Chapter 4 生成离散随机变量 (Generating Discrete Random Variables)

4.1 逆变换法 Inverse Transform Method

collapsed:: true

- **模型简述：**
  - 不论何种随机变量，定有一个相对应的分布函数，且由分布函数的性质可以确定其在$(0,1)$区间式单调递增的，故具有（广义）反函数。模拟的思路即为通过生成均匀分布随机数$U(0,1)$模拟其分布函数数值F，再通过寻找分布函数的反函数确定其随机变量X的数值。
- **模拟目的：**
  - 生成一系列离散型随机变量X，其概率密度函数服从：
    - $P\{X=x_j\}=p_j$，$j=0,1,\cdots$，$\sum_j p_j=1$
- **模拟方法：**
  - 生成随机数U（服从$U(0,1)$的均匀分布）
  - 令
    - $$X=\begin{cases} x_0 & \text{If } U<p_0 \\ x_1 & \text{If } p_0\le U<p_0+p_1 \\ \vdots & \\ x_j & \text{If } \sum_{i=0}^{j-1}p_i\le U<\sum_{i=0}^{j}p_i \\ \vdots & \end{cases}$$
  - 对于$0<a<b<1, p\{a\le U<b\}=b-a$，有：
    - $$p\{X=x_j\}=p\left\{\sum_{i=0}^{j-1}p_i\le U<\sum_{i=0}^{j}p_i\right\}=p_j$$
  - 此时的X即为所求.
- **注意：**
  - 1. 算法表达：
    - Generate a random number U
      If $U<p_0$ set $X=x_0$ and stop
      If $U<p_0+p_1$ set $X=x_1$ and stop
      If $U<p_0+p_1+p_2$ set $X=x_2$ and stop
  - 2. 若x_i是顺序排列的，即$x_0<x_1<\cdots$，且记$F(x_k)=\sum_{i=0}^{k}p_i$，则：
    - $X=x_j$，if $F(x_{j-1}\le U<F(x_j))$
    - 换言之，该过程即为寻找$F^{-1}(U)$对应的X
- **例4a**
  - 要求：生成随机变量X满足：
    - $p_1=0.20,\quad p_2=0.15,\quad p_3=0.25,\quad p_4=0.40\quad \text{where } p_j=P\{X=j\}$
  - 解法一：
    - Generate U

- If $U < 0.20$ set $X = 1$ and stop
  If $U < 0.35$ set $X = 2$ and stop
  If $U < 0.60$ set $X = 3$ and top
  Otherwise set $X = 4$
- 解法二：
  - Generate U
  - If $U < 0.40$ set $X = 4$ and stop
    If $U < 0.65$ set $X = 3$ and stop
    If $U < 0.85$ set $X = 1$ and stop Otherwise set $X = 2$

- ### 例4d 生成几何分布随机变量
  - 要求：生成随机变量X满足参数为p的几何分布，即：
    - $P\{X = i\} = pq^{i-1}, \quad i \geq 1, \quad \text{ where } q = 1 - p$
  - 解：
    - 由几何分布的含义，X可认为是n次独立实验中首次成功的时间，且每次实验的成功概为p，故有：
      - $$\begin{aligned}\sum_{i=1}^{j-1} P\{X = i\} = 1 - P\{X > j - 1\} \\ = 1 - P\{\text{ first } j - 1 \text{ trials are all failures }\} \\ = 1 - q^{j-1}, \quad j \geq 1\end{aligned}$$
    - 故可以生成随机数U并令：
      - $$1 - q^{j-1} \leq U < 1 - q^j$$
      - $$\Rightarrow q^j < 1 - U \leq q^{j-1}$$
    - 因此X为：
      - $$X = \text{Min}\left\{j : q^j < 1 - U\right\} \quad \cdots (\star)$$
    - 下需解出j的具体数值。由对数函数的单调性，对$\star$式集合中不等式两侧求对数，有：
      - $$\begin{aligned}X = \text{Min}\{j : j \log(q) < \log(1 - U)\} \\ = \text{Min}\left\{j : j > \frac{\log(1 - U)}{\log(q)}\right\}\end{aligned}$$
    - 若用记号$\text{Int}(x)$表示“不大于x的最大整数”，则有：
      - $$X = \text{Int}\left(\frac{\log(1 - U)}{\log(q)}\right) + 1$$
      - 其等价于：
      - $$\boxed{X \equiv \text{Int}\left(\frac{\log(U)}{\log(q)}\right) + 1}$$

