3장. 재미있는 확률문제 5개

재미있는 이야기라고 해서 시작하는 경우 치고 제대로 재미있는 경우가 많지 않은 것은 사실입니다. 그런 것을 요새 썰렁하다고 하더군요. 그럼에도 불구하고 이 장에서 '재미있는' 이야기를 하고 싶군요. 그 이유는 내가 이런 문제들은 처음 접하였을 때 무척 재미를 느꼈고 (어떤 문제에 대하여는 20여 년이 지난) 지금까지도 그 여운이 남아 있기 때문입니다. 그러니 여러분들은 "도대체 무슨 문제이기에 저 사람이 재미있어 했는가" 하는 궁금증으로 이 장을 읽기 바랍니다.

첫 번째 문제인 도박꾼의 파산(gambler's ruin)은 카지노의 원리를 이해하는 데 도움이 됩니다. 그것을 제대로 이해하면 무엇에 도움이 될까요? 돈을 벌 수 있는 방법을 배우게 될 것이라고 기대하지는 마십시오. 그 대신, 돈을 덜 잃을 수 있는 방법을 알게 될 것입니다.

두 번째는 쿠폰 수집(coupon collection) 문제입니다. n종의 쿠폰을 적어도 한 장씩 모아야 하는 경우 모두 몇 장의 쿠폰이 필요할까요? 여러분이 이것을 배우게 되면 초등학교 다니는 조카들에게 써먹을 기회가 있을 것입니다. 그 녀석들이 말귀를 알아들을 수 있을까는 잘 모르겠습니다만.

세 번째 문제는 뷔퐁의 바늘(Buffon's needle)인데 원주율 π의 추정과 관련이 있는 고전적인 문제입니다. 기하적 확률이 시작된 문제이기도 하고요.

네 번째 문제는 TV 쇼 "Let's Make a Deal"에서 나왔습니다. 3개의 방문이 있고 그 중 한 방에는 대단한 경품이 걸려 있습니다. 그것을 맞추는 문제입니다. 순간의 결정이 중요하니까 잘 생각해야 합니다.

다섯 번째는 산불(forest fire)이 얼마만큼 어떻게 번지는가에 관한 확률과정을 모의시행을 통하여 이해하는 것입니다. 자연적인 발화는 과히 나쁘지 않습니다.

차례 :

3.1 도박꾼의 파산 (Gambler's Ruin)

'도박꾼의 파산'을 첫 문제로 꼽는 이유는 두가지입니다. 하나는 이 문제가 카지노의 원리를 명쾌하게 보여주기 때문입니다. 따라서 여러분들은 라스베가스 같은 데 가게 되는 경우 어떻게 행동해야 할 것인가를 배우게 될 것입니다. 다른 하나는 이 문제의 풀이가 수학적 관점에서 무척 재미있기 때문입니다. 고등학교 수준의 수열을 사용하여 풀 수 있는 정점(頂點)의 문제입니다. 문제와 풀이는 다음과 같습니다.

<u>파산확률</u> (ruin probability)

도박꾼 H가 k 달러를 들고 카지노에 들어갑니다. 그는 매번 1 달러를 걸고 이길 확률이 p, 질 확률이 q인 게임을 합니다 ($p+q=1$, $p,q>0$). 그는 돈을 다 잃거나 (파산선언), 아니면 그의 돈이 n이 될 때(목표달성)까지 게임을 계속합니다. 그가 파산할 확률은 얼마입니까? (k와 n은 모두 정수이고 $1<k<n$ 입니다.)

```
                          k-1   k+1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
0  1                 확률q     k     확률p              n
```

풀이는 다음과 같습니다. 그 확률을 x_k 라고 합시다. 즉

$$x_k = P\{\text{Ruin} \mid k\},\ 0 < k < n$$

으로 나타냅시다. 1 게임 후, H의 잔고는 $k+1$ (p의 확률로), 또는 $k-1$ (q의 확률로)이 될 것입니다. 따라서

점화식: $\quad x_k = p\ x_{k+1} + q\ x_{k-1}$

초기조건: $\quad x_0 = 1,\quad x_n = 0$

을 얻습니다. 점화식으로부터

$$(p+q)\,x_k = p\ x_{k+1} + q\ x_{k-1} \quad\Rightarrow\quad p\,(x_{k+1} - x_k) = q\,(x_k - x_{k-1})$$

$$\therefore\ (x_{k+1} - x_k) = r\,(x_k - x_{k-1}),\quad r \equiv q/p. \tag{1}$$

경우 1: $r=1$ (즉 $p=q$). 이 경우엔 (1)이

$$x_{k+1} - x_k = x_k - x_{k-1} = \cdots = x_1 - x_0,\quad k = 0, 1, \cdots, n-1$$

로 정리되므로 $\{x_k\}$는 등차수열입니다. 따라서

$$x_k = x_0 + k(x_1 - x_0), \quad k = 1, 2, \cdots, n.$$

초기조건에 의하여 $x_0 = 1$, $x_n = 0$ 이므로

$$x_n = x_0 + n(x_1 - x_0) = 0. \quad \Rightarrow \quad x_1 - x_0 = -1/n.$$

그러므로

$$x_k = 1 - \frac{k}{n}, \quad k = 1, 2, \cdots, n.$$

경우 2: $r \neq 1$ (즉 $p \neq q$).

이 경우엔 (1)로부터 $\{x_{k+1} - x_k\}_{k=0}^{n-1}$이 등비수열이 됩니다. 따라서

$$x_{k+1} - x_k = r^k(x_1 - x_0), \quad k = 0, 1, \cdots, n-1.$$

이 식을 순서대로 나열하면

$$\begin{aligned} x_k - x_{k-1} &= r^{k-1}(x_1 - x_0), \\ x_{k-1} - x_{k-2} &= r^{k-2}(x_1 - x_0), \\ &\vdots \\ x_1 - x_0 &= r^0 \ (x_1 - x_0). \end{aligned}$$

위 식들을 모두 더하면

$$x_k - x_0 = \sum_{i=0}^{k-1} r^i \ (x_1 - x_0) = \frac{1 - r^k}{1 - r} \ (x_1 - x_0).$$

초기조건에 의하여 $x_0 = 1$, $x_n = 0$ 이므로

$$x_n - x_0 (= -1) = \frac{1 - r^n}{1 - r} \ (x_1 - x_0)$$

$$\Rightarrow \quad (x_1 - x_0) = \frac{-1}{(1 - r^n)/(1 - r)} = -\frac{1 - r}{1 - r^n}.$$

그러므로

$$x_k = 1 - \frac{1 - r^k}{1 - r^n} = \frac{r^k - r^n}{1 - r^n}, \quad k = 1, 2, \cdots, n.$$

<그림 1>을 보세요. ■

도박꾼이 불리한 $p < 0.5$ 인 게임에서 도박꾼 H가 목표를 설정하지 않고 (즉 n을 무한대로 놓고) 게임을 반복한다면 그는 필패(必敗)할 수밖에 없습니다. 그 이유는

$$\lim_{n \to \infty} x_k = \lim_{n \to \infty} \frac{r^k - r^n}{1 - r^n} = 1 \;\; (\because \; r = q/p \; > 1)$$

이기 때문입니다.

