Exercise 1

Problem A: Stochastic simulation by the probabilty integral transform and bivariate techniques

1.

We are going to generate samples from an exponential distribution with rate parameter λ, and the number of samples is n. Let $X \sim Exp(\lambda)$. This gives pdf and cdf

$$f(x) = \lambda \exp(-\lambda x)$$

$$F(x) = 1 - \exp(-\lambda x)$$

The inversion method can be used for generate samples from the exponential distribution. First random variable U is generated from the standard uniform distribution in interval $[0, 1]$. Then $X = F^{-1}(U)$. The algorithm is then

$$u \sim U[0,1]$$

$$x = -\frac{1}{\lambda}\log(u)$$

$$\text{return } x$$

```
generate_exponential <- function(n, lambda)
{
  u <-runif(n)
  x <- -(1/lambda) * log(u)
  return(x)
}
```

Below the function is used with $n = 10000$ and $\lambda = 3$. The result is plotted against the theoretical distribution.

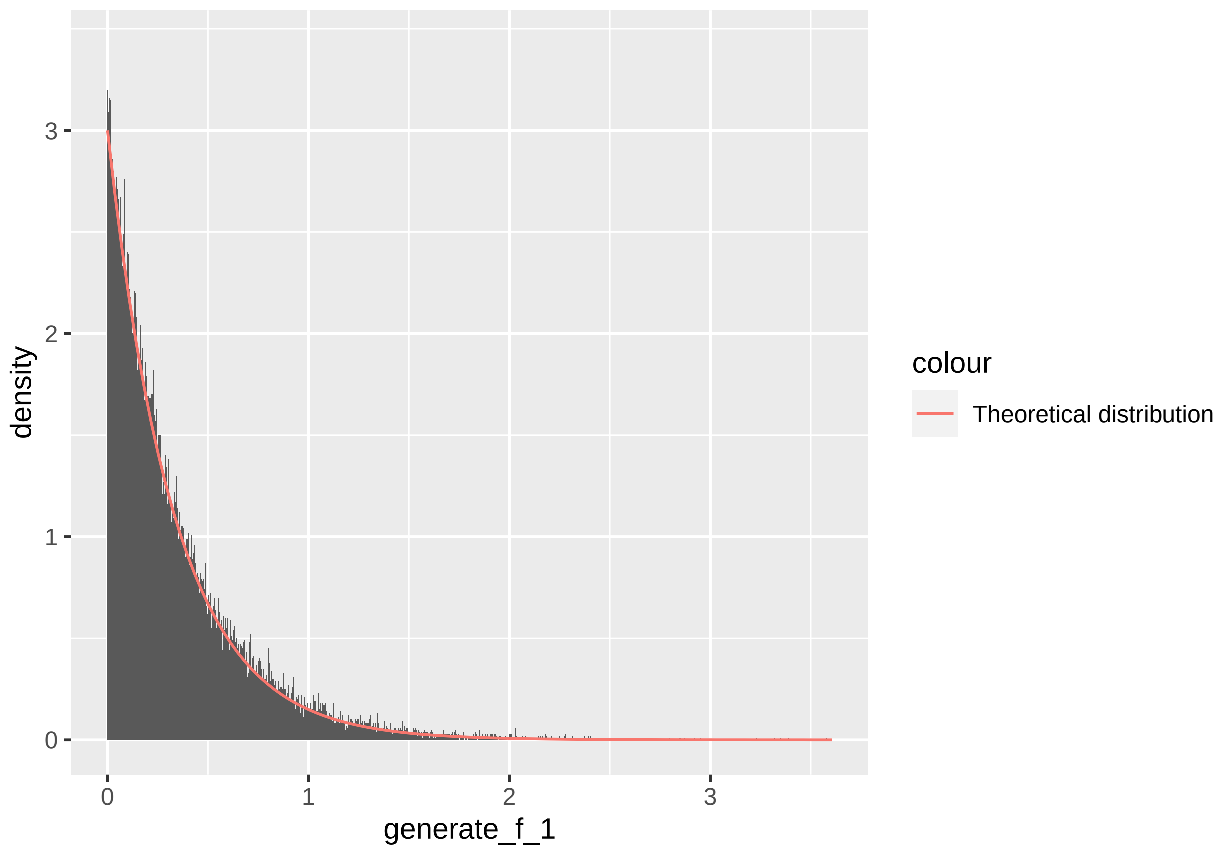
```
library(ggplot2)

theoretical_f_1 <-function(x,lambda)
{
  return(lambda*exp(-lambda*x))
}
set.seed(2)
n=100000
lambda=3
generate_f_1=generate_exponential(n, lambda)


ggplot()+
  geom_histogram(
    data=as.data.frame(generate_f_1),
```

```
    mapping=aes(x=generate_f_1,y=..density..),
    binwidth=0.001

  )+
  stat_function(
  fun=theoretical_f_1,
  args=list(lambda=lambda),
  aes(col="Theoretical distribution")
)
```



