09-04-2020-stat-comp

Jikhan Jeong

9/4/2020

Random Number Generation by Inverse Transform

- Name: Jikhan Jeong
- Date: 09-04-2020 Ref: https://github.com/mariarizzo/SCR/tree/master/SCR-1e/SCR1e-code (https://github.com/mariarizzo/SCR/tree/master/SCR-1e/SCR1e-code)

```
setwd("C:/Users/jikhan.jeong/Documents/R/0_stat_comp")
getwd()
```

```
## [1] "C:/Users/jikhan.jeong/Documents/R/0_stat_comp"
```

```
library(rmarkdown)
```

```
rv <- sample(0:1, size =10, replace = TRUE); rv
```

```
##  [1] 0 0 0 0 0 0 0 1 1 1
```

```
multinomial <- sample(1:3, size= 100, replace = TRUE, prob= c(0.2, 0.3, 0.5))
table(multinomial)
```

```
## multinomial
##  1  2  3
## 29 20 51
```

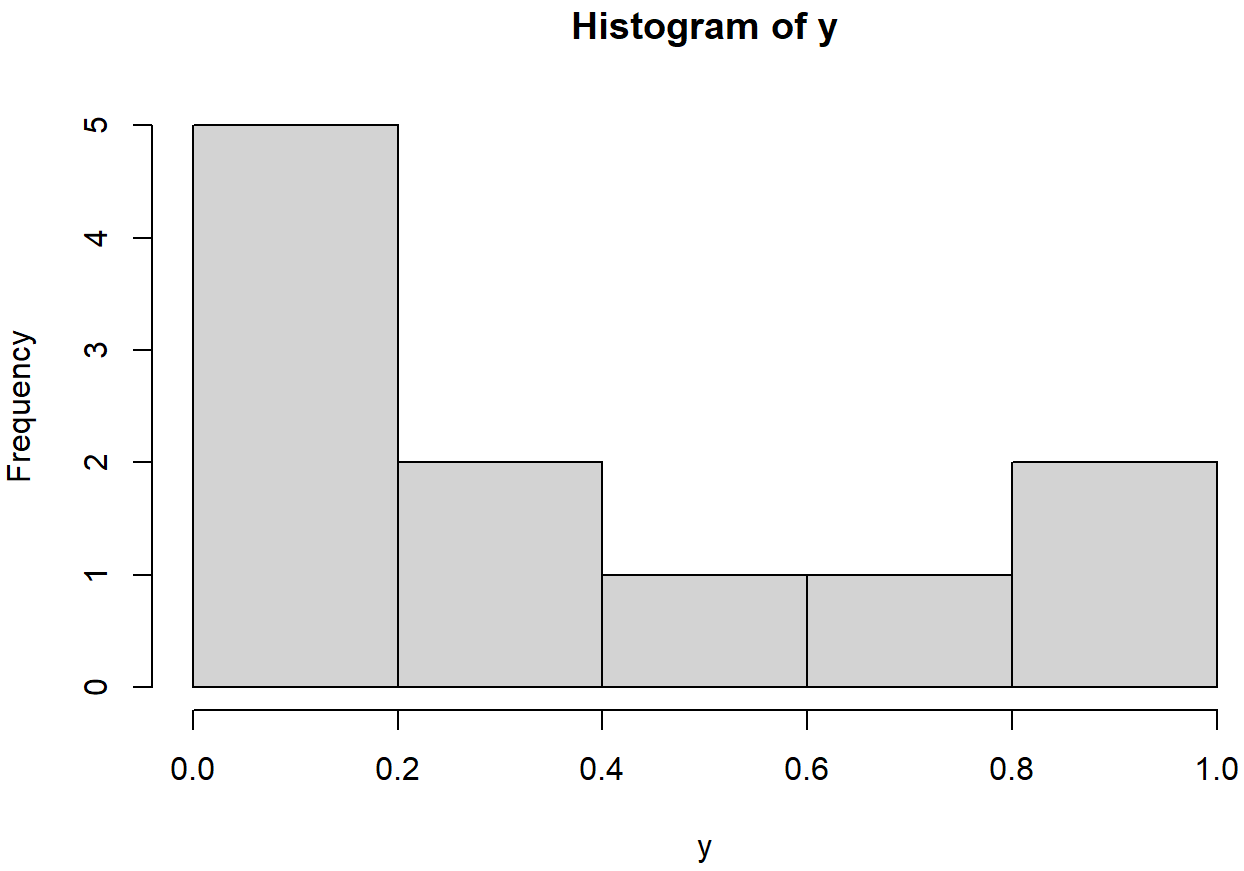
```
x = seq(0,1, by=0.1);x
```

```
##  [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