$p = 0.5$인 게임을 공정하다고 합니다. 그러나 이 경우에서도 도박꾼 H가 목표를 설정하지 않고 게임을 반복한다면 역시 필패(必敗)할 수밖에 없습니다. 왜냐하면

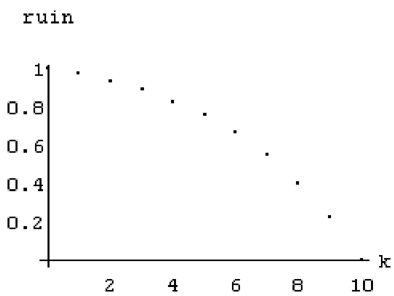
$$\lim_{n \to \infty} x_k = \lim_{n \to \infty} \left(1 - \frac{k}{n}\right) = 1$$

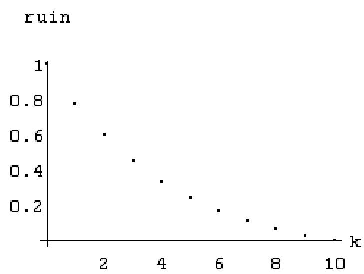
이기 때문입니다.

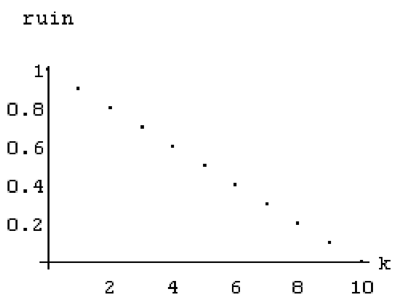
도박꾼이 유리한 $p > 0.5$인 게임에서는 도박꾼 H가 목표를 설정하지 않고 게임을 반복하더라도 도박꾼이 파산하지 않을 수 있습니다 (다시 말하면, 그가 카지노의 돈을 모두 차지할 가능성이 있는 것입니다). 왜냐하면 $r = q/p \; < 1$이므로

$$\lim_{n \to \infty} x_k = \lim_{n \to \infty} \frac{r^k - r^n}{1 - r^n} = r^k \; < 1 \;\; (\because \; r = q/p \; < 1)$$

입니다. 그러나 꿈에서 깨십시오. 그런 불리한 게임을 주관하는 카지노는 세상에 없습니다.







<그림 1> 파산확률 x_k for $n = 10$
左上 → $p < 0.5$인 경우
左下 → $p = 0.5$인 경우
右上 → $p > 0.5$인 경우

여기서 얻는 결론은 명확합니다. 카지노에 들어가기 전에 목표 n을 명확히 설정하고 목표가 달성되면 더 이상 욕심을 내지 말고 자리를 떠야 한다는 것입니다. 그렇지 않으면 1의 확률로 도박꾼이 파산하게 되어 있습니다.

많은 사람들은 돈을 따러 카지노에 가는 것이 아니라 즐거운 시간을 보내려고 카지노에 갑니다. 그렇다면 얼마동안 즐거울 수 있을까요? 즉 몇 번이나 게임을 할 수 있을까, 그 기대값 D_k를 계산하여 봅시다.

게임 수의 기대값 (expected duration)

1 게임 후, H의 잔고는 $k+1$ (p의 확률로), 또는 $k-1$ (q의 확률로)이 됩니다. 따라서

점화식: $D_k = p\ D_{k+1} + q\ D_{k-1} + 1$

초기조건: $D_0 = 0,\ \ D_n = 0$

을 얻습니다. 이 점화식 풀이는 약간 어렵습니다만 한번 해볼 만 합니다. 여기서는 유도과정을 생략하고 결과만 제시합니다.

경우 1: $p = 0.5$인 경우.

$$D_k = k\,(n-k), \qquad \text{for } k = 0, 1, \cdots, n.$$

경우 2: $p \neq 0.5$인 경우.

$$D_k = \frac{k}{q-p} - \frac{n}{q-p} \cdot \frac{1-(q/p)^k}{1-(q/p)^n}, \quad \text{for } k = 0, 1, \cdots, n.$$

예컨대 $p = 0.5$, $k = 5$인 경우에서, $n = 10$이면 게임 수 기대값은 25입니다. 그러나 공정게임에서는 $\lim_{n \to \infty} D_k = \infty$입니다. 그러므로 카지노 입장에서는 공정게임을 주관할 리가 없습니다. 묵은 손님을 내보내고 새 손님을 받아야 장사가 잘 될 테니까요.

$p < 0.5$인 게임에서는 $\lim_{n \to \infty} D_k = \dfrac{k}{q-p} < \infty$입니다. ■

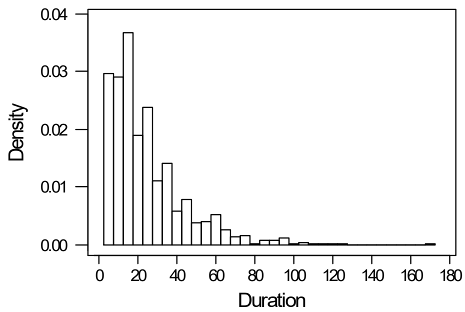
게임 수 T는 확률변수입니다. k 달러를 갖고 시작한 도박꾼에게 목표가 n 달러인 경우에서 T는 최소 $\min(k, n-k)$이고 최대 ∞입니다. 앞에서는 T의 기대값을 구하였을 뿐, T의 확률분포를 구한 것은 아닙니다. 그것을 구할 수 있을까요? 수학적으로는 어려워 보입니다. 몬테칼로는 어떨까요?

<표 1> 도박꾼의 파산 문제를 위한 몬테칼로 프로그램

```
/* Game Duration Time for Gambler's Ruin Problem */
/* ruin.iml                                      */

proc iml;
    p = 18/38;     q = 1-p;
    k = 5;         n = 10 ;
    Nrepeat=1000;
    S = j(Nrepeat,1,0);    /* Value 1 = Ruin, Value 0 = Success */
    T = j(Nrepeat,1,0);    /* Real Value := Game Duration Time  */
    do repeat=1 to Nrepeat;
       position = k;
       time = 0;
       do while ((position > 0) & (position < n));
          if uniform(0) < p then position = position+1;
          else position = position-1;
          time = time + 1;
       end;
       if position = 0 then S[repeat] = 1;
       T[repeat] = time;
    end;
    mean1 = S[+]/Nrepeat;
    mean2 = T[+]/Nrepeat;
    print Nrepeat mean1 mean2;
    print S T;
quit;
```

<표 1>이 게임 수의 확률분포를 구하기 위한 SAS/IML 프로그램입니다. 룰렛 게임처럼 도박꾼이 이길 확률이 $p = 18/38$이고 도박꾼이 $k = 5$달러로 시작하되 그의 목표가 $n = 10$인 경우에 대하여 몬테칼로 모의시행한 결과 <그림 2>의 결과가 나왔습니다 (반복수 Nrepeat = 1,000). 참고로 모의시행과 이론적인 평균값을 비교해보겠습니다. 아래 표를 보세요. 상당히 일치하지 않습니까? '컴퓨터의 힘'을 다시 한 번 느낄 수 있습니다.

