

# TMA4300 Computer Intensive Statistical Methods

## Exercise1, Spring 2014

*Mateusz Samiec*

## Problem A: Stochastic simulation by the probability integral transform

1. R function that generates samples from an exponential distribution with rate parameter  $\lambda$ .

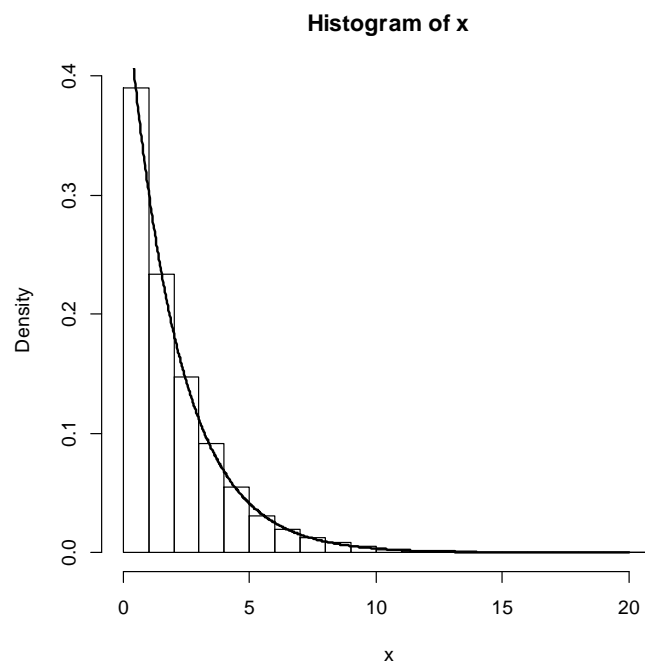
The function takes two arguments as input, the rate parameter of the exponential distribution  $\lambda$  and the number of samples to generate  $n$ .

```
#Exponential distribution
rExponent <- function(n,lambda){
  -log(runif(n))/lambda
}

#Shape of Exp. Dist.
x <- rExponent(n=10000,lambda=0.5)
hist(x,freq=FALSE)
t <- 0:1000/50
lines(t,dexp(t,rate=0.5),lwd=2)

#Properties
> mean(x)
[1] 2.00606
> var(x)
[1] 4.000849
```

As we see empirical mean and variance of the vector of generated samples are close to the theoretical moments:  $E[x] = \frac{1}{\lambda} = 2$ ,  $Var[x] = \frac{1}{\lambda^2} = 4$ . ( $\lambda = 0.5$ ).



2. The following probability function is under the consideration:

$$f(x) = \frac{ce^{\alpha x}}{(1 + e^{\alpha x})^2}, \quad -\infty < x < \infty.$$

a) In order to find the value of normalizing constant  $c$ , probability density function must be integrated from minus infinity to infinity. Thus we have:

$$F(x) = \int f(x)dx = \int \frac{ce^{\alpha x}}{(1 + e^{\alpha x})^2} dx = \frac{ce^{\alpha x}}{\alpha(1 + e^{\alpha x})}$$

Proof:

$$f(x) = F(x)' = \frac{c}{\alpha} \left[ \frac{\alpha e^{\alpha x}(1 + e^{\alpha x}) - \alpha e^{\alpha x} e^{\alpha x}}{(1 + e^{\alpha x})^2} \right] = \frac{ce^{\alpha x}}{(1 + e^{\alpha x})^2}$$

Further on:

$$\lim_{x \rightarrow \infty} F(x) = \lim_{x \rightarrow \infty} \frac{c \frac{e^{\alpha x}}{e^{\alpha x}}}{\alpha \left( \frac{1}{e^{\alpha x}} + \frac{e^{\alpha x}}{e^{\alpha x}} \right)} = \frac{c}{\alpha(0 + 1)} = \frac{c}{\alpha}$$

Because:

$$\lim_{x \rightarrow \infty} F(x) = 1$$

We have that:

$$c = \alpha$$

b) Cumulative distribution function was found in point A2 a) and it has the following form for normalizing constant  $c = \alpha$ :

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

In order to find the inverse of  $F(x)$  the following steps are performed:

$$\begin{aligned} p &= \frac{e^{\alpha x}}{1 + e^{\alpha x}} \Big| \cdot (-1) \\ \frac{1}{p} &= \frac{1}{e^{\alpha x}} + 1 \Big| - 1 \\ \frac{1}{p} - 1 &= \frac{1}{e^{\alpha x}} \Big| \ln() \\ \ln\left(\frac{1}{p} - 1\right) &= -\alpha x \Big| /(-\alpha) \end{aligned}$$

The final form is:

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

- c) R function that generates samples from  $f$  use the inverse technique. This means that random variables are generated from the inverse cumulative function where the argument is an uniformly distributed random variable  $p \sim U(0, 1)$ .

```
#Function that generate from pdf "f"
rF <- function(n,alpha){
  u <- runif(n)
  -log(1/u-1)/alpha
}
```

In order to check if the defined function is working properly the additional function was defined.

```
#Probability density function
f <- function(t, alpha){
  alpha*exp(alpha*t)/(1+exp(alpha*t))^2
}
```

Now the results may be compared with the real value. Therefore mean value, variance value and plot of distribution are compared.

```
> x <- rF(10000, 0.2)
> mean(x)
[1] 0.01408445
> var(x)
[1] 83.03979
> t <- -1000:1000/10
> Var(t,.2)
[1] 82.24666
```

As we see the mean value equal zero which is correct. We know it should be zero directly from the pdf function. Variances also confirm correctness of the defined R method `rF(n,alpha)`.

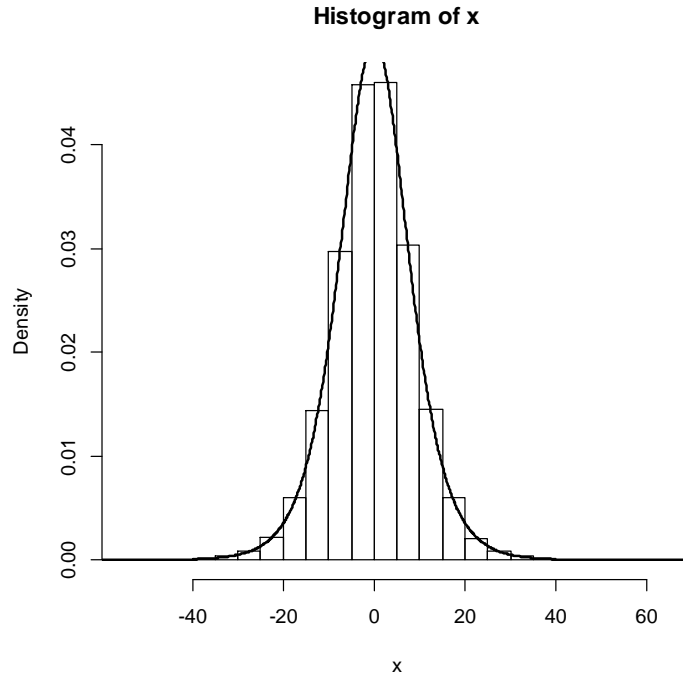
The function `Var(t, 3.2)` is a function defined in order to compute the variance from the function  $Var[x] = \int_{-\infty}^{\infty} x^2 f(x) dx - \mu^2$  where  $\mu = 0$ .

```
#Variance of the pdf
Var<- function(t, alpha){
  v=0
  v = sum(t^2*f(t, alpha))
  return(v)
}
```

