Random Variables

Boyie Chen

9/24/2019

R.V. - Review

Binomial X: r.v. representing the total number of 'H' for tossing a fair coin 10 times $X \sim Bin(n,p)$ $i.e. f_X(x) = (nCx)p^x(1-p)^{n-x}$ $X \sim Bin(n = 10, p = 0.5)$ $i.e. f(x) = (10Cx)0.5^x(1-0.5)^{10-x}$ eg. $f_X(x = 6)$ # 擲 10 次硬幣有 6 次正面朝上的機率

```
choose(10, 6) #C10 取 5
```

```
## [1] 210
```

```
factorial(10)/(factorial(6)*factorial(4)) #10!/(5!5!)
```

```
## [1] 210
```

```
choose(10,6)*(0.5^6)*(1-0.5)^(10-6) #value of f(x=6)
```

```
## [1] 0.2050781
```

```
dbinom(x = 6, size = 10, prob = 0.5) #equivalent to f(x=6), give the density(or pmf)
```

```
## [1] 0.2050781
```

```
pbinom(6, 10, 0.5) #give the CDF of X
```

```
## [1] 0.828125
```

```
# 算看看 F(X=6) = Σ P(X=0)+P(X=1)+P(X=2)+P(X=3)+P(X=4)+P(X=5)+P(X=6)
dbinom(0,10,0.5)+dbinom(1,10,0.5)+dbinom(2,10,0.5)+dbinom(3,10,0.5)+dbinom(4,10,0.5)+dbinom(5,10,0.5)+dl
```

```
## [1] 0.828125
```

```
#or use for loop
  cdf = 0
  for(i in 1:6){
    cdf = cdf+dbinom(i, 10, 0.5)
  }
  cdf
```

```
## [1] 0.8271484
```

```
qbinom(0.8271484, size = 10, prob = 0.5) #inverse CDF, given prob. then output quantile
```

```
## [1] 6
```

Remember the def of quantile, if CDF is not strictly increase, given p, find the min x s.t. $F_X(X = x) \geq p$

```
qbinom(0.65, 10, 0.5) #all 3 give the same output
```

```
## [1] 6
```

```
qbinom(0.7, 10, 0.5)
```

```
## [1] 6
```

```
qbinom(0.8, 10, 0.5)
```

```
## [1] 6
```

```r
pbinom(6, 10, 0.5) # 擲 10 次硬幣，正面朝上的次數小於等於 6 的機率
```

```
## [1] 0.828125
```

```r
pbinom(5, 10, 0.5) # 擲 10 次硬幣，正面朝上的次數小於等於 5 的機率
```

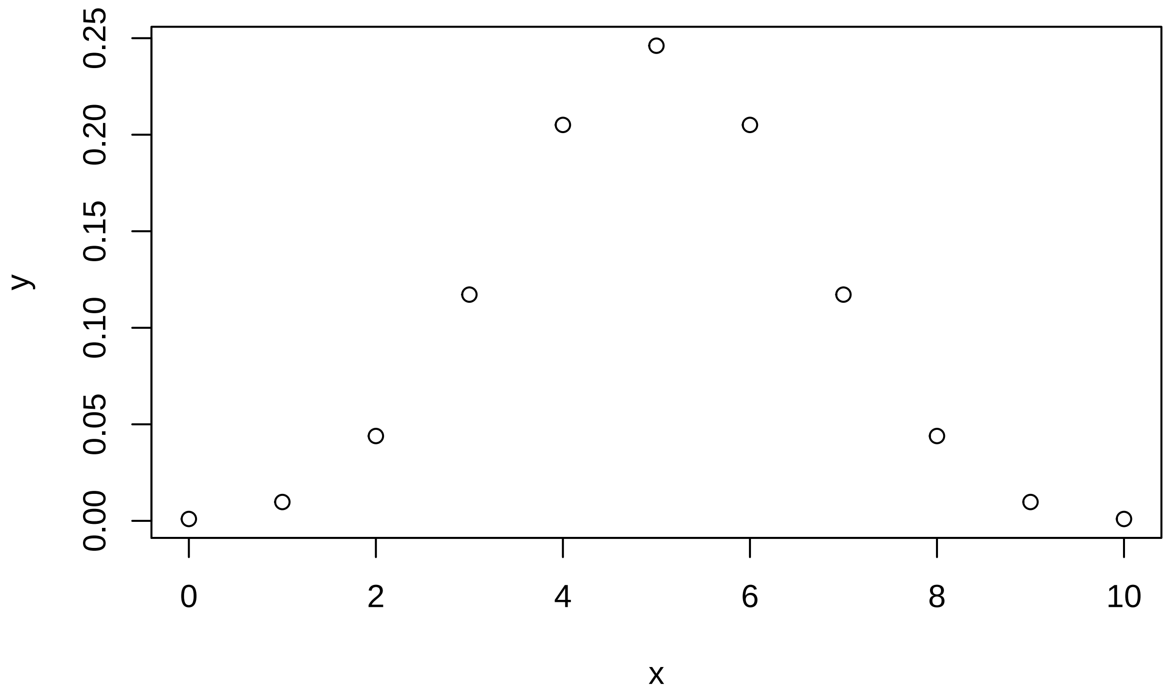
```
## [1] 0.6230469
```

```r
pbinom(6, 10, 0.5)-pbinom(5, 10, 0.5) # 相減就是正面朝上恰好為 6 次的機率
```

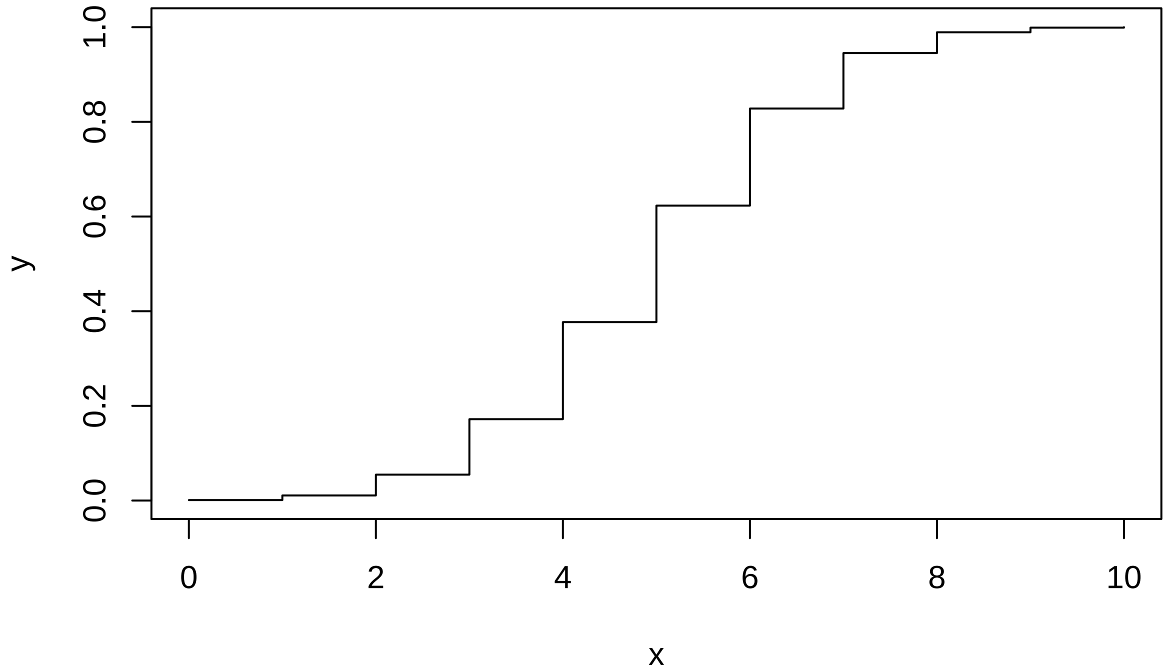
```
## [1] 0.2050781
```

Plotting

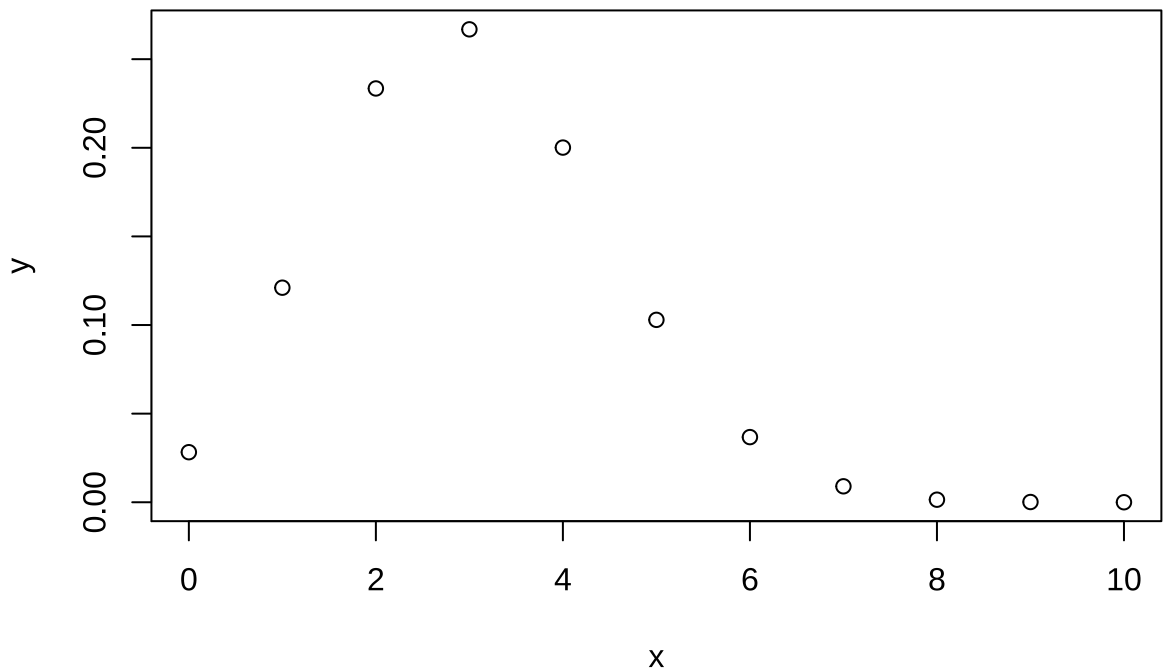
```r
#plot the pdf of Binomial
x = seq(from = 0, to = 10, by = 1)
y = dbinom(x, size = 10, prob = 0.5)
plot(x, y)
```



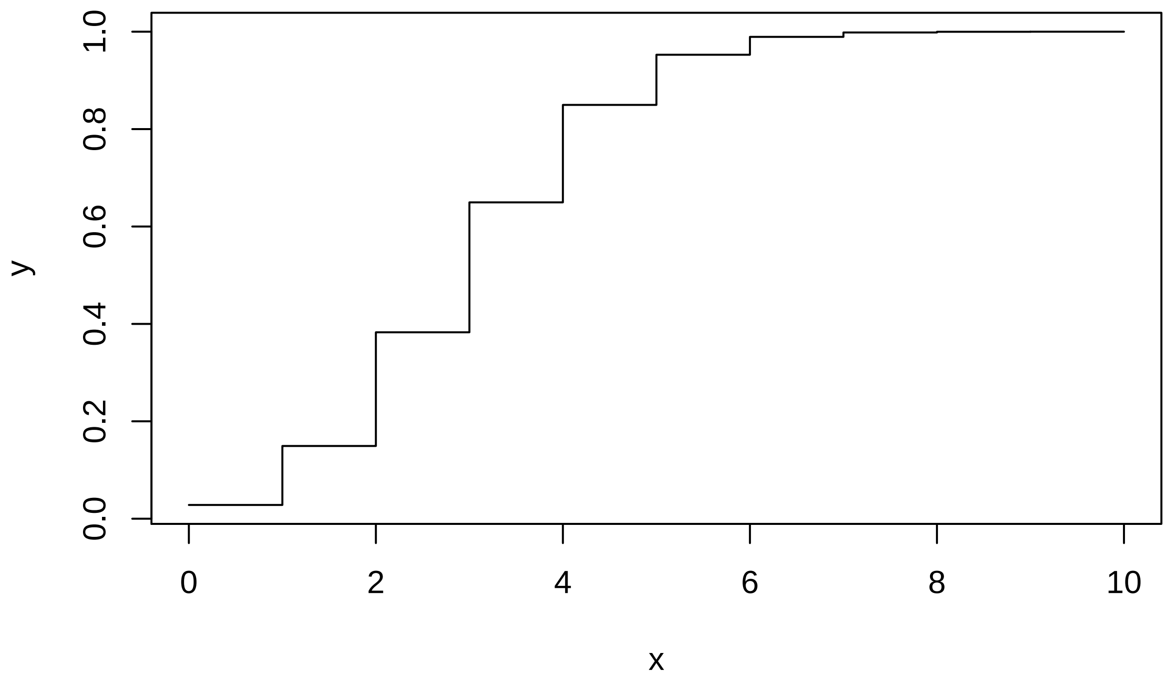

```r
#plot the cdf of Binomial
y = pbinom(x, 10, 0.5)
plot(x, y, type ='s')
```



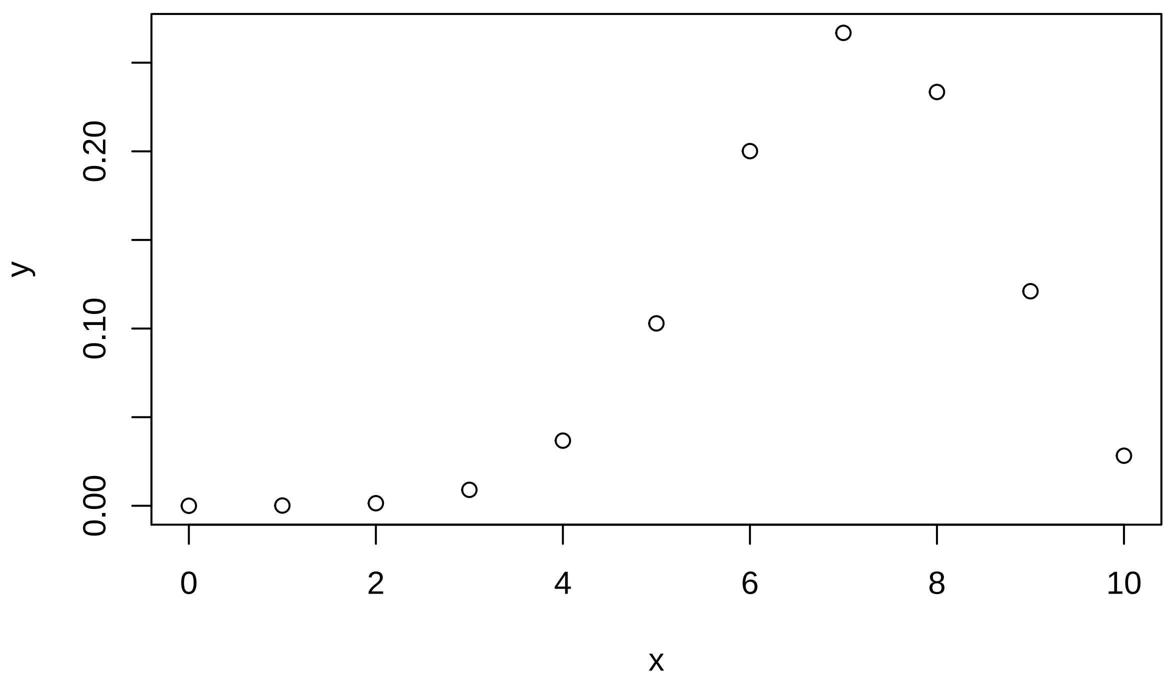

```r
#What if I change the Prob?
y = dbinom(x, 10, prob = 0.3)
plot(x, y) #pdf of Bin(10, 0.3)
```



