Monte Carlo

Randomness

What types of problems can we solve with the help of random numbers?

We can compute (potentially) complicated averages:

1. Where does "the average" web surfer end up? (PageRank)
2. How much is my stock portfolio/option going to be worth?
3. What are my odds to win a certain competition?

Random number generators

- Computers are deterministic - operations are reproducible
- How do we get random numbers out of a determinist machine?

Demo “Playing around with random number generators”

- Pseudo-random numbers
    - Numbers and sequences appear random, but they are in fact reproducible
    - Good for algorithm development and debugging
- How truly random are the pseudo-random numbers?

Example: Linear congruential generator

$$x_o = seed$$

$$x_{n+1} = (a\,x_n + c\,)\,(mod\ M)$$

a: $multiplier$

c: $increment$

M: $modulus$

- If we keep generating numbers using this algorithm, will we eventually get the same number again? Can we define a period?

Demo “Random numbers”

Good random number generator

- Random pattern
- Long period
- Efficiency
- Repeatability
- Portability

Random variables

We can think of a random variable X as a function that maps the outcome of unpredictable (random) processes to numerical quantities.

Examples:

- How much rain are we getting tomorrow?
- Will my buttered bread land face-down?

random variable
$X = 80\%$

We don't have an exact number to represent these random processes, but we can get something that represents the **average** case.

To do that, we need to know how likely each individual value of X is.

Discrete random variables

Each random value X takes values x_i with probability p_i

for $i = 1, \dots, m$ and $\sum_{i=1}^{m} p_i = 1$

Example:

Random variable $\longrightarrow$ X = # top of die after each roll

Possible values x_i :

$x_1 = 1 \longrightarrow p_1 = 1/6$

$x_2 = 2 \longrightarrow p_2 = 1/6$

$\vdots$

$x_6 = 6 \longrightarrow p_6 = 1/6$

Coin toss example

Random variable X: result of a toss can be heads or tails

$x_1 =$ X =1: toss is heads $\longrightarrow p_1 = 0.5$

$x_2 =$ X = 0: toss is tail $\longrightarrow p_2 = 0.5$

Expected value : $E(x) = \sum_{i=1}^{m} p_i x_i$

Roll : $E(x) = \frac{1}{6}(1) + \frac{1}{6}(2) + \frac{1}{6}(3) + \ldots + \frac{1}{6}(6) = \frac{7}{2}$

Toss : $E(x) = \frac{1}{2}(1) + \frac{1}{2}(0) = 0.5$

Coin toss example

(*) expected value

increase M ⇒ better estimate for expected value

increase N ⇒ decrease variance

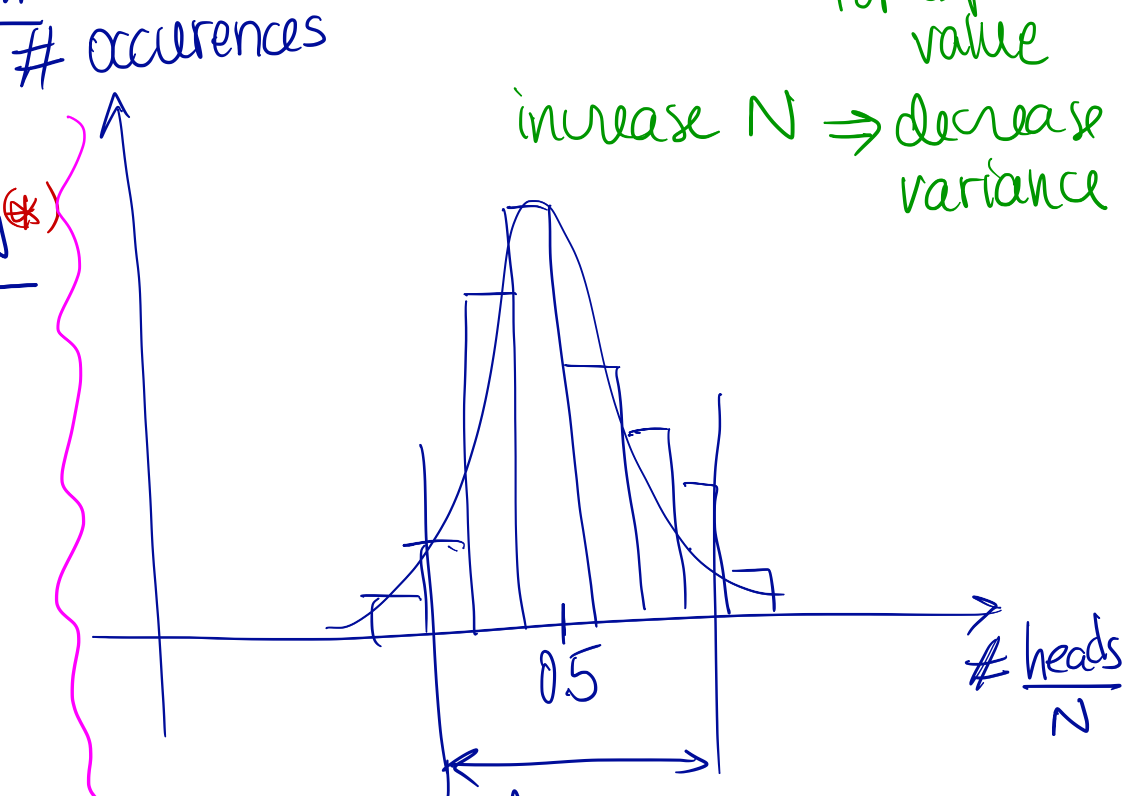
Numerical experiment:

$\boxed{N = 100}$ toss

	#heads	#heads/N (*)
1	45	0.45
2	52	0.52
3	54	0.54
⋮		
M	51	0.51

M = number of numerical experiments

LIVE DEMO



decrease variance

⇓

increase N

Texas Holdem Game

Question: for each starting pair of cards, what is the probability of winning?

- 1 Numerical experimental $\Longrightarrow$ play N games

- Game : set of 7 cards

$\Downarrow$

tie, win or loss

Starting hand

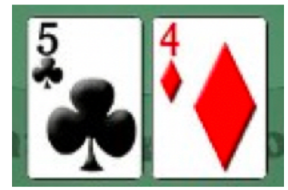
dealer hand

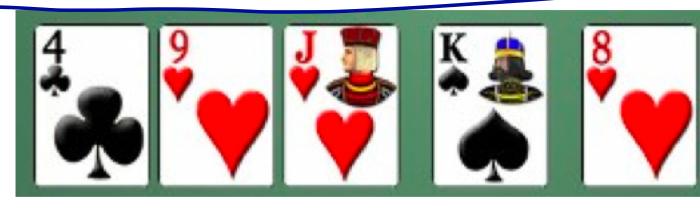
Opponent hand

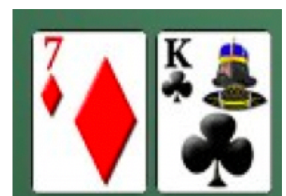
Texas Holdem Game

Question: for each starting pair of cards, what is the probability of winning?

→ Fixed

Starting hand (deterministic variable **S**): 

Dealer hand (random variable **D**): 

Opponent hand (random variable **O**): 

↳ each "game" generates these 7 cards at random

for i = 1, N (games)

generate D, O

who win(S, D, O) → use poker rules to decide who wins.

Texas Holdem Game

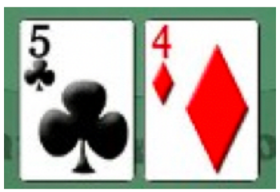
odd of start hand winning $= \dfrac{\#\text{time s win}}{N}$

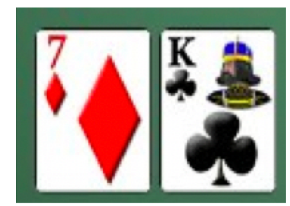
$$\boldsymbol{X} = Win(\boldsymbol{S}, \boldsymbol{O}, \boldsymbol{D})$$

$X = [1,0,0]$: starting hand wins

$X = [0,1,0]$: starting hand loses (opponent wins)

$X = [0,0,1]$: tie





Numerical experiment of $N=50$ games

game 1 $\longrightarrow$ $X = \text{WhoWin}(S,O,D) = [0,1,0]$

2 $\longrightarrow$ $X = [1,0,0]$

$\vdots$

$N \longrightarrow$ $X = [0,1,0]$

$\oplus$ [#time s win, #time O win, #time tie]