4.2 生成泊松分布随机变量 Generating Poisson Random Variables

collapsed:: true

- ## 模型简述：
  - 该算法的思路与4.1一致：通过生成U模拟泊松分布的分布函数，再寻找其对应的自变量值X。不同之处在于泊松分布的函数正常计算较为复杂，故采取递推方式计算。
  - 递推的思路：首先仍生成一均匀分布U代表泊松分布函数，然后开始循环讨论。从$F(i)=P\{X=i\}, i=0$开始，看$F(i)$的值是否大于生成的U的值。若是则该i即为想要模拟的x，若不是则$i++$，继续讨论。
  - 简而言之，对于递推形式，算法的核心在于依次比较$F(0), F(1), F(2), \cdots$与U的大小，第一个使得$F(i)>U$的i即为所求的x。
- ## 模拟目的：
  - 生成服从参数为λ的泊松分布的随机变量X，即
    - $$p_i=P\{X=i\}=e^{-\lambda}\frac{\lambda^i}{i!} \quad i=0,1,\ldots$$
- ## 模拟方法：
  - 首先，由于Poisson分布等分布函数涉及阶乘等，计算复杂度较高，通常采用递推的方式进行计算，即：
    - $$\frac{p_{i+1}}{p_i}=\frac{\frac{e^{-\lambda}\lambda^{i+1}}{(i+1)!}}{\frac{e^{-\lambda}\lambda^i}{i!}}=\frac{\lambda}{i+1}$$
      $$\boxed{p_{i+1}=\frac{\lambda}{i+1}p_i, \quad i\geqslant 0}$$
  - 下对泊松分布进行模拟：
    - STEP 1: Generate a random number U.
      STEP 2: $i=0, p=e^{-\lambda}, F=p$.
      STEP 3: If $U<F$, set $X=i$ and stop.
      STEP 4: $p=\lambda p/(i+1), F=F+p, i=i+1$.
      STEP 5: Go to Step 3.
- ## 代码实现：

```
% function X=cspoirnd(lam,n)
% This function will generate Poisson(lambda)

function x=cspoirnd(lam,n)
x=zeros(1,n);
j=1;
while j<n
  flag =1;
    % initialize quantities
    u=rand(1);
    i=0;
    p=exp(-lam);
    F=p;
    while flag % generate the variate needed
      if u<=F % then accept
          x(j)=i;
            flag=0;
```

```
            j=j+1;
        else % move to next probability
          p=lam*p/(i+1);
            i=i+1;
            F=F+p;
        end
    end
end
```

- ## 4.3 生成二项分布随机变量 Generating Binomial Random Variables

collapsed:: true

- ### 模型概述：
  - 与泊松分布的生成方法类似；
  - 注意的是当np或泊松分布中的λ较大时，算法有较大的改进空间。但讲义中并未提及，故略去。
- ### 模拟目的：
  - 生成二项分布(n,p)随机变量X，即：
    - $$P\{X=i\}=\frac{n!}{i!(n-i)!}p^i(1-p)^{n-i},\quad i=0,1,\ldots,n$$
- ### 模拟方法：
  - 同样采用递归形式：
    - $$P\{X=i+1\}=\frac{n-i}{i+1}\frac{p}{1-p}P\{X=i\}$$
      - 注：在算法实现时，注意到$p/(1-p)$为常数（与i无关）故可以另记之方便计算.
  - 实现算法为：
    - STEP 1: Generate a random number U.
      STEP $2: c=p/(1-p), i=0, \mathrm{pr}=(1-p)^n, F=\mathrm{pr}$
      STEP 3: If $U<F$, set $X=i$ and stop.
      STEP 4: $\mathrm{pr}=[c(n-i)/(i+1)]\mathrm{pr}, F=F+\mathrm{pr}, i=i+1$.
      STEP 5: Go to Step 3.
  - 说明：
    - 这里c即为上述的常数项，pr为递归形式的$P\{X=i\}$，F为累积的分布函数；
    - 要注意到循环的次数总是比确定的X值大一。显然，即使$X=0$，也要经过一次循环比较才能确定；
    - 根据二项分布的性质，当$p>1/2$时可以通过上述算法生成$Y\sim b(n,1-p)$，$X=n-Y$即为所求；
    - 另一种实现方法为模拟n次实验的结果.
- ### 代码实现：

```
% set up storage space for the variables
X=zeros(1,100);
% These are the x's in the domain
x=0:2;
```

```
% These are the prob. masses.
pr=[0.3 0.2 0.5];
% Generate 100 rv's from the desired distribution.
for i=1:100
  u=rand; %generate U
    if u<=pr(1)
      X(i)=x(1);
    elseif u<=sum(pr(1:2))
      % it has to be between 0.3 and 0.5
      X(i)=x(2);
    else
      X(i)=x(3);
        % it has to be between 0.5 and 1.
    end
end
```

-

- # Chapter 5 生成连续随机变量 (Generating Continuous Random Variables)
  - ## 5.1 逆变换法 Inverse Transform Algorithm

    collapsed:: true

    - ### 原理:
      - 若U为$(0,1)$区间随机数，则任意连续型随机变量可以通过下式确定:
        - $X = F^{-1}(U)$
    - ### 例5b: 生成指数分布随机变量
      - 要求：生成随机变量$X \sim \exp(\lambda)$，即:
        - $$F(x) = 1 - e^{-\lambda x}$$
      - 解:
        - 令
          - $$u = F(x) = 1 - e^{-\lambda x}$$
        - 从上式中解出x有:
          - $$x = -\frac{1}{\lambda}\log(1-u)$$
        - 等价于:
          - $$\boxed{X = -\frac{1}{\lambda}\log(u)}$$
      - 代码实现:

```
% set up the parameters.
lam =2;
% generate the rv's
uni=rand(1,n);
X=-log(uni)/lam;
```