	파산 확률	게임수 평균
모의시행	0.632	24.7
이 론	0.629	24.5



<그림 2> 게임 수 분포 (몬테칼로 시행결과)

3.2 쿠폰 수집 (Coupon Collection)

n종의 쿠폰(카드)을 모두 모으기 위해서는 모두 몇 장의 쿠폰을 사야 할까요? 예컨대 '러・리・브・스・토'라고 카드에 임의로 한 글자씩 적힌 카드가 수백장 들어있는 박스가 있다고 합시다. 몇 장을 까야 '러브스토리' 5장이 모두 모아질까요?

<u>쿠폰 수집</u>

모두 n종의 쿠폰을 모아야 한다고 합시다. 첫 번째 쿠폰은 어느 것이어도 '성공'입니다. 그 다음이 문제인데 두 번째 쿠폰은 첫 번째 수집된 쿠폰과 다른 종류여야 합니다. 따라서 두 번째 쿠폰을 얻기까지의 실패 수 X_2는 성공의 확률이 $(n-1)/n$인 기하분포를 따릅니다.

그리고, 세 번째 쿠폰은 첫 번째와 두 번째 수집된 쿠폰과 다른 종류여야 합니다. 그러므로 두 번째 쿠폰을 얻은 후 세 번째 쿠폰을 얻기까지의 실패 수 X_3는 성공의 확률이 $(n-2)/n$인 기하분포를 따릅니다.

이런 식으로, 마지막 쿠폰은 이제까지 수집된 $n-1$종의 쿠폰과 다른 종류여야 합니다. 그러므로 $n-1$번째 쿠폰을 얻은 후 마지막 쿠폰을 얻기까지의 실패 수 X_n은 성공의 확률이 $1/n$인 기하분포를 따릅니다.

결국, n종의 쿠폰(카드)을 모두 모으기 위해서

$$S_n \equiv 1 + (1 + X_2) + (1 + X_3) + \cdots + (1 + X_n)$$

장의 쿠폰을 사야 하는데, 여기서 X_i 는 Geometric$(1-(i-1)/n)$ 분포를 따르는 독립적인 확률변수입니다 $(i = 2, 3, \cdots, n)$. 따라서

$$E\{S_n\} = 1 + \sum_{i=2}^{n} \{1 + E(X_i)\} = \sum_{i=1}^{n} \left\{1 + \frac{1-(n-i+1)/n}{(n-i+1)/n}\right\}$$

$$= \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i} \qquad (2)$$

로 표현됩니다. 예컨대 $n = 5$ 인 경우

$$E\{\mathrm{S}_5\} = 5\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}\right) = 11\frac{5}{12}$$

입니다. ■

기대값 (2)를 $n = 365$ 에 대하여 계산해볼까요? (생일이 각기 다른 365명의 사람들을 집합시키기 위해서는 모두 몇 명에게 생일을 물어보아야 할까요?) 불가능하지는 않아도 손가락으로 계산하기에는 상당히 어려운 계산이 될 것입니다. 그러나 근사값은 다음과 같이 쉽게 계산할 수 있습니다.

$$\sum_{i=1}^{n} \frac{1}{i} \le 1 + \sum_{i=2}^{n} \int_{i-1}^{i} \frac{1}{x}\, dx = 1 + \log_e n,$$

$$\sum_{i=1}^{n} \frac{1}{i} \ge \sum_{i=1}^{n} \int_{i-1}^{i} \frac{1}{x+1}\, dx = \log_e (n+1).$$

그런데 $\lim_{n \to \infty} \frac{1 + \log_e n}{\log_e (n+1)} = 1$ 이므로

$$E\{S_n\} \sim n \cdot \frac{1 + \log_e n + \log_e (n+1)}{2} \qquad (3)$$

가 되는데, 더 간략히 표현하면

$$E\{S_n\} \sim n \log_e n, \quad \text{큰 } n \text{에 대하여}$$

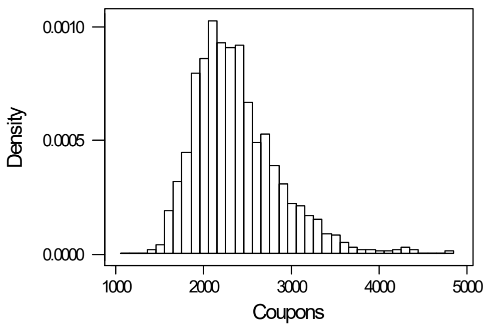
입니다. 수치예로서, $n = 365$ 에 대하여 정확한 기대값 (2)는 2364.65 인데 이에 대한 근사값인 $n \log_e n$ 은 2153.46 이고 (3)의 근사값은 2336.46 입니다.

참고로 $\sum_{i=1}^{n} \frac{1}{i} \simeq \log_e n + \gamma$ ($\gamma = 0.57\cdots$ 는 오일러 상수)입니다.

지금까지 쿠폰 수 S_n 의 기대값을 구해보았습니다만, S_n 의 확률분포를 직접 구할 수 있을까요? 수학적 접근은 조금 어려워 보이는군요. 그러므로 몬테칼로 방법을 적용해봅시다. <표 2>가 이를 위한 SAS/IML 프로그램입니다. 1,000번의 모의시행 결과 평균이 2367.01 로 나왔고 분포는 <그림 3>과 같이 나왔습니다.

<표 2> 쿠폰 수집 문제를 위한 몬테칼로 프로그램

```
/* Coupon Collection - How many coupons do we need to buy? */
/* coupon.iml                                              */

proc iml;
   n = 365;
   Nrepeat = 1000;
   Xrecord = j(Nrepeat,1,0);
   sum = 0;
   do repeat=1 to Nrepeat;
      sum0=0;
      do coupon=2 to n;
          theta=1-(coupon-1)/n;
          success = 0;
          do while (success = 0);
              if uniform(0) < theta then success=1;
              sum0 = sum0 + 1- success;
          end;
      x = n + sum0;
      Xrecord[repeat] = x;
      end;
      sum = sum + x;
   end;
   mean = sum/Nrepeat;
   print Nrepeat n mean;
   print Xrecord;
quit;
```



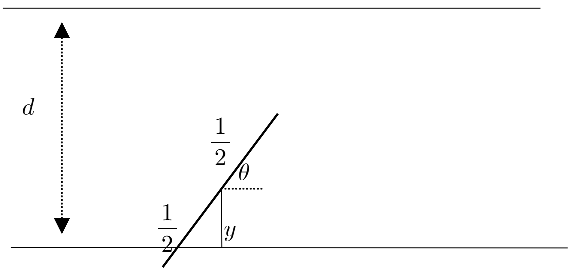
<그림 3> 쿠폰 수 분포 ($n = 365$, 몬테칼로 시행결과)

3.3 뷔퐁의 바늘 (Buffon's Needle)

18세기 프랑스 학자인 뷔퐁(Buffon)이 만들어낸 바늘 문제를 풀어봅시다. 이 문제에서 흥미로운 요소는 원주율 π의 추정 문제와 관련이 있다는 사실입니다.

