计算物理作业九

于浩然 PB19020634 2021.10.20

1 作业题目

考虑泊松分布、指数分布，并再自设若干个随机分布 (它们有相同或不同的 μ 和 σ^2)，通过 Monte Carlo 模拟，验证中心极限定理成立 ($N=2,5,10$).

2 算法简介

2.1 中心极限定理

概率论中大数法则和中心极限定理是 Monte Carlo 方法应用于统计计算的基础. 大数法则说，若随机量序列 f_i 有期望值 μ，则

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}f_i\to\mu \tag{1}$$

而中心极限定理指出了当 N 有限时，平均值的误差分布. 即

$$P\left\{\left|\frac{\langle f\rangle-\mu}{\sigma_f/\sqrt{N}}\right|<\beta\right\}\to\Phi(\beta) \tag{2}$$

其中 $\Phi(\beta)$ 是 Gauss 正态分布. 因此

$$\sigma_s=|\langle f\rangle-\mu|\propto\frac{\sigma_f}{\sqrt{N}}=\frac{1}{\sqrt{N}}\sqrt{\langle f^2\rangle-\langle f\rangle^2} \tag{3}$$

上式中 σ_f 是函数 $f(x)$ 的标准差，σ_s 是积分值的标准差.

若要验证此定理，可以对满足任一分布的随机变量 x 进行 N 次抽样，计算统计量 $\frac{\langle x\rangle-\mu}{\sigma/\sqrt{N}}$，重复上述过程 n 次取平均，可得到对应分布抽样 N 次所得统计量的近似概率分布. 将该概率分布与标准正态分布比较，若随着 n 的增大两者越来越接近，则可认为中心极限定理成立.

2.2 标准正态分布

期望为 0，方差为 1 的正态分布称为 **标准正态分布**. 其概率密度分布函数为

$$f(x)=\frac{1}{\sqrt{2\pi}}e^{-x^2/2} \tag{4}$$

其累积函数为

$$\Phi(x)=\int_{-\infty}^{x}f(t)\mathrm{d}t \tag{5}$$

2.3 分布一：泊松分布

若 X 服从参数为 λ 的泊松分布，记为 $X \sim P(\lambda)$. 其概率密度分布函数 (离散分布) 为

$$P(x=k)=\frac{\lambda^k}{k!}e^{-\lambda}, k=0,1,\cdots \tag{6}$$

其期望值和标准差值可由参数 λ 确定：

$$\mu=\sigma^2=\lambda \tag{7}$$

不妨取参数 $\lambda=2$，并构造如下统计量：

$$g=\frac{\bar{X}-\mu}{\sigma/\sqrt{N}}=\sqrt{\frac{N}{2}}(\bar{X}-2) \tag{8}$$

此分布中随机变量 X 取值为可数无穷多个，但实际上当 X 较大时取值概率已相当小，故我们忽略后面的部分进行抽样.

$\lambda=2$ 时 $P(X=k)=\frac{2^k}{k!}e^{-2}$，这时 $P(X=7)=0.0034$，$P(X=8)=0.00086$ ，$P(X=9)=0.00019$，我们可以认为当 $k>8$ 时，$P(X=k)\approx 0$. 抽样时，只需对 $X=k(k=1,2,\cdots,8)$ 进行抽样. 但由于后面仍有无穷多项，仍需重新归一化.

2.4 分布二：指数分布

若 X 服从参数为 λ 的指数分布，记为 $X \sim E(\lambda)$. 其概率密度分布函数 (连续分布) 为

$$f(x)=\lambda e^{-\lambda x}, \quad x\geq 0 \tag{9}$$

同样，我们可确定期望与方差：

$$\langle x\rangle=\frac{1}{\lambda}, \quad \sigma^2=\frac{1}{\lambda^2} \tag{10}$$

取参数 $\lambda=1$，可构造统计量：

$$g=\frac{\bar{x}-1/\lambda}{1/\lambda\sqrt{N}}=\sqrt{N}(\bar{x}-1) \tag{11}$$

与 Poisson 分布同理，数值计算模拟时必须选取一段有限区间. 绘制参数 $\lambda=1$ 的指数分布的概率密度函数图像展示如下：

显见，可忽略 $x>6$ 的部分，抽样时仅对 $0\leq x\leq 6$ 进行即可.

