统计算法基础第一次作业

姓名：王凯栋　　学号：PB20071441　　日期：2023/3/17

目录

一. 实验目的

- 了解随机数产生的基本方式
- 了解计算机程序解决概率论基本问题的方法
- 了解 R 语言的基本使用

二. 实验过程

- 完成统计计算若干题目 (推导及代码)
- 完成统计计算使用 R 若干代码问题

三. 实验内容

统计计算

2.4 洗好一副编号分别为 $1, 2, \cdots, 54$ 的纸牌，依次抽取出来，若第 i 次抽取到编号 i 的纸牌则称为成功抽取。编写程序估计成功抽取个数 T 的期望和方差，推导理论公式并与模拟结果进行比较。

```r
N <- 1000000  # 模拟次数
success_count_list <- replicate(N, sum(sample(1:54) == 1:54))  # 记录每次模拟中成功抽取的

# 计算期望和方差
mean <- mean(success_count_list)
variance <- var(success_count_list)

cat(" 期望: ", mean, "\n")
```

```
## 期望:  0.99943
```

```r
cat(" 方差: ", variance, "\n")
```

```
## 方差:  0.9999947
```

理论推导如下：记时间 X_i 为

$$X_i = \begin{cases} 1 & Thei-thcardissuccessfullydrawnforthefirsttime \\ 0 & otherwise \end{cases}$$

则 $T = \sum_{i=1}^{54} X_i$, 且 X_i 之间相互独立

因为 $E(X_i) = P(X_i = 1) = \frac{1}{54}, Var(X_i) = E(X_i^2) - (EX_i)^2 = \frac{1}{54}(1 - \frac{1}{54})$

所以则 $ET = \sum_{i=1}^{54} EX_i = \sum_{i=1}^{54} \frac{1}{54} = 1$

$Var(T) = \sum_{i=1}^{54} Var(X_i) = \sum_{i=1}^{54} \frac{1}{54}(1 - \frac{1}{54}) = 1 - \frac{1}{54} = \frac{53}{54}$

- Conclusion: 可见，理论推导的结果与程序运行结果十分相近。

2.9 编写例 2.2.7 中原始的和改进的生成泊松分布随机数的算法并用 R 程序实现，比较两种算法得到的序列是否相同。

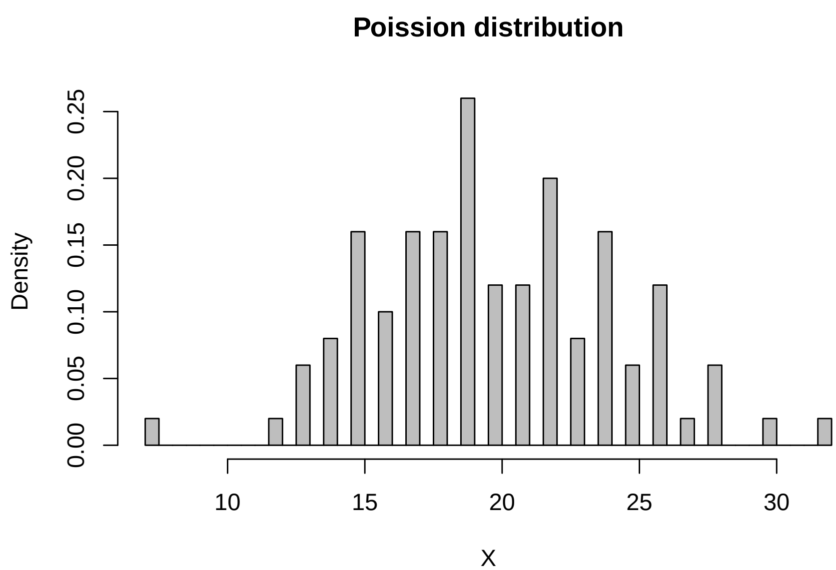
取 $N = 100, \lambda = 20$

- 原始算法

```r
N=100
lambda=20
set.seed(123)
U=runif(N,0,1)
X=NULL
for(i in 1:N){
  X[i]=0
  k=0
  p=exp(-lambda)
  F=p
  while(U[i]>F){
    p=lambda/(k+1)*p
    F=F+p
    k=k+1
  }
  X[i]=k
}
X
```

```
##   [1] 17 24 19 25 27 13 20 26 20 19 28 19 22 21 14 26 17 13 18 28 26 22 21 32 22
##  [26] 22 20 21 17 15 28 26 22 24 12 20 23 16 18 17 15 19 19 18 15 15 17 19 17 25
##  [51] 13 19 24 15 21 16 15 23 26 18 22 14 19 17 24 19 24 24 24 19 23 21 22  7 20
##  [76] 16 18 21 18 15 17 22 19 24 14 19 30 26 25 16 15 22 18 22 18 16 23 14 19 20
```

```r
hist(X, breaks = 50, freq = FALSE, col = 'gray', main = 'Poission distribution')
```