- ## 5.2 接受拒绝法 Rejection Method

collapsed:: true

- ### 模拟目的:
  - 通过辅助分布$g(x)$生成服从较复杂的密度函数$f(x)$的随机变量.
- ### 模拟方法:
  - 首先确定一个辅助的"建议分布"Y，已知其概率密度函数为$g_Y(y)$，用来产生候选样本；
    - 注：理论上Y可服从任意分布，而在实际计算中通常采取与目标分布$f(x)$形状较为接近的分布。
  - 另生成一个$U(0,1)$用于后续比较；
  - 计算一个常数c，使得对于$\forall x$，都有$f(x)/g(x) \leq c$.
    - 注：为了计算方便，常常选择满足条件的c中的最小值。
  - 若不等式$U \leq \frac{f(Y)}{cg(Y)}$成立，则接受 Y（令$X = Y$），否则则重新生成进行比较。
  - 算法实现：
    - STEP 1: Generate Y having density g.
      STEP 2: Generate a random number U.
      STEP 3: If $U \leqslant \frac{f(Y)}{cg(Y)}$, set $X = Y$. Otherwise, return to Step 1 .
- ### 原理:
  - 令 X 为想要生成的指定分布的随机数, 令 N 为必要迭代次数:

$$\begin{aligned}
P\{X \leq x\} &= P\left\{Y_N \leq x\right\} \\
&= P\{Y \leq x \mid U \leq f(Y) / \operatorname{cg}(Y)\} \\
&= \frac{P\{Y \leq x, U \leq f(Y) / \operatorname{cg}(Y)\}}{K} \\
&= \frac{\int P\{Y \leq x, U \leq f(Y) / \operatorname{cg}(Y) \mid Y = y\} g(y) d y}{K} \\
&= \frac{\int_{-\infty}^x(f(y) / \operatorname{cg}(y)) g(y) d y}{K} \\
&= \frac{\int_{-\infty}^x f(y) d y}{K c}
\end{aligned}$$

  其中 $\mathrm{K}=\mathrm{P}(\mathrm{U} \leq f(Y) / \operatorname{cg}(Y)\}$. 令 $x \rightarrow \infty$ 可知 $\mathrm{K}=1 / c$ 证毕.

  - 注：在每次循环判断时，若$U > f/cg$，则在下一次循环时事实上可以不再重新生成随机数，而是可以令$\frac{U-f(Y)/\operatorname{cg}(Y)}{1-f(Y)/\operatorname{cg}(Y)} = \frac{cUg(Y)-f(Y)}{cg(Y)-f(Y)}$作为下一次的$U$以减少计算。

- ### 注：
  - 该方法由Von Neumann创造，其中的Y为(a,b)区间的均匀分布；
  - 由于每次接受的概率为：$P(U \leq f(Y)/cg(Y)) = 1/c$，故平均循环次数的几何平均为$c$
  - 在循环中若拒绝，即$U > f(Y)/cg(Y)$，此时并不需要重新生成随机数，而是可以通过下面的公式直接利用先前拒绝的U计算出新的随机数，以减少运算量：
    - $\frac{U-f(Y)/cg(Y)}{1-f(Y)/cg(Y)} = \frac{cUg(Y)-f(Y)}{cg(Y)-f(Y)}$
- ### 例5d
  - 要求：
    - 生成随机变量X服从：
    - $f(x) = 20x(1-x)^3$
  - 解：
    - 令
      - $g(x) = 1, \ \ 0 < x < 1$
    - 下求解最优c：
      - 已知：
        - $\frac{f(x)}{g(x)} = 20x(1-x)^3$
      - 通过求导可知上式的极大值点为$x = 1/4$，极大值为$135/64$
      - 故$c = 135/64$
    - 因此有：
      - $\frac{f(x)}{cg(x)} = \frac{256}{27}x(1-x)^3$
    - 下开始模拟过程：
      - 生成随机数U_1, U_2；
      - 若$U_2 \leqslant \frac{256}{27}U_1(1-U_1)^3$则接受，令$X = U_1$，否则重复上述操作。
- ### 例5f
  - 要求：生成标准正态随机数$Z \sim N(0,1)$
  - 解：
    - 先考虑$X = |Z|$的分布，即：
      - $$f(x) = \frac{2}{\sqrt{2\pi}}e^{-x^2/2} \quad 0 < x < \infty$$
    - 令$g(x)$为$\exp(1)$的概率密度函数，故有：
      - $$\frac{f(x)}{g(x)} = \sqrt{2/\pi}e^{x-x^2/2}$$
    - 可求其最大值得到最优的c值：
      - $c = \max\frac{f(x)}{g(x)} = \sqrt{\frac{2e}{\pi}}$
    - 故有：

- $$\begin{aligned}\frac{f(x)}{cg(x)} &= \exp\left\{x-\frac{x^2}{2}-\frac{1}{2}\right\} \\ &= \exp\left\{-\frac{(x-1)^2}{2}\right\}\end{aligned}$$
- 因此可以按下过程生成随机数：
  - STEP 1: Generate Y, an exponential random variable with rate 1.
    STEP 2: Generate a random number U.
    STEP 3: If $U \leqslant \exp\left\{-(Y-1)^2/2\right\}\star$, set $X=Y$. Otherwise, return to Step 1 .
- 在生成了绝对值正态分布后，我们可以令Z以相等的概率等于X或$-X$，即有标准正态函数的分布。

- 改进：
  - 对上述$\star$式左右取对数，有：
    - $$-\log U \geqslant (Y-1)^2/2$$
  - 根据计算又知$-\log U$服从$\exp(1)$分布，故算法可改进为：
    - STEP 1: Generate Y_1, an exponential random variable with rate 1.
      STEP 2: Generate Y_2, an exponential random variable with rate 1 .
      STEP 3: If $Y_2-(Y_1-1)^2/2>0$, set $Y=Y_2-(Y_1-1)^2/2$ and go to Step 4 .
      Otherwise, go to Step 1.
      STEP 4: Generate a random number U and set
      $$Z=\begin{cases} Y_1 & \text{if} \quad U \leqslant \frac{1}{2} \\ -Y_1 & \text{if} \quad U > \frac{1}{2}\end{cases}$$
- 通过生成标准正态Z，其余正态函数可以通过$\mu+\sigma Z$生成.