2

a)

We want to find the cumulative distribution function and the inverse of the cumulative distribution function when the probability density function is

$$g(x) =$$

The cumulative distribution $G(x)$ is given by

$$G(x) = \int_{-\infty}^{x} g(t)dt$$

Thus, for $0 < x < 1$, the cdf becomes

$$G(x) = \int_0^x ct^{\alpha-1} = c \cdot \left[\frac{1}{\alpha}t^\alpha\right]_0^x = \frac{c}{\alpha}x^\alpha.$$

For $1 \leq x$, the cdf is given by

$$G(x) = \int_0^1 ct^{\alpha-1} + \int_1^x ce^{-t}dt = c \cdot \left[\frac{1}{\alpha}t^\alpha\right]_0^1 + c \cdot \left[-e^{-t}\right]_1^x = \frac{c}{\alpha} - ce^{-x} + e^{-1} = c \cdot \left(\frac{1}{\alpha} - e^{-x} + \frac{1}{e}\right)$$

The constant c can be found by solving the following equation for c,

$$\begin{aligned}
\int_0^1 ct^{\alpha-1} + \int_1^\infty ce^{-t}dt = 1 \\
\frac{c}{\alpha} + c \cdot \left[-e^{-t}\right]_1^\infty = \frac{c}{\alpha} + \frac{c}{e} \\
\frac{c}{\alpha} + \frac{c}{e} = 1 \quad \Longrightarrow c = \frac{\alpha e}{e+\alpha}.
\end{aligned}$$

The inverse of this cumulative distribution function can be found by solving the following equation for x,

$$y = G(x).$$

For $0 < x < 1$, we have

$$y = \frac{ex^\alpha}{e+\alpha} \implies y(e+\alpha) = ex^\alpha \implies x = \left(\frac{u(e+\alpha)}{e}\right)^{\frac{1}{\alpha}}$$

Thus, the inverse of the cumulative distribution function is

$$G^{-1}(y) = \left(\frac{y(e+\alpha)}{e}\right)^{\frac{1}{\alpha}}$$

for $0 < G^{-1}(y) < 1 \implies 0 < y < \frac{e}{e+\alpha}$. For $1 \leq x$, the following equation is solved for x

$$\begin{aligned}
y = 1 - \frac{\alpha e^{-x+1}}{e+\alpha} &\implies x = 1 - \ln\left(\frac{(1-y)(e+\alpha)}{\alpha}\right) = \ln\left(\frac{\alpha e}{(1-y)(e+\alpha)}\right) \\
&\implies G^{-1}(y) = \ln\left(\frac{\alpha e}{(1-y)(e+\alpha)}\right)
\end{aligned}$$

For $x = 1$, we have

$$1 = \ln\left(\frac{e\alpha}{(1-y)(e+\alpha)}\right) \implies e = \frac{e\alpha}{(1-y)(e+\alpha)} \implies y = 1 - \frac{\alpha}{\alpha+e}$$

When $x = \infty$, $y = 1$. Therefore the inverse cumulative function is

b

The inversion method is used to generate samples from g.