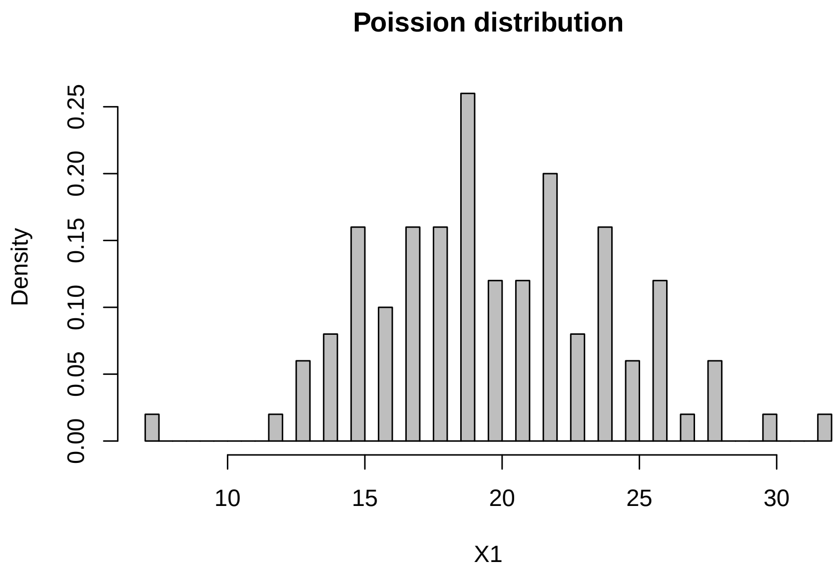
- 改进算法

```
N=100
lambda=20
set.seed(123)
U1=runif(N,0,1)
X1=NULL
for(i in 1:N){
  X1[i]=0
  p=exp(-lambda)
  F=NULL
  F_0=p
  K=floor(lambda)
  for(j in 0:K){
    p=lambda/(j+1)*p
    if(j==0){
      F[j+1]=F_0+p
    }else{
```

```
      F[j+1]=F[j]+p
    }
  }
  if(U1[i]>F[K] & U1[i]<=F[K+1]){
    X1[i]=K+1
  }else if(U1[i]<=F[K]){
    mark=K
    while(U1[i]<=F[mark]){
      mark=mark-1
    }
    X1[i]=mark+1
  }else{
    sig=K+1
    while(U1[i]>F[sig]){
      p=lambda/(sig+1)*p
      F[sig+1]=F[sig]+p
      sig=sig+1
    }
    X1[i]=sig
  }
}
X1
```

```
##   [1] 17 24 19 25 27 13 20 26 20 19 28 19 22 21 14 26 17 13 18 28 26 22 21 32 22
##  [26] 22 20 21 17 15 28 26 22 24 12 20 23 16 18 17 15 19 19 18 15 15 17 19 17 25
##  [51] 13 19 24 15 21 16 15 23 26 18 22 14 19 17 24 19 24 24 24 19 23 21 22  7 20
##  [76] 16 18 21 18 15 17 22 19 24 14 19 30 26 25 16 15 22 18 22 18 16 23 14 19 20
```

```
hist(X1, breaks = 50, freq = FALSE, col = 'gray', main = 'Poission distribution')
```