```
y = x^2
y[1:2]
```

```
## [1] 0.00 0.01
```

```
hist(y)
```




```
rnorm(10, 0,1) # normal distributin (size, mean, sd)
```

```
##  [1] -0.4900142 -0.4206600 -1.1019793 -0.1258808  0.2944888 -0.6987916
##  [7] -0.6069504 -0.2416197 -1.4635268 -2.0991689
```

```
rt(10,2) # student t (size, degree of freedom)
```

```
##  [1] -1.0953023 -1.0695771 -2.4103998  0.3566476  1.2967568 -5.6236909
##  [7] -0.9472847 -0.1824785 -3.0361890  2.3653622
```

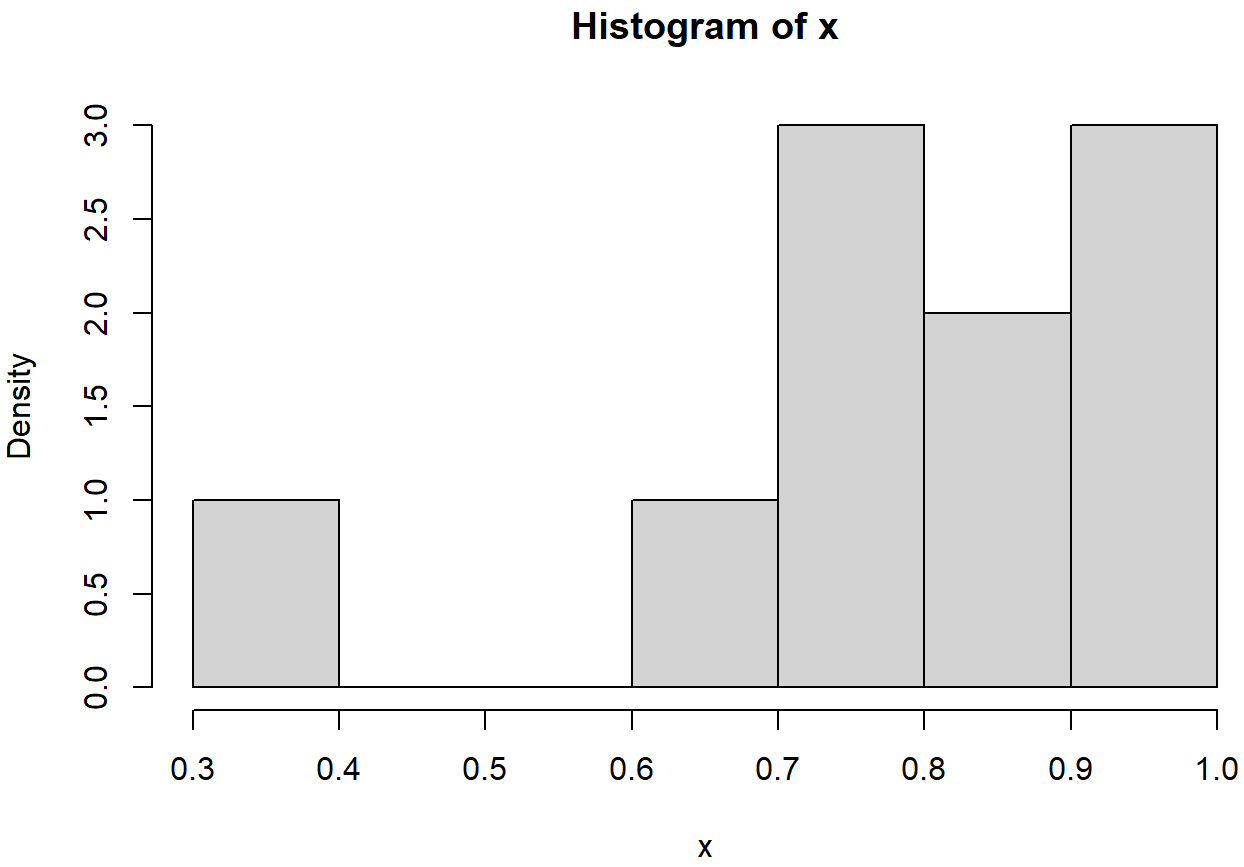
equation arrays

$$
\begin{aligned}
(a+b)^2 \\
a^2+2ab+b^2
\end{aligned}
$$

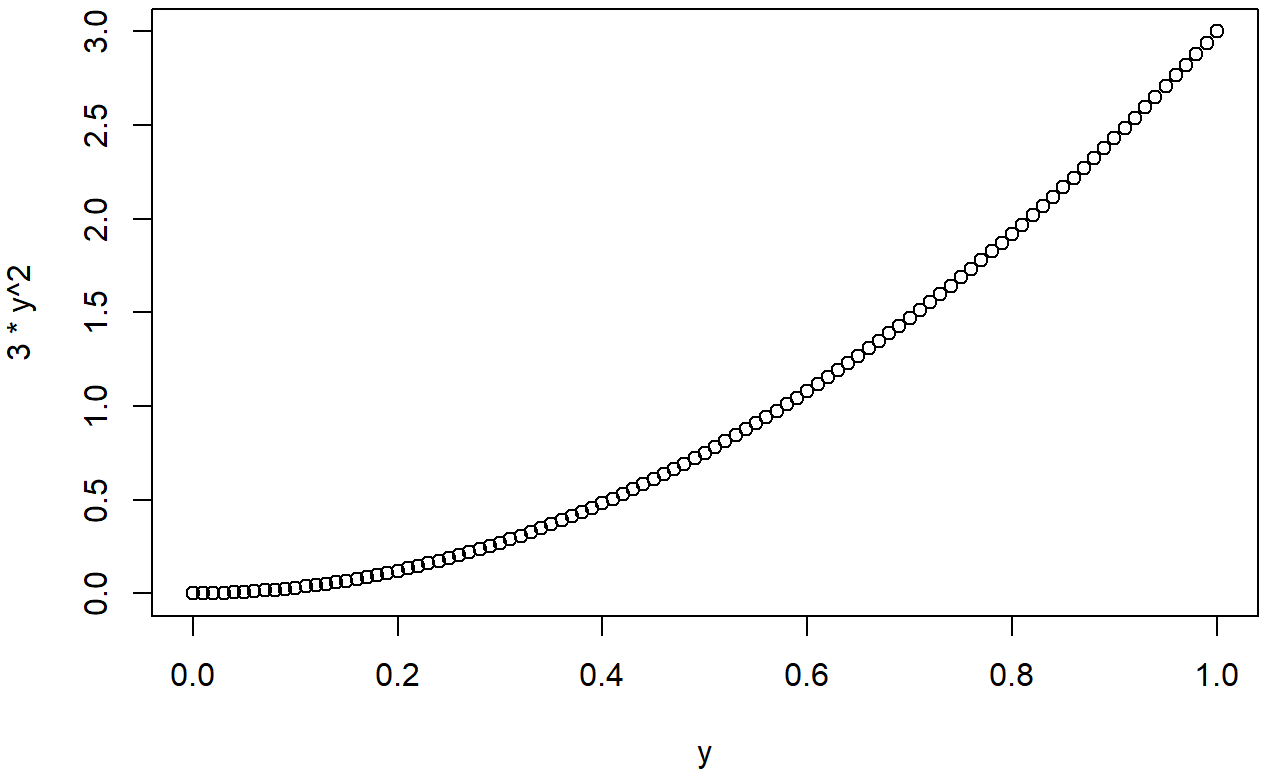
Inverse transform method, continuous case