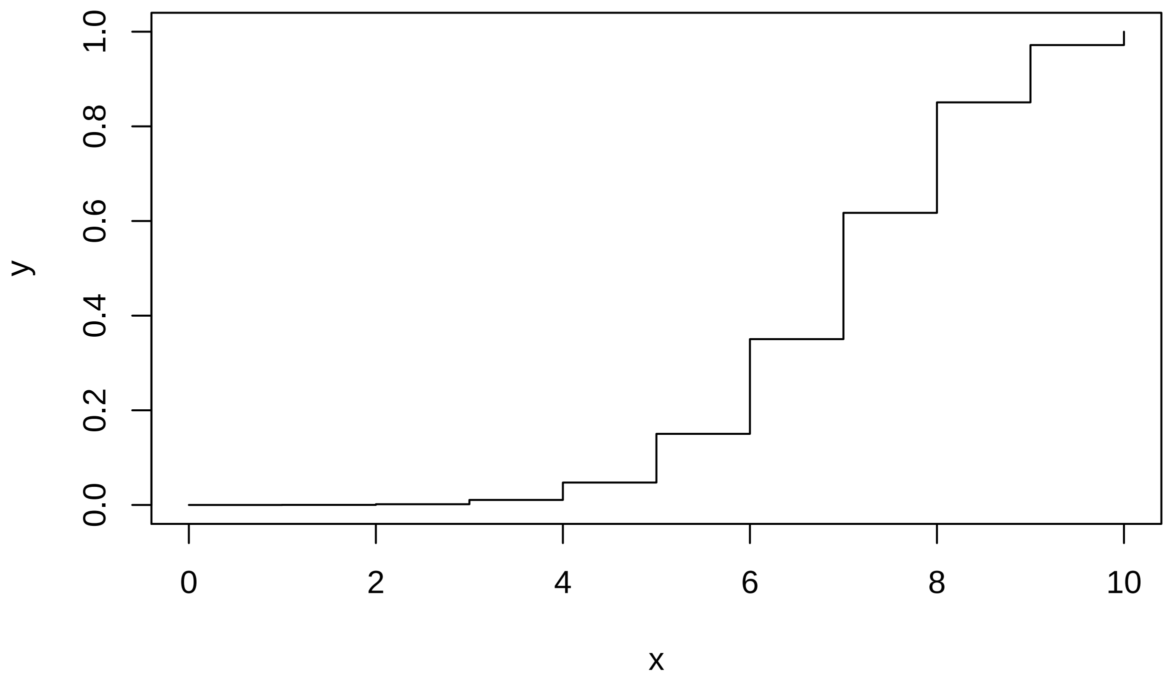

```r
y = pbinom(x, 10, 0.3)
plot(x, y, type = 's') #cdf of Bin(10, 0.3)
```




```
y = dbinom(x, 10, prob = 0.7)
plot(x, y) # 左偏
```




```
y = pbinom(x, 10, 0.7)
plot(x, y, type = 's') # 比較慢才跑到 1
```



Uniform Dist.

```r
dunif(x = 0.3, min = 0, max = 1) #give pdf of U(0,1) # 任何在上下界內的 x density 都一樣
```

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

```r
dunif(x = 0.03, min = 0, max = 1) #note that, this is NOT Prob.
```

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

```r
dunif(x = 0.003, min = 0, max = 1)
```

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

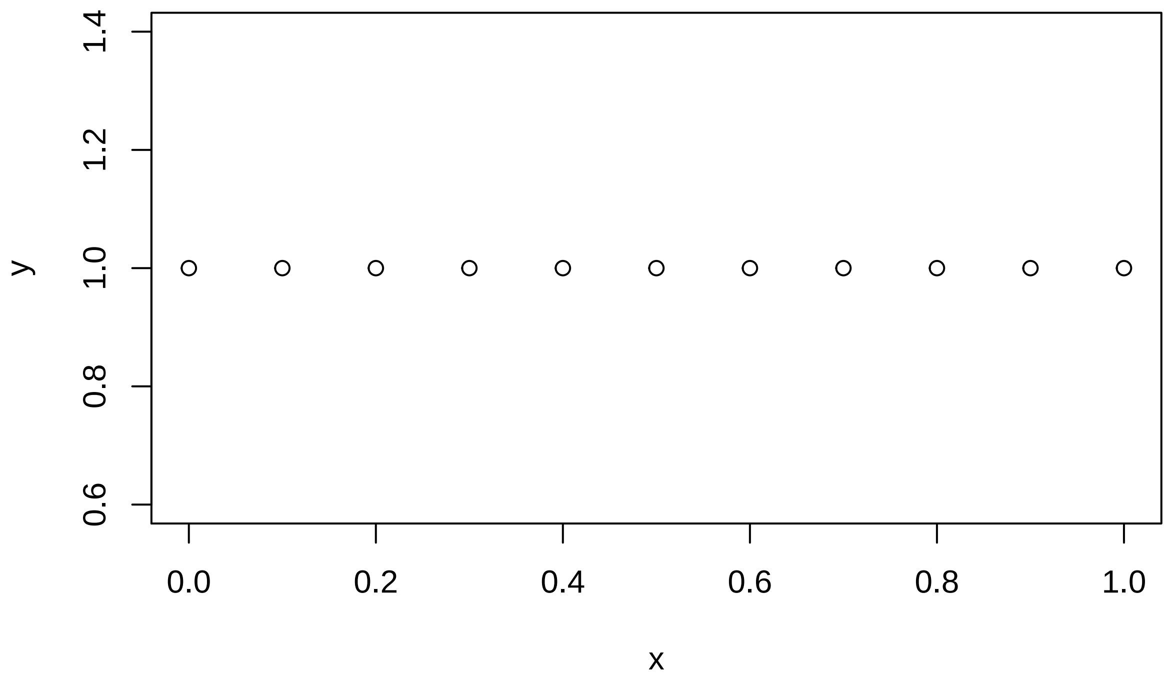
```r
dunif(2, 0, 1) # 不在上下界的 x 就會是 0
```

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

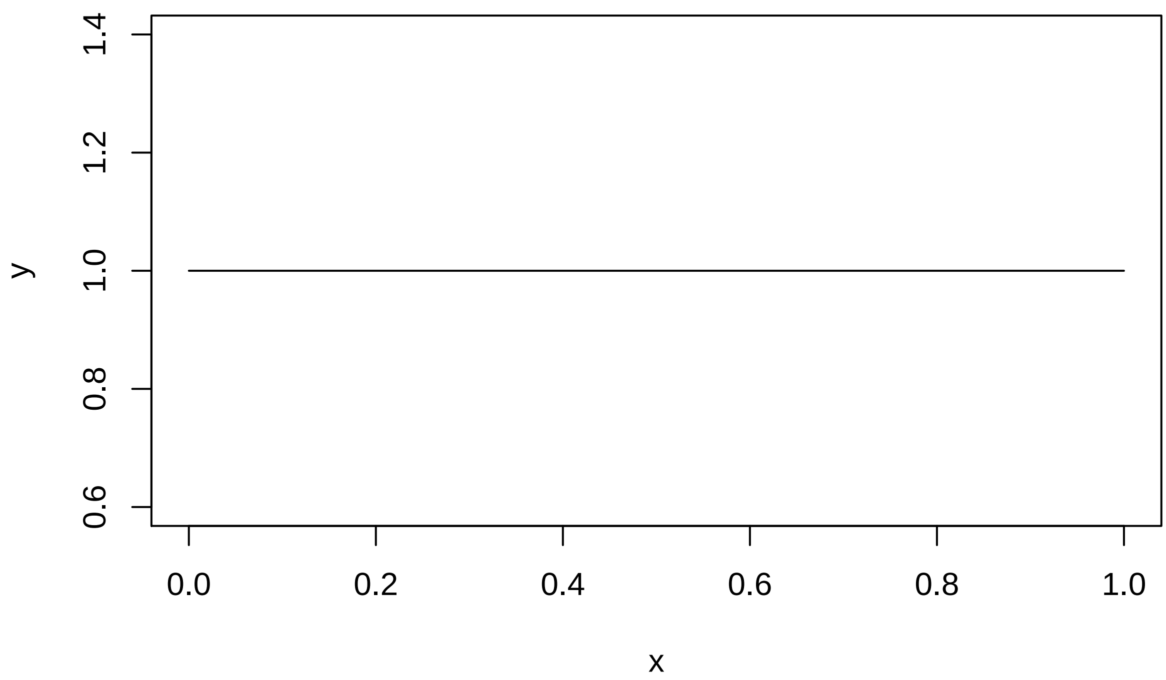
```r
dunif(x = 1, min = 0, max = 100) #height=1/(upper-lower)
```

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

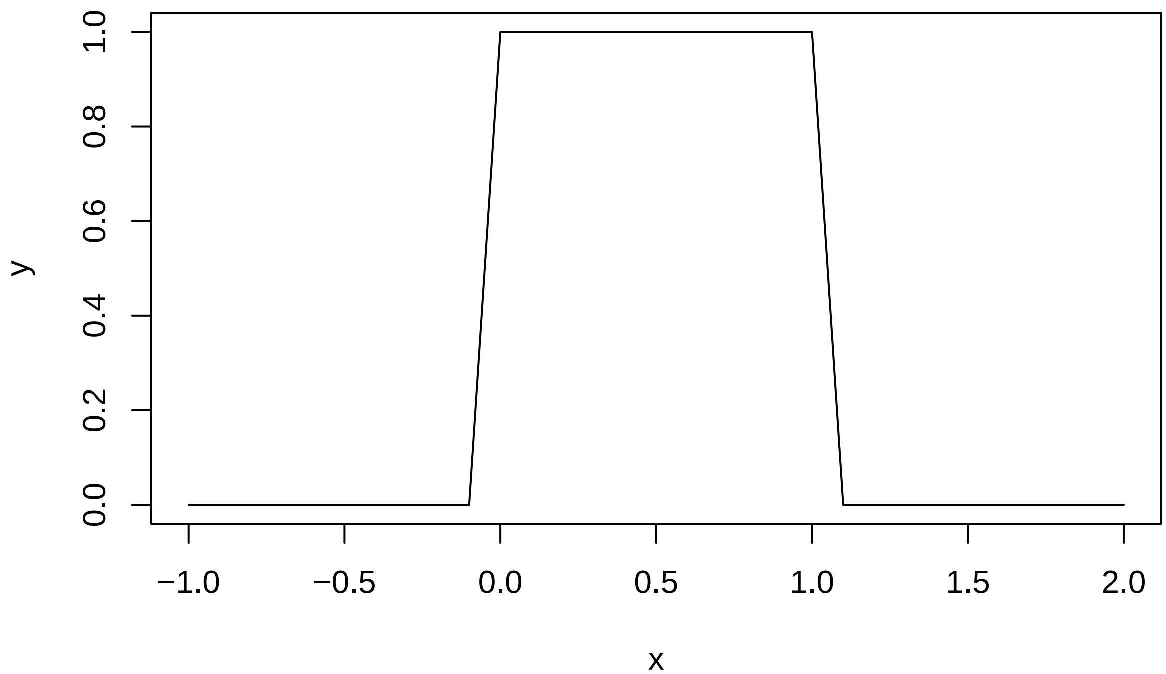
```r
#plot the pdf of U(0, 1)
x = seq(0, 1, by = 0.1) #10+1 pts
y = dunif(x, min = 0, max = 1) #y = x*1/(upper-lower)
plot(x, y, type = 'p') #plot with points
```



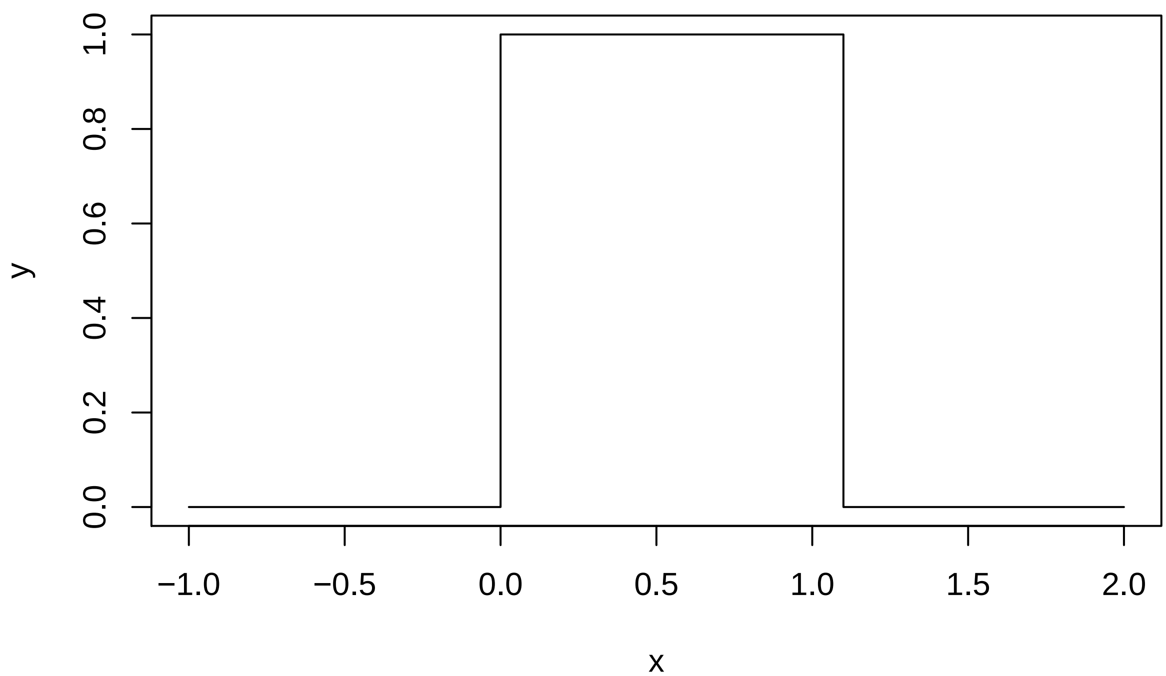

```r
plot(x, y, type = 'l') #plot with lines
```



