《统计算法基础》第三次作业

姓名：王凯栋　　学号:PB20071441　　日期：2023/4/8

目录

一. 实验目的

- 复习 Copula Model 的相关问题
- 了解分层重要抽样减小方差的方法
- 掌握 Monte Carlo 方法在统计推断中的应用

二. 实验过程

- 完成 Copula 与分层抽样相关习题
- 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

1.(Copula model) Consider a copula $C(u_1, u_2, \cdots, u_d)$.Prove that for any $u_i \in [0,1], i = 1, 2, \cdots, d$,we have

$$max\{1 - d + \sum_{i=1}^{d} u_i, 0\} \leq C(u_1, u_2, \cdots, u_d) \leq min\{u_1, u_2, \cdots, u_d\}$$

解：由定义，考虑一个多元 Cdf F 以及边缘 cdf$F_1, F_2, \cdots, F_d$, 则 Copula 可以定义为

$$C(u_1, u_2, \cdots, u_d) = P(F_1(X_1) \leq u_1, F_2(X_2) \leq u_2, \cdots, F_d(X_d) \leq u_d)$$

所以，容易知道，由于

$$(F_1(X_1) \leq u_1, F_2(X_2) \leq u_2, \cdots, F_d(X_d) \leq u_d) \quad (F_i(X_i) \leq u_i, 1 \leq i \leq d)$$

由 $F_i(X_i) \sim U(0,1)$, 所以对任意的 i 有

$$C(u_1, u_2, \cdots, u_d) = P(F_1(X_1) \leq u_1, F_2(X_2) \leq u_2, \cdots, F_d(X_d) \leq u_d) \leq P(F_i(X_i) \leq u_i) = u_i$$

所以

$$C(u_1, u_2, \cdots, u_d) \leq min\{u_1, u_2, \cdots, u_d\}$$

因为 $C(u_1, u_2, \cdots, u_d)$ 表示的是一个概率，所以它大于等于 0 必然是成立的

$C(u_1, u_2, \cdots, u_d) \geq 1-d+\sum_{i=1}^{d} u_i$ 当且仅当 $1-C(u_1, u_2, \cdots, u_d) \leq \sum_{i=1}^{d}(1-u_i)$

$$\begin{aligned}
&1 - C(u_1, u_2, \cdots, u_d) \\
=&1 - P(F_1(X_1) \leq u_1, F_2(X_2) \leq u_2, \cdots, F_d(X_d) \leq u_d) \\
=&P(\exists F_i, X_i, s.t F_i(X_i) \geq u_i) \\
\leq&\sum_{i=1}^{d} P(F_i(X_i) \geq u_i)(Union\ Bound) \\
=&\sum_{i=1}^{d}(1 - u_i)
\end{aligned}$$

综上，有

$$max\{1 - d + \sum_{i=1}^{d} u_i, 0\} \leq C(u_1, u_2, \cdots, u_d) \leq min\{u_1, u_2, \cdots, u_d\}$$

2.(Stratified importance sampling):《统计计算使用 R》：page 135/例 5.13

在例 5.10 中我们通过重要函数 $f_3(x) = e^{-x}/(1 - e^{-1}), 0 < x < 1$ 得到最好的结果。通过 10000 次重复实验我们得到了估计值 $\hat{\theta} = 0.5257801$

和估计标准误差 0.0970314。现在我们把区间 $(0,1)$ 分成五个子区间 $(a_{j-1},a_j), j=1,\cdots,5$, 其中 $a_0=0, a_5=1, a_1,a_2,a_3,a_4$ 为 f_3 五个分位数，即 $\int_{a_{j-1}}^{a_j} f_3(x)dx=\frac{1}{5}, j=1,2,\cdots,5.$

