Project 1
Computational Numeric Statistics

João Camacho, N. 56861, Analysis and Engineering of Big Data
Ana Mendes, N. 57144, Analysis and Engineering of Big Data
Lia Schmid , N. 57629, Cienceas Cognitivas
Emanuele Vivoli, N. 57284, Engineering Informatics
Simão Gonçalves, N. 54896, Analysis and Engineering of Big Data

CONTENTS

I. EXERCISE 1 - BOX-MULLER METHOD

The Box-Muller method is a method for generating samples from two independent random variables X, Y $\sim N_1(0,1)$.

A. Derivation of Box-Muller Method

Let's consider X, Y where each follow $N_1(0,1)$. If we plot (X,Y) on the Cartesian Plane as a point, we can represent that in polar coordinates.

We will first consider the distance, R, between the point (X,Y) and the origin as in [1]. We know from geometry that $R^2 = X^2 + Y^2$. The sum between two squared standard Normal variables is distributed as $\chi^2_{df=2}$. It is also a known that

$$\chi^2_{df=2} \overset{d}{\sim} Gamma(1, \frac{1}{2})$$

which is itself equivalent to $Exp(0.5)$. Furthermore, using the Inverse Transform Method, we can sample from $e \sim Exp(\frac{1}{2})$ using the $u \sim U(0,1)$ through $e = -2log(u)$.

Now, if we take the joint distribution of X,Y we have:

$$\begin{aligned} f(x,y) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} \\ &= \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)} \end{aligned} \tag{1}$$

To determine the joint density of R^2 and θ, $g(R,\theta)$ we make the change of variables:

$$\begin{cases} R^2 = x^2 + y^2 \\ \theta = tan^{-1}(\frac{y}{x}) \end{cases}$$

As shown in [2]:

$$\begin{aligned} f(x,y) &= |J| g(R^2, \theta) \\ &= |J| g(x^2+y^2, tan^{-1}(\frac{y}{x})) \end{aligned} \tag{2}$$

where $|J|$ is the absolute value of the Jacobian and because of (2) we can determine $g(R^2,\theta)$ as follow:

$$g(R^2,\theta) = \frac{1}{|J|} f(x,y) \tag{3}$$

We need the term from the Jacobian which is the absolute value of the determinant of partial derivatives of R^2 and θ with respect to x and y.

$$J = \begin{vmatrix} \frac{\partial R^2(x,y)}{\partial x} & \frac{\partial R^2(x,y)}{\partial y} \\ \frac{\partial \theta(x,y)}{\partial x} & \frac{\partial \theta(x,y)}{\partial x} \end{vmatrix}$$

This result is easily achieved and returns $J = 2$:

$$J = \begin{vmatrix} 2x & 2y \\ -\frac{y}{x^2+y^2} & \frac{x}{x^2+y^2} \end{vmatrix} = \frac{2x^2}{x^2+y^2} + \frac{2y^2}{x^2+y^2} = 2 \quad (4)$$

Because of (1), (3) and (4) the joint density $g(R^2, \theta)$ is thus given by:

$$g(R^2, \theta) = \frac{1}{2}\frac{1}{2\pi}e^{-\frac{1}{2}R^2} \quad (5)$$

This result is the product of the exponential density (having mean 2), $\frac{1}{2}e^{-\frac{1}{2}R^2}$, and the uniform density (defined over $[0, 2\pi]$). In other words, once we have the radius *R* of the coordinates, the angle θ is uniformly distributed over the circumference with radius *R*. Thus, θ can be any angle as long as it is uniformly distributed. So we can take:

$$\theta = 2\pi U_2 \quad , \quad U_2 \sim U(0,1) \quad (6)$$

Putting all these results together, we have:

$$\begin{cases} R^2 = -2log(u) \Leftrightarrow R = \sqrt{-2log(u_1)} \\ \theta = 2\pi u_2 \end{cases} \quad (7)$$

$$\begin{cases} X = Rcos(\theta) \\ Y = Rsin(\theta) \end{cases}$$

$$\begin{cases} X = \sqrt{-2log(u_1)}cos(2\pi u_2) \\ Y = \sqrt{-2log(u_1)}sin(2\pi u_2) \end{cases}$$

And with this we can generate samples from two independent Normal distributions.

B. Exercise

We are asked to do 3 tasks

1) Create the Box-Muller algorithm in R
2) Generate a sample of size 10 000 of a N(0,4)
3) Plot the histogram of the samples with the true $p.d.f_{N(0,4)}$ superimposed.