5.3 极坐标法正态随机数 Polar Method for Generating Normal Random Variables

collapsed:: true

- **基本知识：**
  - 极坐标：
    - $R^2=X^2+Y^2$
    - $\tan\Theta=\frac{Y}{X}$
  - 正态分布：
    - 由于X, Y独立，其联合概率密度函数为
      - $$\begin{aligned} f(x, y) &= \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \\ &= \frac{1}{2\pi} e^{-(x^2+y^2)/2}\end{aligned}$$
  - 函数变换：
    - $P(X\in C, Y\in D)=\iint_{X\in C, Y\in D} f(x,y)dxdy=\iint_{u\in C', v\in D'} f(g_1(u,v), g_2(u,v))|J|dudv$

- **Box-Muller变换**
  - 将X,Y的联合密度函数转化到极坐标系中，有：
    - 令$d=x^2+y^2,\theta=\arctan(y/x)$,故：
    - $$f(d,\theta)=\frac{1}{2}\frac{1}{2\pi}e^{-d/2},\quad 0<d<\infty,0<\theta<2\pi$$
  - 注意到，上述密度函数可以认为是均值为2的指数分布($\frac{1}{2}e^{-d/2}$)与$(0,2\pi)$的均匀分布的密度函数($\frac{1}{2\pi}$)乘积
    - 故有：R^2与Θ彼此独立，R^2服从均值为2的指数分布，Θ服从$(0,2\pi)$的均匀分布
  - 故可以按如下步骤以极坐标法生成正态分布：
    - STEP1: 生成随机数U_1,U_2
    - STEP2: 令$R^2=-2\log U_1$，$\Theta=2\pi U_2$
    - STEP3: 令
      - $$\begin{aligned}X&=R\cos\Theta=\sqrt{-2\log U_1}\cos\left(2\pi U_2\right)\\Y&=R\sin\Theta=\sqrt{-2\log U_1}\sin\left(2\pi U_2\right)\end{aligned}\qquad(\star)$$
  - 说明：Box-Muller变换在计算时的效率较低，这是因为其中在STEP3中涉及到了三角函数$\cos\sin$的计算（而这一计算耗时较长）。为了改进这一特点，下不再生成随机角度Θ，而是直接通过模拟直角三角形的三边长度生成随机三角函数$\cos\Theta,\sin\Theta$，具体方法如下：
- **极坐标法：B-M变化的改进***
  - 改进思路：不再计算模拟的随机角度的三角函数，而是通过模拟单位圆（面）直接计算三角函数的具体数值。
  - 改进步骤：
    - 引入单位圆
      - 若$U\sim(0,1)$，则$2U-1\sim(-1,1)$，故令
        - $$\begin{aligned}V_1&=2U_1-1\\V_2&=2U_2-1\end{aligned}$$
      - 不断生成随机数对(V_1,V_2)并保留满足$V_1^2+V_2^2\leq 1$的部分，则有(V_1,V_2)在如下图所示的单位圆上均匀分布：
        -
      - 对于该随机数对(V_1,V_2)对应的极坐标方程，可知其对应的R^2服从$(0,1)$的均匀分布，而Θ服从$(0,2\pi)$的均匀分布。
    - 模拟$\cos\sin$：
      - $$\begin{aligned}\sin\Theta&=\frac{V_2}{R}=\frac{V_2}{\left(V_1^2+V_2^2\right)^{1/2}}\\\cos\Theta&=\frac{V_1}{R}=\frac{V_1}{\left(V_1^2+V_2^2\right)^{1/2}}\end{aligned}$$
    - 对B-M的改进：
      - 将上述模拟的$\sin\Theta,\cos\Theta$代入B-M中的$(\star)$，有

- $$X=(-2\log U)^{1/2}\frac{V_1}{\left(V_1^2+V_2^2\right)^{1/2}}$$
  $$Y=(-2\log U)^{1/2}\frac{V_2}{\left(V_1^2+V_2^2\right)^{1/2}}$$
- 再令 $S=R^2$，则有：
  - $$X=(-2\log S)^{1/2}\frac{V_1}{S^{1/2}}=V_1\left(\frac{-2\log S}{S}\right)^{1/2}$$
    $$Y=(-2\log S)^{1/2}\frac{V_2}{S^{1/2}}=V_2\left(\frac{-2\log S}{S}\right)^{1/2}$$
- 综上，可知新的模拟步骤为：
  - STEP1: 生成随机数U_1,U_2
  - STEP2: 令 $V_1=2U_1-1, V_2=2U_2-1, S=V_1^2+V_2^2$
  - STEP3: 若$S>1$，返回STEP1
  - STEP4: 否则可按如下原则生成一对标准正态分布：
    - $$X=\sqrt{\frac{-2\log S}{S}}V_1,\quad Y=\sqrt{\frac{-2\log S}{S}}V_2$$