- Conclusion: 上面便是用两种不同的算法生成 Poisson 分布的方法，从对比来看，二者生成的数完全相同。

2.11 用变换法生成如下分布的随机数：并比较经验分布与理论分布

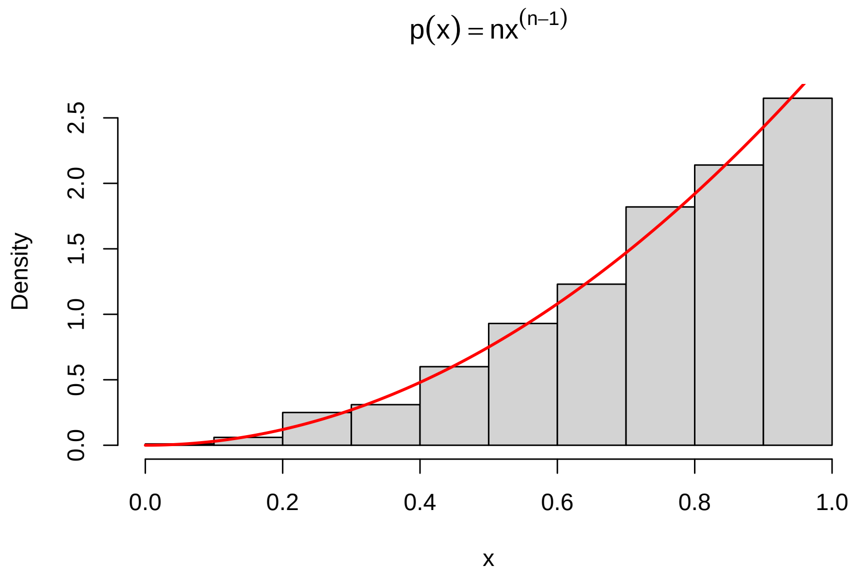
(2)$Beta(n,1)$ 分布，密度为

$$p(x)=nx^{n-1},x\in[0,1]$$

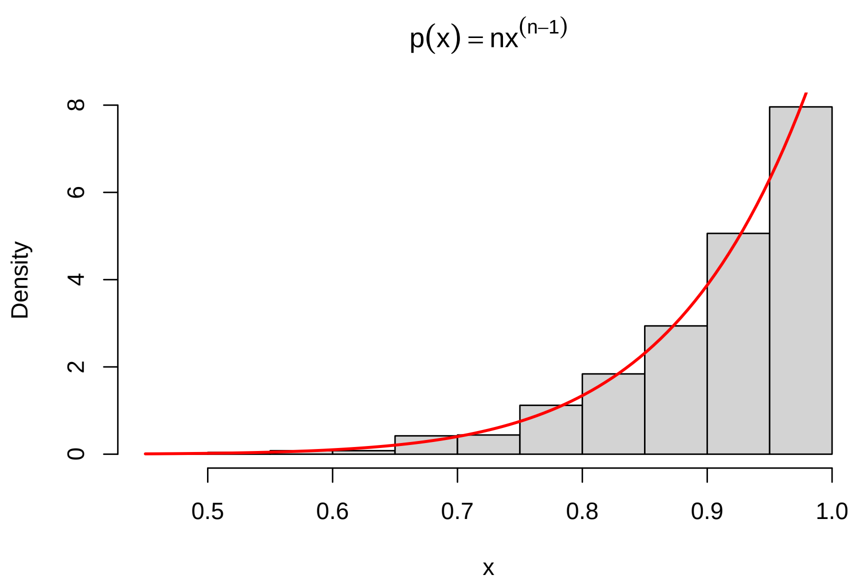
* 取 n=3,10,20

n=3

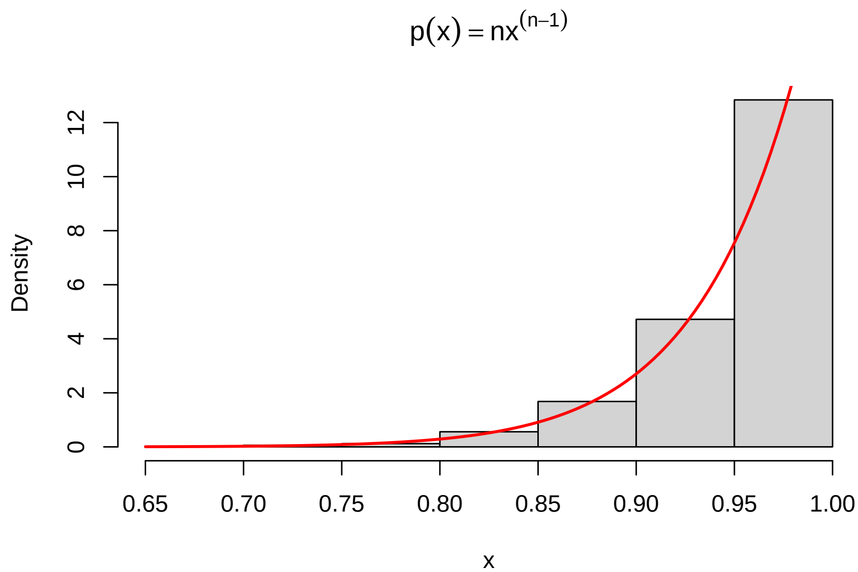
```
beta <- function(n){
set.seed(12)
u <- runif(1000)
x <- u^(1/n)
hist(x,prob=TRUE,main= expression(p(x)==n*x^(n-1)))
curve(n*x^(n-1),col="red",add=TRUE,lwd=2)
}
beta(3)
```


$p(x) = nx^{(n-1)}$
Density
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.2
0.4
0.6
0.8
1.0
x

```
beta(10)
```


$p(x) = nx^{(n-1)}$
Density
0
2
4
6
8
0.5
0.6
0.7
0.8
0.9
1.0
x

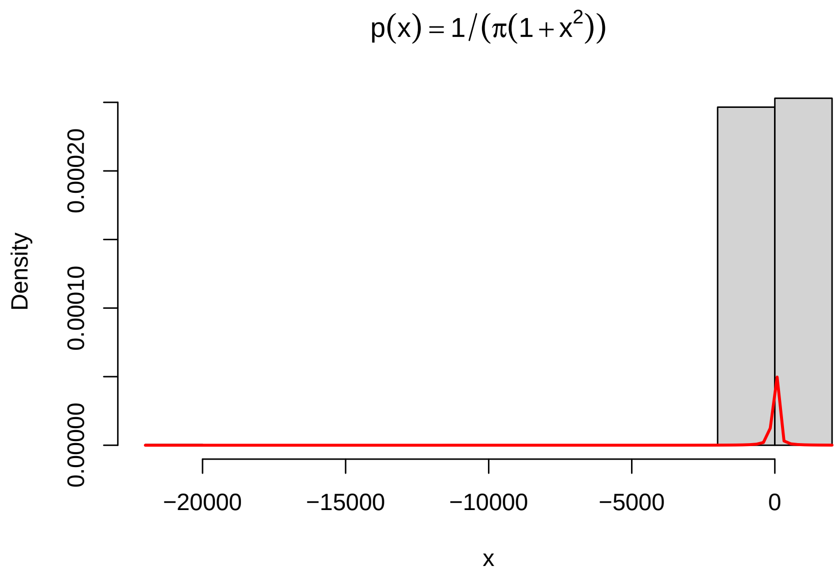
```
beta(20)
```



(4)Cauchy 分布，密度为

$$p(x) = \frac{1}{\pi(1+x^2)}, x \in (-\infty, \infty)$$

```
set.seed(12)
u <- runif(1000)
x <- tan(u*pi)
hist(x,prob=TRUE,main= expression(p(x)==1/(pi*(1+x^2))))
curve(1/(pi*(1+x^2)),col="red",add=TRUE,lwd=2)
```



(6) 威布尔分布，密度为

$$p(x)=\frac{\alpha}{\eta}x^{\alpha-1}e^{-\frac{x^{\alpha}}{\eta}},x>0(\alpha>0,\eta>0)$$

令 $Y=X^{\alpha}$，则有 $X=Y^{\frac{1}{\alpha}}$。则有

$$p_Y(y)=p_X(x)\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right|$$

$$=\frac{\alpha}{\eta}(y^{\frac{1}{\alpha}})^{\alpha-1}e^{-\frac{(y^{\frac{1}{\alpha}})^{\alpha}}{\eta}}\cdot\frac{1}{\alpha}y^{\frac{1-\alpha}{\alpha}}$$

$$=\frac{1}{\eta}e^{-\frac{y}{\eta}}$$

可以看出，Y 的概率密度函数是一个 $exp(\frac{1}{\eta})$ 分布。因此，我们可以使用 Γ 分布的随机数来生成威布尔分布的随机数。具体步骤如下：

(1) 生成一个 $exp(\frac{1}{\eta})$ 分布的随机数 y；

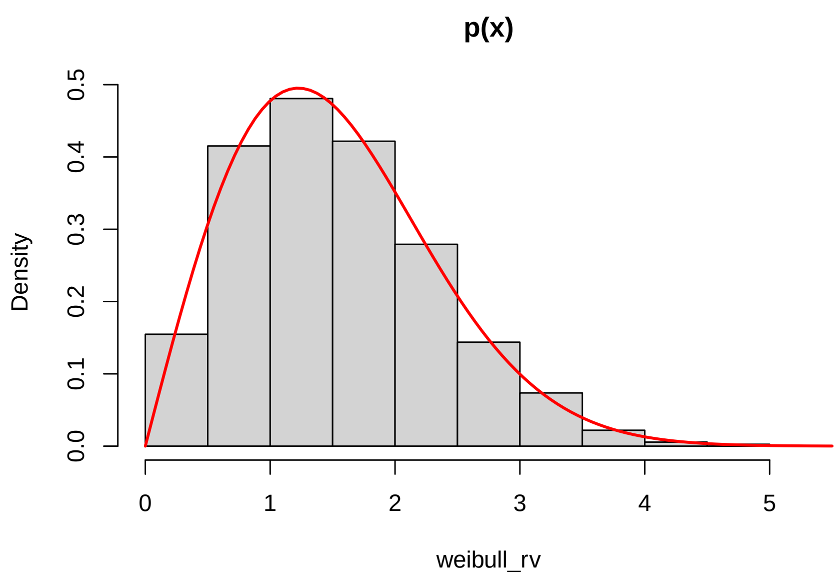
(2) 计算 $x=y^{\frac{1}{\alpha}}$；

(3)x 即为威布尔分布的一个随机数。

我们取 $\alpha=2,\eta=3$ 进行测试

```
set.seed(123)  # 设置随机数种子，方便复现结果
n <- 10000  # 生成随机数的个数
alpha <- 2  # 威布尔分布的参数
eta <- 3  # 威布尔分布的参数

# 生成威布尔分布的随机数
exp_rv <- rexp(n,1/eta)
weibull_rv <- exp_rv^(1/alpha)
hist(weibull_rv,prob=TRUE,main="p(x)")
curve(alpha/eta*x^{alpha-1}*exp(-x^{alpha}/eta),col="red",add=TRUE,lwd=2)
```



