HW1

杜金鸿 15338039

目录

习题三

2. 考虑定积分

$$I = \int_{-1}^{1} e^x \mathrm{d}x = e - e^{-1}$$

(1) 用随机模拟方法计算定积分 I, 分别用随机投点法、平均值法、重要抽样法和分层抽样法计算。

```r
set.seed(0)
h <- function(x){
        exp(x)
}
I <- integrate(h,-1,1)$value
cat('Numeric Integration    : ',I)
```

```
## Numeric Integration    :  2.350402
```

随机投点法

当 $x \in [-1,1]$, $h(x) = e^x \in [e^{-1}, e]$。

```r
set.seed(0)
# 随机投点法
# Stochastic Point Method
SPM <- function(
    h,        # density h(x) to be integrated
    from,     # left end point of x
    to,       # right end point of x
    M,        # the upper bound of h(x) in [from,to]
    N         # the number of points to be generated
){
    x <- runif(N, min = from, max = to)
    y <- runif(N, min = 0, max = M)
    hx <- h(x)
    p_hat <- mean(y<=hx)
    I_hat <- p_hat*M*(to-from)
    VarI_hat <- I_hat*(M*(to-from)-I_hat)/N
        return(list(I_hat,VarI_hat))
}

N <- 100000
SPMresult <- SPM(h,-1,1,exp(1),N)
I1 <- SPMresult[[1]]
```

```
VarI1 <- SPMresult[[2]]
cat('Stochastic Point Method: ',I1)
```

```
## Stochastic Point Method:  2.334569
```

平均值法

```
set.seed(0)
# 平均值法
# Mean Value Method
MVM <- function(
    h,        # density h(x) to be integrated
    from,     # left end point of x
    to,       # right end point of x
    N         # the number of points to be generated
){
    x <- runif(N, min = from, max = to)
    hx <- h(x)
    I_hat <- (to-from)* mean(hx)
    VarI_hat <- mean(((to-from)*hx-I_hat)^2)/N
        return(list(I_hat,VarI_hat))
}

MVMresult <- MVM(h,-1,1,N)
I2 <- MVMresult[[1]]
VarI2 <- MVMresult[[2]]
cat('Mean Value Method: ',I2)
```

```
## Mean Value Method:  2.350865
```

重要抽样法

$$
\begin{aligned}
e^x &= 1 + x + \frac{x^2}{2!} + \cdots \\
&\approx 1 + x
\end{aligned}
$$

取

$$
g(x) = \frac{1}{3}\left(\frac{3}{2} + x\right), \qquad x \in [-1, 1]
$$

要产生 $g(x)$ 的随机数可以用逆变换法，密度 $g(x)$ 的分布函数 $G(x) = \frac{1}{6}\left(\frac{3}{2}+x\right)^2 - \frac{1}{24}, \qquad x \in [-1, 1]$, 的反函数为

$$
G^{-1}(y) = \sqrt{6y + \frac{1}{4}} - \frac{3}{2}, \quad 0 < y < 1
$$

因此，取 $U_i \overset{iid}{\sim} U(0,1)$, 令 $X_i = \sqrt{2U_i} - \frac{3}{2}$, $i = 1, 2, \ldots, N$, 则重要抽样法的积分公式为

$$\hat{I}_3 = \frac{1}{N} \sum_{i=1}^{N} \frac{e^{X_i}}{\frac{1}{3}\left(\frac{3}{2} + x\right)}$$

当 $g(x)$ 在被积区域 C 以为取值不为 0 时，需要计算其在 C 上的条件概率，因此以下函数可能需要输入 min_G 和 max_G 用来处理这种特殊情况。

```
set.seed(0)
# 重要抽样法
# Important Sampling Method
ISM <- function(
    h,        # density h(x) to be integrated
    g,        # test density g(x)
    G_inv,    # inverse function of G(x)
    N,        # the number of points to be generated
        min_G = 0,
        max_G = 1
){
    U <- runif(N,min_G,max_G)
    X <- G_inv(U)
    I_hat <- mean(h(X)/g(X)*(max_G-min_G))
    VarI_hat <- mean((h(X)/g(X)*(max_G-min_G)-I_hat)^2)/N
        return(list(I_hat,VarI_hat))
}

g <- function(x){return((3/2+x)/3)}
G_inv <- function(y){return(sqrt(6*y+1/4)-3/2)}
ISMresult <- ISM(h,g,G_inv,N)
I3 <- ISMresult[[1]]
VarI3 <- ISMresult[[2]]
cat('Number Important Sampling Method: ',I3)
```

```
## Number Important Sampling Method:  2.350796
```

分层抽样法

```
set.seed(0)
# 分层抽样法
# Stratified Sampling Method
SSM <- function(
    h,        # density h(x) to be integrated
```

```r
    from,     # left end point of x
    to,       # right end point of x
    level,
    N         # the number of points to be generated
){
    interval <- seq(from,to,length.out=level+1)
    I_hat <- 0
    VarI_hat <- 0
        for (i in c(1:level)){
        MVMresult <- MVM(h,interval[i],interval[i+1],as.integer(N/level))
        I_hat <- I_hat + MVMresult[[1]]
        VarI_hat <- VarI_hat + MVMresult[[2]]
    }
        return(list(I_hat,VarI_hat))
}
SSMresult <- SSM(h,-1,1,10,N)
I4 <- SSMresult[[1]]
VarI4 <- SSMresult[[2]]
cat('Stratified Sampling Method: ',I4)
```

```
## Stratified Sampling Method:  2.350112
```

(2) 设估计结果为 I，如果需要以 95% 置信度保证计算结果精度在小数点后三位小数，这四种方法分别需要计算多少次被积函数值?