Plots of distributions were obtained using the following code.

```
#Plots of denisities
x <- rF(100000,0.2)
hist(x,freq=FALSE)
t <- -1000:1000/10
lines(t,f(t,0.2),lwd=2)
```

As we see distribution of generated random variables has very similar shape as the probability density function.



3. Under the consideration is the following probability density function:

$$g(x) = \begin{cases} cx^{\alpha-1}, & 0 < x < 1, \\ ce^{-x}, & 1 \leq x, \\ 0, & \text{otherwise,} \end{cases}$$

where  $\alpha \in (0, 1)$  and  $c$  is normalizing constant.

a) In order to find cumulative distribution function we have to integrate  $g(x)$ .

Thus we have:

$$G(x) = \int_{-\infty}^x g(x)dx = \begin{cases} \int_0^1 cx^{\alpha-1}dx = c \frac{x^\alpha}{\alpha} + C_1, \\ \int_1^\infty ce^{-x}dx = -ce^{-x} + C_2, \\ \int_{-\infty}^0 0dx = C_3, \end{cases}$$

where  $C_1, C_2, C_3$  are integration constants.

Because we know that:

$$\lim_{x \rightarrow -\infty} G(x) = 0, \quad \lim_{x \rightarrow \infty} G(x) = 1 \quad \text{and} \quad G(x) \in C^1$$

we can derive that:

$$C_1 = 0, \quad C_2 = 0, \quad C_3 = \frac{c}{e} + \frac{c}{\alpha}$$

and

$$\lim_{x \rightarrow \infty} G(x) = -\frac{c}{e^\infty} + \frac{c}{e} + \frac{c}{\alpha} = c \left( \frac{1}{e} + \frac{1}{\alpha} \right) = c \frac{e + \alpha}{e\alpha} = 1$$

thus we obtain normalizing value:

$$c = \frac{e\alpha}{e + \alpha}$$

Eventually cumulative distribution function has a following form

$$G(x) = \begin{cases} \frac{e}{e + \alpha} x^\alpha, & 0 < x < 1, \\ -\frac{e\alpha}{e + \alpha} \cdot \frac{1}{e^x} + 1, & 1 \leq x, \\ 0, & x \leq 0 \end{cases}$$

In order to find the inverse of  $G(x)$  the following steps have to be performed:

for  $0 \leq x < 1$

$$x = \left( \frac{e + \alpha}{e\alpha} \right)^{1/\alpha}, \quad 0 \leq p < \frac{e}{e + \alpha}$$

for  $1 \leq x$

$$p = -\frac{e\alpha}{e + \alpha} \cdot \frac{1}{e^x} + 1$$

$$\frac{e\alpha}{e + \alpha} \cdot \frac{1}{e^x} = 1 - p \quad \left| \cdot \frac{e^x}{1 - p} \right.$$

$$\frac{e\alpha}{e + \alpha} \cdot \frac{1}{1 - p} = e^x \quad \left| \cdot \ln() \right.$$

$$x = \ln \left( \frac{e\alpha}{e + \alpha} \right) + \ln \left( \frac{1}{1 - p} \right), \quad \frac{e}{e + \alpha} \leq p < 1$$

for  $0 < x$

no event

The final form of the inverse cumulative distribution function  $G(x)^{-1}$  is:

$$G(x)^{-1} = \begin{cases} \left( \frac{e + \alpha}{e\alpha} \right)^{1/\alpha}, & 0 \leq p < \frac{e}{e + \alpha} \\ \ln \left( \frac{e\alpha}{e + \alpha} \right) + \ln \left( \frac{1}{1 - p} \right), & \frac{e}{e + \alpha} \leq p < 1 \end{cases}$$

- b) R function that generates samples from  $g(x)$  use the inverse technique. This means that random variables are generated from the inverse cumulative function where the argument is an uniformly distributed random variable  $p \sim U(0, 1)$ .

```
#Inverse cumulative distribution function
rInvertedG <- function(n, alpha){
  z = rep(NA,n)
  c = (alpha*exp(1))/(alpha+exp(1))
  for(i in 1:n){
    u <- runif(1)
    if(u < (exp(1)/(exp(1) + alpha)))
      z[i] = ((exp(1) + alpha) / exp(1) *
        u)^(1/alpha)
    else
      z[i] = log( 1/(1 - u), base = exp(1)) +
        log(exp(1) * alpha/(exp(1) + alpha),
          base = exp(1))
  }
  return(z)
}
```

In order to check if the defined function is working properly the additional function was defined. This function allows to plot pdf.

```
#Probability dneisity function of g(x)
g_pdf <- function(x, alpha){
  c = (alpha*exp(1))/(alpha+exp(1))
  (x<1)*(c*x^(alpha-1)) + (!(x<1))*c*exp(-x)
}
```

Now the results may be compared with the real value. Therefore mean value, variance value and plot of distribution are compared.

```
#Mean and variance of the vector of the generated
#samples
> mean(rInvertedG(10000, 0.7))
[1] 0.730486
> var(rInvertedG(10000, 0.7))
[1] 0.6829017
```

Now, these properties have to be compared with theoretical values. Thus formulas for mean and variance were derived.

The derivation of the mean is as follows:

$$\begin{aligned}
E[x] &= \int_0^{\infty} xg(x)dx = c \int_0^1 x \cdot x^{\alpha-1}dx + c \int_1^{\infty} xe^{-x}dx \\
&= c \left[ \int_0^1 x^{\alpha}dx - \frac{x}{e^x} \Big|_1^{\infty} + \int_1^{\infty} e^{-x}dx \right] = c \left[ \frac{x^{\alpha+1}}{\alpha+1} \Big|_0^1 - \frac{x}{e^x} \Big|_1^{\infty} - \frac{1}{e^x} \Big|_1^{\infty} \right] \\
&= c \left[ \frac{1}{\alpha+1} - \frac{x+1}{e^x} \Big|_1^{\infty} \right] = c \left[ \frac{1}{\alpha+1} - \frac{2}{e} \right] = \frac{\alpha e}{\alpha+e} \left[ \frac{1}{\alpha+1} - \frac{2}{e} \right]
\end{aligned}$$

Because:

$$c = \frac{\alpha e}{\alpha+e}, \quad \lim_{x \rightarrow \infty} \frac{x+1}{e^x} = 0 \quad (\text{De L'Hospital Rule})$$

Therefore for  $\alpha = 0.7$ , the mean value  $\mu = 0.737006$ . As we see this is close to the empiric value.