13: 设 $\alpha\sim U(0,2\pi),R\sim Exp(\frac{1}{2})$ 与 α 独立，令

$$\begin{cases}X=\sqrt{R}cos\alpha\\Y=\sqrt{R}sin\alpha\end{cases}$$

证明：X,Y 相互独立且都服从 $N(0,1)$ 分布 (并用代码验证)

证明：首先，容易将 X, Y 表示成 R, α 的函数，表示如下

$$\begin{cases} R = X^2 + Y^2 \\ \alpha = arctan(\frac{Y}{X}) \end{cases}$$

所以对应的 $Jocabi$ 变换矩阵为

$$|\frac{\partial(R,\alpha)}{\partial(X,Y)}| = \begin{vmatrix} 2X & 2Y \\ -\frac{Y}{X^2+Y^2} & \frac{X}{X^2+Y^2} \end{vmatrix} = 2$$

因为 α, R 是独立的，所以他们的联合密度函数为

$$f_{R,\alpha}(r,\alpha) = \frac{1}{4\pi}e^{-\frac{r}{2}}, r > 0, 0 < \alpha < 2\pi$$

所以 (X, Y) 的联合密度为

$$f_{x,y}(x,y) = \frac{1}{4\pi}e^{-\frac{x^2+y^2}{2}} \times |2| = \frac{1}{2\pi}e^{-\frac{x^2+y^2}{2}} = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}$$

(X, Y) 联合密度函数因为变量可分离，所以 X, Y 是独立的，且有

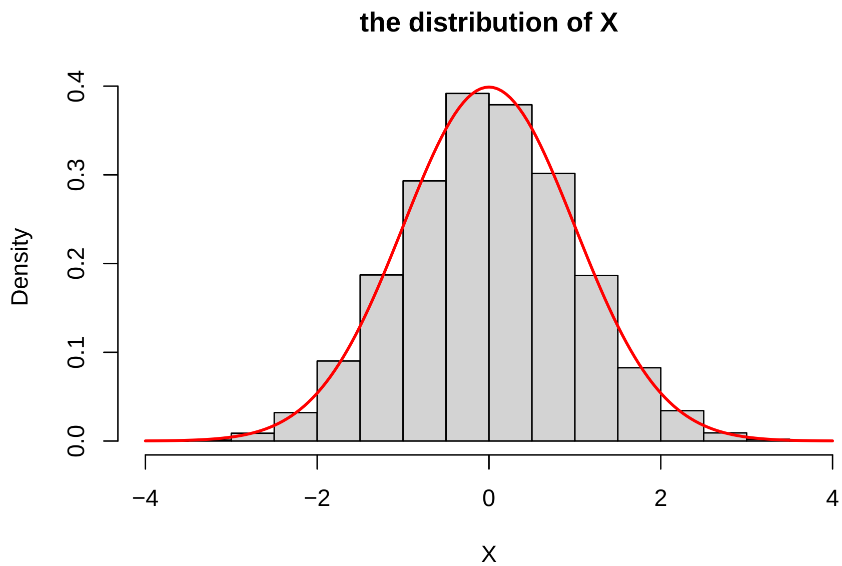
$$f_X(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$$

$$f_Y(y) = \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}$$

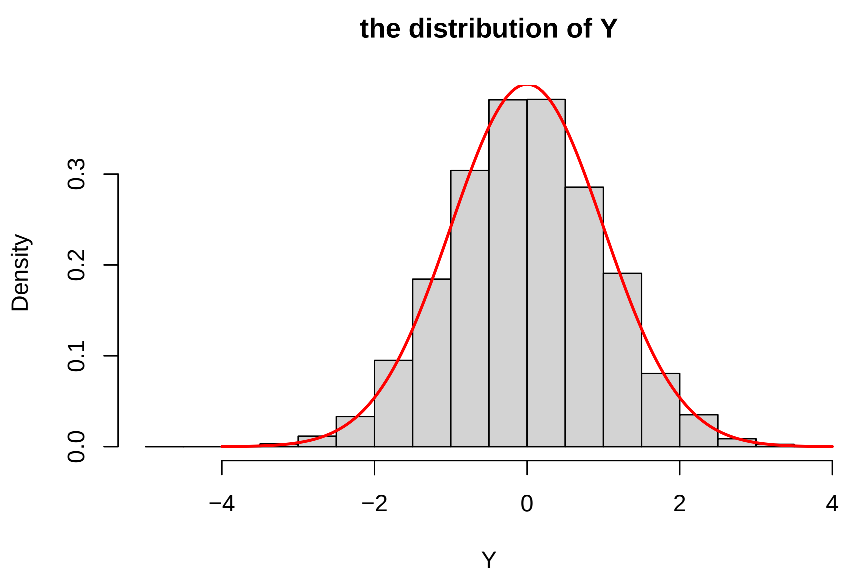
所以它们都服从标准正态分布，证毕

代码验证如下

```
N=10000
set.seed(11)
alpha=runif(N,0,2*pi)
R=rexp(N,1/2)
X=sqrt(R)*cos(alpha)
Y=sqrt(R)*sin(alpha)
x<- seq(-4,4,0.05)
hist(X,prob=TRUE,main= "the distribution of X")
lines(x,dnorm(x,0,1),col="red",lwd=2)
```




```r
hist(Y,prob=TRUE,main= "the distribution of Y")
lines(x,dnorm(x,0,1),col="red",lwd=2)
```



19 设 $X \sim B(n,p)$,k 为满足 $0 \le k \le n$ 的给定的整数，随机变量 Y 的分布函数为 $P(Y \le y) = P(X \le y|X \ge k)$. 记 $\alpha = P(X \ge k)$。分别用逆变换法和舍选法生成 Y 的随机数。当 α 取值较大的时候还是取值小的时候舍取法不可取？