1) Create the Box-Muller algorithm in R: The following is the box muller algorithm implemented in R:

Listing 1. Box-Muller algorithm in R

```
sim.norm<-function(n, mu, std){

  x<-vector()
  y<-vector()

  # Box-Muller algorithm
  theta=2*pi*runif(n,0,1)
  R=sqrt(-2*log(runif(n,0,1)))

  # add ~N(0,1) observations to the samples
  x=c(x,R*cos(theta))
  y=c(y,R*sin(theta))

  # convert all observations from N(0,1)
  # to N(mu, std)
  x=x*std+mu
  y=y*std+mu

  # return results in a table format
  samples = matrix(c(x,y),ncol=2)
  colnames(samples) = c("X","Y")
  rownames(samples) = seq(from=1,
  to=nrow(samples),by=1)
  samples = as.table(samples)
  return(samples)
}
```

Note that the Box-Muller method generates samples from N(0,1) but we want to generate samples from the N(0,4). To solve this, we used a property of linear combinations. If we take:

$$y = ax + b \quad , \quad x \sim N(\mu, \sigma)$$

The new mean of the distribution will be $a \cdot \mu + b$ and the new σ will be $\sigma + b$ since:

$$\begin{cases} E[Y] = E[a \cdot X + b] = aE[X] + E[b] = a \cdot \mu + b \\ Var[Y] = Var[a \cdot X + b] = a^2 * Var[X] = a^2 \cdot \sigma^2 \\ \sigma[Y] = \sqrt{Var[Y]} = a \cdot \sigma \end{cases}$$

Furthermore, Y is also normally distributed since its p.d.f. keeps the same form as the Normal Distribution:

$$p.d.f._Y = \frac{1}{\sigma^* \cdot \sqrt{2\pi}}e^{-\frac{1}{2}(\frac{y-\mu^*}{\sigma^*})^2} \quad (8)$$

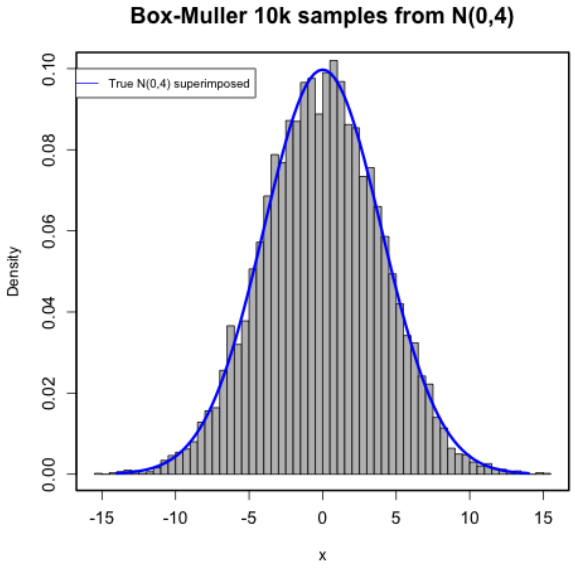
Where σ^* and μ^* are the new values after applying the transformation $y = a \cdot x + b$
Therefore, from the standard Normal Distribution we can easily find a transform that converts it to a N(μ^*,σ^*), by taking the transform:

$$y = \sigma^* \cdot X + \mu^*$$

2) Generate a sample of size 10 000 of a N(0,4): We only need to run our algorithm for 5000 iterations as we can merge the samples from X and Y since they are iid random variables. With this, the code histogram of the generated samples from the Box-Muller method is found below:

Listing 2. Box-Muller: generation of 10k samples in R

```
set.seed(123)
mu = 0
std = 4
n = 10000
samples = sim.norm(n/2,0,4)
samples = c(samples[,"X"], samples[,"Y"])
```



II. Exercise 2 - Acceptance-Rejection Method

A. Description of Acceptance-rejection method

Let's suppose we want to generate independent samples from a probability distribution $f(x)$ which is difficult to sample. We can pick another distribution from which we can more easily sample called the *candidate density function*, Y with pdf $g(y)$, and then reject observations that are unlikely under the *target density function* $f(x)$.

B. Derivation of Acception Rejection

Let's take a constant, M, that satisfies the following condition:

$$\frac{f(x)}{g(x)} \leq M \tag{9}$$

M represents the maximum of the function $h(x) = \frac{f(x)}{g(x)}$. This means that $f(x) \leq Mg(x)$. If we generate a sample, x_c, from g(x), the probability of accepting it as a sample from f(x) has to be proportional to the probability of generating that same sample from f(x).
An intuitive explanation for this is that if the sample x_c generated from g(x) had, for example, a high pdf value in $g(x_c)$, but $f(x)$ happened to have a much lower pdf value on that point, then we have to accept x_c with a low probability. That same probability of accepting is actually α defined as below:

$$\frac{1}{M}\frac{f(x_c)}{g(x_c)} \tag{10}$$

If $Mg(x_c)$ is very close to $f(x_c)$ then it is very likely that x_c will be accepted as a sample from $f(x)$. Otherwise, the probability of accepting will be low.

C. Exercise

In this exercise we will:

1) create a function in R for generating a sample from f using the acceptance-rejection method;
2) generate a sample of size 10 000 of f;
3) plot the histogram of the samples with the true p.d.f superimposed;
4) display hit-and-miss plot.