在第 j 个子区间 (a_{j-1},a_j) 上根据密度

$$\frac{5e^{-x}}{1-e^{-1}}, x\in(a_{j-1},a_j)$$

生成随机变量，实现过程留作练习

解：

因为

$$f(x)=\frac{e^{-x}}{1-e^{-1}}$$

then,

$$F(x)=\int_0^x f(x)dx=\frac{1}{1-e^{-1}}-\frac{e^{-x}}{1-e^{-1}}$$

$$F^{-1}(u)=-log(1-(1-e^{-1})u)$$

由定义 $a_j=F^{-1}(j/5), j=1,2,3,4$

在第 j 个子区间 (a_{j-1},a_j) 上根据密度

$$\frac{5e^{-x}}{1-e^{-1}}, x\in(a_{j-1},a_j)$$

这个概率密度 CDF 对应的逆为

$$F^{-1}(u)=-log(e^{-a_{j-1}}-\frac{(1-e^{-1})u}{5})$$

```
F_h <-function(u){
  -log(1-(1-exp(-1))*u)
}

# 找出几个分位数
a <- c(0,F_h(1/5),F_h(2/5),F_h(3/5),F_h(4/5),1)
# 定义各个区间上的被积函数
g1 <- function(x){
```

```
  exp(-x-log(1+x^2))*(x>a[1])*(x<a[2])
}
g2 <- function(x){
  exp(-x-log(1+x^2))*(x>a[2])*(x<a[3])
}
g3 <- function(x){
  exp(-x-log(1+x^2))*(x>a[3])*(x<a[4])
}
g4 <- function(x){
  exp(-x-log(1+x^2))*(x>a[4])*(x<a[5])
}
g5 <- function(x){
  exp(-x-log(1+x^2))*(x>a[5])*(x<a[6])
}
# 定义重要函数
f <- function(x){
  5*exp(-x)/(1-exp(-1))
}
# 给定初值
N <- 10000
m =1000
estimates <- numeric(m)
T <- numeric(5)


# 计算 Monte Carlo 估计
for(i in 1:m){
# 实现各个分层的随机数产生方式
u1 <- runif(N/5,a[1],a[2])
f_1 <- -log(exp(-a[1])-(1-exp(-1))*u1/5)
u2 <- runif(N/5,a[2],a[3])
f_2 <- -log(exp(-a[2])-(1-exp(-1))*u2/5)
u3 <- runif(N/5,a[3],a[4])
```

```r
f_3 <- -log(exp(-a[3])-(1-exp(-1))*u3/5)
u4 <- runif(N/5,a[4],a[5])
f_4 <- -log(exp(-a[4])-(1-exp(-1))*u4/5)
u5 <- runif(N/5,a[5],a[6])
f_5 <- -log(exp(-a[5])-(1-exp(-1))*u5/5)

# 计算每一层得到的估计
  T[1] <- mean(g1(f_1)/f(f_1))
  T[2] <- mean(g2(f_2)/f(f_2))
  T[3] <- mean(g3(f_3)/f(f_3))
  T[4] <- mean(g4(f_4)/f(f_4))
  T[5] <- mean(g5(f_5)/f(f_5))
  estimates[i] <- sum(T)
}
# 计算估计的均值与误差
mean = mean(estimates)
sd = sd(estimates)
rate = ((0.0970314)^2-sd^2)/(0.0970314)^2

cat(" 得到的 Monte Carlo 估计为: ",mean," 方差缩减了",rate*100,"%")
```

```
## 得到的Monte Carlo估计为:  0.5203761 方差缩减了 99.99997 %
```

3.(Monte Carlo for estimation)

- 《统计计算使用 R》：第六章 6.6.

用蒙特卡洛方法来估计正态情况下的偏度 $\sqrt{b_1}$ 的 0.025，0.05，0.95，0.975 分位数，使用密度的（具有精确方差公式的）正态近似来计算式 $Var(\hat{x_q}) = \frac{q(1-q)}{nf(x_q)^2}$ 中估计的标准误差。将估计分位数和大样本近似 $\sqrt{b_1} \approx N(0,6/n)$ 的分位数进行比较。

解：偏度的计算公式为

$$b = \frac{m_3}{m_2^{3/2}}$$

其中 $m_2 = \frac{1}{n}\sum_i (x_i-\mu)^2, m_3 = \frac{1}{n}\sum_i (x_i-\mu)^3$

下面以标准正态分布为例来计算, 所以对应的 pdf 为:

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

```
set.seed(123)
m <- 1000
N <- 1000
f<- function(x){
  1/sqrt(2*pi)*exp(-x^2/2)
}
y <- numeric(m)
for(i in 1:m){
  x <- rnorm(N,0,1)
  m2 <- sum((x-mean(x))^2)/N
  m3 <- sum((x-mean(x))^3)/N
  y[i]=m3/(m2^(3/2))
}
sig <- c(0.025,0.05,0.95,0.975)
z <- sort(y)[sig*m]
var <- numeric(4)
sd <- numeric(4)
for(i in 1:4){
  var[i]=(sig[i]*(1-sig[i]))/(m*f(z[i])^2)
  sd[i]=sqrt(var[i])
}
cat("Monte Carlo 得到的四个分位数对应的标准误差分别为: ",sd)
```

```
## Monte Carlo得到的四个分位数对应的标准误差分别为:  0.01253264 0.01743206 0.01740661 0
```

```
# 大样本对比
estimate <- matrix(0,2,4)
for(i in 1:4){
  estimate[1,i]=z[i]
```

```
  estimate[2,i]=qnorm(sig[i],0,sqrt(6/m))
}
estimate
```

```
##              [,1]       [,2]      [,3]      [,4]
## [1,] -0.1588662 -0.1342207 0.1228547 0.1482187
## [2,] -0.1518182 -0.1274098 0.1274098 0.1518182
```

可见, 我们通过 Monte Carlo 得到的估计与大样本估计十分相近。

- 《统计计算》: 习题三 15: 比较（1）（2）（3）三种置信区间。固定 $p = 0.7$, 通过 Monte Carlo 探索三种置信区间的置信水平与 n 和 α 的关系。

解：设 $X \sim b(1,p)$,$X_1, X_2, \cdots, X_n$ 为样本。令 $S_0 = \sum_{i=1}^n X_i, \hat{p} = S_0/n = \frac{1}{n}\sum_{i=1}^n X_i$

该检验对应的零假设为

$$H_0 : p = 0.7$$

(1) 利用正态近似。当 n 很大时

$$\frac{\hat{p} - p}{\sqrt{\frac{1}{n}\hat{p}(1-\hat{p})}}$$

近似服从 N(0,1), 于是得到置信区间

$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{1}{n}\hat{p}(1-\hat{p})}$$

```
set.seed(123)
p <- 0.7
m <- 1000
n <- c(100,1000,10000)
alpha <- c(0.05,0.5,0.95)
level <- matrix(0,3,3)
for(i in 1:length(n)){
  for(j in 1:length(alpha)){
```

```r
    lower = numeric(m)
    upper = numeric(m)
    for(k in 1:m){
      s <- rbinom(n[i],1,p)
      p_hat <- mean(s)
      lower[k] <- p_hat-qnorm(1-alpha[j]/2)*sqrt(1/n[i]*p_hat*(1-p_hat))
      upper[k] <- p_hat+qnorm(1-alpha[j]/2)*sqrt(1/n[i]*p_hat*(1-p_hat))
    }
    level[i,j] <- mean((lower<p)&(upper>p))
  }
}
colnames(level) <- c("alpha=0.05","alpha=0.5","alpha=0.95")
rownames(level) <- c("n=100","n=1000","n=10000")
level
```

```
##          alpha=0.05 alpha=0.5 alpha=0.95
## n=100         0.959     0.488      0.073
## n=1000        0.952     0.504      0.022
## n=10000       0.957     0.498      0.038
```

可见, 第一个置信区间，得到的置信水平估计与 $1-\alpha$ 十分接近，且随着 n 的增大，它们之间的误差逐渐减小, 且 α 比较大的时候，较小的 n 就可以比较接近真实的置信水平，但是 α 较小时，虽然 n 越大越精确，但是即使达到 10000 也仍然有比较大的误差。

