Homework 1

Dario Stagnitto

29/4/2018

Ex 1.A

Sample survey: Suppose we are going to sample 100 individuals from a county (of size much larger than 100) and ask each sampled person whether they support policy Z or not. Let $Y_i = 1$ if person i in the sample supports the policy, and $Y_i = 0$ otherwise.

Assume $Y_1, \ldots., Y_{100}$ are, conditional on θ, i.i.d. binary random variables with expectation θ. Write down the joint distribution of $Pr(Y_1 = y_1, \ldots, Y_{100} = y_{100} \mid \theta)$ in a compact form. Also write down the form of $P(\sum_{i=1}^{100} Y_i = y \mid \theta)$

So what we have is:

- $Y_i \mid \theta \sim Ber(\theta)$
- $Y_i \in \{0, 1\}$
- $f(y_i \mid \theta) = \theta^y (1-\theta)^{1-y}$

Being i.i.d. our joint distribution will be:

$$P(Y_1 = y_1, \ldots., Y_{100} = y_{100} \mid \theta) = \prod_{i=1}^{100} f(y_i \mid \theta) = \theta^{\sum_{i=1}^{100} y_i} (1-\theta)^{100-\sum_{i=1}^{100} y_i}$$

Since $P(\sum_{i=1}^{100} Y_i = y \mid \theta)$ is sum of 100 $Ber(\theta)$ conditionally iid, using the Bernoulli's properties it can be written as a $Bin(100, \theta)$. So we have:

$$P(\sum_{i=1}^{100} Y_i = y \mid \theta) = \binom{100}{y} \theta^y (1-\theta)^{100-y}$$

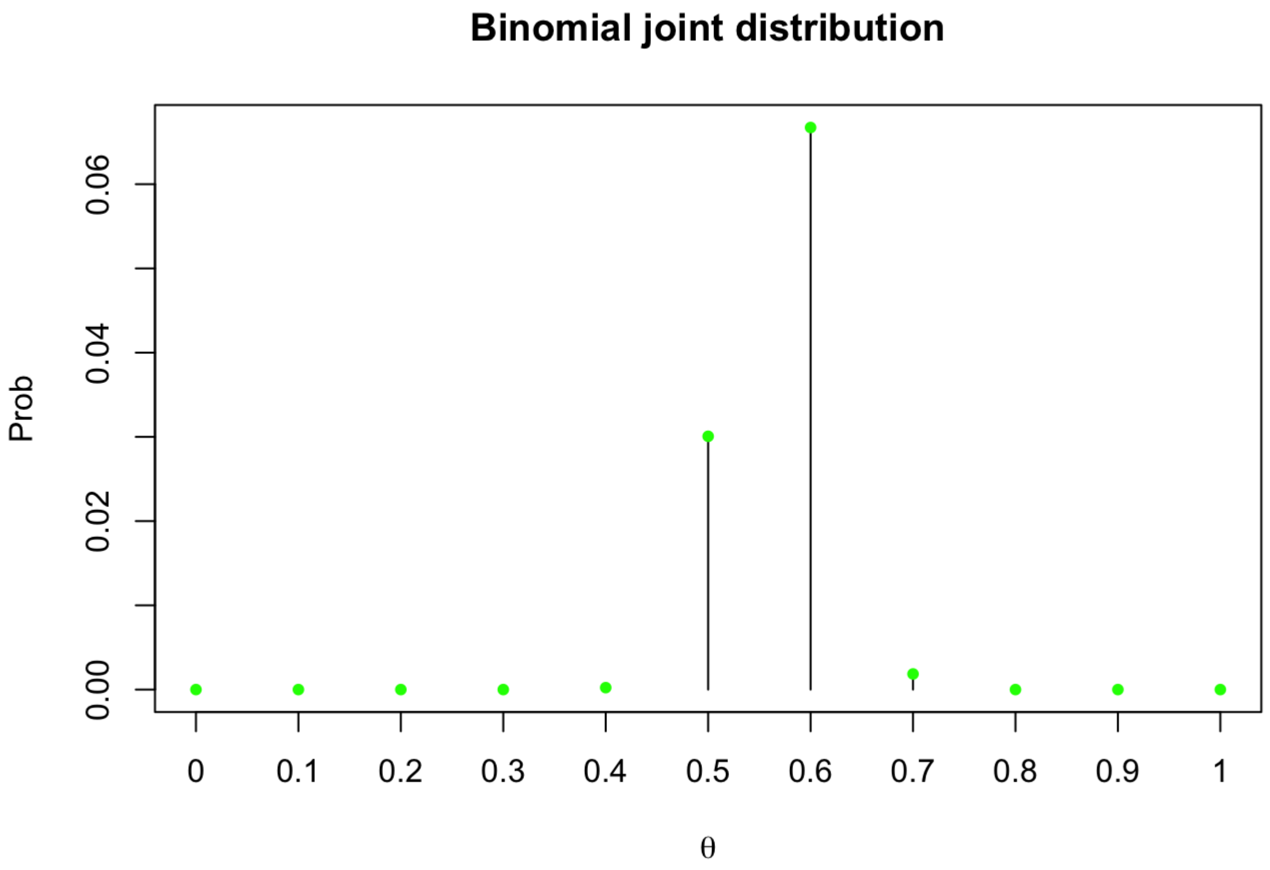
Ex 1.B

For the moment, suppose you believed that $\theta \in \{0.0, 0.1, \ldots, 0.9, 1.0\}$. Given that the results of the survey were $\sum_{i=1}^{n} Y_i = 57$, compute:

$$Pr(\sum_{i=1}^{n} Y_i = 57 \mid \theta)$$

for each of these 11 values of θ and plot these probabilities as a function of θ.

```
seque <- seq(0,1, by = 0.1)

a = NULL
for (i in 1:length(seque)){

  a[i] <- dbinom(x = 57, size = 100, prob = seque[i])
}

plot(a, type = "h", xaxt = "n", ylab = "Prob", xlab = expression(theta), main = "Binomial joint distribution")
axis(1, at = (1:11), labels = seque)
points(a, pch = 20, col = "green")
```

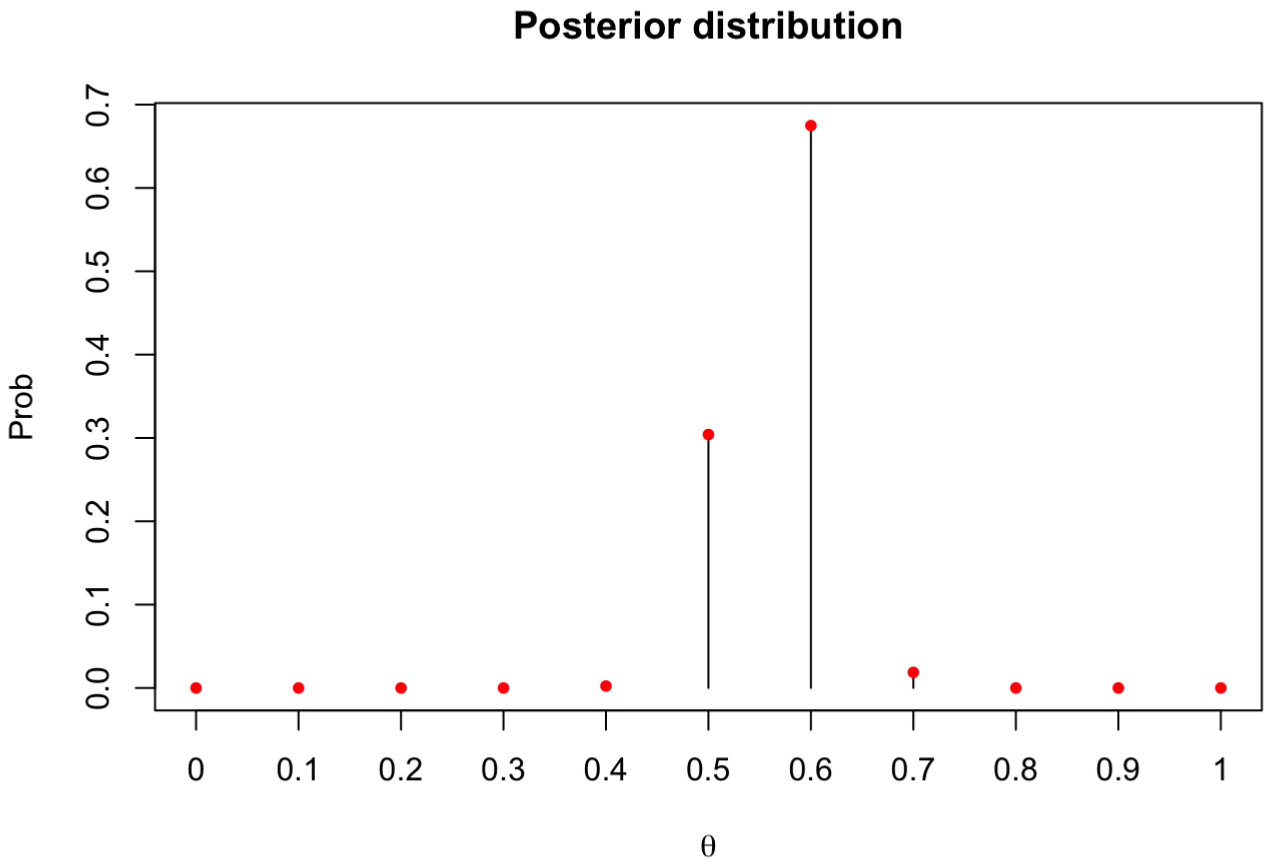


Ex 1.C

Now suppose you originally had no prior information to believe one of these $\theta - values$ **over another,and so** $Pr(\theta = 0.0) = Pr(\theta = 0.1) = ... = Pr(\theta = 0.9) = Pr(\theta = 1.0)$**. Use Bayes rule to compute** $\pi(\theta \mid \sum_{i=1}^{n} Y_i = 57)$ **for each θ-value. Make a plot of this posterior distribution as a function of θ.**

$$\pi(\theta \mid \sum_{i=1}^{n} Y_i = 57) = \frac{P(\sum_{i=1}^{n} Y_i = 57 \mid \theta) \cdot \pi(\theta)}{\sum_{j=1}^{n} P(\sum_{i=1}^{n} Y_i = 57 \mid \theta_j) \cdot \pi(\theta_j)}$$