考虑直接抽样，对 (9) 式积分求出指数分布的累积函数：

$$\xi(x)=\int_0^x e^{-t}\mathrm{d}t=1-e^{-x} \tag{12}$$

求反函数：

$$x=-\ln(1-\xi)=-\ln(\xi) \tag{13}$$

其中 ξ 为均匀分布随机变量，ξ 与 $1-\xi$ 等价. 在编写抽样程序时，可由上式从均匀随机数 ξ 抽出服从指数分布的 x.

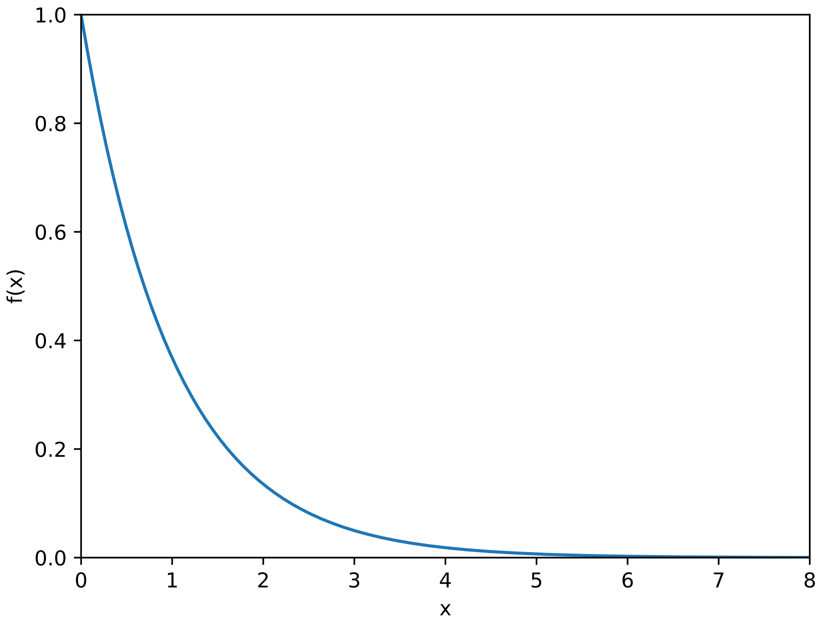


图 1: $\lambda = 1$ 的指数分布概率密度函数 $f(x)$

2.5 分布三：自设离散分布

设分布函数为

$$P(x=k) = 1/5, \quad k = 1,2,3,4,5 \tag{14}$$

容易计算其期望和方差：

$$\langle X \rangle = 3, \quad \sigma^2 = \langle X^2 \rangle - \langle X \rangle^2 = 2 \tag{15}$$

可构造统计量：

$$g = \frac{\bar{X} - 3}{\sqrt{2/N}} \tag{16}$$

2.6 分布四：自设连续分布

设分布函数为

$$f(x) = x, \quad 0 \le x \le 1 \tag{17}$$

易得期望和方差

$$\langle x \rangle = \frac{1}{3}, \quad \sigma^2 = \langle x^2 \rangle - \langle x \rangle^2 = \frac{5}{36} \tag{18}$$

可构造统计量：

$$\frac{\bar{x} - 1/3}{\sqrt{5/36}/\sqrt{N}} = 2\sqrt{\frac{N}{5}}(3\bar{x} - 1) \tag{19}$$

同样考虑进行直接抽样，累积函数为

$$\xi(x) = \int_0^x f(t)\mathrm{d}t = \frac{x^2}{2} \tag{20}$$

求反函数:

$$x = \sqrt{2\xi} \tag{21}$$

于是可由上式抽样.