(2) 利用正态近似，

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i-\hat{p})^2 = \frac{n}{n-1}\hat{p}(1-\hat{p})$$

n 很大时

$$\frac{\hat{p}-p}{\sqrt{\frac{1}{n}S^2}}$$

近似服从 $N(0,1)$, 于是得到置信区间

$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{1}{n}S^2}$$

```
set.seed(123)
p <- 0.7
m <- 1000
n <- c(100,1000,10000)
alpha <- c(0.05,0.5,0.95)
level1 <- matrix(0,3,3)
for(i in 1:length(n)){
  for(j in 1:length(alpha)){
    lower = numeric(m)
    upper = numeric(m)
    for(k in 1:m){
      s <- rbinom(n[i],1,p)
      p_hat <- mean(s)
      lower[k] <- p_hat-qnorm(1-alpha[j]/2)*sqrt(1/n[i]*p_hat*(1-p_hat)*n[i]/(n[i]-1))
      upper[k] <- p_hat+qnorm(1-alpha[j]/2)*sqrt(1/n[i]*p_hat*(1-p_hat)*n[i]/(n[i]-1))
    }
    level1[i,j] <- mean((lower<p)&(upper>p))
  }
}
colnames(level1) <- c("alpha=0.05","alpha=0.5","alpha=0.95")
rownames(level1) <- c("n=100","n=1000","n=10000")
level1
```

```
##          alpha=0.05 alpha=0.5 alpha=0.95
## n=100         0.959     0.565      0.073
## n=1000        0.952     0.504      0.022
## n=10000       0.957     0.505      0.038
```

由于样本量比较大，所以 $\frac{n}{n-1}$ 与 1 的区别非常小，除了 $\alpha = 0.5$, 其他得到的结果与 (1) 十分相近。

(3)Wilson 置信区间。利用正态近似，n 很大时

$$\frac{\hat{p}-p}{\sqrt{\frac{1}{n}p(1-p)}}$$

近似服从 $N(0,1)$, 解关于 p 的不等式

$$|\frac{\hat{p}-p}{\sqrt{\frac{1}{n}p(1-p)}}| \le z_{1-\alpha/2}$$

得到置信区间 ($\lambda = z_{1-\alpha/2}$)

$$\frac{\hat{p}+\frac{\lambda^2}{2n}}{1+\frac{\lambda^2}{n}} \pm \frac{\lambda\sqrt{\frac{\lambda^2}{4n}+\hat{p}(1-\hat{p})}}{\sqrt{n}(1+\frac{\lambda^2}{n})}$$

```
set.seed(123)
p <- 0.7
m <- 1000
n <- c(100,1000,10000)
alpha <- c(0.05,0.5,0.95)
level <- matrix(0,3,3)
for(i in 1:length(n)){
  for(j in 1:length(alpha)){
    lower = numeric(m)
    upper = numeric(m)
    for(k in 1:m){
      s <- rbinom(n[i],1,p)
      p_hat <- mean(s)
      lamba = qnorm(1-alpha[j]/2)
      base = (p_hat+lamba^2/(2*n[i]))/(1+lamba^2/n[i])
      sd = lamba*sqrt(lamba^2/(4*n[i])+p_hat*(1-p_hat))/(sqrt(n[i])*(1+lamba^2/n[i]))
      lower[k] <- base - sd
      upper[k] <- base + sd
    }
    level[i,j] <- mean((lower<p)&(upper>p))
  }
}
colnames(level) <- c("alpha=0.05","alpha=0.5","alpha=0.95")
rownames(level) <- c("n=100","n=1000","n=10000")
level
```

```
##          alpha=0.05 alpha=0.5 alpha=0.95
## n=100         0.947     0.565      0.073
## n=1000        0.954     0.504      0.022
## n=10000       0.956     0.498      0.038
```

可见，得到的置信区间也非常接近 $1-\alpha$, 且在 α 比较大时，得到的结果不太精确，但是随着 n 的增大，较为靠近真实值.