首先我们可以写出 Y 的 cdf 为

$$\begin{aligned}
P(Y \le y) &= P(X \le y|X \ge k) \\
&= \frac{P(X \le y, X \ge k)}{P(X \ge k)} \\
&= \frac{P(X \le y) - P(X < k)}{1 - P(X < k)} \\
&= \frac{F_X(y) - \sum_{i=0}^{k-1} \binom{n}{i} p^i (1-p)^{n-i}}{1 - \sum_{i=0}^{k-1} \binom{n}{i} p^i (1-p)^{n-i}} \\
&= \frac{\sum_{i=0}^{y} \binom{n}{i} p^i (1-p)^{n-i} - \sum_{i=0}^{k-1} \binom{n}{i} p^i (1-p)^{n-i}}{1 - \sum_{i=0}^{k-1} \binom{n}{i} p^i (1-p)^{n-i}} \\
&= \frac{\sum_{i=k}^{y} \binom{n}{i} p^i (1-p)^{n-i}}{\sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}} \\
&= \frac{\sum_{i=k}^{y} \binom{n}{i} p^i (1-p)^{n-i}}{\alpha}
\end{aligned}$$

因此我们得到了 Y 的概率密度函数如下

$$f_Y(y) = \begin{cases} \frac{\binom{n}{y} p^y (1-p)^{n-y}}{\alpha} & k \le y \le n \\ 0 & otherwise \end{cases}$$

- 逆变换法

算法为：(1) 生成 $U \sim Uniform(0,1)$ 的随机数

(2) 计算 $c_0 = 0$ 和 $c_i = \sum_{j=0}^{i-1} f_Y(j)$，即 c_i 表示 Y 在 $[0,i]$ 上的累积概率

(3) 找到最小的 i 使得 $c_{i-1} < U \le c_i$

(4)$Y = i - 1$，即为生成的随机数

假设 p=0.4,n=100,k=20

```r
set.seed(124)
generate_Y <- function(n, p, k, size) {
  alpha <- 1 - sum(dbinom(0:(k-1), n, p))
  c <- cumsum(dbinom(k:n, n, p) / alpha)
  Y <- numeric(size)
  for (i in 1:size) {
    u <- runif(1)
    Y[i] <- sum(c < u) - 1
  }
  return(Y)
}
generate_Y(100,0.4,20,100)
```

```
##   [1] 12 18 19 18 15 16 20 19 26 16 23 24 22 24 18 11 20 22 25 10 21 20 12 18 17
##  [26] 15 24 16 24 18 23 17 18 24 13 28 15 27 14 17 14 24 21 16 18 17 27 16 18 14
##  [51] 18 24 19 15 16 17 15 22 15 20 19 18 11 16 20  3 19 20 17 15 21 25 23 25 20
##  [76] 17 15 27 17 14 18 15 27 23 19 26 21 15 18 15 19 15 14 17 17 16 19 19 24 21
```

- 舍选法

可以使用舍选法（rejection sampling）来生成随机数。具体来说，我们可以使用一个简单的分布作为提议分布（proposal distribution），例如二项分布 $B(m, q)$，其中 m 是一个大于等于 n 的整数，q 是一个介于 0 和 1 之间的概率。我们需要保证二项分布的支撑集包含了目标分布 $P(Y)$ 的支撑集，即 $[0, n]$。

舍选法的步骤如下：

(1) 从提议分布 $B(m, q)$ 中生成一个随机数 Z

(2) 以概率 $\frac{P(Y=Z)}{q \cdot B(Z;m,q)}$ 接受 Z，否则拒绝 Z

(3) 重复步骤 1 和步骤 2，直到生成所需的随机数

其中，$P(Y = Z)$ 可以根据题目中给出的 $P(Y \leq y)$ 求得。如果 $Z > y$，则接受 Z 的概率为 0；如果 $Z \leq y$，则接受 Z 的概率为 $\frac{P(Y=Z)}{q \cdot B(Z;m,q)}$，其中 $q \cdot B(Z; m, q)$ 是提议分布在 Z 处的概率密度函数值。

```r
set.seed(124)
generate_Y_reject <- function(n, p, k, size) {
  m <- n + 100  # 可调整的参数，需要大于等于 n
  q <- 0.5  # 可调整的参数，需要介于 0 和 1 之间
  c <- dbinom(k:n, n, p) / (q * dbinom(k:n, m, q))
  Y <- numeric(size)
  i <- 1
  while (i <= size) {
    Z <- sample(0:m, 1)
    if (Z <= n && runif(1) < c[Z - k + 1]) {
      Y[i] <- Z
      i <- i + 1
    }
  }
  return(Y[1:size])
}
generate_Y_reject(100,0.4,20,100)
```

```
##   [1] 64  4 21 25 26 10 61 16 12 29 40 29 59 12 42 28 64 52 66 10 25 35 64 42 18
##  [26]  7 45 45 57  3 30 44 34 24 42 24 58  9  3 58 36 27 40 58 46 15  8  1 12 67
##  [51] 23 53 41 50  6 32 27 32 53 45 35 34 32 14 44 65 26 56 25  9 33 22 21 42 29
##  [76] 63  8 13 43 50 23 35 27 15 40 55 62 27 22 10 42 56  7 59  3 46  9  2 51 59
```

当 α 较大时，可以考虑使用舍选法（rejection sampling）来生成随机数。这是因为当 α 很大时，条件概率 $P(X \leq y|X \geq k)$ 的分子和分母都比较大，计算复杂度也会比较大，而且在逆变换法中需要使用二项分布的累积分布函数，计算量也比较大。相比之下，舍选法使用一个简单的提议分布，计算量相对较小，且可以通过调整参数来平衡接受率和效率。

23 设随机变量 X 的分布为

$$P(X=k)=\frac{1}{2^{k+1}}+\frac{2^{k+1}}{3^k}, k=1,2,\cdots$$

给出模拟此随机变量的算法

解：

$$P(X=k)=\frac{1}{2^{k+1}}+\frac{2^{k+1}}{3^k}=\frac{3^k+2^{2k+2}}{2^{k+1}3^k}, k=1,2,\cdots$$

