

## 금융공학프로그래밍 II

2015. 12. 11

### 1. Binomial Tree 함수 (20pts)

- 함수원형: `std::vector<double*> binomialTree(double s, double u, int steps);`
- s에서 시작하고 1 step 마다 상승률이 u (>1) 하락률이 1/u (<1) 인 **이항트리** 리턴
- steps = 이항트리의 스텝수
- vector의 n번째 원소 = n번째 스텝 (time n)의 값들로 구성된 double\* 배열

### 2. Series 클래스 (30pts)

- series의 **평균**과 **표준편차**를 계산하는 클래스
- 생성자: `Series(std::vector<double> data)`
- 멤버변수: `std::vector<double> data_`
- 멤버함수: ① `double mean()`, ② `double stdev()`

$$\text{mean} = \frac{1}{N} \sum_{i=1}^N x_i \quad \text{stdev} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \text{mean})^2}$$

### 3. Dist / Binomial / Poisson 클래스 (50pts)

- 이산확률분포 (이항분포와 포아송분포)의 **pdf**와 **cdf**를 계산하는 클래스

#### Dist (반드시 추상클래스)

- 생성자: `Dist(std::string name)`
- 멤버변수: `std::string name_`
- 멤버함수: ① `double pdf(int x) = 0` (확률밀도함수), ② `double cdf(int x)` (누적확률분포 함수), ③ `int factorial(int n)`, ④ `std::string getName()`

$$\text{cdf}(x) = \sum_{i=0}^x \text{pdf}(i)$$
$$\text{factorial}(n) = n! = n \times (n-1) \times (n-2) \times \dots \times 1$$

#### Binomial (Dist 에서 상속받아 구현)

- 생성자: `Binomial(int n, double p)` (name\_ 변수를 "Binomial" 로 초기화)
- 멤버함수: `double pdf(int x)` (오버라이딩 함수)
- 추가 멤버변수: `int n_, double p_`

$$\text{pdf}(x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

#### Poisson (Dist 에서 상속받아 구현)

- 생성자: `Poisson(double m)` (name\_ 변수를 "Poisson" 으로 초기화)
- 멤버함수: `double pdf(int x)` (오버라이딩 함수)
- 추가 멤버변수: `double m_`

$$\text{pdf}(x) = \frac{m^x \times e^{-m}}{x!}$$

※ Binomial과 Poisson 클래스는 생성자와 pdf 외의 다른 함수를 포함하지 말 것

## 출력결과

```
C:\Windows\system32\cmd.exe

<1>
64
128      32
256      64      16
512      128      32      8
1024     256      64      16      4
2048     512      128      32      8      2

<2>
mean = 5.5
stdev = 3.02765

<3>
      Binomial      Poisson
-----
0      0.028248      0.028248      0.049787      0.049787
1      0.121061      0.149308      0.149361      0.199148
2      0.233474      0.382783      0.224042      0.423190
3      0.266828      0.649611      0.224042      0.647232
4      0.200121      0.849732      0.168031      0.815263
5      0.102919      0.952651      0.100819      0.916082
6      0.036757      0.989408      0.050409      0.966491
7      0.009002      0.998410      0.021604      0.988095
8      0.001447      0.999856      0.008102      0.996197
9      0.000138      0.999994      0.002701      0.998898
10     0.000006      1.000000      0.000810      0.999708

계속하려면 아무 키나 누르십시오 . . .
```