5.4 生成齐次泊松过程 Generating a Poisson Process

collapsed:: true

- **模拟原理**
  - 由以λ为参数的Poisson过程的性质可知：每两次事件发生的时间间隔独立同分布于以λ为参数的指数分布；
  - 因此若模拟一系列均匀分布$U_1,...,U_n$，并令$X_1=-\frac{1}{\lambda}\log U_i$，则$X_i\sim\exp(\lambda)$即可表示事件*i-1*与事件*i*发生的时间间隔；
  - 另外，记$S(I)=\sum_{i=1}^{j}X_i$，则$S(I)$可表示累积到第*j*个事件发生时的时间。
- **模拟过程**
  - STEP1: 初始化$t=0,\quad I=0$;
  - STEP2: 生成随机数U;
  - STEP3: 令$t=t-\frac{1}{\lambda}\log U$；若$t>T$，则停止；
  - STEP4: 令 $I=I+1, S(I)=t$;
  - STEP5: 返回到STEP2.
- **说明：**
  - 最终的I反映了截止到时间T发生的事件个数，$S(I)$为发生到第I个事件所需的时间。

Chapter 6 模拟数据的统计分析 Statistical Analysis of Simulated Data

引入

- simulation的目的是为了在某些随机模型中模拟某些变量的数值θ
- 每一次模拟都会生成一个随机变量X，我们期望这个随机变量的值能够接近目标数值θ
- 在总体的模拟过程中，我们将生成k个这样的随机变量$X_1,...X_k$，其中这些随机变量X_i都独立同分布且均值为θ，随后我们便用样本均值$\bar{X}=\sum_{i=1}^{k}X_i/k$作为期望数值$\theta$的估计量

- 因此本章的研究内容即为k的大小对于θ的估计精度的影响，计算在给定置信度的情况下相应置信区间等。

6.1 样本均值与方差 Sample Mean and Variance

基本概念

- 记$X_1,...X_n$为独立同分布随机变量，以θ,σ^2分别表示总体期望与方差，即$\theta=E(X_i),\sigma^2=Var(X_i)$；
- 样本均值sample mean：
  - $$\bar{X}\equiv\sum_{i=1}^{n}\frac{X_i}{n}$$
    - 由于总体均值θ未知，因此我们用$\bar{X}$估计θ
  - 考虑样本均值期望：
    - $$\begin{aligned}E[\bar{X}]&=E\left[\sum_{i=1}^{n}\frac{X_i}{n}\right]\\&=\sum_{i=1}^{n}\frac{E\left[X_i\right]}{n}\\&=\frac{n\theta}{n}=\theta\end{aligned}$$
    - 因此$\bar{X}$是θ的无偏估计
  - 考虑样本均值的均方误(MeanSquareError)
    - $$\begin{aligned}E\left[(\bar{X}-\theta)^2\right]&=\operatorname{Var}(\bar{X})\quad(\text{ since } E[\bar{X}]=\theta)\\&=\operatorname{Var}\left(\frac{1}{n}\sum_{1}^{n}X_i\right)\\&=\frac{1}{n^2}\sum_{i=1}^{n}\operatorname{Var}\left(X_i\right)\quad\text{(by independence)}\\&=\frac{\sigma^2}{n}\quad\left(\text{ since }\operatorname{Var}\left(X_i\right)=\sigma^2\right)\end{aligned}$$
  - 综上：当$\sigma/\sqrt{n}$较小时，$\bar{X}$是一个较好的估计
- 样本方差：
  - 定义：
    - $$S^2=\frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2}{n-1}$$
  - 性质：
    - $$E\left[S^2\right]=\sigma^2$$
  - 因此可以通过S^2估计σ^2

- **判断停止算法：**
  - **内容**
    - 1） 选择可接受的参数标准差d
    - 2） 生成至少100项随机值
      - 3）继续生成随机数，直到满足$\frac{S}{\sqrt{k}} < d$时停止，其中S为这k项随机数的样本标准差
    - 4） θ的估计值通过$\bar{X}$给出
  - **说明：**
    - 为提高算法的计算效率，下给出一递归算法计算样本的均值与方差，这样的递归内容使得每一个新生成的随机数给出后不用重新计算。迭代方法如下：
      - 令$S_1^2 = 0, \bar{X}_0 = 0$，则：
        - $$\bar{X}_{j+1} = \bar{X}_j + \frac{X_{j+1} - \bar{X}_j}{j+1}$$ $$S_{j+1}^2 = \left(1 - \frac{1}{j}\right)S_j^2 + (j+1)\left(\bar{X}_{j+1} - \bar{X}_j\right)^2$$
- **例 8a （对于点估计的理解）**
  - 问题描述：
    - 假设有一服务系统，其中在每天下午5点后不再允许新顾客进入。已知每天的顾客光临服务情况服从统一分布状况，并且我们希望能够研究最后一位顾客离开服务系统的时间。
    - 假设：期望有至少95%的把握使得估计的时间与真值之间的差距不超过15秒
  - 解答：
    - 为了满足题设要求，可以连续生成随机数对顾客到来情况进行模拟。已知一共生成k个数值（$k \geq 100$），并且需要满足$1.96S/\sqrt{k} < 15$
    - 故估计的预期时间则为k项数值的平均值
- **对于概率p 的估计**
  - 当估计概率p时，可以构造随机变量X，满足：
    - $$X_i = \begin{cases} 1 \text{ 以概率p} \\ 0 \text{ 以概率1-p} \end{cases}$$
  - 此时可知X_i服从Bernoulli分布，故由分布的方差特点可有如下估计：
    - $Var(X_i) = \bar{X}_n(1 - \bar{X}_n)$
  - 故有如下判断停止算法：
    - 1）选择可接受的估计标准差d
    - 2）生成至少100项随机数
    - 3）继续生成，直到满足$\left[\bar{X}_k\left(1 - \bar{X}_k\right)/k\right]^{1/2} < d$停止
    - 4）参数p的估计为$\bar{X}_k$
- **例 8c**
  - 题目描述：
    - 在例8a的情况下，若想研究在下午5:30仍然有顾客的概率，我们则可以模拟多天的情况，并记：