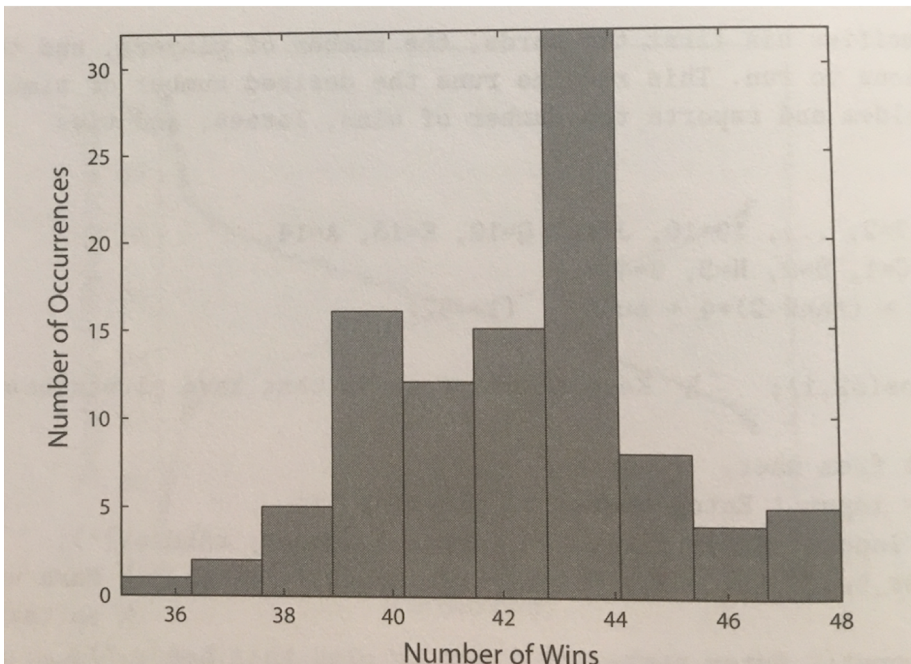
Texas Holdem Game

increase N → reduce variance

Starting hand: pair of aces

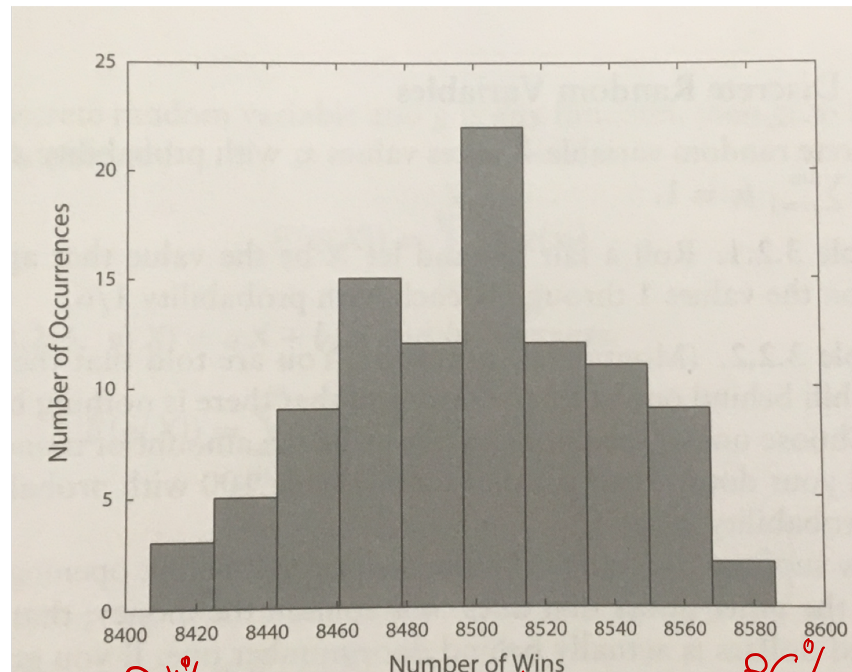
M =

Plotting the number of wins for 100 numerical experiments



72% 84% 96%

N = 50 games



84% 85% 86%

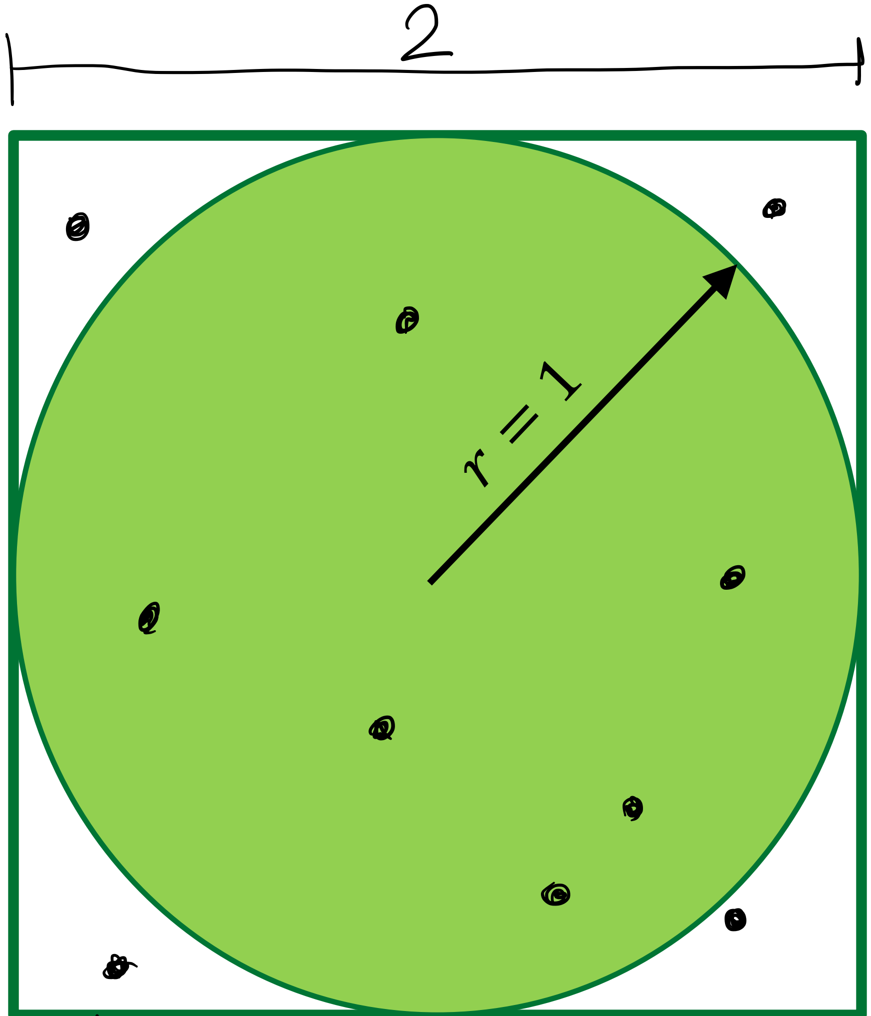
N = 10,000 games

Monte Carlo methods

- You just implemented an example of a Monte Carlo method!
- Algorithm that compute APPROXIMATIONS of desired quantities based on randomized sampling

→ Often used to approximate areas/volumes of complicated surfaces.

Example: Approximate the number π



1 numerical experiment:

- sample N points inside domain
- count # points that are inside circle $\rightarrow N_\circ$
- $A_\square \propto N_\square = N$
- $A_\circ \propto N_\circ$