由强大数定理

$$\hat{p}_1 = \frac{1}{N}\sum_{i=1}^{n}\xi_i \xrightarrow{a.s.} p \qquad N \to \infty$$

$$\hat{I}_1 = \hat{p}_1 M(b-a) \xrightarrow{a.s.} I \qquad N \to \infty$$

由中心极限定理

$$\frac{\sqrt{N}(\hat{p}_1 - p)}{\sqrt{p(1-p)}} \xrightarrow{D} N(0,1) \qquad N \to \infty$$

$$\sqrt{N}(\hat{I}_1 - I) = M(b-a)(\hat{p}_1 - p) \xrightarrow{D} N(0,[M(b-a)]^2 p(1-p)) \qquad N \to \infty$$

因此渐进方差为

$$Var\hat{I}_1 = \frac{1}{N}[M(b-a)]^2 p(1-p)$$

令

$$\mathbb{P}\left(|\hat{I}_1 - I| < 10^{-3}\right) = \mathbb{P}\left(\left|\frac{\hat{I}_1 - I}{\sqrt{\frac{1}{N}[M(b-a)]^2 p(1-p)}}\right| < \frac{10^{-3}}{\sqrt{\frac{1}{N}[M(b-a)]^2 p(1-p)}}\right)$$
$$= 1 - \alpha$$

得

$$\frac{10^{-3}}{\sqrt{\frac{1}{N}[M(b-a)]^2 p(1-p)}} = z_{1-\frac{1}{2}\alpha}$$
$$N = 10^6[M(b-a)]^2 p(1-p) z^2_{1-\frac{1}{2}\alpha}$$

我们也可用渐进方差的估计值 $\frac{1}{N}[M(b-a)]^2\hat{p}_1(1-\hat{p}_1) = \frac{1}{N}\hat{I}_1[M(b-a)-\hat{I}_1]$ 来估计置信区间，则

$$N = 10^6 \hat{I}_1[M(b-a)-\hat{I}_1] z^2_{1-\frac{1}{2}\alpha}$$

```r
a <- -1
b <- 1
M <- exp(1)-exp(-1)
p <- I/(b-a)/M
N1 <- ceiling(10^6*(M*(b-a))^2*p*(1-p)*qnorm(0.975)^2)
N12 <- ceiling(10^6*I1*(M*(b-a)-I1)*qnorm(0.975)^2)
cat('Required number of tests by Stochastic Point Method is : ',N1,' or ',N12)
```

```
## Required number of tests by Stochastic Point Method is :  21221723  or  21220759
```

平均值法由中心极限定理，

$$\sqrt{N}(\hat{I}_2 - I) \xrightarrow{D} N(0, (b-a)^2 Var(h(U))) \qquad N \to \infty$$

其中

$$Var(h(U)) = \int_a^b [h(u) - \mathbb{E}h(U)]^2 \frac{1}{b-a} \mathrm{d}u$$

因此 $\hat{I}_2 \xrightarrow{D} N(I, \frac{1}{N}(b-a)^2 Var(h(U)))$, 设 $\{Y_i\}$ 的样本方差为 S_N^2，则 I 的 $1-\alpha$ 近似 95% 置信区间为

$$\hat{I}_2 \pm \frac{z_{0.975} S_N}{\sqrt{N}}$$

, 令

$$\frac{z_{0.975} S_N}{\sqrt{N}} \le 10^{-3}$$

得

$$N \ge z^2_{0.975} \times 10^6 S_N^2$$

```
N2 <- ceiling(qnorm(0.975)^2*10^6*(VarI2*N))
cat('Required number of tests by Mean Value Method is : ',N2)
```

```
## Required number of tests by Mean Value Method is :  6660247
```

重要抽样法
$X_i \overset{iid}{\sim} g(x)$,

$$\begin{aligned}
Var\hat{I}_3 &= \sum_{i=1}^{N} Var\left(\frac{h(X_i)}{g(X_i)}\right) \\
&= \frac{1}{N}\left[\mathbb{E}\left(\frac{h^2(X)}{g^2(X)}\right) - I^2\right] \\
&= \frac{1}{N}\left[\int_{-1}^{1} \frac{e^{2x}}{\frac{1}{3}\left(\frac{3}{2}+x\right)} \mathrm{d}x - I^2\right] \\
&\approx \frac{3.368662}{N}
\end{aligned}$$

则

$$\mathbb{P}\left(|\hat{I}_3 - I| < 10^{-3}\right) = \mathbb{P}\left(\left|\frac{\hat{I}_3 - I}{\sqrt{\frac{3.368662}{N}}}\right| < \frac{10^{-3}}{\sqrt{\frac{0.1946727}{N}}}\right)$$

令 $\frac{10^{-3}}{\sqrt{\frac{0.1946727}{N}}} = z_{0.975}$, 则 $N = 0.1946727 \times 10^6 z_{0.975}^2$

```
N3 <- (integrate(function(x){exp(2*x)/(3/2+x)*3},-1,1)$value - I^2)*10^6*qnorm(0.975)^2
cat('Required number of tests by Importance Sampling Method is : ',N3)
```

```
## Required number of tests by Importance Sampling Method is :  747827.1
```

