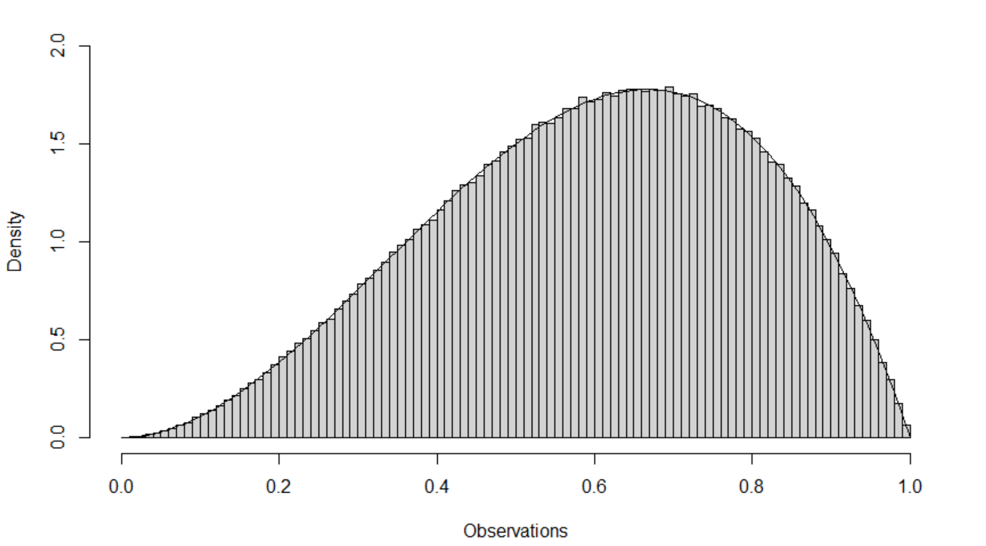
1. Generate a random sample of size 1000 from the Beta(3,2) distribution using the acceptance-rejection method. Graph the histogram of the sample with the theoretical Beta(3,2) density superimposed as the Figure below.

```
> #先定義Beta distribution的pdf
> f = function(x){
+   if((0<x) && (x<1)){
+     ((x^2)*(1-x))/beta(3,2)
+   }else{
+     return(0)
+   }
+ }
>
> #定義g(x) 選擇Uniform distribution
> g <- function(x) {
+   dunif(x)
+ }
>
>
> #計算M,max(f(t)/g(t)),for all t
> M = max(sapply(seq(0, 1, by = 0.001), function(x) f(x) / g(x)))
>
> #Acceptance-Rejection Method
> rejectionBeta = function(f,M){
+   while(TRUE){
+     x = runif(1) #從g分布中找x,且f(x)/g(x) <= M
+     y = runif(1) #用Uniform distribution找機率
+     if(y<f(x)/M*g(x)){
+       return(x)
+     }
+   }
+ }
>
> #找出服從Beta distribution的隨機變數
> set.seed(21)
> nrep = 1000000 #大樣本更為符合想要找到的Distribution
> Observations = rep(0,nrep)
> for (i in 1:nrep) {
+   Observations[i] = rejectionBeta(f,M)
+ }
>
> #畫圖確認
> hist(Observations,breaks = seq(0,1,by=0.01),freq = FALSE,ylim = c(0,2),main = "")
> y = seq(0,1,0.01)
> fy = ((y^2)*(1-y))/beta(3,2) #Beta distribution(3,2)的pdf
> lines(y,fy)
```