f(x) = $3x^2$// F(x) = x^3// $F^{-1}(x) = u^{1/3}$//

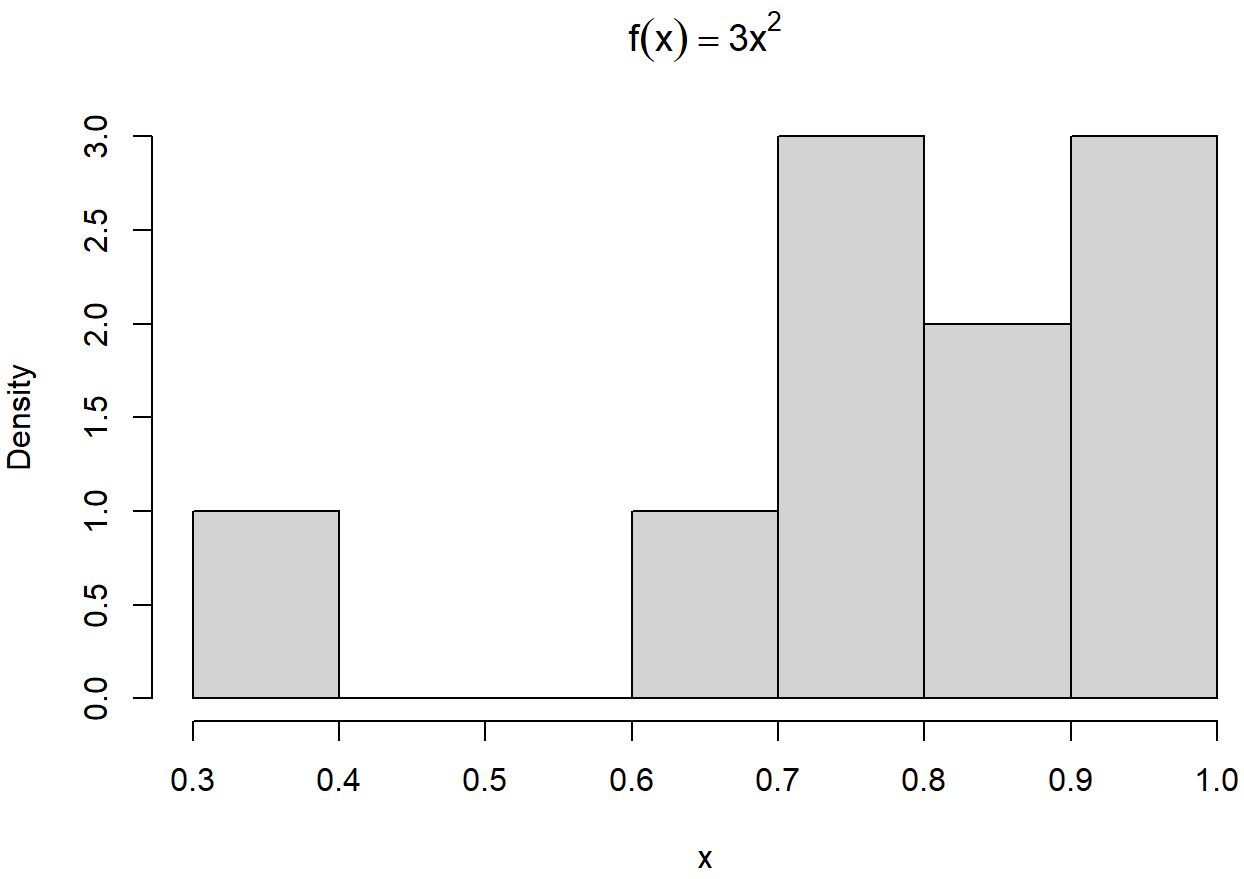
```
n <- 10
u <- runif(n)
x <- u^(1/3)
hist(x, prob=TRUE) # density histogram of sample from uniform
```



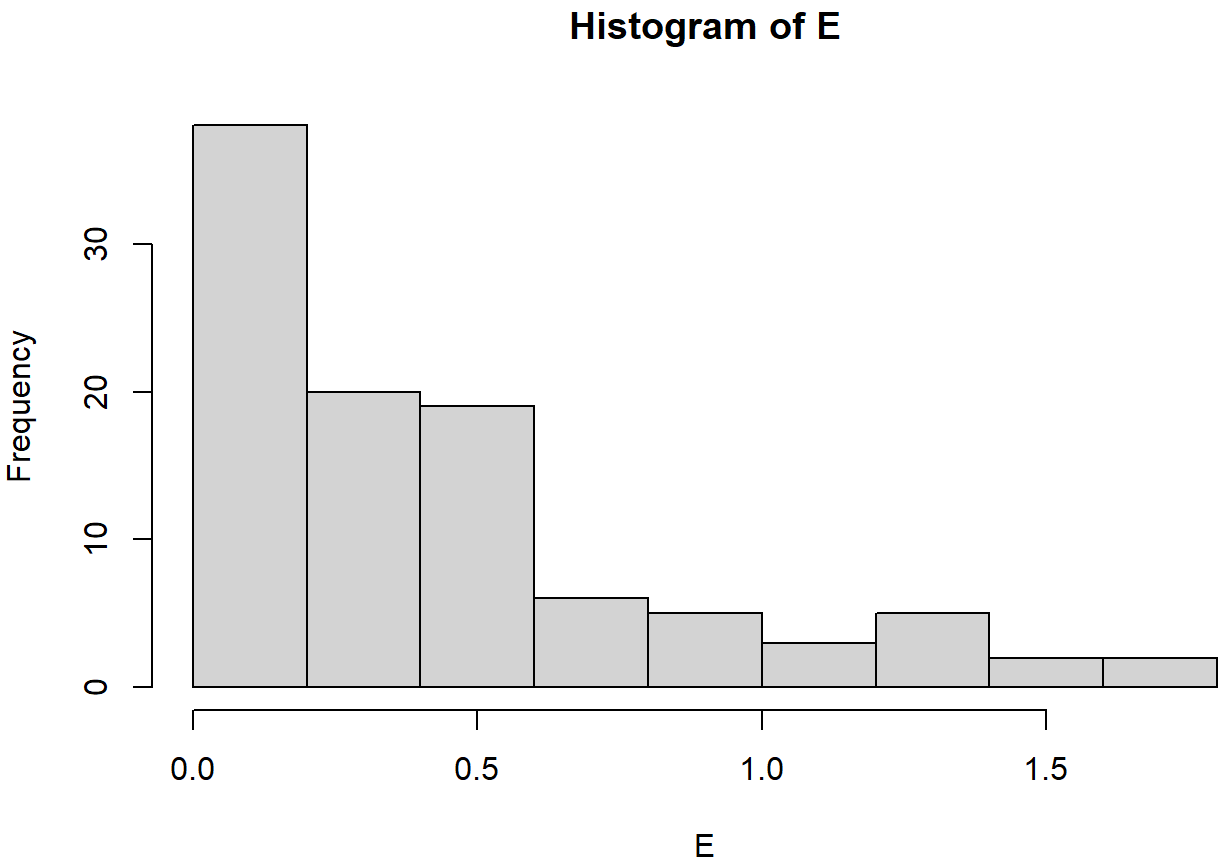