The derivation of the variance is as follows:

$$Var[x] = \int_0^{\infty} x^2 g(x)dx - \mu^2$$

where:

$$\begin{aligned}
\int_0^{\infty} x^2 g(x)dx &= c \int_0^1 x^2 \cdot x^{\alpha-1}dx + c \int_1^{\infty} x^2 e^{-x}dx \\
&= c \left[ \int_0^1 x^{\alpha+1}dx + \int_1^{\infty} x^2 e^{-x}dx \right]
\end{aligned}$$

where:

$$\begin{aligned}
\int x^2 e^{-x}dx &= -x^2 e^{-x} + 2 \int x e^{-x}dx = -\frac{x^2}{e^x} + 2 \left( -\frac{x}{e^x} + \int e^{-x}dx \right) \\
&= -\frac{x^2}{e^x} + 2 \left( -\frac{x}{e^x} - \frac{1}{e^x} \right) = -\frac{x^2 + x + 2}{e^x}
\end{aligned}$$

Therefore:

$$\begin{aligned}
\int_0^{\infty} x^2 g(x)dx &= c \left[ \frac{x^{\alpha+2}}{\alpha+2} \Big|_0^1 - \frac{x^2 + x + 2}{e^x} \Big|_1^{\infty} \right] \\
&= c \left[ \frac{1}{1+\alpha} - \left( 0 - \frac{1+2+2}{e} \right) \right] = \frac{\alpha e}{\alpha+e} \left[ \frac{1}{\alpha+1} + \frac{5}{e} \right] \\
Var[x] &= \frac{\alpha e}{\alpha+e} \left[ \frac{1}{\alpha+1} + \frac{5}{e} \right] - \mu^2
\end{aligned}$$

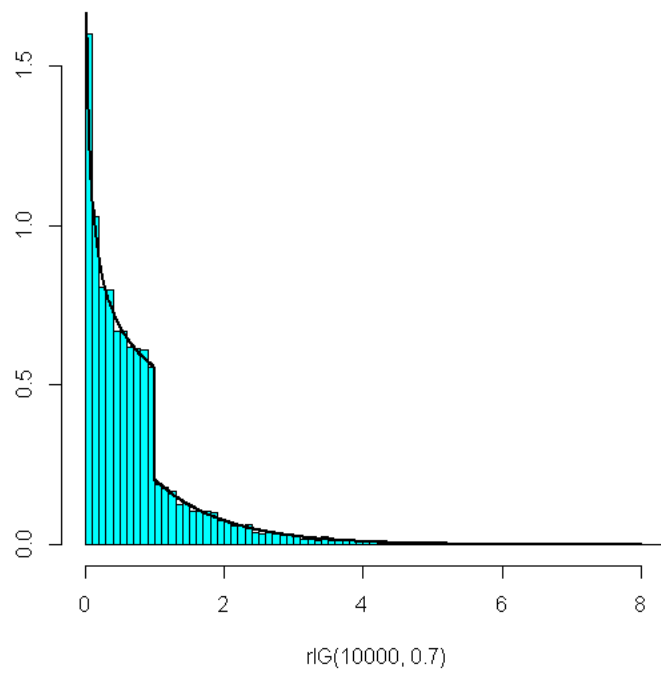
For  $\alpha = 0.7$ , the variance value  $\sigma^2 = 0.686897$ . As we see this is close to the empiric value.

Plots of distributions were obtained using the following code.



```
#Plots of denisities
library(MASS)
truehist(rInvertedG(10000, 0.7))
t <- 0:800/100
lines(t, g_pdf(t, 0.7), lwd=2)
```

As we see distribution of generated random variables has very similar shape as the probability density function.



## Problem B: Rejection sampling and importance sampling

1. A rejection sampling algorithm to simulate from  $f(\theta|y)$  using the  $U(0,1)$  density as the proposal density.

The first step before function is coded is to find scaling coefficient that allow to get the highest acceptance probability as possible for particular proposal density. For this reason I checked the maximum value of the target function using the R function optimize.

```
> optimize(function(x) (2+x)^125*(1-x)^(18+20)*x^34,
lower=0, upper=1, maximum= T)
$maximum
[1] 0.6268101
$objective
[1] 1.838839e+29
```

From now on the in every acceptance step coefficient  $c = 1.838839e + 29$  is taken into account. For such value uniform distribution times factor  $c$  is tangent witch target pdf. The code is enclosed bellow.

```
#Rajection sampling algorithm
RejectionSampling <- function(n){
  y <- numeric(n)
  #optimal coefficient
  c <- 1.838839e+29
  accepted.samples <- 0
  #acceptance step
  while(accepted.samples < n){ #while loop suffice
    u <- runif(1)
    if(c* runif(1) < MassFunction(u)){
      accepted.samples <- accepted.samples+1
      y[accepted.samples] <- u
    }
  }
  return(y)
}
```

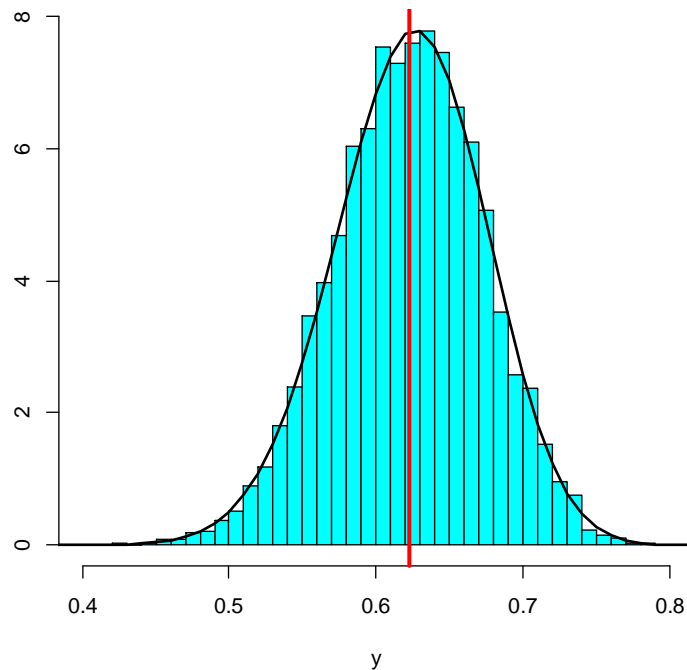
2. In order to estimate posterior mean of  $\theta$  the rejection function defined in the previous point is used. The estimation is made basing on the 10000 samples. Bellow the code and the results are shown.

```
> y <- RejectionSampling(10000)
> mean(y)
[1] 0.623168
> var(y)
[1] 0.002569723
```

