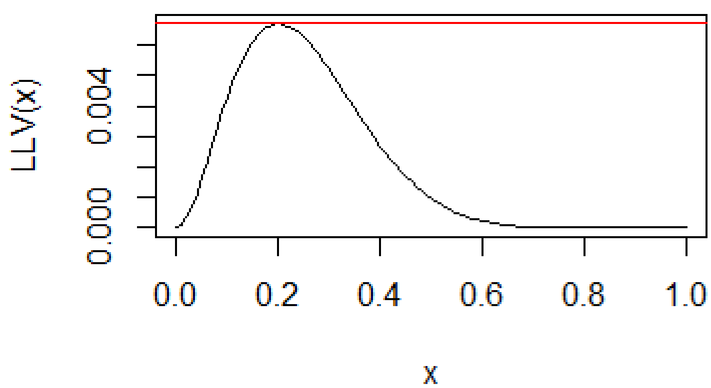

Figure 6: Plot of the likelihood of exercise 8.4

We want to compute the a-r algorithm with the new choises of the candidate distribution q equal to the prior $\pi(\theta)$ having $f(\theta)$ as target distrubution a posteriori computed with the approximation formula of $f(\theta|x) \propto \pi(\theta)L_x(\theta)$

```
pigreco=function(theta){
  dbeta(theta,0.5,0.5)
}
curve(pigreco(x),0,1)
curve(k*pigreco(x),from=0,to=1,col="red",ylim=c(0,0.025))
```

```
curve(LLV(x)*pigreco(x),0,1,col="black",add=TRUE)
```

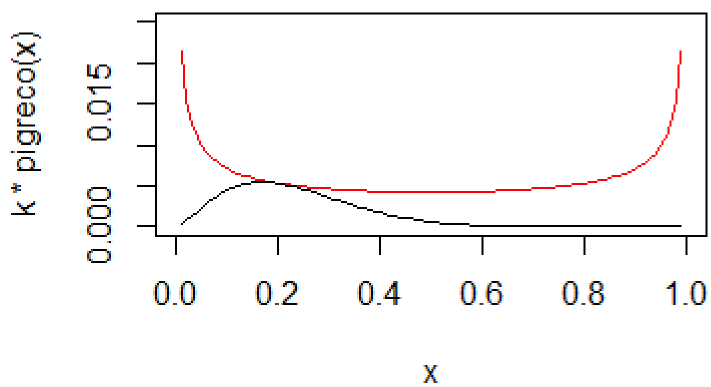

Figure 7: Plot of the prior and posterior distr. of exercise 8.4

We use as prior $\pi(theta) \sim Beta(\alpha = 0.5, \beta = 0.5)$ and $L(\theta) \sim \theta^s(1-\theta)^{n-s}$ with $s = sum(x)$. The posterior will be $\pi(\theta|x) \sim Beta(\alpha + s_1, \beta + n - s)$. We are going to compute the algorithm defined as $J \leq \frac{\pi(\theta)L(\theta)}{k\pi(\theta)}$

```
proposal=rbeta(1000,shape1=0.5,shape2=0.5)
rug(proposal,col="red")
rapporto=LLV(proposal)*(pigreco(proposal))/(k*pigreco(proposal))
J=runif(length(proposal))
accepted_or_not=(J<=rapporto)
```

We now write the constant m(x) as:

$$m(x) = \int_{\Theta} L(\theta)\pi(\theta)\, d\theta = \int_0^1 \frac{(\theta^s(1-\theta)^{n-s})(\theta^{0.5-1}(1-\theta)^{0.5-1})}{B(0.5, 0.5)}\, d\theta \tag{41}$$

Where:

$$B(\alpha, \beta) = \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}\, d\theta = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{42}$$

```
rug(proposal[accepted_or_not],col="black",0.02)
simulati_da_AccRej=proposal[accepted_or_not]
hist(simulati_da_AccRej,freq=FALSE)
s1=(0.5+sum(x))
s2=(0.5-sum(x)+length(x))
curve(dbeta(x,shape1=s1,shape2=s2),add=TRUE)
mx=beta(sum(x)+0.5,length(x)-sum(x)+0.5)/beta(0.5,0.5)
curve(LLV(x)*pigreco(x)/mx,0,1,col="black",add=TRUE,lwd=3)
mx/k
```

We can see how m/k is close to the number of accepted numbers.