4.(Monte Carlo hypothesis test)《统计计算使用 R》：第六章 6.2，6.A

- 6.2 将例 6.9 中 t 检验的备择假设换成 $H_1: \mu \neq 500$, 并保持显著水平 $\alpha = 0.05$ 不变，绘制该检验的经验功效曲线。

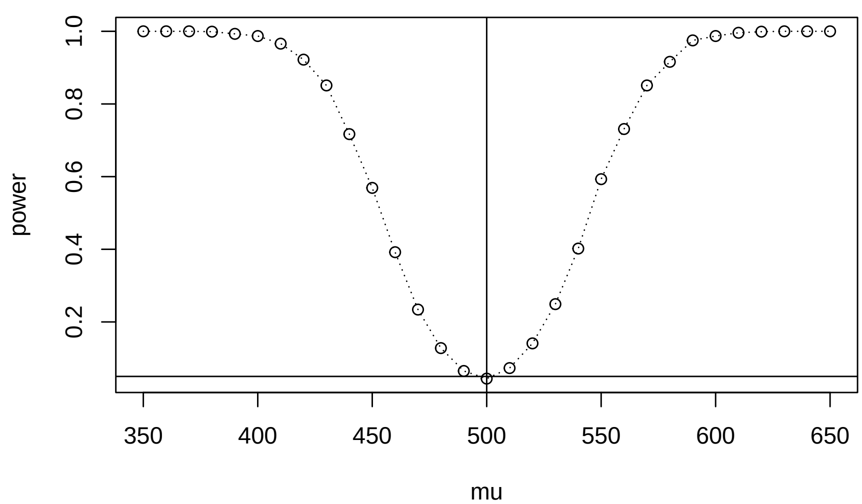
```
n<-20
m<-1000
mu0 <- 500
sigma <- 100
mu <- c(seq(350,650,10)) #alternative
M <- length(mu)
power <- numeric(M)
for(i in 1:M){
  mu1 <- mu[i]
  pvalues <- replicate(m,expr={
    #simulate under alternative mu1
    x <- rnorm(n,mean = mu1,sd = sigma)
    ttest = t.test(x,
                   alternative = "two.sided",mu = mu0)
    ttest$p.value
  })
  power[i] <- mean(pvalues <= 0.05)
}
```

```
# 画图
```

```
plot(mu,power)
abline(v = mu0,lty = 1)
abline(h = 0.05,lty = 1)

#add standard errors
se <- sqrt(power * (1-power)/m)

lines(mu,power,lty = 3)
```



从上面的功效曲线可以看出，经验检验功效 $\hat{\pi}(\theta)$ 在 θ 接近 $\theta_0 = 500$ 时比较小，当 θ 远离 θ_0 时开始增大。

- 6.A 当抽样总体非正态时，使用蒙特卡罗模拟来研究 t 检验的经验第一类错误率是否约等于理论显著水平 α.t 检验对微小的正态性偏离是稳定的. 讨论对下列抽样总体进行模拟的结果：(i)$\chi^2(1)$;(ii)Uniform(0,2);(iii)Exponential(1). 在每种情况下都检验 $H_0: \mu = \mu_0, H_1: \mu \neq \mu_0$, 其中 μ_0 分别为 $\chi^2(1), Uniform(0,2)$ 和 $Exponential(1)$ 的均值

解:

为 $\chi^2(1), Uniform(0,2)$ 和 $Exponential(1)$ 的均值均为 1，检验为双边检验

```
n <-c(20,200,2000)
alpha <- 0.05
mu0 <- 1
typeerror <- matrix(0,3,3)
colnames(typeerror) <- c("chisq(1)","Uniform(0,2)","Exponential(1)")
rownames(typeerror) <- c("n=20","n=200","n=2000")
# chisq(1)
for(i in 1:length(n)){
  m <- 1000
  p <- numeric(m)
  for(j in 1:m){
    x <- rchisq(n[i],mu0)
    ttest <- t.test(x,alternative = "two.sided",mu=mu0)
    p[j] <- ttest$p.value
  }
  typeerror[i,1] <- mean(p<alpha)

}

# Uniform(0,2)
for(i in 1:length(n)){
  m <- 10000
  p <- numeric(m)
  for(j in 1:m){
    x <- runif(n[i],0,2)
    ttest <- t.test(x,alternative = "two.sided",mu=mu0)
    p[j] <- ttest$p.value
  }
  typeerror[i,2] <- mean(p<alpha)
}
```

```
# Exponential(1)
for(i in 1:length(n)){
  m <- 10000
  p <- numeric(m)
  for(j in 1:m){
    x <- rexp(n[i],mu0)
    ttest <- t.test(x,alternative = "two.sided",mu=mu0)
    p[j] <- ttest$p.value
  }
  typeerror[i,3] <- mean(p<alpha)
}

typeerror
```

```
##          chisq(1) Uniform(0,2) Exponential(1)
## n=20       0.110       0.0531         0.0806
## n=200      0.057       0.0516         0.0497
## n=2000     0.049       0.0466         0.0526
```