分层抽样法

$$\mathbb{P}\left(|\hat{I}_4 - I| < 10^{-3}\right) = \mathbb{P}\left(\left|\frac{\hat{I}_4 - I}{\sqrt{Var\hat{I}_{31} + Var\hat{I}_{32}}}\right| < \frac{10^{-3}}{\sqrt{Var\hat{I}_{31} + Var\hat{I}_{32}}}\right)$$

令 $\frac{10^{-3}}{\sqrt{VarI_{31} + VarI_{32}}} = z_{0.975}$, 因为

$$\begin{aligned}
Var\hat{I}_{31} &= \frac{S_{31}^2}{\frac{N}{2}} \\
Var\hat{I}_{32} &= \frac{S_{32}^2}{\frac{N}{2}}
\end{aligned}$$

则 $N = 2(S_{31}^2 + S_{32}^2)10^6 z_{0.975}^2$

```r
N4 <- ceiling(2*VarI4*N*10^6*qnorm(0.975)^2)
cat('Required number of tests by Stratified Sampling Method is : ',N4)
```

```
## Required number of tests by Stratified Sampling Method is :  185747
```

(3) 用不同的随机数种子重复以上的估计 B 次，得到 $\hat{I}_j$, $j=1,2,\ldots,B$, 由此估计 $\hat{I}$ 的抽样分布方差，与 **(2)** 的结果进行验证。

(4) 称

$$MAE(\hat{I}) = E|\hat{I} - I|$$

为 $\hat{I}$ 的平均绝对误差。从 **(3)** 得到的 $\hat{I}_j$, $j=1,2,\ldots,B$ 中估计 $MAE(\hat{I})$。比较这四种积分方法的平均绝对误差大小。

```r
B <- 15
EI <- rep(0,4)
VarI <- rep(0,4)
MAE <- rep(0,4)
N <- 10000
for (i in c(1:B)){
        set.seed(i)
    SPMresult <- SPM(h,-1,1,exp(1),N)
    EI[1] <- EI[1] + SPMresult[[1]]
    VarI[1] <- VarI[1] + SPMresult[[2]]
    MAE[1] <- MAE[1] + abs(SPMresult[[1]]-I)
    MVMresult <- MVM(h,-1,1,N)
    EI[2] <- EI[2] + MVMresult[[1]]
    VarI[2] <- VarI[2] + MVMresult[[2]]
    MAE[2] <- MAE[2] + abs(MVMresult[[1]]-I)
    ISMresult <- ISM(h,g,G_inv,N)
    EI[3] <- EI[3] + ISMresult[[1]]
    VarI[3] <- VarI[3] + ISMresult[[2]]
    MAE[3] <- MAE[3] + abs(ISMresult[[1]]-I)
    SSMresult <- SSM(h,-1,1,10,N)
    EI[4] <- EI[4] + ISMresult[[1]]
    VarI[4] <- VarI[4] + SSMresult[[2]]
    MAE[4] <- MAE[4] + abs(SSMresult[[1]]-I)
}
cat('Estimated integration of I is : ',EI/B)
```

```
## Estimated integration of I is :  2.356678 2.344971 2.349727 2.349727
```

```r
cat('Estimated variance of I is : ',VarI/B)
```

```
## Estimated variance of I is :  0.0007257867 0.0001724218 1.95034e-05 2.411844e-06
```

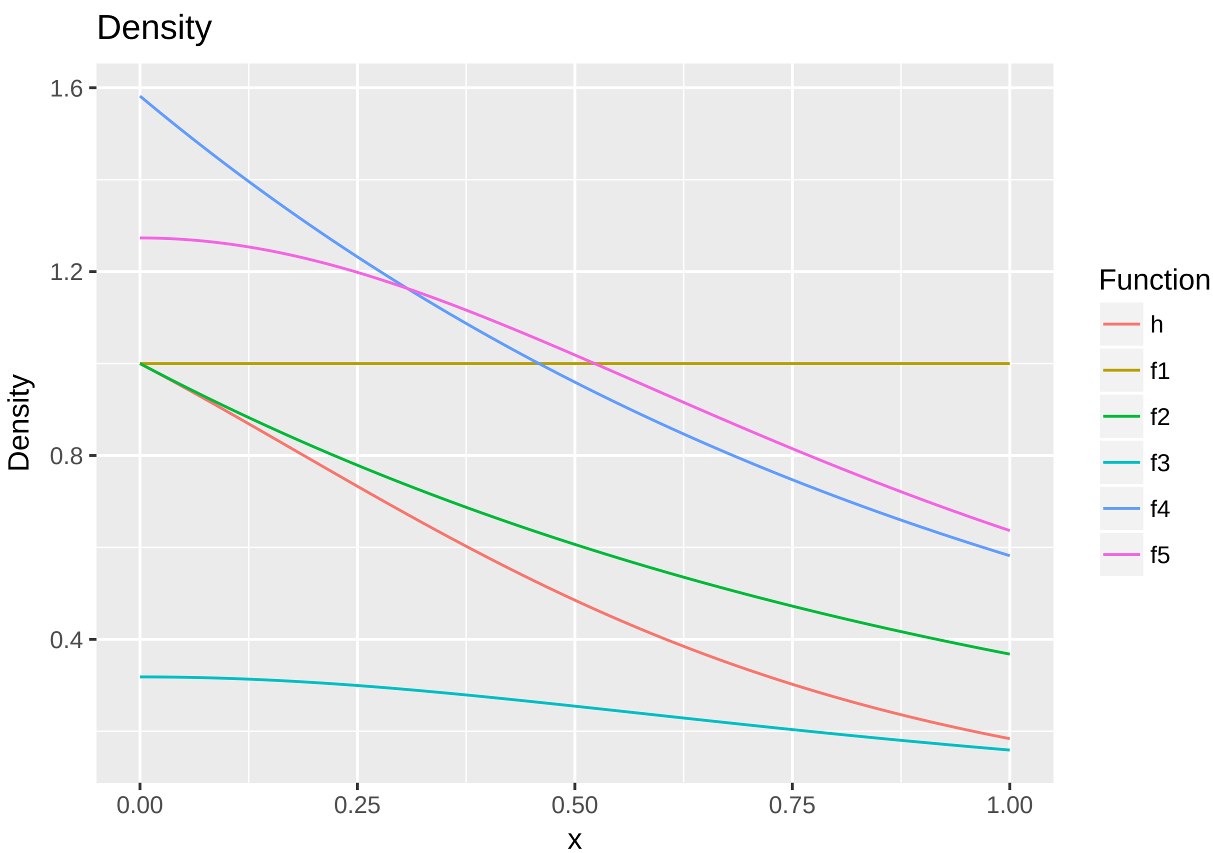
```r
cat('Mean absolute error of I is : ',MAE/B)
```

```
## Mean absolute error of I is :  0.01757928 0.008726125 0.003573075 0.001436255
```

