

Project 1 - TMA4300

Magnus Grytten & Petter J. Gudbrandsen

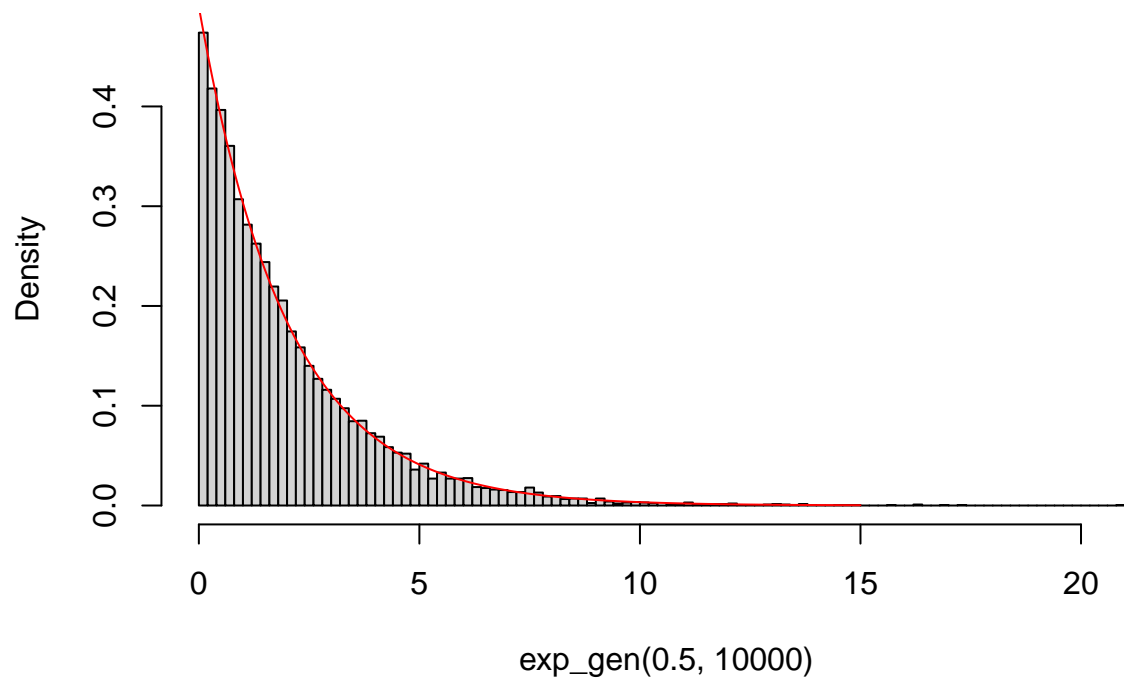
A

1.

In this project all red lines in plots are theoretical density functions.

```
exp_gen <- function(lam, n) {  
  u = runif(n)  
  -(1/lam)*log(u)  
}  
  
hist(exp_gen(0.5,10000),breaks = 100, freq = FALSE)  
lines(c(0:1500)/100,dexp(c(0:1500)/100,0.5),col="red")
```

Histogram of exp_gen(0.5, 10000)



2.

(a)

We integrate to find the cumulative distribution function,

$$G(x) = \int_{-\infty}^x g(x) dx = \begin{cases} \frac{c}{\alpha} x^{\alpha} & 0 < x \leq 1 \\ -ce^{-x} + \frac{c}{e} + \frac{c}{\alpha} & 1 \leq x \\ 0 & \text{otherwise} \end{cases}$$

We find the normalizing constant,

$$\lim_{x \rightarrow \infty} G(x) = \frac{c}{e} + \frac{c}{\alpha} = 1 \Rightarrow c = \frac{e\alpha}{e + \alpha}$$

The inverse of the cumulative distribution function then becomes,

$$G^{-1}(x) = \begin{cases} \sqrt[\alpha]{\frac{\alpha}{c}x} & 0 < x \leq \frac{c}{\alpha} \\ -\ln(-\frac{x}{c} + \frac{1}{e} + \frac{1}{\alpha}) & \frac{c}{\alpha} \leq x \\ 0 & \text{otherwise} \end{cases}$$

(b)

```
invG <- function(x,c,a){
  if (x < c/a){
    k <- ((a/c)*x)**a
  }
  else if (c/a <= x){
    k <- -log(-x/c + 1/exp(1) + 1/a)
  }
  else {
    print("Error")
  }
  k
}

g_gen <- function(a,n){
  c <- (exp(1)*a) / (exp(1)+a)
  u <- runif(n)

  ifelse(u < c/a,
        ((a/c) * u) ^ (1/a),
        -1 * log((1/exp(1)) + (1/a) - (u/c)))
}

g_dens <- function(x,a){
  c <- (exp(1)*a) / (exp(1)+a)

  gx <- c(1:length(x))
  for (i in c(1:length(x))){
    if (x[i] <= 0){
      gx[i] <- 0
    }
  }
}
```

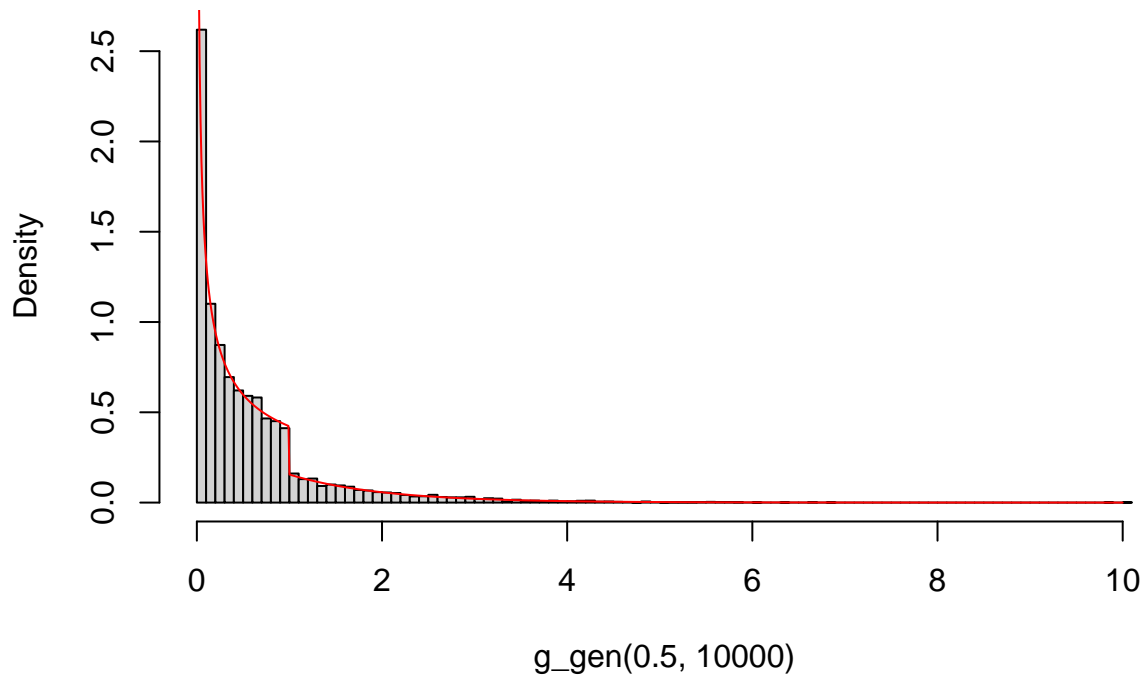
```

    }
    else if (1 <= x[i]){
      gx[i] <- c*exp(-x[i])
    }
    else{
      gx[i] <- c*x[i]^(a-1)
    }
  }
  gx
}

hist(g_gen(0.5,10000),breaks = 100, freq = FALSE)
lines(c(1:1000)/100,g_dens(c(1:1000)/100,0.5),col="red")

