

# *Monte Carlo Simulation*

*Problem Solving Using Randomness*

# randomness

- 의미

- 패턴이나, 예측 가능성, 규칙성이 없음.
- 예> 주사위 굴리기, 난수 발생, 로또 추첨

# 몬테카를로 시뮬레이션

- *randomness*에 기반한 시뮬레이션을 설계
- 시뮬레이션을 반복 수행함으로써 특정 이벤트 발생 확률을 근사적으로 계산
  - 충분한 시도를 해야 함.
- 수학적으로 계산이 어려운 문제도 해결 가능
  - 포커 게임
  - 생일이 같은 확률
  - PI (아날로그를 디지털로 전환)
  - ...

# 문제 1

- 주사위를 굴렀을 때 1이 나올 확률은?

- 방법 1: 수학적으로 확률을 계산할 수 있다.

- 이 경우에는 간단함. 하지만 계산이 어려운 문제들도 있음.

- 방법 2: 직접 주사위를 굴리면서 (1이 나온 회수) / (전체 시도 회수) 근사값을 찾을 수도 있다.

- 문제는 정확성.
    - 정확성을 높이려면 일단 *randomness*가 전제되어야 함.
    - 또한 많이 시도해야 함. 단 시간이 오래 걸림.
    - 컴퓨터를 이용하면 많은 시도를 하는 것이 가능함.

# import

- 다른 파이썬 모듈(파일)에 정의된 라이브러리를 사용할 때 사용
  - `import random`
    - 랜덤 관련 라이브러리 함수들을 사용하기 위함
  - `random.randint()`
    - `random` 모듈의 `randint()` 함수 호출

# random 모듈

- `randint(from, to)`

- `from`부터 `to`까지의 정수들 중에서 임의의 정수를 반환

- `randint(1, 6)`

- 1~6 사이의 정수 반환

- `random()`

- 0 ~ 1 사이의 임의의 실수를 반환

# 문제 /

- 파이썬 코드

```
import random
```

```
def rollDice():
```

```
    ...
```

```
def calDiceProb(tries, target) :
```

```
    ...
```

```
print(calDiceProb(100, 1))
```

```
print(calDiceProb(100, 6))
```

## 문제2

- 주사위를 두 개 굴렀을 때 동일한 숫자가 나올 확률은?
  - 방법1: 수학적으로 확률을 계산할 수 있다.
    - $6/36$
  - 방법2: 직접 주사위를 굴리면서 근사값을 찾을 수도 있다.



# 문제2

- 파이썬 코드

```
import random
```

```
...
```

```
def calDiceProb2(tries):
```

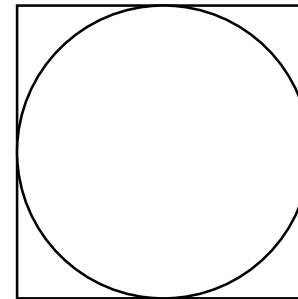
```
    ...
```

```
print(calDiceProb2(100))
```

```
print(calDiceProb2(1000))
```

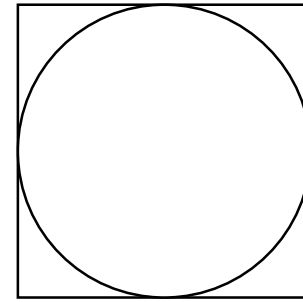
# 문제3

- 원주율 ( $\pi$ )를 계산하시오.
- 원주율
  - 한 변 2인 정사각형과 그 안에 내접한 반지름이 1인 원
  - 원의 넓이  $= \pi * r * r = \pi$
  - 정사각형 넓이  $= 2 * 2 = 4$
  - $\pi : 4 = \text{원의 넓이} : \text{정사각형 넓이}$
  - $\pi = (4 * \text{원의 넓이}) / \text{정사각형 넓이}$   
 $= 4 * (\text{원의 넓이} / \text{정사각형 넓이})$



# 문제3

- 넓이의 비를 어떻게 측정할까?
  - 넓이는 점들의 모임
  - 수많은 화살을 랜덤하게 쏘아서 사각형 안의 화살 개수 대비 원 안의 화살의 개수의 비를 구하면 됨.
  - 수많은 공을 랜덤하게 떨어 뜨려서 사각형 안의 공 개수 대비 원 안의 공의 개수의 비를 구하면 됨.

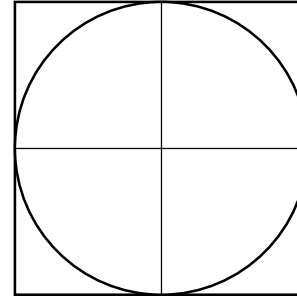


# 문제3

- 화살 쏘는 시뮬레이션?