3. 设 $h(x)=\dfrac{e^{-x}}{1+x^2}, x\in(0,1)$**，用重要抽样法计算积分** $I=\int_0^1 h(x)\mathrm{d}x$**，分别采用如下的试抽样密度：**

$$
\begin{aligned}
f_1(x) &= 1, \quad x\in(0,1)\\
f_2(x) &= e^{-x}, \quad x\in(0,\infty)\\
f_3(x) &= \frac{1}{\pi(1+x^2)}, \quad x\in(-\infty,\infty)\\
f_4(x) &= (1-e^{-1})^{-1}e^{-x}, \quad x\in(0,1)\\
f_5(x) &= \frac{4}{\pi(1+x^2)}, \quad x\in(0,1)
\end{aligned}
$$

(1) 作 $h(x)$ **和各试抽样密度的图形，比较其形状。**



$f_2(x)$，$f_4(x)$ 与 $h(x)$ 形状相似。

```r
library(ggplot2)
library(reshape2)
library(latex2exp)

h <- function(x){
        return(exp(-x)/(1+x^2)*(x<=1 & x>=0))
```

```r
}
f1 <- function(x){
        return(as.integer(x<=1 & x>=0))
}
f2 <- function(x){
        return(exp(-x)*(x>=0))
}
f3 <- function(x){
        return(1/(1+x^2)/pi)
}
f4 <- function(x){
        return(exp(-x)*(x<=1 & x>=0)/(1-exp(-1)))
}
f5 <- function(x){
        return(4/(1+x^2)/pi*(x<=1 & x>=0))
}
x <- seq(0,1,length.out = 1000)
result <- data.frame(h=h(x),f1 = f1(x), f2 = f2(x),f3 = f3(x), f4 = f4(x),f5 = f5(x), x=x)
ggplot(data = melt(result,id = 'x',variable.name="Function") ,
                aes(x=x, y=value, colour=Function)) +
    geom_line()+
    labs(title="Density", x = 'x', y = 'Density')
```

(2) 取样本点个数 $N = 10000$，分别给出对应于不同试抽样密度的估计 $\hat{I}_k$，$k = 1,2,3,4,5$，以及 $Var(\hat{I}_k)$ 的估计。

$$
\begin{aligned}
F_1(x) &= x\mathbb{1}_{[0,1]} & F_1^{-1}(y) &= y\mathbb{1}_{[0,1]} \\
F_2(x) &= (1-e^{-x})\mathbb{1}_{[0,1]} & F_2^{-1}(y) &= -\ln(1-y)\mathbb{1}_{[0,1-e^{-1}]} \\
F_3(x) &= \left(\frac{\arctan x}{\pi}+\frac{1}{2}\right)\mathbb{1}_{[0,1]} & F_3^{-1}(y) &= \tan\left[\left(y-\frac{1}{2}\right)\pi\right]\mathbb{1}_{[\frac{1}{2},\frac{3}{4}]} \\
F_4(x) &= \frac{1-e^{-x}}{1-e^{-1}}\mathbb{1}_{[0,1]} & F_4^{-1}(y) &= -\ln[1-(1-e^{-1})y]\mathbb{1}_{[0,1]} \\
F_5(x) &= \frac{4\arctan x}{\pi}\mathbb{1}_{[0,1]} & F_5^{-1}(y) &= \tan\left(\frac{y\pi}{4}\right)\mathbb{1}_{[0,1]}
\end{aligned}
$$