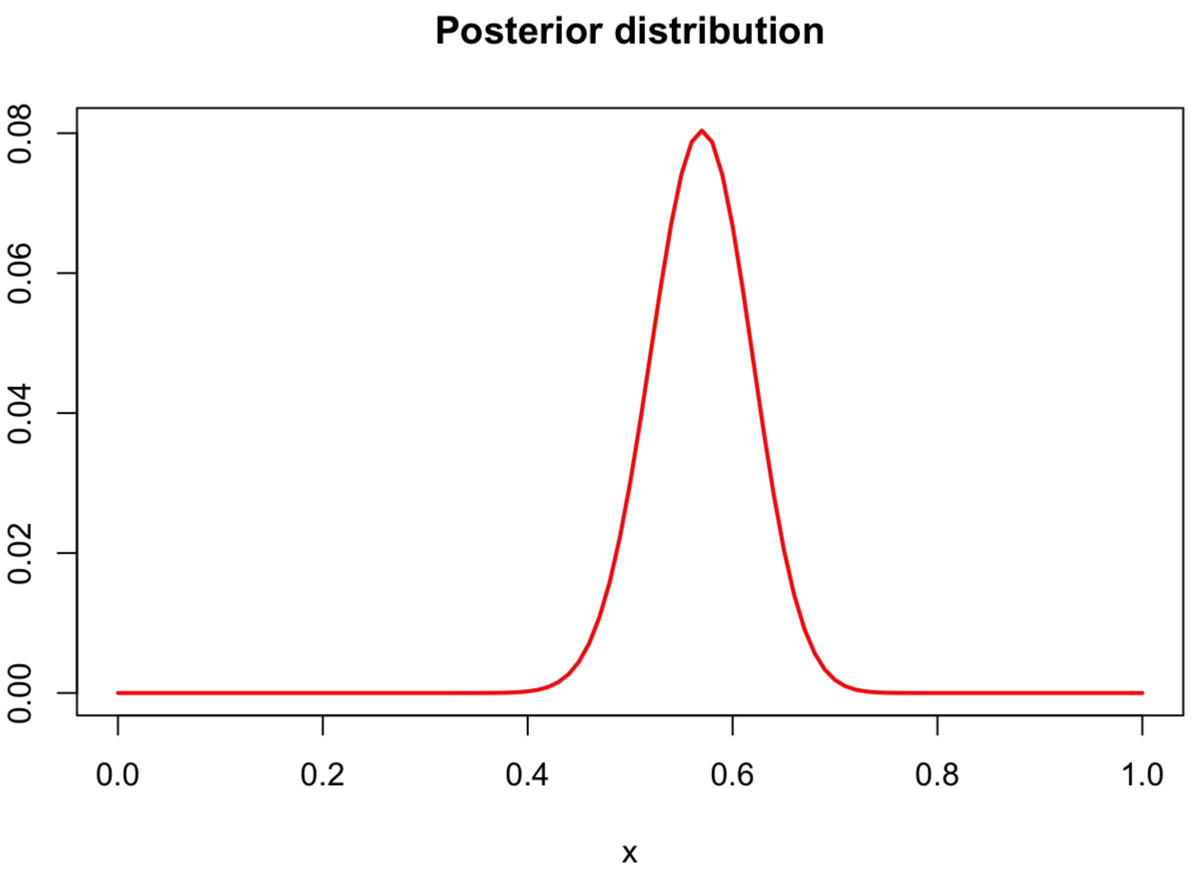
```
numer <- a*1/11
deno <- sum(a*1/11)

posterior <- numer/deno

plot(posterior, type = "h", xaxt = "n", ylab = "Prob", xlab = expression(theta), main = "Posterior distribution")
axis(1, at = (1:11), labels = seque)
points(posterior, pch = 20, col = "red")
```



Ex 1.D

Now suppose you allow θ to be any value in the interval $\theta = [0, 1]$. Using the uniform prior density for $\theta \in [0, 1]$, so that $\pi(\theta) = I[0, 1](\theta)$, plot $\pi(\theta) \cdot Pr(\sum_{i=1}^{n} Yi = 57 \mid \theta)$ as a function of θ.

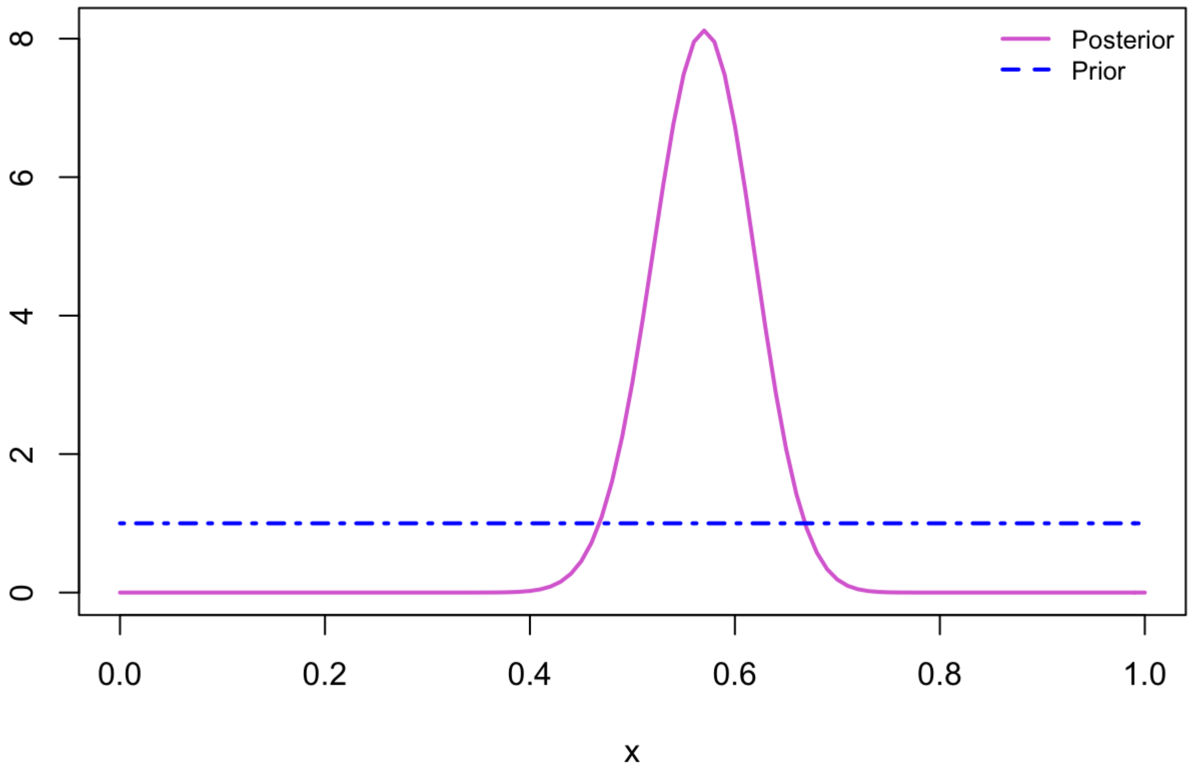
```r
prior <- function(theta){ dunif(theta,0,1) }
posterior <- function(theta){prior(theta)*dbinom(x = 57, size = 100, prob = theta)}
curve(posterior(x), col = "red", lwd = 2, main = "Posterior distribution", xlab = "x", ylab = "")
```



Ex 1.E

Plot the posterior density as a function of θ.

```
curve(dbeta(x,58,44), col = "orchid", lwd = 2, ylab = "")
curve(dbeta(x,1,1),col = "blue", add = T, lwd = 2, lty = 4)
legend("topright", legend = c("Posterior","Prior"), col = c("orchid","blue"),lty = 1:4,cex = 0.8, lwd = 2, bty =
"n" )
```



Ex 2.A

Derive the general formula of the prior predictive distribution for a single observation X.

$$X_i \mid \theta \sim N(\theta, \sigma_x^2)$$

$$\theta \sim N(\theta_0, \sigma_\theta^2)$$

$$P(X') = \int_{-\infty}^{\infty} P(X', \theta)\mathrm{d}\theta = \int_{-\infty}^{\infty} P(X' \mid \theta) \cdot P(\theta)\mathrm{d}\theta$$

where $P(X' \mid \theta)$ is the Likelihood and $P(\theta)$ is the prior distribution.

So we have that:

$$X' = (X' - \theta) + \theta$$

Knowing θ:

$$(X' - \theta) \sim N(0, \sigma_x^2)$$

$$X' \sim N(0, \sigma_x^2) + N(\theta_0, \sigma_\theta^2) = N(\theta_0, \sigma_x^2 + \sigma_\theta^2)$$

Ex 2.B

Derive the general formula of the posterior predictive distribution for a single observation X.

$$Pr(X' \mid X) = \int_{-\infty}^{+\infty} Pr(X' \mid \theta, X) \cdot \pi(\theta \mid X) d\theta = Pr(X' \mid \theta) \cdot \pi(\theta \mid X) \sim N(\cdot) \cdot N(\theta', \sigma_\theta'^2)$$

$$X' = (X' - \theta) + \theta$$

$$(X' - \theta) \sim N(0, \sigma_x^2)$$

$$\theta \sim N(\theta', \sigma_\theta'^2)$$

$$X' \sim N(0, \sigma_x^2) + N(\theta', \sigma_\theta'^2) = N(\theta', \sigma_x^2 + \sigma_\theta'^2)$$

Ex 2.C

Elicit your prior distribution on the unknown θ in such a way that your prior mean is 0 and you believe that the unknown θ is in the interval [–5, 5] with prior probability 0.96.

```
griglia <- seq(2,3, by = 0.001)

quant.fun <- function(sd){qnorm((1-0.96)/2,0,sd)+5}

p <- quant.fun(griglia)
which.min(abs(quant.fun(griglia)-0)) #cerchiamo quelli più vicini a 0
```

```
## [1] 436
```

