

STAT3003: 统计算法基础

随机变量的产生 (1)

曾靖

中国科学技术大学管理学院



大纲

1. 随机数
2. 逆变换法
3. 接受拒绝算法

本节内容来自于《统计计算使用 R 》第 3 章,《统计计算》第 2 章

1 随机数

1 随机数

- 在随机模拟中，如何生成随机数是一个很重要的问题。目前普通的 CPU 和服务端实际上并不能产生真正的随机数，而是产生伪随机数。对于某一个指定的种子，计算机按照某种固定的递推算法得到了一个序列。所以，从同一种子出发，得到的随机数序列是相同的。虽然这种伪随机数并不是随机的，但是伪随机数序列的表现与真实的分布 $F(x)$ 产生的随机数列很难区分。
- 真正的随机数 (True Random Number, TRN) 需要依赖物理世界中的不可预测过程，例如：
 - 热噪声（电子器件里的热噪声电压）
 - 量子随机数（基于光子到达时间、量子测量塌缩等量子不可预知性）
 - 放射性衰变

1 随机数

- 可重复性 (reproducibility): 为了得到可重现的结果, 随机模拟应该从固定不变的种子开始模拟
- 随机数生成思想:
 1. 生成满足均匀分布的随机数;
 2. 通过某种变换得到满足其他分布的随机数。

1.1 随机种子

`set.seed(seed, kind = NULL)`: 固定随机序列的种子

- `seed`: 种子编号。从编号 `seed` 的种子运行相同的模拟程序就可以得到相同的结果;
- `kind`: 随机数生成器 (random number generator) 类型。

例子 1.1

```
round(runif(5), 4)
```

```
## [1] 0.9608 0.1305 0.9962 0.8367 0.6505
```

```
round(runif(5), 4)
```

```
## [1] 0.4960 0.6315 0.5352 0.3739 0.7973
```

```
set.seed(5)
```

```
runif(5)
```

```
## [1] 0.2002145 0.6852186 0.9168758 0.2843995 0.1046501
```

```
set.seed(5)
```

```
runif(5)
```

```
## [1] 0.2002145 0.6852186 0.9168758 0.2843995 0.1046501
```

1.2 随机抽取

`sample(x, size, replace = FALSE, prob = NULL)`: 从一个有限集合中无放回或有放回地随机抽取

- `x` 用以存储有限集合的向量，也可以为一个正整数（此时集合为 `1:x`）；
- `size` 指定抽样个数，即样本数；
- `replace` 指定是否为有放回抽样，`TRUE` 是有放回抽样，`FALSE` 是无放回抽样；
- `prob` 指定以各种权重抽取，默认是等概率。

例子 1.2

- 有放回抽样 (Bernoulli distribution)

```
sample(0:1, size = 10, replace = TRUE)
```

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

- 无放回等概率抽样:

```
set.seed(5)
```

```
sample(1:100, size = 6, replace = FALSE)
```

```
## [1] 66 57 79 75 41 85
```

思考

1. 如何利用 `sample` 函数构造多项分布? `replace = TRUE` 还是 `replace = FALSE`? 例如, 考虑多项分布为掷骰子 10 次后各面出现的频数。
2. 能否利用 `sample` 函数构造一个数列的置换?

1.3 如何生成简单的随机变量

- `runif(n)` 产生 n 个标准均匀分布随机数;

```
x <- runif(5)
```

```
sprintf("%.2f", x)
```

```
## [1] "0.96" "0.11" "0.27" "0.49" "0.32"
```

- `rnorm(n)` 产生 n 个标准正态分布随机数。

```
x <- rnorm(5)
```

```
sprintf("%.2f", x)
```

```
## [1] "0.15" "-0.83" "1.22" "1.00" "0.58"
```

1.4 其他相关概念

- p: probability, 分布函数
- q: quantile, 分位数
- d: density, 概率密度函数
- r: random, 随机数

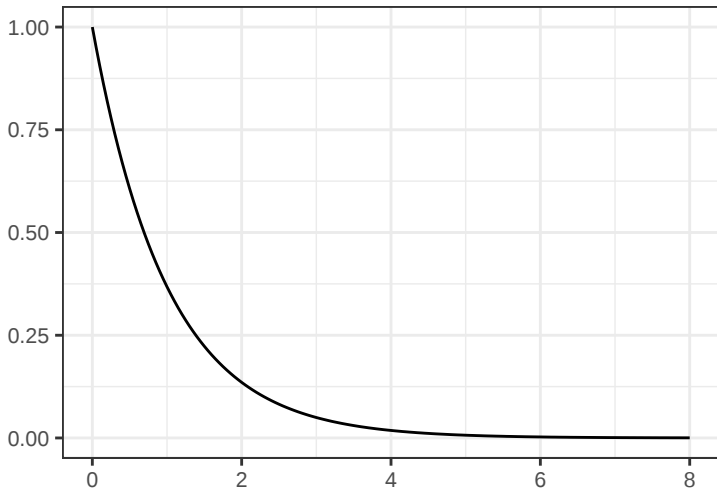
常见分布

Distribution	CDF	Parameters
正态分布	<code>pnorm</code>	<code>mean, sd</code>
t 分布	<code>pt</code>	<code>df</code>
均匀分布	<code>punif</code>	<code>min, max</code>
二项分布	<code>pbinom</code>	<code>size, prob</code>
卡方分布	<code>pchisq</code>	<code>df</code>
指数分布	<code>pexp</code>	<code>rate</code>
泊松分布	<code>ppois</code>	<code>lambda</code>
F 分布	<code>pf</code>	<code>df1, df2</code>
伽玛分布	<code>pgamma</code>	<code>shape, rate or shape</code>
几何分布	<code>pgeom</code>	<code>prob</code>
对数正态分布	<code>plnorm</code>	<code>meanlog, sdlog</code>
负二项分布	<code>pnbinom</code>	<code>size, prob</code>