Let X be a continuous random variable with probability density function (p.d.f.) proportional to $f(x) = xe^{-x}$, $x > 0$
The function f is a probability density function if

$$F(x) = \int_{-\infty}^{+\infty} f(x)dx = 1$$

In this case

$$\int_0^{+\infty} xe^{-x}dx \tag{11}$$

Using definite integrals by parts

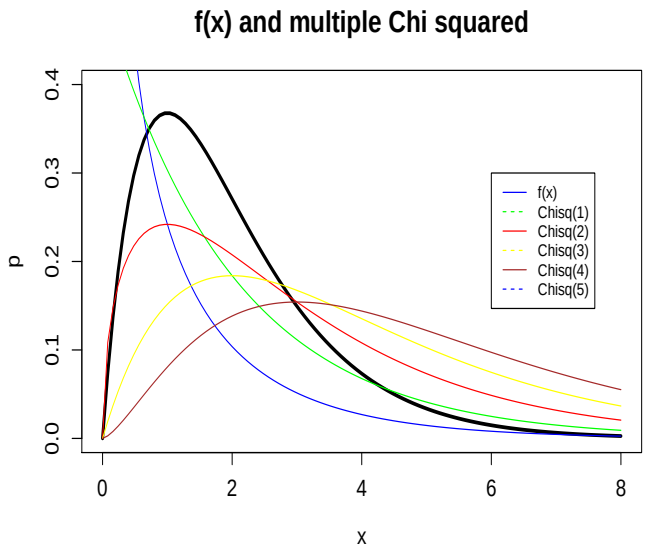
$$\int_a^b udv = uv\Big|_a^b - \int_a^b vdu \tag{12}$$

Let $u = x$ and $v' = e^{-x}$ then $u' = 1$ and $v = -e^{-x}$

$$\begin{aligned}\int_0^{+\infty} f(x)dx &= -xe^{-x}\Big|_0^{+\infty} - \int_0^{+\infty} -e^{-x}dx \\ &= -xe^{-x}\Big|_0^{+\infty} - \left(-e^{-x}\right)\Big|_0^{+\infty} \\ &= \lim_{x\to\infty} -xe^{-x} - \left[\lim_{x\to\infty} -e^{-x} - (e^0)\right] \\ &= 0 - (0 - 1) = 1\end{aligned}$$

Therefore, although the question suggested that X had a probability density function proportional to f(x), we found that f(x) has an integral equal to exactly 1 over the domain $x > 0$, therefore, the pdf of X is equal to f(x).

The support of our random variable X is the interval $[0, +\infty[$, the work sheet also suggests that we choose a χ^2_n as the candidate density function, which satisfies the support. To choose the χ^2 which best approximates the form of the distribution of $f(x)$, we plotted multiple χ^2 pdfs against $f(x)$:



From the plot above, we choose χ^2_3 (red) as the candidate density function $g(x)$ because its form is the most similar to that of $f(x)$ (black), and that similarity allows the algorithm to minimize the rejection rate.

$$g(x) = \frac{1}{\Gamma(\frac{3}{2})2^{\frac{3}{2}}} e^{-\frac{x}{2}} x^{\frac{3}{2}-1} = \frac{1}{\Gamma(1+\frac{1}{2})2\sqrt{2}} e^{-\frac{x}{2}} x^{\frac{1}{2}}$$

$$\overset{\Gamma(x+1)=x\Gamma(x)}{=} \frac{1}{\frac{1}{2}\Gamma(\frac{1}{2})2\sqrt{2}} e^{-\frac{x}{2}} x^{\frac{1}{2}} \overset{\Gamma(\frac{1}{2})=\sqrt{\pi}}{=} \frac{1}{\sqrt{\pi}\sqrt{2}} e^{-\frac{x}{2}} x^{\frac{1}{2}}$$

$$h(x) = \frac{f(x)}{g(x)} = \frac{xe^{-x}}{\frac{1}{\sqrt{\pi}\sqrt{2}} e^{-\frac{x}{2}} x^{\frac{1}{2}}} =$$

$$= \frac{\sqrt{2}\sqrt{\pi} e^{\frac{x}{2}} x}{e^x x^{\frac{1}{2}}} = \sqrt{2}\sqrt{\pi} e^{-\frac{x}{2}} x^{\frac{1}{2}}$$

To determine the minimum value M such that $\frac{f(x)}{g(x)} \leq M$, is equivalent to determine the maximum value of $\frac{f(x)}{g(x)}$ and take M equal to that maximum. Furthermore, because the factor $\sqrt{2\pi}$ does not affect the search for local extremes we omit it in the derivation of the maximum.