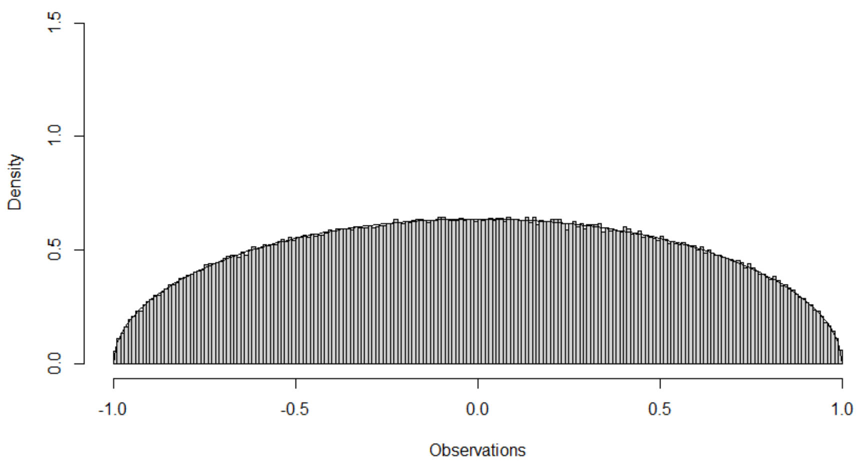


2. Generate a random sample of size 1000 from the pdf

$$f(x) = \frac{2}{\pi R^2}\sqrt{R^2 - x^2}, \quad -R \le x \le R$$

using the acceptance-rejection method. Graph the histogram of the sample with the theoretical density superimposed.

```
> #給定常數r(預設為1)，產生符合目標分配的直方圖，並畫出目標分配，確認是否合理
> generationR = function(r = 1){
+ 
+ #先把目標分配定義出來
+   f = function(x){
+     if((x>=-1*r)&&(x<=r)){
+       (2*sqrt(r^2-x^2))/(pi*r^2)
+     }else{
+       0
+     }
+   }
+ 
+ #選擇canditate分配
+   g = function(x){
+     dnorm(x)
+   }
+ 
+ #找出拒絕上限 M
+   M = max(sapply(seq(0, r, by = 0.001), function(x) f(x) / g(x)))
+ 
+ #Acceptance-Rejection Method
+   rejectionK = function(f,M){
+     while(TRUE){
+       x = rnorm(1)
+       y = runif(1)
+       if(y<f(x)/(M*g(x))){
+         return(x)
+       }
+     }
+   }
+ 
+ #找出服從目標分配的隨機變數
+   set.seed(21)
+   n = 1000000
+   Observations = rep(0,n)
+   for(i in 1:n){
+     Observations[i]=rejectionK(f,M)
+   }
+ 
+ #畫圖確認
+   hist(Observations,breaks = seq(-1*r,r,by=0.01),freq = FALSE,ylim = c(0,M),main = "")
+   y = seq(-1*r,r,0.01)
+   fy =  (2*sqrt(r^2-y^2))/(pi*r^2)
+   lines(y,fy)
+ }
> 
> #執行函數
> generationR()
```

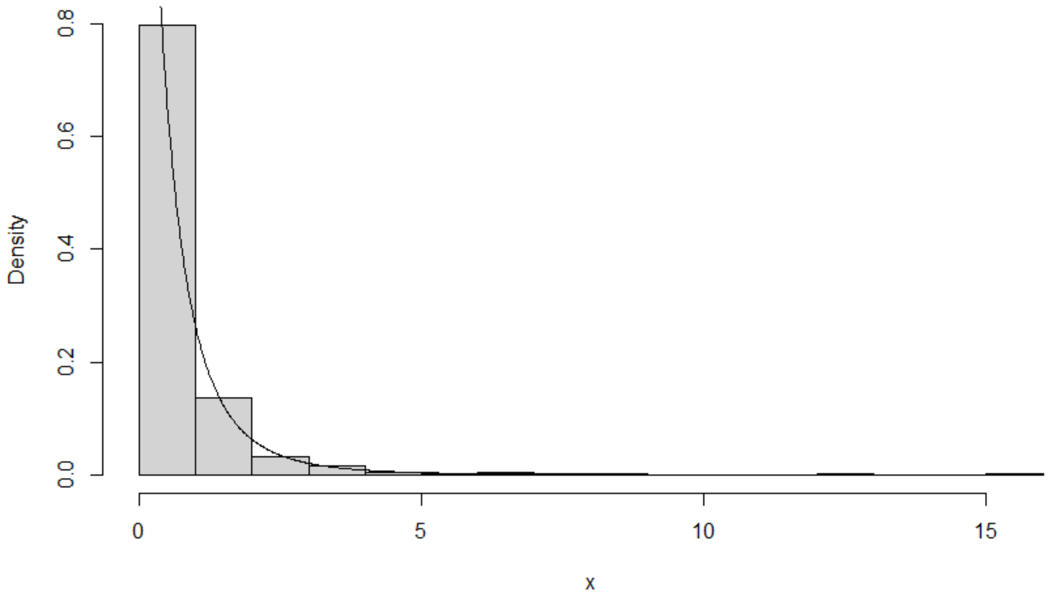


3. The continuous random variable X with positive support is said to have the Pareto distribution if its probability density function is given by