例子 1.5: 指数分布的概率密度函数

```
x <- seq(0, 8, .05)
data <- data.frame(x = x, y = dexp(x))
ggplot(data, aes(x = x, y = y)) + geom_line() +
  labs(x = "", y = "",
        title = "Exponential distribution") +
  theme_bw() +
  theme(plot.title = element_text(size = 8, hjust = 0.5),
        axis.text = element_text(size = 8),
        axis.title = element_text(size = 8))
```

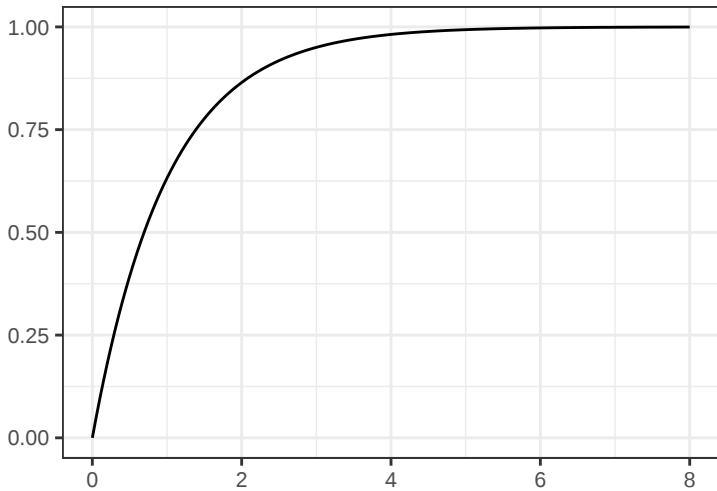
Exponential distribution



例子 1.6: 指数分布的分布函数

```
x <- seq(0, 8, .05)
data <- data.frame(x = x, y = pexp(x))
ggplot(data, aes(x = x, y = y)) + geom_line() +
  labs(x = "", y = "",
        title = "Exponential distribution") +
  theme_bw() +
  theme(plot.title = element_text(size = 8, hjust = 0.5),
        axis.text = element_text(size = 8),
        axis.title = element_text(size = 8))
```

Exponential distribution



例子 1.7: 指数分布的随机数

```
x <- rexp(1000, 0.5)
data <- data.frame(x = x)
```

思考: 如何做图来近似验证随机数的分布符合指数分布?

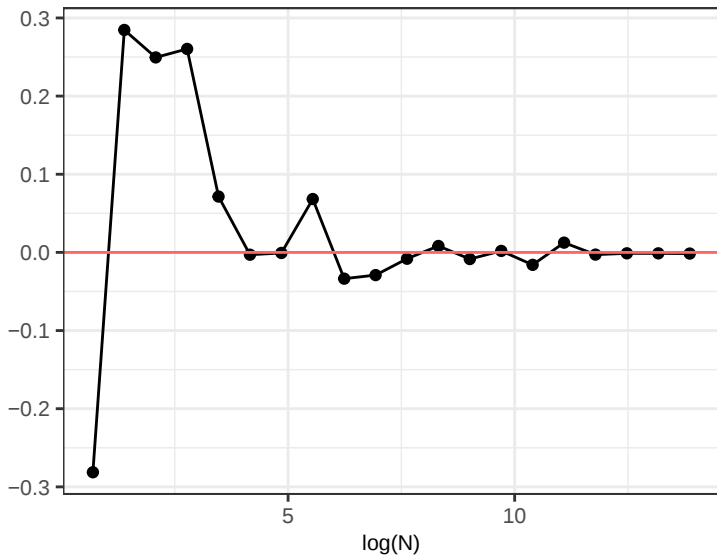
例子 1.8: 大数定律

- Normal distribution

```
N <- 2^c(1:20)
y <- sapply(N, function(n){
  x <- rnorm(n, 0, 1)
  mean(x)
})
```

思考: 如何用 *for* 循环来重写?

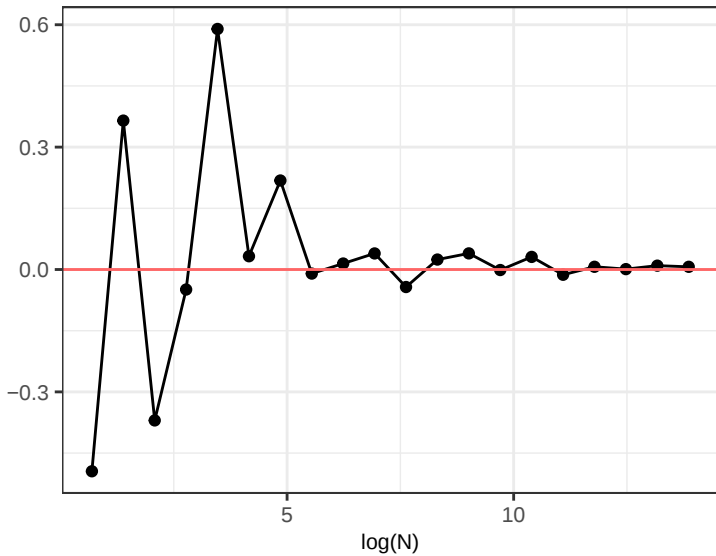
```
data <- data.frame(N = N, mean = y)
ggplot(data, aes(x = log(N), y = mean))+
  geom_point()+
  geom_line()+
  geom_abline(intercept = 0, slope = 0,
              color = '#FF6666')+
  labs(y="")+
  theme_bw()+
  theme(plot.title = element_text(size = 8, hjust = 0.5),
        axis.text = element_text(size = 8),
        axis.title = element_text(size = 8))
```