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

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

Now solving $h^{*'}(x) = 0$.:

$$h^{*'}(x) = 0 \Leftrightarrow \frac{1}{2} e^{-\frac{x}{2}} x^{\frac{1}{2}} (-1 + x^{-1}) = 0$$

$$\Leftrightarrow \underbrace{e^{-\frac{x}{2}} = 0}_{I} \vee \underbrace{x^{\frac{1}{2}} = 0}_{II} \vee \underbrace{(-1 + x^{-1}) = 0}_{III}$$

$I : e^{-\frac{x}{2}} = 0$: no solution
$II : x^{\frac{1}{2}} = 0 \Leftrightarrow x = 0$
$III : -1 + x^{-1} \Leftrightarrow x = 1$

Since only values for x where $x > 0$ are permitted for the exponential function, we just have to check whether there really is a local extreme at $x = 1$. Therefore, we consider the sign change:
$h^{*'}_{x \to 1_{x<1}}(x) > 0$ and $h^{*'}_{x \to 1_{x>1}}(x) < 0$

This means that the function has a local minimum at $x = 1$.
Now to get the value for M, we just have to compute:

$$M = h(1) = \sqrt{2\pi} e^{-\frac{1}{2}} 1^{\frac{1}{2}} \approx 1.52$$

Consequently, P('accepting candidate sample x_c') = $\alpha(x_c)$, with $\alpha(x_c)$ defined as

$$\alpha(x_c) = \frac{1}{M} \frac{f(x_c)}{g(x_c)} = \frac{1}{1.52} \sqrt{2\pi} e^{-\frac{x_c}{2}} x_c^{\frac{1}{2}} \qquad (13)$$

We implemented in R the function `sim()` that generates a sample of f with the following algorithm:

Step 1: Generate a candidate observation x_c from g, for this we will use the following result:

Let Z_1, Z_2, Z_3 be 3 independent random variables such that $Z_i \sim N(0,1)$. Then

$$\sum_{i=1}^{3} Z_i^2 \sim \chi^2_3$$

Step 2: Compute the probability of accepting x_c as

$$\alpha = \frac{1}{1.52} h(x_c)$$

Step 3: Generate an observation u from a U(0,1) distribution.
Step 4: If $u \leq \alpha$ set $x = x_c$. Otherwise go back to step 1.
Step 5: Repeat previous steps until you reach the desired sample size.

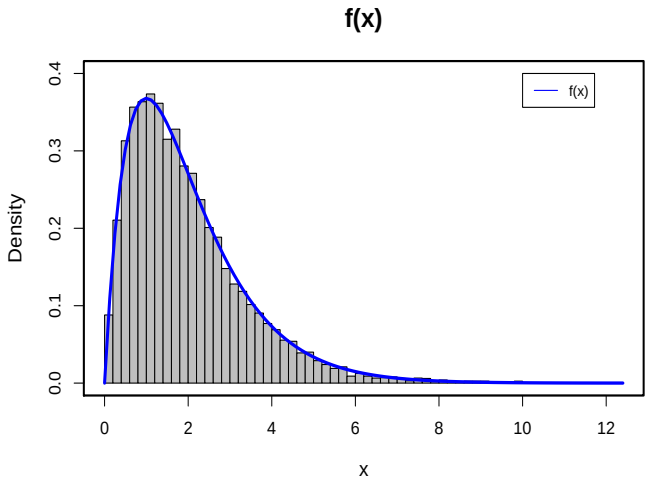
Listing 3. Acceptance-Rejection Method in R

```
sim<-function(n){
  x<-vector()
  yx <- vector()
  #rejected candidates
  rej_x<- vector()
  # y pos of rejected candidates
  yrejx<- vector()
  for (i in 1:n){
    u<-1
    alpha<-0
    k<-0
    while(u>alpha){
      if(k!=0){
        rej_x <- c(rej_x,xc);
        yrejx <- c(yrejx,u*M*g(xc) )
      }
```

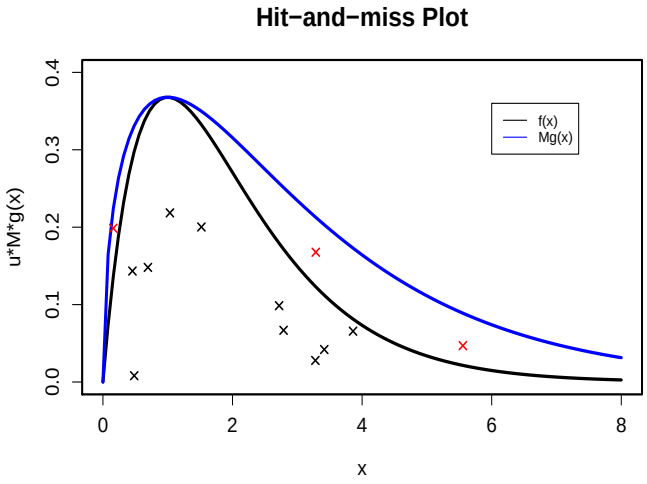
```
        xc<-sim.quisquared(1,3)
        alpha=(1/M)*h(xc)
        u<-runif(1,0,1)
        k <- k+1
      }
      x=c(x,xc)
      yx <- c(yx,u*M*g(xc))
    }
return(list(x=x, rej_x=rej_x, yx=yx,
      yrejx=yrejx))
}

set.seed(123)
fl=sim(10000)
ff=fl$x
rejection_rate= length(fl$rej_x)/
      (length(fl$rej_x)+length(fl$x))
```

As seen above, we also calculated the rejection rate which is the ratio between the number of rejections and the total number of observations that were generated. In our case, for a simulation that generates a sample of size 10000, the rejection rate obtained was 33.45%. The histogram of the generated samples is the following:



Lastly, we did a new simulation to generate a sample of size 10 and created the hit-and-miss plot, where the accepted points are below the f(x) pdf, and the rejected are colored as red, above the f(x) pdf.