In order to check real values the target function  $f(\theta|\mathbf{y})$ , R function `integrate` and equations for mean  $E[x] = \int_{-\infty}^{\infty} xf(x)dx$  and variance  $Var[x] = \int_{-\infty}^{\infty} x^2f(x)dx - \mu^2$  are used. Look at the code enclosed bellow.

```
> integrate(function(x)x*(2+x)^125*(1-
x)^(18+20)*x^34/const, lower=0, upper=1)
0.6228062 with absolute error < 1.7e-07
> integrate(function(x)(x-0.6228062 )^2*(2+x)^125*(1-
x)^(18+20)*x^34/const, lower=0, upper=1)
0.002594921 with absolute error < 5e-08
```

As we see empirical results are proved by the theoretical values. Additionally the shapes of the target and sampled distributions are compared (red line is estimated posterior mean).



The plot was generated using the following code.

```
const <- 2.357695e+28 #normalizing constant
y <- RejectionSampling(10000)
truehist(y)
t <- 0:100/100
# MassFunction(t) is a target function f(teta|y)
lines(t, MassFunction(t)/const,lwd = 2)
abline(v = mean(y), col = "red", lwd = 3)
```

The normalizing constant `const` and `MassFunction` are derived from the following code.

```
#Definition of the target function f(teta|y)
MassFunction <- function(x){
  x <- (2+x)^125*(1-x)^(18+20)*x^34
  return(x)
}

#Inverted normalizing constant given as follows
> integrate(function(x) (2+x)^125*(1-x)^(18+20)*x^34,
lower=0, upper=1)
2.357695e+28 with absolute error < 1.8e+21
```

3. The amount of random numbers that we need to generate on average in order to obtain one sample of  $f(\theta|y)$  might be calculated acceptance probability formula.

$$P(c \cdot g(\theta) \leq f(\theta|y)) = \int_0^1 \frac{f(\theta|y)}{c \cdot g(\theta)} g(\theta) d\theta = \int_0^1 \frac{f(\theta|y)}{c} d\theta$$

Because as it was proved in the previous point the target function integrate to:

$$\int_0^1 f(\theta|y) d\theta = 2.357695e + 28$$

We have that

$$\int_0^1 \frac{f(\theta|y)}{c} d\theta = \frac{2.357695e + 28}{1.838839e + 29} \approx 0.128$$

Because single trials are independent the number of trials up to the first success is geometrically distributed with parameter  $1 / \int_0^1 \frac{f(\theta|y)}{c} d\theta$ . Thus the expected number of trials up to the first success is  $\frac{1}{0.128} \approx 7.813$ . Since it has to be integer we round up and eventually we have that the 8 trials have to be performed to get one acceptance.

4. In order to use importance sampling weights to estimate posterior mean under the new prior based on the samples from part (2) the following equations have to be implemented:

$$\hat{E}[\theta] = \frac{1}{\sum_{l=1}^L w_l} \sum_{l=1}^L \theta w_l$$

where

$$w_l = \frac{f(\theta^{(l)})}{g(\theta^{(l)})} - \text{so called importance weight.}$$

Bearing in mind the fact that the new prior  $g(\theta)$  has Beta distribution  $Be(2,5)$  the R code implementation of the new estimator has the following form.

```
> y <- RejectionSampling(10000)      #samples form part (2)
> mean(y)
[1] 0.623081                        #mean under the old Be(1,1) prior
#importance smapling weights
#new prioir Be(2,5) instead of Be(1,1)
> w <- MassFunction(y)/dbeta(y,2,5)
> sum <- sum(w)
> posterior_mean <- 1/sum*sum(y*w)
> posterior_mean
[1] 0.6369494                      #mean under the new prior Be(2,5)
```

As we see the new posterior estimator doe not differ a lot from the previous one for the uniform prior.

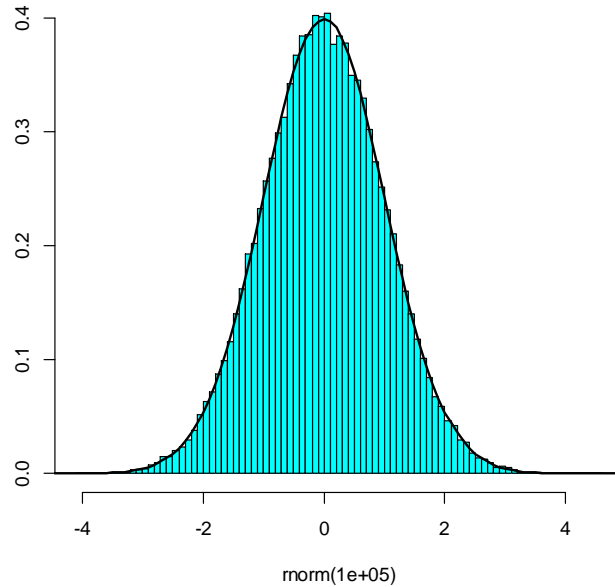
### Problem C: Stochastic simulation by bivariate techniques and rejection sampling

1. An R function that uses Box-Muller algorithm to generate a vector of n independent samples from the standard normal distribution.

```
#Box-Muller algorithm
rnorm1 <- function(n){
  u = runif(n)
  v = runif(n)
  x=rep(0,n)
  for (i in 1:n){
    x[i] = sqrt(-2*log(u[i]))*cos(2*pi*v[i])
  }
}
#Ploting routine
truehist(rnorm(100000))
t <- -100:100/10
lines(t,dnorm(t,mean=0,sd=1),lwd=2)

> mean(rnorm(100000))
[1] 0.004150459
> var(rnorm(100000))
[1] 0.9988284
```

As we see plot, mean value and variance value confirm the correctness of the implemented method.