```

Histogram of g_gen(0.5, 10000)



3.

(a) We integrate to find the cumulative distribution function,

$$F(x) = \int_{-\infty}^x f(x) dx = \frac{c}{\alpha} \left(1 - \frac{1}{1 + e^{\alpha x}}\right)$$

$$\lim_{x \rightarrow \infty} F(x) = \frac{c}{\alpha} = 1 \quad \Rightarrow \quad c = \alpha$$

(b) Setting in the normalizing constant from (a) we find the cumulative distribution function,

$$F(x) = 1 - \frac{1}{1 + e^{\alpha x}}$$

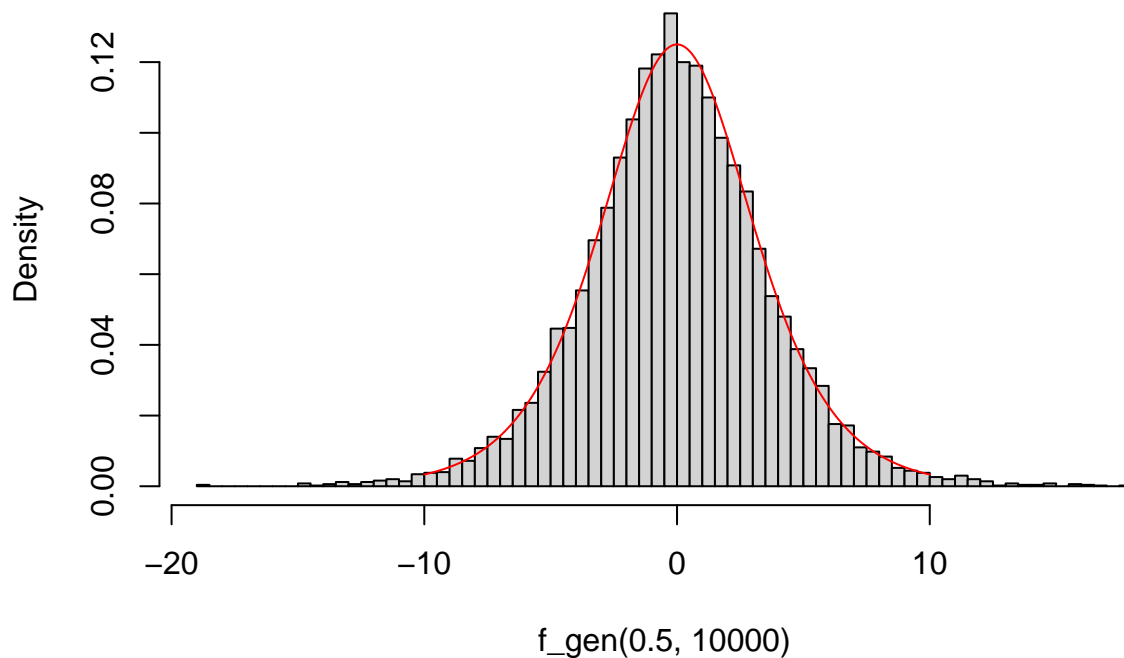
The inverse of the cumulative distribution function then becomes,

$$F^{-1}(x) = \frac{\ln(\frac{x}{1-x})}{\alpha}$$

(c)

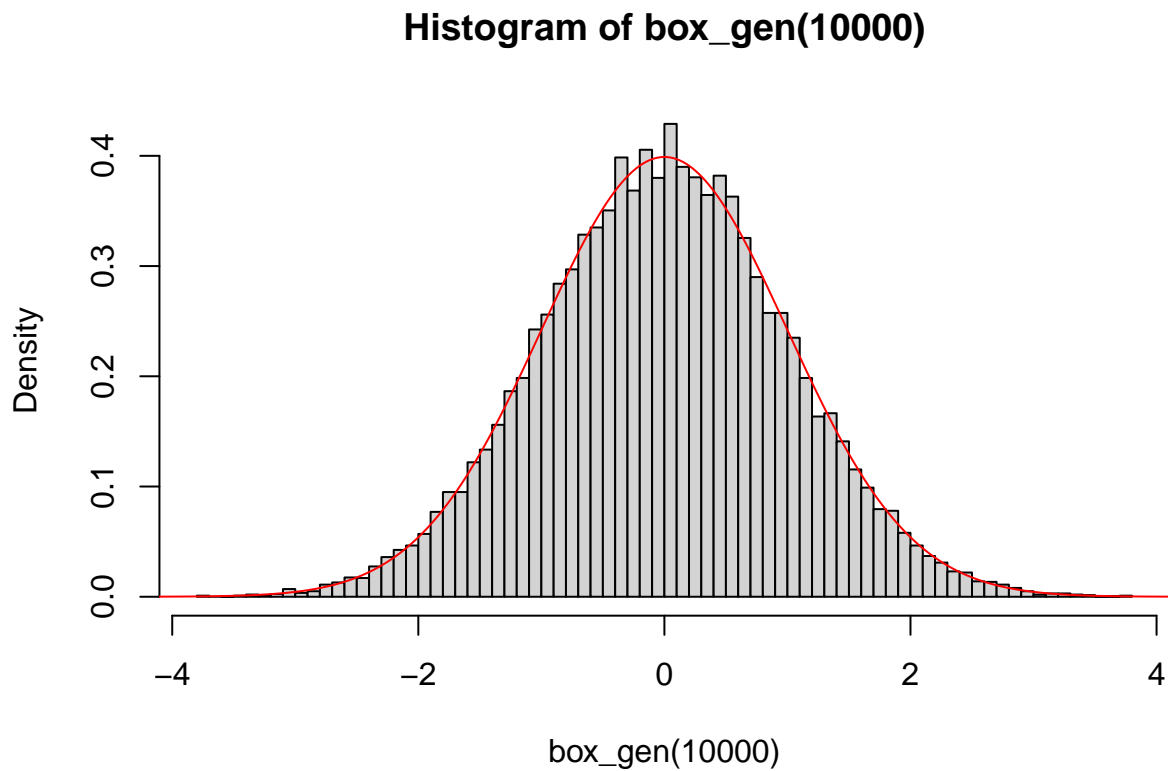
```
invF <- function(x,a){  
  k <- log(x/(1-x))/a  
}  
  
f_gen <- function(a, n) {  
  u <- runif(n)  
  v <- invF(u,a)  
  v  
}  
  
f_dens <- function(x,a){  
  (a*exp(a*x))/((1+exp(a*x))^2)  
}  
  
hist(f_gen(0.5,10000),breaks = 100, freq = FALSE)  
lines(c(-1000:1000)/100,f_dens(c(-1000:1000)/100,0.5),col="red")
```