III. Exercise 3 - Monte Carlo Integration

Monte Carlo integration refers to a numerical integration that uses properties of random variables. It is technique for approximating complicated and even analytically impossible to solve integral calculations.

A. Theory (Generalized MC integration)

If X is a continuous random variable with $p.d.f.$ function f and $g : \mathbb{R} \to \mathbb{R}$ is continuous, then $Y = g(X)$ is a random variable and:

$$E[Y] = E[g(X)] = \int_{D_X} g(x)f(x)dx \tag{14}$$

From equation (14) comes the basic idea of the Monte Carlo integration method. Since probabilities and expectations can in fact be described as integrals, it is quite immediate how the Monte Carlo method for ordinary integrals extends into probability theory. Since g and X are respectively function and random variable as previously defined, note that calculating the expected value of $g(X)$ is actually equivalent to computing $\mathcal{I}$ for a suitable choice of the g function:

$$\mathcal{I} = \int_{D_X} g(x)f(x)dx = E[g(X)]$$

and the sample mean is an unbiased estimator of the expectation, so Monte Carlo method approximates the integration as follows:

$$\mathcal{I}_{MC} = \frac{1}{m}\sum_{i=1}^{m} g(x_i) \approx E[g(X)]$$

B. Theory (Uniform approach MC integration)

Suppose that we want to integrate the one-dimensional function $g(x)$ in the interval $[a, b]$ as follow:

$$\mathcal{I} = \int_a^b g(x)dx$$

so for the (14) we can find a random variable X that has a $p.d.f.$ function f with support on the interval $[a, b]$. One idea

could be a random variable $X \sim U(a,b)$ with density function $f(x) : \frac{1}{b-a}$ and so our integration can be:

$$\mathcal{I} = \int_a^b g(x)dx = \int_a^b (b-a)\frac{1}{b-a}g(x)dx =$$

$$= (b-a)\int_a^b g(x)f(x)dx = (b-a)E[g(X)] \qquad (15)$$

so, sampling multiple times the $r.v.$ X we can (after enough amounts of samples) calculate the expectation of the function $g(\cdot)$ and multiply it by the constant $(b-a)$ to obtain the Monte Carlo integral approximation.

C. Practical exercise (MC Integration)

The exercise is to compute the integral:

$$\mathcal{I} = \int_0^1 \frac{\sqrt{-log(x)}}{2}dx \qquad (16)$$

and starting from equation (15) we can set $a = 0$ and $b = 1$, so we sample $X \sim U(0,1)$ and $b - a = 1 - 0 = 1$ leaves (16) without changes. We can see a proof of $\mathcal{I}_{MC}$ truly being an unbiased estimator:

$$\begin{aligned}
E[\mathcal{I}_{MC}] &= E[\frac{1}{m}\sum_{i=1}^{m} g(X_i)] \\
&= \frac{1}{m}\sum_{i=1}^{m} E[g(X_i)] \\
&= E[g(X)] \\
&= \int_{-\infty}^{\infty} \frac{\sqrt{-log(x)}}{2} f_X(x)dx \\
\Big(X \sim U(0,1) \Rightarrow & f_X(\cdot) : [0,1] \to [0,1] \Rightarrow \mathcal{D}_X = \{0 \le x \le 1\}\Big) \\
&= \int_0^1 \frac{\sqrt{-log(x)}}{2}dx \\
&= \mathcal{I}
\end{aligned}$$

To compute the integral approximation $\mathcal{I}_{MC}$ and the variance of the $\mathcal{I}_{MC}$ we can proceed as follow. The integral approximation:

$$\mathcal{I}_{MC} = \frac{1}{m}\sum_{i=1}^{m} g(X_i)$$

and from the variance of the $\mathcal{I}_{MC}$:

$$\begin{aligned}
Var(\mathcal{I}_{MC}) &= Var(\frac{1}{m}\sum_{i=1}^{m} g(X_i)) \\
&= \frac{1}{m^2}\sum_{i=1}^{m} Var(g(X_i)) \\
&= \frac{1}{m^2} m Var(g(X)) \\
&= \frac{Var(g(X))}{m}
\end{aligned}$$

we can obtain a way to approximate the variance of the integral approximation as:

$$Var(\mathcal{I}_{MC}) \approx \frac{\left(\frac{1}{m}\sum_{i=1}^{m}(g(X_i) - \mathcal{I}_{MC})^2\right)}{m}$$

D. Theory (Control variables)

Monte Carlo estimator has two important values: the size of the simulation m and the variance of the approximation $Var(\mathcal{I}_{MC})$.