- Student's t -distribution with degree of freedom = 2.

```
N <- 2^c(1:20)
y <- sapply(N, function(n){
  x <- rt(n, df = 2)
  mean(x)
})
data <- data.frame(N = N, mean = y)
```

```
ggplot(data, aes(x = log(N), y = mean))+  
  geom_point()+  
  geom_line()+  
  geom_abline(intercept = 0, slope = 0,  
              color = '#FF6666')+  
  labs(y="")+  
  theme_bw()+  
  theme(plot.title = element_text(size = 8, hjust = 0.5),  
        axis.text = element_text(size = 8),  
        axis.title = element_text(size = 8))
```

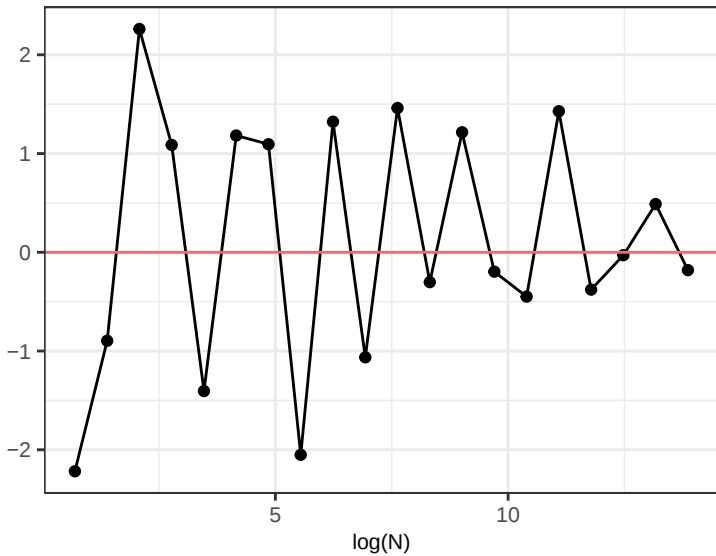


- Student's t -distribution with degree of freedom = 1.

思考: t 分布自由度为 1 的时候满足服从大数定律吗?

```
N <- 2^c(1:20)
y <- sapply(N, function(n){
  x <- rt(n, df = 1)
  mean(x)
})
data <- data.frame(N = N, mean = y)
```

```
ggplot(data, aes(x = log(N), y = mean))+  
  geom_point()+  
  geom_line()+  
  geom_abline(intercept = 0, slope = 0,  
              color = '#FF6666')+  
  labs(y="")+  
  theme_bw()+  
  theme(plot.title = element_text(size = 8, hjust = 0.5),  
        axis.text = element_text(size = 8),  
        axis.title = element_text(size = 8))
```



例子 1.9: 中心极限定理

思考: 如何验证均匀分布的中心极限定理?

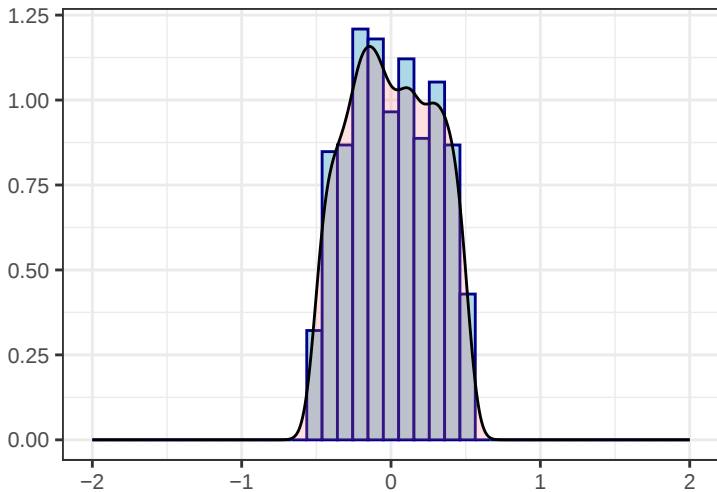
```
iter <- 1000
N <- c(1, 10, 100)

data <- lapply(N, function(n){
  x <- replicate(iter, mean(runif(n)))
  stat <- (x-0.5)*sqrt(n)
})
```