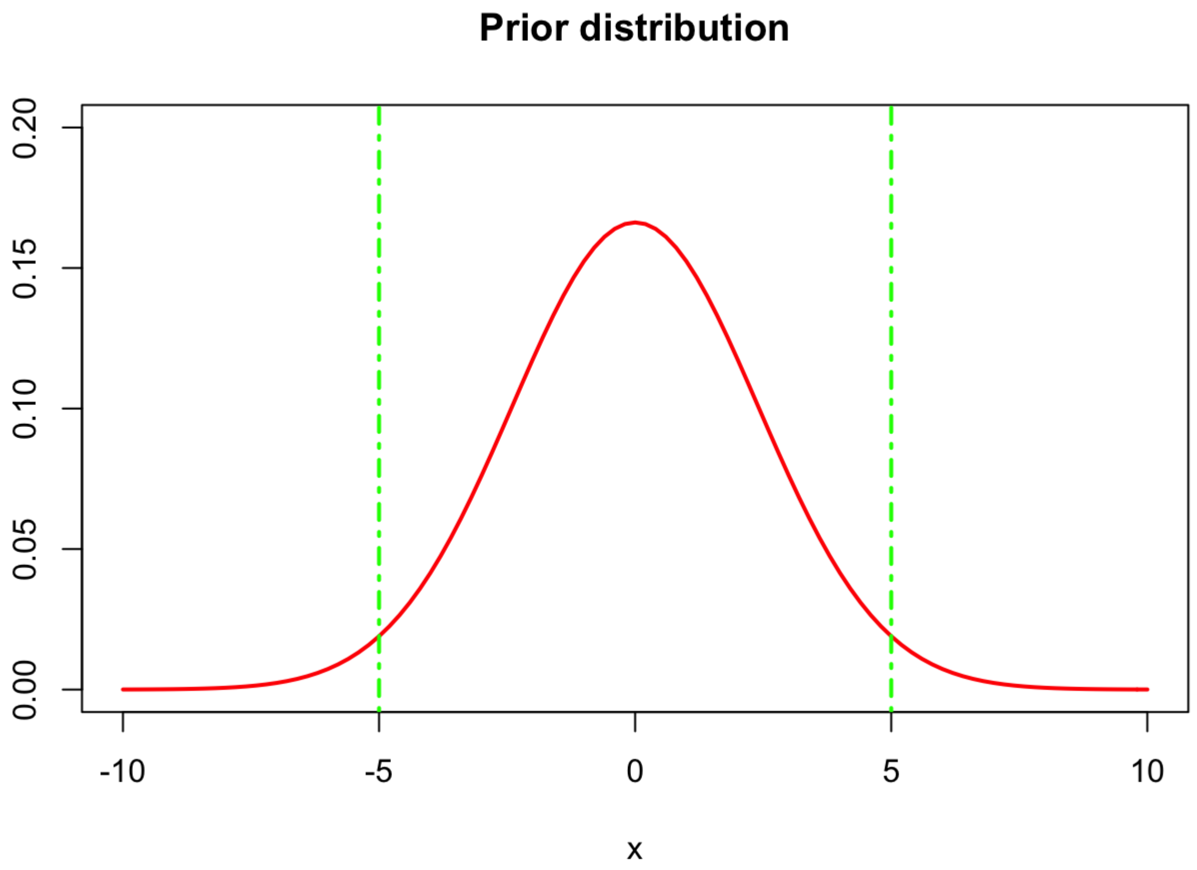
```
p[436]
```

```
## [1] -0.0008785974
```

```
# 436*0.001 + 2 = 2.436

qnorm((1-0.96)/2,0,2.4)
```

```
## [1] -4.928997
```

```
curve(dnorm(x,0,2.4), xlim = c(-10,10), col = "red", ylim = c(0,.2), ylab = "",
      main = "Prior distribution", lwd = 2) #prior distribution
abline(v = c(-5,5), col = "green", lwd = 2, lty = 4)
```

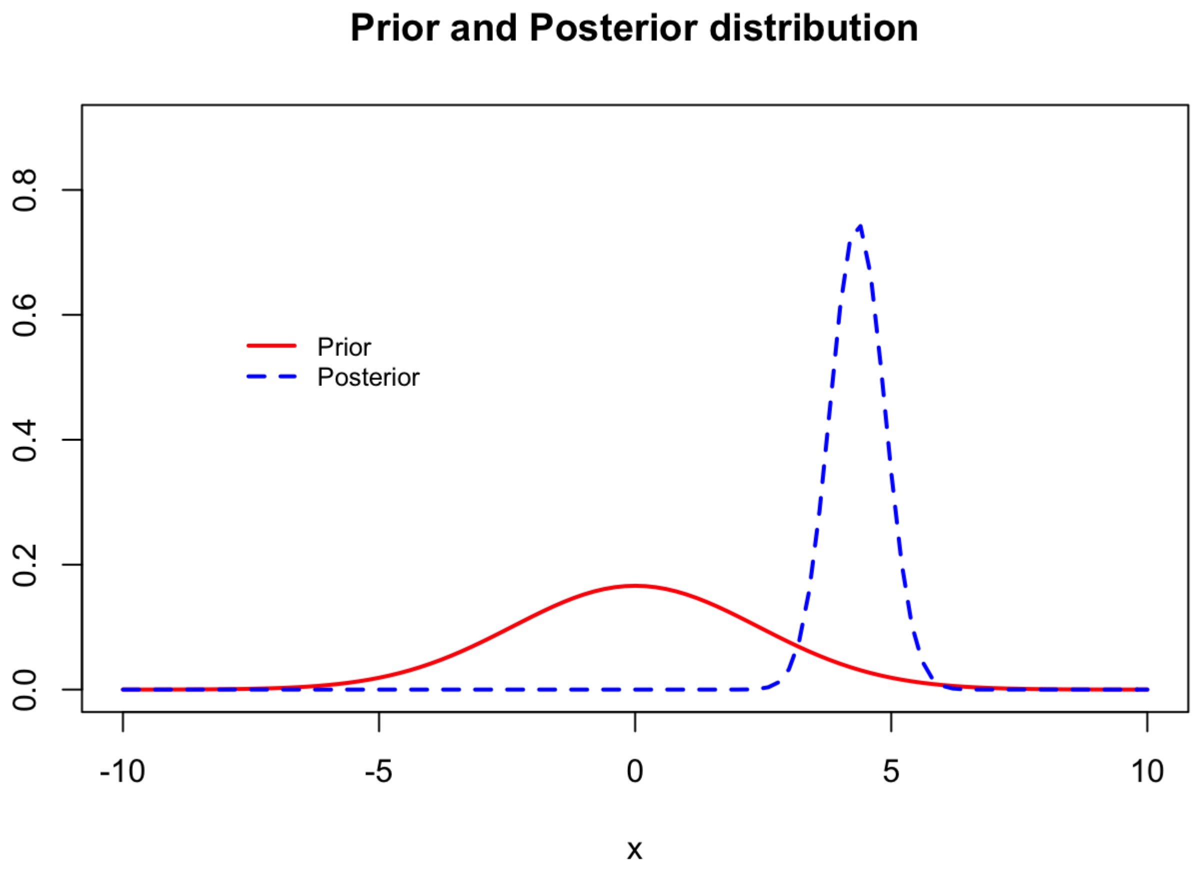


Ex 2.D

Assume that the known value of λ is 1/3 and suppose you have observed the following data, derive your posterior distribution and represent it graphically.

$$-1.25, 8.77, 1.18, 10.66, 11.81, -6.09, 3.56, 10.85, 4.03, 2.13$$

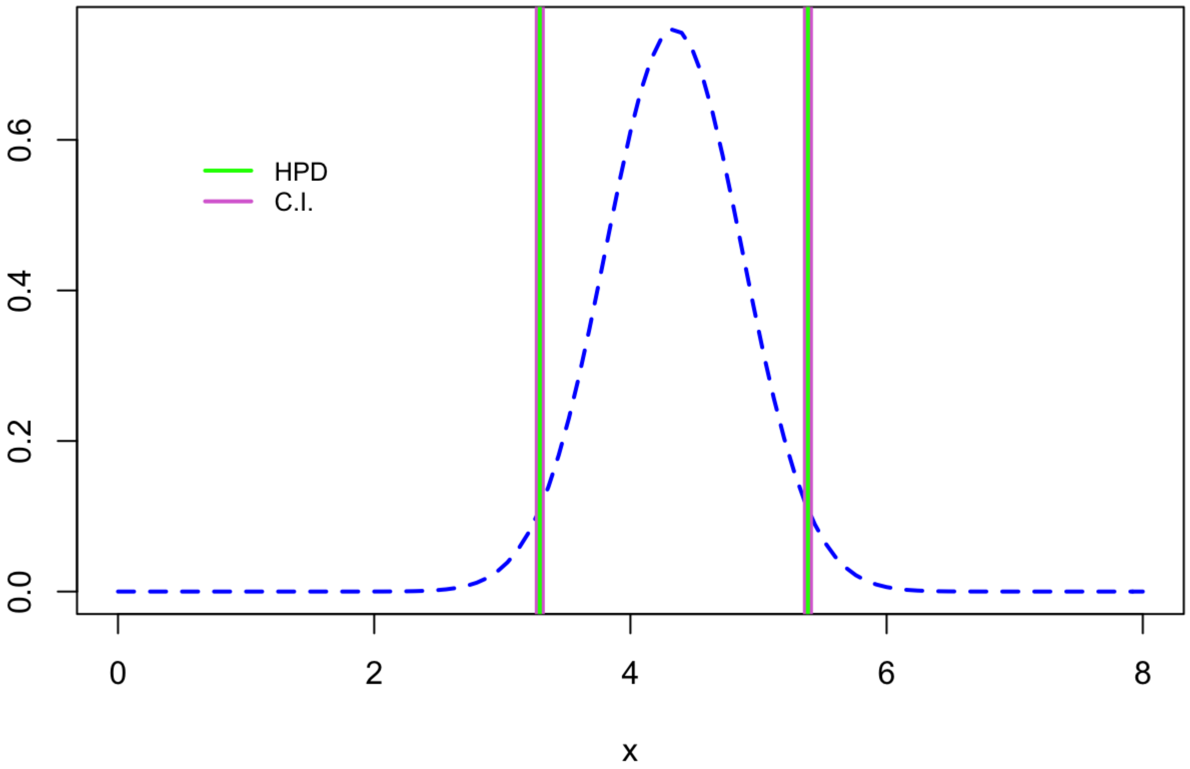
```
vec <- c(-1.25,8.77,1.18,10.66,11.81,-6.09,3.56,10.85,4.03,2.13)

sigma_2 <- 3
y_bar <- mean(vec)

n <- length(vec)

tau_0 <- 2.4**2
mu_0 <- 0



mu_n <- (1/tau_0*mu_0)/((1/tau_0)+n*(1/sigma_2)) + ((n*(1/sigma_2)*y_bar)/((1/tau_0)+(n*(1/sigma_2))))
tau_n <- 1/((1/tau_0)+n/(sigma_2))

post <- function(theta){ dnorm(theta, mean = mu_n, sd = sqrt(tau_n))}



curve(dnorm(x,0,2.4), xlim = c(-10,10), col = "red", ylim = c(0,0.9),main = "Prior and Posterior distribution", ylab = "", lwd = 2)
curve(post(x), add = T, lwd = 2, col = "blue", lty = 2:4) #posterior
legend(x = -8,y = 0.6, legend = c("Prior","Posterior"), col = c("red","blue"),lty = 1:4,cex = 0.8, lwd = 2, bty = "n")
```