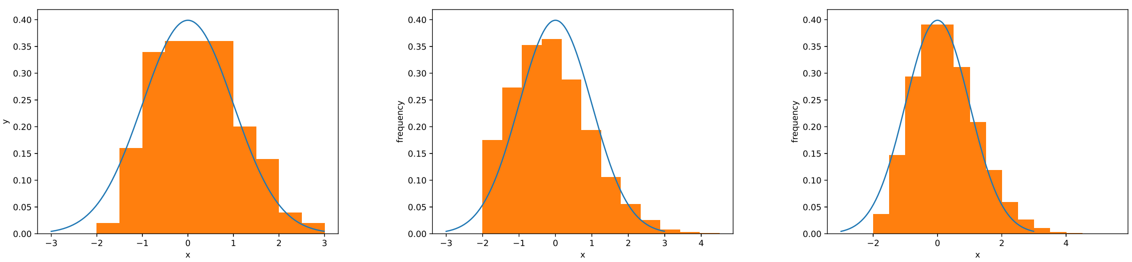
3 编程实现

用 FORTRAN90 进行编程，分别将离散抽样的两个子程序和连续 (直接抽样) 的两个子程序装在两个模块中，便于管理与维护. 源代码较长，展示于报告最后.

4 计算结果

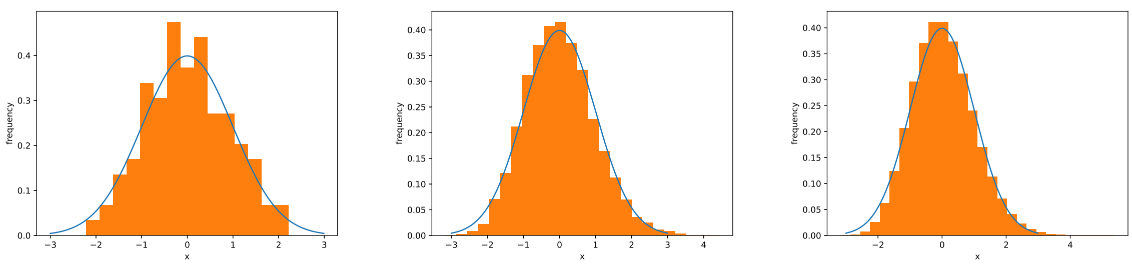
适当选取分割区间，用各组数据绘制直方图.

4.1 泊松分布抽样结果



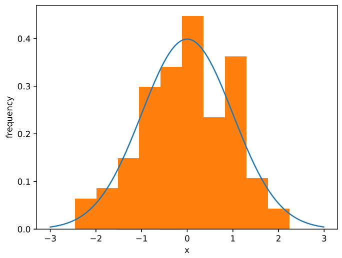
(a) $n = 10^2$ (b) $n = 10^4$ (c) $n = 10^6$

图 2: 泊松分布抽样结果 (N=2)

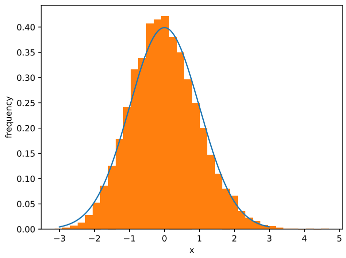


(a) $n = 10^2$ (b) $n = 10^4$ (c) $n = 10^6$

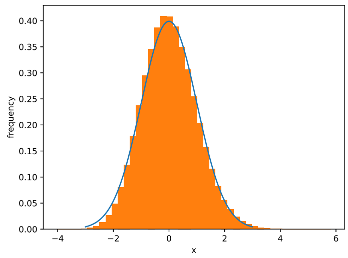
图 3: 泊松分布抽样结果 (N=5)



(a) $n = 10^2$



(b) $n = 10^4$

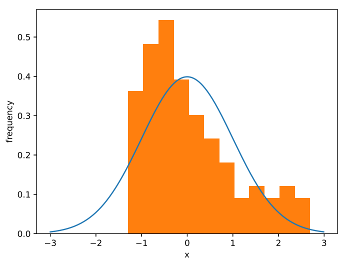


(c) $n = 10^6$

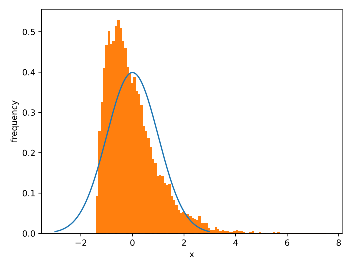
图 4: 泊松分布抽样结果 (N=10)

由于泊松分布是离散抽样，故选取区间只能比较宽才能等到理想的效果. 可见实际峰比理论峰要偏左，这是由于我们舍弃了 k 较大时的情形，是一个“近似”的泊松分布.