Histogram of f_gen(0.5, 10000)



4.

```
box_gen <- function(n) {  
  x1 <- runif(n) * 2*pi  
  x2 <- exp_gen(0.5, n) #right?  
  
  y1 <- sqrt(x2) * sin(x1)  
  y2 <- sqrt(x2) * cos(x1)  
  
  c(y1,y2)  
}  
  
hist(box_gen(10000), breaks=100, freq = FALSE)  
lines(c(-500:500)/100, dnorm(c(-500:500)/100), col="red")
```



5.

```
dnormal <- function(my,sig,n) {  
  
  d = length(my)  
  
  y <- matrix(nrow=n, ncol=d)  
  x <- matrix(box_gen(d*n/2), d, n)
```

```

A = t(chol(sig))

my_matrix <- matrix(rep(my,n), nrow=d, ncol=n)

y <- my_matrix + A%%x

y
}

my <- c(1,2)
sig <- matrix(c(5,1,1,3), nrow=2, ncol=2)
print(sig)#Given Sigma = Cov[x,y] matrix

```

```

##      [,1] [,2]
## [1,]    5    1
## [2,]    1    3

```

```

y <- dnormal(my,sig,10000)
cov(t(y))#Estimated Sigma from samples

```

```

##      [,1] [,2]
## [1,] 5.083455 1.047834
## [2,] 1.047834 2.941718

```

```

print(my)#Given my = E[x] vector

```

```

## [1] 1 2

```

```

rowMeans(y)#Estimated my from samples

```

```

## [1] 0.968905 2.008550

```

B

1.

(a) We have,

$$\frac{f(x)}{g(x)} = \begin{cases} \frac{1}{\alpha\Gamma(\alpha)}e^{-x} & 0 < x \leq 1 \\ \frac{1}{\alpha\Gamma(\alpha)}x^{\alpha-1} & 1 \leq x \end{cases} \leq c$$

Since both e^{-x} and $x^{\alpha-1}$ are strictly decreasing for $|\alpha| < 1$ they are both \leq than their values at the left boundaries. Giving us,

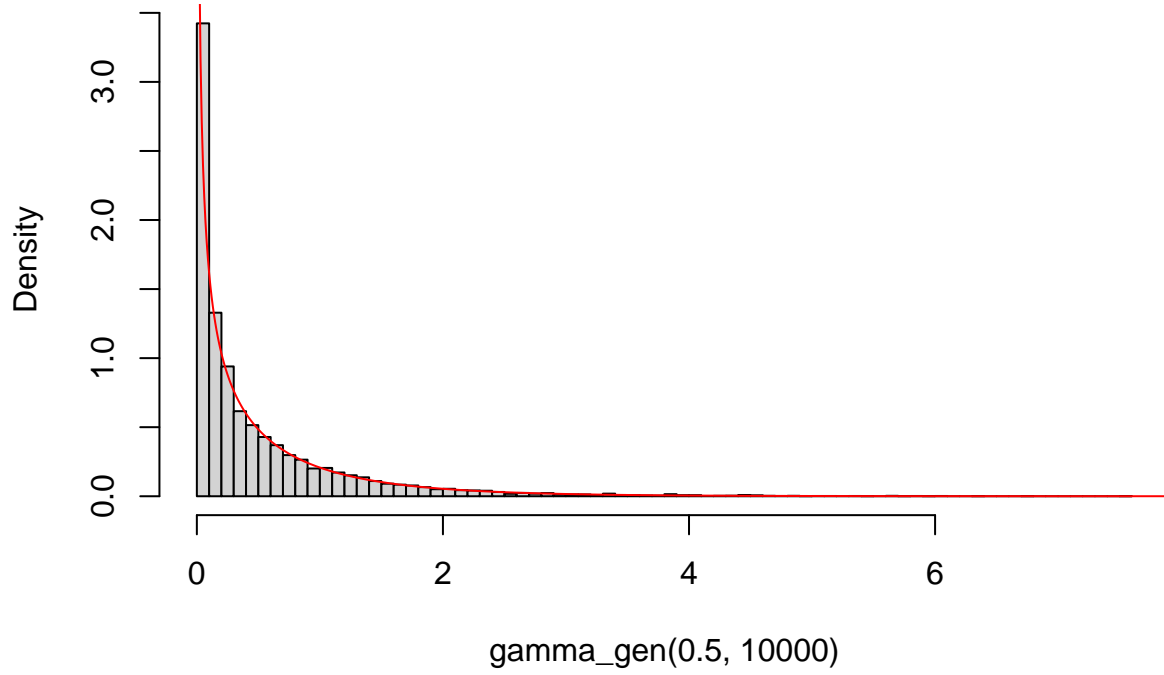
$$\frac{f(x)}{g(x)} \leq \frac{1}{\alpha\Gamma(\alpha)} = c$$

Giving us the acceptance probability $c^{-1} = \alpha\Gamma(\alpha)$.

(b)

```
gamma_gen <- function(alph, n) {  
  if(1/(alph*gamma(alph)) < 1){  
    c <- 1  
  }  
  else{  
    c <- 1/(alph*gamma(alph))  
  }  
  
  i <- 0  
  y <- c(1:n)*0  
  
  while (i < n){  
    x <- g_gen(alph,1)  
    u <- runif(1)  
  
    ifelse(x<1,  
      a <- (1/c)*1/(alph*gamma(alph))*exp(-x),  
      a <- (1/c)*1/(alph*gamma(alph))*x^(alph-1))  
    if (u<=a) {  
      y[i+1] <- x  
      i <- i+1  
    }  
  }  
  y  
}  
  
hist(gamma_gen(0.5,10000), breaks=100, freq = FALSE)  
lines(c(0:1000)/100,dgamma(c(0:1000)/100,0.5),col="red")
```

Histogram of gamma_gen(0.5, 10000)



2.