2. Under the consideration is the gamma distribution function with parameters  $\alpha \in (0,1)$  and  $\beta = 1$ :

$$f(x) = \begin{cases} \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x}, & 0 < x, \\ 0, & \text{otherwise.} \end{cases}$$

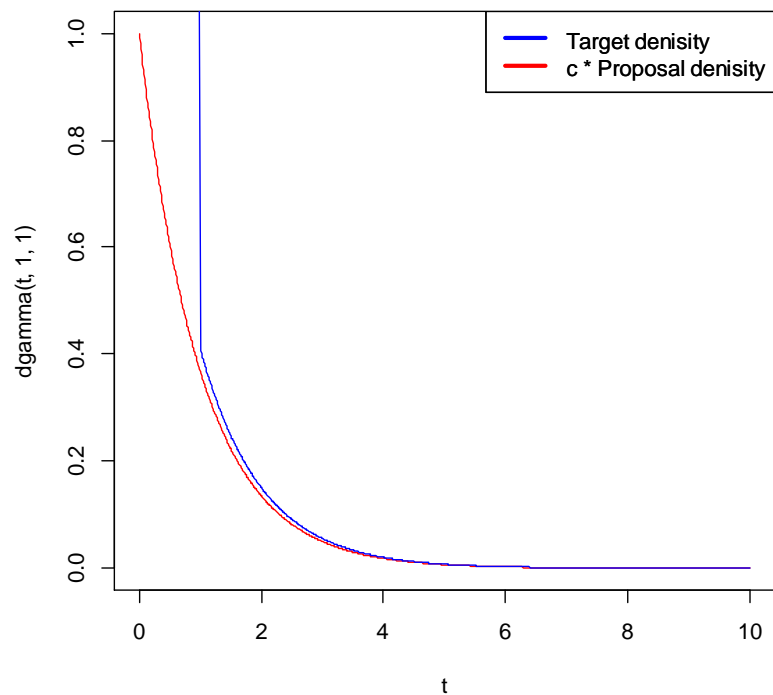
Here rejection sampling is used to generate from this distribution by proposing samples from the function  $g(x)$  that was considered in Problem A3.

- a) Expression for the acceptance probability in the rejection sampling algorithm.

In order to find acceptance probability experiments for different settings  $\alpha \in (0,1)$  and  $c \geq 1$  were performed. After number of them, the one that seems to be relevant was chosen. This is  $\alpha = 0.7$  and  $c = 2$ .

Compared curves of both densities and code, that was used to obtain those curves, are enclosed bellow.

```
#visualization of target and proposal densities
c <- 2 #c
t <- 0:1000/100
plot (t,dgamma(t,1,1), type="l", col="red")
lines(t, c*g_pdf(t, 0.7), col="blue", lwd=1)
legend("topright", lty = c(1,1),lwd=c(3,3),
      col=c("blue","red"), c("Target denisity", "c *
      Proposal denisity"))
```



b) R function that generates a vector of  $n$  independent samples  $f$ .

To do this rejection sampling was used. The R function has the following implementation.

```
#Rejection sampling from Gamma
rGammaReject <- function(n, alpha){
  c <- 2 #chosen c coefficient
  y <- numeric(n)
  accepted.samples <- 0
  while (accepted.samples < n) {
    x <- rInvertedG(1, 0.7)
    f.at.x <- dgamma(x, alpha, 1) #target dist.
    g.at.x <- g_pdf(x, 0.7)       #proposal dist.
    accept <- f.at.x / (c*g.at.x)
    if (runif(1) < accept) {
      accepted.samples <- accepted.samples + 1
      y[accepted.samples] <- x
    }
  }
  return(y)
}
```

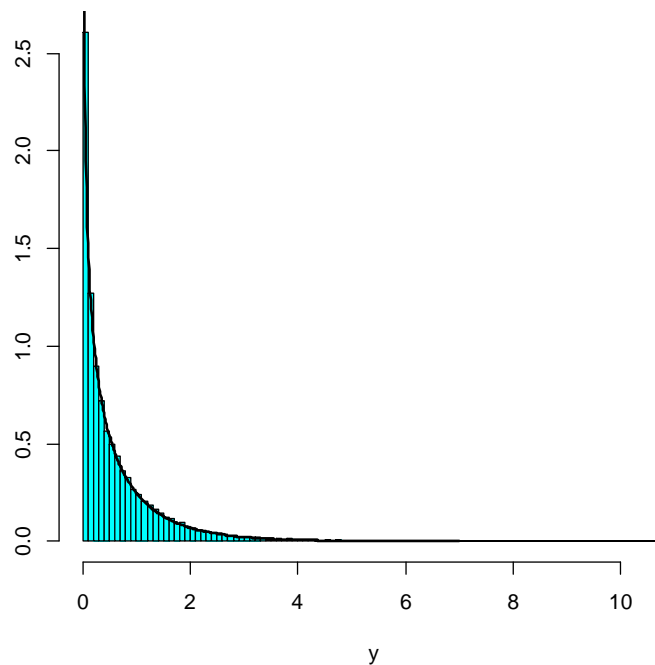
Now the results may be compared with the real values. Therefore mean value, variance value and plot of distribution are compared.

```
y <- rGammaReject(30000, 0.6)
> mean(y)
[1] 0.6010811
> var(y)
[1] 0.5978176
```

Because for  $Gamma(\alpha, \beta)$  has the following properties:  $E[x] = \frac{\alpha}{\beta} = \frac{0.6}{1} = 0.6$  and  $Var[x] = \frac{\alpha}{\beta^2} = \frac{0.6}{1} = 0.6$ , sampled density fits to the real  $Gamma(\alpha, \beta)$  pdf.

It might be observed on the plot, that is enclosed below the code implementation which is responsible for the plotting routines.

```
#Test - comparition
y <- rGammaReject(30000, 0.6)
truehist(y)
t <- 0:700/100
lines(t, dgamma(t, 0.6, 1), lwd=2)
```





3. Under the consideration is the gamma distribution function with parameters  $\alpha > 1$  and  $\beta = 1$ :

$$f(x) = \begin{cases} \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x}, & 0 < x, \\ 0, & \text{otherwise.} \end{cases}$$

Here the ratio of uniform method is used to simulate from this distribution.

- a) In order to find the values  $a = \sqrt{\sup f^*(x)}$ ,  $b_+ = \sqrt{\sup (x^2 f^*(x))}$  and  $b_- = -\sqrt{\sup (x^2 f^*(x))}$  the differentials of these functions were get. Thus we have:
- $$f^*(x)' = (x^{\alpha-1} e^{-x})' = (\alpha - 1)x^{\alpha-2} e^{-x} - x^{\alpha-1} e^{-x} = \frac{1}{e^x} ((\alpha - 1)x^{\alpha-2} - x^{\alpha-1})$$

$$\frac{1}{e^x} ((\alpha - 1)x^{\alpha-2} - x^{\alpha-1}) = 0$$

$$\ln(\alpha - 1) = \ln(x)(\alpha - 1 - \alpha + 2)$$

$$x = \alpha - 1$$

It is sufficient since  $f^*(\alpha - 1)'' < 0$ .