$$f(x) = \frac{\beta\alpha^{\beta}}{(x+\alpha)^{\beta+1}}, \quad x > 0.$$

Generate a random sample of size 1000 from the Pareto distribution with $\alpha = 2$ and $\beta = 4$ using "inverse transformation method". Compare the empirical and theoretical distributions by graphing the histogram of the sample and superimposing the Pareto density curve.

```
> #產生Pareto distribution(α = 2,β = 4)的隨機變數，並畫出直方圖
> generationP = function(a = 2,b = 4){
+   n = 1000
+   u = runif(n)
+   x = ((16/(1-u))^(1/b))-a #inverse transformation method
+   hist(x,prob = TRUE,main = "")
+   y = seq(0,8,0.01)
+   lines(y,(b*(a^b))/((y+a)^(b+1)))
+ }
>
> #執行函數
> generationP()
```



4. Please compare the efficiencies of two methods for generating the Wishart Distribution with $n = 100$ and covariance $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$.

a. A random sample of size n from multinormal distribution(d, mu = 0, sigma)

```
> #先定義共變異數矩陣
> Sigma = matrix(c(1,0.5,0.5,2),nrow = 2,ncol = 2)
>
> #產生X ~ Multinormal distribution(n,mean,Sigma)
> rmvn.eigen = function(n,mu,Sigma){
+   d = length(mu)
+   ev = eigen(Sigma,symmetric = TRUE)
+   lamda = ev$values
+   V = ev$vectors
+   R = V %*% diag(sqrt(lamda)) %*% t(V)
+   Z = matrix(rnorm(n*d),nrow = n,ncol = d)
+   X = Z %*% R + matrix(mu,n,d, byrow = TRUE)
+   X
+ }
>
> #產生一個X ~ Wishart distribution(d = 2, Sigma)
> w = function(d,Sigma){
+   mu = rep(0,d)
+   x = rmvn.eigen(1,mu,Sigma)
+   M = t(x) %*% x
+   M
+ }
>
> #產生n組X ~ Wishart distribution(d = 2, Sigma)
> W = function(n,d,Sigma){
+   x = list()
+   for (i in 1:n) {
+     x[[i]] = w(d,Sigma)
+   }
+   x
+ }
```

b. A method based on Bartlett's decomposition

```
> #產生A矩陣
> a = function(d){
+   # 產生 d x d 的矩陣
+   A <- matrix(0, nrow = d, ncol = d)
+
+   # 生成符合 N(0, 1) 分佈的隨機數填充 A[i, j] (i > j)
+   for (i in 1:d) {
+     for (j in 1:(max(i - 1, 1))) {
+       A[i, j] <- rnorm(1)
+     }
+   }
+
+   # 生成符合 sqrt(chisq(n-i+1)) 分佈的隨機數填充對角線 A[i, i]
+   for (i in 1:d) {
+     A[i, i] <- sqrt(rchisq(1, df = d - i + 1))
+   }
+   A
+ }
>
> #產生一組X ~ Wishart distribution(d = 2, Sigma)
> w2 = function(d,Sigma){
+   L = chol(Sigma)
+   A = a(d)
+   x = L %*% A %*% t(A) %*% t(L)
+   x
+ }
>
>
> #產生n組X ~ Wishart distribution(d = 2, Sigma)
> Bartlett_decomposition = function(n,d,Sigma){
+   x = list()
+   for (i in 1:n) {
+     x[[i]] = w2(d,Sigma)
+   }
+   x
+ }
```

Compare which method is more efficiency

```
> system.time({
+   W(1000000,2,Sigma)
+ })
使用者   系統   流逝
 10.75    0.28   21.58
>
> system.time({
+   Bartlett_decomposition(1000000,2,Sigma)
+ })
使用者   系統   流逝
  9.50    0.25   18.68
```

The method based on Bartlett's decomposition is more efficiency.