Ex 2.E

Derive your favorite point estimate and interval estimate and motivate your choices.

```
library(TeachingDemos)
C.I <- qnorm(c(0.025,0.975), mean = mu_n, sd = sqrt(tau_n))
hpd <- hpd(qnorm, mean = mu_n,sd = sqrt(tau_n), conf = 0.95)

curve(post(x), lwd = 2, col = "blue", lty = 2:4, xlim = c(0,8), ylab = "")
legend(x = 0.5,y = 0.6, legend = c("HPD","C.I."), col = c("green","orchid"), cex = 0.8, lwd = 2, bty = "n")

abline(v = C.I, col = "orchid", lwd = 5)
abline(v = hpd, col = "green", lwd = 2)
```



```
diff(hpd)
```

```
## [1] 2.093214
```

```
C.I[2] - C.I[1]
```

```
## [1] 2.093214
```

As point estimate, the best choice is taking the mean of the posterior distribution that is equal to **4.34.**

In this case, the way how we choose the confidence interval is not so relevant because for the Gaussian distribution, being symmetric, both coincide.

Ex 3.A

Consider the following statistical model where $X_i \mid \theta$ are i.i.d with

$$X_i \mid \theta \sim f(x \mid \theta) = \frac{1}{20} I_{[\theta-10,\theta+10](x)}$$

Provide a fully Bayesian analysis for these data explaining all the basic ingredients and steps for carrying it out. In particular, compare your final inference on the unknown $\theta = E[X \mid \theta]$ with the one you have derived in the previous point 2).

$$X_i \mid \theta \sim Unif(\theta - 10, \theta + 10)$$

$$\pi(\theta) \sim N(0, 2.4^2)$$

$$L(\theta) = \prod_{i=1}^{N} \frac{1}{20} I_{[\theta-10,\theta+10](x_i)} = \frac{1}{20^n} I_{[x_n-10,x_1+10]}(\theta)$$

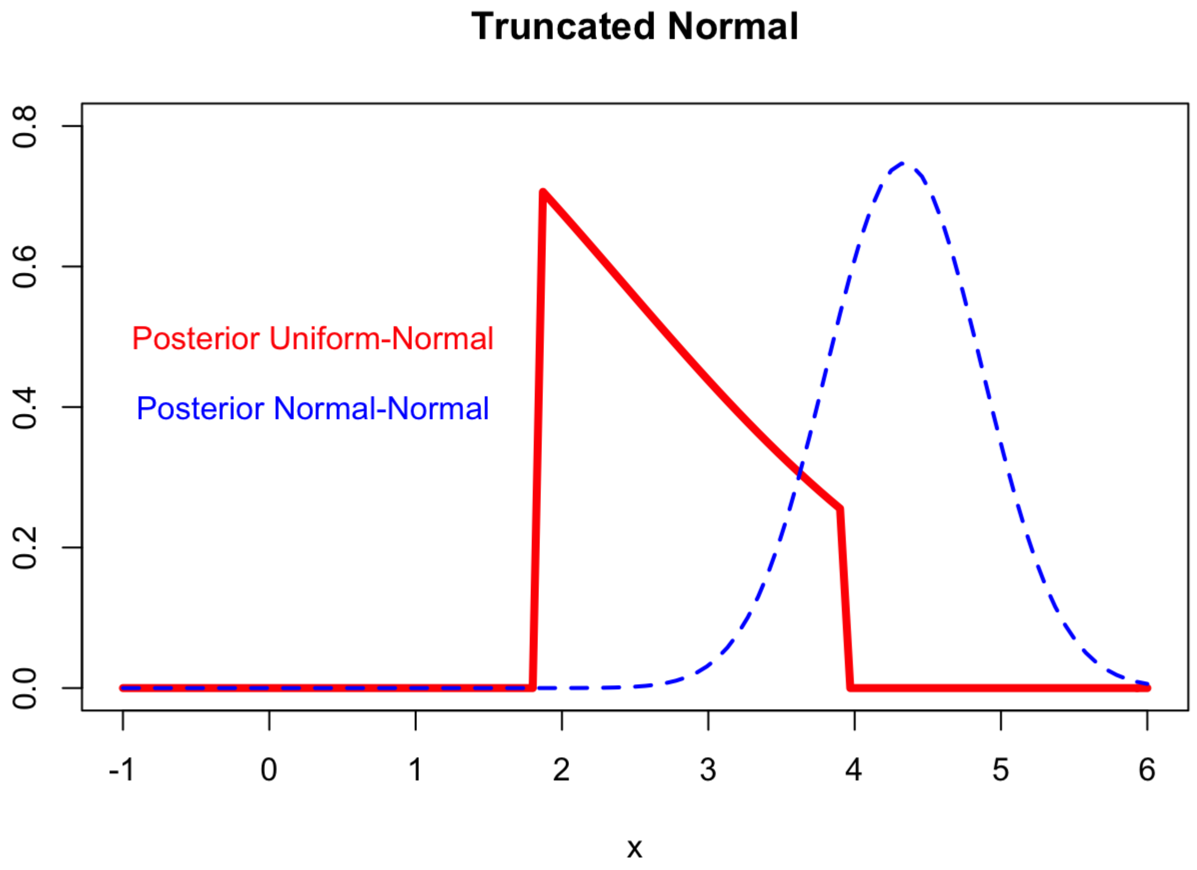
Finally, our posterior will be:

$$\pi(\theta \mid X) = \pi(\theta) L(\theta)$$

```
library(truncnorm)
```

```
## Warning: package 'truncnorm' was built under R version 3.4.3
```

```
likeli <- function(theta){ (1/(20^n)) * (theta<(min(vec)+10))*(theta>(max(vec)-10))}

post.3 <- function(theta){dtruncnorm(theta,mean = 0,sd = 2.4, a = 1.81, b = 3.91)}

curve(post.3(x), xlim = c(-1,6),ylim = c(0,0.8), ylab = "", col = "red", lwd = 4, main = "Truncated Normal")
curve(post(x), add = T, lwd = 2, col = "blue", lty = 2:4)
text(0.3, 0.5, col = 'red', 'Posterior Uniform-Normal', lwd = 1.5)
text(0.3, 0.4, col = 'blue', 'Posterior Normal-Normal', lwd = 1.5)
```

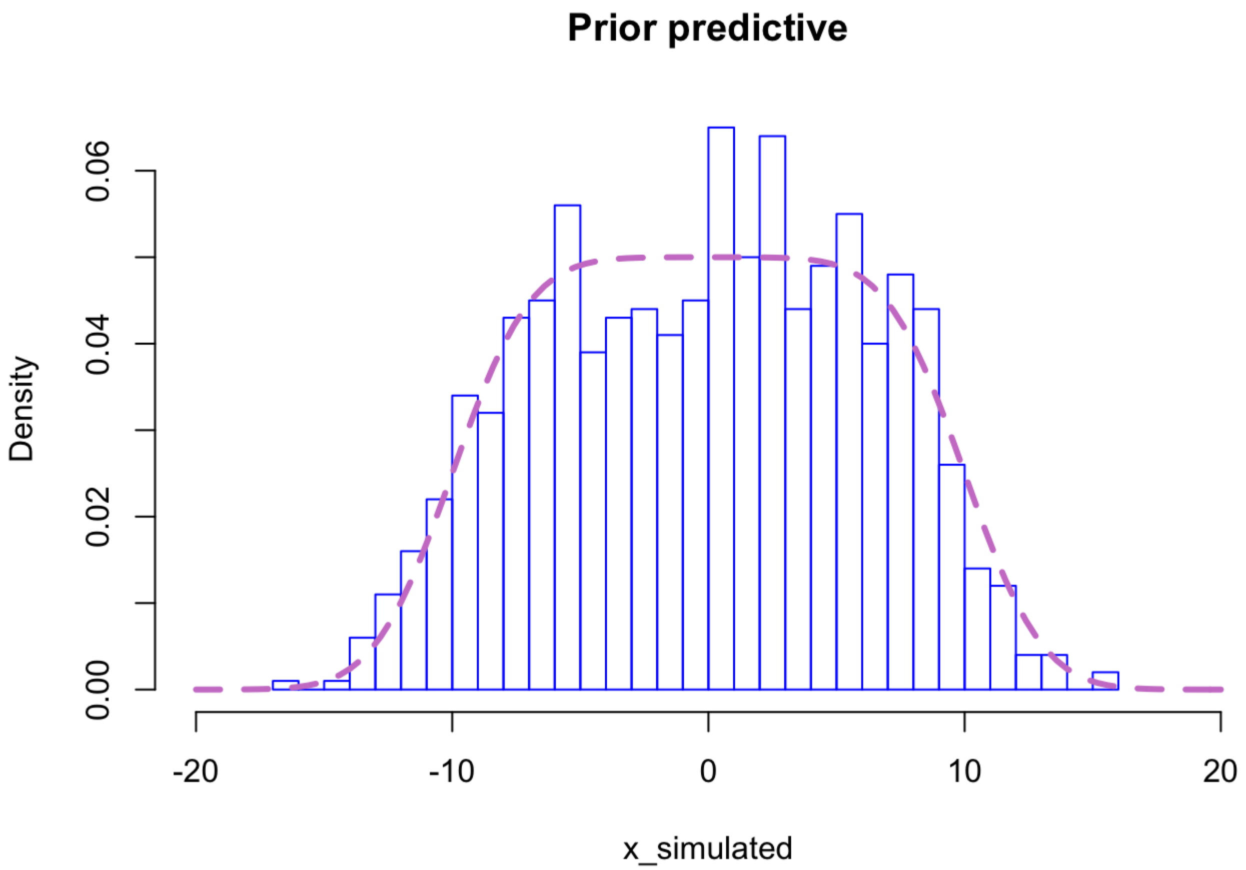


Our posterior distribution turned out to be a **truncated Normal**.