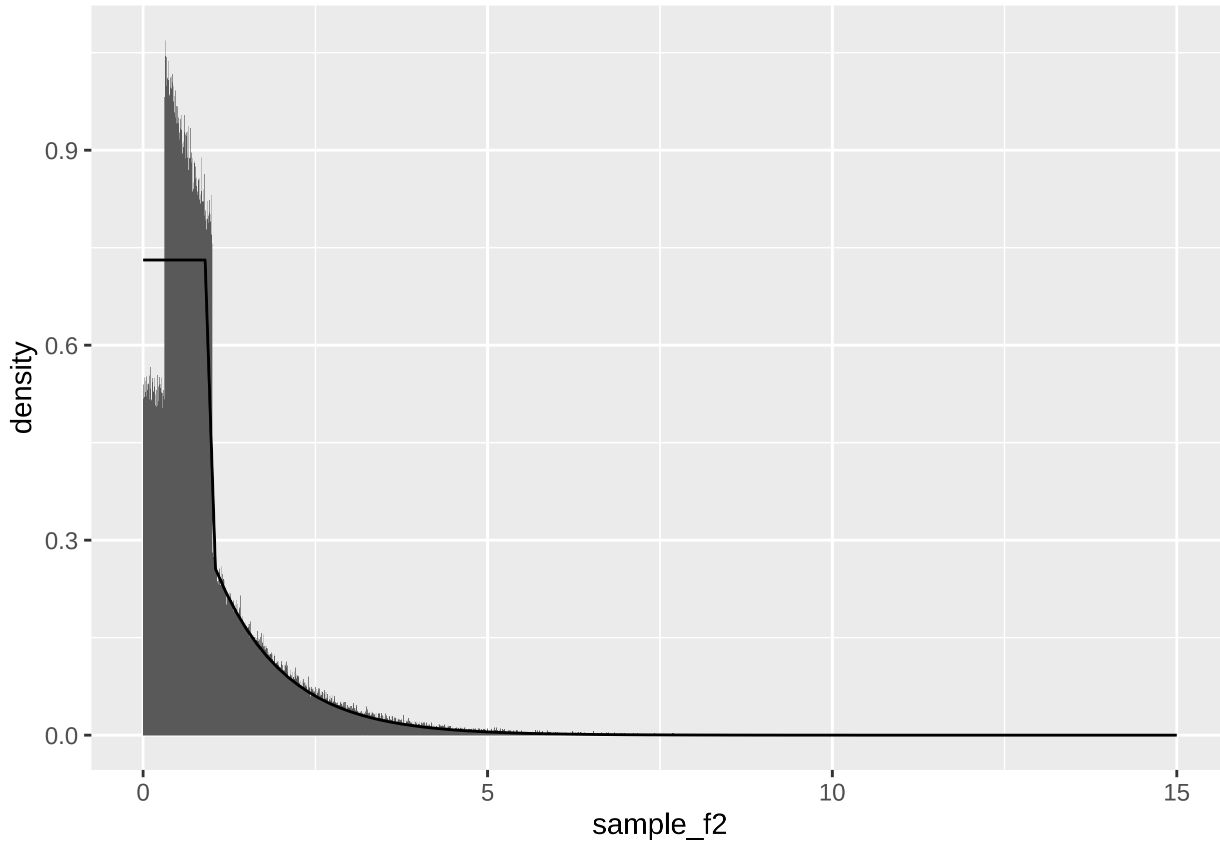
```r
generate_function <-function(n, alpha)
{
  u1<-runif(n/2,0,exp(1)/(alpha+exp(1)))
  u2<-runif(n/2,1-(exp(1)/(alpha+exp(1))),1)
  x1<-(u1*((alpha+exp(1)))/exp(1))^(1/alpha)
  x2<-log((alpha+exp(1))/((1-u2)*(alpha+exp(1))))
  x<-append(x1,x2)
}
```

The result for $\alpha = 2$ and $n = 10000$ is plotted against the theoretical distribution.

```r
theo_gx <- function(x, alpha) {
const <- alpha * exp(1) / (alpha + exp(1)) # Normalizing constant
func <- rep(0, length(x)) # Vector of zeros of same length as x
left <- x > 0 & x < 1 # The PDF has one value for 0 < x < 1
right <- x >= 1 # ... and one value for x >= 1
func[left] <- const * x[left]^(alpha - 1) # The value to the left
func[right] <- const * exp(-x[right]) # The value to the right
return(func)
}


n=1000000
alpha=1
sample_f2<-generate_function(n,alpha)
ggplot()+
  geom_histogram(
    data=as.data.frame(sample_f2),
    mapping=aes(x=sample_f2,y=..density..),
    binwidth=0.001

  )+
  stat_function(
  fun=theo_gx,
  args=list(alpha=alpha)
)
```



3.)