可见 $\chi^2(1)$ 和 $Exp(1)$ 在 n 比较小的时候，第一型错误率和 α 还有比较大的差距，但是当 n 逐渐增大，这三个非正态总体的第一型错误率都接近 $\alpha = 0.05$, 正好验证了中心极限定理。

5.(Markov chain Monte Carlo)《统计计算使用 R》：第九章 9.4，9.6

- 9.4 实现一个随机游动 Metropolis 样本生成器来生成标准 Laplace 分布。通过一个正态分布来模拟增量。对由方差不同的建议分布所生成的链条进行比较。此外，计算每个链条的接受率。

解：标准 Laplace 分布对应的概率密度函数为：

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

对应的接受概率为：

$$\alpha(X_t,y)=min\{1,\frac{f(Y)}{f(X_t)}\}$$

```
f <- function(x){
  1/2*exp(-abs(x))
}

#Metropolis 算法函数
rw.Metropolis <- function(sigma,x0,N){
  x <- numeric(N)
  x[1] <- x0
  u <- runif(N)
  k <- 0 # 计算接受次数
  for(i in 2:N){
    y <- rnorm(1,x[i-1],sigma)
    if(u[i]<=f(y)/f(x[i-1])){
      x[i] <- u[i]
      k <- k+1# 接受后次数 +1
    }
    else{
      x[i] <- x[i-1]
    }
  }
  return(list(x=x,k=k))
}
```

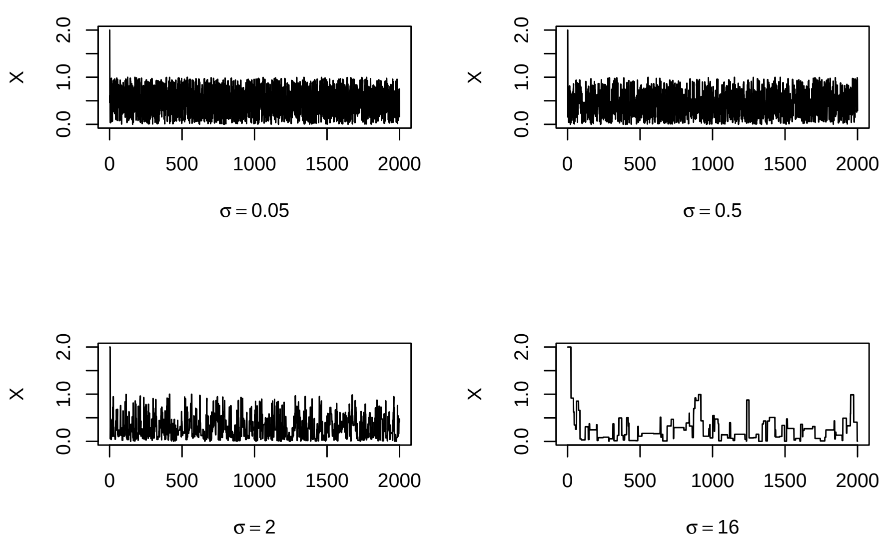
```
# 比较接受率
N <- 2000
sigma <- c(0.05,0.5,2,16)
x0 <- 2 # 给定初值为 2
rw1 <- rw.Metropolis(sigma[1],x0,N)
rw2 <- rw.Metropolis(sigma[2],x0,N)
```

```
rw3 <- rw.Metropolis(sigma[3],x0,N)
rw4 <- rw.Metropolis(sigma[4],x0,N)
print(c(rw1$k,rw2$k,rw3$k,rw4$k)/N)
```

```
## [1] 0.9785 0.8300 0.4325 0.0575
```