n	1	2	3	4	5
$\hat{I}_n$	0.5251946	0.5245498	0.526542	0.5249511	0.5222991
$Var\hat{I}_n$	6.076767×10^{-6}	9.3588286×10^{-7}	1.9685198×10^{-6}	9.2876292×10^{-7}	1.9879219×10^{-6}

```
N <- 10000
F1_inv <- function(y){
        return(y)
}
F2_inv <- function(y){
        return(-log(1-y)*(y>=0 & y<=1-exp(-1)))
}
F3_inv <- function(y){
        return(tan((y-1/2)*pi)*(y>=1/2 & y<=3/4))
}
F4_inv <- function(y){
        return(-log(1-(1-exp(-1))*y))
}
F5_inv <- function(y){
        return(tan(y*pi/4))
}
set.seed(0)
ISMresult1 <- ISM(h,f1,F1_inv,N)
ISMresult2 <- ISM(h,f2,F2_inv,N,0,1-exp(-1))
ISMresult3 <- ISM(h,f3,F3_inv,N,1/2,3/4)
ISMresult4 <- ISM(h,f4,F4_inv,N)
ISMresult5 <- ISM(h,f5,F5_inv,N)
cat(ISMresult1[[1]],ISMresult2[[1]],ISMresult3[[1]],ISMresult4[[1]],ISMresult5[[1]],'\n')
```

```
## 0.5251946 0.5245498 0.526542 0.5249511 0.5222991
```

```
cat(ISMresult1[[2]],ISMresult2[[2]],ISMresult3[[2]],ISMresult4[[2]],ISMresult5[[2]],'\n')
```

```
## 6.076767e-06 9.358829e-07 1.96852e-06 9.287629e-07 1.987922e-06
```

```
cat(integrate(h,0,1)$value)
```

```
## 0.5247971
```

(3) 分析 $Var(\hat{I}_k)$ 的大小差别的原因。

$\hat{I}_1$，$\hat{I}_3$ 和 $\hat{I}_3$ 的方差较大，原因是其密度函数与 $h(x)$ 的形态差异较大。事实上，

$$\begin{aligned}
Var\left(\frac{h(X)}{g(X)}\right) &= \mathbb{E}\left(\frac{h^2(X)}{g^2(X)}\right) - \left[\mathbb{E}\left(\frac{h(X)}{g(X)}\right)\right]^2 \\
&= \mathbb{E}\left(\frac{h^2(X)}{g^2(X)}\right) - \left[\int_C h(x)\mathrm{d}x\right]^2 \\
&\geq \left[\mathbb{E}\left(\frac{|h(X)|}{g(X)}\right)\right]^2 - \left[\int_C h(x)\mathrm{d}x\right]^2 \\
&= \left[\int_C |h(x)|\mathrm{d}x\right]^2 - \left[\int_C h(x)\mathrm{d}x\right]^2
\end{aligned}$$

当且仅当 $\dfrac{|h|(X)}{g(X)}$ 为常数时等号成立，因此两者形状越接近方差越小。

(4) 把 $(0,1)$ 区间均分为 10 段，在每一段内取 $N=1000$ 个样本点用平均值法计算积分值，把各段的估计求和得到 I 的估计 $\hat{I}_6$，估计其方差。

```r
set.seed(0)
N <- 1000
SSMresult <- SSM(h,0,1,10,N*10)
I6 <- SSMresult[[1]]
VarI6 <- SSMresult[[2]]
I6
```

```
## [1] 0.5247939
```

```r
VarI6
```

```
## [1] 6.164339e-08
```

(5) 用例 3.2.7 的分层抽样方法计算积分的估计 $\hat{I}_7$ ，估计 $Var(\hat{I}_7)$ 并与前面的结果进行比较。

例 3.2.7 中分层抽样法在每一层只取一个抽样点。为了与前面实验统一样本数，取层数 $m=10$，实验次数 $N\times 10$。结果比前面均精确。

```r
set.seed(0)
# Stratified Sampling Method
SSM2 <- function(
```

```r
    h,        # density h(x) to be integrated
    from,     # left end point of x
    to,       # right end point of x
    m,        # level
    N         # the number of points to be generated
){
    samples <- t(matrix(h(from+(to-from)*(c(1:m)-1+ runif(m*N,0,1))/m),m,N))
    I_samples <- apply(samples, 1, mean)
    I_hat <- mean(I_samples)
    VarI_hat <- var(I_samples)/N
    return(list(I_hat,VarI_hat))
}
SSM2result <- SSM2(h,0,1,10,N*10)
I7 <- SSM2result[[1]]
VarI7 <- SSM2result[[2]]
cat('I7    = ',I7,'\n')
```

```
## I7    =  0.5248379
```

```r
cat('VarI7 = ',VarI7,'\n')
```

```
## VarI7 =  5.962721e-09
```