为了模拟随机变量 X，我们可以按照如下步骤进行：

首先，容易得到，上面的数求和并不是 1，而是 $\frac{9}{2}$，所以做归一化，给概率密度乘以 $\frac{2}{9}$, 可以得到

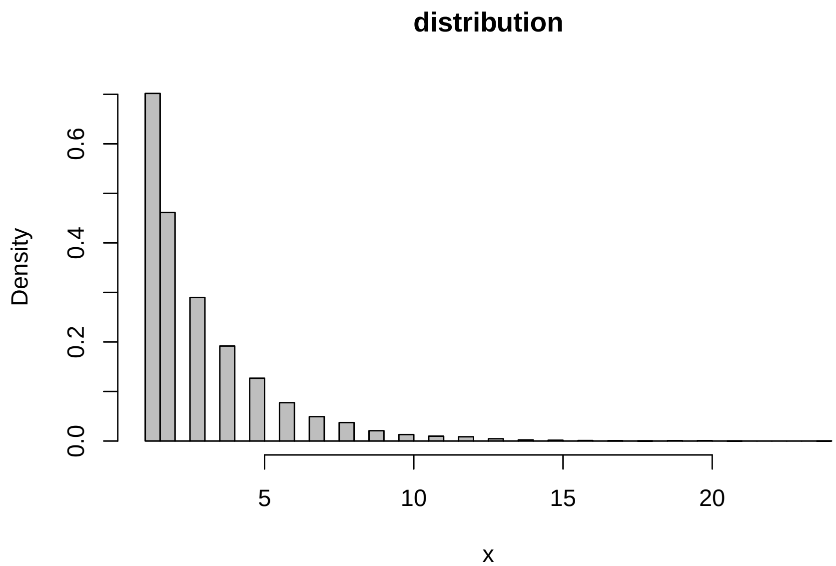
$$P(X=k)=\frac{2}{3^2}\times\frac{3^k+2^{2k+2}}{2^{k+1}3^k}=\frac{3^k+2^{2k+2}}{2^k3^{k+2}}, k=1,2,\cdots$$

(1) 生成一个服从 $U(0,1)$ 均匀分布的随机数 u。

(2) 令 $i=1$。

(3) 当 $u\geqslant 1-\frac{1}{9}(\frac{1}{2})^i-\frac{8}{9}(\frac{2}{3})^i$ 时，令 $i=i+1$，并返回步骤 3；否则，令 $X=i$，输出 X 并停止。

```
set.seed(123)
n=10000
u <- runif(n,0,1)
x=NULL
for(k in 1:n){
i=1
mark=1-1/9*(1/2)^{i}-8/9*(2/3)^{i}
while (mark<=u[k]) {
  i=i+1
  mark=1-1/9*(1/2)^{i}-8/9*(2/3)^{i}
}
x[k]=i
}
hist(x, breaks = 50, freq = FALSE, col = 'gray', main = 'distribution')
```



24 设随机变量 X 的分布函数为

$$p(x) = \frac{1}{2}e^{-|x|}, -\infty < x < \infty$$

称 X 服从双指数分布或 $Laplace$ 分布。分别用逆变换法或复合法生成 X 的随机数

- 逆变换法

该分布的 cdf 为：

$$F(x) = \int_{-\infty}^{x} p(x)dx$$
$$= \begin{cases} \frac{1}{2}e^{x} & x \leq 0 \\ 1 - \frac{1}{2}e^{-x} & x > 0 \end{cases}$$

求出 $F(x)$ 的反函数 $x = F^{-1}(u)$，其中 u 是 $[0,1]$ 均匀分布的随机数。

当 $0 \leq u \leq 1/2$ 时，

$$F(x) = \frac{1}{2}(1 - e^{-x}) = u$$

解得：

$$x = -\ln(1 - 2u)$$

当 $1/2 < u \leq 1$ 时，

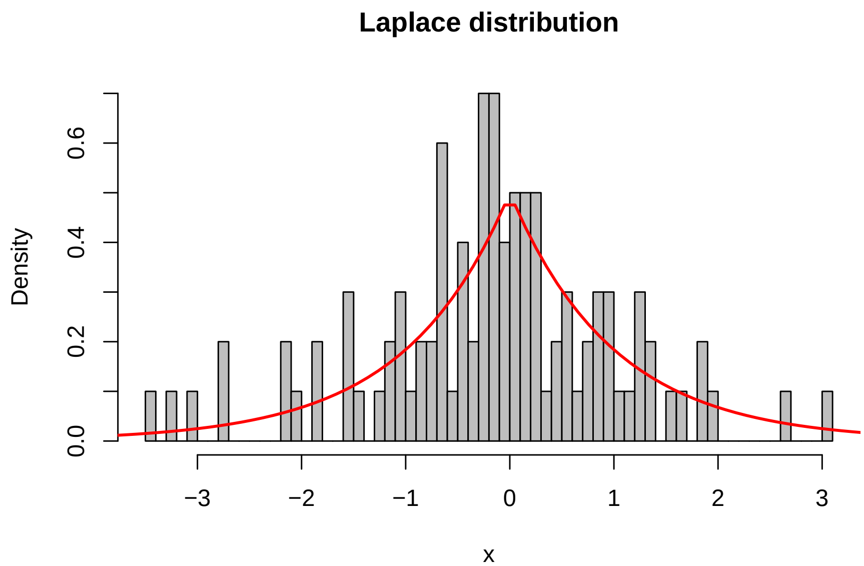
$$F(x) = \frac{1}{2}(e^x + 1) = u$$

解得：

$$x = \ln(2u - 1)$$

```
set.seed(125)
N=100
x=NULL
u <- runif(N,0,1)
for(i in 1:N){
  if(u[i]<=1/2){
    x[i]=-log(1-2*u[i])
  }else{
    x[i]=log(2*u[i]-1)
  }
}
hist(x, breaks = 50, freq = FALSE, col = 'gray',
     main = 'Laplace distribution')

# 绘制理论分布
t <- seq(-5, 5, length.out = 100)
p <- 0.5 * exp(-abs(t))
lines(t, p, col = 'red', lwd = 2)
```



- 复合法

为了生成服从双指数分布的随机数,可以使用复合法(composition method)。复合法的步骤如下:

(1) 选择一个容易生成的分布 $G(x)$,使得存在一个函数 $h(x)$,使得 $h(x)$ 在 $(-\infty, 0)$ 和 $(0, \infty)$ 上分别是单调递增的,且 $h(x)$ 的导数存在且连续。