We consider the probability density function

$$f(x) = \frac{ce^{\alpha x}}{(1+e^{\alpha x})^2}, \quad \infty < x < \infty, \quad \alpha > 0$$

a) To find the normalizing constant, we consider the integral I of the pdf over R

$$I = \int_{-\infty}^{\infty} f(x)dx = 1 \implies \int_{-\infty}^{\infty} \frac{ce^{\alpha x}}{(1+e^{\alpha x})^2} dx = 1$$

Let $u = 1 + e^{\alpha x}$. This means that $\frac{du}{dx} = \alpha e^{\alpha x}$ and $u(-\infty) = 1$ and $u(\infty) = \infty$. By using variable change the integral becomes

$$I = \int_{1}^{\infty} \frac{c}{\alpha} u^{-2} du = \frac{c}{\alpha} \left[-u^{-1}\right]_1^{\infty} = \frac{c}{\alpha}$$

$$I = 1 \implies \frac{c}{\alpha} = 1 \implies c = \alpha$$

The pdf is therefore

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

The cumulative distribution function is given by

$$F(x) = \int_{-\infty}^{x} \frac{\alpha e^{\alpha t}}{(1+e^{\alpha t})^2} dt.$$

By using $u = 1 + e^{\alpha t}$, we get

$$F(x) = \int_{-\infty}^{1+e^{\alpha x}} u^{-2} du = \left[-u^{-1}\right]_1^{1+e^{\alpha x}} = \frac{-1}{1+e^{\alpha x}} + 1 = \frac{e^{\alpha x}}{1+e^{\alpha x}}$$

The inverse cumulative distribution is found by solving $y = F(x)$ for x.

$$y = \frac{e^{\alpha x}}{1+e^{\alpha x}} \implies e^{\alpha x} = \frac{y}{1-y} \implies x = \frac{1}{\alpha} \ln\left(\frac{y}{1-y}\right)$$

This means that the inverse cumulative distribution function is

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

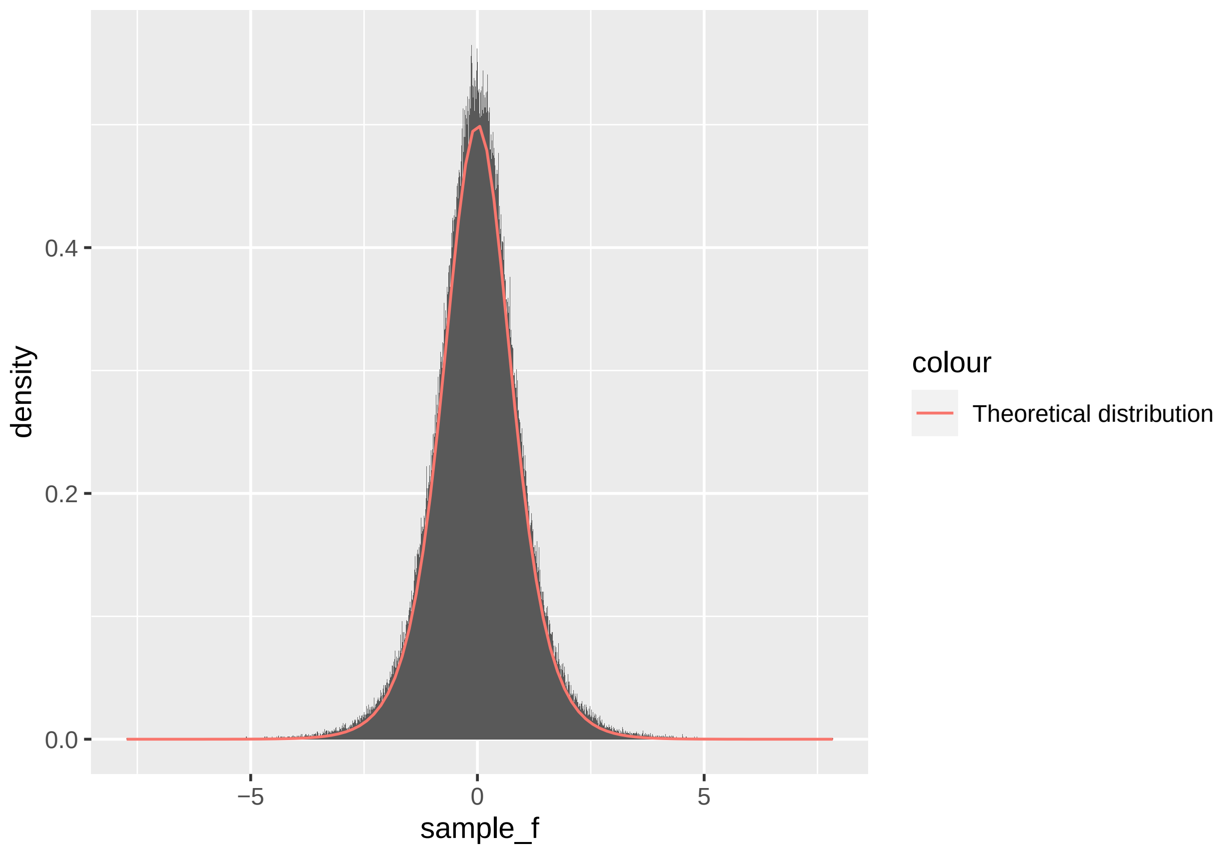
c) In the following chunk there is code for a function generating samples from f by using the inversion method

```
generate_f <- function(n, alpha)
{
  u<-runif(n)
  x<-(1/alpha)*log(u/(1-u))
  return(x)
}
```