(a)

we start by finding a .

$$\begin{aligned}
 \frac{d}{dx} f^*(x) &= 0 \\
 (\alpha - 1)x^{\alpha-2}e^{-x} - x^{\alpha-1}e^{-x} &= 0 \\
 x^{\alpha-1}((\alpha - 1) - x) &= 0 \\
 \Downarrow \\
 x = 0, \quad x = \alpha - 1 \\
 a = \sqrt{\sup_x f^*(x)} &= \sqrt{f^*(\alpha - 1)} = \sqrt{(\alpha - 1)^{\alpha-1}e^{-\alpha+1}}
 \end{aligned}$$

We use the same method for b_+ ,

$$\begin{aligned}
 \frac{d}{dx} (x^2 f^*(x)) &= 0 \\
 (\alpha + 1)x^\alpha e^{-x} - x^{\alpha+1}e^{-x} &= 0 \\
 \Downarrow \\
 x = 0, \quad x = \alpha + 1
 \end{aligned}$$

$$b_+ = \sqrt{\sup_x x^2 f^*(x)} = \sqrt{x^2 f^*(\alpha + 1)} = \sqrt{(\alpha + 1)^{\alpha+1} e^{-\alpha-1}}$$

Since $f^*(x)$ is 0 for all $x \leq 0$ $b_- = 0$.

(b)

```
#Function generating samples from the gamma distribution using the ratio of uniforms method
rou <- function(n,alph){

  a <- sqrt((alph-1)^(alph-1)*exp(-(alph-1)))
  b <- sqrt((alph+1)^(alph+1)*exp(-(alph+1)))

  f_star <- function(x,alph){
    x^(alph-1)*exp(-x)
  }

  y <- c(1:n)*0
  tries <- 0
  i <- 0
  while (i < n){
    tries <- tries +1
    x1 <- runif(1,0,a)
    x2 <- runif(1,0,b)

    if(x1 <= sqrt(f_star(x2/x1,alph))){
      i <- i+1
      y[i] <- x2/x1
    }

  }

  y
}

#Altered version of rou, implemented in log scale. Does not work for alpha = 1
rou_log <- function(n,alph,returnTries = FALSE){

  la <- (alph-1)/2*log(alph-1) - (alph-1)/2
  lb <- (alph+1)/2*log(alph+1) - (alph+1)/2

  y <- c(1:n)*0
  tries <- 0
  i <- 0
  while (i < n){
    tries <- tries +1
    #These samples are uniformly distributed after taking the exponetial
    l1 <- la - exp_gen(1,1)
    l2 <- lb - exp_gen(1,1)

    #log(x1) <= log(sqrt(f*(x2/x1)))
    if( l1 <= (alph-1)/2 *(l2-l1) - exp( l2-l1 )/2 ){
      i <- i+1
    }
  }

  y
}
```

```

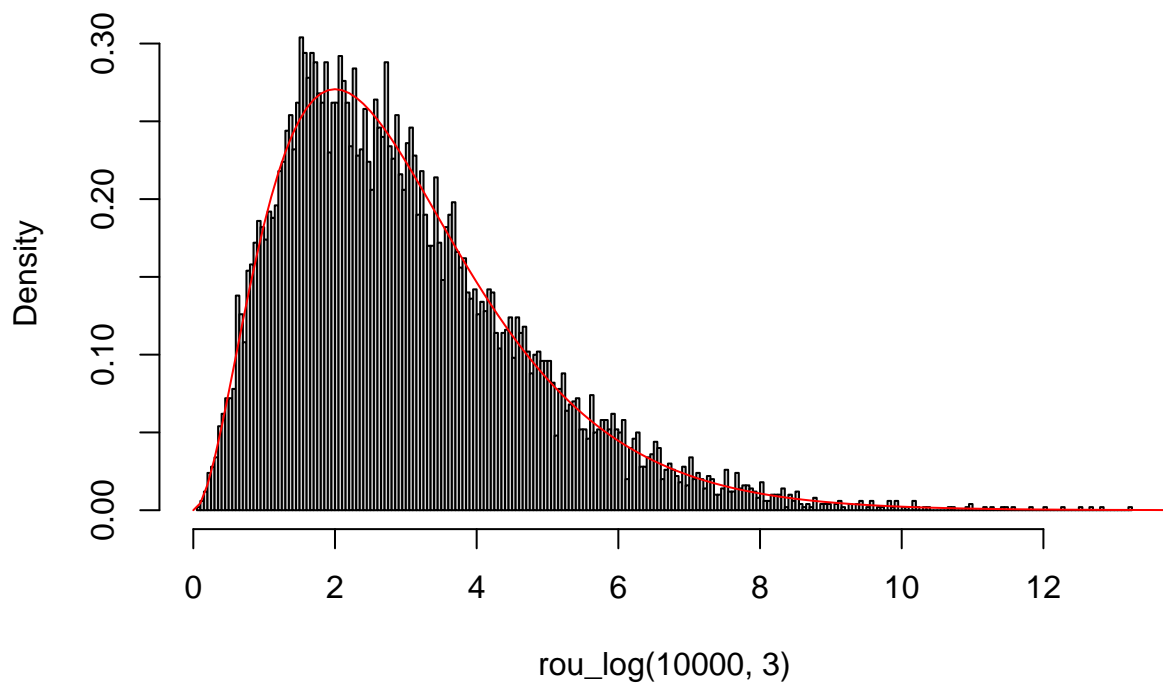
    y[i] <- exp(l2-l1)
  }

}
if(returnTries){
  tries
}
else{
  y
}
}
}

hist(rou_log(10000,3),breaks=200,freq=FALSE)
lines(c(0:150)/10,dgamma(c(0:150)/10,3),col="red")