(2) 求出反函数 $H^{-1}(x)$,其中 $H(x) = \int_{-\infty}^{x} h(t)dt$。

(3) 生成 $U \sim U(0, 1)$,然后令 $X = H^{-1}(U)$。

对于双指数分布,我们可以选择指数分布作为 $G(x)$,即

$$G(x) = \begin{cases} e^{x} & x < 0 \\ e^{-x} & x \geq 0 \end{cases}$$

此时,

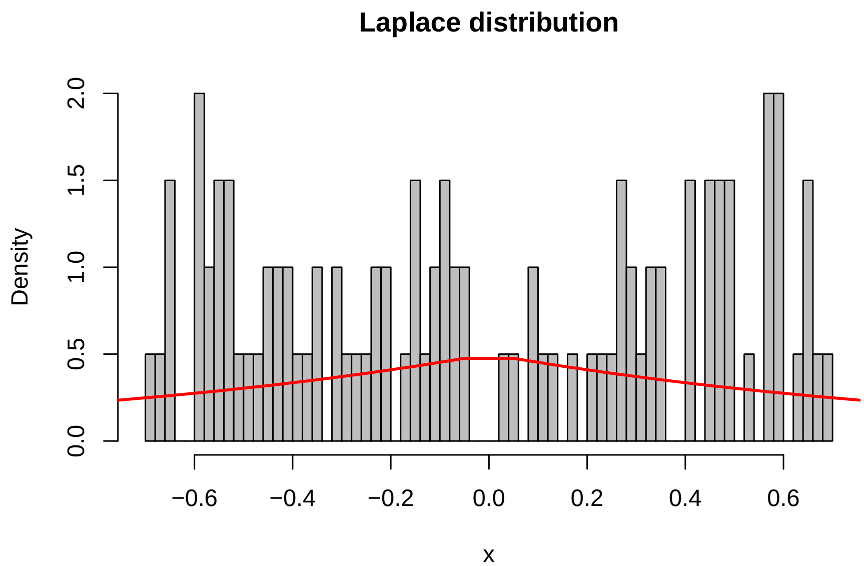
$$h(x) = \frac{1}{2}e^{-|x|}$$

$$H(x) = \begin{cases} \frac{1}{2}e^{-x} & x < 0 \\ 1 - \frac{1}{2}e^{-x} & x \geq 0 \end{cases}$$

$$H^{-1}(x) = \begin{cases} -\ln(2-2x) & 0 < x < \frac{1}{2} \\ \ln(2x) & \frac{1}{2} \le x < 1 \end{cases}$$

```r
set.seed(123)
n=100
u <- runif(n,0,1)
x=NULL
for(i in 1:n){
  if(u[i]<1/2){
    x[i]=-log(2-2*u[i])
  }else{
    x[i]=log(2*u[i])
  }
}
hist(x, breaks = 50, freq = FALSE, col = 'gray', main = 'Laplace distribution')

# 绘制理论分布
t <- seq(-5, 5, length.out = 100)
p <- 0.5 * exp(-abs(t))
lines(t, p, col = 'red', lwd = 2)
```



统计计算使用 R

3.11 生成一个大小为 1000，服从正态位置混合变量的随机样本，混合变量的分量分别服从 $N(0,1)$ 分布和 $N(3,1)$ 分布，混合密度为 p_1 和 $p_2=1-p_1$. 对 $p_1=0.5$ 画出叠加了密度曲线的直方图，对不同的 p_1 值进行重复，并观察混合变量的经验分布是否是双峰的. 推测能够生成使混合变量为双峰的 p_1 的值.

生成服从正态位置混合分布的随机样本，可以使用以下步骤：

(1) 生成服从二项分布的随机变量 Z，其中 $n=1$，$p=p_1$。

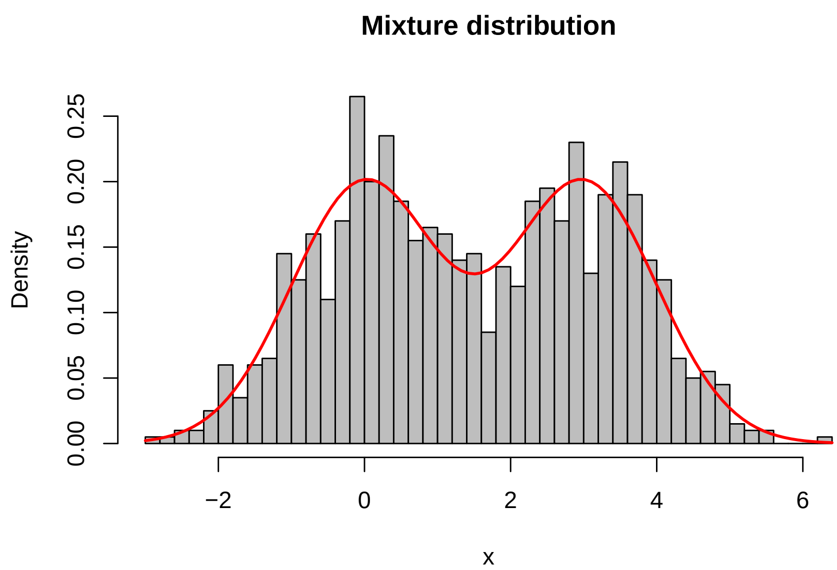
(2) 根据 Z 的值，以概率 p_1 和 $1-p_1$ 选择正态分布 $N(0,1)$ 和 $N(3,1)$ 中的一个，生成随机数。

(3) 重复步骤 1 和步骤 2，直到得到所需的样本大小为止。

```
set.seed(123)
n <- 1000
p1 <- 0.5
```

```
z <- rbinom(n, size = 1, prob = p1)
x <- rnorm(n, mean = ifelse(z == 1, 3, 0), sd = 1)

hist(x, breaks = 50, freq = FALSE, col = 'gray', main = 'Mixture distribution')
curve(p1 * dnorm(x, mean = 0, sd = 1) + (1 - p1) * dnorm(x, mean = 3, sd = 1), add = TR
```