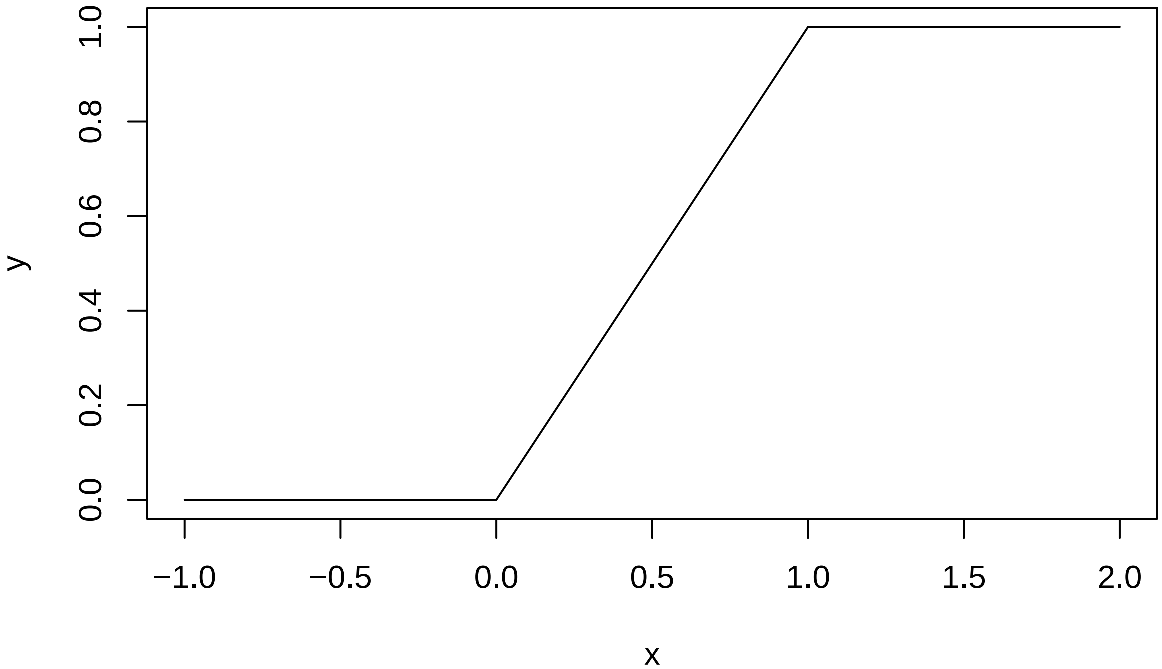

```r
#expand the range of X, not only supp(X)
x = seq(-1, 2, by = 0.1) #30+1 pts
y = dunif(x, min = 0, max = 1)
plot(x, y, type = 'l') #zoom in and see sth strange...
```




```r
plot(x, y, type = 's') #plot with jumping stairs
```




```r
#plot the cdf of U(0,1)
y = punif(x, 0, 1)
plot(x, y, type = 'l') # 斜率是 1/(upper-lower)
```




```r
##R.V. - generating random numbers
set.seed(9487) # 設定亂數產生器的起始值，使具有重現性
runif(n = 10, min = 0, max = 1) #give 10 random numbers from U(0, 1)
```

```
##  [1] 0.23417131 0.50284959 0.75148000 0.21552222 0.02665175 0.05534443
##  [7] 0.54685807 0.72360818 0.44475163 0.44919418
```

```r
rnorm(10, mean = 0, sd = 1) #give 10 random numbers from N(0, 1)
```

```
##  [1]  1.6639615  2.3295673  1.5629792 -0.9226910 -1.2932827  0.7322225
##  [7]  0.2592743  0.1011436  0.4548977  0.8561027
```

```r
set.seed(1234)
rbinom(n = 1, size = 10, prob = c(0.5, 0.5))
```

```
## [1] 3
```

```r
set.seed(1234)
sum(sample(x = c(0, 1), size = 10, replace = T, prob = c(0.5, 0.5)))
```

```
## [1] 3
```

```r
set.seed(1234)
sum(replicate(expr = sample(x = c(0, 1), size = 1, replace = T, prob = c(0.5, 0.5)), n = 10))
```

```
## [1] 3
```

Functions

來寫一個幫大家調分的函數

```r
stupid_score_modifier = function(input){
  output = input +1
  return(output)
}
stupid_score_modifier = function(score){
```

```
  score = score +1
  return(score)
}
stupid_score_modifier(3)
```

```
## [1] 4
```

來改良一下，只調一分太沒意思

```
smart_score_modifier = function(score){
  score = sqrt(score)*10
  return(score)
}
smart_score_modifier(1)
```

```
## [1] 10
```

現在有全班 30 個人的成績：

```
score0 = sample(10:100, 30) #10 到 100 分隨便抽 30 個成績出來
score0
```

```
##  [1] 25 13 95 79 88 87 23 65 71 99 91 30 49 93 76 14 75 56 96 57 12 50 41
## [24] 51 52 11 63 58 47 98
```

```
summary(score0)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   42.50   57.50   58.83   85.00   99.00
```

來看看調完分後

```
score1 = smart_score_modifier(score0)
summary(score1) # 可以看到分布改變，但這公平嗎？
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   33.17   65.16   75.83   73.90   92.18   99.50
```

```
max(score0)
```

```
## [1] 99
```

假設我想讓調分是個平移，而且平移到滿分（或特定分數）為止

```
score_modifier = function(score, highestGrade){
  shift = highestGrade - max(score)
  score = score + shift
  return(score)
}
score_modifier(score0, 90)
```

```
##  [1] 16  4 86 70 79 78 14 56 62 90 82 21 40 84 67  5 66 47 87 48  3 41 32
## [24] 42 43  2 54 49 38 89
```

或者我想讓低於中位數或 x 分的不調，其餘低於 60 的調到及格，超過 60 的調一樣的幅度，滿分還是滿分

```
score0[score0<50]
```

```
##  [1] 25 13 23 30 49 14 12 41 11 47
```

```
score_modifier2 = function(score, threshold_X){
  belowX = score[score<threshold_X]
```

```r
  btwXto60 = score[score<60 & score>=threshold_X]
  above60 = score[score>60]
  print(" 低於門檻分數"); print(belowX)
  print(" 可以調到及格分數"); print(btwXto60)
  print(" 最高調到滿分"); print(above60)
}
score_modifier2(score0, 50)
```

```
## [1] "低 於 門 檻 分 數"
##  [1] 25 13 23 30 49 14 12 41 11 47
## [1] "可 以 調 到 及 格 分 數"
## [1] 56 57 50 51 52 58
## [1] "最 高 調 到 滿 分"
##  [1] 95 79 88 87 65 71 99 91 93 76 75 96 63 98
```

```r
score_modifier2 = function(score, X){
  shift = 60 - min(score[score<60 & score>=X])
  for(i in 1:length(score)){
    if(score[i]<X){
      score[i] = score[i]
    }
    else if(score[i]>=X & score[i]<60){
      score[i] = 60
    }
    else{
      score[i] = score[i]+shift
    }
  }
  return(score)
}
score_modifier2(score0, 50)
```

```
##  [1]  25  13 105  89  98  97  23  75  81 109 101  30  49 103  86  14  85
## [18]  60 106  60  12  60  41  60  60  11  73  60  47 108
```

隨機實驗

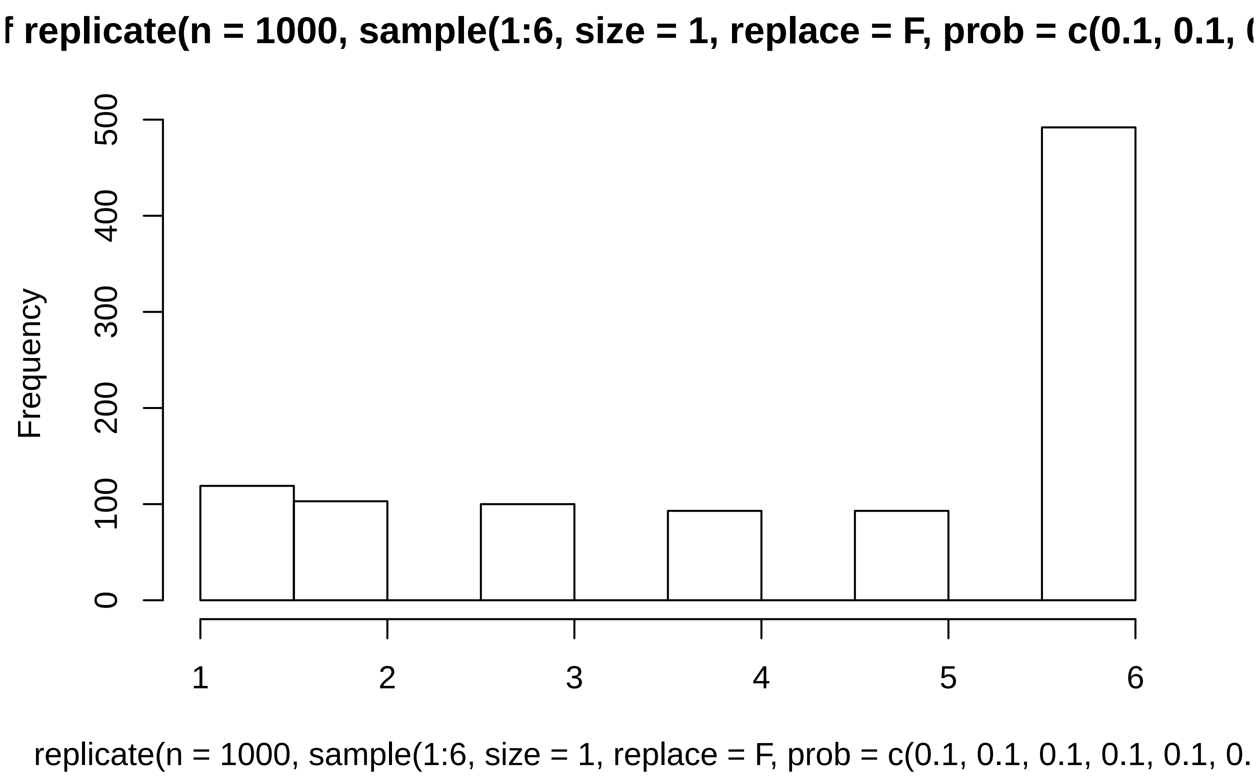
```r
set.seed(9487)
sample(1:6, size = 1, replace = F)
```

```
## [1] 3
```

```r
sample(1:6, size = 1, replace = F, prob = c(0.1, 0.1, 0.1, 0.1, 0.1, 0.5))
```

```
## [1] 3
```

```r
hist(replicate(n = 1000, sample(1:6, size = 1, replace = F, prob = c(0.1, 0.1, 0.1, 0.1, 0.1, 0.5))))
```



toss a coin a binomial appoach to approximate normal

```r
set.seed(9487)
toss = sample(c("H", "T"), size = 10, replace = T, prob = c(0.5, 0.5))
toss
```

```
##  [1] "T" "H" "H" "T" "T" "T" "H" "H" "T" "T"
```

```r
toss[] == 'T'
```

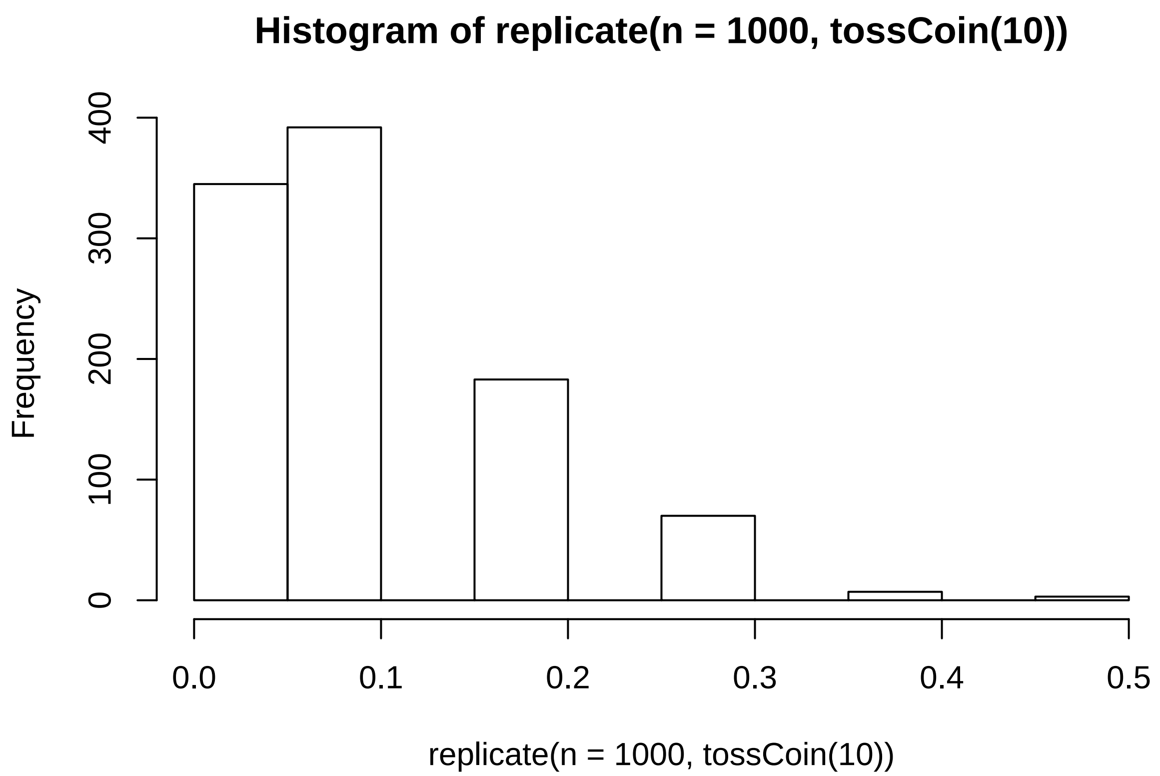
```
##  [1]  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
```

```r
sum(toss[] == 'T')
```