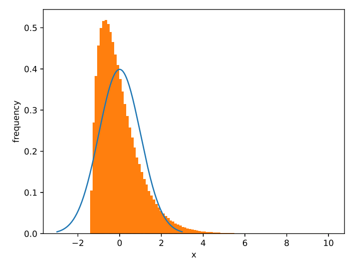
4.2 指数分布抽样结果



(a) $n = 10^2$

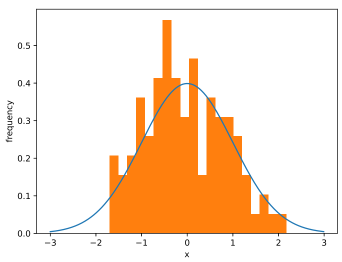


(b) $n = 10^4$

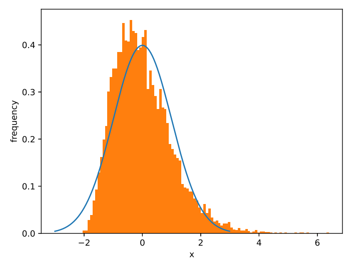


(c) $n = 10^6$

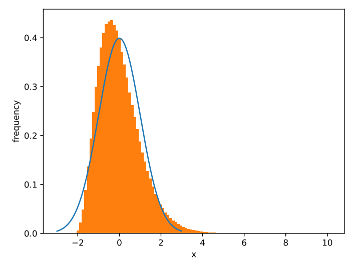
图 5: 指数分布抽样结果 (N=2)



(a) $n = 10^2$

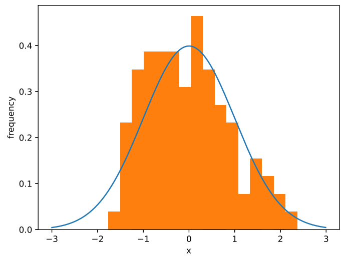


(b) $n = 10^4$

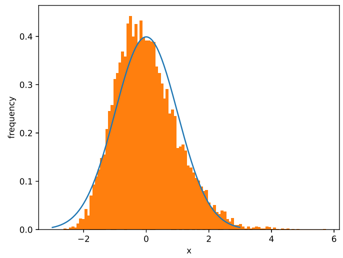


(c) $n = 10^6$

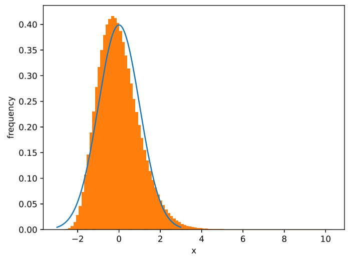
图 6: 指数分布抽样结果 (N=5)



(a) $n = 10^2$



(b) $n = 10^4$

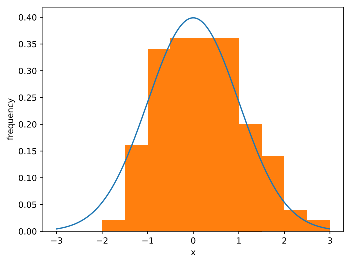


(c) $n = 10^6$

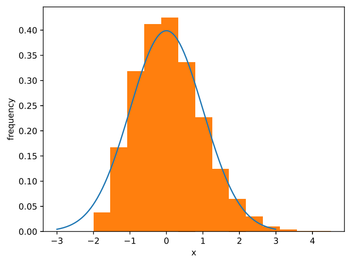
图 7: 指数分布抽样结果 (N=10)

我们容易看出此时频率直方图也明显地趋近与正态曲线，但是峰偏左的现象比前者更为明显，尤其是当 N 取 2 时，这也是我们取部分区间抽样所导致的，并不影响我们证明中心极限定理.