Now I am looking for  $b_+$ .

$$(x^2 f^*(x))' = (x^{\alpha+1} e^{-x})' = \frac{1}{e^x} ((\alpha + 1)x^\alpha - x^{\alpha+1})$$

$$\ln(\alpha + 1) = \ln(x)(\alpha + 1 - \alpha) \Rightarrow x = \alpha + 1$$

It is sufficient since  $(x^2 f^*(x))'' > 0$

Because  $b_-$  exists only for  $x = 0$  we have that:  $b_- = 0$

Summarizing:

$$\begin{aligned} a &= \left( \frac{\alpha - 1}{e} \right)^{\frac{\alpha-1}{2}} \\ b_+ &= \left( \frac{\alpha + 1}{e} \right)^{\frac{\alpha+1}{2}} \\ b_- &= 0 \end{aligned}$$

- b) R function that generates a vector of  $n$  independent samples from  $Gamma(\alpha, 1), \alpha > 1$ .

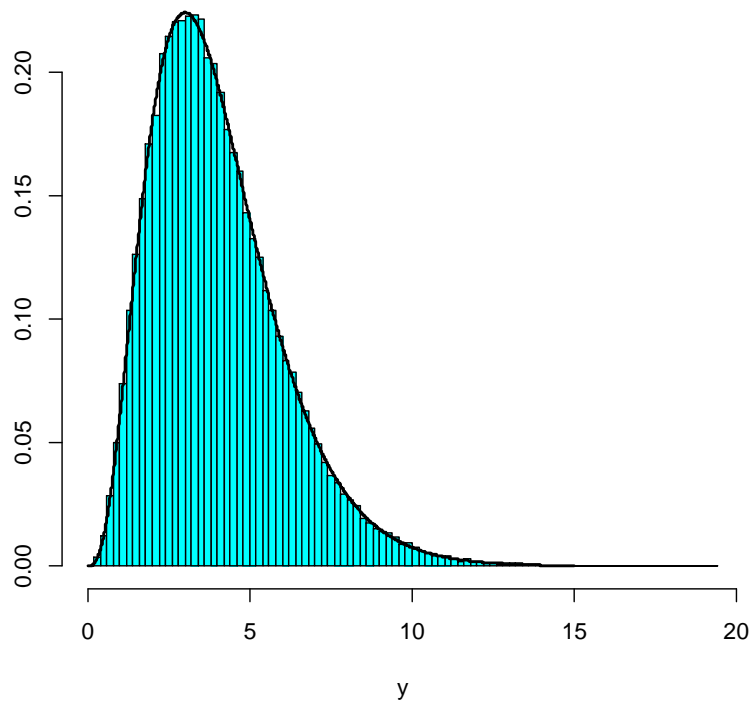
```
#Ratio of uniform method for Gamma
rGamma3 <- function(n, alpha){
  y <- numeric(n)
  accepted.samples <- 0
  a <- ((alpha - 1)/exp(1))^(0.5*alpha - 0.5)
  b_plus <- ((alpha+1)/exp(1))^(0.5*alpha +0.5)
  while(accepted.samples < n){
    u <- a*runif(1)
    v <- b_plus*runif(1)
    x <- v/u
    accepted <- (alpha - 1)*log(x) - x
    if(2*log(u) < accepted){
      accepted.samples <- accepted.samples + 1
      y[accepted.samples] <- x
    }
  }
  return(y)
}
```

Now the results may be compared with the real values. Therefore mean value, variance value and plot of distribution are compared.

```
y <- rGamma3(100000, 4)
> mean(y)
[1] 4.000726
> var(y)
[1] 4.044997
```

Because for  $Gamma(\alpha, \beta)$  has the following properties:  $E[x] = \frac{\alpha}{\beta} = \frac{4}{1} = 4$  and  $Var[x] = \frac{\alpha}{\beta^2} = \frac{4}{1} = 4$ , sampled density fits to the real  $Gamma(\alpha, \beta)$  pdf. It might be observed on the plot, that is enclosed below the code implementation which is responsible for the plotting routines.

```
#Test - comparition
library(MASS)
y <- rGamma3(100000, 4)
truehist(y)
t <- 0:1500/100
lines(t, dgamma(t, 4, 1), lwd=2)
```



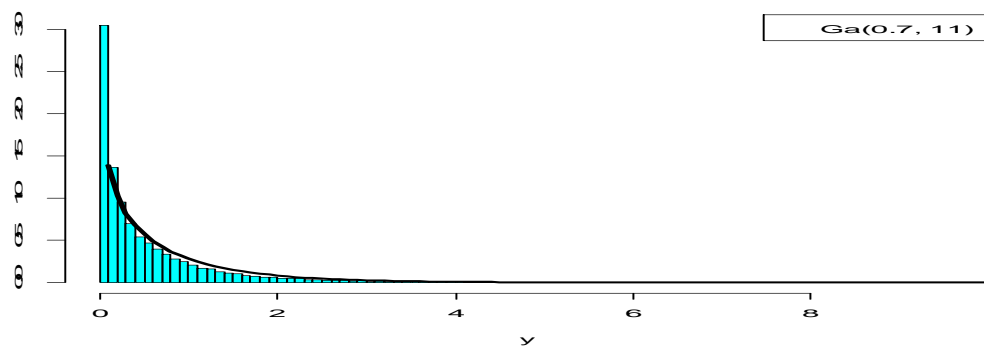
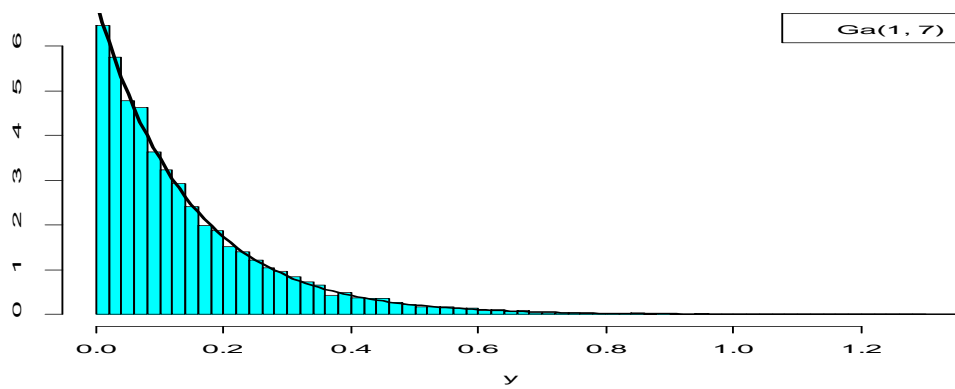
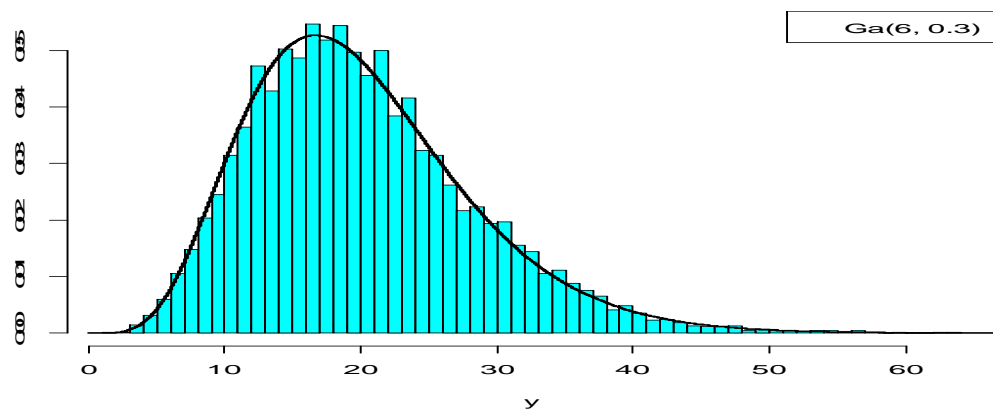
4. R function that generates a vector of  $n$  independent samples from a gamma distribution with parameters  $\alpha > 0$  i  $\beta > 0$ .