- $$X_i = \begin{cases} 1 \text{ 第i天5:30仍有顾客} \\ 0 \text{ 没有顾客} \end{cases}$$
- 通过生成至少100天的数值内容，使得至少k满足$[p_k(1-p_k)/k]^{1/2} < d$，其中$p_k = \bar{X}_k$, d为可接受的标准差d

6.2 总体均值区间估计 Interval Estimates of a Population Mean

区间估计方法

- 假设$X_1, \ldots, X_n$为独立同分布随机变量，均值θ，方差为σ^2，则当n较大时，根据中心极限定理有：
  - $$\sqrt{n}\frac{(\bar{X}-\theta)}{\sigma} \dot{\sim} N(0,1)$$
- 此外，若σ也未知，通过S可以估计有：
  - $$\sqrt{n}(\bar{X}-\theta)/S \dot{\sim} N(0,1)$$
- 若Z为正态分布，则有分位数：$P\{Z > z_\alpha\} = \alpha$
  - 故：
    - $$P\left\{-z_{\alpha/2} < Z < z_{\alpha/2}\right\} = 1-\alpha$$
- 代入估计值，有
  - $$P\left\{-z_{\alpha/2} < \sqrt{n}\frac{(\bar{X}-\theta)}{S} < z_{\alpha/2}\right\} \approx 1-\alpha$$
  - 等价于
    - $$P\left\{-z_{\alpha/2} < \sqrt{n}\frac{(\theta-\bar{X})}{S} < z_{\alpha/2}\right\} \approx 1-\alpha$$
- 故推出$1-\alpha$的置信估计区间：$\bar{X} \pm z_{\alpha/2}S/\sqrt{n}$

区间估计停止算法

- 1）确定α, l
- 2）生成至少100项随机数，直到生成的总数k满足$2z_{\alpha/2}S/\sqrt{k} < l$，其中$S$为$k$个样本标准差（通过上文给出的递推算法不断更新）
- 3）若$\bar{x}, s$是$\bar{X}, S$的观测值，则参数θ的$1-\alpha$置信区间（长度小于l）为$\bar{x} \pm z_{\alpha/2}s/\sqrt{k}$

对于概率的区间估计

- 假设$X_1, \ldots, X_n$为概率p的Bernoulli分布，即
  - $$X_i = \begin{cases} 1 \text{ with probability } p \\ 0 \text{ with probability } 1-p \end{cases}$$
- 故有
  - $$\sqrt{n}\frac{(\bar{X}-p)}{\sqrt{\bar{X}(1-\bar{X})}} \dot{\sim} N(0,1)$$

- 因此可确定α的置信区间：
  - $$P\left\{-z_{\alpha/2}<\sqrt{n}\frac{(\bar{X}-p)}{\sqrt{\bar{X}(1-\bar{X})}}<z_{\alpha/2}\right\}=1-\alpha$$
  - 等价于
    - $$P\left\{\bar{X}-z_{\alpha/2}\sqrt{\bar{X}(1-\bar{X})/n}<p<\bar{X}+z_{\alpha/2}\sqrt{\bar{X}(1-\bar{X})/n}\right\}=1-\alpha$$
- 因此p的置信区间为
  - $$p_n \pm z_{\alpha/2}\sqrt{p_n\left(1-p_n\right)/n}$$

6.3 均值均方误的脱靴法估计 Bootstrapping Technique for Estimating MSE

MSE 定义

- 假设$X_1,\ldots,X_n$为独立同分布随机变量，分布函数为F
- 我们想要研究关于这些变量的分布有关的某种参数，记为$\theta(F)$
  - 例如平均数、中位数等
- 同时还可以通过这些已知的随机变量计算出一个对于参数θ的某种估计，记为$g(X_1,\ldots,X_n)$
- 为了评估估计值与真实值之间的偏差，引入MSE，其定义为：
  - $MSE(F)\equiv E_F[g(X_1,\ldots,X_n)-\theta(F)]^2$
    - 其中，E_F是说明期望是在X都服从分布F的条件下给出的
- 如果对于F的分布有较好的了解，便可以直接计算MSE；但在大多数时候，我们对于整体位置参数的分布也并不清楚，因此首先需要对F进行估计，才能最终对MSE进行估计。