```

Histogram of rou_log(10000, 3)



```

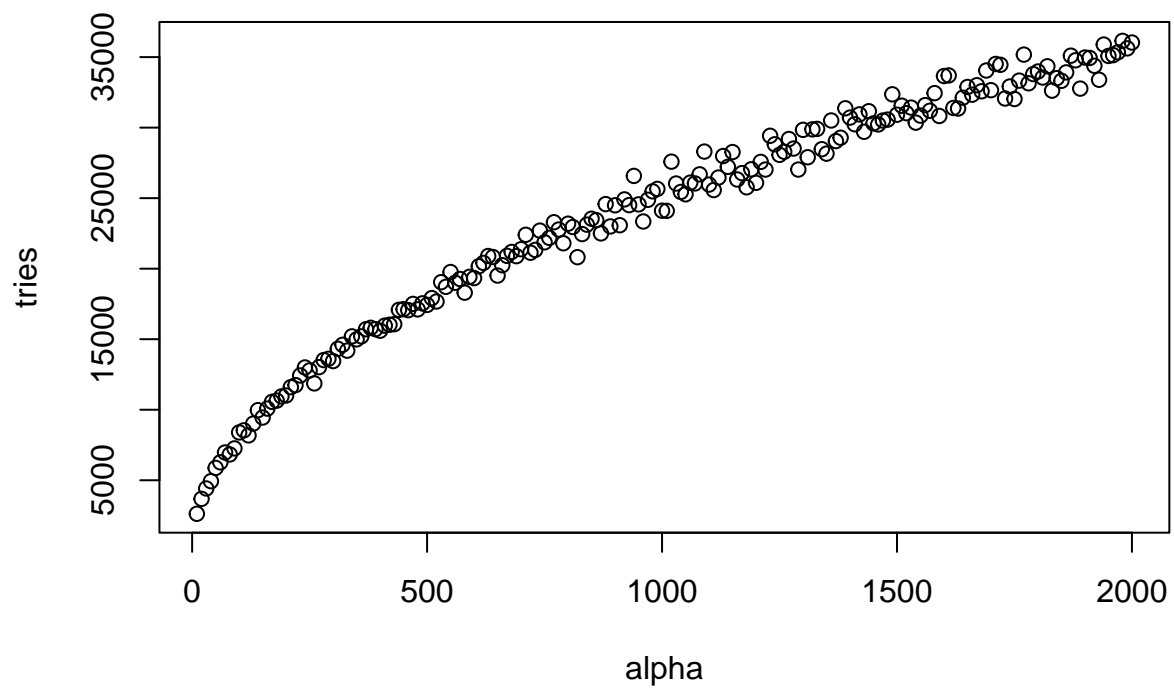
plot_tries <- function(){

  alpha <- c(1:200)*10

  tries <- c(1:200)*0
  for (i in 1:200){
    tries[i] <- rou_log(1000,alpha[i],TRUE)
  }
  plot(alpha,tries)
}

```

```
}
plot_tries()
```

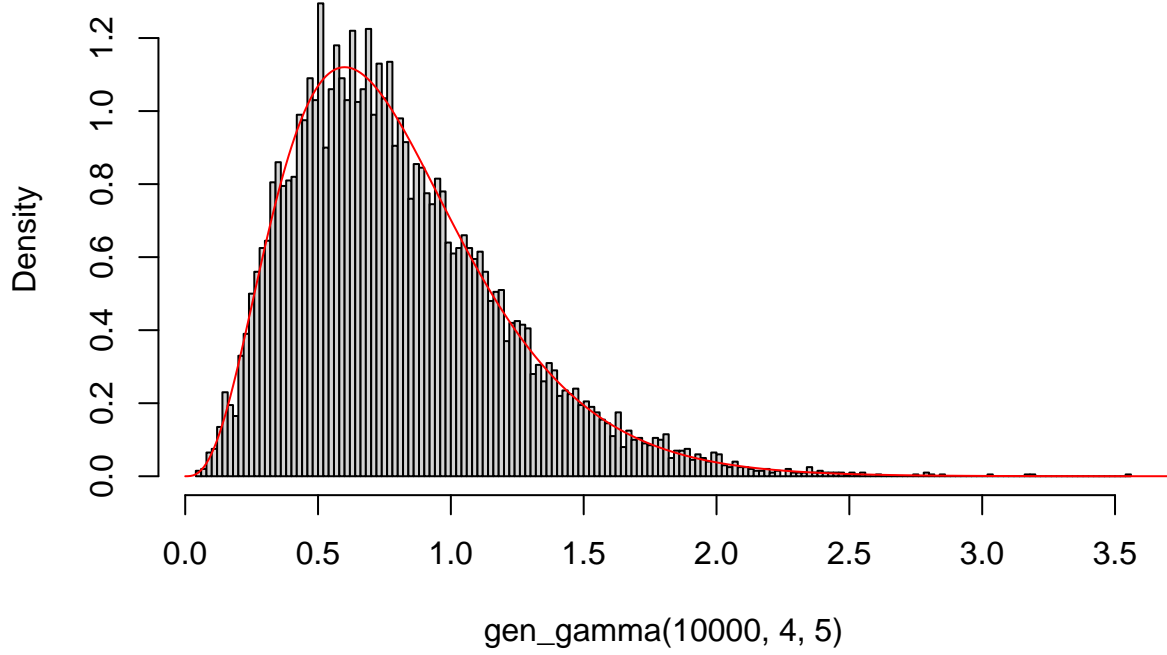


3.

```
gen_gamma <- function(n,alpha,beta){
  x <- c(1:n)*0
  ifelse(alpha == 1,
    x <- rou(n,alpha)/beta,
    x <- rou_log(n,alpha)/beta)
  x
}

hist(gen_gamma(10000,4,5),breaks=200,freq=FALSE)
lines(c(0:1500)/100,dgamma(c(0:1500)/100,4,rate=5),col="red")
```

Histogram of gen_gamma(10000, 4, 5)



4.

(a)

We have $x \sim \text{Gamma}(\alpha, 1)$ and $y \sim \text{Gamma}(\beta, 1)$, we want the distribution of $z = \frac{x}{x+y}$. We create another stochastic variable $w = x + y$, since it's a sum of two gamma distributions, it is also a gamma distribution with density,

$$f_W(w) = \frac{1}{\Gamma(\alpha + \beta)} w^{\alpha + \beta - 1} e^{-w}$$

We find the inverses, $x = zw$ and $y = w(1 - z)$, giving the Jacobian,

$$|J| = \begin{vmatrix} w & -w \\ z & (1 - z) \end{vmatrix} = w$$

Since x and y are independent their joint distribution is the product of their individual distributions. $f_{X,Y}(x, y) = f_X(x)f_Y(y)$.

$$\begin{aligned} f_{Z,W}(z, w) &= f_{X,Y}(g^{-1}(z, w))|J| = f_X(g_1^{-1}(z, w))f_Y(g_2^{-1}(z, w))|J| \\ &= \frac{1}{\Gamma(\alpha)} (zw)^{\alpha-1} e^{-zw} \frac{1}{\Gamma(\beta)} (w(1-z))^{\beta-1} e^{-w(1-z)} w \\ &= \frac{1}{\Gamma(\alpha)\Gamma(\beta)} z^{\alpha-1} (1-z)^{\beta-1} \cdot w^{\alpha+\beta-1} e^{-w} \end{aligned}$$

Since we have that $f_{Z,W}(z, w) = g(z) \cdot h(w)$ we know that z and w are independent variables, and thus we can find the distribution of z by dividing the joint distribution by the distribution of w .

