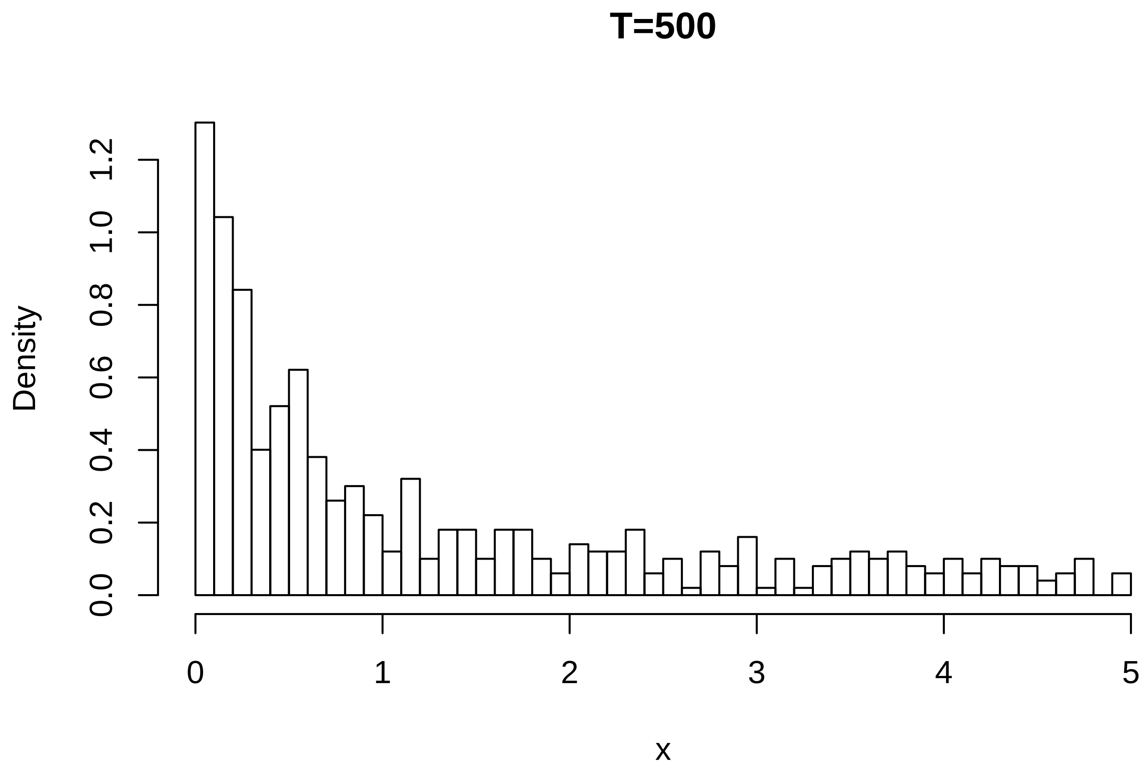
Gibbs

We write a function with upperbounds and sizes as parameters. To generate the given random variables, we use Inverse Transform Sampling. i.e. we calculate CDF for the truncated exponential distribution and find its inverse function.

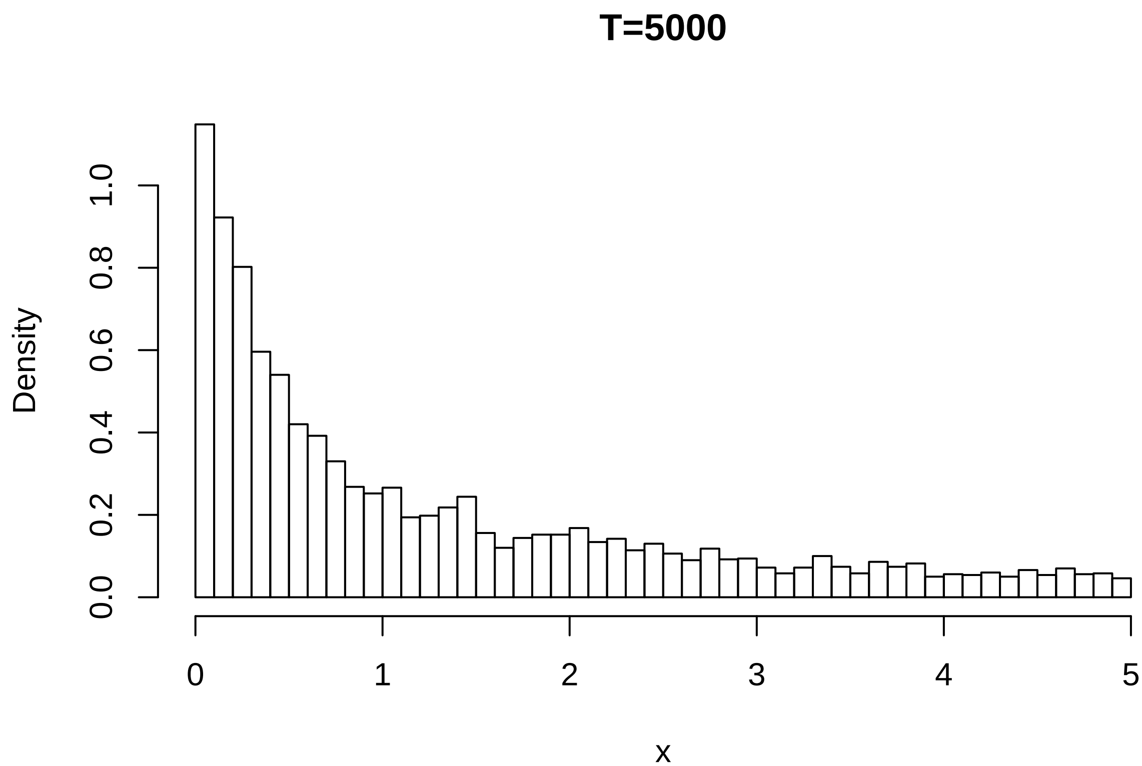
```
gibbs<-function(B,size){

  B=B
  x=numeric()
  y=numeric()

  y[1]=runif(1,0,5)# random starting value
  x[1]= -log(1-runif(1,0,1)*(1-exp(-y[1]*B)))/y[1]# manual calculated inverse cdf
  for (i in 2:size){# gibbs sampling algorithm
    y[i]= -log(1-runif(1,0,1)*(1-exp(-x[i-1]*B)))/x[i-1]
    x[i]= -log(1-runif(1,0,1)*(1-exp(-y[i]*B)))/y[i]
  }
  x=x[-size/10]#1/10 sample size used as burn-in
  return(x)
}
```