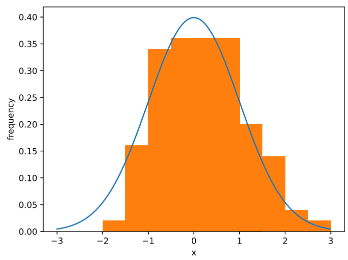
4.3 自设离散分布抽样结果



(a) $n = 10^2$

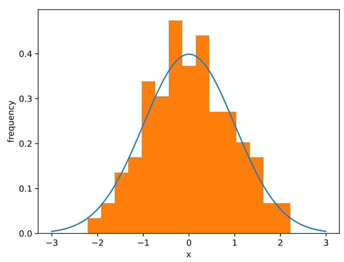


(b) $n = 10^4$

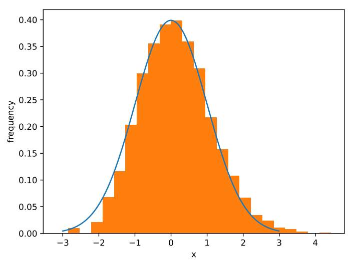


(c) $n = 10^6$

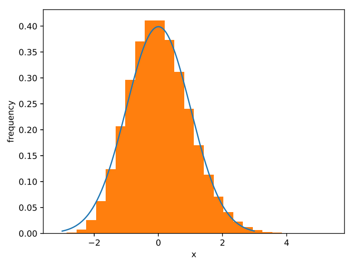
图 8: 自设离散分布抽样结果 (N=2)



(a) $n = 10^2$



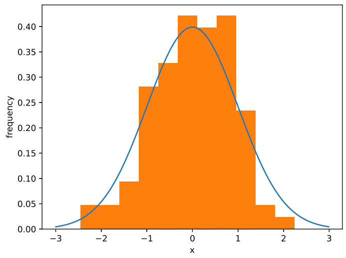
(b) $n = 10^4$



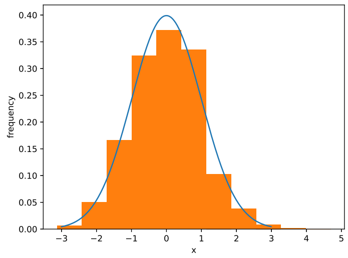
(c) $n = 10^6$

图 9: 自设离散分布抽样结果 (N=5)

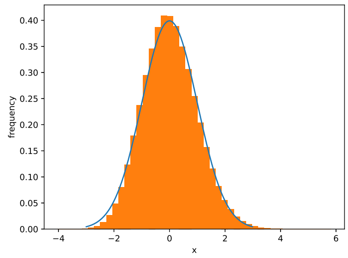
由于所设函数较简单，$N = 2$ 时频率直方图与正态曲线拟合不太好，但是当 $N = 5, 10$ 时仍可明显看出数据的正态分布趋势.



(a) $n = 10^2$



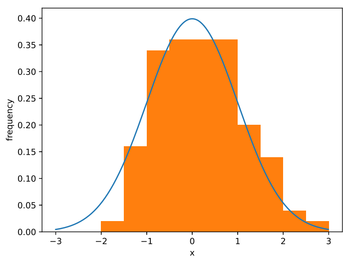
(b) $n = 10^4$



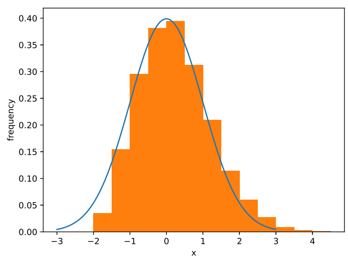
(c) $n = 10^6$

图 10: 自设离散分布抽样结果 (N=10)