<u>뷔퐁의 바늘</u>

<그림 4>와 같이 동일한 간격으로 평행선 금이 무수하게 나있는 마루 바닥에 바늘을 던집니다. 그러면 어떤 때에는 바늘이 금에 다 있기도 하고 어떤 때에는 금에 다 있지 않기도 하겠지요.



<그림 4> 뷔퐁의 바늘

바늘의 길이를 1이라고 하겠습니다. 이것은 금 간격 d와의 상대적으로만 관계되므로 그렇게 가정하여도 일반성을 해치지 않을 것입니다. 단, $d \ge 1$이라고 가정하겠습니다. 그렇게 가정함으로써 바늘이 두 금에 걸리는 경우를 원천적으로 배제할 수 있겠지요. 여기서 임의적 현상을 다음 두 변수로 기술하겠습니다.

Θ = 바늘이 금의 기선과 이루는 각 $(0 \le \Theta \le \pi)$,

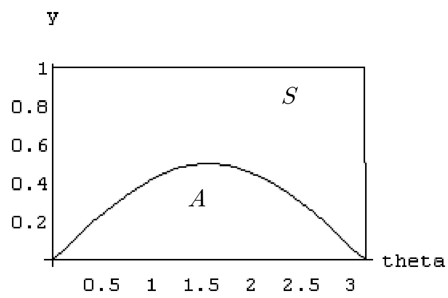
Y = 바늘의 중점에서 가장 가까운 금까지의 거리 $(0 \le Y \le d/2)$.

바늘이 임의로 던져지는 경우, 이 두 변수는 독립적으로 Uniform$(0,\pi)$ 분포와 Uniform$(0,d/2)$ 분포를 따른다고 할 수 있습니다. 따라서 전체공간은

$$S = \{(\theta, y) : 0 \le \theta \le \pi, 0 \le y \le d/2\}$$

가 되겠고 바늘이 금에 걸리는 사건은

$$A = \left\{(\theta, y) : 0 \le y \le \frac{1}{2}\sin\theta\right\}$$



<그림 5> 전체공간 S와 사건 A의 도시(圖示) : $d = 2$

로 나타내집니다 (여기서 $\frac{1}{2}$은 바늘의 절반 길이입니다). 그러므로 바늘이 금에 걸릴 확률이

$$P\{A\} = \frac{\text{area}(A)}{\text{area}(S)}$$

로 표현됩니다. 그런데 $\text{area}(S) = \frac{d}{2}\pi$이고

$$\text{area}(A) = \int_0^{\pi}\int_0^{\sin\theta/2} dy\, d\theta = [-\cos\theta/2]_0^{\pi} = 1$$

이므로

$$P\{A\} = \frac{2}{d\,\pi} \tag{4}$$

입니다. 금 사이 간격 d가 클 수록 바늘이 금에 걸칠 확률이 작아지는 것은 당연합니다. ■

원주율 π에 관한 수학사는 인류의 역사와 더불어 왔다고 해도 과언이 아닙니다. 이에 대한 논문은 무수히 많고 우리말로 나온 책도 한 권 있을 정도이니까요? π 값을 알고자 하는 목적으로 앞의 식 (4)를 활용해보면 어떨까요? 즉 다음과 같은 베르누이 시행 $Z_1, \cdots, Z_n$을 실제 해보는 것입니다.

1) 길이가 1인 바늘을 평행한 금 사이 간격이 $d \geq 1$인 바닥에 던져 바늘이 금에 걸리면 1, 그렇지 않으면 0으로 기록합니다.
2) 그런 장난을 n번 반복합니다. 그 결과로 $z_1, \cdots, z_n (= 0,1)$을 얻었다고 합시다. 그리고 그 합계를 $s\,(= \sum_{i=1}^{n} z_i)$라고 합시다.

따라서 $P\{A\}$는 s/n으로 추정됩니다. 그러므로

$$\frac{s}{n} \simeq \frac{2}{d\pi}$$

의 관계가 성립하게 됩니다. 이로부터 π에 대한 근사값

$$\pi \simeq \frac{2}{d} / \frac{s}{n} = \frac{2}{dp}, \text{ 여기서 } p = \frac{s}{n} : \text{표본비율}$$

을 얻게 됩니다. 이제 이론이 세워졌으니 실행을 하여야 하지 않겠습니까? 실제실험을 해본다면 말할 나위 없이 좋겠습니다만 여기서는 모의시행(simulation)으로 대신해 보겠습니다. 그 단계는 다음과 같아야 할 것입니다.

1) θ를 Uniform$(0, \pi)$ 분포로부터, y를 Uniform$(0, d/2)$ 분포로부터 독립적으로 만듭니다.

2) $y \le \frac{1}{2}\sin\theta$ 이면 z를 1로, 그렇지 않으면 z를 0으로 기록합니다.

3) 앞의 두 단계를 n번 반복하여 p를 산출합니다.

4) p로부터 π를 추정합니다.

그러나 이 알고리즘은 내재된 모순을 갖고 있습니다. 원주율 π를 추정하는 것이 목적인데, 단계 1에서 θ를 Uniform$(0,\pi)$ 분포로부터 만들어낼 수 있겠습니까? 그럼 포기해야 할까요? 쉽게 그럴 수는 없지요. 단계 1을 다음과 같이 바꾸어 봅니다.

1*) θ를 Uniform$(0,4)$ 분포로부터, y를 Uniform$(0, d/2)$ 분포로부터 독립적으로 만듭니다. 이 때, $\frac{1}{2}\sin\theta \ge 0$ 이여야 다음 단계로 넘어갑니다.

여기서 π를 4로 대치한 이유는 (담배 피우던 호랑이도 알고 있던 사실인) $\pi \le 4$이기 때문이고 $\frac{1}{2}\sin\theta \ge 0$ 이어야 한다고 조건을 단 이유는 $\frac{1}{2}\sin\theta$가 실제 길이여야 의미가 있기 때문입니다. <그림 4>를 보세요.

<표 3>이 몬테칼로 프로그램입니다 ($d = 2$). 모의시행의 결과는 다음과 같습니다 (N은 단계 1*가 수행된 회수이고 n이 실제로 단계 2까지 간 회수입니다).

반복수 N	유효 반복수 n	비율 $p(=s/n)$	π의 추정치
1,000	781	0.3073	3.3542
10,000	7,831	0.3152	3.1730
100,000	78,569	0.3207	3.1186
1,000,000	785,464	0.3177	3.1478

<표 3> 뷔퐁의 바늘 문제 풀이를 위한 몬테칼로 프로그램

```
/* Buffon's Needle Simulation */
/* needle.iml                 */

proc iml;
    d = 2;
    Nrepeat = 10000;
    pi = 4;
    count = 0;
    success = 0;
    do repeat=1 to Nrepeat;
        theta = uniform(0)*pi;
        y = uniform(0)*d/2;
        temp = sin(theta);
        if temp >= 0 then count = count + 1;
        if y <= 0.5*temp then success = success + 1;
    end;
    p = success/count;
    estimate = 2/(d*p);
    print Nrepeat count success p[format=12.4] estimate[format=12.4];
end;
```

반복수 N이 1,000,000, 유효 반복수 n이 785,000 정도가 되어야 π 값이 우리가 현재 알고 있는 $3.14\cdots$ 정도가 되는군요. 이제 끝을 낼까요? 하나만 더 하기로 합시다. 소위 연구설계(research design)의 문제인데, 앞에서 금 간격 d를 2로 잡았지만 다르게 잡을 수도 있지 않겠습니까? (추정이론에 관하여 전혀 예비지식이 없는 학생들은 다음 기회에 읽기 바랍니다.)