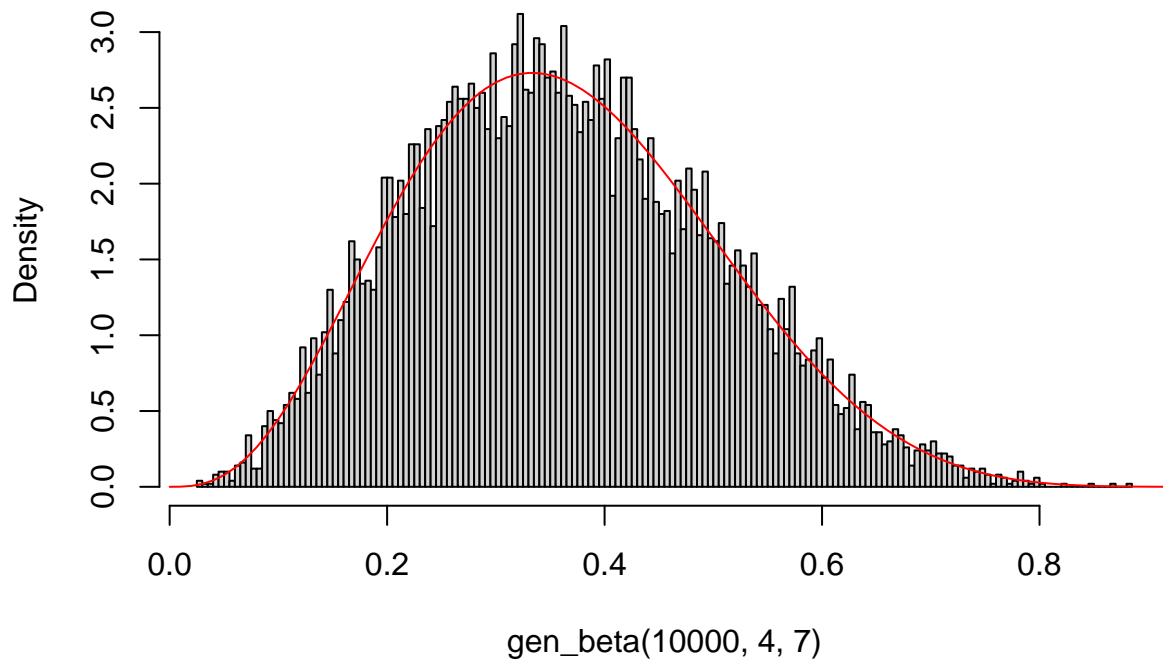
$$f_Z(z) = \frac{f_{Z,W}(z,w)}{f_W(w)} = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} z^{\alpha-1} (1-z)^{\beta-1}$$

(b)

```
gen_beta <- function(n,alpha,beta){
  x <- gen_gamma(n,alpha,1)
  y <- gen_gamma(n,beta,1)
  z <- x/(x+y)
  z
}

hist(gen_beta(10000,4,7),breaks=200,freq=FALSE)
lines(c(0:100)/100,dbeta(c(0:100)/100,4,7),col = "red")
```

Histogram of gen_beta(10000, 4, 7)



C

1.

```
n = 100000
samples <- box_gen(n)[1:n]
```

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n I(X_i > 4)$$

$$E[\hat{\theta}] = E[I(X_i > 4)] = \overline{I(X_i > 4)}$$

$$\text{Var}[\hat{\theta}] = \frac{1}{n^2} \sum_{i=1}^n \text{Var}[I(X_i > 4)] = \frac{1}{n} \text{Var}[I(X_i > 4)]$$

$$SD[\hat{\theta}] = \frac{SD[I(X_i > 4)]}{\sqrt{n}}$$

This gives the confidence interval $E[\hat{\theta}] \pm z_{0.05/2} \cdot SD[\hat{\theta}]$. Here $z_{0.05/2}$ is the 0.025-quantile of a standard normal distribution.

```
# Expected theta
h <- samples > 4 # We set h(x) = I(x > 4)
exp_theta <- sum(h) / n

# 95% confidence interval
sd_theta <- sd(h) / sqrt(n)
z <- 1.960
conf_int1 <- c(exp_theta - z*sd_theta, exp_theta + z*sd_theta)
conf_int1
```

```
## [1] -7.718447e-06 4.771845e-05
```

```
# Interval size
conf_int1[2] - conf_int1[1]
```

```
## [1] 5.543689e-05
```

2.

$$h(x) = I(X > 4)$$

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

$$g(x) = cxe^{-\frac{1}{2}x^2}, x > 4$$

$$\int_4^{\infty} g(x) dx = 1 \implies c(e^{-8} - e^{-\frac{1}{2}x^2}) = 1 \implies c = e^8$$

$$G(x) = \int_4^x g(x) dx = 1 - e^{-\frac{1}{2}x^2 - 8}$$

$$G^{-1}(x) = \sqrt{-2 \cdot \ln(1-x) - 8}$$

$$w(x) = \frac{f(x)}{g(x)} = \frac{1}{2\pi e^8 x}$$

$$E[\hat{\theta}] = \frac{\sum_{i=1}^n h(x_i) \cdot w(x_i)}{n} = \frac{1}{2\pi e^8 n} \sum_{i=1}^n \frac{1}{x_i}$$

$$Var[\hat{\theta}] = \frac{1}{(2\pi e^8 n)^2} Var\left[\sum_{i=1}^n \frac{1}{x_i}\right] = \frac{1}{4\pi^2 e^{16} n} Var\left[\frac{1}{x}\right]$$