- 화살의 좌표:

- $-1$ 과  $1$ 사이의 수들 중에서 임의의  $x, y$  를 뽑으면 화살의 좌표가 됨.
    - 또는  $0$ 과  $1$ 사이의 수들 중에서 임의의  $x, y$  를 뽑으면 화살의 좌표가 됨.



- 넓이의 비

- 화살이 원 안에 있는 회수 / 전체 회수
  - 원 안에 있는 지의 여부는 원점에서 거리가  $1$ 이하이면 됨.

# 문제3

- 파이썬 코드

## 문제4 (knapsack)

- 각각 무게가  $weight_i$ , 가치가  $val_i$ 인  $n$ 개의 물건들을 배낭에 넣으려고 한다. 배낭에 넣을 수 있는 최대 무게는  $W$ 이다. 배낭에 넣은 물건들의 가치를 최대로 하는 물건의 조합을 구하시오.
- 아이디어
  - 각 아이템의 포함 여부를 임의로 결정 (using random 함수)
  - 가치와 무게를 계산하여 무게가  $W$  이하이면 기존의 최대 가치와 비교하여 최대치를 기억.
  - 위 작업을 충분히 반복한 후 최대 가치와 해당 경우 반환

# 문제4 (knapsack)

- 파이썬 코드

```
import random
...
class Knapsack(object):
    ...
    def findBestCaseMontecarlo(self, tries=10000):
        ...
```

# 문제4 (knapsack)

- 파이썬 코드

```
names = ['0', '1', '2', '3', '4', '5', '6', '7']  
values = [10, 30, 20, 14, 23, 11, 15, 18]  
weights = [5, 8, 3, 7, 9, 2, 6, 1]
```

```
max_weight = 20
```

```
knapsack = Knapsack(names, values, weights, max_weight)
```

```
(chosen_items, max_value) = knapsack.findBestCaseMontecarlo(100)  
print(chosen_items, max_value)
```



# 문제5

- 52장의 카드에서 임의로 5장의 카드를 뽑을 때 플러시일 확률은?
  - 어려워졌지만, 수학적으로 확률을 계산할 수 있다.
    - $(4 * {}_{13}C_5) / {}_{52}C_5 = 0.002$ .
  - 또는 직접 카드를 뽑으며 근사값을 찾을 수도 있다.
    - 시뮬레이션 모델링이 필요함
- 52장의 카드에서 임의로 7장의 카드를 뽑을 때 플러시일 확률은?
- 52장의 카드로 3명이 포커 (5 또는 7)를 할 때, 플러시일 확률은?

# 문제5

- 파이썬 코드 (1단계)
  - `card` 클래스 정의: 한 장의 카드를 모델링 (무늬, 숫자)
  - `player` 클래스 정의: 여러 장의 카드를 갖는다 (`card`의 리스트)

```
import random
```

```
class Card:
```

```
    def __init__(self, kind, number):  
        self.kind = kind  
        self.number = number
```

```
class Player:
```

```
    def __init__(self):  
        self.cards = []
```

# 문제5

- 파이썬 코드 (2단계)

- *porker* 클래스 정의 :

- 멤버변수: 카드들의 리스트, 플레이어들의 리스트, 플레이어의 수, 몇 장의 카드를 나누어 줄지 나타내는 변수 (5 or 7?)
    - 메서드: 생성자, 카드 생성, 카드 섞기, 플레이어 생성, 카드 출력, ...

# 문제5

- 파이썬 코드 (2단계)

- *poker* 클래스 정의 : 생성자 정의

```
class Poker:
    def __init__(self, playerCnt=1, distCardCnt=5):
        self.distCardCnt = distCardCnt
        self.playerCnt = playerCnt
        self.cards = []
        self.players = []
        self.generateCards()
        self.shuffleCards()
        self.createPlayers()
```

# 문제5

- 파이썬 코드 (2단계)

- poker 클래스 정의 : 카드 생성 메서드 `generateCards()` 정의

```
class Poker:
```

```
...
```

```
def generateCards(self):
```

```
    self.cards = []
```

```
    kinds = ['spade', 'heart', 'diamond', 'clover']
```

```
    for i in range(4):
```

```
        for j in range(13):
```

```
            card = Card(kinds[i], j+1)
```

```
            self.cards.append(card)
```

```
    return
```

# 문제5

- 파이썬 코드 (2단계)

- *poker* 클래스 정의 : 카드 섞기 메서드 *shuffleCards()* 정의

```
class Poker:
```

```
...
```

```
def shuffleCards(self):  
    random.shuffle(self.cards)
```

# 문제5

- 파이썬 코드 (2단계)

- poker 클래스 정의 : 플레이어 생성 메서드 `createPlayers()` 정의

```
class Poker:
```

```
...
```

```
def createPlayers(self):
```

```
    for j in range(self.playerCnt):
```

```
        player = Player()
```

```
        self.players.append(player)
```

```
def printCards(self):
```

```
    for card in self.cards:
```

```
        print(card.kind + " " + str(card.number))
```