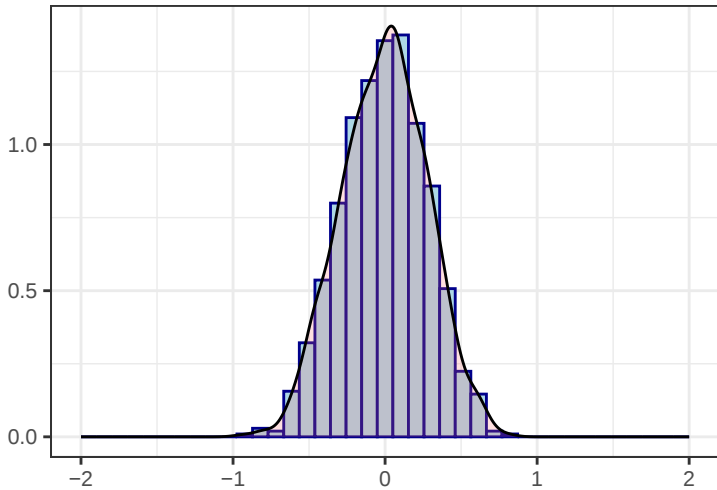
An example with $n = 1$.

```
data_x <- data.frame(x = data[[1]])
ggplot(data_x, aes(x = x))+
  geom_histogram(aes(y=after_stat(density)),
                 color="darkblue", fill="lightblue",
                 bins = 40)+
  scale_x_continuous(limits = c(-2, 2))+
  geom_density(alpha=.2, fill="#FF6666")+
  labs(x = "", y = "", title = "n = 1")+
  theme_bw()+
  theme(plot.title = element_text(size = 8, hjust = 0.5),
        axis.text = element_text(size = 8),
        axis.title = element_text(size = 8))
```

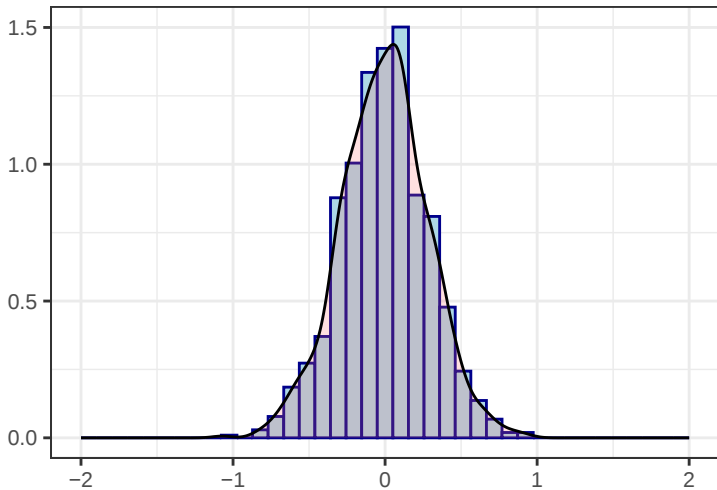
$n = 1$



$n = 10$



$n = 100$



- Student's t -distribution with degree of freedom = 3.

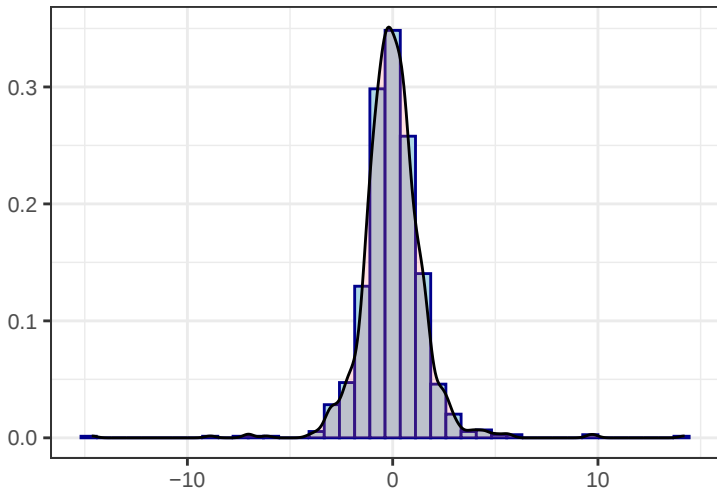
```
iter <- 1000
N <- c(1, 10, 100)

data2 <- lapply(N, function(n){
  x <- replicate(iter, mean(rt(n, 3)))
  stat <- x*sqrt(n)
})
```

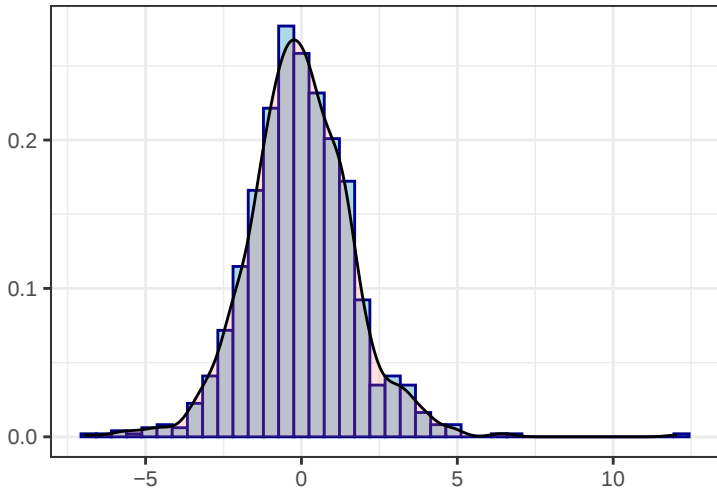
An example with $n = 1$.

```
data_x <- data.frame(x = data2[[1]])
ggplot(data_x, aes(x = x))+
  geom_histogram(aes(y=after_stat(density)),
                 color="darkblue", fill="lightblue",
                 bins = 40)+
  geom_density(alpha=.2, fill="#FF6666")+
  labs(x = "", y = "", title = "n = 1")+
  theme_bw()+
  theme(plot.title = element_text(size = 8, hjust = 0.5),
        axis.text = element_text(size = 8),
        axis.title = element_text(size = 8))
```