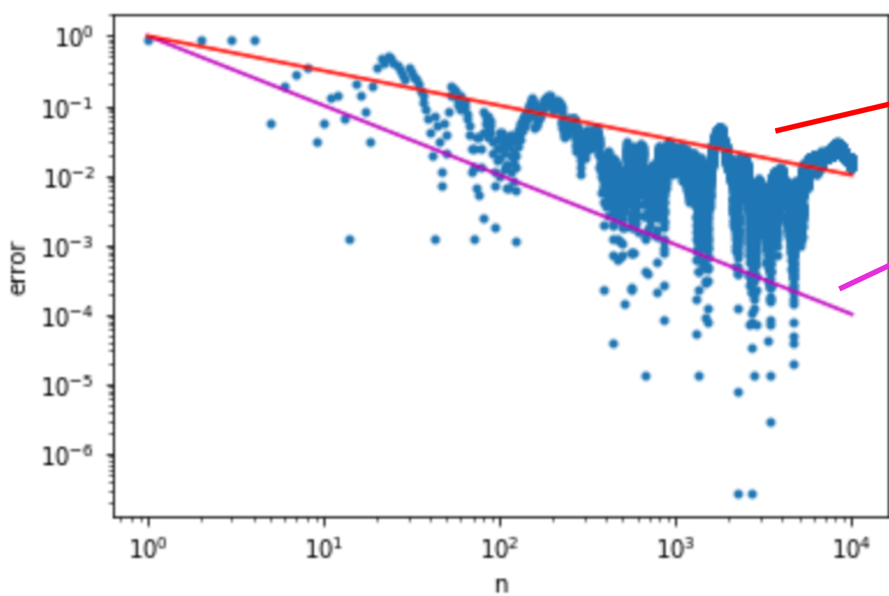
$$\pi r^2 = 4N_\circ / N$$

$$\frac{A_\square}{A_\circ} = \frac{N}{N_\circ} \Rightarrow A_\circ = \frac{N_\circ}{N} A_\square \Rightarrow \boxed{A_\circ = \frac{4N_\circ}{N}} \qquad \boxed{(\pi)_{approx} = \frac{4N_\circ}{N}}$$

What can we learn about this simple numerical experiment?

- What is the cost of this numerical experiment? What happens to the cost when we increase the number of sampling points (n)?

- Does the method converge? What is the error?



$$error = O\left(\frac{1}{\sqrt{n}}\right) = O(n^{-1/2})$$

$$error = O\left(\frac{1}{n}\right) = O(n^{-1})$$

- CONS: Slow convergence rate when using Monte Carlo Methods
- PROS: Efficiency does not degrade with increase in the dimension of the problem (try to modify the demo to approximate the area of an sphere)