# 문제5

- 파이썬 코드 (2단계)

- poker 클래스 정의 : 현재 생성된 카드들을 출력하는 메서드 `printCards()` 정의

```
class Poker:
```

```
...
```

```
def createPlayers(self):
```

```
    for j in range(self.playerCnt):
```

```
        player = Player()
```

```
        self.players.append(player)
```

```
def printCards(self):
```

```
    for card in self.cards:
```

```
        print(card.kind + " " + str(card.number))
```



# 문제5

- 파이썬 코드 (2단계)

- poker 중간 실행 코드

```
playerCnt = 1
```

```
distCardCnt = 5
```

```
poker = Poker(playerCnt, distCardCnt)
```

```
poker.printCards()
```

# 문제5

- 파이썬 코드 (3단계)

- poker 클래스 정의 : 카드를 플레이어에게 배분하는 메서드 `playCards()` 정의

```
class Poker:
```

```
...
```

```
def playCards(self):
```

```
    for i in range(self.distCardCnt):
```

```
        for j in range(self.playerCnt):
```

```
            card = self.cards.pop()
```

```
            self.players[j].cards.append(card)
```

# 문제5

- 파이썬 코드 (3단계)

- poker 클래스 정의 : 각 플레이어에게 배분된 카드를 플레이어별로 출력하는 메서드  
printPlayerCards() 정의

```
class Poker:
```

```
...
```

```
def printPlayerCards(self):
```

```
    player_num = 1
```

```
    for player in self.players:
```

```
        print("\nplayer ", player_num, ":\n")
```

```
        player.printCards()
```

```
        player_num += 1
```

# 문제5

- 파이썬 코드 (3단계)

- *player* 클래스 정의 : 각 플레이어에게 배분된 카드를 출력하는 메서드 *printCards()* 정의

```
class Player:
```

```
...
```

```
def printCards(self):
```

```
    for card in self.cards:
```

```
        print(card.kind + " " + str(card.number))
```

# 문제5

- 파이썬 코드 (3단계)

- poker 중간 실행 코드2

```
playerCnt = 3  
distCardCnt = 5
```

```
poker = Poker(playerCnt, distCardCnt)  
#poker.printCards()  
poker.playCards()  
poker.printPlayerCards()
```

# 문제5

- 파이썬 코드 (4단계)

- poker 클래스 정의 : 플러시 계수 메서드 `countFlush()` 정의

```
class Poker:
```

```
...
```

```
def countFlush(self):
```

```
    count = 0
```

```
    for player in self.players:
```

```
        if (self.isFlush(player.cards) == True):
```

```
            count += 1
```

```
    return count
```

# 문제5

- 파이썬 코드 (4단계)

- poker 클래스 정의 : 플러시 판단 메서드 *isFlush()* 정의

```
class Poker:
```

```
...
```

```
def isFlush(self, targetCards):
```

```
    kindCnt = {'spade':0, 'heart':0, 'diamond':0, 'clover':0}
```

```
    for card in targetCards:
```

```
        kindCnt[card.kind] += 1
```

```
    for key in kindCnt.keys():
```

```
        if (kindCnt[key] >= 5):
```

```
            return True;
```

```
    return False
```

# 문제5

- 파이썬 코드 (4단계)

- poker 최종 실행 코드

```
playerCnt = 1
```

```
distCardCnt = 5
```

```
gameCnt = 10000
```

```
flushCnt = 0
```

```
for i in range(gameCnt):
```

```
    poker = Poker(playerCnt, distCardCnt)
```

```
    poker.playCards()
```

```
    flushCnt += poker.countFlush()
```

```
print(flushCnt)
```

```
print(flushCnt / (gameCnt * playerCnt))
```



# Summary

- Randomness를 활용한 문제 해결 기법

- 몬테카를로 시뮬레이션

- randomness 기반 시뮬레이션을 설계하여 특정 이벤트 발생 확률을 경험적으로 계산
    - 충분한 시도를 해야 함.
    - 수학적으로 계산이 어려운 문제도 해결 가능
      - PI (아날로그를 디지털로 전환)
      - 포커 게임
      - 생일이 같은 확률
      - ...

# homework candidate

- $y = x * x$  을 0부터 1까지 적분한 값을 몬테카를로 시뮬레이션을 이용하여 계산하시오.



↳ 넓이 구하기

## homework candidate

- 포커 클래스에 스트레이트 계수 함수를 추가하고 확률을 구하시오.

## homework candidate

- 플레이어 5명이고 7 포커를 할 때, 한 번의 게임에서 플러시가 두 개 이상 나올 확률을 계산하시오.