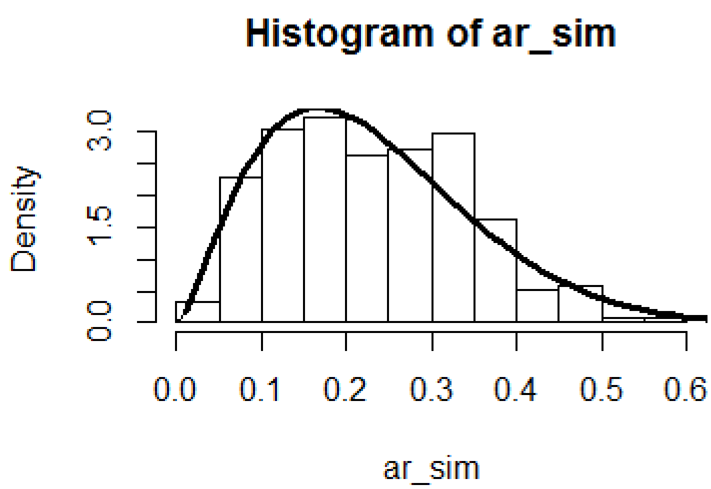

Figure 8: Plot of exercise 8.4

9 Acceptance Rejection with Cauchy distribution

We want to simulate from a standard Normal distribution defined as

$$f(x) = \frac{e^{\frac{-x^2}{2}}}{\sqrt{2\pi}} \sim N(0,1) \tag{43}$$

using pseudo-random deviates from a standard Cauchy

$$g(x) = \frac{1}{\pi(1+x^2)} \tag{44}$$

We find the optimal value M making the relation:

$$M = \frac{f(x)}{g(x)} \tag{45}$$

We generate x values from g(x) and U from Unif(0,1). If

$$U \leq \frac{f(x)}{Mg(x)} \tag{46}$$

we accept the x

Code in R:

```
rel=(dnorm(x,0,1)/dcauchy(x,0,1))
m=max(rel) #it makes sqrt(2pi/e)
N=10000

x = rcauchy(N,0,1)
numb = c()
val= c()

for(i in 1:length(x)){
  U = runif(n = 1, min =  0, max = 1)
  if(dcauchy(x[i], 0, 1)*m*U <= dnorm(x[i], 0, 1)) {
    val[i]=x[i]
  }
  else if(dcauchy(x[i],0,1)*m*U > dnorm(x[i], 0, 1)) {
  }
}

val2=val[!is.na(val)]
numb=length(val2)
acc=abs((numb / N)-(1/m))
```

Looking to the low value of:

$$|\frac{n}{N} - \frac{1}{k}| \tag{47}$$

```
    [1] 0.004955376
```

We can see from the small result how the frequency is very close to $\frac{1}{k}$ so we can accept $\frac{1}{k}$ as the acceptance probability

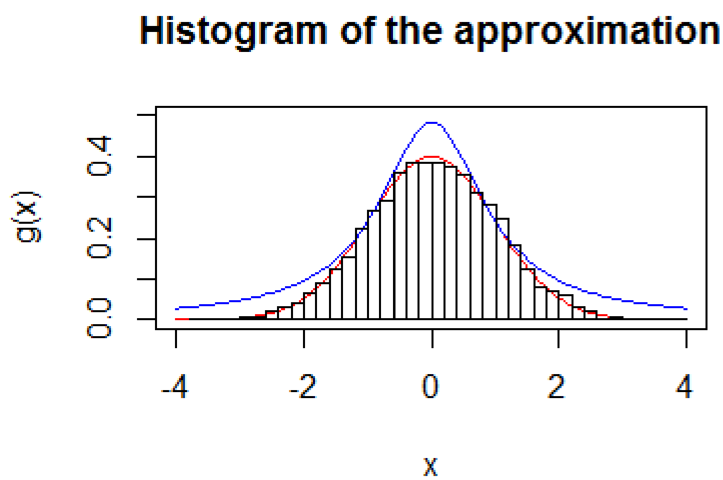

Figure 9: Histogram of exercise 9