Gibbs Sampling

Chi-Chun Liu, Junxiao Bu

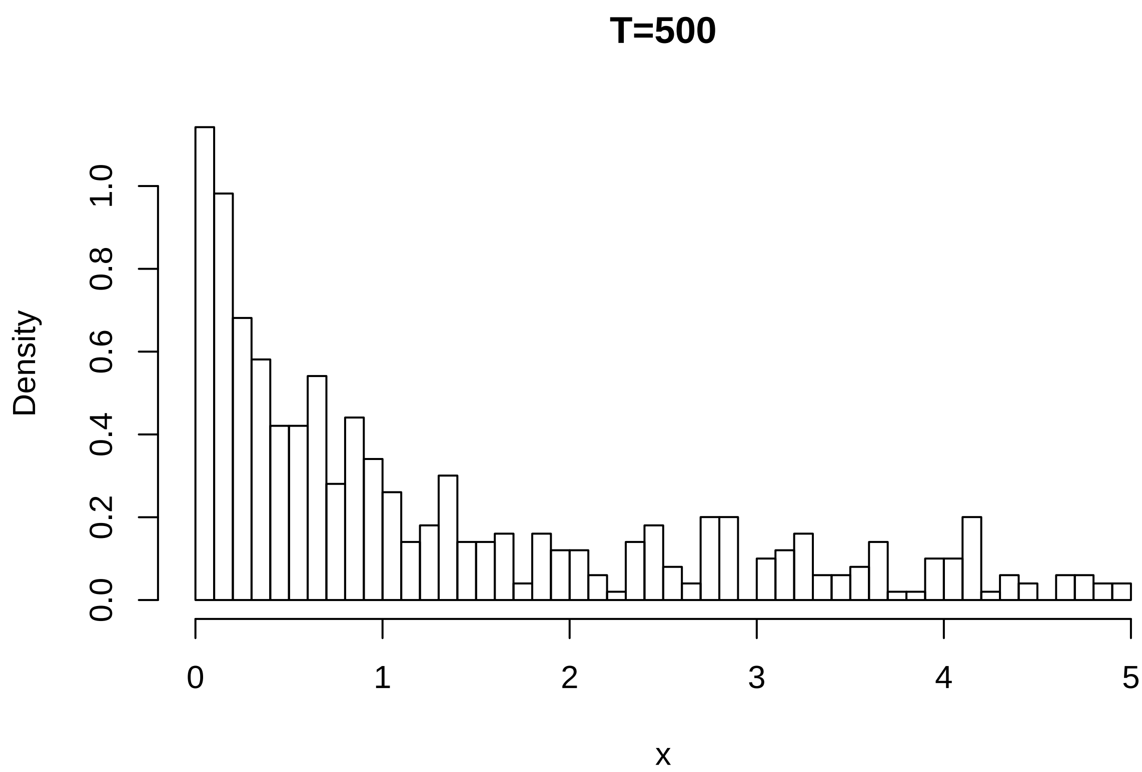
We write a function with upperbounds and sizes as parameters. To generate the given random variables, we use Inverse Transform Sampling. i.e. we calculate CDF for the truncated exponential distribution $\frac{1-\exp(-yx)}{1-\exp(-yB)}$ and find its inverse function. $x = \frac{-\log((1-z)(1-\exp(-yB))}{y}$ and plug in a unif(0,1) r.v.

Gibbs sampling generates r.v's from the marginal distribution.