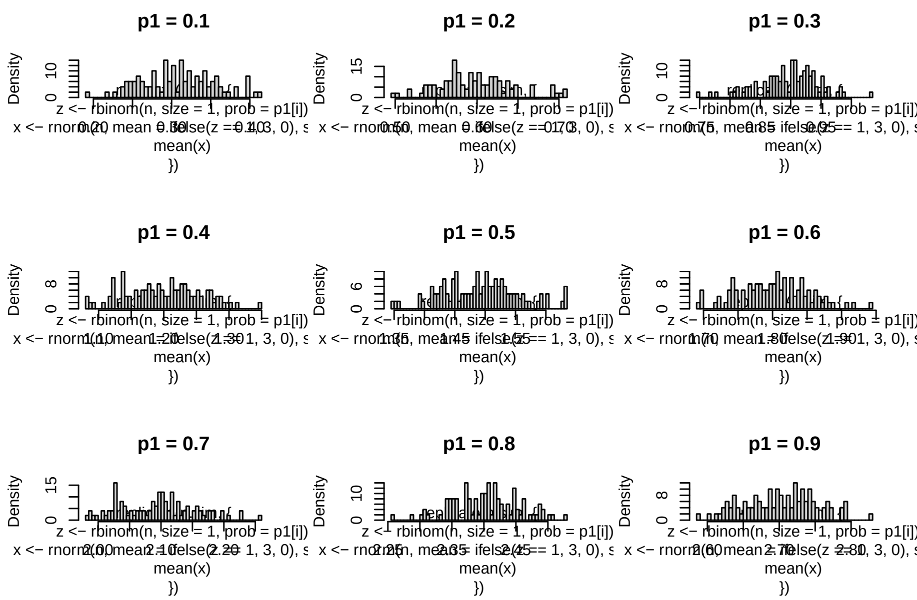


可见，对 $p_1 = 0.5$, 其经验分布为双峰的

尝试用不同的 p_1 值重复生成混合分布，观察其经验分布是否为双峰的

```
p1 <- seq(0.1, 0.9, by = 0.1)
n.sim <- 100
n <- 1000
par(mfrow = c(3, 3))
for (i in seq_along(p1)) {
  hist(replicate(n.sim, {
    z <- rbinom(n, size = 1, prob = p1[i])
    x <- rnorm(n, mean = ifelse(z == 1, 3, 0), sd = 1)
    mean(x)
```

```
  }), breaks = 50, freq = FALSE, main = paste0('p1 = ', p1[i]))
}
```



从上面的 9 张经验分布图可以看出

(1) 当 p 接近 0.5 时，经验分布呈现双峰

(2) 当 p 接近 0 或者 1 时，经验分布接近单峰

3.14 使用 *Choleski* 分解法生成 200 个服从三维多元正态分布的随机观测值，该分布具有均值向量 $(0,1,2)$ 和协方差矩阵

$$\begin{pmatrix} 1.0 & -0.5 & 0.5 \\ -0.5 & 1.0 & -0.5 \\ 0.5 & -0.5 & 1.0 \end{pmatrix}$$

使用 R 中的"pairs" 图对每一对变量画出一列散点图。对每一对变量（视觉上）验证其位置和相关系数与相应的双正态分布的理论参数大致吻合。

解：使用 *Choleski* 分解法生成三维多元正态分布的随机观测值，可以按照以下步骤进行：

(1) 定义均值向量和协方差矩阵。

(2) 对协方差矩阵进行 *Choleski* 分解，得到一个下三角矩阵 L，使得 $\Sigma = LL^T$。

(3) 生成一个三维的标准正态分布随机向量 $Z = (Z_1, Z_2, Z_3)$。

(4) 计算 $X = LZ + \mu$，其中 μ 为均值向量。

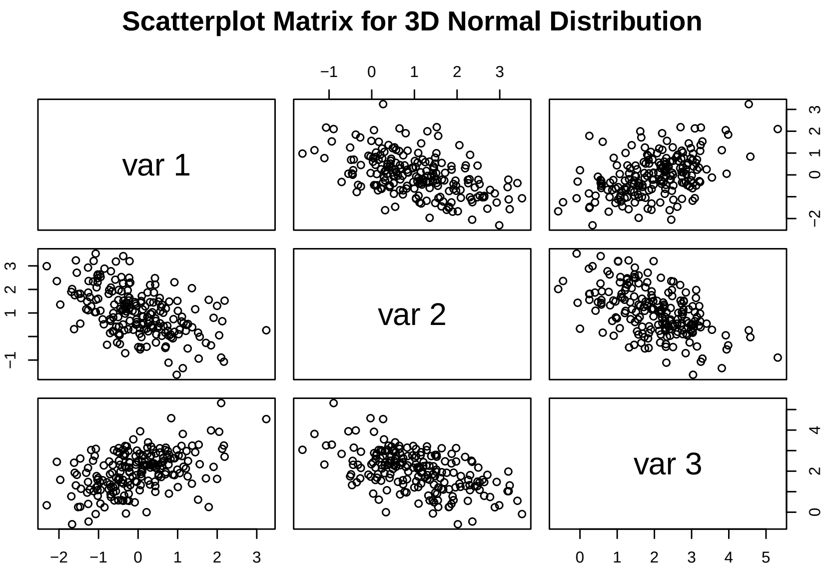
(5) 重复上述步骤多次，得到多个服从三维多元正态分布的随机观测值。

```
# 定义均值向量和协方差矩阵
mu <- c(0, 1, 2)
sigma <- matrix(c(1.0, -0.5, 0.5,
                  -0.5, 1.0, -0.5,
                  0.5, -0.5, 1.0), nrow=3)

# 对协方差矩阵进行 Choleski 分解
L <- chol(sigma)

# 生成 200 个服从三维多元正态分布的随机观测值
set.seed(123)
n <- 200
Z <- matrix(rnorm(n*3), ncol=3)
X <- t(t(Z %*% L) + mu)

# 对每一对变量画出一列散点图
pairs(X, main="Scatterplot Matrix for 3D Normal Distribution")
```



从散点图矩阵可以看出，每一对变量的散点图都大致呈现双正态分布的形状，其中一个峰值位于均值附近，另一个峰值位于距离均值较远的位置。此外，每一对变量的相关系数也与协方差矩阵中对应的元素大小大致相符。因此，可以认为生成的随机观测值服从三维多元正态分布，并且每一对变量都符合双正态分布的理论参数。