실제시행이든 아니면 모의시행이든 간에 그것을 베르누이 변수들인 $Z_1, \cdots, Z_n$으로 표현하겠습니다. 이들은 Bernoulli$\left(\frac{2}{d}\cdot\frac{1}{\pi}\right)$ 분포로부터의 iid 확률변수들입니다. 원래 문제는 π를 추정하는 것이지만, 논의를 간결하게 하기 위하여 $\frac{1}{\pi}\,(=\eta)$를 추정하는 문제로 바꾸어 생각합시다. 그러면,

$$\hat{\eta} = \frac{\frac{S}{n}}{\frac{2}{d}}\ , \quad \text{여기서}\ S = \sum_{i=1}^{n} Z_i$$

입니다. 따라서

$$Var\ \hat{\eta} = \left(\frac{d}{2}\right)^2 \frac{2}{d\pi}\left(1-\frac{2}{d\pi}\right)\Big/ n = \frac{1}{\pi n}\left(\frac{d}{2}-\frac{1}{\pi}\right)$$

인데, 이것은 d의 단조증가함수입니다. 그러므로 $\hat{\eta}$의 분산은 d가 가장 작을 때 최소화됩니다. 이 문제의 기술에서 금 간격 d는 $d \geqq 1$이여야 한다고 했으므로 η를 추정하기 위하여 가장 좋은 d는 1입니다.

직관적으로는, $\eta\,(=\pi^{-1})$가 잘 추정되면 π도 잘 추정될 것이므로 π의 추정에서도 역시 $d=1$이 좋다고 할 수 있습니다만, 이에 관한 본격적인 논의는 이제까지보다 약간 복잡하므로 여기서는 생략하기로 하겠습니다 (연습문제 3.5).

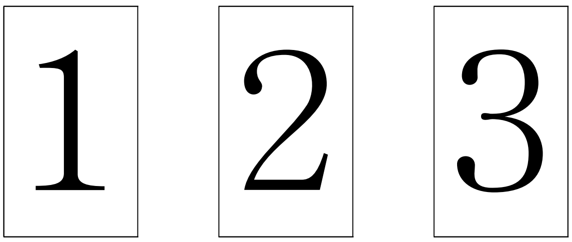
다음은 $d=1$로 수행한 모의시행의 결과입니다. 앞의 결과와 비교해 보십시오.

반복수 N	유효 반복수 n	비율 $p\,(=s/n)$	π의 추정치
1,000	764	0.6113	3.2716
10,000	7,853	0.6443	3.1040
100,000	78,436	0.6393	3.1284
1,000,000	785,178	0.6364	3.1429

3.4 바꿀까? 말까? (Let's Make a Deal)

이 문제는 'Let's Make a Deal'이라는 TV 쇼에서 나왔습니다. 'Monty Hall' 문제라고도 하지요. 쇼의 내용은 다음과 같습니다.

0) 출연자가 1, 2, 3의 숫자가 적힌 세 개의 방문 앞에 섭니다. 그 중 1개 방에는 큰 경품이 있습니다만 다른 두 방에는 당나귀 그림뿐입니다. 물론 출연자는 어느 방에 경품이 있는지 전혀 모릅니다.



1) 출연자에게 1개 문을 선택하게 합니다. 그러나 열어볼 수는 없습니다.
2) 사회자가 남은 두 개의 문 중에서 1개 문을 엽니다.
3) 이제 마지막으로, 출연자는 사회자가 보여준 방을 제외한 나머지 2개 방 중에서 한 방의 문을 최종 선택합니다. 그는 마음을 바꿀 수도 있고 바꾸지 않을 수도 있습니다.

문제는 단계 3에서 출연자가 처음 선택하였던 문을 여는 것이 확률적으로 좋은가, 아니면 마음을 바꿔 다른 문을 선택하는 것이 좋은가 하는 것입니다. 여러분이 출연자라면 어떻게 하겠습니까? 경품이 스포츠 카라고 생각하고 신중하게 결정하십시오.

마음을 바꾸는 것이 좋습니까? 초지일관(初志一貫)이 낫습니까? 아니면, 어떻게 하든지 마찬가지입니까?

사실, 이 문제에 대하여는 많은 논란이 있었다고 합니다. 수학자들도 각자 다른 주장을 하였다니까요. 이런 혼란이 생긴 이유는 사회자가 어떤 방법으로 남은 두 개의 문 중에서 1개 문을 여는가에 대하여 단계 2에 명확히 기술되어 있지 않았기 때문입니다. 여기서는 다음 규칙을 따르는 것으로 하겠습니다.

사회자 규칙 : 사회자는 남은 두 개의 문 중에서 경품이 들어 있지 않은 방의 문을 엽니다. 그러한 방이 2개이면 그 중 1개의 방문을 선택합니다.

따라서, 출연자가 처음 선택한 방이 가짜인 경우에는 사회자는 선택의 여지가 없이 남은 2개의 문 중에서 가짜 방의 문을 열게 됩니다. 그러나 출연자가 처음 선택한 방이 진짜인 경우에는 사회자에게는 남은 2개의 문 중에서 1개 문을 임의로 선택하여 엽니다. 이제 답을 말하기로 하지요.

마음을 바꾸는 것이 좋습니다. 그러면 성공확률은 2/3입니다. 초지일관하는 경우엔 성공확률이 1/3입니다.

이상하다고요? 잘 생각해보세요. 돈이 걸려 있는 이런 문제에 대하여는 공동의 공간을 써가면서까지 친절하게 설명하고 싶지 않군요. 대신 몬테칼로 방법을 제시하기로 합니다. 직접 경험해 보는 것이 더 확실하지 않겠습니까?

<표 4>의 프로그램에서 변수들의 정의는 다음과 같습니다.

U = 경품이 들어 있는 방의 문 번호,
X = 출연자가 처음 선택한 문 번호,
V = 사회자가 연 문의 번호,
Y = 출연자가 최종 선택한 문 번호.