可见，方差较小时，生成的链条接受率比较高，因为接受率在 $[0.5, 0.85]$ 范围内，Markov 链有比较好的性质，所以这四个选择的方差中，$\sigma = 0.5$ 有比较好的性质。

```
# 比较方差不同的链条分布
par(mfrow=c(2,2))
rw <- cbind(rw1$x,rw2$x,rw3$x,rw4$x)
for(j in 1:4){
  plot(rw[,j],type = "l",xlab = bquote(sigma ==.(round(sigma[j],3))),ylab = "X",
       ylim = range(rw[,j]))
}
```



- 9.6 Rao 给出了一个关于四个纲 197 种动物的基因连锁的例子。群体大小为 $(125, 18, 20, 34)$。假设响应的多项分布的概率为

$$(\frac{1}{2}+\frac{\theta}{4}, \frac{1-\theta}{4}, \frac{1-\theta}{4}, \frac{\theta}{4})$$

给定观测样本，使用本章中的一种方法估计 θ 的后验分布。

解：

$$f(\beta|x_1, x_2, \cdots, x_4) \propto p(x_1, x_2, \cdots, x_4|\theta)\pi(\theta) \propto (2+\theta)^{x_1}(1-\theta)^{x_2+x_3}\theta^{x_4}$$

Metropolis 算法的接受概率为：

$$\alpha(\beta_t, Y) = min\{1, \frac{f(Y|x_1, \cdots, x_4)}{f(\beta_t|x_1, \cdots, x_4)}\}$$

群体的观测样本已经给定，为：

```
win <- c(125,18,20,34)
```

不妨假定建议分布为均匀分布 $U(0, 1)$

```
set.seed(3)
prob <- function(y,x,win){
  ((2+y)/(2+x))^(win[1])*((1-y)/(1-x))^(win[2]+win[3])*(y/x)^(win[4])
}




m <- 5000
u <- runif(m)
v <- runif(m,0,1)
x[1] <- 0.25
k <- 0
```

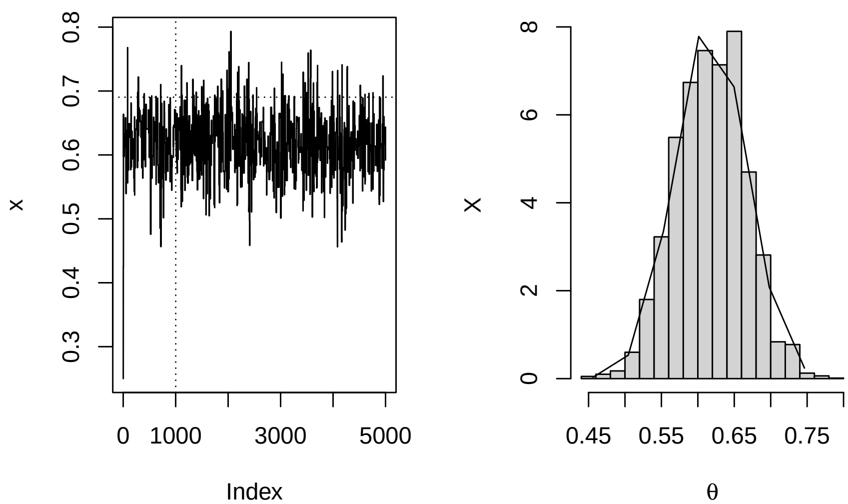
```
for(i in 2:m){
  y <- v[i]
  if(u[i]<= prob(y,x[i-1],win)) x[i] <- y
  else{
    x[i] <- x[i-1]
    k <- k+1
  }
}

k/m
```

```
## [1] 0.8392
```

拒绝率为 0.8392

```
b = win[4]*4/197
burn = 1000
par(mfrow = c(1,2))
plot(x,type = "l")
abline(h = b,v = burn+1,lty = 3)
xb <- x[-(1:burn)]
hist(xb,prob= TRUE,xlab = bquote(theta),ylab = "X",main = "")
z <- seq(min(xb),max(xb),sd(xb))
lines(z,dnorm(z,mean(xb),sd(xb)))
```




```
print(mean(xb))
```

```
## [1] 0.617731
```

```
print(sd(xb))
```

```
## [1] 0.04836383
```

上面分别为 θ 的概率分布图，对应的点估计为 $\theta = 0.62$