Many times the stopping condition for the sampling, in order to improve the Monte Carlo estimate, is based on obtaining a value for the estimation of the variance less than some threshold. In some of these cases the computational cost of the approximation can increase and become huge. To avoid this some methods have been developed to reduce the variance $Var(\mathcal{I}_{MC})$ of a Monte Carlo estimation $\mathcal{I}_{MC}$. One of these methods is called *Control variables*.

Assume that there exists a function $h(\cdot)$ of which we know the Expectation $\mu = E[h(X)]$ and that is correlated with $g(X)$.

In this case it is possible to prove that $\forall c \in \mathbb{R}$ the integral:

$$\mathcal{I}_C = g(X) + c(h(X) - \mu)$$

is an unbiased estimator of $\mathcal{I}$.

The variance of the estimator is calculated as:

$$\begin{aligned}
&\text{Var}(\text{I}_C) = Var(g(X) + c(h(X) - \mu)) \\
&= \text{Var}(\text{g(X) + c } \cdot h(X) \underbrace{- c\mu}_{constant}) \\
&= \text{Var}(\text{g(X) + c } \cdot h(X)) \\
&\Big(\text{Var(aX + bY)} = \\
&\text{a}^2 Var(X) + b^2 Var(Y) + 2abCov(X,Y)\Big) \\
&= \underbrace{Var(g(X))}_{c} + \underbrace{\boldsymbol{c^2} Var(h(X))}_{ax^2} + \underbrace{2\boldsymbol{c} Cov(g(X), h(X))}_{bx}
\end{aligned}$$

this can be seen as a quadratic function of the c variable and since we want the smallest variance possible, we can choose the c value that minimizes the variance value.

The x correspondent to the minimum value of a quadratic function $f(x) = ax^2 + bx + c$ is written as $x = -\frac{b}{2a}$ and in our case it means that the optimum is:

$$c^* = \frac{-\cancel{2}Cov(g(X), h(X))}{\cancel{2}Var(h(X))} \qquad (17)$$

that makes the variance (by substitution):

$$\begin{aligned}
Var(\mathcal{I}_C) =& Var(g(X))\frac{-Cov^2(g(X), h(X))}{Var(h(X))} \\
=& Var(g(X))(1 - \rho^2)
\end{aligned}$$

with $\rho = Cor(g(X), h(X))$.

So, given $X_1, X_2, \ldots, X_m$ a sample of the $r.v.$ X, the *Monte Carlo variable-control-based* estimator of $\mathcal{I}$ is:

$$\mathcal{I}_{MC_{cv}} = \frac{1}{m}\sum_{i=1}^{m}\Big(g(X_i) + c^*(h(X_i) - \mu)\Big) \qquad (18)$$

that has respectively expectation and variance as:

$$E[\mathcal{I}_{MC_{cv}}] = \mathcal{I}$$
$$Var(\mathcal{I}_{MC_{cv}}) = Var(g(X))(1-\rho^2)$$

E. Practical exercise (Control variable)

Given the integral in (16), and given one sample $X_1, X_2, \ldots, X_m$ of the $r.v.$ X, first we calculate

$$c^* = \frac{-Cov\Big(\frac{\sqrt{-log(X)}}{2}, X\Big)}{Var(X)}$$

and then the Integral approximation

$$\mathcal{I}_{MC_{cv}} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\sqrt{-log(X_i)}}{2} + c^*(X_i - \mu)\right)$$

where $\mu = \frac{1}{m}\sum_{i=1}^{m} X_i$ (sample mean).

As last thing, the following:

$$Var_{\%red} = 100 \cdot \Big(1 - \frac{Var(\mathcal{I}_{MC_{cv}})}{Var(\mathcal{I}_{MC})}\Big)$$

is percentage of variance reduction that is achieved when using $\mathcal{I}_{MC_{cv}}$ instead of $\mathcal{I}_{MC}$.

F. Exercise

a) Use the R function integrate() to compute the value of I.

```
#Defining the g function
g <- function(x){sqrt(-log(x))/2}

#Use the R function integrate() to
#     calculate I
I <- integrate(g, lower = 0, upper = 1)
I$value
#0.4431136
```

b) Describe and implement in R the Monte Carlo method of size m = 10000 for estimating I. Report an estimate of the variance of the Monte Carlo estimator $\hat{I}_{CM}$ of I.

```
set.seed(456)
m=10000
x=runif(m,0,1)
I_MC <- mean(g(x)); I_MC
#0.4401598
error <- abs(I_MC - I$value); error
#0.002953797

# estimate of the variance
Var_I_MC <- var(g(x))/m; Var_I_MC
#5.195264e-06
```

c) Describe and implement in R the Monte Carlo method of size m = 10000 based on control variables for estimating I. Report an estimate of the variance of the Monte Carlo estimator $\hat{I}_C$ of I.

```
m = 10000
c_st <- -cov(g(x), x)/var(x); c_st
#0.7720018

I_c<-mean(g(x)+c_st*(x-mean(x)))
#0.4401598

#Estimated variance
Var_I_C<-(var(g(x))
-((cov(g(x),x))^2/var(x)))/m
#2.764839e-07
#We can see that this variance is lower
#than the previous result
```

d) What's the percentage of variance reduction that is achieved when using $\hat{I}_C$ instead of $\tilde{I}_{MC}$.

```
Var_red_p<-((Var_I_MC-Var_I_C)/Var_I_MC)*100
#94.67816%
```

The use of control variables improved the precision of the estimation by a considerable amount, this approach is use full if precision is an issue for the application.