<표 4> 'Let's Make a Deal' 문제 풀이를 위한 몬테칼로 프로그램

```
/* Let's Make a Deal                                       */
/* rule=1 for "change rule", rule = 0 for "no change rule" */
/* lmad.iml                                                */

proc iml;
   rule = 1;
   Nrepeat = 1000;
   sum = 0;
   do repeat=1 to Nrepeat;
      t1 = uniform(0);
      if t1 <= 1/3 then u = 1;
         else if t1 <= 2/3 then u = 2;
            else u = 3;
      t2 = uniform(0);
      if t2 <= 1/3 then x = 1;
         else if t2 <= 2/3 then x = 2;
            else x = 3;
      if x ^= u then do;
         v = mod(x,3)+1;
         if v = u then v = mod(u,3)+1;
      end;
      if x = u then do;
         t3 = uniform(0);
         if t3 <= 1/2 then v = mod(u,3)+1;
         else v = mod(u+1,3)+1;
      end;
      if rule=0 then y = x;
      if rule=1 then do;
         y = mod(x,3)+1;
         if y = v then y = mod(v,3)+1;
      end;
      if u = y then success = 1;  else success = 0;
      sum = sum + success;
   /* print u x v y success; */
   end;
   p = sum / Nrepeat;
   print rule Nrepeat sum p;
quit;
```

<표 4>의 프로그램에서 'rule=0'은 마음을 바꾸지 않는 결정규칙(no change rule)이고 'rule=1'은 마음을 바꾸는 결정규칙(change rule)입니다. 마음을 바꾸지 않는 결정규칙으로 1000번을 모의시행하였더니 성공률이 0.339, 마음을 바꾸는 결정규칙으로 1000번을 모의시행한 결과로는 성공률이 0.689가 나오는군요. 그러므로 몬테칼로 시행을 통하여도 마음을 바꾸는 결정규칙이 마음을 바꾸지 않는 결정규칙보다 훨씬 좋다는 것을 알 수 있습니다.

<표 4>와 사실상 동일하지만 웹(World Wide Web)에 'Let's Make a Deal'이 자바 애플렛(Java Applet)으로 구현되어 있습니다. 만든 이는 미국 South Carolina 대학의 R. Webster West 교수입니다. WWW 주소는 다음과 같습니다.

http://www.amstat.org/publications/jse/v6n3/applets/LetsMakeaDeal.html

또는

http://www.stat.sc.edu/~west/javahtml/LetsMakeaDeal.html

여러분도 그곳을 방문해보길 바랍니다. 이 자바 애플렛에서는 <그림 6>과 같은 동적 그래픽이 뜨기 때문에 실감 있게 게임을 할 수 있습니다. 물론 각 규칙에 따른 결과도 자동으로 기록이 되어 나옵니다.

<그림 6> 자바 애플렛 Let's Make a Deal

3.5 산불 (Forest Fire)

마지막으로, 산불이 번지는 과정을 확률모형으로 만들어 보겠습니다. 숲을 정사각형 그리드(grid)라고 하고 그리드의 각 정수 점(i_1, i_2)에 나무가 한 그루씩 있다고 하겠습니다. 그리고 정 중앙에서 발화(發火)하는 것으로 하지요. 즉

$$S = \{(i_1, i_2) \mid -n \le i_1 \le n, -n \le i_2 \le n\}$$

의 원점$(0,0)$으로부터 불이 붙기 시작한다고 전제합니다 (0시에). 불이 번져가는 과정에 대하여 다음과 같은 가정을 하겠습니다.

1) 시각 t에 불이 붙은 나무는 다음 시각 $t+1$에 불에 타 없어진다.
2) (i_1, i_2)에 있는 나무가 시각 t에 불이 붙으면, 그 다음 시각에

$$(i_1+1, i_2), (i_1-1, i_2), (i_1, i_2+1), (i_1, i_2-1)$$

   에 각각 p의 확률로, 그리고 독립적으로 불이 옮겨 붙는다. 물론, 이미 불에 타 없어진 나무에는 다시 불이 붙지 않는다.

가정은 단 두가지로 간단하지만 불이 번지는 경로는 생각보다 훨씬 복잡합니다. 예를 들어$(1,0)$에 불이 붙는 경우의 보기는 다음과 같습니다.

$$\begin{tikzcd}
(0,1) & (1,1) \\
(0,0) \arrow[r] & (1,0)
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
(0,1) \arrow[r] & (1,1) \arrow[d] \\
(0,0) \arrow[u] & (1,0)
\end{tikzcd}$$

time1 　　　　　　　　　　 time3

이 두 가지 경우 외에도 수많은 경우가 있습니다. 그러므로 이 문제에도 역시 순수 수학적 접근은 매우 어려워 보입니다. 그러나 그 과정의 알고리즘은 단순하기 때문에 몬테칼로 모의시행은 쉽습니다.

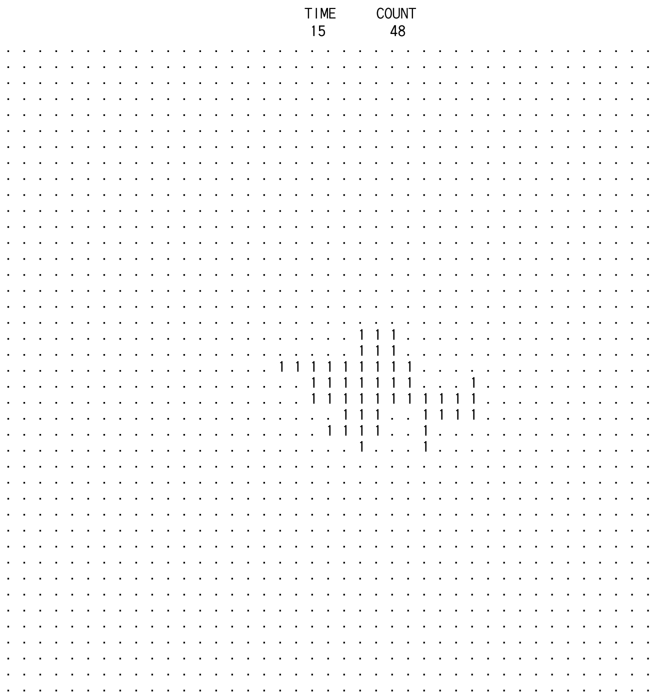
<표 5>의 프로그램은 $n=20$인 경우 즉, 41×41 그리드의 숲에서 정중앙에서 발화된 산불이 어떻게 번지는가를 보여줄 것입니다. 확률 p는 0.5로 하였습니다.

산불이 어떤 자국을 남겼는지 <그림 7>을 보세요. (출력에서 TIME은 불이 지속된 시간이고 COUNT는 불에 탄 나무의 수입니다.) 예측하기 어려운 복잡한 양상들을 보여주고 있습니다. 그 중 첫 번째 것(A1)과 네 번째 것(A4)은 자그맣게 되었습니다만 두 번째 것(A2)과 세 번째 것(A3)은 불이 상당히 많이 번진 결과를 보여줍니다. 그러나 A2와 A3에서도 불이 타지 않은 곳이 곳곳에 남아 있어 종자 보존이 가능하다는 것을 알 수 있습니다. 생태적 관점에서는 산불이 나쁘지 않다고 합니다.