$$SD[\hat{\theta}] = \frac{1}{2\pi e^8 \sqrt{n}} SD\left[\frac{1}{x_i}\right]$$

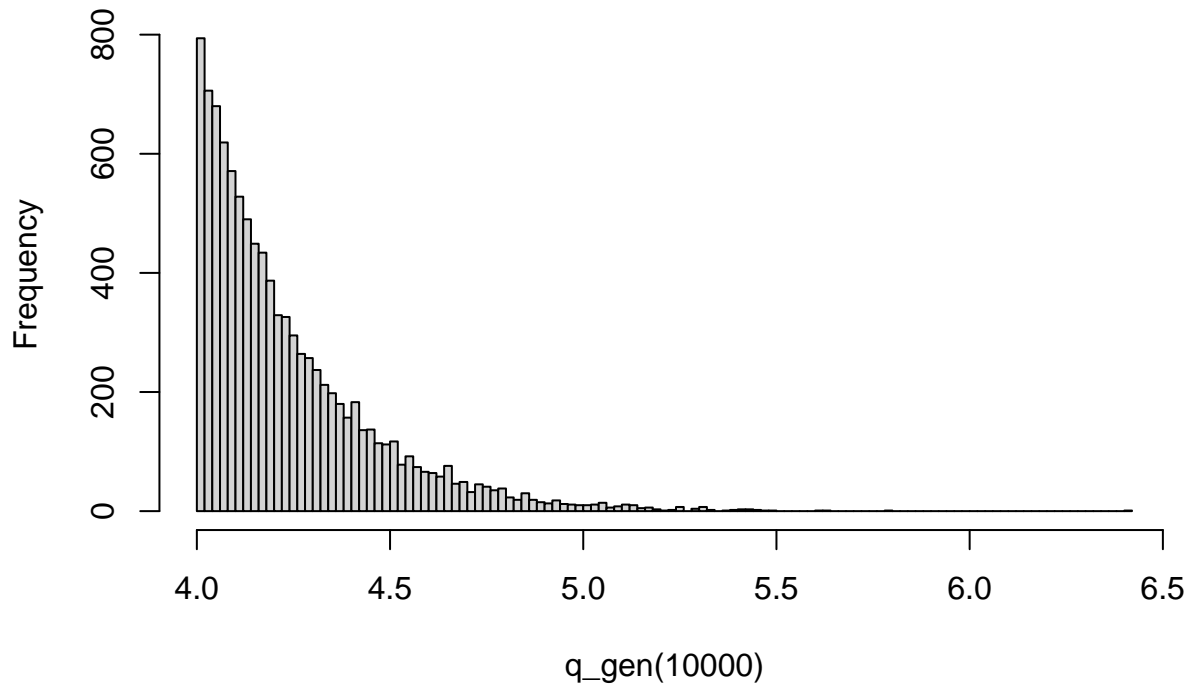
This gives the confidence interval $E[\hat{\theta}] \pm z_{0.05/2} \cdot SD[\hat{\theta}]$. Here $z_{0.05/2}$ is the 0.025-quantile of a standard normal distribution.

```
q_gen <- function(n) {
  u <- runif(n)

  sqrt(-2*log(1-u) + 16)
}

hist(q_gen(10000), breaks = 100)
```

Histogram of q_gen(10000)



```
n <- 100000
q_samples <- q_gen(n)
exp_theta2 <- sum(1/q_samples) * (1/(2*pi*n*exp(8)))
sd_theta2 <- sd(1/q_samples) * (1/(2*pi*sqrt(n)*exp(8)))
z <- 1.960
conf_int2 <- c(exp_theta2 - z*sd_theta2, exp_theta2 + z*sd_theta2)

conf_int1[2] - conf_int1[1]
```

```
## [1] 5.543689e-05
```

```
conf_int2[2] - conf_int2[1]
```

```
## [1] 7.661948e-09
```

The confidence interval in task 2 is less by a factor of 10^4 . To get that accuracy in task 1 we would need $n \gg 10^8$ samples (too big for our laptops).

3.

(a)

```
anti_q_gen <- function(n){  
  u <- runif(n)  
  anti_u <- 1 - u  
  
  c(sqrt(-2*log(1-u) + 16), sqrt(-2*log(1-anti_u) + 16))  
}
```

(b)

```
n_3 <- 50000 # We use half the size of n since anti_q_gen(n) generates 2n variates  
anti_q_samples <- anti_q_gen(n_3)  
exp_theta3 <- sum(1/anti_q_samples) * (1/(2*pi*n_3*exp(8)))  
sd_theta3 <- sd(1/anti_q_samples) * (1/(2*pi*sqrt(n_3)*exp(8)))  
z <- 1.960  
conf_int3 <- c(exp_theta3 - z*sd_theta3, exp_theta3 + z*sd_theta3)  
  
conf_int3[2] - conf_int3[1] # task 3
```

```
## [1] 1.08921e-08
```

```
conf_int2[2] - conf_int2[1] # task 2
```

```
## [1] 7.661948e-09
```

The interval is about 30% bigger in task 3 compared to task 2.

D

1.

```
y <- c(125, 18, 20, 34)  
  
f_theta_y_not_normalized <- function(theta){###  
  (((2+theta)^(y[1]))*((1-theta)^(y[2]+y[3]))*(theta^(y[4])))  
}
```



```

normalizing_c <- unlist(integrate(f_theta_y_not_normalized, 0, 1)[1])

# Posterior distribution
f_theta_y_dens <- function(theta){
  (((2+theta)^(y[1]))*((1-theta)^(y[2]+y[3]))*(theta^(y[4]))) / normalizing_c
}

gen_f_theta_y <- function(n){
  # c = max(f/g). g(theta|y) is 1 and max(f(theta|y)) = (sqrt(53809)+15)/394 is
  # found by solving d/d(theta) f(theta) = 0.
  c <- f_theta_y_dens((sqrt(53809)+15)/394)

  i <- 0
  y <- (1:n)*0

  while (i < n){
    x <- runif(1)

    # The acceptance probability is f(x)/(c*g(x)). g(x) is always 1 in beta(1, 1)
    accept_prob <- f_theta_y_dens(x)/c

    u <- runif(1)
    if (u <= accept_prob) {
      y[i+1] <- x
      i <- i+1
    }
  }

  y
}