经验分布函数 F_e

- 在总体分布函数未知的情况下，可以通过下面定义的经验分布对总体分布进行9评估：
  - $F_e(x)=\frac{\text{number of } i \ \text{ s.t.}\{X_i\le x\}}{n}$
  - 即满足$X_i<x$的随机变量的占比
- 经验分布逼近总体函数
  - 由强大数定律，可知当$n\to\infty$，经验分布以概率1收敛于总体分布：
    - $F_e(x)\to F(x)\ \ ,a.s.$

MSE的*bootstrap* 估计

- 用经验分布近似估计总体分布，则可以得到目标参数θ的近似估计$\theta(F_e)$，进而可以得到MSE的估计：
  - $MSE(F_e)=E_{F_e}[g(X_1,\ldots,X_n)-\theta(F_e)]^2$
- 需要特别指出的，估计的真正问题在于本身想要研究的参数θ已经是采用样本估计而来的（$g(\ldots X_i\ldots)$）；而又由于对总体的分布情况未知，在评价估计的效果时，所谓的评价标准（即θ也是未知的。因此相当于在用估计出的标准来评价估计的效果。这种“自导自演”的合理性的保证是该方法的精髓所在。粗略地来说，该方法相当于在有限的样本中反复进行抽样模拟总体分布，因此对于参数的估计与对于总体分布的估计，这两个部分应当是“独立”的。

- ### 例8d：期望的Bootstrapping MSE估计
  - 本例中想要研究的参数为总体的均值$\theta(F)=E[X]$，对于此，我们可以通过样本的均值$\bar{X}=\sum X_i/n$来近似估计。下面想要通过Bootstrapping法求出这一个估计的MSE（的估计）。
  - 不妨设抽出的样本为$x_1,x_2,\ldots,x_n$
  - 则可知：$\theta(F_e)=\bar{x}=\sum x_i/n$
  - 则有：
    - $MSE(F_e)=E_{F_e}[g(X_1,\ldots,X_n)-\theta(F_e)]^2=E_{F_e}[\sum X_i/n-\bar{x}]^2$
  - 又因为$\bar{x}=E_{F_e}[X]=E_{F_e}[\sum X_i/n]$
  - 故
    - $$\begin{aligned}MSE(F_e)&=E_{F_e}[\frac{\sum X_i}{n}-\bar{x}]^2\\&=E_{F_e}[\frac{\sum X_i}{n}-E_{F_e}(\frac{\sum X_i}{n})]\\&=Var_{F_e}(\sum^n\frac{X_i}{n})\\&=\frac{Var_{F_e}(X)}{n}\end{aligned}$$
  - 又知道
    - $$\begin{aligned}\mathrm{Var}_{F_e}(X)&=E_{F_e}\left[(X-E_{F_e}[X])^2\right]\\&=E_{F_e}\left[(X-\bar{x})^2\right]\\&=\frac{1}{n}\left[\sum_{i=1}^n(x_i-\bar{x})^2\right]\end{aligned}$$
  - 故将Var代入MSE中，有：
    - $\mathrm{MSE}(F_e)=\frac{\sum_{i=1}^n(x_i-\bar{x})^2}{n^2}$
  - 需要注意的是，若采用正常的估计方法，求的的MSE为$\sum(x_i-\bar{x})^2/(n(n-1))$，这与Bootstrapping法得到的结果相当接近
- 一般情况下的说明：
  - 对于Bootstrapping法，我们想要通过一组样本$x_1,x_2,\ldots,x_n$来评估估计$g(X_1,X_2,\ldots,X_n)$的好坏。这种方法的核心之处即在于对于这组数据的反复利用，即让随机向量$(X_1,X_2,\ldots,X_n)$有放回地"抽取"$\{x_1,x_2,\ldots,x_n\}$，得到n组随机向量$(X_{11},X_{12},\ldots,X_{1n})\ldots(X_{n1},X_{n2},\ldots,X_{nn})$，相应地可以计算得$n$组目标参数的估计值$g_1(X_{11},X_{12},\ldots,X_{1n}),\ldots,g_n(X_{n1},X_{n2}\ldots,X_{nn})$，再根据此进行计算。
  - 因此我们有：
    - $$\begin{aligned}\mathrm{MSE}(F_e)&=E_{F_e}\left[(g(X_1,\ldots,X_n)-\theta(F_e))^2\right]\\&=\sum P\cdot[g(x_{i1},x_{i2},\ldots,x_{in})-\theta(F_e)]^2\end{aligned}$$
  - 又由于X_i等可能地取到$\{x_1,x_2,\ldots,x_n\}$中地任何值，因此每一组随机向量可能取值的概率为$P=1/n^n$
  - 最终，我们有：
    - $\mathrm{MSE}(F_e)=\sum_{i_n}\cdots\sum_{i_1}\frac{[g(x_{i_1},\ldots,x_{i_n})-\theta(F_e)]^2}{n^n}$
  - 然而这种方法由于n^n的存在，在n较大时无法实际计算，故需要引入简化形式。