9. 用随机模拟法计算二重积分 $\int_0^1 \int_0^1 e^{(x+y)^2} \mathrm{d}y\mathrm{d}x$, 用对立变量法改善精度。

```
# Adaptive Multivariate Integration
library(cubature)
I <- adaptIntegrate(function(X){return(exp((X[1]+X[2])^2))},c(0,0),c(1,1),tol=1e-5)$integral

# Mean value method
set.seed(0)
N <- 10000000
U1 <- runif(N)
U2 <- runif(N)
h <- function(U1,U2){
    return(exp((U1+U2)^2))
}
In <- h(U1,U2)*(1-0)^2
In_dual <- h(1-U1,1-U2)*(1-0)^2
I_hat <- mean(In)
Var_I_hat <- mean((In-I_hat)^2)/N
I_hat_dual <- mean(In+In_dual)/2
Var_I_hat_dual <- (mean((In-I_hat_dual)^2)+cov(In,In_dual))/2/N
cat('I                     = ',I,'\n')
```

```
## I                     =  4.899159
```

```
cat('Mean Value Method     = ',I_hat,'\n')
```

```
## Mean Value Method     =  4.894959
```

```
cat('Variance of MVM       = ',Var_I_hat,'\n')
```

```
## Variance of MVM       =  3.547827e-06
```

```
cat('Dual Variable Method  = ',I_hat_dual,'\n')
```

```
## Dual Variable Method  =  4.898595
```

```
cat('Variance of DVM       = ',Var_I_hat_dual,'\n')
```

```
## Variance of DVM       =  1.140933e-06
```

16. 不同的检验法有不同的功效，在难以得到功效函数的显式表达式的时候，模拟方法可以起到重要补充作用。对无截距项的回归模型

$$y_i = bx_i + \epsilon_i, \qquad i = 1, 2, \ldots, n$$

$\epsilon_1, \epsilon_2, \ldots, \epsilon_n \overset{iid}{\sim} N(0, \sigma^2)$**。为检验** $H_0 : b = 0$**，有如下两种检验方法：**

(1) $b = 0$ **时** $y_i = \epsilon_i$ **，于是在** H_0 **下**

$$t = \frac{\overline{y}}{\sqrt{\frac{1}{n}\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

服从 $t(n-1)$ **分布。设** λ **为** $t(n-1)$ **分布的** $1 - \frac{\alpha}{2}$ **分位数，取否定域为** $\{|t| > \lambda\}$**。**

(2) 令

$$\hat{b} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2} \qquad U = \hat{b}^2 \sum_{i=1}^{n} x_i^2$$