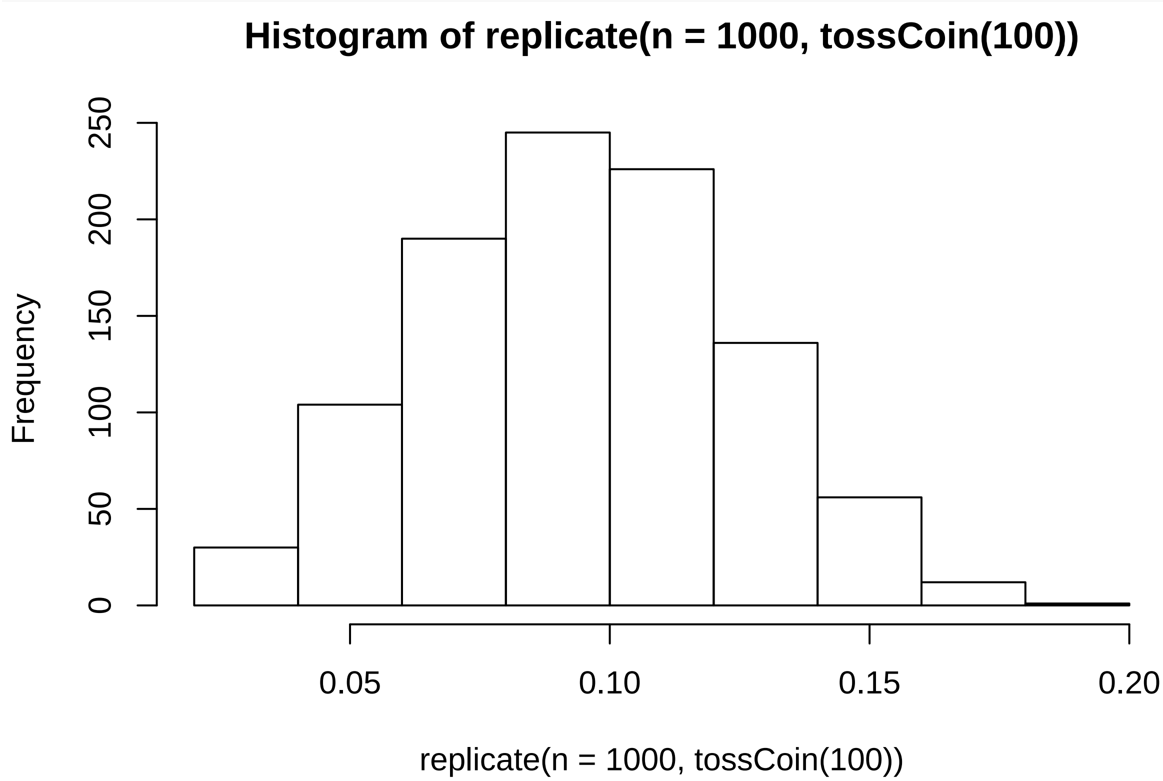
```
## [1] 6
```

也可以寫一個 function 來擲硬幣

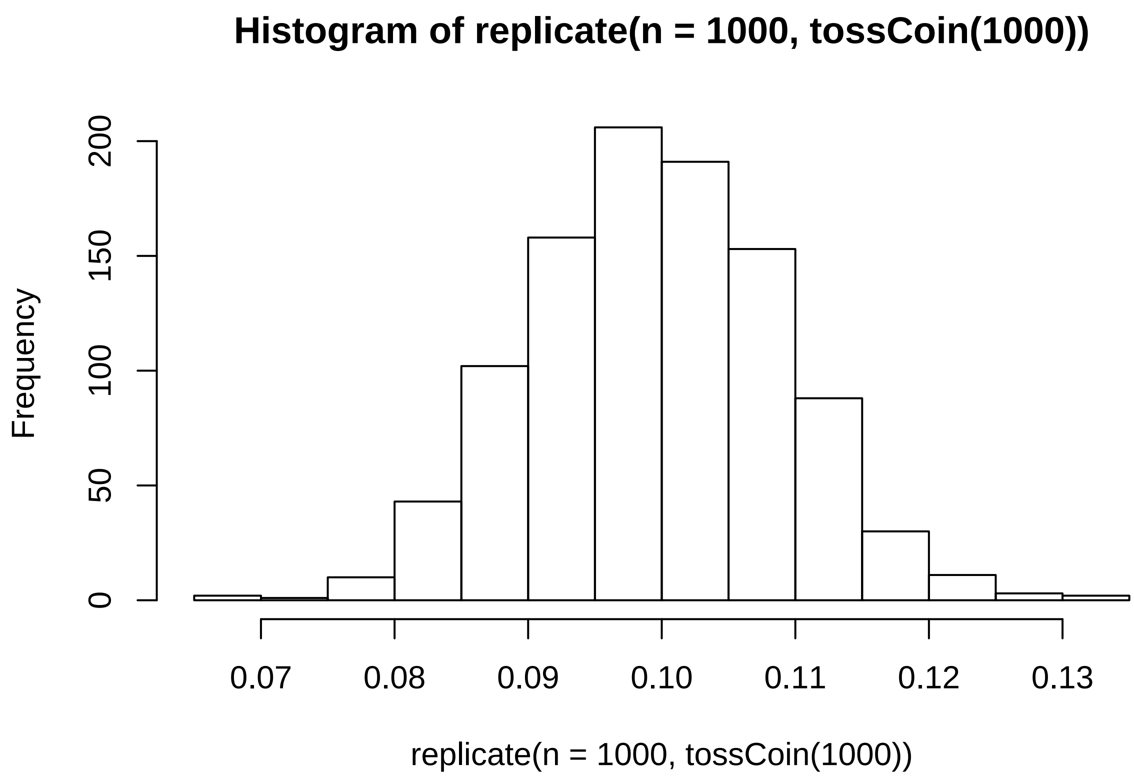
```r
tossCoin = function(n){
  toss = sample(c('H', 'T'), size = n, replace = T, prob = c(0.9, 0.1))
  return(sum(toss[] == 'T')/n)
}
hist(replicate(n = 1000, tossCoin(10)))#n = 10, do not converge to norm
```




```r
hist(replicate(n = 1000, tossCoin(100)))
```

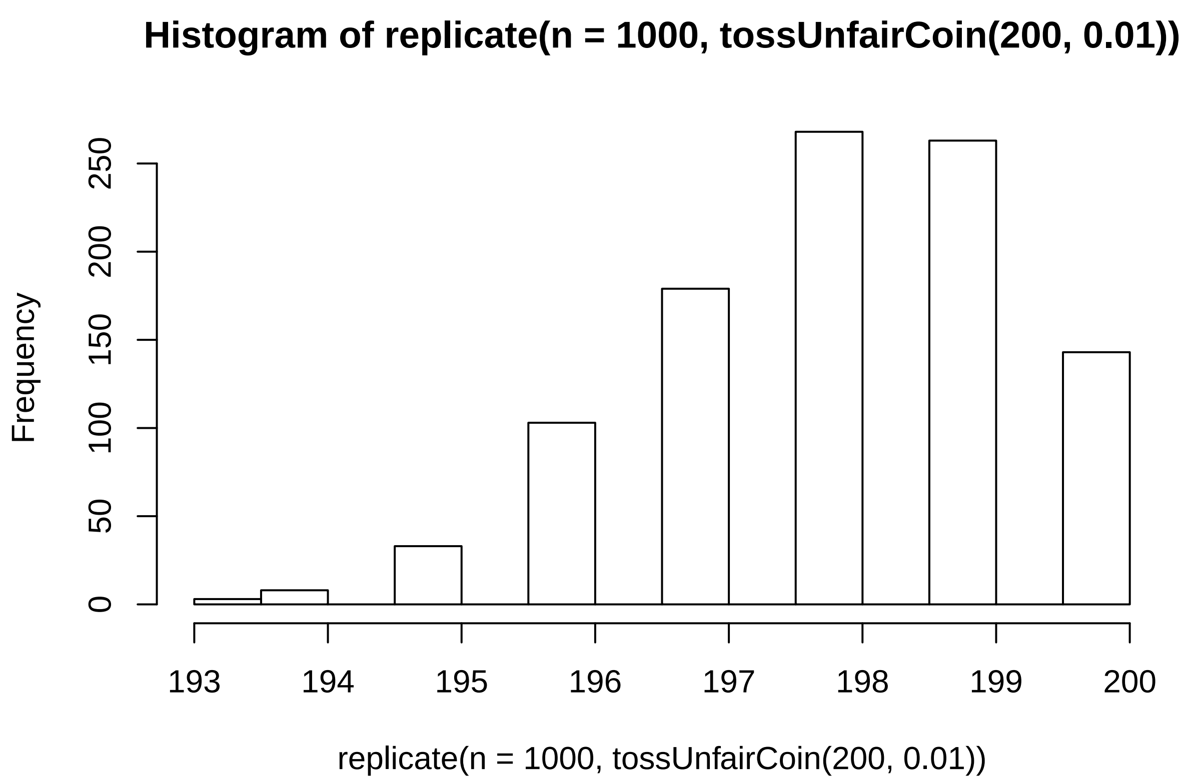



```r
hist(replicate(n = 1000, tossCoin(1000))) #var decrease
```

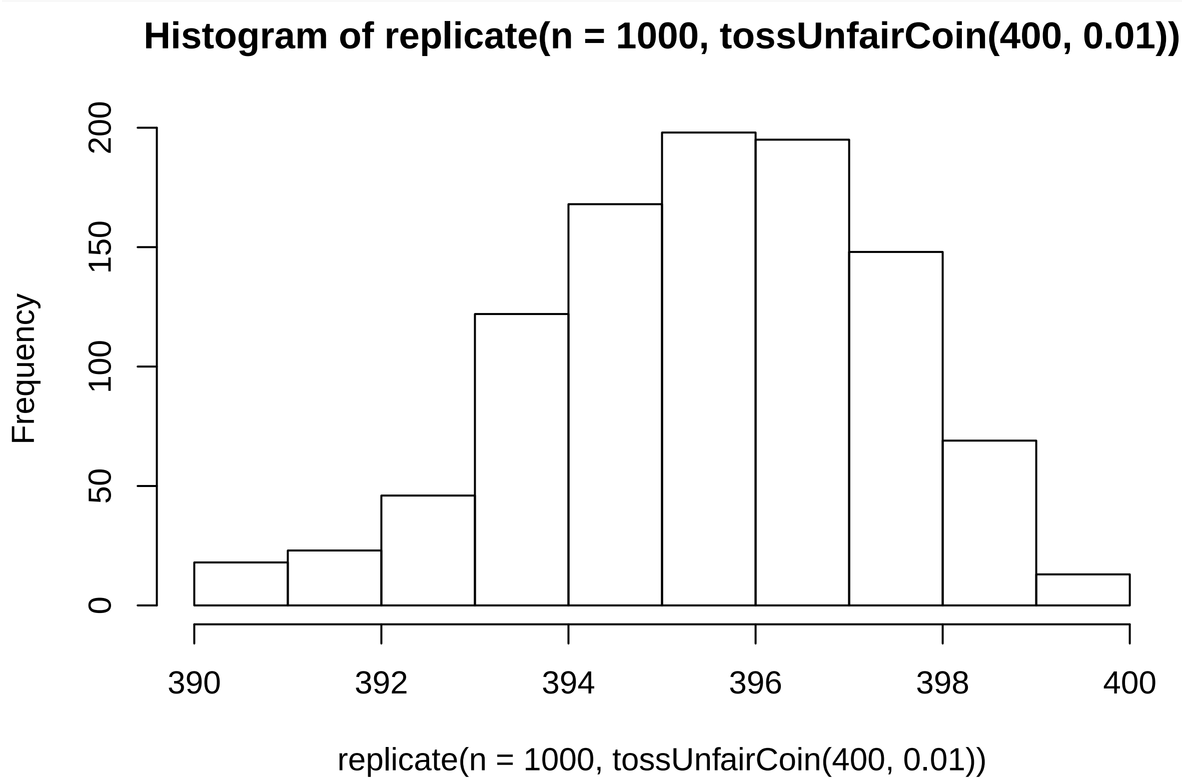


或擲一個不公正的硬幣

```
tossUnfairCoin = function(n, p){
  toss = sample(c('H', 'T'), size = n, replace = T, prob = c(p, 1-p))
  return(sum(toss[] == 'T'))
}
hist(replicate(n = 1000, tossUnfairCoin(200, 0.01))) # 左偏，尾巴在左邊
```

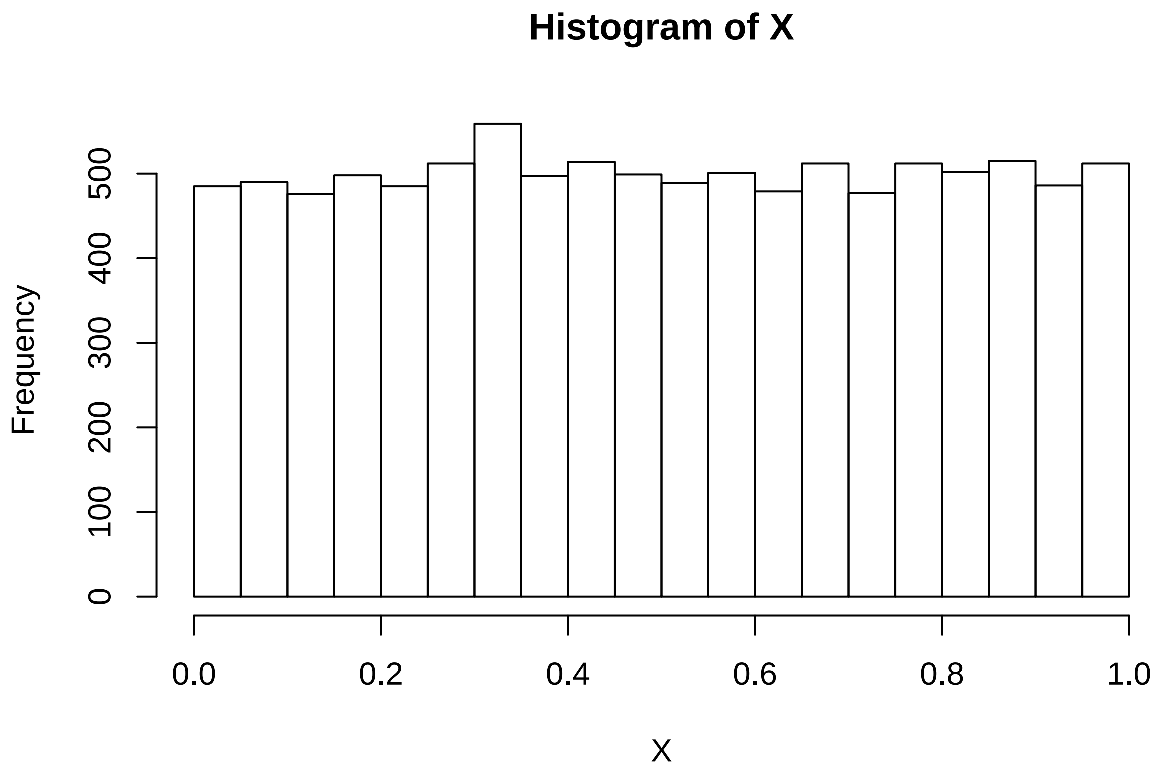



```r
hist(replicate(n = 1000, tossUnfairCoin(400, 0.01))) # 隨著重複次數增加，越容易收斂到常態
```