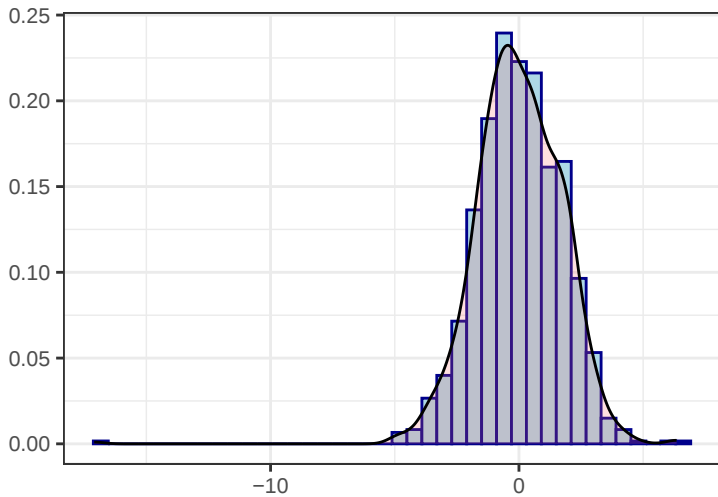
$n = 1$



$n = 10$



$n = 100$



2 逆变换法

2.1 连续型随机变量的逆变换法

Theorem 1 (概率积分变换). X 是一个连续的随机变量, 具有累积分布函数 $F_X(x)$, 则我们有 $U = F_X(X) \sim U(0, 1)$ 。

Proof: For $x \in [0, 1]$, we have

$$\begin{aligned}P(F(X) \leq x) &= P(X \leq F^{-1}(x)) \\&= (F \circ F^{-1})(x) = x\end{aligned}$$

Hence, the random variables $F(X) \sim \text{Unif}[0, 1]$.

□

同理, 我们可以证明 $F_X^{-1}(U) \sim X$ 。据此, 我们可以通过 F_X 的逆变换和均匀分布来构造随机变量 X 。

Algorithm 1

1. 求出随机变量 X 的累积分布函数的逆 F_X^{-1} ;
2. 生成一个标准均匀分布的随机变量 U ;
3. 对其做逆变换 $F_X^{-1}(U)$ 即可得到 $F_X^{-1}(U) \sim X$ 。

例子 2.1

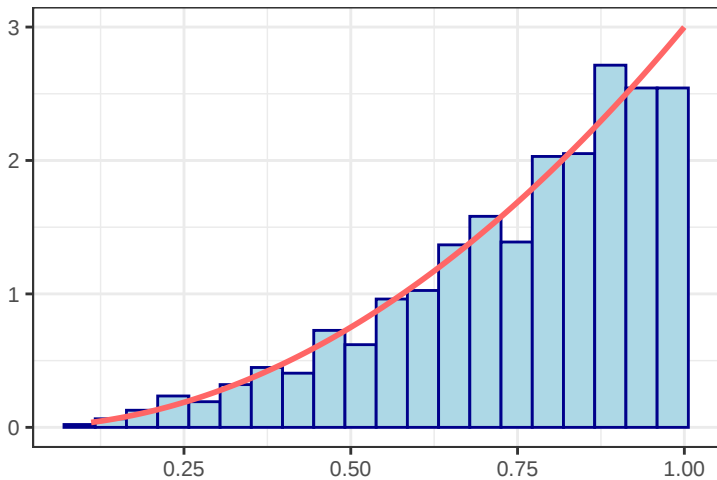
思考: 模拟一个密度函数为 $f_X(x) = 3x^2$ ($0 < x < 1$) 的随机样本。

$$F_X(x) = x^3, \quad 0 < x < 1,$$

$$F_X^{-1}(u) = u^{1/3}$$

```
u <- runif(1000)
x <- u^(1/3)
data_x <- data.frame(x = x)
ggplot(data_x, aes(x = x)) +
  geom_histogram(aes(y = after_stat(density)), bins = 20,
    color = "darkblue", fill = "lightblue") +
  stat_function(fun = function(x) 3 * x^2,
    color = "#FF6666", linewidth = 1) +
  labs(x = "", y = "",
    title = expression(f(x) == 3 * x^2)) +
  theme_bw() +
  theme(plot.title = element_text(size = 8, hjust = 0.5),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 8))
```

$$f(x) = 3x^2$$



2.2 离散变量的逆变换法

- 设 X 是一个离散型的随机变量, 取值满足 $x_1 < \cdots < x_n$, 且

$$P(X = x_i) = p_i, \quad i = 1, \dots, n.$$

那么定义累计分布函数的逆

$$\begin{aligned} F_X^{-1}(u) &= \{x_i : F_X(x_{i-1}) < u \leq F_X(x_i)\} \\ &= \min\{x_i : F_X(x_i) \geq u\}, \quad 0 < u < 1. \end{aligned}$$

思考 1: 写出 X 对应的累积分布函数的逆。

思考 2: 如何验证 $F_X^{-1}(U)$ 服从 X 的分布?