- ## 简化Bootstrapping
  - 由于上文所述的原因，我们引入下面的简便方式：
    - 生成一组包含n个数的随机数$X_1^{(1)}, X_2^{(1)}, \ldots, X_n^{(1)}$
    - 计算$Y_1 = [g(X_1^{(1)}, \ldots, X_n^{(1)}) - \theta(F_e)]^2$
    - 同理可以生成$X_1^{(2)}, \ldots, X_n^{(2)}$并计算$Y_2$
    - 以此类推，一共得到r项：$Y_1, Y_2, \ldots, Y_n$
    - 最终有：$MSE(F_e) = \sum_{i=1}^{r} Y_i / r$
  - 说明：该简化版本的含义是非常符合直觉的。在正常的计算过程中，需要对一大串随机变量的计算结果（不妨记为ϕ）求解期望，然而期望的求解是相当计算复杂的。而对于其的改进版本相当于通过模拟的方法近似代替期望，即生成n组ϕ_i，并对$\phi_1, \ldots, \phi_n$求解平均值，以均值代替期望，大大降低了计算复杂度。经验而言，大约生成100组ϕ，即$r \geq 100$时基本上即可满足一般需求。
- ## 例8e：均值极限状态
  - 现考虑均值的极限状态。考虑顾客在某系统内花费的时间。记W_i为第i个顾客在系统内花费的时间(需注意到这些W_i并非独立同分布的)，且$i \geq 1$，现考虑：
    - $\theta \equiv \lim_{n \to \infty} \frac{W_1 + \ldots + W_n}{n}$
  - 首先证明这一极限(即θ)的存在性：
    - 记N_i为第i天到达的顾客数，则D_i为第i天的所有顾客在系统中花费的时间总和，即有：
      - $$\begin{aligned} &D_1 = W_1 + \ldots + W_{N_1} \\ &D_2 = W_{N1+1} + \ldots + W_{N1+N2} \\ &\ldots \\ &D_i = W_{N1+\ldots+N_{i-1}} + \cdots + W_{N_1+\cdots+N_i} \end{aligned}$$
    - 则θ可表示为：
      - $\theta = \lim_{m \to \infty} \frac{D_1 + D_2 + \cdots + D_m}{N_1 + N_2 + \cdots + N_m}$
      - 上下同除以m，有：
      - $\theta = \lim_{m \to \infty} \frac{(D_1 + D_2 + \cdots + D_m)/m}{(N_1 + N_2 + \cdots + N_m)/m}$
    - 由于每天的顾客数量以及时间分布服从相同的概率分布，且彼此独立，故D_i与N_i分别是独立同分布的。则由强大数定律可以保证，上下两项分别在极限状态的均值可以以概率1收敛于其期望，因此我们有：
      - $\theta = \frac{E[D]}{E[N]}$
        - $E[D]$为每天顾客在系统中花费的期望时间，$E[N]$为每天系统中期望到达的顾客数量。
  - 因此我们可以通过样本均值估计期望，即得到θ的估计如下：
    - $\hat{\theta} = \frac{\bar{D}}{\bar{N}} = \frac{(D_1 + \cdots + D_k)/k}{(N_1 + \cdots + N_k)/k} = \frac{D_1 + \cdots + D_k}{N_1 + \cdots + N_k}$
  - 接着我们希望能够给出θ的估计的MSE:
    - $\mathrm{MSE} = E\left[\left(\frac{\sum_{i=1}^{k} D_i}{\sum_{i=1}^{k} N_i} - \theta\right)^2\right]$
  - 通过Bootstrapping给出MSE的估计：
    - 假设D_i, N_i的观测值为d_i, n_i，$i = 1, \ldots, k$
    - 故有经验联合分布函数：
      - $P_{F_e}\{D = d_i, N = n_i\} = \frac{1}{k}, \quad i = 1, \ldots, k$
    - 根据经验分布函数，可以由此求得：
      - $$E_{F_e}[D] = \bar{d} = \sum_{i=1}^{k} d_i / k, \quad E_{F_e}[N] = \bar{n} = \sum_{i=1}^{k} n_i / k$$

- 由上面推导的$\theta(F)=\frac{E[D]}{E[N]}$，有
  - $\theta\left(F_e\right)=\frac{E_{F_e}[D]}{E_{F_e}[N]}=\frac{\bar{d}}{\bar{n}}$
- 因此，通过Bootstrapping给出经验MSE:
  - $\operatorname{MSE}\left(F_e\right)=E_{F_e}\left[\left(\frac{\sum_{i=1}^k D_i}{\sum_{i=1}^k N_i}-\frac{\bar{d}}{\bar{n}}\right)^2\right]$
- 而上面同样证明，若进行正式的期望计算，需要计算的次数为k^k次，因此在此通过r次独立实验进行简化模拟：
  - 生成第1组k维独立随机向量：$D_i^{(1)}, N_i^{(1)},\ \ i=1,2,\ldots,k$
    - 根据上面的推导，计算：
      - $Y_1=\left(\frac{\sum_{i=1}^k D_i^1}{\sum_{i=1}^k N_i^1}-\frac{\bar{d}}{\bar{n}}\right)^2$
  - 以此类推，生成r组(取100左右)这样的数：$Y_1,\ldots,Y_r$
  - 这一组数的均值$\sum_{i=1}^r Y_i/r$即可作为MSE的近似模拟

- # Chapter 7 降低方差手段 Variance Reduction Techniques

  collapsed:: true

  - ## 7.1 对偶变量 Use of Antithetic Variables
  - ## 7.2 控制变量 Use of Control Variates
  - ## 7.3 条件假设 Variance Reduction by Conditioning
  - ## 7.4 分层抽样 Stratified Sampling

    -
-
  -