$$Q = \sum_{i=1}^{n} y_i^2 - U \qquad F = \frac{U}{\frac{Q}{n-1}}$$

设 λ' **为** $F(1, n-1)$ **的** $1 - \alpha$ **分位数，取否定域为** $\{F > \lambda'\}$**。**

对不同的 b, σ^2, n, α **以及不同的** $\{x_i\}$ **模拟比较这两种检验方法的功效。**

$$Power = \mathbb{P}\{\text{Reject } H_0 | H_1 \text{ is true}\}$$

当 b 接近 0 时，$Power$ 接近 0。当 b 远大于 0 时，$Power$ 接近 1。

当 b 较接近 0 并且 σ 较小时，F 检验的功效明显比 t 检验大，即犯第二类错误的概率更小。在其他情况下，两者较接近。

当 σ 较大并且 n 较小时，两种检验的功效都较小。

```
t_test <- function(
    b,
    sigma,
    n,
    x,
    alpha,
        N = 10000
){
    Power <- 0
        for (i in c(1:N)){
        epsilon <- rnorm(n,0,sigma)
```

```
        y <- b*x+epsilon
        y_mean <- mean(y)
        tvalue <- y_mean/sqrt(mean((y-y_mean)^2)/(n-1))
        Power <- Power + (abs(tvalue)>qt(1-alpha/2,n-1))
    }
        return(Power/N)
}


F_test <- function(
    b,
    sigma,
    n,
    x,
    alpha,
    N = 10000
){
    Power <- 0
    for (i in c(1:N)){
        epsilon <- rnorm(n,0,sigma)
        y <- b*x+epsilon
        b_hat <- crossprod(x,y)/crossprod(x)
        U <- b_hat^2*crossprod(x)
        Q <- crossprod(y)-U
        Fvalue <- U/Q*(n-1)
        Power <- Power + (Fvalue>qf(1-alpha,1,n-1))
    }
    return(Power/N)
}

b_list <- c(0.01,1)
sigma_list <- c(0.5,1,5)
n_list <- c(100,1000)
alpha_list <- c(0.01,0.05,0.1)
cat('For x = 1:n')
```

```
## For x = 1:n
```

```
for (b in c(1:2)){
    for (sigma in c(1:3)){
        for (n in c(1:2)){
            for (alpha in c(1:3)){
                x <- c(1:n_list[n])
                print(sprintf('b=%3.2f, sigma=%4.1f, n=%4d, alpha=%.2f, t Power=%f, F Power=%f',
                              b_list[b],sigma_list[sigma],
                              n_list[n],alpha_list[alpha],
                              t_test(b_list[b],sigma_list[sigma],n_list[n],
                                     x,alpha_list[alpha]),
                              F_test(b_list[b],sigma_list[sigma],n_list[n],
                                     x,alpha_list[alpha])))
            }
        }
    }
}
```

```
## [1] "b=0.01, sigma= 0.5, n= 100, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 0.5, n= 100, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 0.5, n= 100, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 0.5, n=1000, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 0.5, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 0.5, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 1.0, n= 100, alpha=0.01, t Power=0.987000, F Power=0.998700"
## [1] "b=0.01, sigma= 1.0, n= 100, alpha=0.05, t Power=0.998900, F Power=0.999800"
## [1] "b=0.01, sigma= 1.0, n= 100, alpha=0.10, t Power=0.999500, F Power=1.000000"
## [1] "b=0.01, sigma= 1.0, n=1000, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 1.0, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 1.0, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 5.0, n= 100, alpha=0.01, t Power=0.061400, F Power=0.076000"
## [1] "b=0.01, sigma= 5.0, n= 100, alpha=0.05, t Power=0.177200, F Power=0.211700"
## [1] "b=0.01, sigma= 5.0, n= 100, alpha=0.10, t Power=0.262700, F Power=0.312900"
## [1] "b=0.01, sigma= 5.0, n=1000, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 5.0, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=0.01, sigma= 5.0, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n= 100, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n= 100, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n= 100, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n=1000, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
```

```
## [1] "b=1.00, sigma= 0.5, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n= 100, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n= 100, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n= 100, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n=1000, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n= 100, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n= 100, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n= 100, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n=1000, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
```

```r
cat('For x = norm')
```

```
## For x = norm
```

```r
for (b in c(1:2)){
    for (sigma in c(1:3)){
        for (n in c(1:2)){
            for (alpha in c(1:3)){
                x <- rnorm(n_list[n],1,3)
                print(sprintf('b=%3.2f, sigma=%4.1f, n=%4d, alpha=%.2f, t Power=%f, F Power=%f',
                              b_list[b],sigma_list[sigma],
                              n_list[n],alpha_list[alpha],
                              t_test(b_list[b],sigma_list[sigma],n_list[n],
                                     x,alpha_list[alpha]),
                              F_test(b_list[b],sigma_list[sigma],n_list[n],
                                     x,alpha_list[alpha])))
            }
        }
    }
}
```

```
## [1] "b=0.01, sigma= 0.5, n= 100, alpha=0.01, t Power=0.009900, F Power=0.028400"
## [1] "b=0.01, sigma= 0.5, n= 100, alpha=0.05, t Power=0.055000, F Power=0.097500"
## [1] "b=0.01, sigma= 0.5, n= 100, alpha=0.10, t Power=0.112000, F Power=0.158200"
## [1] "b=0.01, sigma= 0.5, n=1000, alpha=0.01, t Power=0.021500, F Power=0.263700"
## [1] "b=0.01, sigma= 0.5, n=1000, alpha=0.05, t Power=0.098100, F Power=0.498200"
## [1] "b=0.01, sigma= 0.5, n=1000, alpha=0.10, t Power=0.169400, F Power=0.625900"
```

```
## [1] "b=0.01, sigma= 1.0, n= 100, alpha=0.01, t Power=0.010500, F Power=0.014400"
## [1] "b=0.01, sigma= 1.0, n= 100, alpha=0.05, t Power=0.053800, F Power=0.064900"
## [1] "b=0.01, sigma= 1.0, n= 100, alpha=0.10, t Power=0.103700, F Power=0.113800"
## [1] "b=0.01, sigma= 1.0, n=1000, alpha=0.01, t Power=0.014500, F Power=0.056500"
## [1] "b=0.01, sigma= 1.0, n=1000, alpha=0.05, t Power=0.061900, F Power=0.168000"
## [1] "b=0.01, sigma= 1.0, n=1000, alpha=0.10, t Power=0.116400, F Power=0.267500"
## [1] "b=0.01, sigma= 5.0, n= 100, alpha=0.01, t Power=0.008900, F Power=0.008600"
## [1] "b=0.01, sigma= 5.0, n= 100, alpha=0.05, t Power=0.050000, F Power=0.052000"
## [1] "b=0.01, sigma= 5.0, n= 100, alpha=0.10, t Power=0.097200, F Power=0.098800"
## [1] "b=0.01, sigma= 5.0, n=1000, alpha=0.01, t Power=0.011200, F Power=0.013300"
## [1] "b=0.01, sigma= 5.0, n=1000, alpha=0.05, t Power=0.050600, F Power=0.050700"
## [1] "b=0.01, sigma= 5.0, n=1000, alpha=0.10, t Power=0.100800, F Power=0.102100"
## [1] "b=1.00, sigma= 0.5, n= 100, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n= 100, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n= 100, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n=1000, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 0.5, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n= 100, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n= 100, alpha=0.05, t Power=0.802100, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n= 100, alpha=0.10, t Power=0.999200, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n=1000, alpha=0.01, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 1.0, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n= 100, alpha=0.01, t Power=0.131600, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n= 100, alpha=0.05, t Power=0.138500, F Power=0.999800"
## [1] "b=1.00, sigma= 5.0, n= 100, alpha=0.10, t Power=0.519500, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n=1000, alpha=0.01, t Power=0.999700, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n=1000, alpha=0.05, t Power=1.000000, F Power=1.000000"
## [1] "b=1.00, sigma= 5.0, n=1000, alpha=0.10, t Power=1.000000, F Power=1.000000"
```

26. 设随机变量 X 和 Y 都取值于 $(0,B)$ 区间 (B 已知)。设 $Y=y$ 条件下 X 的条件分布密度为

$$f(x|y) \propto e^{-yx}, \quad x \in (0,B)$$

$X=x$ 条件下 Y 的条件分布密度为

$$f(y|x) \propto e^{-xy}, \quad y \in (0,B)$$

编写 R 程序用 Gibbs 抽样方法对 (X,Y) 抽样，估计 $\mathbb{E}X$ 和 $\rho(X,Y)$。

$\because$

$$\begin{aligned} f(x|y) &\propto e^{-yx}\mathbb{1}_{[0,B]}(x) \\ F(x|y) &\propto \left(\frac{1}{y}-\frac{1}{y}e^{-yx}\right)\mathbb{1}_{[0,B]}(x) \end{aligned}$$