例子 2.2: 几何分布

Question: 几何分布的概率质量函数和分布函数分别为

$$f(x) = q^x p \quad F_X(x) = 1 - q^{x+1}, x = 0, 1, 2, \dots,$$

其中 $q = 1 - p$ 。求该几何分布的 $F_X^{-1}(u)$ 。

例子 2.2: 几何分布

Question: 几何分布的概率质量函数和分布函数分别为

$$f(x) = q^x p \quad F_X(x) = 1 - q^{x+1}, x = 0, 1, 2, \dots,$$

其中 $q = 1 - p$ 。求该几何分布的 $F_X^{-1}(u)$ 。

Solution: 可以求得其逆变换为

$$F_X^{-1}(u) = \min\{x : 1 - q^{x+1} \geq u\} = \left\lceil \frac{\log(1 - u)}{\log(1 - p)} \right\rceil - 1.$$

其中 $\lceil t \rceil$ 表示上取整，即取不小于 t 的最小整数。

模拟代码

```
n <- 1000
p <- 0.25
u <- runif(n)
x <- ceiling(log(1-u)/log(1-p)) - 1
tab <- table(x)
```

```
options(width = 40)
## 比较相对频数和理论概率
round(rbind(tab/n, dgeom(as.numeric(names(tab)),
                          prob = p)), 3)
```

```
##           0           1           2           3           4
## [1,] 0.246 0.199 0.163 0.100 0.080
## [2,] 0.250 0.188 0.141 0.105 0.079
##           5           6           7           8           9
## [1,] 0.063 0.037 0.032 0.017 0.013
## [2,] 0.059 0.044 0.033 0.025 0.019
##          10          11          12          13          14
## [1,] 0.010 0.013 0.007 0.007 0.002
## [2,] 0.014 0.011 0.008 0.006 0.004
##          15          17          18          21
## [1,] 0.004 0.005 0.001 0.001
## [2,] 0.003 0.002 0.001 0.001
```

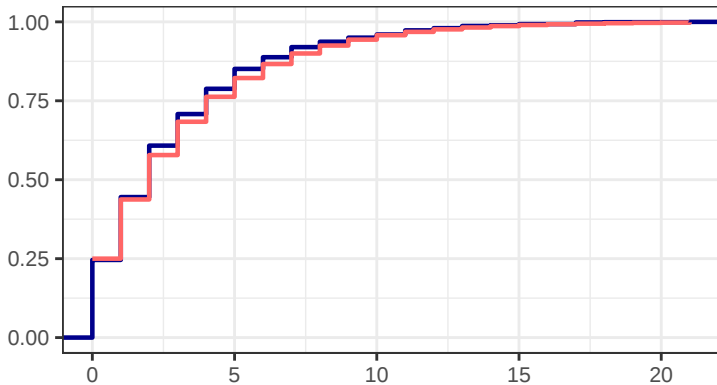
```

geom22 <- data.frame(x = 0:max(x),
                     F = pgeom(0:max(x), prob = p))
p22 <- ggplot(data.frame(x = x), aes(x = x)) +
  stat_ecdf(aes(color = "Empirical CDF"),
            geom = "step", linewidth = 0.8) +
  geom_step(data = geom22,
            aes(x = x, y = F, color = "Theoretical CDF"),
            linewidth = 0.8) +
  scale_color_manual(values = c("Empirical CDF" = "darkblue",
                                "Theoretical CDF" = "#FF6666"))
labs(x = "", y = "", title = "Geometric distribution",
      color = "") +
theme_bw() +
theme(
  plot.title = element_text(size = 8, hjust = 0.5),
  axis.text = element_text(size = 8),
  axis.title = element_text(size = 8),
  legend.position = "top"
)

```

Geometric distribution

— Empirical CDF — Theoretical CDF



例子 2.3：泊松分布

Question: 对于 Poisson 分布,

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, \dots,$$

如何计算该泊松分布的 $F_X^{-1}(u)$?

例子 2.3: 泊松分布

Question: 对于 Poisson 分布,

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, \dots,$$

如何计算该泊松分布的 $F_X^{-1}(u)$?

Solution: 可以利用递推关系计算 $F(k)$, $k = 0, 1, 2, \dots$ 。

- 令 $p_k = P(X = k)$ 。则 $p_0 = e^{-\lambda}$, $p_{k+1} = \frac{\lambda}{k+1} p_k$ 。
- $F(k+1) = F(k) + p_{k+1} = F(k) + \frac{\lambda}{k+1} p_k$
- 对给定的 u , 返回第一个满足 $F(k) \geq u$ 的 k , 即 $F_X^{-1}(u) = \min\{k : F(k) \geq u\}$ 。

模拟代码

```
n <- 500; lambda <- 1
x <- rep(NA_integer_, n)
for (i in 1:n){
  # 初始化  $k=0$ ,  $p$  为  $p_0$ ,  $f$  为  $F(0)$ 
  k <- 0; p <- exp(-lambda); f <- p
  # 生成  $U \sim Unif(0, 1)$ 
  u <- runif(1)
  repeat{
    # 若  $U \leq F(k)$ , 当前  $k$  即为逆变换结果
    if(u <= f) break
    # 否则利用递推式更新  $p_{(k+1)}$  和  $F(k+1)$ 
    p <- lambda/(k+1)*p; f <- f + p;
    # 进入下一个候选取值
    k <- k + 1
  }
  x[i] <- k
}
```

```
tab <- table(x)
round(rbind(tab/n, dpois(as.numeric(names(tab)),
                        lambda)), 3)
```

```
##           0      1      2      3      4      5
## [1,] 0.344 0.380 0.188 0.066 0.014 0.008
## [2,] 0.368 0.368 0.184 0.061 0.015 0.003
```