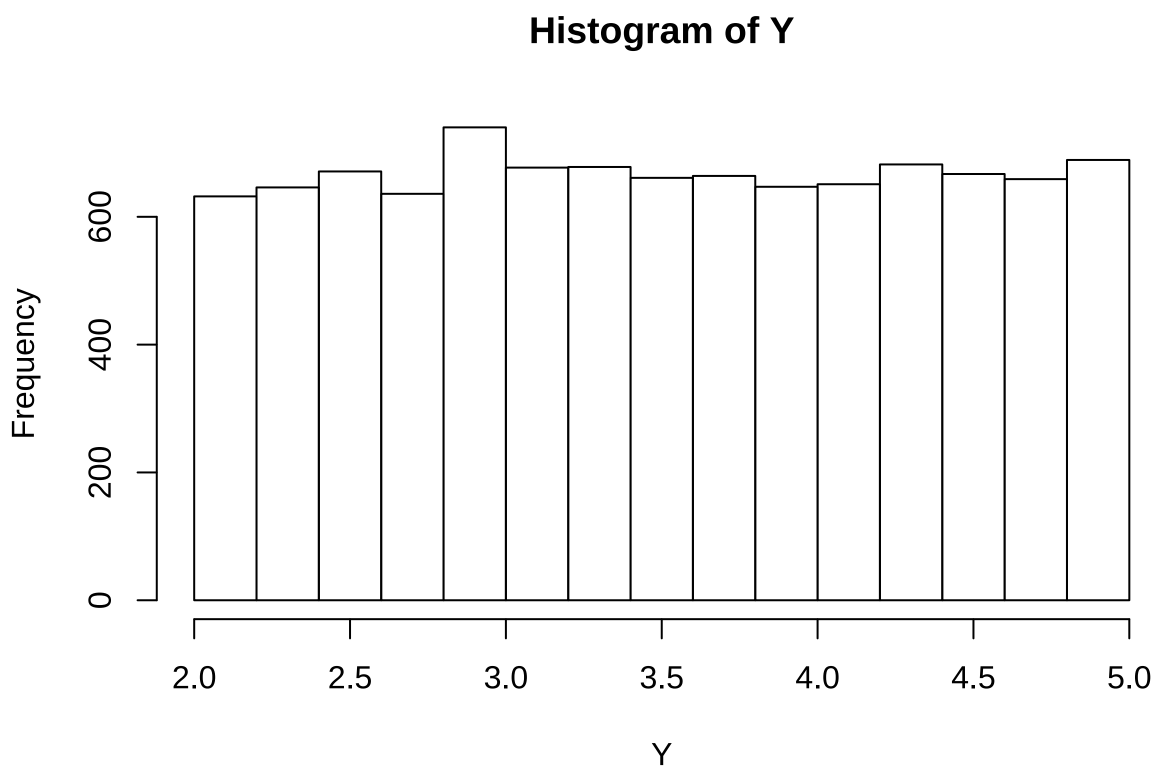


Optional

```r
##Shift U(0, 1) to U(a, b)
X = runif(10000, 0, 1)
hist(X)
```



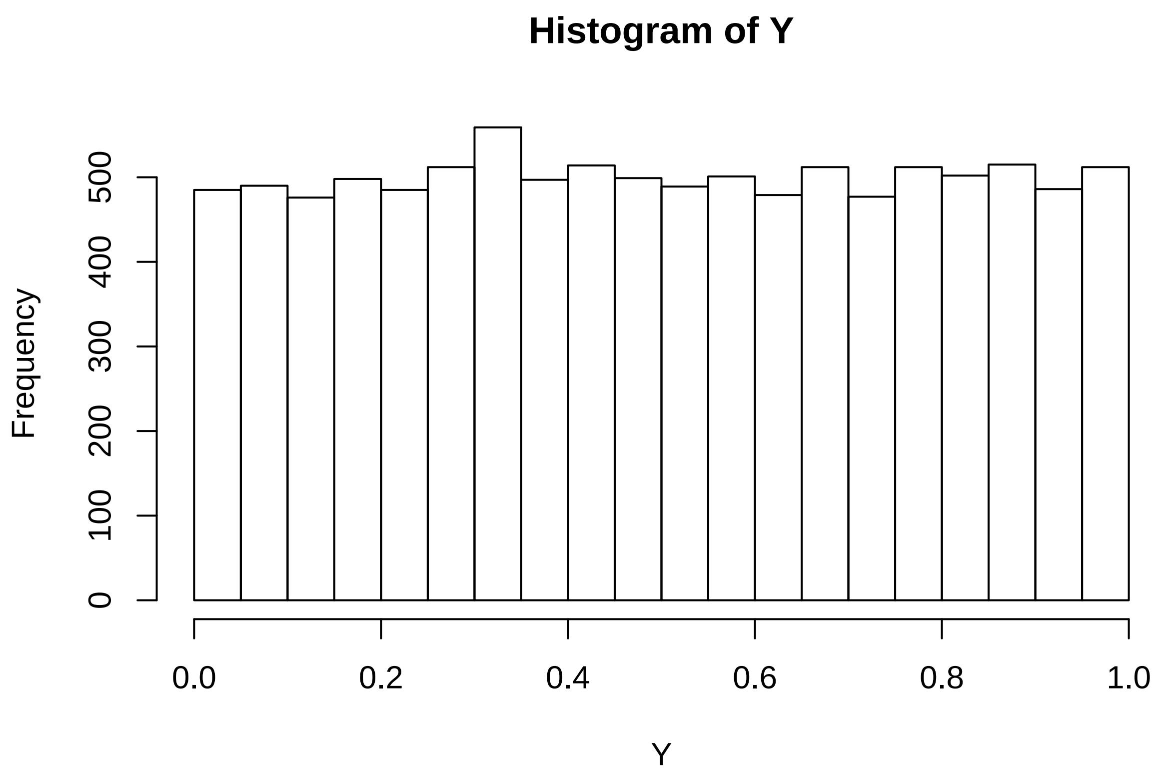

```r
Y = 3*X+2 #trivial
hist(Y) #look at the range of Y
```




```
#or
Y = punif(X) #plug in to its own cdf, note: F_X(x) = x, where X~U(0,1)
head(Y-X)
```

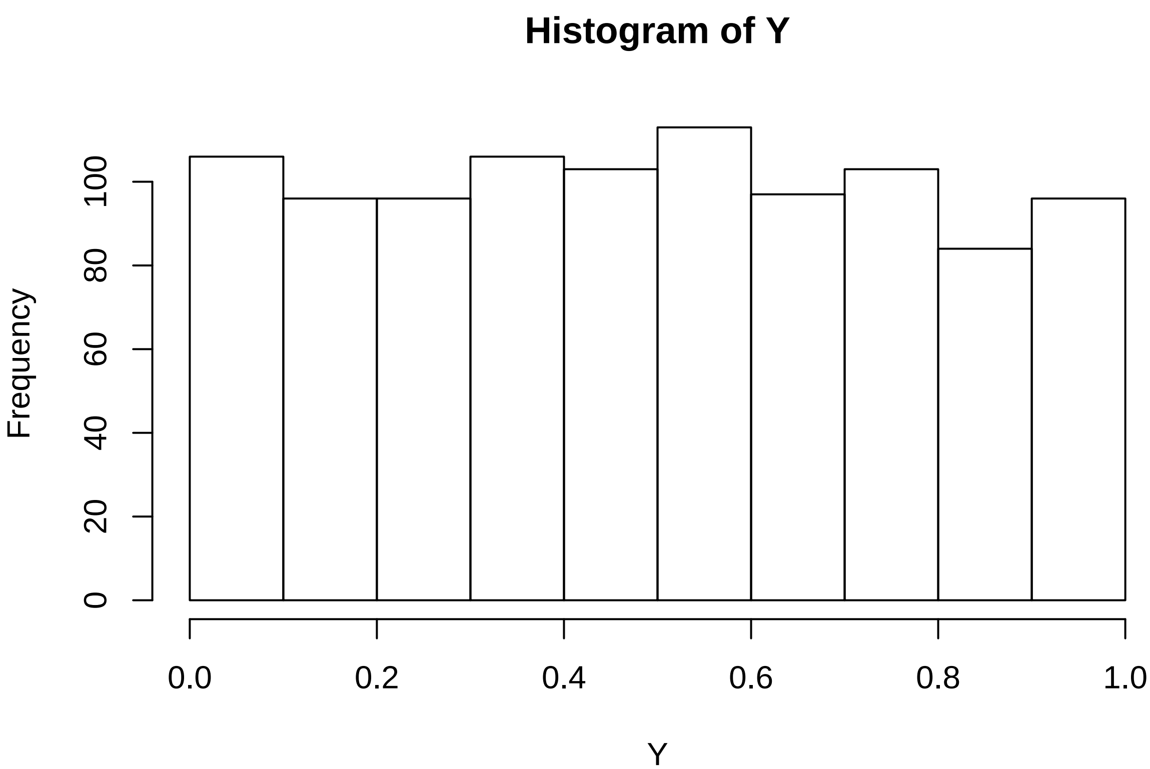
```
## [1] 0 0 0 0 0 0
```

```
hist(Y) #still being U(0,1)
```



How about U(a,b)? What happens after pluging in into its own cdf?

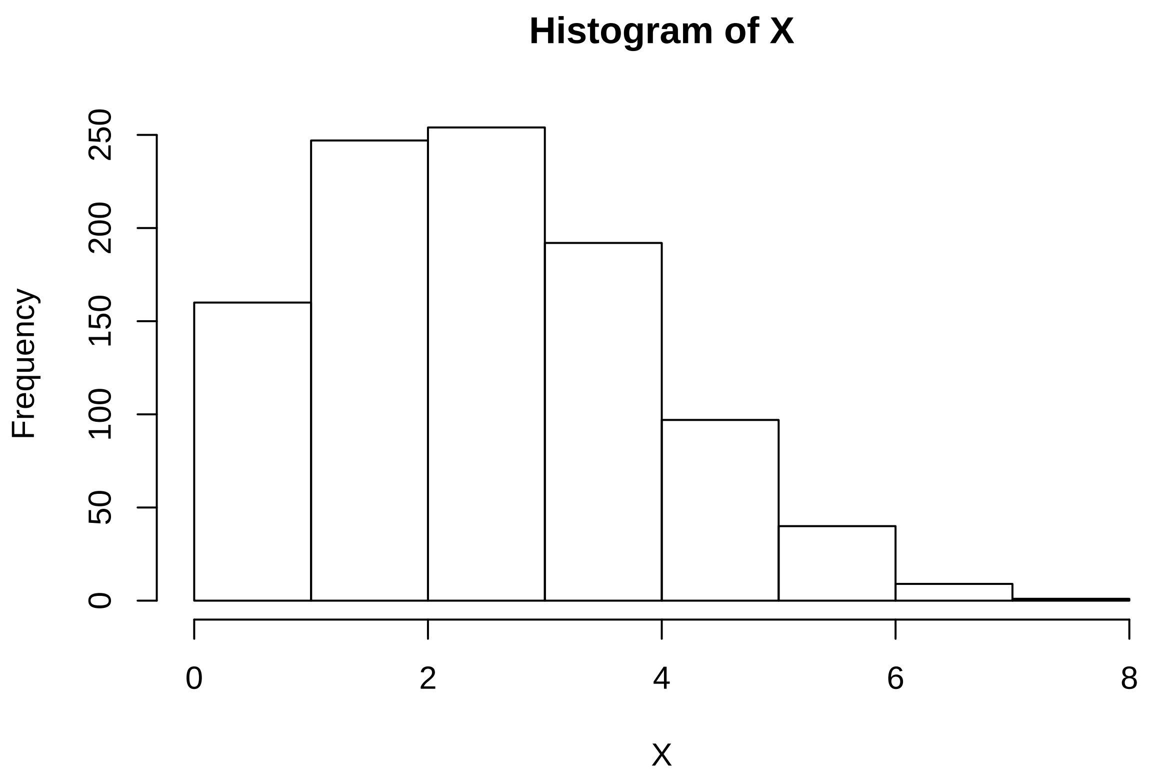
```
X = runif(1000, 2, 5)
Y = (X-2)/3 # 先平移後伸縮
hist(Y) #become U(0,1)
Y = punif(X, min = 2, max = 5) #Or I plug it in its own cdf
hist(Y) #Also become U(0,1)
```



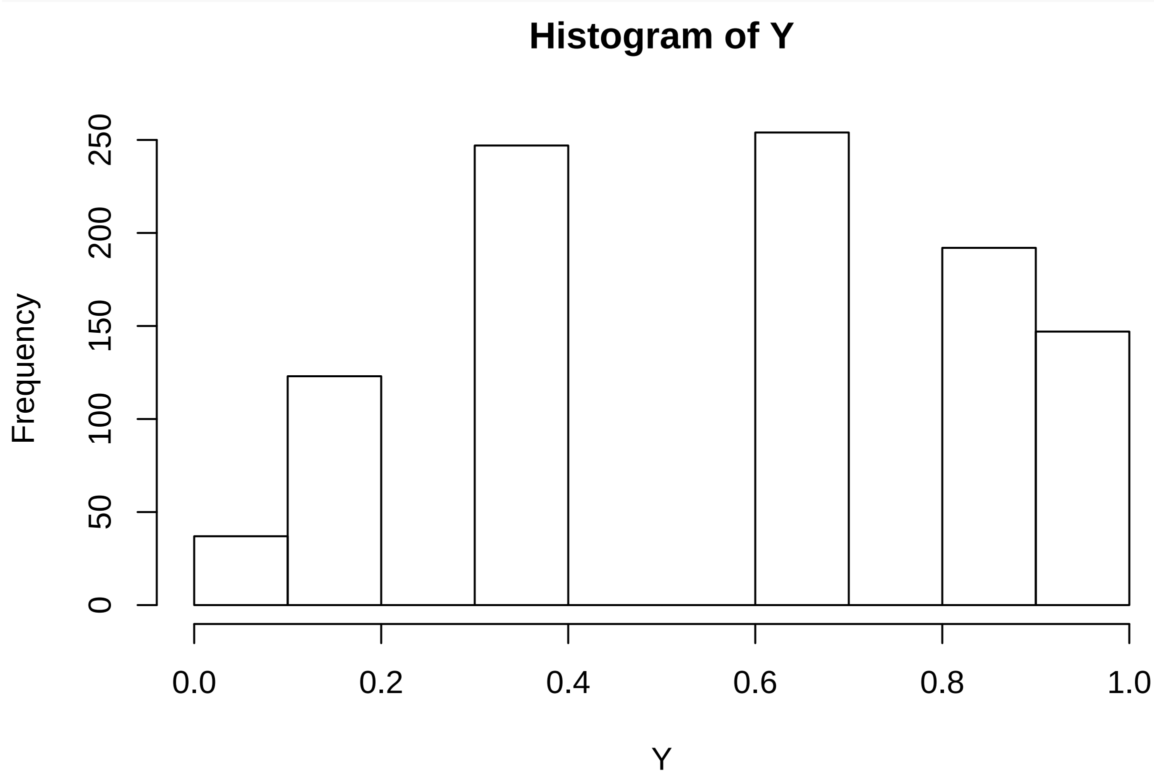
Does it hold on other r.v.?