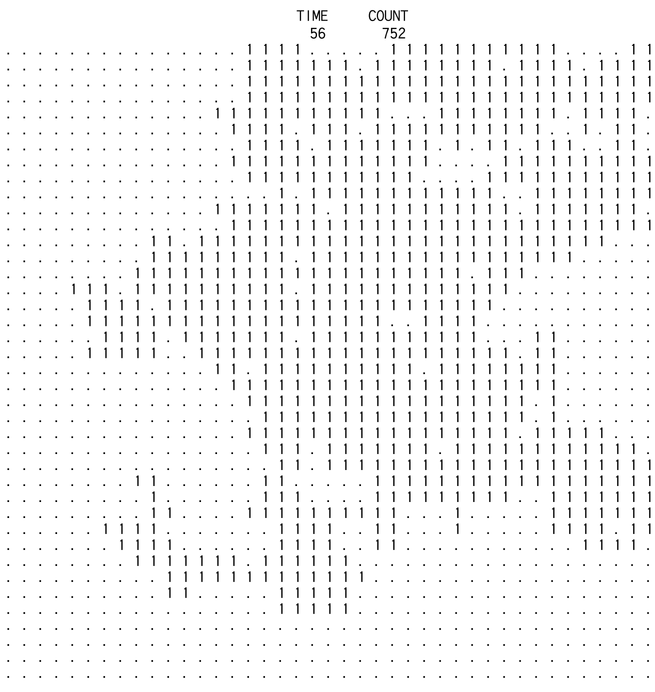
<표 5> 산불과정을 보기 위한 몬테칼로 프로그램

```
/* Forest Fire Process Simulation */
/* fire.iml                       */

proc iml;
    p = 0.5;  n=20;
    nn = 2*n+1;
    FIRE1 = j(nn,nn,'.');
    FIRE2 = j(nn,nn,'.');
    MAP = j(nn,nn,'.');
    status = 1;
    FIRE1[n+1,n+1] = '0';
    time = 0;

    do while (status > 0);
        status = 0;
        do i1 = 1 to nn;
        do i2 = 1 to nn;
          if FIRE1[i1,i2] = '0' then do;
              if (i1 < nn) then do;
                 if (uniform(0) < p) & (MAP[i1+1,i2] = '.') then
                    do;  FIRE2[i1+1,i2] = '0';  status = 1;  end;  end;
              if (i1 > 1)  then do;
                 if (uniform(0) < p) & (MAP[i1-1,i2] = '.') then
                    do;  FIRE2[i1-1,i2] = '0';  status = 1;  end;  end;
              if (i2 < nn) then do;
                 if (uniform(0) < p) & (MAP[i1,i2+1] = '.') then
                    do;  FIRE2[i1,i2+1] = '0';  status = 1;  end;  end;
              if (i2 > 1)  then do;
                 if (uniform(0) < p) & (MAP[i1,i2-1] = '.') then
                    do;  FIRE2[i1,i2-1] = '0';  status = 1;  end;  end;
              MAP[i1,i2] = '1';
           end;
        end;
        end;
        FIRE1 = FIRE2;
        FIRE2 = j(nn,nn,'.');
        time = time + 1;
    /*  print time MAP[format=1.0];    */
    /*  print time FIRE1[format=1.0];  */
        end;

    count=0;
    do i1 = 1 to nn;
    do i2 = 1 to nn;
        if MAP[i1,i2] = '1' then count=count+1;
    end;  end;
    print time count, MAP[format=1.0];
quit;
```

A1

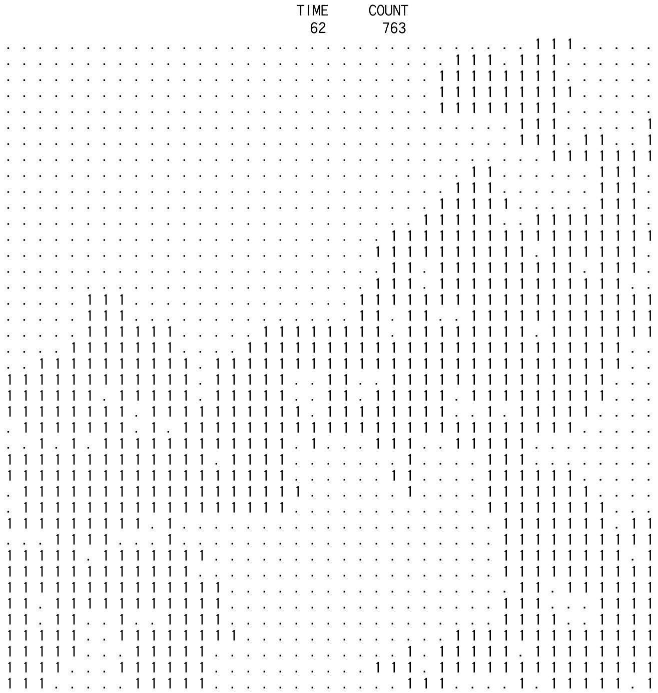


A2

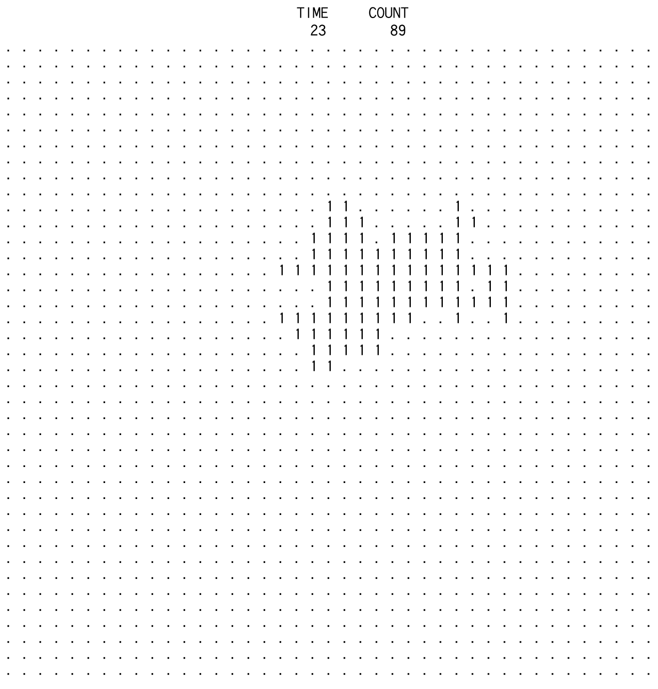


<그림 7> 산불이 꺼진 후 (1은 불에 탄 나무, ㆍ는 불에 타지 않은 나무)

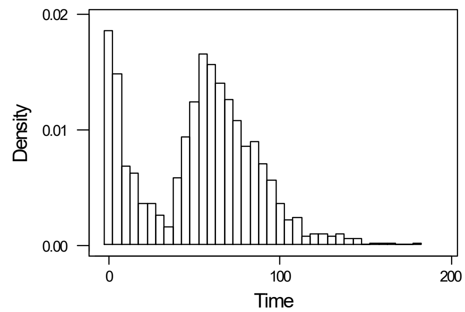
A3



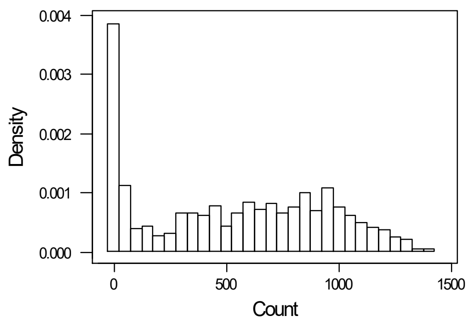
A4



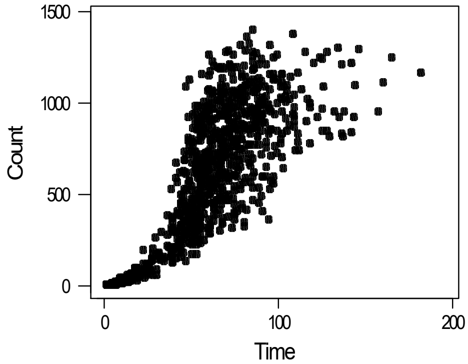
<그림 7> 산불이 꺼진 후 (계속)