```

2.

```

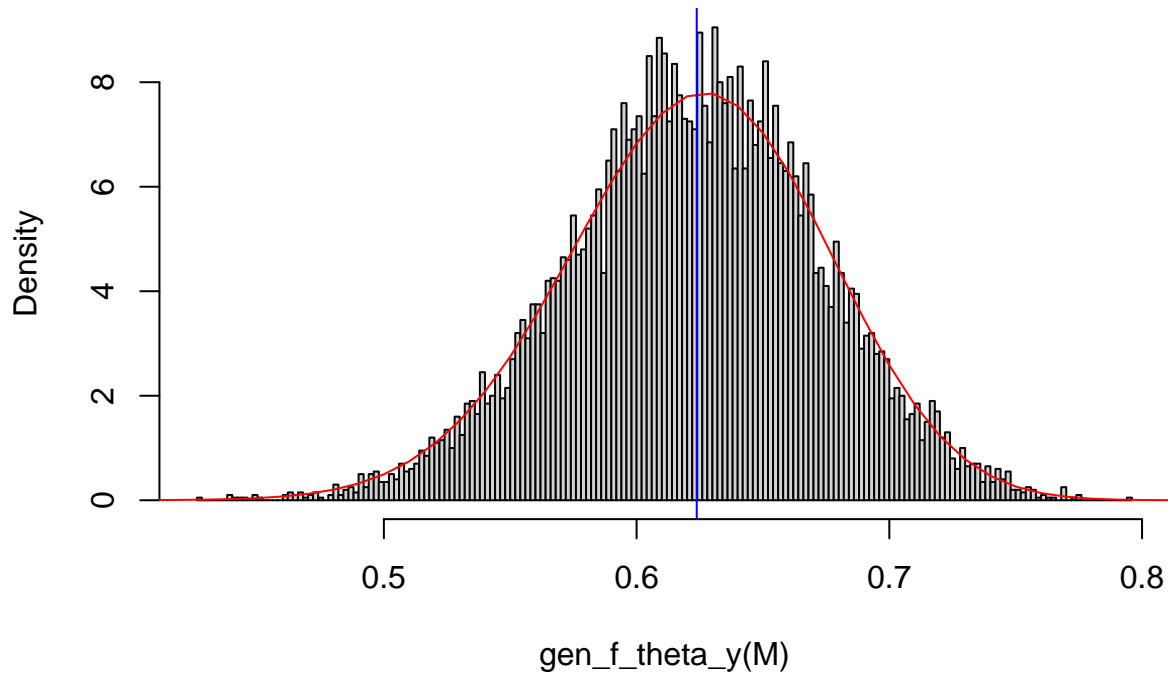
M = 10000
samples <- gen_f_theta_y(M)

# E[f(theta|y)], the posterior mean
post_mean <- sum(samples)/M

hist(gen_f_theta_y(M), breaks=200, freq=FALSE)
lines((0:100)/100, f_theta_y_dens(c(0:100)/100), col="red")
abline(v = post_mean, col = 'blue')

```

Histogram of $\text{gen_f_theta_y}(M)$



```
post_mean_num_int <- unlist(integrate(function(theta) theta*f_theta_y_dens(theta), 0, 1)[1])
post_mean_num_int
```

```
##      value
## 0.6228061
```

3.

```
# Similar to gen_f_theta_y(n), but finds the mean amount of tries per value generated
tries_for_1 <- function(n){

  # c = max(f/g). g(theta/y) is 1 and max(f(theta/y)) = (sqrt(53809)+15)/394 is
# found by solving d/d(theta) f(theta) = 0.
  c <- f_theta_y_dens((sqrt(53809)+15)/394)

  i <- 0
  tries <- 0

  while (i < n){
    tries <- tries + 1
    x <- runif(1)
    accept_prob <- f_theta_y_dens(x)/c
  }
```

```

    u <- runif(1)
    if (u <= accept_prob) {
      i <- i+1
    }
  }

  tries / n
}

tries_for_1(M)

```

```
## [1] 7.8251
```

The theoretically computed result is $\text{tries} = 1 / (\text{overall acceptance probability})$. The overall acceptance probability is:

$$P(U \leq \frac{1}{c} \cdot \frac{f(x)}{g(x)}) = \int_{-\infty}^{\infty} \frac{f(x)}{cg(x)} g(x) dx = \int_{-\infty}^{\infty} \frac{f(x)}{c} dx = c^{-1}$$

The number of tries is therefore c which we computed in D1.

```

num_tries <- f_theta_y_dens((sqrt(53809)+15)/394)
num_tries

```

```
##      value
## 7.799308
```

4.

```

beta_dens <- function(z,alpha,beta){
  ((gamma(alpha+beta))/(gamma(alpha)*gamma(beta)))*(z^(alpha-1))*((1-z)^(beta-1))
}

new_f_theta_y_dens <- function(theta){
  beta_dens(theta,1,5)*f_theta_y_dens(theta)
}

imp_sample <- function(x){

  hx = x
  fx = new_f_theta_y_dens(x)
  gx = f_theta_y_dens(x)
  wx = fx/gx

  sum(hx*wx)/sum(wx)
}

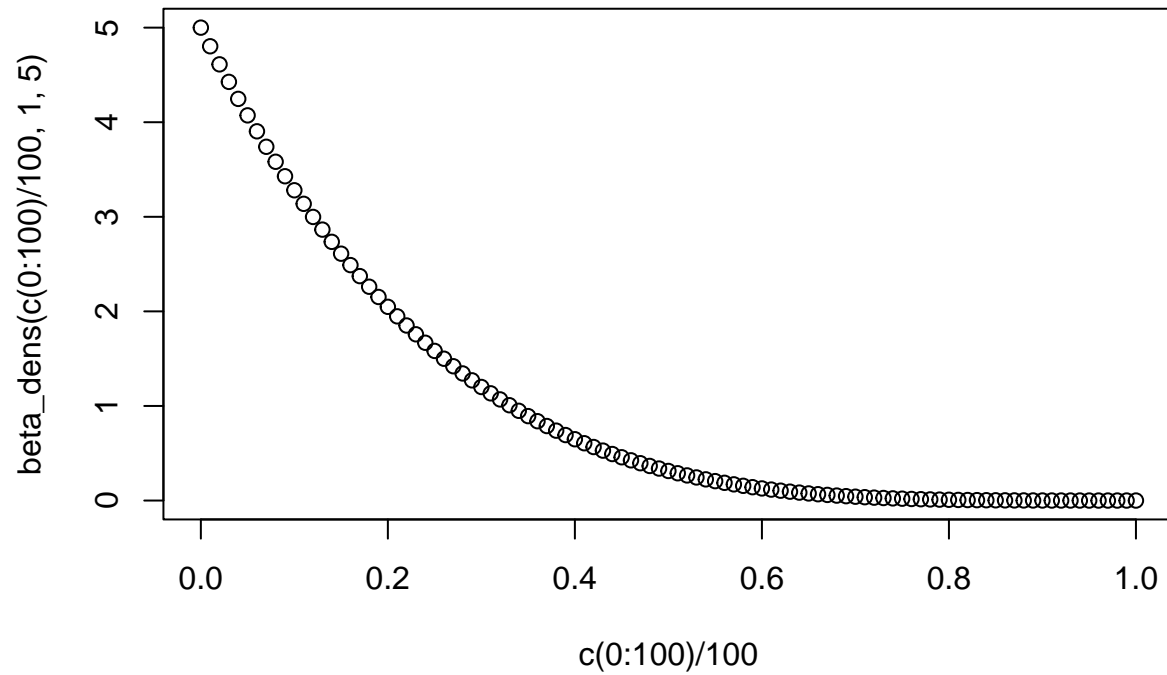
imp_sample(samples)

```

```
## [1] 0.596358
```

Looking at the density of $Beta(1, 5)$,

```
plot(c(0:100)/100,beta_dens(c(0:100)/100,1,5))
```



We see that it is weighted towards the lower part of the domain. It makes sense that the mean of the new posterior density is lower than the mean of the old posterior density, since the new density is the product of the old density and $Beta(1, 5)$