IV. Exercise 4 - Hypothesis testing

A. Theory of hypothesis testing

1) Basics:

- two-tailed test: $H_0 : \sigma^2 = \sigma_0^2$ vs. $H_1 : \sigma^2 \neq \sigma_0^2$
- right-tailed test: $H_0 : \sigma^2 \leq \sigma_0^2$ vs. $H_1 : \sigma^2 > \sigma_0^2$
- left-tailed test: $H_0 : \sigma^2 \geq \sigma_0^2$ vs. $H_1 : \sigma^2 < \sigma_0^2$

2) Kolmogorov-Smirnov test: It is a test of the equality of continuous probability distributions to compare our with the χ^2_{n-1} distribution as a reference probability distribution. It quantifies the distance between the empirical distribution function of our sample and the cumulative distribution function of the reference distribution.

H_0 : samples are drawn from the same distribution vs.

H_1 : samples are drawn from different distributions.

3) Binomial test: Performs an exact test of a simple null hypothesis about the probability of success in a Bernoulli experiment. In this case to see whether the empirical and theoretical α depart significantly.

B. Exercises

Let $X_1, ..., X_n$ be a random sample from the population $X \sim N(1, 3)$.

We consider the number of simulations $m = 1000$ and a sample size of $n = 25$.

1) Does $\frac{(n-1)S^2}{2} \sim \chi^2_{n-1}$ *?:* In this exercise we will validate via a Monte Carlo simulation study if:

$$\chi = \frac{(n-1)S^2}{2} \sim \chi^2_{n-1} \tag{19}$$

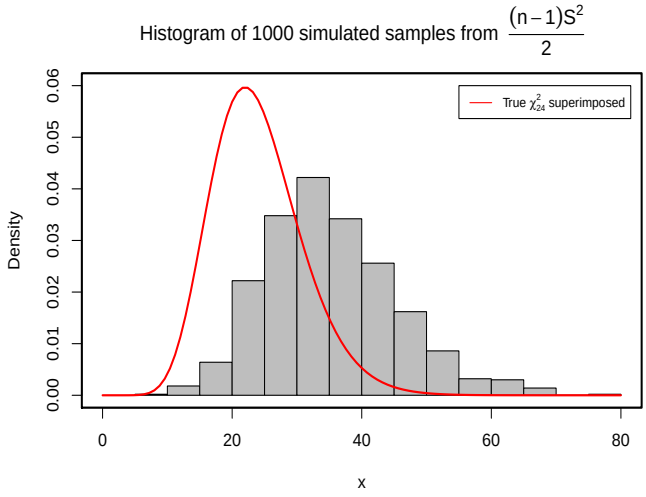
To validate (19), we simulate $\chi_1, ..., \chi_{1000}$ using Monte Carlo.

Listing 4. Generating 1000 samples in R

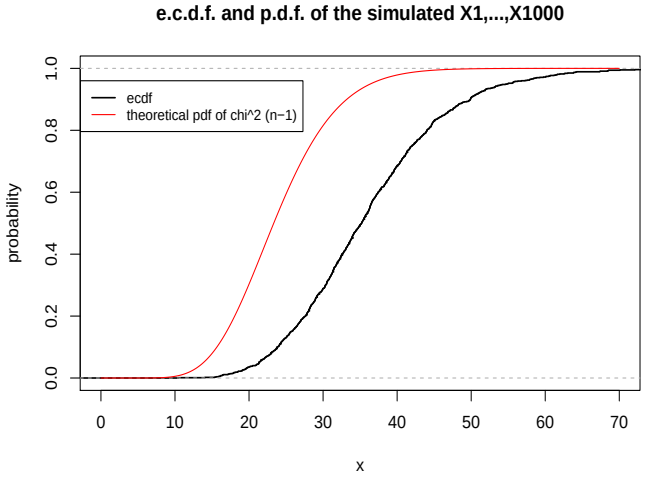
```
m=1000; n=25; alpha=0.05;
mu=1; sigma=sqrt(3)

xi <- vector()
for(i in 1:m){
  sample <- rnorm(n,mu,sigma)
  xi <- c(xi, (n-1) * var(sample) / 2)
  xi
}
```

The histogram of the simulated $\chi_1, ..., \chi_{1000}$ with the theoretical density superimposed.



It's clear from this plot that our samples aren't following χ^2_{24}.
If we plot the empirical cumulative distribution (ecdf) of $\chi_1 \cdots \chi_{1000}$ with the theoretical cumulative probability function (pdf) superimposed we can also take the same conclusion. As expected, the ecdf is more to the right than χ^2_{24} since in the histogram plot, most of our samples are focused in an x area that is to the right from most of the χ^2_{24} density.