$\therefore$

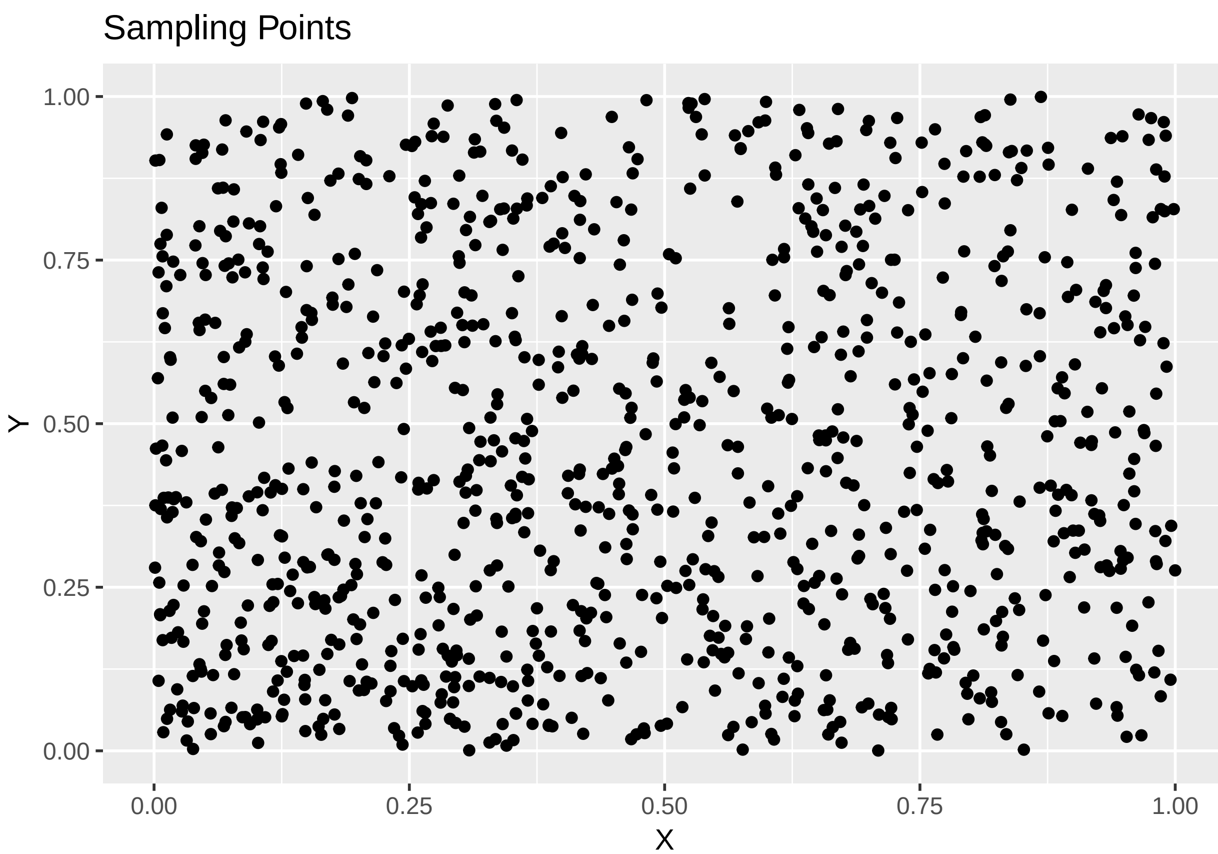
$$\begin{aligned} f(x|y) &= \frac{ye^{-yx}}{1-e^{-yB}}\mathbb{1}_{[0,B]} \\ F(x|y) &= \frac{1-e^{-yx}}{1-e^{-yB}}\mathbb{1}_{[0,B]} \\ F_{x|y}^{-1}(z|y) &= -\frac{1}{y}\ln[1-(1-e^{-yB})z] \end{aligned}$$

```
F_inv <- function(X,
                  B
){
    U <- runif(2)
    X_next <- rev(-log(1-(1-exp(-X*B))*U)/X)
        return(X_next)
}

GibbsSampling <- function(
    F_inv,
    B,
    N
){
    X <- matrix(0,N,2)
    X[1,] <- runif(2,0,B)
        for (i in c(2:N)){
        X[i,] <- F_inv(X[i-1,],B)
    }
        return(X)
}
```

```
set.seed(0)
B <- 1
N <- 1000
samples <- GibbsSampling(F_inv,B,N)
X <- samples[,1]
Y <- samples[,2]


ggplot(data = NULL, aes(x = X, y = Y)) +
  # 散点图函数
  geom_point() +
  coord_cartesian(xlim=c(0,B),ylim=c(0,B))  +
  labs(title="Sampling Points")
```




```
EX <- mean(X)
rho <- mean((X-mean(X))*(Y-mean(Y)))/sqrt(mean((X-mean(X))^2)*mean((Y-mean(Y))^2))
cat('EX        = ',EX,'\n')
```

```
## EX        =  0.4612906
```

```
cat('rho(X,Y) = ',rho,'\n')
```

```
## rho(X,Y) =  0.06515843
```