Ex 3.B

Write the formula of the prior predictive distribution of a single observation and explain how you can simulate i.i.d random draws from it. Use the simulated values to represent approximately the predictive density in a plot and compare it with the prior predictive density of a single observation of the previous model.

```
sim <- 1000
sim_theta <- rnorm(sim,mu_0,sqrt(tau_0))
x_simulated <- runif(sim,sim_theta-10,sim_theta+10)


prior.pred <- function(x){ (pnorm(x+10,mu_0,sqrt(tau_0)) - pnorm(x-10,mu_0,sqrt(tau_0)))/20}

hist(x_simulated,freq = F,xlim = c(-20,20),breaks = 40,border = 4,main = 'Prior predictive')
curve(prior.pred,add=T,col=rgb(0.8,0.5,0.8),lwd=3,lty=2)
```



Ex 3.C

Consider the same discrete (finite) grid of values as parameter space θ for the conditional mean Θ in both models. Use this simplified parametric setting to decide whether one should use the Normal model rather than the Uniform model in light of the observed data.

To decide whether it's better use the Normal model rather than the Uniform model, we can compare the two models implemented above in the following way:

$$BF = \frac{Pr(X \mid M_1)}{Pr(X \mid M_2)} = \frac{\int_\Theta \pi(\theta) f_1(X \mid \theta) d\theta}{\int_\Theta \pi(\theta) f_2(X \mid \theta) d\theta}$$

where **BF** is the **Bayesian Factor**.

```
# Function for the uniform-normal model
mod_2 <- function(theta)
{
  aux <- 0
  for (theta in sim_theta)
  {
    aux <- aux + prod((vec>=theta-10)*(vec<=theta+10)/20) * dnorm(theta,0,sqrt(tau_0))
  }
  return((aux))
}



# Function for the normal-normal model
mod_1 <- function(theta)
{
  aux <- 0
  for (theta in sim_theta)
  {
  aux <- aux + dnorm(theta,0,sqrt(tau_0)) * prod(dnorm(vec,theta,sqrt(3)))
  }
  return((aux))
}

# Evaluate the Bayes Factor
BF <- mod_1(sim_theta)/mod_2(sim_theta)
BF
```

```
## [1] 6.494642e-18
```

In according to the following table,

Values	BF Interpretation
< 1	Supports model 2
1-3	Weak support for model 1

Values	BF Interpretation
3-20	Support for model 1
20-150	Strong support for model 1
Over 150	Very strong support for model 1

the model 2 turns out better than the model 1.

Ex 4.A

The **A-R algorithm** is a basic technique used to generate observations from a distribution. Also called **Rejection sampling**, this algorithm is based on the observation that to sample a random variable one can perform a uniformly random sampling and keep the samples in the region under the graph of its density function.

The steps are the following:

- Generate a random variable Y distributed as G
- Generate U (independent from Y)
- If

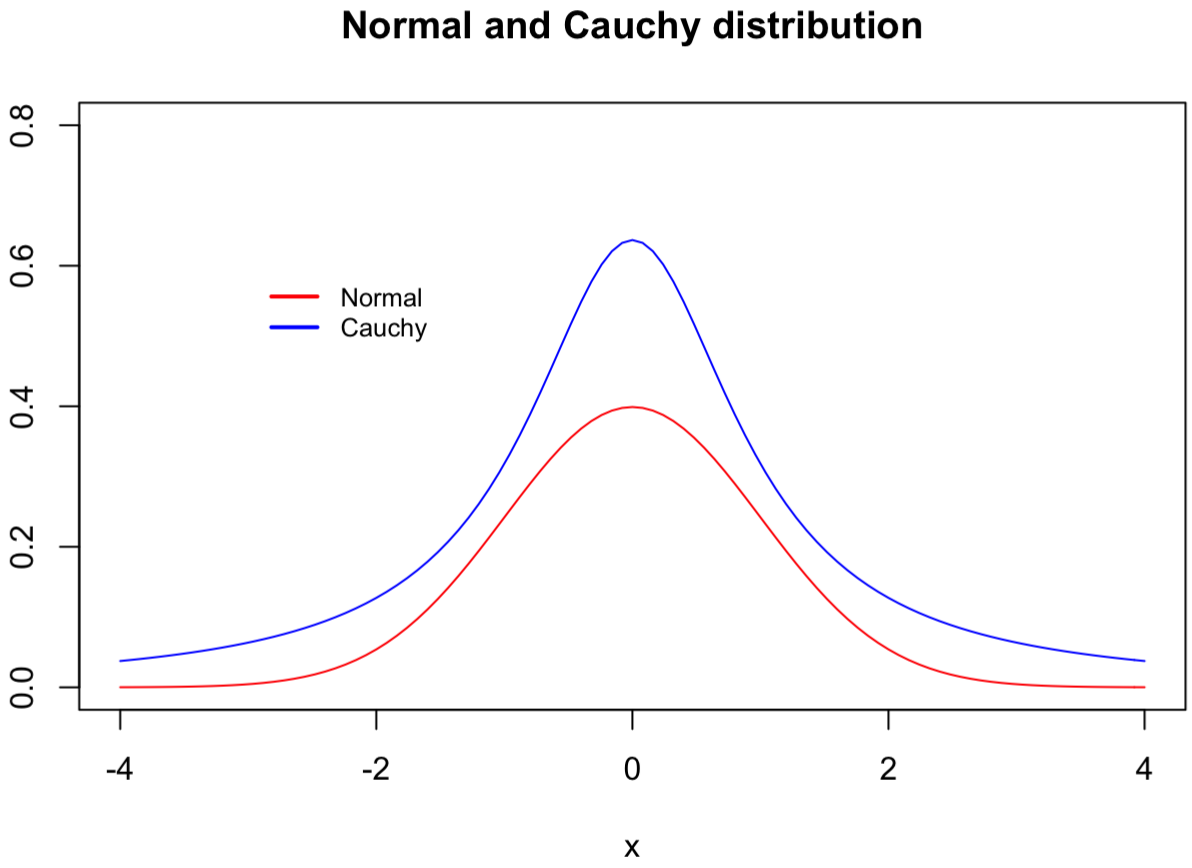
$$U \leq \frac{f(Y)}{g(Y)}$$

then X = Y so we **accept** otherwise **reject** and the algorithm starts again. Taking into account our task, we have that our *f(X)* is a Normal distribution and $g(\cdot)$ our Cauchy distribution.

Ex 4.B

Provide your R code for the implementation of the A-R

```
k <- 2
curve(dcauchy(x)*k, xlim = c(-4,4),col = "blue", ylim = c(0,.8), ylab = "", main = "Normal and Cauchy distribution")
curve(dnorm(x), col = "red", add = T)
legend(x = -3, y = 0.6, legend = c("Normal","Cauchy"), col = c("red","blue"),cex = 0.8, lwd = 2, bty = "n")
```

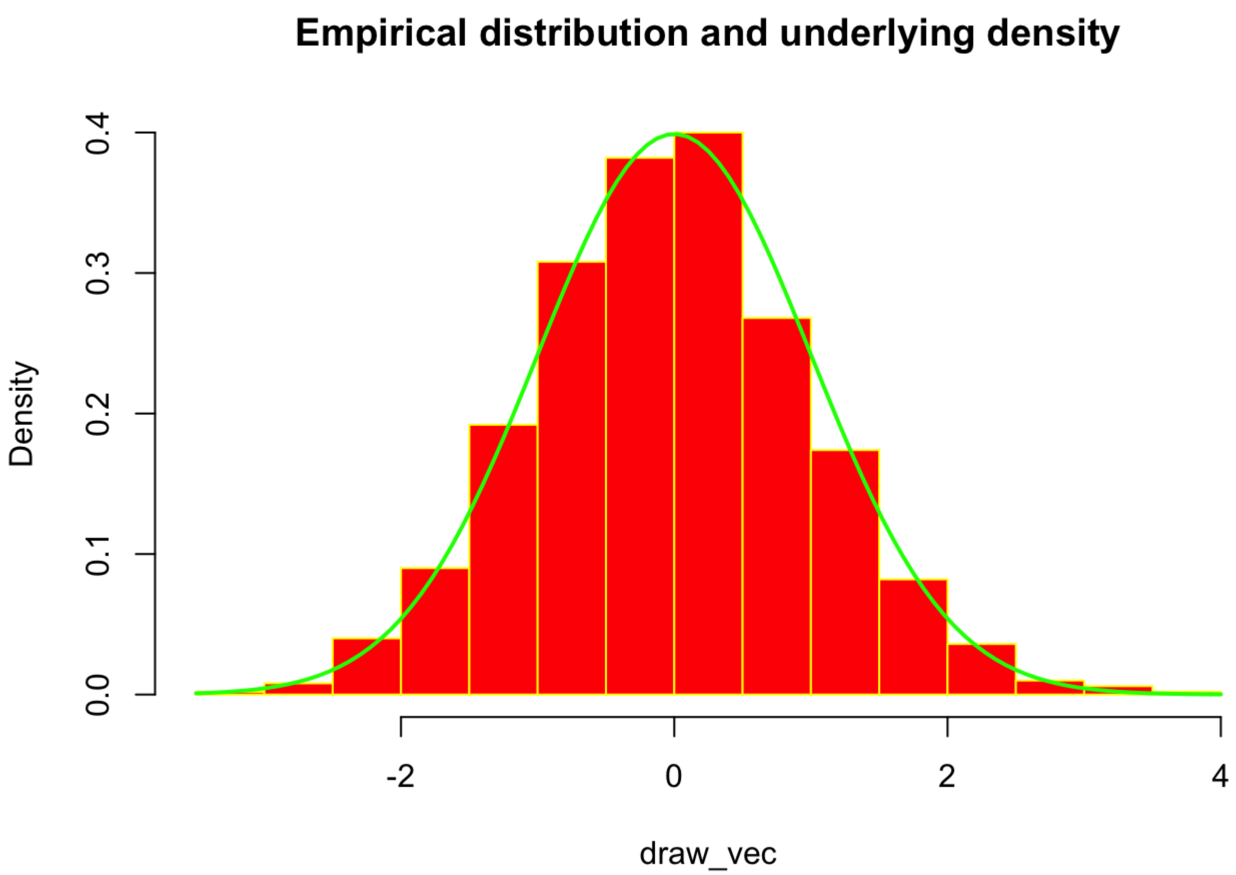


```
q <- function(x){
  dcauchy(x)
}

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] 2.374009
##
## $computational_effort
## [1] 1
```

```r
mcsize=1000
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)
  draw_vec[i] = DD$draw
  effort_vec[i] = DD$computational_effort

}
```

```r
hist(draw_vec,freq=FALSE, col = "red", border = "yellow", main = "Empirical distribution and underlying density")
curve(f(x),add=TRUE, col = "green", lwd = 2)
```



Ex 4.C

Evaluate numerically (approximately by MC) the acceptance probability.

```
accept.prob <- mcsize/sum(effort_vec)
accept.prob
```

```
## [1] 0.5107252
```

Ex 4.D

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

In the code I simulated the A-R algorithm for 1000.