3 接受拒绝法

3.1 接受拒绝法 (Acceptance-rejection method)

- 如果分布函数 $F(x)$ 难以求逆, 如何抽样?
- 如果来自于概率密度函数 $g(y)$ 的随机变量比较好产生, 我们就可以借助 $g(y)$ 来产生概率密度函数为 $f(x)$ 的随机变量 X 。

算法流程:

假设随机变量 X 和 Y 的密度函数为 f 和 g , 并存在常数 c 使得

$$\frac{f(x)}{cg(x)} \leq 1, \quad \forall x \in \{t : f(t) > 0\}.$$

1. 生成一个来自于 $\text{Unif}(0, 1)$ 的随机变量 U ;
2. 生成一个独立于 U 的随机变量 Y , 使其服从概率密度函数为 $g(y)$;
3. 如果 $U \leq \frac{f(Y)}{cg(Y)}$, 则令 $X = Y$, 否则拒绝 Y , 返回第 2 步。

Think: c 可以小于 1 吗?

算法原理：概率空间

- 算法本身的概率空间：由两个独立的随机变量构成： $Y \sim g$ 和 $U \sim \text{Unif}(0, 1)$ 。定义接受事件

$$A = \left\{ U \leq \frac{f(Y)}{cg(Y)} \right\}.$$

- 输出变量 X 的概率分布：算法只在事件 A 发生时输出 Y ，因此 X 的分布是 Y 在事件 A 下的条件分布，即 $X \sim \mathcal{L}(Y \mid A)$ 。

算法原理:

1. X 的 CDF 为

$$P(X \leq x) = P(Y \leq x \mid A) = \frac{P(Y \leq x, A)}{P(A)}$$

2. 一方面,

$$P(Y \leq x, A) = \int_{-\infty}^x \int_0^{\frac{f(y)}{cg(y)}} f_{Y,U}(y, u) du dy.$$

3. 另一方面,

$$P(A) = P(Y \leq \infty, A)$$

4. 因此, $P(X \leq x) = F(x)$.

接受拒绝算法的性质

Question: 记 $T \in \{1, 2, \dots\}$ 为接受一次需要的试验次数。请回答：

1. T 满足什么分布？
2. 平均需要多少次实验才能接受一次？

接受拒绝算法的性质

Question: 记 $T \in \{1, 2, \dots\}$ 为接受一次需要的试验次数。请回答：

1. T 满足什么分布？
2. 平均需要多少次实验才能接受一次？

Solution: 每次试验的接受概率为 $P(A) = 1/c$ ，因此

$$T \sim \text{Geom}(1/c), \quad E(T) = c.$$

也就是说，平均需要重复 c 次才能接受一次。为了提高效率，应尽量减小 c 。

提议分布 Y 的选择

接受—拒绝法中，我们希望提议分布满足：

- Y 容易产生，例如来自前面介绍的常见分布；
- g 的 support set 与 f 一致，且函数形状尽量接近 f ；
- 这样可以减小 c ，降低接受一个样本所需的平均试验次数。

例子 3.1

产生一个来自于以下密度函数的样本：

$$f(x) = 60x^3(1-x)^2, 0 \leq x \leq 1.$$

- $f(x)$ 在 $x = 0.6$ 处取得最大值 $f(0.6) = 2.074$.
- 令 $g(y)$ 为 $U(0, 1)$ 分布的密度函数，即 Y 从均匀分布 $U(0, 1)$ 产生。
- 令 $c = 2.074$ ，则 $f(x)/(cg(x)) \leq 1$.
- 从均匀分布 $U(0, 1)$ 产生变量 U ，如果 $U \leq \frac{f(Y)}{cg(Y)} = 60y^3(1-y)^2/2.074$ ，接受，并令 $X = Y$ 。否则，拒绝。

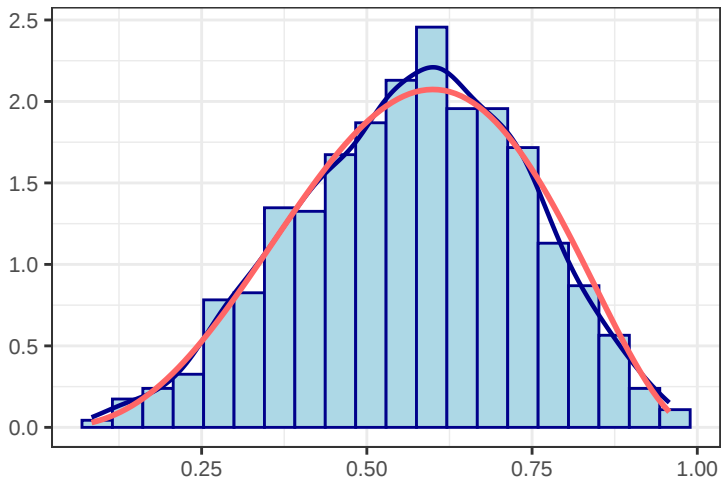
模拟代码

```
n <- 1000          # 需要的样本数
k <- 0             # 用以记录接受的次数
j <- 0             # 用以记录总共产生的随机数数量
x <- numeric(n)
while(k < n){
  u <- runif(1)
  j <- j + 1
  y <- runif(1)
  if(u < 60*y^3*(1-y)^2/2.074){
    # 接受
    k <- k + 1
    x[k] <- y
  }
}
print(sprintf("%d %d %.3f %.3f", k, j, k/j, 1/2.074))
```

```
## [1] "1000 2149 0.465 0.482"
```

```
p31 <- ggplot(data.frame(x = x), aes(x = x)) +  
  geom_histogram(  
    aes(y = after_stat(density)), bins = 20,  
    color = "darkblue", fill = "lightblue") +  
  geom_density(color = "darkblue", linewidth = 0.8) +  
  stat_function(fun = function(x) 60 * x^3 * (1 - x)^2,  
               color = "#FF6666", linewidth = 1) +  
  labs(x = "", y = "",  
       title = expression(f(x) == 60 * x^3 * (1 - x)^2)) +  
  theme_bw() +  
  theme(  
    plot.title = element_text(size = 8, hjust = 0.5),  
    axis.text = element_text(size = 8),  
    axis.title = element_text(size = 8)  
  )
```