To check that the function works properly, an example with using the function $\alpha = 2$ and $n = 1000000$ is plotted against the theoretical distribution.

```
theoretical_f <- function(x, alpha) {
return(alpha * exp(alpha * x) / (1 + exp(alpha * x))^2)
}
```

```
library(ggplot2)
n=1000000
alpha=2
sample_f <- generate_f(n,alpha)
ggplot()+
  geom_histogram(
    data=as.data.frame(sample_f),
    mapping=aes(x=sample_f,y=..density..),
    binwidth=0.001,
  ) +
  stat_function(
    fun=theoretical_f,
    args=list(alpha=alpha),
    aes(col="Theoretical distribution")
  )
```



4.

We use the Box-Muller algorithm to represent independent variables which are standard normal distributed. Let $X \sim N(0,1)$ and $Y \sim N(0,1)$ be independent. The joint distribution of these two variables is

$$f(x,y) = f_X(x) \cdot f_Y(y) = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}}$$

By using polar coordinates where $x^2 + y^2 = r^2$, the joint distribution becomes

$$f(r) = \frac{1}{2\pi} e^{-\frac{r^2}{2}}.$$

This is a joint distribution of $r^2 \sim \exp(1/2)$ and $X_1 \sim \text{Unif}(0, 2\pi)$. This means that