2) Kolmogorov-Smirnov two-sided test: Now we will perform a Kolmogorov-Smirnov two-sided test to test the following hypothesis:
H_0 : both samples drawn from the χ^2_{24} distribution
H_1 : samples are drawn from different distributions.

```
ks.test(xi, pchisq, n-1,
        alternative = "two.sided")
```

p-value $< 2.2 \cdot 10^{-16}$

Since p-value $< \alpha = 0.05$, it means we have sufficient statistical evidence to reject H_0.

3) Theoretical quantiles comparison: Now we move on to calculate the theoretical quantiles 0.90, 0.95, 0.975 with the empirical quantiles. We expect a priori to get higher values for the empirical quantiles because the histogram of the samples is to the right of the χ^2_{24} density function.

Listing 5. quantiles comparison in R

```
qchisq(c(0.9,0.95,0.975), n-1)
quantile(xi, probs = c(0.9,0.95,0.975))
```

	90%	95%	97.5%
Theoretical quantiles	33.19624	36.41503	39.36408
Empirical quantiles	49.00081	54.54538	58.28360

4) Hypothesis test for σ^2*:* In question b) we are supposed to perform the following hypothesis test:

$$H_0 : \sigma^2 \leq 3 \quad vs. \quad H_1 : \sigma^2 > 3 \tag{20}$$

The test statistic is the following:

$$X^2 = \frac{(n-1)S^2}{\sigma_0^2} \sim \chi^2_{n-1}$$

And the critical region for α is:

$$R_\alpha =]\chi^2_{24,1-\alpha}; +\infty[$$

The pvalue is then computed as: p-value = $P(X > X_{obs}) = 1 - P(X_{obs} \leq X_{obs})$, where

$$X_{obs} = \frac{(24)S^2}{2}$$

Listing 6. computation of empirical p-values in R

```
p <- numeric(m)
for(i in 1:m){
  x=rnorm(n,mu,sigma)
  X2 = (n-1)*sd(x)^2/sigma^2
  p[i] = 1-pchisq(X2, n-1)
  p      # empirical p-values

  phat=mean(p<alpha)
  phat
}
```

The empirical signicance level obtained was $\hat{\alpha} = 0.042$. We can now perform a Binomial test to see if $\hat{\alpha}$ departs signicantly from α:

Listing 7. binomial test for the empirical significance level

```
binom.test(phat*m,m,p=0.05)
bin.test$p.value
```

$p = 0.2760517$

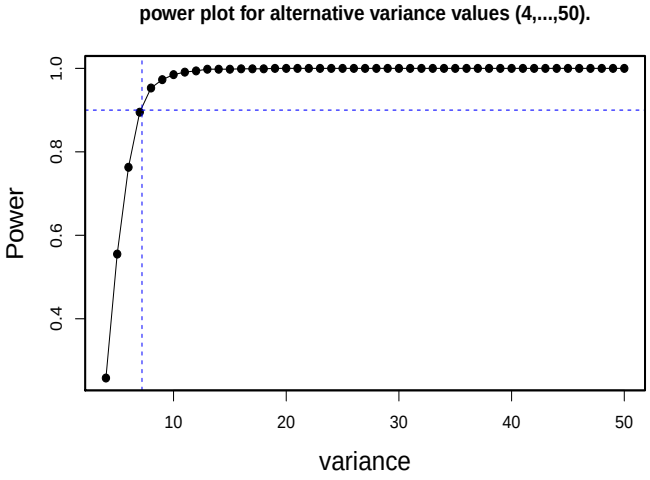
$\Rightarrow$ The empirical and theoretical α level do not depart significantly.

5) Power plot for alternative σ_i: Finally, we want to make a power plot for alternative variance values from 4 to 50, and answer the question: "How far from H_0 does one need to be so that the power of the test gets higher than 90%?"

We know that the power of a test is that test's ability to correctly reject the H_0. Therefore, for each value of the variance v_1, we will count the fraction of how many samples out of the 1k samples we generated with that variance v_1 the test was able to correctly reject.

Listing 8. simulating the power of the test for multiple values of the variance in R

```
v1=4:50
sd1 = sqrt(v1)
nv1=length(v1)
power=vector()
set.seed(789)
for(sds in sd1){
  p=vector()
  for(i in 1:m){
    x=rnorm(n,mu,sds)
    X2 = (n-1)*sd(x)^2/sigma^2
    p[i] = 1 - pchisq(X2, n-1)
  }
  power=c(power,mean(p<alpha))
}
```



$\Rightarrow$ We can see in the power plot that one need to have $\sigma^2 > 7$ to obtain a test power greater 90 %.

REFERENCES

[1] Sheldon M Ross. *Simulation*. Academic press, 2002.

[2] Sheldon M Ross. *Introduction to Probability Models, ISE*. Academic press, 2006.