X~Bin(n = 10, p = 0.3)

```
X = rbinom(n = 1000, size = 10, prob = 0.3)
hist(X)
```



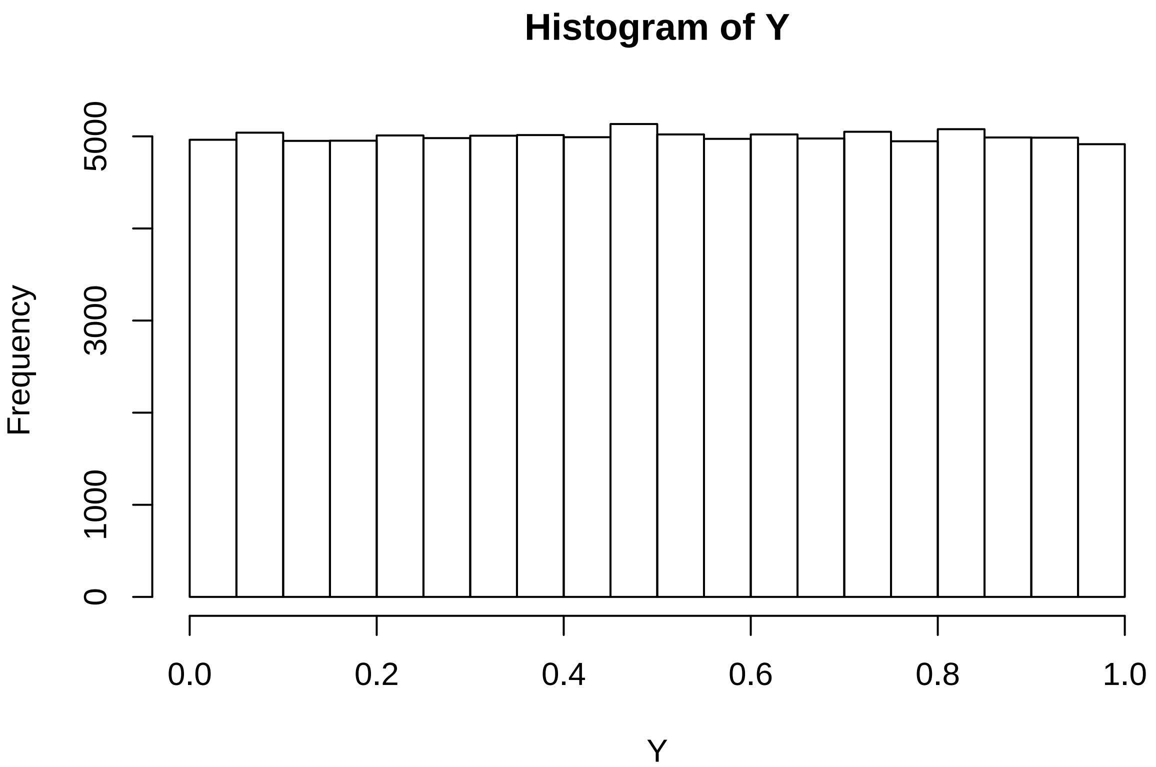

```
Y = pbinom(X, size = 10, prob = 0.3)
hist(Y) #not that clear, but we can see some pattern
```



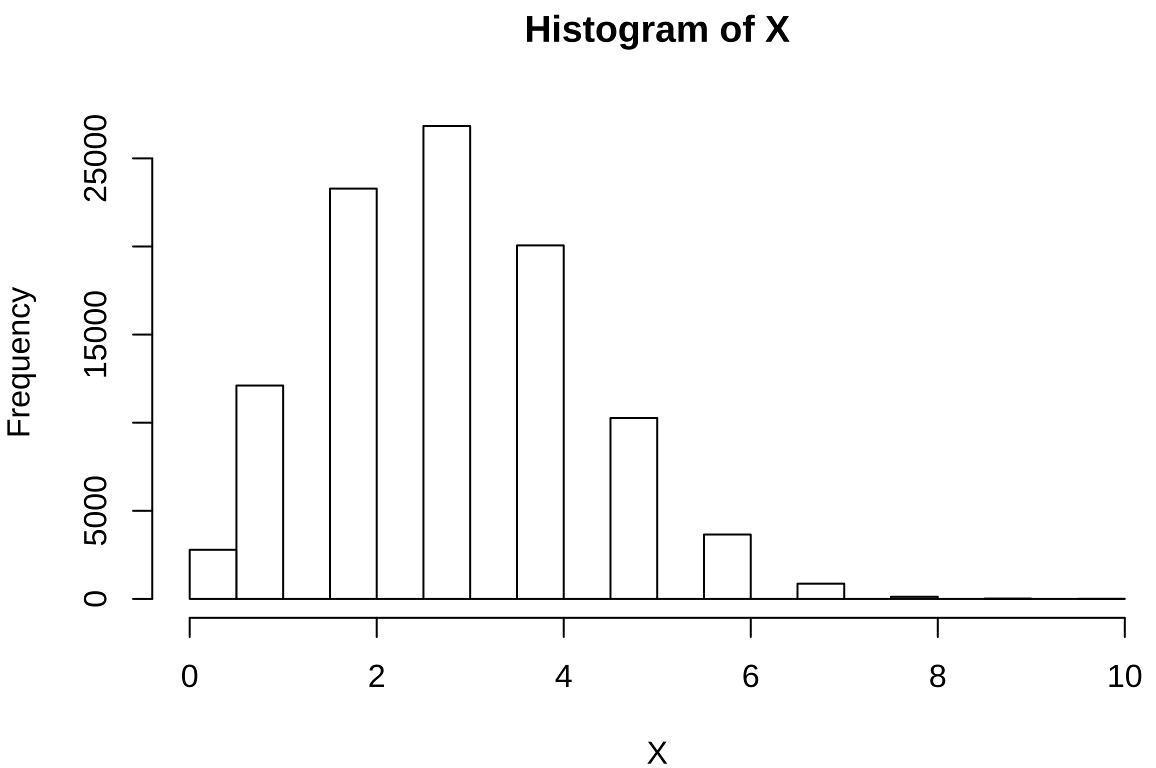

```
                    #the lower&upper bound of Y are (0,1)
```

Can we do another direction? Generating Bin(n = 10, p = 0.3) from U(0, 1)

```r
Y = runif(100000, 0, 1)
hist(Y)
```




```r
X = qbinom(Y, size = 10, prob = 0.3)
hist(X)
```



機率積分轉換 Probability Integral Transform

use U(0,1) & CDF of Standard Normal Dist to generate N(0, 1)

```r
#These are what we need
qnorm(0.95) #input Prob., return quantile i.e. inverse cdf
```

```
## [1] 1.644854
```

```r
pnorm(1.96 ,mean = 0, sd = 1) #cdf of N(0, 1)
```

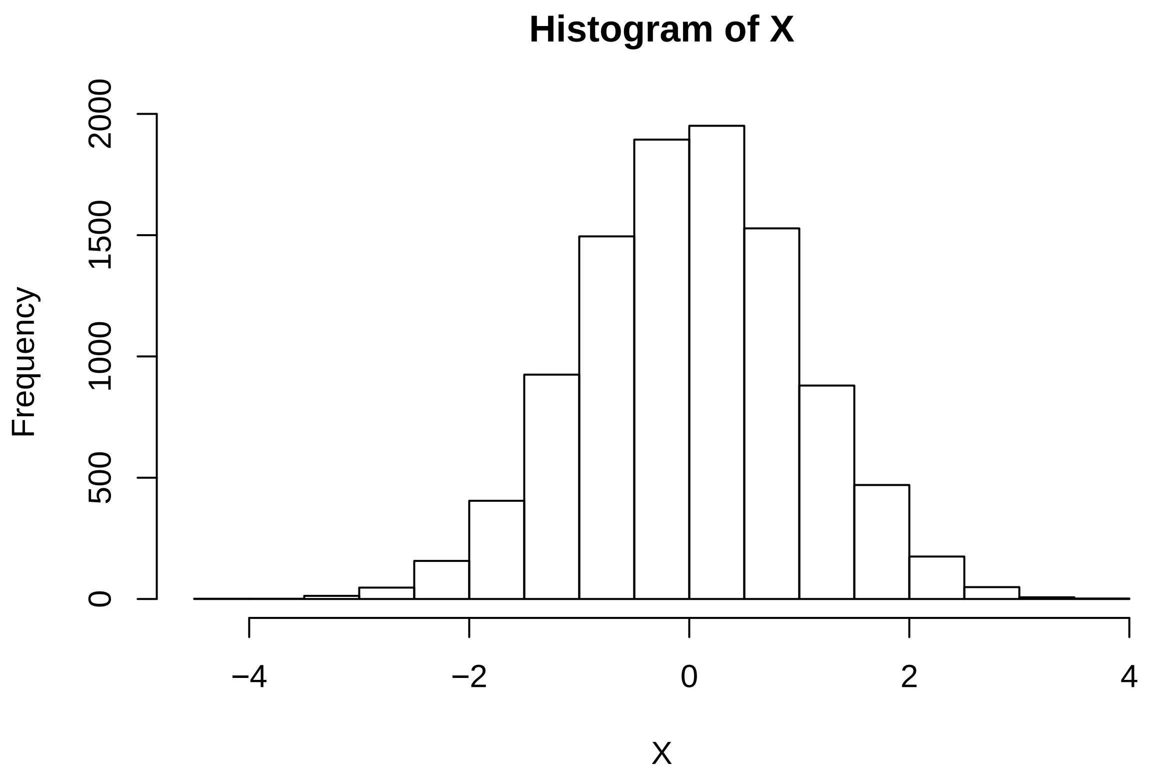
```
## [1] 0.9750021
```

```r
dnorm(10) #return pdf
```

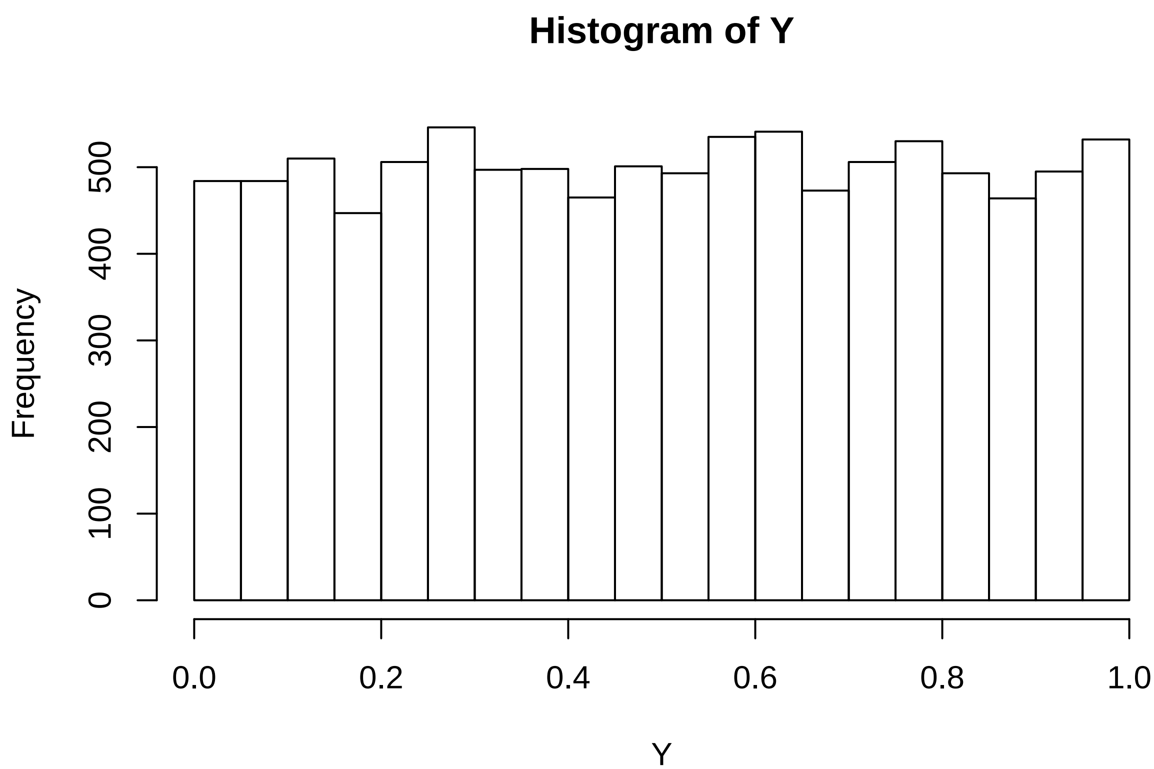
```
## [1] 7.694599e-23
```

By PIT, we can make all r.v. from N(0,1) to U(0,1)

```r
X = rnorm(10000, 0, 1)
hist(X)
```

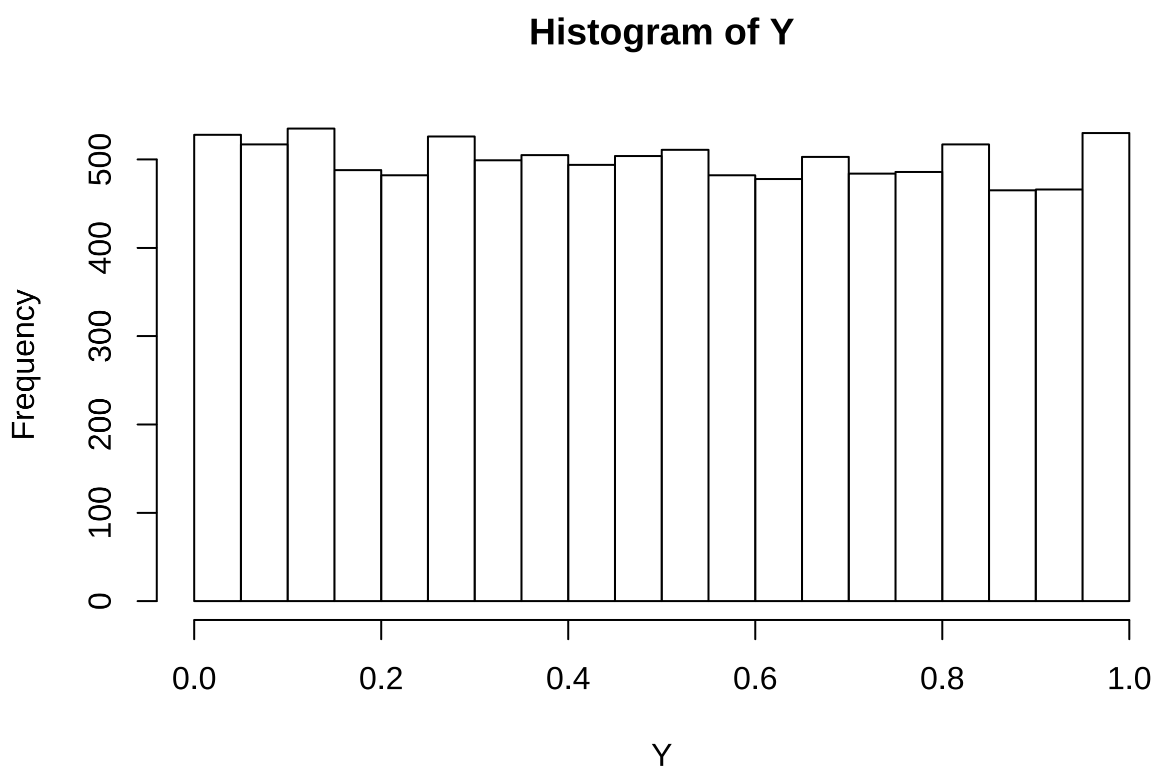



```r
Y = pnorm(X) #Y = Fx(X)
hist(Y)
```