$$X = r\cos(X_1)$$

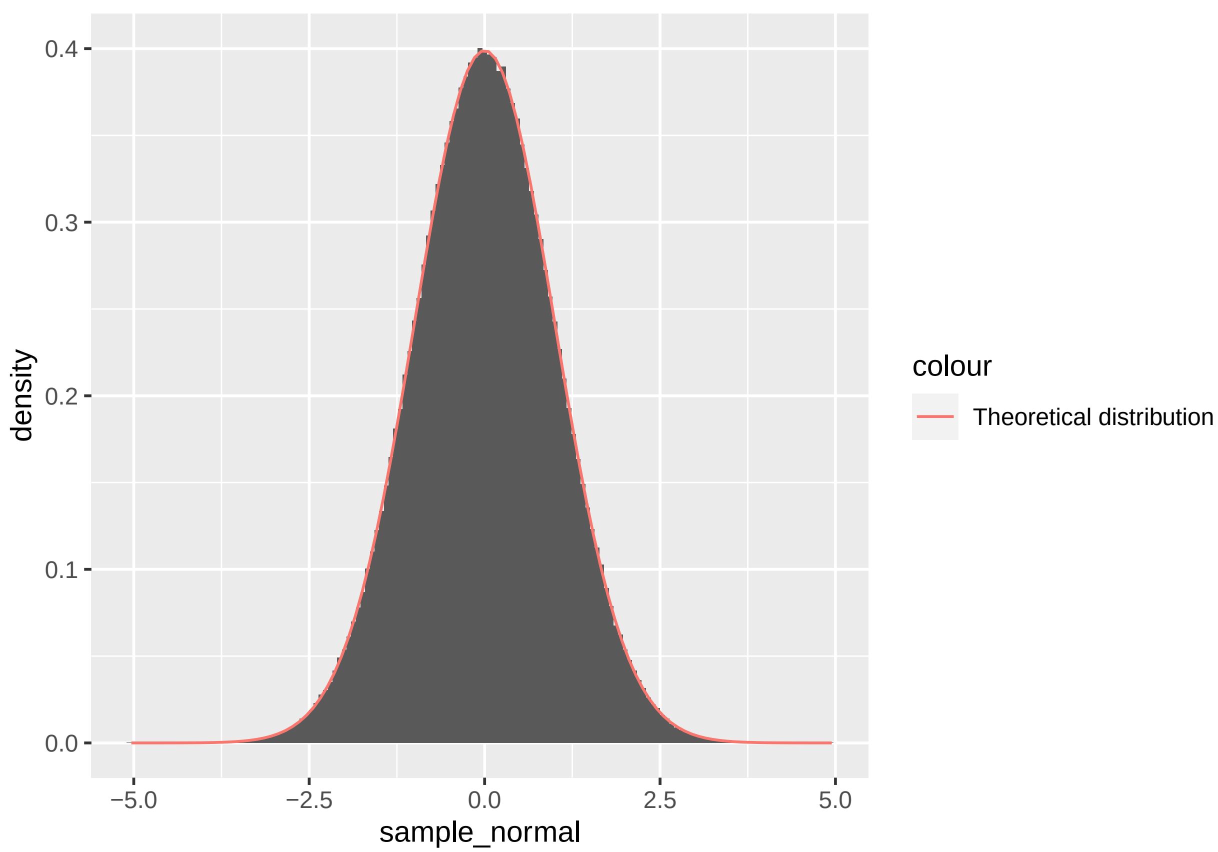
$$Y = r\sin(X_1)$$

are normal distributed. $r \sim \sqrt{-2\log(\text{Unif}(0,1))}$ and $X_2 \sim 2\pi\text{Unif}(0,1)$

In the following chunk, the Box-Muller algorithm is implemented. We first draw two samples from $\text{Unif}(0,1)$. We then calculate r and X_1 and at last return $X = r\cos(X_1)$ which is standard normal distributed.

```
generate_from_normal <- function(n)
{
  u1<-runif(n)
  u2<-runif(n)
```

```
  r<-sqrt(-2*log(u1))
  x_1<-2*pi*u2
  x=r*cos(x_1)
  return(x)
}
n=1000000
sample_normal<-generate_from_normal(n)
ggplot()+
  geom_histogram(
    data=as.data.frame(sample_normal),
    mapping=aes(x=sample_normal, y=..density..),
    binwidth=10/150
  )+
  stat_function(
    fun=dnorm,
    args=list(mean=0,sd=1),
    aes(col="Theoretical distribution")
  )
```



5. We want to to simulate from a d-variate normal distribution with mean vector μ and covariance matrix Σ. We use the function form the previous task to simulate form the standard normal distribution. Let $Z \sim \text{Normal}(0, I_d)$, where I_d is the identity matrix. Then

$$X = \mu + DZ \sim Normal(\mu, DD^T)$$

Have to find D such that $\Sigma = DD^T$.

```
generate_d_normal <-function(n,mu,cov,d)
{
  D<-t(chol(cov))
  z<-generate_from_normal(d*n)
  Z<-matrix(z.nrow=d,ncol=n)
  X<-mu+D %*% Z
  return(X)
}
```

To test wheter the function works, we use $\mu = [1, 7, 2]^T$ and Σ

```
n<-10000
mu<-c(3,4,5)
cov_mat<-cbind(c(1,2,3), c(2,3,4), c(3,4,5))
cov_mat
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    2    3    4
## [3,]    3    4    5
```

```
# sample_normal_d<-generate_d_normal(n,mu,cov_mat,3)
```

Problem B: The gamma distribution

1.

The gamma distribution with parameters $\alpha \in (0, 1)$ and $\beta = 1$ has probability density function

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

a) The acceptance probability γ in the rejection sampling algorithm is given by

$$\gamma = \frac{1}{c} \cdot \frac{f(x)}{g(x)}.$$

where $g(x)$ is the proposal density. We use the density in problem A.2.

Problem C: Monte Carlo integration and variance reduction

We want to use Monte Carlo integration to find $\theta = P(X > 4)$ when $X \sim N(0, 1)$ 1. The parameter θ is estimated by using $n = 100000$ samples from the standard normal distribution. The inverse cumulative distribution function for the standard normal distribution is

```r
generate_from_normal<-function(n)
{

}
```