```
y <- seq(0,1, .01)
plot(y, 3*y^2)
```



```
hist(x, prob=TRUE, main = expression(f(x)==3*x^2))
```

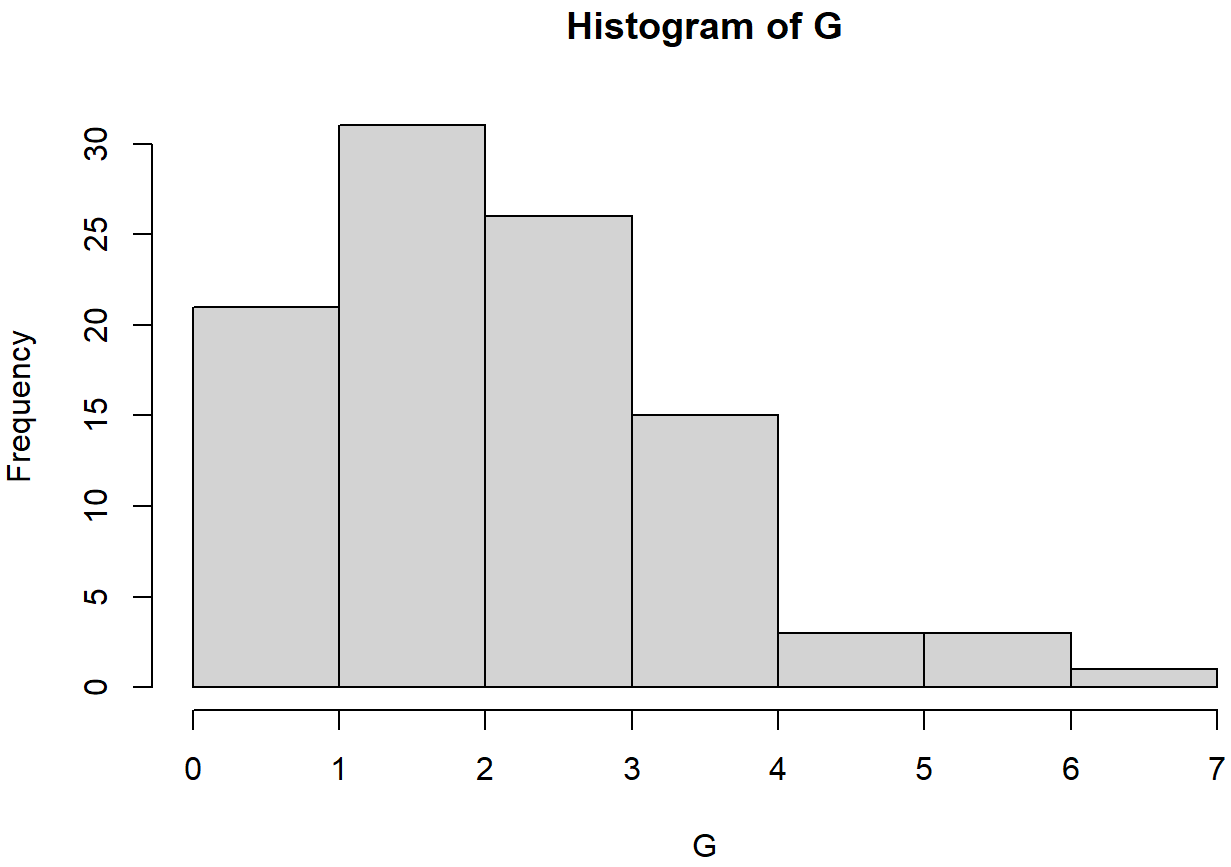