Now we generate x and histograms Obviously, the shape is closer to given pdf when T is larger.

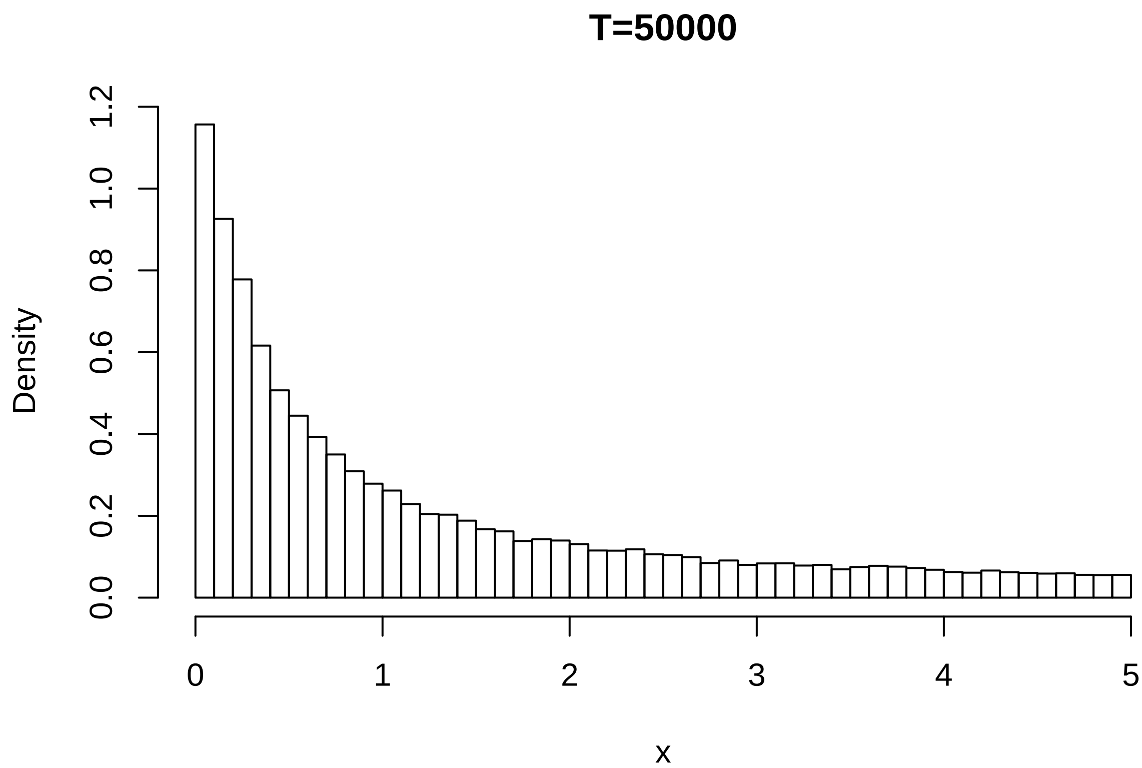
```
a1=gibbs(5,500)
hist(a1,breaks=40,freq=FALSE,xlab="x",main="T=500")
```




```
a2=gibbs(5,5000)
hist(a2,breaks=40,freq=FALSE,xlab="x",main="T=5000")
```




```r
a3=gibbs(5,50000)
hist(a3,breaks=40,freq=FALSE,xlab="x",main="T=50000")
```



Calculate the estimate of the expectation of X

```
mean(a1)#T=500
```

```
## [1] 1.285308
```

```
mean(a2)#T=5000
```

```
## [1] 1.271836
```

```
mean(a3)#T=50000
```

```
## [1] 1.266437
```