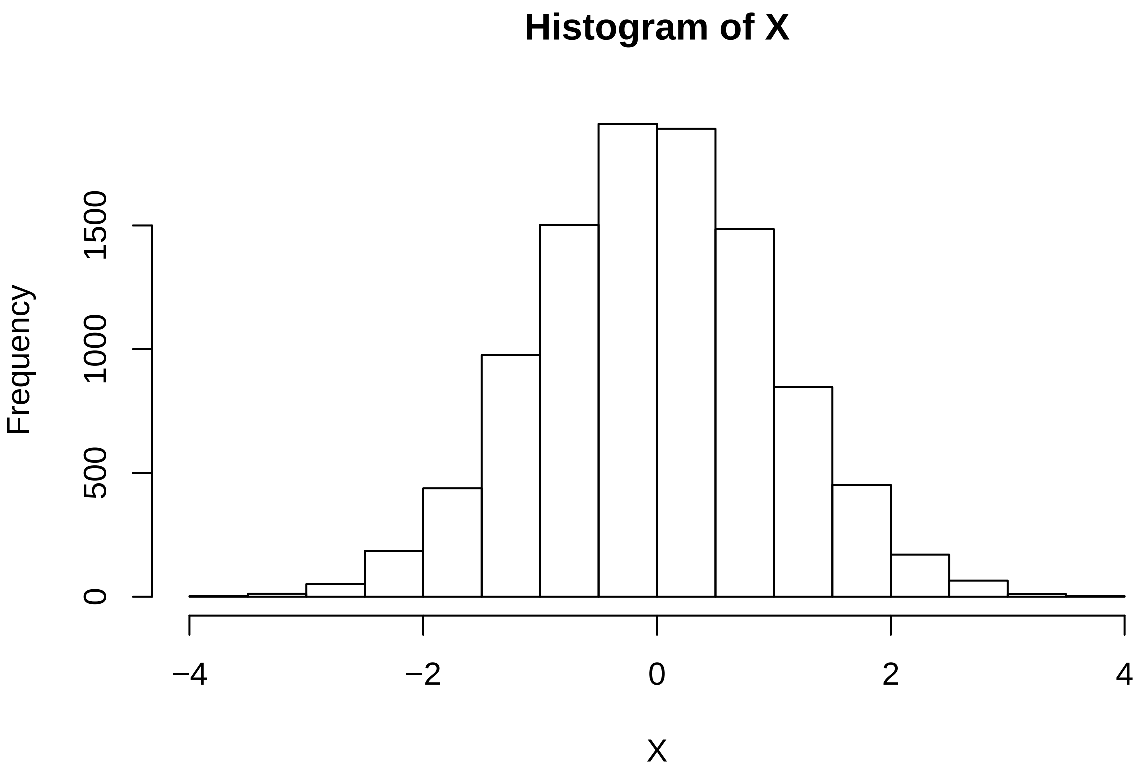


Also, we can have the inverse transform and generate N(0,1) form U(0,1)

```
Y = runif(10000, 0, 1)
hist(Y)
```

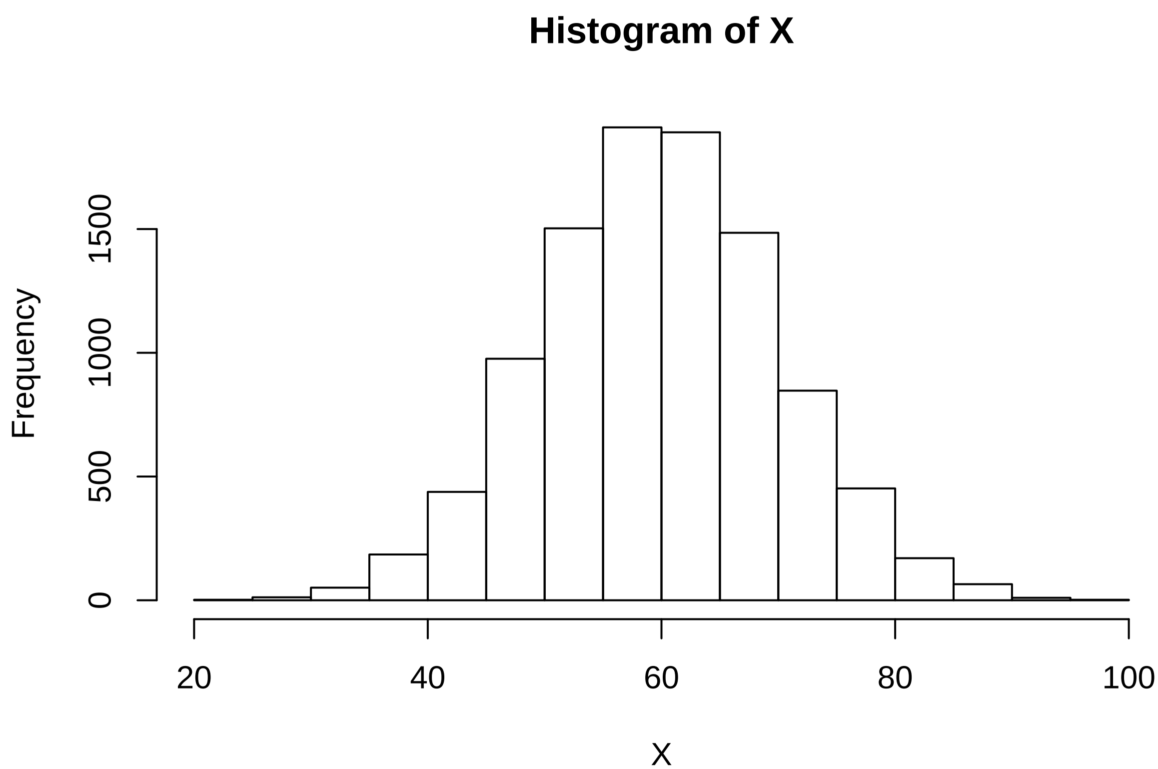


```r
X = qnorm(Y) #Fx^-1(Y) = X
hist(X)
```



if we want N(60, 10), just modify the qnorm

```r
X = qnorm(Y, mean = 60, sd = 10)
hist(X)
```



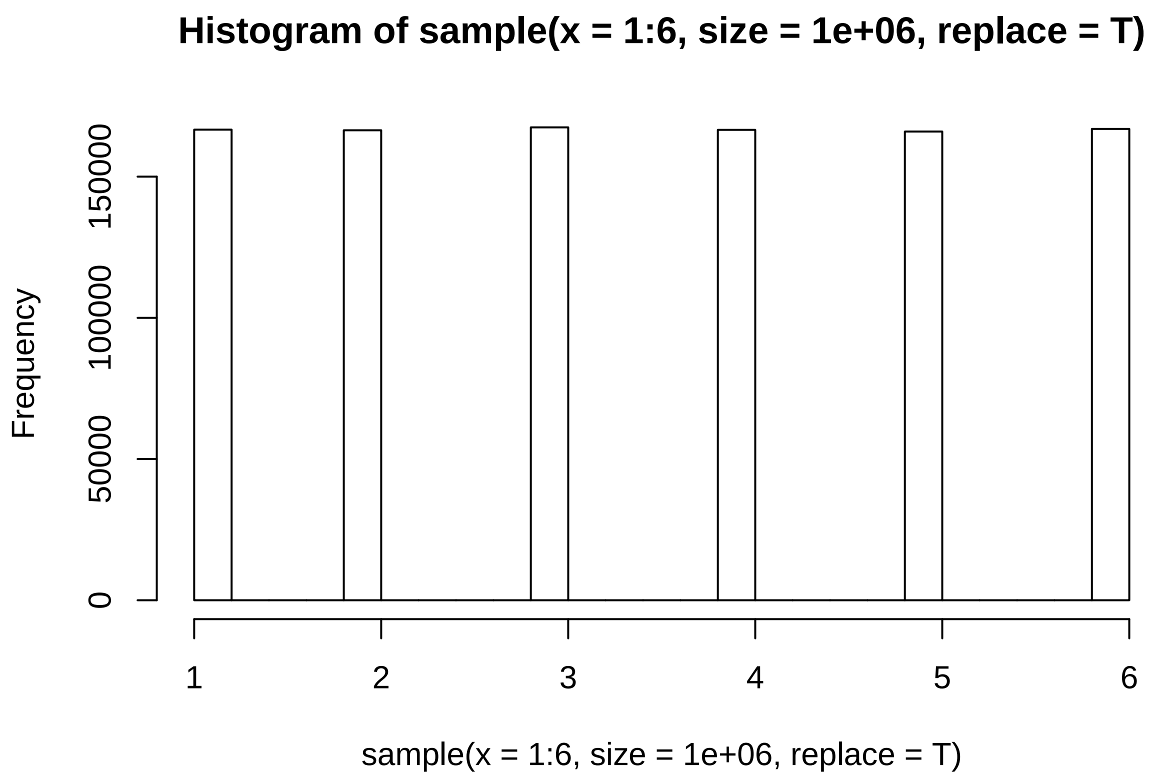
Practice 1

請用亂數模擬擲一顆骰子時所出現的點數 ans:

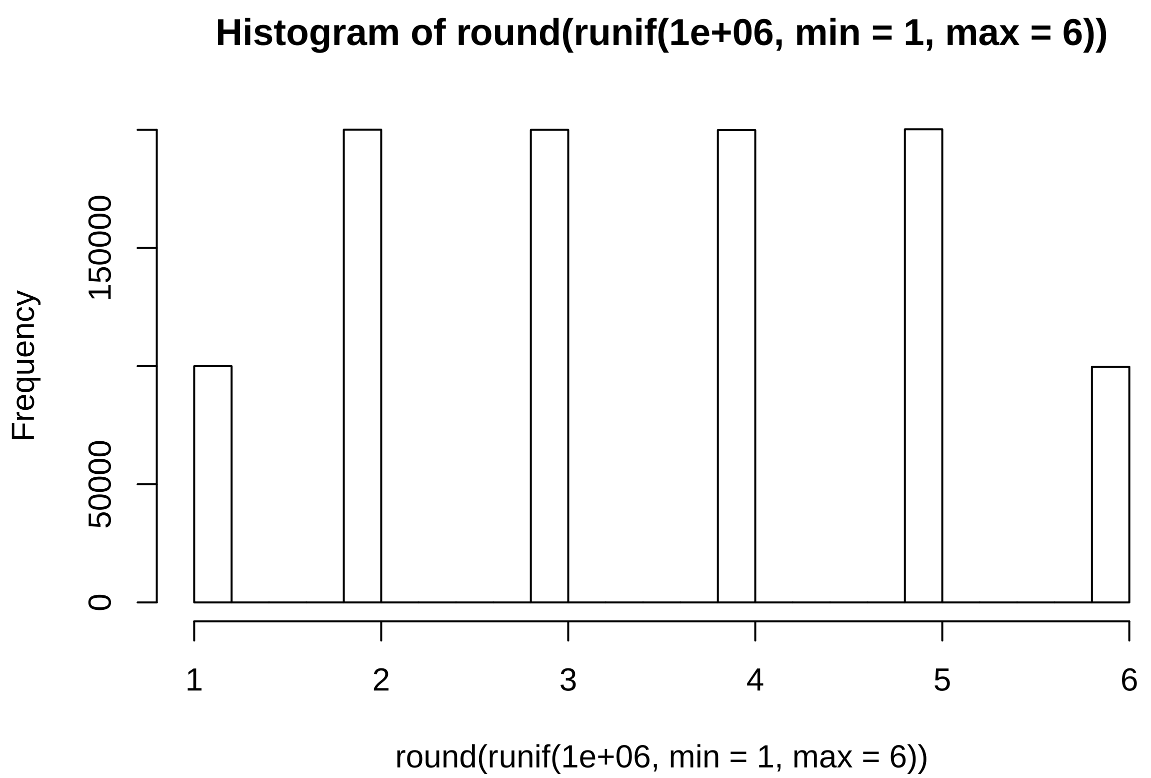
```r
sample(x = 1:6, size = 1)
```

```
## [1] 4
```

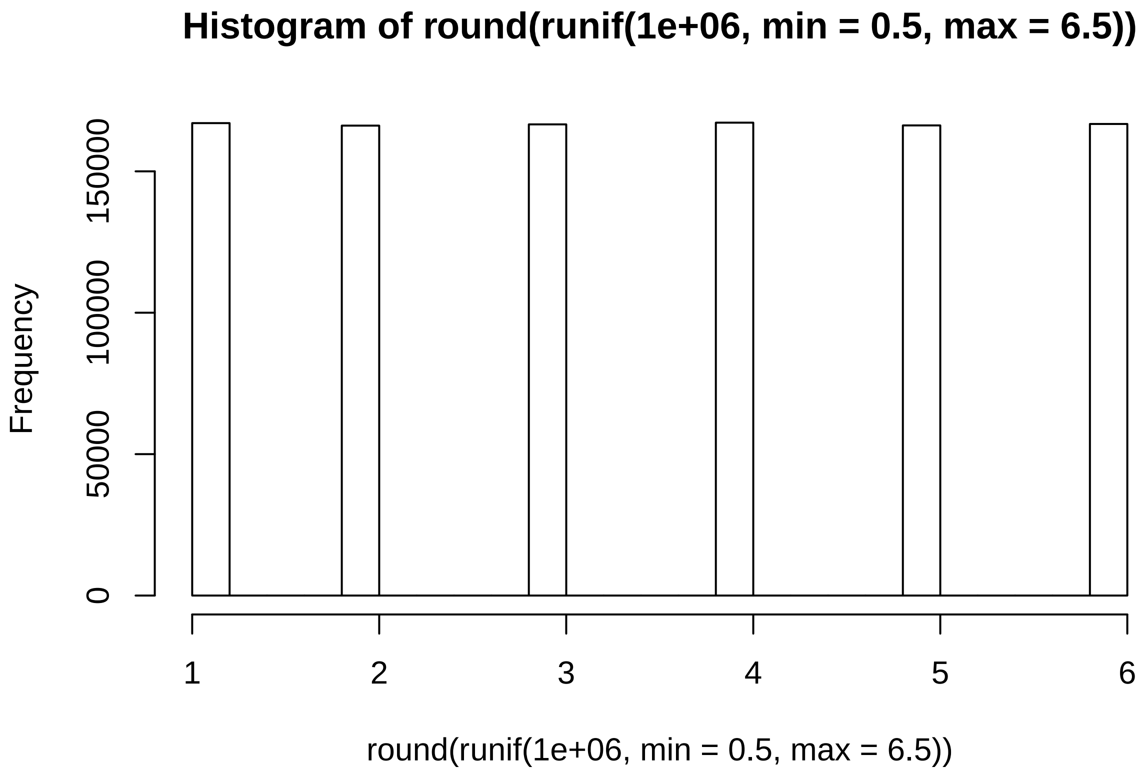
```r
hist(sample(x = 1:6, size = 1000000, replace = T))
```




```r
# 可以用 runif 嗎?
hist(round(runif(1000000, min = 1, max = 6))) #1&6 的 freq 明顯少很多
```



```r
hist(round(runif(1000000, min = 0.5, max = 6.5)))
```



Practice2

請試著以 sample() 模擬一個擲出不公正硬幣的隨機事件，其中擲出 Head 的機率為 0.9，擲出 Tail 的機率為 0.1，一次擲 10 枚並試著計算此 10 玫硬幣中共有幾枚為 Tail hint: use boolean operator and sum() to count the number of Tail let toss = samp(…); sum(toss[] == 'T') ans:

```r
sample(c('H', 'T'), size = 10, replace = T, prob = c(0.9, 0.1))
```

```
##  [1] "H" "T" "H" "H" "H" "H" "H" "H" "H" "H"
```

```r
toss = sample(c('H', 'T'), size = 10, replace = T, prob = c(0.9, 0.1))
sum(toss[] == 'T')
```

```
## [1] 2
```

Practice 3

請用 function 及亂數模擬擲 n= 1,2,3,4,5 顆骰子時所出現的點數和 ans:

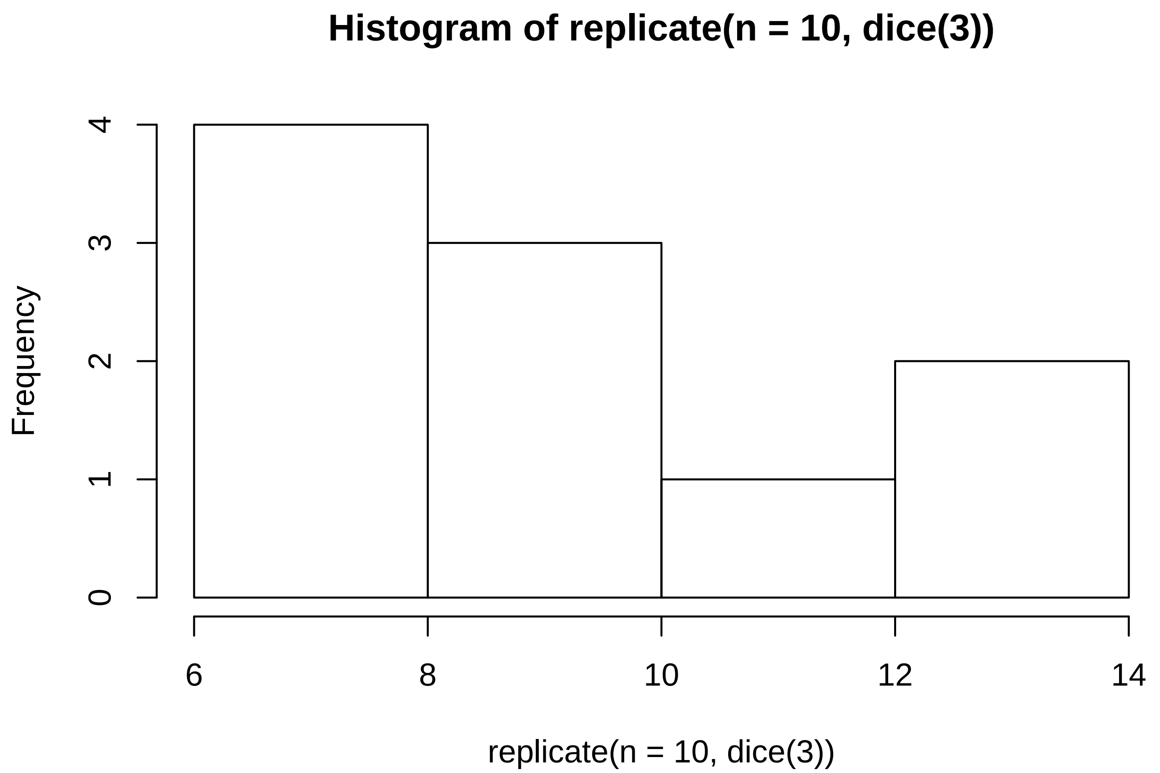
```r
dice = function(n){
  a = sample(x = 1:6, size = n, replace = T)
  return(sum(a))
}
rep(dice(1), 10) # 不能用 rep 函數，why?
```

```
##  [1] 5 5 5 5 5 5 5 5 5 5
```

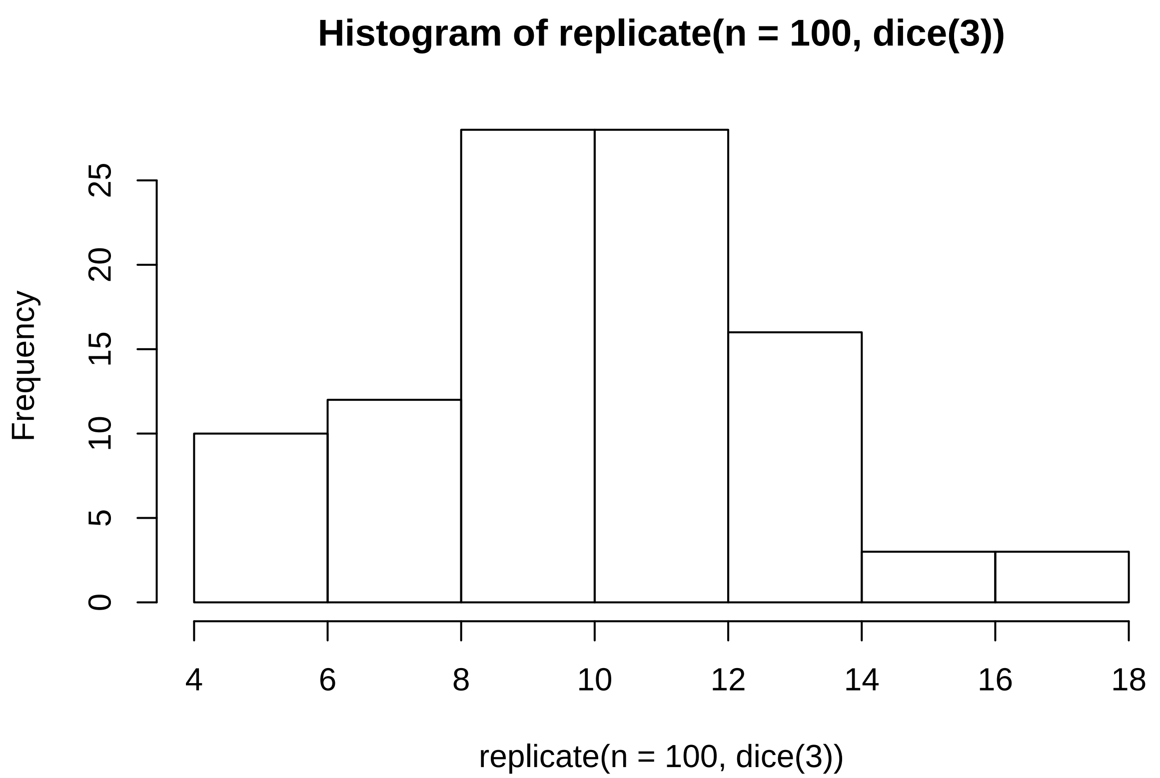
```r
replicate(n = 10, dice(1)) #replicate 函數才能重新運算
```

```
##  [1] 4 6 3 4 1 1 4 3 4 4
```

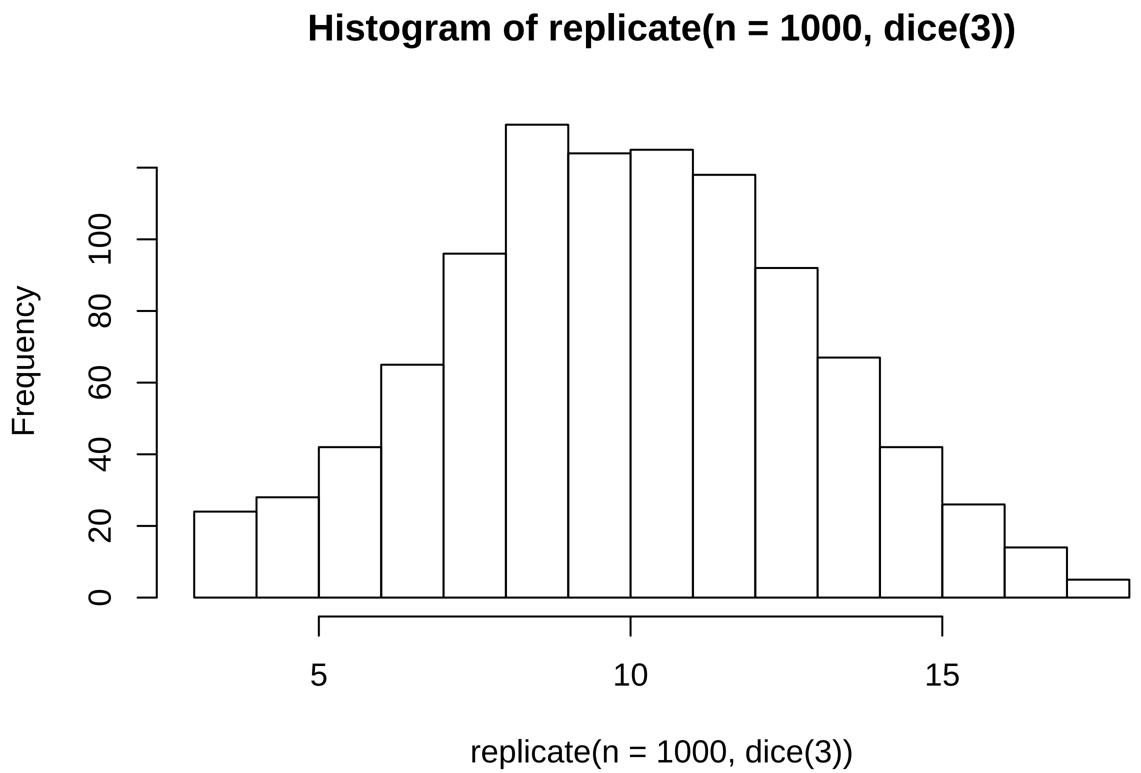
```r
hist(replicate(n = 10, dice(3)))
```



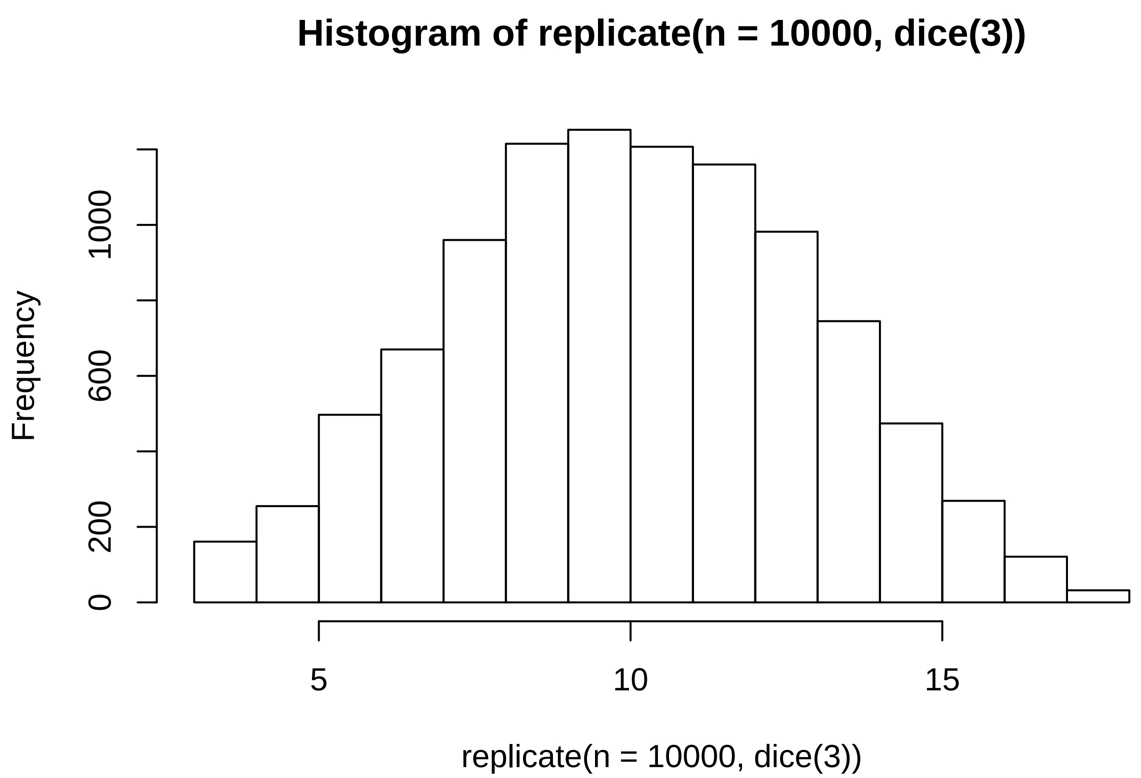

```r
hist(replicate(n = 100, dice(3)))
```



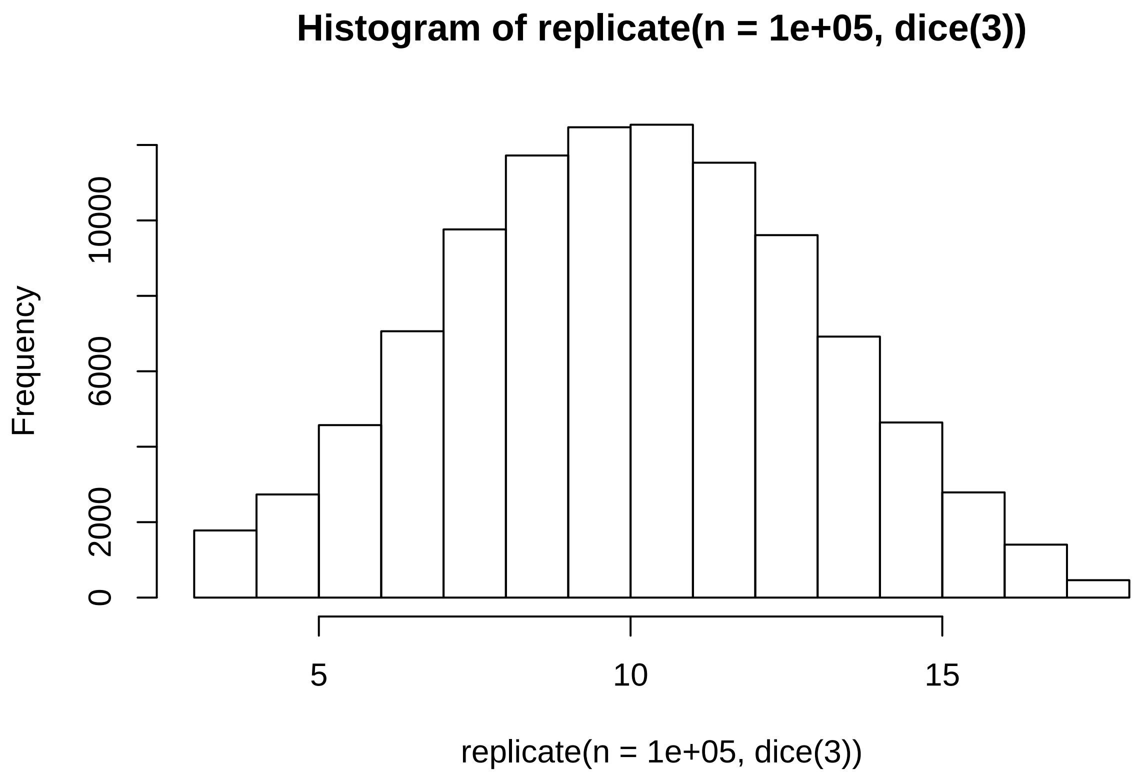
```r
hist(replicate(n = 1000, dice(3)))
```




```r
hist(replicate(n = 10000, dice(3)))
```



```r
hist(replicate(n = 100000, dice(3)))
```



Practice 4

假設台大學生共 n 人，其身高服從常態，且 mean = 160, sd = 10 今隨機從台大學生中抽出 1 人，其平均身高超過 175 公分的機率為何？以 1-pnorm() 來計算理論機率值請試著只以 rnorm() 及 function 來計算 hint: 利用邏輯判斷回傳布林值，再計算 True 的個數所佔的比例 => 將相對頻率看作為機率 ans:

```r
1-pnorm(175, mean = 160, sd = 10)
```

```
## [1] 0.0668072
```

```r
hight = function(n){
  x = rnorm(n, mean = 160, sd = 10)
  freq = length(x[x>175])/length(x)
  return(freq)
}
hight(1000)
```

```
## [1] 0.073
```