```
# 1. simulate from uniform
# 2. x = -1(1/lambda)* log(U)
lam =2
n = 100; U =runif(n)
E = -(1/lam)*log(U)
hist(E)
```



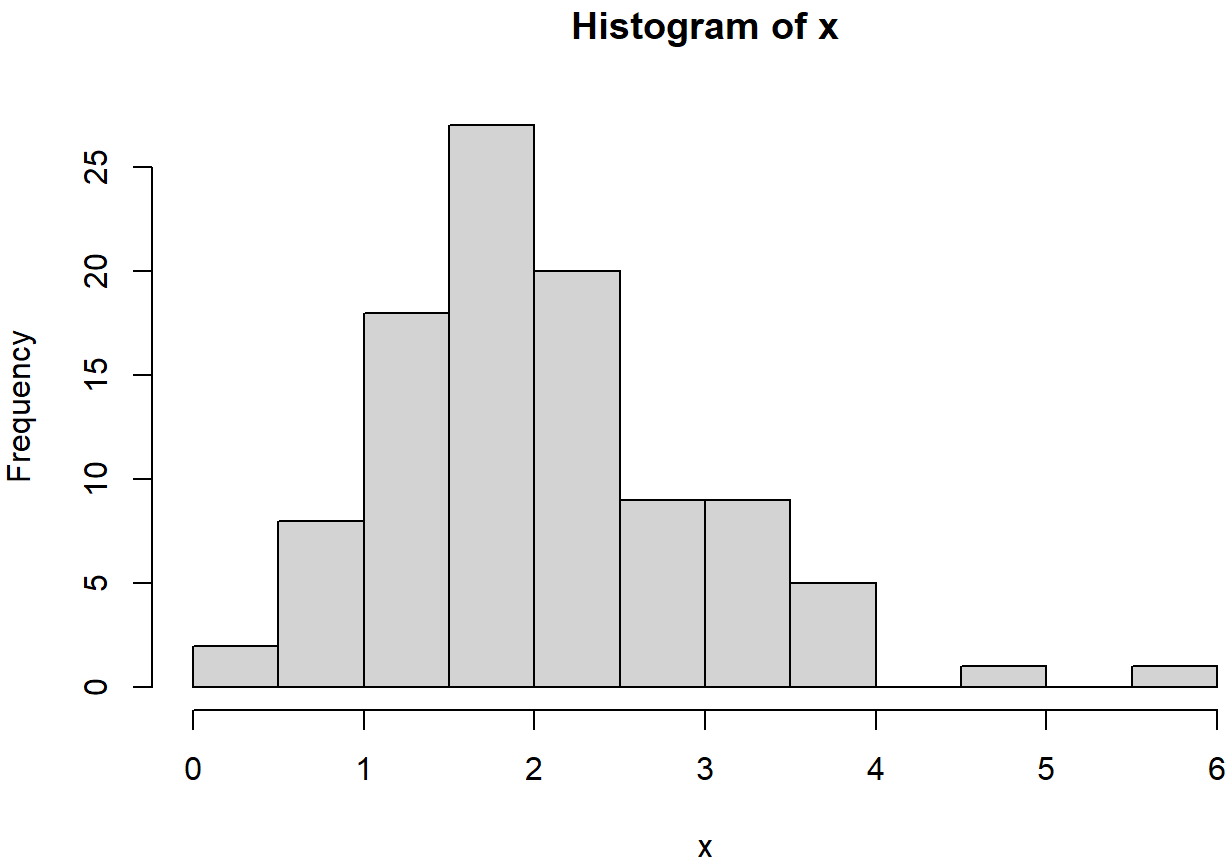

```
# simulate Bernoulli(p) from a uniform distribution
p = 0.5
n = 100; U=runif(n)
x =ifelse(u<(1-p),0,1)
```