<그림 8> 산불 시간의 분포



<그림 9> 불에 탄 나무수 분포



<그림 10> 산포도(scatterplot)

산불 시간(TIME)과 불에 탄 나무 수(COUNT)의 분포도 흥미롭습니다. <그림 8>과 <그림 9>를 보세요. 특히 산불시간에 대하여는 명백하게 2개의 봉우리를 가진 이봉분포(二峰分布, bimodal distribution)를 볼 수 있습니다. <그림 10>의 산포도는 이들 두 변수간의 관계를 보여줍니다. 100 시간을 넘어가서는 시간이 길어져도 불에 탄 나무시간이 더 이상 증가하지 않는 것을 볼 수 있는데 이것은 불에 탈 수 있는 자원이 상당히 소진되었기 때문으로 생각됩니다. 즉 잔불만 남은 것이지요.

※ <그림 8, 9, 10>은 <표 5>의 프로그램을 1,000번 반복수행한 결과를 토대로 만든 것입니다. 그렇게 하기 위해서는 <표 5>에 반복 수행문을 삽입해 넣어야 합니다. 그리고 마지막 프린트 문을 꺼 놓는 것이 좋습니다. (연습문제 3.8)

※ <표 5>의 프로그램에서 루프 안에 있는 프린트 문을 살리면 MAP은 시각 t 현재 불에 탄 숲의 모습을, FIRE1은 시각 t 현재 불에 타고 있는 나무들을 계속 보여줍니다. 전이 확률 p를 1로 하면 불이 마름모 꼴로 번지는 것을 볼 수 있습니다.

※ <표 5>의 프로그램은 매우 비효율적으로 짜여 있습니다. 여러분들이 손을 봐서 프로그램을 개선시켜 보세요.

산불 과정은 산불에 뿐만 아니라 사회과학으로도 응용가능하다고 생각됩니다. 가령 유언비어에 대하여 그렇습니다. 유언비어가 생산되고 그 주변에 퍼지는 과정을 생각해 봅시다. 산불과 유언비어의 공통점과 차이점은 무엇입니까? 차이점이 있다면 그것을 추가로 모형화하면 될 것입니다 (탐구문제).

이제 이 장을 마칩니다. 핵심을 요약하자면, 확률문제를 이해하는 데 수학적 분석과 몬테칼로 모의시행이 모두 중요하다는 것입니다. 서로가 상대방에 대한 대안적 역할을 한다고 말할 수 있습니다.

3.A 연습문제

3.1 카지노를 상대로 A가 5불을 갖고 도박을 합니다. 그의 목표는 10불입니다. B도 A와 마찬가지입니다. 그런데 A는 한 게임에 1불이 왔다 갔다 하는 게임을 하고 B는 한 게임에 0.1불이 걸린 게임을 합니다. 게임당 이길 확률 p가 0.5인 경우와 18/38인 경우에서 A와 B의 파산확률을 비교해보세요. 어느 쪽이 더 낫습니까? 평균 게임수에서는 어떤 차이가 있습니까?

3.2 n종의 카드를 모두 모으기까지 필요한 카드수 S_n의 평균은 이미 3.2절에서 구했습니다. 이 번에는 S_n의 분산을 구해보세요. 그리고

$$\lim_{n\to\infty} n^{-2}\ Var(S_n) = \frac{\pi^2}{6}, \quad 즉 \quad Var(S_n) \sim \frac{\pi^2}{6}n^2$$

임을 보이세요. 더불어, $n = 365$인 경우에 $Var(S_n)$의 정확한 값, 수학적 근사 값, 몬테칼로 근사값을 구하여 비교해보세요. $\left[참고\ :\ \sum_{k=1}^{\infty}\frac{1}{k^2} = \frac{\pi^2}{6} \right]$

3.3 뷔퐁의 바늘 문제에 대한 변형입니다. 수평방향과 수직방향으로 무수한 평행선 금이 있는 바닥에 길이 1인 바늘을 던집니다. 이 때 수평 평행선 사이의 간격과 수직 평행선 사이의 간격은 $d \geq 1$로 같습니다. 따라서 타일바닥을 생각하면 됩니다. 바늘이 수직선 또는 수평선 금에 걸리는 수 $X(=0,1,2)$에 대한 기대값을 구해보세요 [힌트: 쉽게 생각하십시오].

3.4* 앞의 문제에서, 어떻게 X의 확률분포를 구할 수 있는지 생각해봅시다.

3.5 뷔퐁의 바늘 문제에서 π에 대한 추정량 $\hat{\pi} = \dfrac{2}{d\,p}$의 분산을 근사적으로 구하세요. d와 p의 정의는 3.2절의 본문에서와 같습니다.

힌트: 델타방법을 활용합니다. 답: $\dfrac{\pi^3}{n}\left(\dfrac{d}{2} - \dfrac{1}{\pi}\right)$.

3.6* 앞의 문제에서, 몬테칼로 방법을 활용하여 π의 추정값을 구하고 그것의 분산까지 구하는 방법에 대하여도 생각해봅시다.

3.7 Let's Make a Deal 문제에서 새로운 사회자 규칙이 “남은 두 개의 문 중에서 사회자가 임의의 1개의 방문을 선택합니다”라고 합시다. 즉, 사회자가 경품을 갖게 될 가능성을 허용한다고 합시다. 그런 경우에도 출연자는 마지막 단계에서 마음을 바꾸는 것이 유리합니까? 아니면 그렇지 않습니까?

3.8 전이확률 p를 0.4로 잡고 산불 과정에 대한 모의시행 프로그램을 1000번 반복하여 산불 시간(TIME)과 불에 탄 나무 수(COUNT)의 분포를 구하세요. 그리고 두 변수의 산포도를 그려보세요.

탐구문제

정사각형 정수 그리드(grid)

$$S = \{ (i_1, i_2) \mid -n \le i_1 \le n, -n \le i_2 \le n \}$$

의 정중앙인(0,0) 으로부터 시작된 유언비어(流言蜚語)의 확산과정에 대하여 생각하여 봅시다.

3.B 읽을만한 책

도박꾼의 파산, 쿠폰 수집, 뷔퐁의 바늘 문제에 대하여는 다음 책을 보십시오.

▪ Ross, S. (1998) *A First Course in Probability*, Fifth Edition. Prentice Hall.

“바꿀까? 말까?”(Monty Hall Problem)와 산불(Forest Fire)에 대하여는 다음 책을 보십시오. 이외에도 그 책에는 여러 재미있는 확률문제들이 여러 개 있습니다.

▪ Siegrist, K.T. (1996) *Interactive Probability*. Duxbury. (Sections 7 and 13)