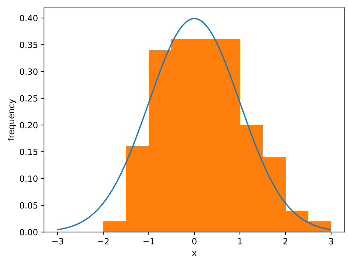
4.4 自设连续分布抽样结果



(a) $n = 10^2$

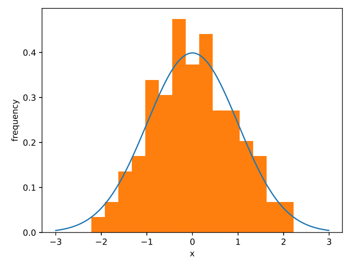


(b) $n = 10^4$

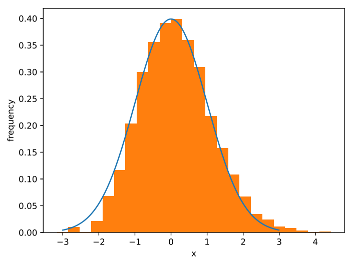


(c) $n = 10^6$

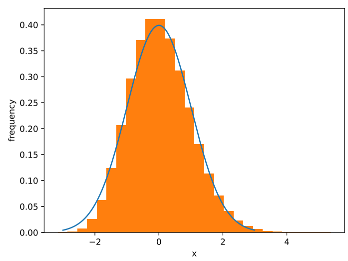
图 11: 自设连续分布抽样结果 (N=2)



(a) $n = 10^2$



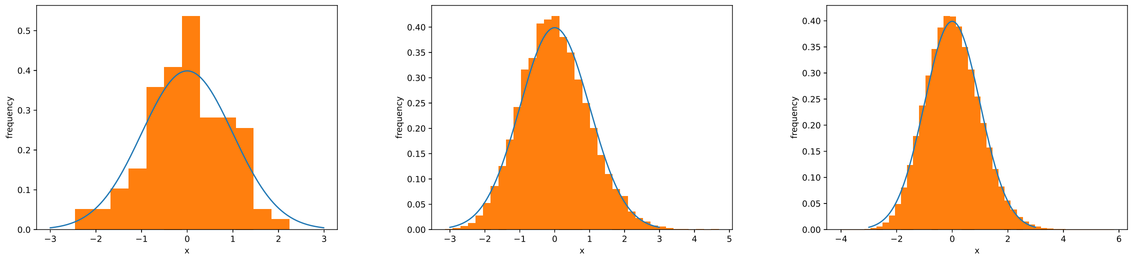
(b) $n = 10^4$



(c) $n = 10^6$

图 12: 自设连续分布抽样结果 (N=5)

与上一节相似，当 N 更大时抽样结果与正态分布的吻合度更高.



(a) $n = 10^2$　(b) $n = 10^4$　(c) $n = 10^6$

图 13: 自设连续分布抽样结果 (N=10)

5 结论

综上，我们验证了泊松分布、指数分布以及自设的离散和连续分布满足中心极限定理.