Once implemented that, I estimated the acceptance probability simply dividing the number of iterations for the sum of the “effort_vec” whose data inside are the number of the iterations which assumes 1 as value.

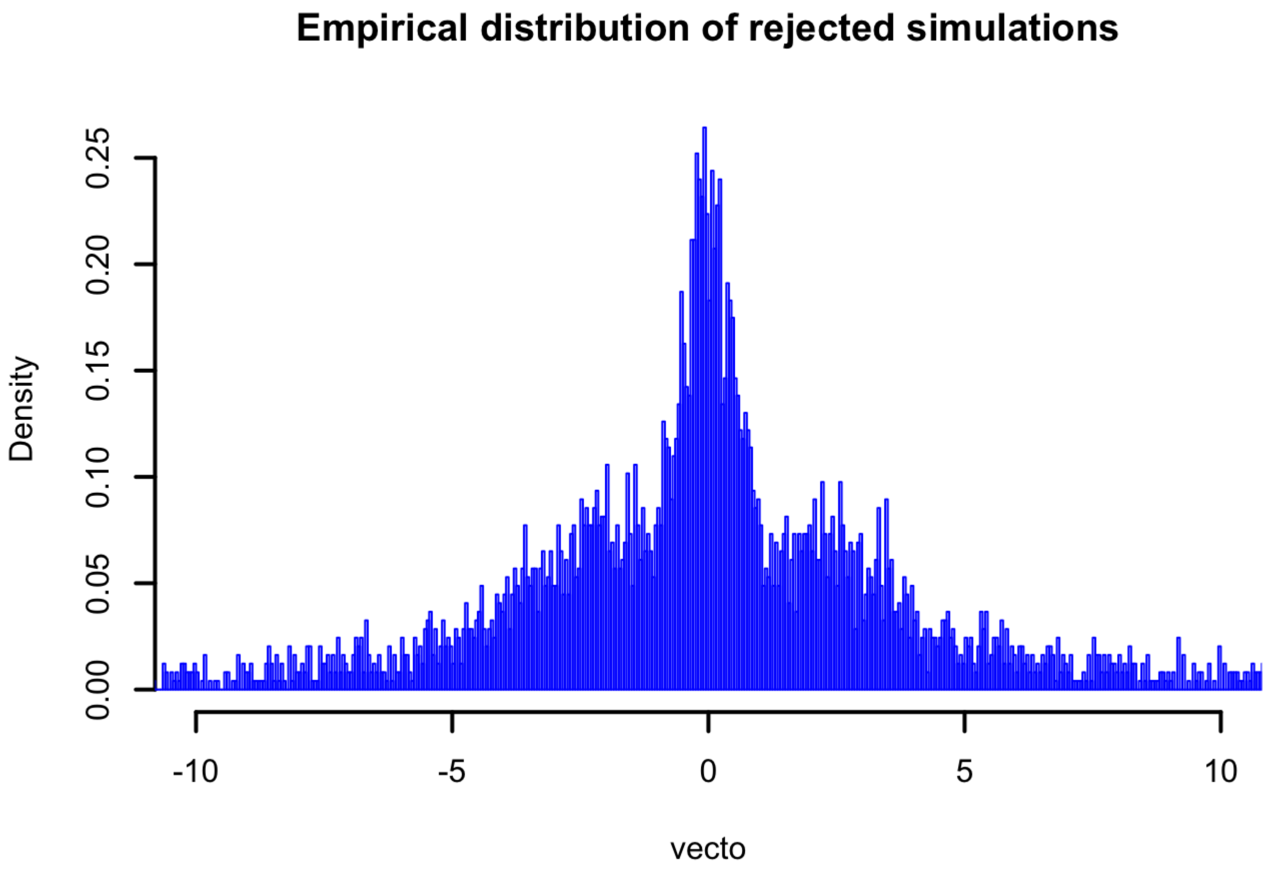
Ex 4.E

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

```
AR.2 <- function(dtarget,dauxiliary,rauxiliary,k){

  count=0
  E=0
  reject <- NULL

  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.2(dtarget=q,dauxiliary=q,rauxiliary=draw_from_q,k)
```

```
## $draw
## [1] -115.0859
##
## $computational_effort
## [1] 2
##
## $rej
## numeric(0)
```

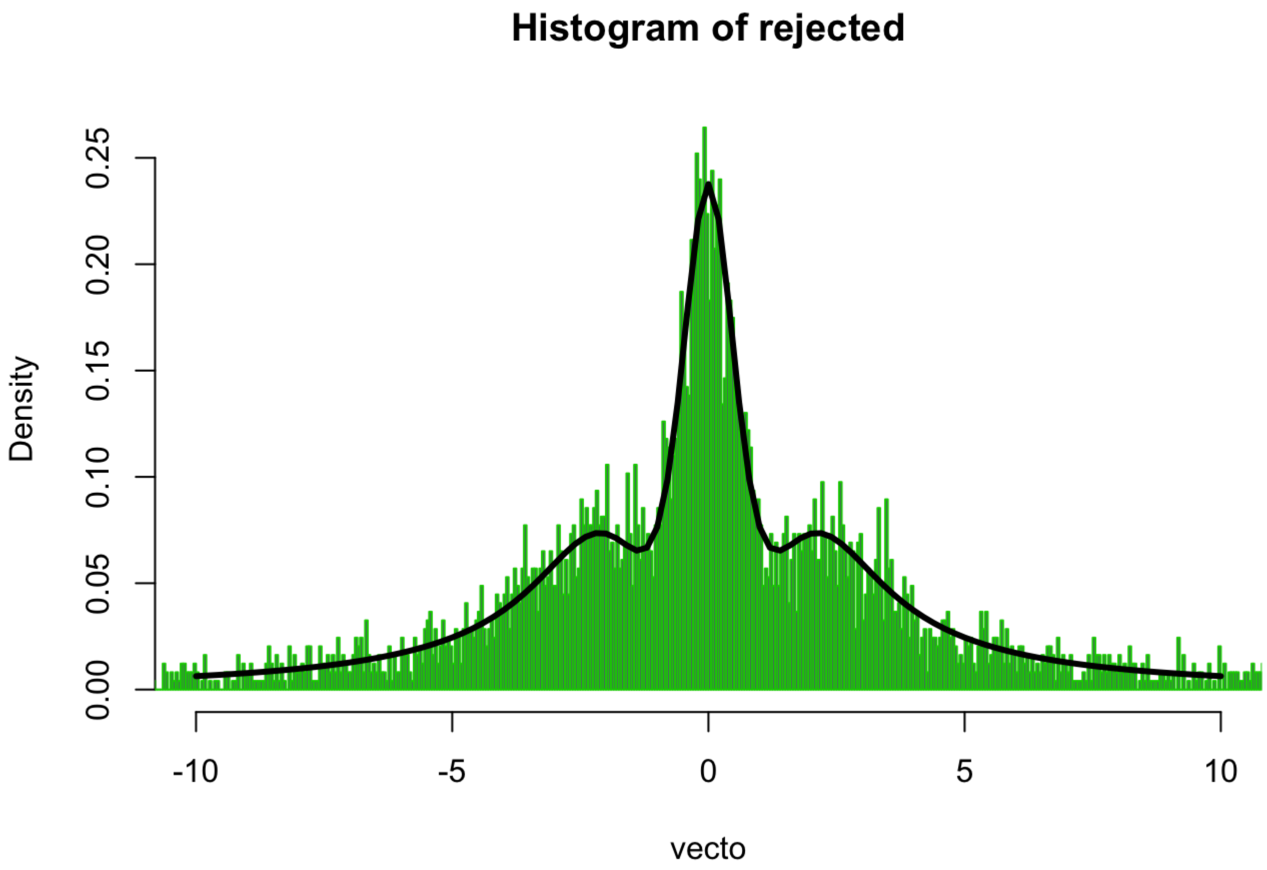
```
mcsize=10000
draw_vec=rep(NA,mcsize)
effort_vec=rep(NA,mcsize)
vecto <- c()

for(i in 1:mcsize){

  DD=AR.2(dtarget=f,dauxiliary=q,rauxiliary=draw_from_q,k = k)
  draw_vec[i] = DD$draw
  effort_vec[i] = DD$computational_effort
  vecto <- c(vecto,DD$rej)

}


hist(vecto, breaks = 100000,xlim = c(-10,10), freq = F, border = "blue", lwd = 2, main = "Empirical distribution
 of rejected simulations")
```



Ex 4.F

Derive the underlying density corresponding to the rejected random variables and try to compare it with the empirical distribution.

```
hist(vecto,breaks = 100000,freq = FALSE,border = 3,xlim=c(-10,10),col = rgb(0.5,0.2,0.7),main='Histogram of rejected')
rejected_distr <- function(x) (dcauchy(x)*(1-(dnorm(x))/(k*dcauchy(x))))/(1-1/k)
curve(rejected_distr(x),add=T,lwd=3)
```



Ex 5.A

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 different ways:

- exact analytic computation

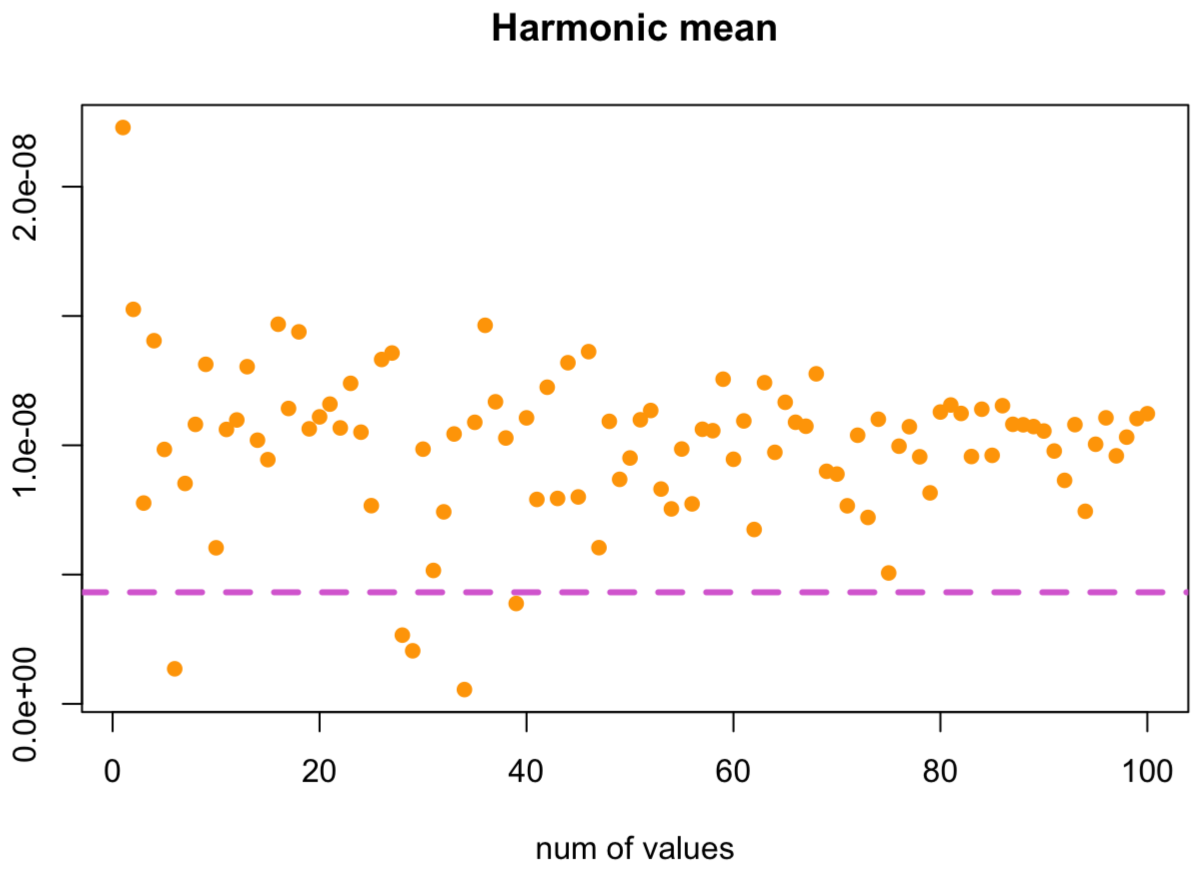
```r
set.seed(123)

n <- 10
M <- 1
if ( M == 1){
  x_obs = rpois(n,2)
}

fun <- function(theta){prod(1/factorial(x_obs))*theta^(24)*exp(1)^(-11*theta)}
integrale <- integrate(fun, lower = 0, upper = Inf)
value <- integrale$value
value
```

```
## [1] 4.314297e-09
```

Ex 5.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.

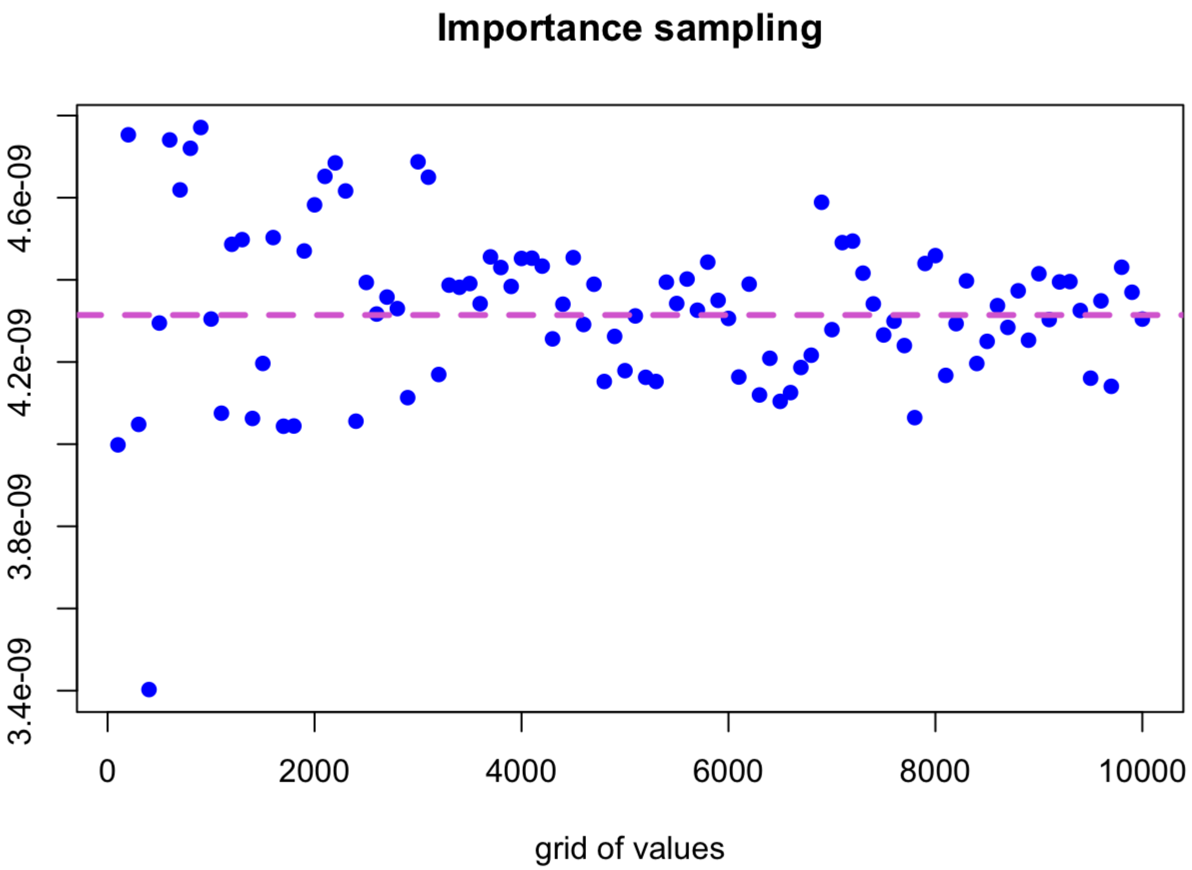
```r
set.seed(123)
obs <- rpois(10,2)
prior <- function(theta){ dgamma(theta,1,1)}

seq <- seq(100,10000, by = 100)

likel.pois <- Vectorize(function(theta){((theta^sum(obs))*exp(-length(obs)*theta))/prod(factorial(obs))})

eps.HM <- function(t){
  sample <- rgamma(t, shape = sum(obs) + 1, rate = 1 + length(obs))
  out <- 1/mean(1/likel.pois(sample))
  return(out)
  }


vec = c()
for (i in seq){
  vec <- c(vec,eps.HM(i))
}

plot(vec, main = "Harmonic mean", xlab = "num of values", ylab = "", col = "orange", pch = 20, lwd = 3)
abline(h = 4.314297e-09, lty = 2, col = "orchid", lwd = 3)
```



Ex 5.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.

```
import.sampl <- Vectorize(function(t){

  val <- abs(rcauchy(t,0,1))
  out <- mean(likel.pois(val)*dgamma(val,1,1)/(2*dcauchy(val,0,1)))

  return(out)
  })


r <- import.sampl(seq)

plot(x = seq,y = r, main = "Importance sampling", xlab = "grid of values", ylab = "", col = "blue", pch = 20, lwd
 = 3)
abline(h = 4.314297e-09, lty = 2, col = "orchid", lwd = 3)
```



Ex 6.A

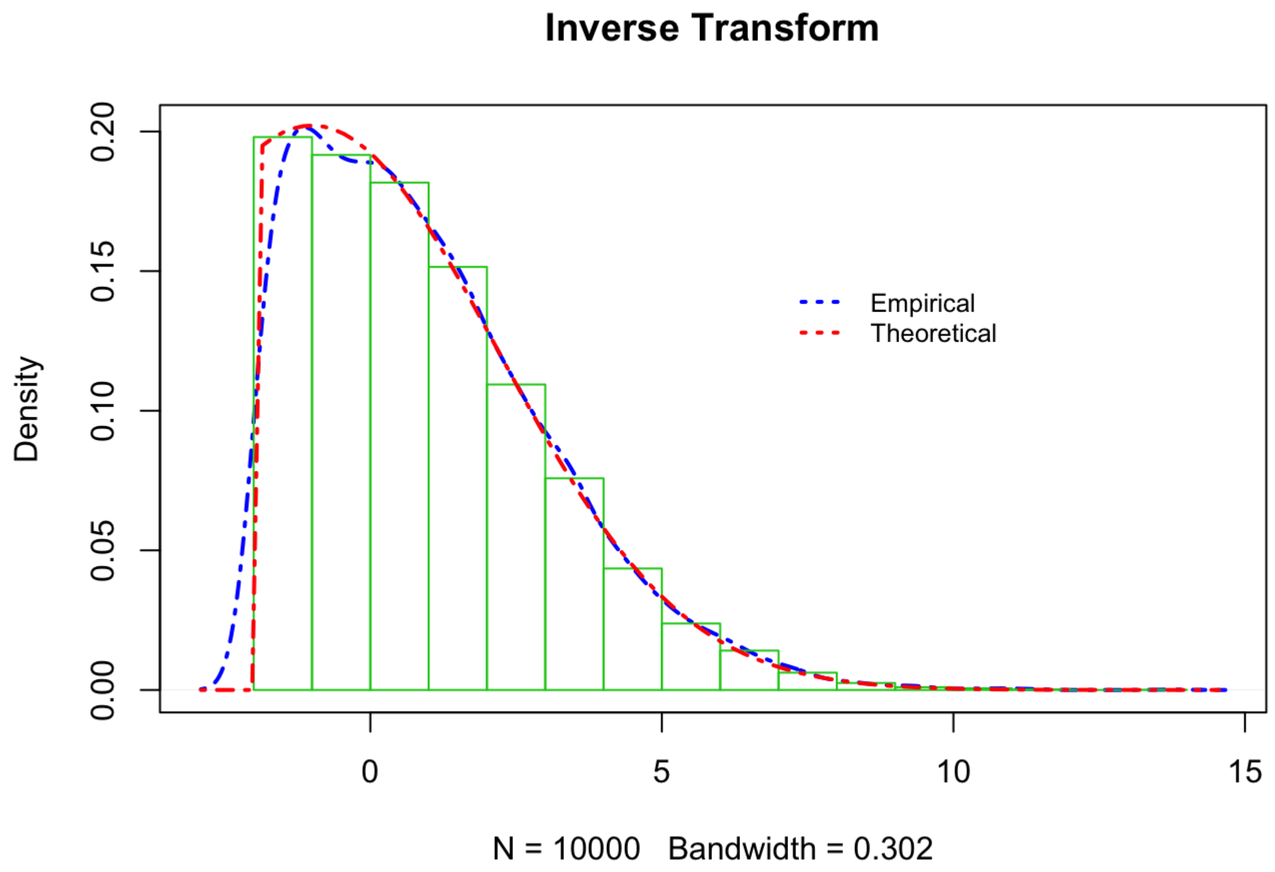
Provide two alternative implemetations of 10000 i.i.d. simulations from the following left-truncated normal distribution