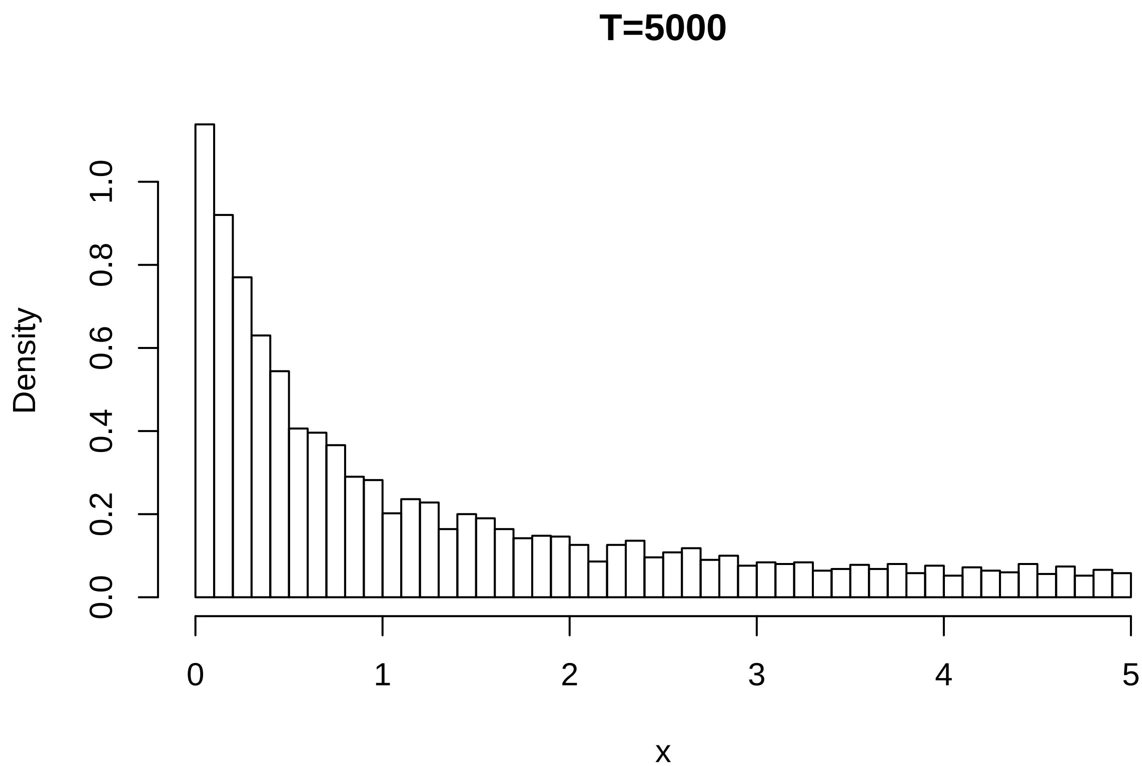
```
gibbs<-function(B,size){

  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.

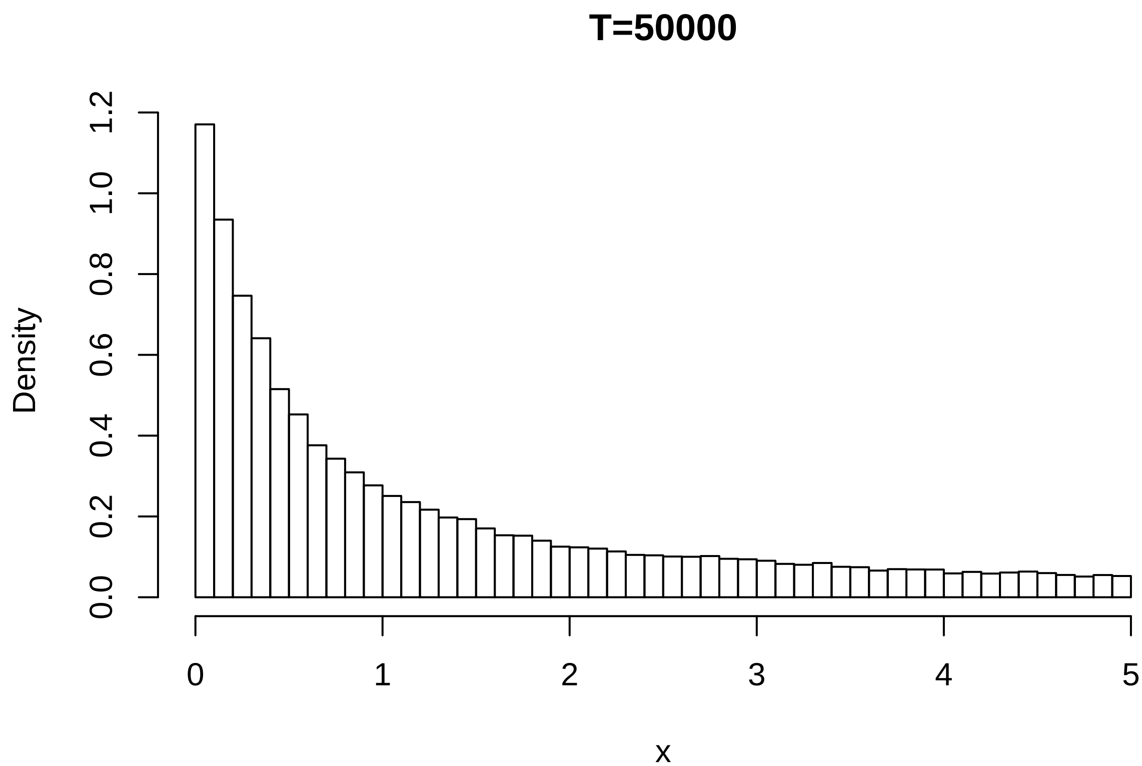
```
a1=gibbs(5,500)
hist(a1,breaks=40,freq=FALSE,xlab="x",main="T=500")
```




```
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

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

```
## [1] 1.291478
```

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

```
## [1] 1.282159
```

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

```
## [1] 1.260556
```