6 源程序

将 FORTRAN90 源代码展示如下：

```fortran
MODULE DscSample !离散抽样模块
IMPLICIT NONE
CONTAINS

SUBROUTINE Poi(N, num, filename) !泊松分布的抽样以及统计量计算
    INTEGER(KIND=4) :: k, N, i, j, l, num
    REAL(KIND=8), DIMENSION(num * N) :: dat
    REAL(KIND=8), DIMENSION(num) :: g, xbar
    REAL(KIND=8), DIMENSION(num, N) :: x
    REAL(KIND=8), DIMENSION(0:8) :: sumprob
    REAL(KIND=8) :: maximum
    CHARACTER(LEN=*) :: filename
    CALL Schrage(num * N, 98412123, 'rand.dat') !共需要(n*N)个随机数
    OPEN (1, file='rand.dat')
    READ (1, *) dat
    CLOSE (1)
    sumprob(0) = f(0)
    DO i = 1, 8
```

```fortran
            sumprob(i) = sumprob(i - 1) + f(i) !求分段节点
        END DO
        maximum = sumprob(8)
        sumprob = sumprob / maximum !重新归一化
        DO i = 1, num
            DO j = 1, N
                DO l = 0, 8
                    IF(dat((i * N - j + 1)) < sumprob(l)) THEN
                        x(i, j) = l !在指定概率区间内则取相应值
                        EXIT
                    END IF
                END DO
            END DO
        END DO
        xbar = SUM(x, DIM=2) / N !通过数组运算求出每次循环中的平均值
        g = SQRT(real(N) / 2) * (xbar - 2) !构造统计量
        OPEN (1, file=trim(filename))
        WRITE (1, *) g
        CLOSE (1)
        CONTAINS
            FUNCTION f(t) !Poisson分布密度函数表达式
                REAL(KIND=8) :: f
                INTEGER(KIND=4) :: t, y, m
                y = 1 !给阶乘值赋初值1
                IF(t .NE. 0) THEN
                    DO m = 1, t
                        y = y * m !求阶乘
                    END DO
                END IF
                f = (2**t * EXP(-2.0)) / y
            END FUNCTION f
END SUBROUTINE Poi

SUBROUTINE Mydsc(N, num, filename) !自设离散分布的抽样与统计量计
    算
        INTEGER(KIND=4) :: k, N, i, j, l, num
        REAL(KIND=8), DIMENSION(num * N) :: dat
        REAL(KIND=8), DIMENSION(num) :: g, xbar
        REAL(KIND=8), DIMENSION(num, N) :: x
```

```
    REAL(KIND=8), DIMENSION(1:5) :: sumprob
    CHARACTER(LEN=*) :: filename
    CALL Schrage(num * N, 98412123, 'rand.dat') !共需要(n*N)个随机数
    sumprob(1) = 1.0 / 5
    DO i = 2, 5
        sumprob(i) = sumprob(i - 1) + 1.0 / 5
    END DO
    DO i = 1, num
        DO j = 1, N
            DO l = 1, 5
                IF(dat((i * N - j + 1)) < sumprob(l)) THEN
                    x(i, j) = l !在指定概率区间内则取相应值
                    EXIT
                END IF
            END DO
        END DO
    END DO
    xbar = SUM(x, DIM=2) / N !通过数组运算求出每次循环中的平均值
    g = (xbar - 3) * SQRT(real(N) / 2) !计算统计量
    OPEN (1, file=filename)
    READ (1, *) g
    CLOSE (1)
END SUBROUTINE Mydsc
END MODULE DscSample

MODULE Sample !连续抽样模块
IMPLICIT NONE

CONTAINS
SUBROUTINE Expt(N, num, filename) !指数分布抽样与统计量计算
    INTEGER(KIND=4) :: N, num, i, j
    CHARACTER(LEN=*) :: filename
    REAL(KIND=8), DIMENSION(num * N) :: dat
    REAL(KIND=8), DIMENSION(num) :: g, xbar
    REAL(KIND=8), DIMENSION(num, N) :: x
    CALL Schrage(num * N, 7521233, 'rand.dat')
    OPEN (1, file='rand.dat')
    READ (1, *) dat
```

```
    CLOSE (1)
    DO i = 1, num
        DO j = 1, N
            x(i, j) = f(dat(i * N - j + 1)) !对0到1间均匀随机数
                直接抽样
        END DO
    END DO
    xbar = SUM(x, DIM=2) / N !通过数组运算求出每次循环中的平均值
    g = SQRT(real(N)) * (xbar - 1)