$$f(x) \propto exp\{-\frac{1}{2}\frac{(x+1)^2}{10}\}I_{(-2,\infty)}(x)$$

using:

- integral transform i.e. using the inverse CDF

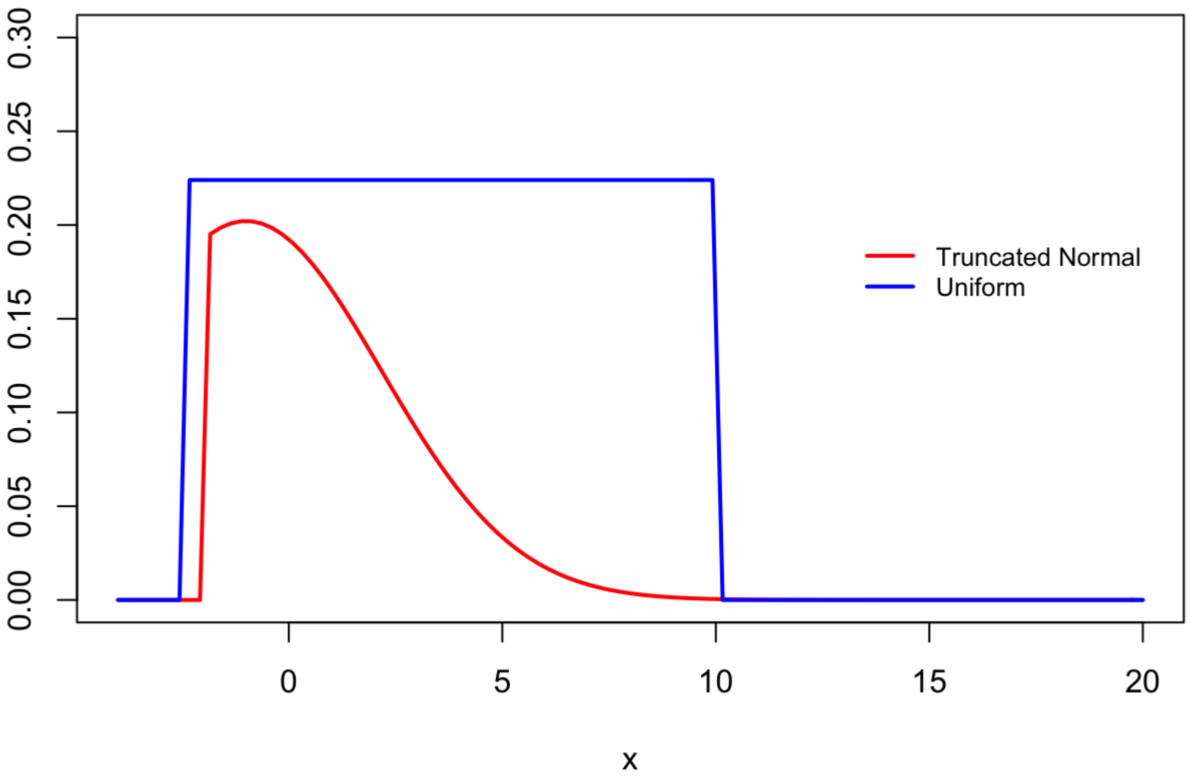
```r
library(truncnorm)
a <- -2
b <- Inf

mu <- -1
sigma2 <- 10
sd <- sqrt(sigma2)

alpha = (a - mu)/sd
beta = (b -mu)/sd

x <- qnorm(pnorm(alpha)+runif(10000)*(pnorm(beta)-pnorm(alpha)))*sd+mu


plot(density(x), col = "blue", lwd = 2, lty = 4, main = "Inverse Transform") #empirical
hist(x,freq = F ,add = T, border = "#1acc1d", cex = 1.3)
curve(dtruncnorm(x, a = a, b = b, mean = -1, sd = sqrt(10)), col = "red",lwd = 2,
      lty = 4,add = T) #theoretical
legend(x = 7,y = 0.15, legend = c("Empirical","Theoretical"), col = c("blue","red"), lty = c("dotted"),cex = 0.8,
 lwd = 2, bty = "n")
```



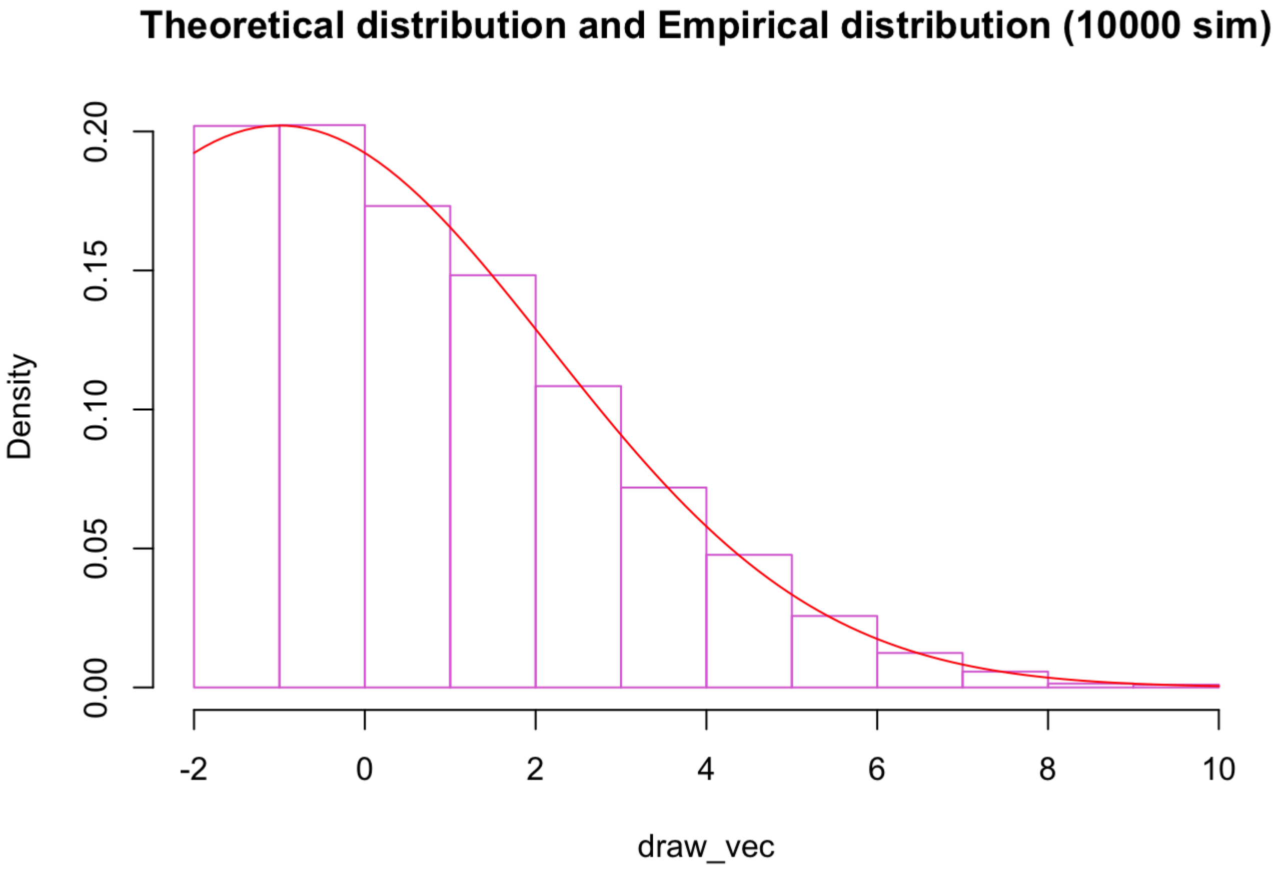
Ex 6.B

- Acceptance-rejection

```
library(truncnorm)

k <- 2.8
curve(dtruncnorm(x, a = a, b = b, mean = -1, sd = sqrt(10)), col = "red", xlim = c(-4,20),ylim = c(0,.3), ylab =
"", lwd = 2)
curve(dunif(x,-2.5,10)*k,col = "blue", ylim = c(0,.8),add = T, lwd = 2)
legend(x = 13,y = 0.20, legend = c("Truncated Normal","Uniform"), col = c("red","blue"),cex = 0.8, lwd = 2, bty =
 "n")
```



```r
funct <- function(x){
  dunif(x,-2,10)
}

draw_from_q=function(n){
  runif(n,-2,10)
}

f <- function(x){
  dtruncnorm(x, a = a, b = b, mean = -1, sd = sqrt(10))
}



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))

}


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

for(i in 1:mcsize){

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

}


hist(draw_vec,freq=FALSE, border = "orchid", main = "Theoretical distribution and Empirical distribution (10000 sim)")
curve(f(x),add=TRUE, col = "red")
```


Theoretical distribution and Empirical distribution (10000 sim)
Density
draw_vec