$$f(x) = 60x^3(1-x)^2$$



例子 3.2

产生满足 $\Gamma(3/2, 1)$ 分布的样本，密度为

$$f(x) = \frac{x^{1/2}e^{-x}}{\Gamma(3/2)}, \quad \forall x > 0.$$

- 考虑到取值范围以及指数函数的存在，我们考虑用指数分布生成 Y 。由于 $\Gamma(3/2, 1)$ 的期望是 $3/2$ ，我们假设 Y 的密度函数为

$$g(y) = \frac{2}{3}e^{-\frac{2}{3}y}.$$

- 简单计算可知

$$\frac{f(x)}{g(x)} \leq c = \frac{3\sqrt{3}}{\sqrt{2\pi e}} \approx 1.257.$$

- 从均匀分布 $\text{Unif}(0, 1)$ 产生变量 U ，如果 $U \leq \frac{f(Y)}{cg(Y)} = \sqrt{2e/3}y^{1/2}e^{-y/3}$ ，接受，并令 $X = Y$ 。否则，拒绝。

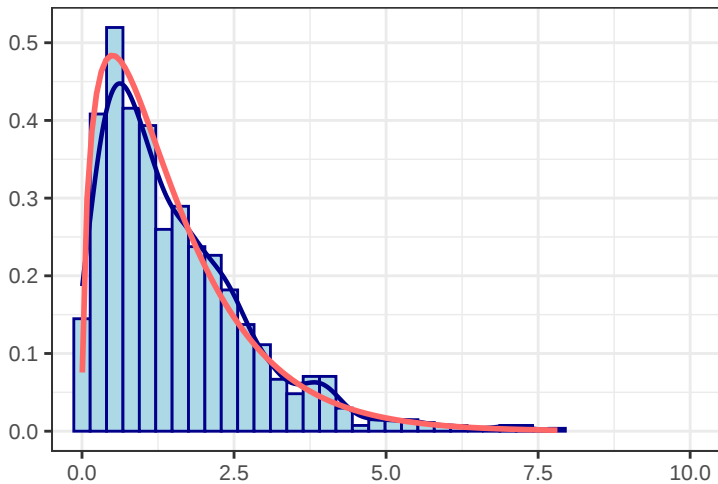
模拟代码

```
n <- 1000          # 需要的样本数
k <- 0             # 用以记录接受的次数
j <- 0             # 用以记录总共产生的随机数数量
x <- numeric(n)
while(k < n){
  u <- runif(1)
  j <- j + 1
  y <- rexp(1, 2/3)
  if(u < sqrt(2*exp(1)/3) * sqrt(y) * exp(-y/3)){
    # 接受
    k <- k + 1
    x[k] <- y
  }
}
print(sprintf("%d %d %.3f %.3f", k, j, k/j, 1/1.257))
```

```
## [1] "1000 1263 0.792 0.796"
```

```
p32 <- ggplot(data.frame(x = x), aes(x = x)) +  
  geom_histogram(  
    aes(y = after_stat(density)), bins = 30,  
    color = "darkblue", fill = "lightblue") +  
  geom_density(color = "darkblue", linewidth = 0.8) +  
  stat_function(fun = dgamma,  
                args = list(shape = 3/2, rate = 1),  
                color = "#FF6666", linewidth = 1) +  
  coord_cartesian(xlim = c(0, 10)) +  
  labs(x = "", y = "", title = expression(Gamma(3/2, 1))) +  
  theme_bw() +  
  theme(  
    plot.title = element_text(size = 8, hjust = 0.5),  
    axis.text = element_text(size = 8),  
    axis.title = element_text(size = 8)  
  )
```

$$\Gamma(3/2, 1)$$



方法总结

逆变换法

- 优点：只要能够计算（或数值求解） F_X^{-1} ，算法直接，没有拒绝样本。
- 缺点：需要先求出并计算 CDF 的逆；对复杂分布，显式求逆可能困难，数值求逆也可能代价较高。

接受拒绝算法

- 优点：不需求解目标分布的 CDF 逆，能处理许多复杂分布。
- 缺点：会产生被拒绝的候选样本， c 较大时效率低；需要合理选择 g 和 c 。