In this task I used “if else” function and three generators of Gamma distributions that were used in previous points of the assignment. Three conditions are considered:  $\alpha \in (0,1)$ ,  $\alpha = 1$  and  $\alpha > 1$ .  $\beta$  is an inverse parameter where  $y = \beta x \sim Ga(\alpha, \beta)$  if  $x \sim Ga(\alpha, 1)$ .

For  $\alpha = 1$  and  $\beta = 1$ , gamma probability distribution function  $Ga(\alpha, \beta)$  has the same shape as Exponential pdf  $E(\lambda)$  for  $\lambda=1$ . The R code of the function is enclosed bellow.

```
#Gamma distribution (alpha > 0, beta > 0)
rGammaGeneral <- function(n, alpha, beta){
  smaple <- numeric(n)
  if(alpha > 0){
    if(alpha < 1) smaple <- 1/beta*rGammal(n, alpha)
    else if(alpha > 1) smaple <- 1/beta*rGamma2(n, alpha)
    else sample <- 1/beta*rExponent(n)
  }
  return(smaple)
}
```

Different combinations of  $\alpha$  and  $\beta$  are enclosed bellow.



## Problem D: Multivariate distributions

1. An R function that generates one realization from a d-variate distribution with given mean vector  $\mu$  and covariance matrix  $\Sigma$ .

Multivariate normal distribution may be simply generated from the standard normal distribution. The transformation has following form:

for  $x \sim N(0, I)$  we have that  $y = \mu + Ax \Rightarrow y \sim N(\mu, AA^T)$

Thus if we choose  $A$  so that  $AA^T = \Sigma$  the distribution may be generated. In order to obtain  $A$  the Cholesky decomposition is used. The R function generating one realization together with mean and variance values is enclosed below.

```
#Multivariate normal distribution
MVrnorm <- function(n, mu, sigma) {
  ncols <- ncol(sigma)
  m <- rep(mu, each = n)
  m + matrix(rnorm(n * ncols), ncol = ncols) %*%
    chol(sigma) #Cholesky decomposition
}

sigma <- matrix(c(1, -0.2, 0.7, -0.2, 1, -0.1, 0.7, -0.1,
1), ncol = 3)
mu <- c( 7, -12, 4)

#Properties
Y <- MVrnorm(100000, mu, sigma)
> cor(y)
      [,1]      [,2]      [,3]
[1,] 1.0000000 -0.1999523  0.7030889
[2,] -0.1999523  1.0000000 -0.1028494
[3,]  0.7030889 -0.1028494  1.0000000
> colMeans(y)
[1]  6.997459 -12.000054  4.001704
> #comparation of integrate multivariate normal
  distribution in R
> x = mvrnorm(100000, mu, sigma)
> var(x)
      [,1]      [,2]      [,3]
[1,] 0.9985284 -0.2009629  0.6972806
[2,] -0.2009629  0.9961674 -0.1007321
[3,]  0.6972806 -0.1007321  0.9929015
> colMeans(x)
[1]  6.996474 -11.993420  3.998335
```

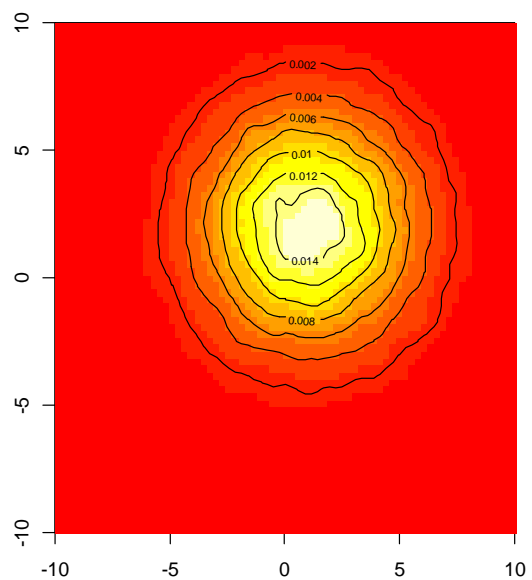
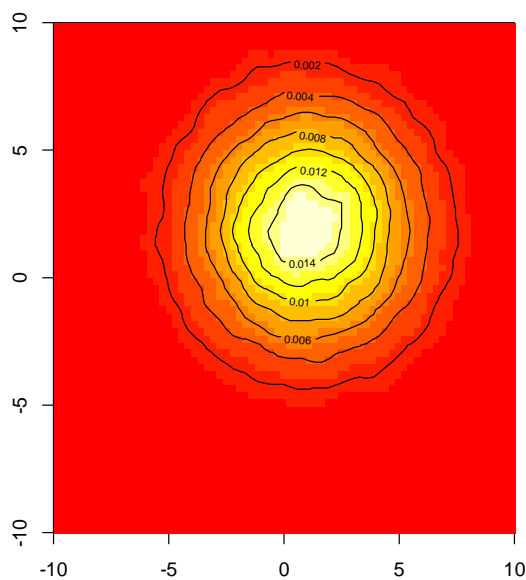
To validate the code, R multivariate normal distributed random variables generator was used. As we see characteristics does not differ significantly.