```
# Gamma distribution
alpha = 2; beta =1; n=100
U = matrix(runif(n*alpha), alpha, n)
E = -log(U) # exp(l)
G = beta*colSums(E)
hist(G)
```

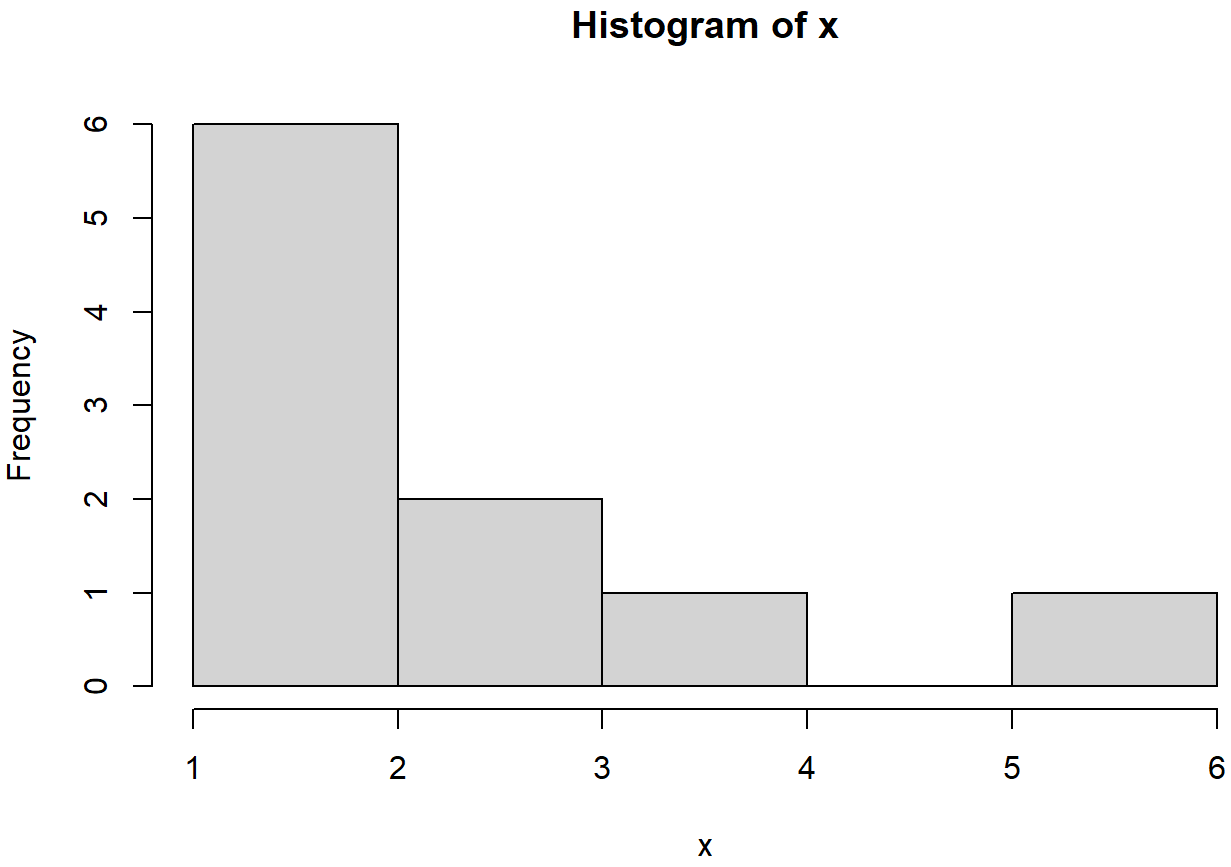



```
gamma_rv_generation = function(nsim=1000, alpha=3, beta=2){
  U = matrix(runif(n*alpha), alpha, n)
  E = -log(U) # exp(l)
  G = beta*colSums(E)
  return (G)
  }
```

```
x = gamma_rv_generation(10,4,0.5); hist(x)
```




```
# Exponetial
lambda <-2
x <- -log(runif(10)/lambda);hist(x)
```



```r
# Two point distribution
# Bernoulli, p =0.4
n <- 10
p <- 0.4
u <- runif(n);u
```

```
##  [1] 0.89173202 0.09651738 0.89002716 0.89728177 0.63764525 0.97496998
##  [7] 0.77723090 0.83145829 0.99769052 0.45548098
```

```r
x <- as.integer(u>.6);x
```

```
##  [1] 1 0 1 1 1 1 1 1 1 0
```

```r
mean(x); var(x)
```

```
## [1] 0.8
```

```
## [1] 0.1777778
```

```r
rbinom(n, size = 1, prob = p) # simple way
```

```
##  [1] 0 1 1 0 1 1 1 1 0 1
```

```r
sample(c(0,1), size = n, replace = TRUE, prob = c(.6,.4))
```

```
##  [1] 1 0 1 0 0 0 1 0 1 0
```