    OPEN (1, file=filename)
    WRITE (1, *) g
    CLOSE (1)
    CONTAINS
        FUNCTION f(t) !指数分布抽样函数
            REAL(KIND=8) :: f, t
            f = - LOG(t) !对f进行积分得到x平均值
        END FUNCTION f
END SUBROUTINE Expt

SUBROUTINE Myctn(N, num, filename) !自设连续分布抽样与统计量计算
    INTEGER(KIND=4) :: N, num, i, j
    CHARACTER(LEN=*) :: filename
    REAL(KIND=8), DIMENSION(num * N) :: dat
    REAL(KIND=8), DIMENSION(num) :: g, xbar
    REAL(KIND=8), DIMENSION(num, N) :: x
    CALL Schrage(num * N, 64563218, 'rand.dat')
    OPEN (1, file='rand.dat')
    READ (1, *) dat
    CLOSE (1)
    DO i = 1, num
        DO j = 1, N
            x(i, j) = f(dat(i * N - j + 1)) !对0到1间均匀随机数
                直接抽样
        END DO
    END DO
    xbar = SUM(x, DIM=2) / N !通过数组运算求出每次循环中的平均值
    g = 2 * SQRT(real(N) / 5) * (3 * xbar - 1)
    OPEN (1, file=filename)
    WRITE (1, *) g
```

```fortran
    CLOSE (1)
    CONTAINS
        FUNCTION f(t) !定义抽样函数
            REAL(KIND=8) :: f, t
            f = SQRT(2 * t)
        END FUNCTION f
END SUBROUTINE Myctn
END MODULE Sample

SUBROUTINE Schrage(num, z0, filename) !Schrage随机数生成器子程序
    IMPLICIT NONE
    INTEGER(KIND=4) :: N = 1, num
    INTEGER :: m = 2147483647, a = 16807, q = 127773, r = 2836,
       In(num), z0
    REAL(KIND=8) :: z(num)
    CHARACTER(LEN=8) :: filename
    In(1) = z0 !将传入值z0作为种子
    z(1) = REAL(In(1))/m
    DO N = 1, num - 1
        In(N + 1) = a*MOD(In(N), q) - r*INT(In(N)/q)
        IF (In(N + 1) < 0) THEN !若值小于零, 按Schrage方法加m
            In(N + 1) = In(N + 1) + m
        END IF
        z(N + 1) = REAL(In(N + 1))/m !得到第N+1个随机数
    END DO
    OPEN (1, file=trim(filename)) !每次运行子程序按照传入参数
       filename生成数据文件
    DO N = 1, num !将随机数按行存入文件
        WRITE (1, *) z(N)
    END DO
    CLOSE (1)
END SUBROUTINE Schrage

PROGRAM MAIN
    USE DscSample
    USE Sample
    IMPLICIT NONE
    INTEGER(KIND=4) :: i, j
    CHARACTER(LEN=1) :: chari
```

```fortran
    DO i = 2, 6, 2
        WRITE (chari, "(I1)") i !将整型数值转化为字符型便于写入文件名
        CALL Poi(2, 10**i, 'poi_2_' // chari // '.dat')
        CALL Poi(5, 10**i, 'poi_5_' // chari // '.dat')
        CALL Poi(10, 10**i, 'poi_10_' // chari // '.dat')
        WRITE (chari, "(I1)") i
        CALL Expt(2, 10**i, 'expt_2_' // chari // '.dat')
        CALL Expt(5, 10**i, 'expt_5_' // chari // '.dat')
        CALL Expt(10, 10**i, 'expt_10_' // chari // '.dat')
        WRITE (chari, "(I1)") i
        CALL Poi(2, 10**i, 'mydsc_2_' // chari // '.dat')
        CALL Poi(5, 10**i, 'mydsc_5_' // chari // '.dat')
        CALL Poi(10, 10**i, 'mydsc_10_' // chari // '.dat')
        WRITE (chari, "(I1)") i
        CALL Poi(2, 10**i, 'myctn_2_' // chari // '.dat')
        CALL Poi(5, 10**i, 'myctn_5_' // chari // '.dat')
        CALL Poi(10, 10**i, 'myctn_10_' // chari // '.dat')
    END DO
END PROGRAM MAIN
```

以及使用多次的 python 脚本：

```python
import numpy as np
import matplotlib.pyplot as plt
import math

plt.rcParams['savefig.dpi'] = 300
plt.rcParams['figure.dpi'] = 300

x = np.arange(-3, 3, 0.01)
y = np.exp(-pow(x, 2) / 2) / np.sqrt(2 * np.pi)
data = np.loadtxt('myctn_2_6.dat')
plt.xlabel('x')
plt.ylabel('frequency')
plt.plot(x, y)
plt.hist(data, bins=16, density=True)
```

```
plt.savefig('myctn_2_6.eps')
plt.show()
```