In order to make the results more palpable, 2-D plots together with the code are enclosed bellow.

```
sigma <- matrix(c(10,0,0,10),2,2)
mu <- c(1,2)
y <- MVRnorm(100000, mu, sigma) #Self defined
x <- mvrnorm(100000, mu, sigma)  #R function

#Plotting routine
library(gplots)
y.kde = kde2d(y[,1], y[,2], n=100)
x.kde = kde2d(x[,1], x[,2], n=100)

par(mfrow=c(1,2))
image(y.kde, xlim=c(-10,10), ylim=c(-10,10))
contour(y.kde, add = T)
image(x.kde, xlim=c(-10,10), ylim=c(-10,10))
contour(x.kde, add = T)
```



- a) “Assume  $z_k \sim \text{Gamma}(\alpha_k, 1)$  for  $k = 1, \dots, K$  independently, and define  $x_k = \frac{z_k}{z_1 + \dots + z_K}$  for  $k = 1, \dots, K$ . Show that then  $x = (x_1, \dots, x_K)$  has a Dirichlet distribution with parameter vector  $\alpha = (\alpha_1, \dots, \alpha_K)$ ”.

It might be done by incorporating the change-of-variables formula and transformation of the original variables  $(z_1, \dots, z_K)$  to  $(x_1, \dots, x_{K-1}, v)$ , where  $v = z_1 + \dots + z_K$ .

$$f_y(y_1, y_2) = f_x(g^{-1}(y_1, y_2)) \cdot \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \dots & \frac{\partial x_m}{\partial y_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial y_n} & \dots & \frac{\partial x_m}{\partial y_n} \end{vmatrix}$$

Bearing in mind that  $z_k = x_k v$  for  $k = 1, \dots, K-1$  and  $z_K = v x_K = v(1 - x_1 - \dots - x_{K-1})$ , the Jacobian of the transformation is as follows:

$$J = \begin{pmatrix} v & 0 & \dots & x_1 \\ 0 & v & \dots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ -v & \dots & -v & x_K \end{pmatrix}$$

Adding the first  $K-1$  rows to the last one, upper diagonal matrix is obtained. Because diagonal elements are  $(v, \dots, v, 1)$  we have that  $\det|J| = v^{K-1}$ . Then pdf of transformed variables  $(x_1, \dots, x_{K-1}, v)$  might be expressed by the equation:

$$f(x_1, \dots, x_{K-1}, v) = v^{K-1} \sum_{k=1}^K \frac{(x_k v)^{\alpha_k - 1} e^{-x_k v}}{\Gamma(\alpha_k)} = v^{\sum \alpha_k - 1} e^{-v} \prod_{k=1}^K \frac{x_k^{\alpha_k - 1}}{\Gamma(\alpha_k)}$$

In order to obtain the probability density function of Dirichlet distributed random variables the previous expression has to be integrated with respect to  $v$ . Thus we have:

$$f(x_1, \dots, x_{K-1}) = \prod_{k=1}^K \frac{x_k^{\alpha_k - 1}}{\Gamma(\alpha_k)} \cdot \int_0^\infty v^{\sum \alpha_k - 1} e^{-v} dv = \Gamma(\sum_{k=1}^K \alpha_k) \cdot \prod_{k=1}^K \frac{x_k^{\alpha_k - 1}}{\Gamma(\alpha_k)}$$

which is the same formula as Dirichlet pdf with parameter vector  $\alpha = (\alpha_1, \dots, \alpha_K)$

$$f(x_1, \dots, x_{K-1}) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\Gamma(\alpha_1) \cdot \dots \cdot \Gamma(\alpha_K)} \cdot x_1^{\alpha_1 - 1} \cdot \dots \cdot x_{K-1}^{\alpha_{K-1} - 1} \cdot (1 - \sum_{k=1}^{K-1} x_k)^{\alpha_K - 1}$$

for  $x_1, \dots, x_k > 0$  and  $\sum_{k=1}^{K-1} x_k < 1$

- b) An R function that generates one representation from a Dirichlet distribution with parameter vector  $\alpha \in (\alpha_1, \dots, \alpha_K)$ .

As was proved in the previous point Dirichlet distributed random variables might be derived from Gamma distribution. The way to do so is to use once again the same transformation:  $x_k = \frac{z_k}{z_1 + \dots + z_K} \sim \text{Dir}(\alpha)$  where  $z_k \sim \text{Gamma}(\alpha_k, 1)$ ,  $k = 1, \dots, K$ .

Therefore the function `rGammaGeneral` defined in the problem C4 is used here with constant parameter  $\beta$ . The R code with Dirichlet function, that takes as the parameter vector  $\alpha \in (\alpha_1, \dots, \alpha_K)$ , is enclosed below.

```
#Direchlet dist.
rdirichlet1 = function(n, alpha) {
  k = length(alpha)
  r = matrix(0, nrow=n, ncol=k)
  for (i in 1:k) {
    r[,i] = rGammaGeneral(n, alpha[i], 2)
  }
  r <- matrix(mapply(function(r, s) {return (r/s)}), r,
    rowSums(r)), ncol=k)
  return(r)
}
```

Now the results may be compared with the real values. Therefore mean value, variance value and plot of distribution are compared.

```
> r <- rdirichlet1(10000, c(1,2,3))
> mean(r[,1])
[1] 0.1641375
> mean(r[,2])
[1] 0.3325303
> mean(r[,3])
[1] 0.5033322
> var(r)
      [,1]      [,2]      [,3]
[1,] 0.019733167 -0.007561047 -0.01217212
[2,] -0.007561047 0.031175994 -0.02361495
[3,] -0.012172120 -0.023614947 0.03578707
```

The real values are calculated manually using the following equations:

$$E[X_i] = \frac{\alpha_i}{\sum_k \alpha_k}; \quad \text{Var}[X_i] = \frac{\alpha_0(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}; \quad \text{Cov}[X_i, X_j] = \frac{-\alpha_i \alpha_j}{\alpha_0^2(\alpha_0 + 1)}; \quad \alpha_0 = \sum_{i=1}^K \alpha_i.$$

$$E[X_1] \approx 0.167; \quad E[X_2] \approx 0.33; \quad E[X_3] = 0.5;$$



$$\text{Var}[X_1] \approx 0.0198; \quad \text{Var}[X_2] \approx 0.0317; \quad \text{Var}[X_3] \approx 0.0357;$$

$$\text{Cov}[X_1, X_2] \approx -0.0079; \quad \text{Cov}[X_1, X_3] \approx -0.0119; \quad \text{Cov}[X_2, X_3] \approx -0.0238$$

As we see the parameters of the  $\text{Dir}(1,2,3)$  distribution coincide. Additionally some plot are attached bellow.

