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 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?

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.

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

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:

Coin toss example

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

X =1: toss is heads

X = 0: toss is tail

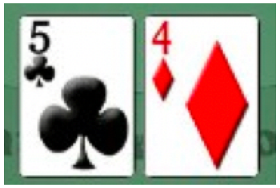
Coin toss example

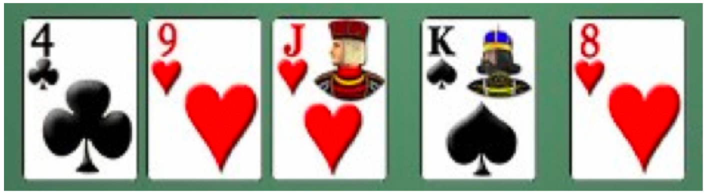
Texas Holdem Game

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

Texas Holdem Game

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

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

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

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

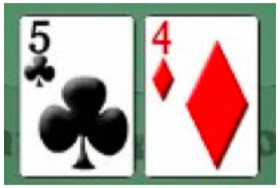
Texas Holdem Game

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

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

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

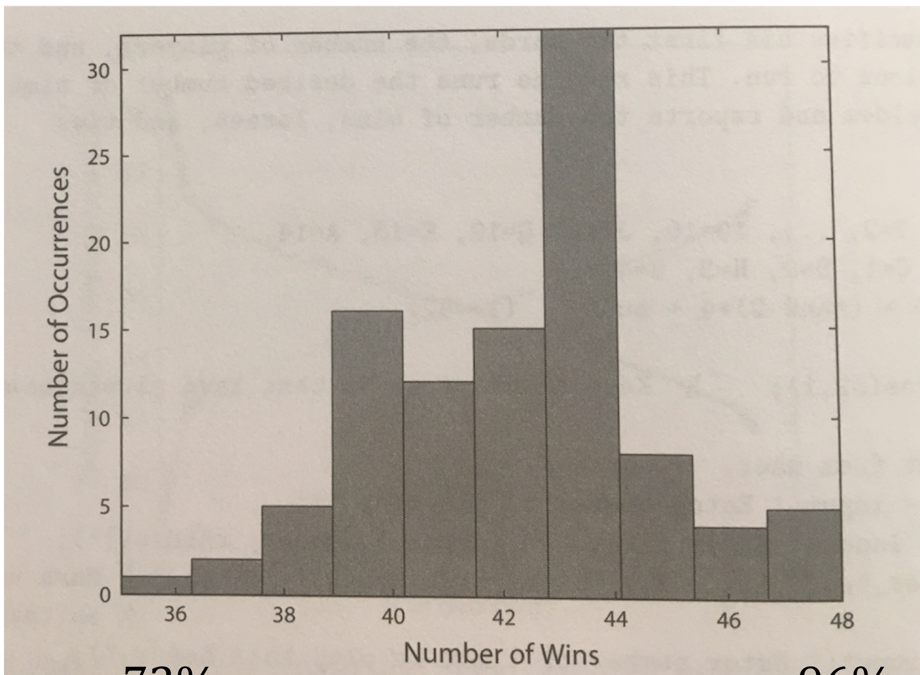
$X = [0,0,1]$: tie



Texas Holdem Game

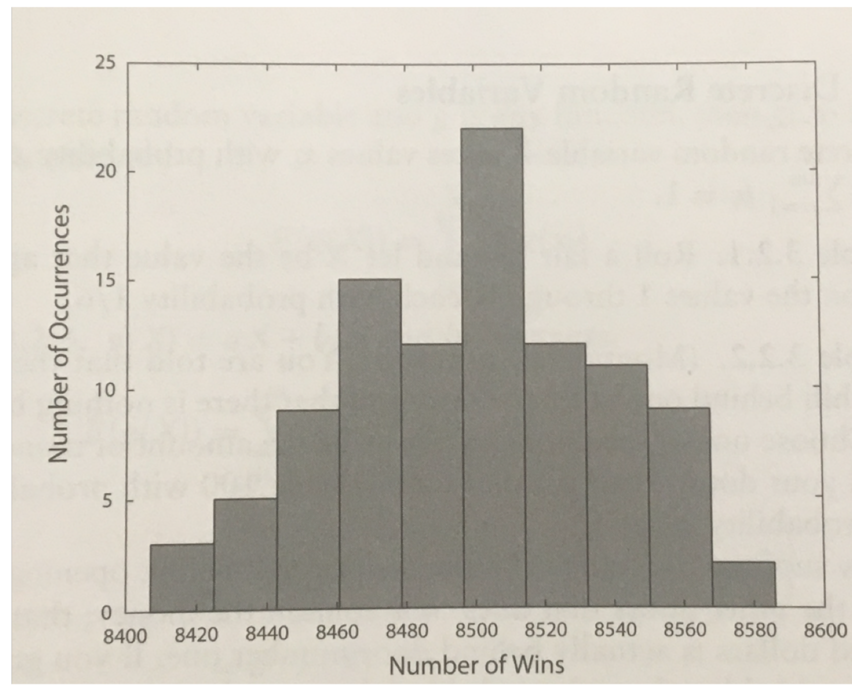
Starting hand: pair of aces

Plotting the number of wins for 100 numerical experiments



72% 84% 96%

50 games



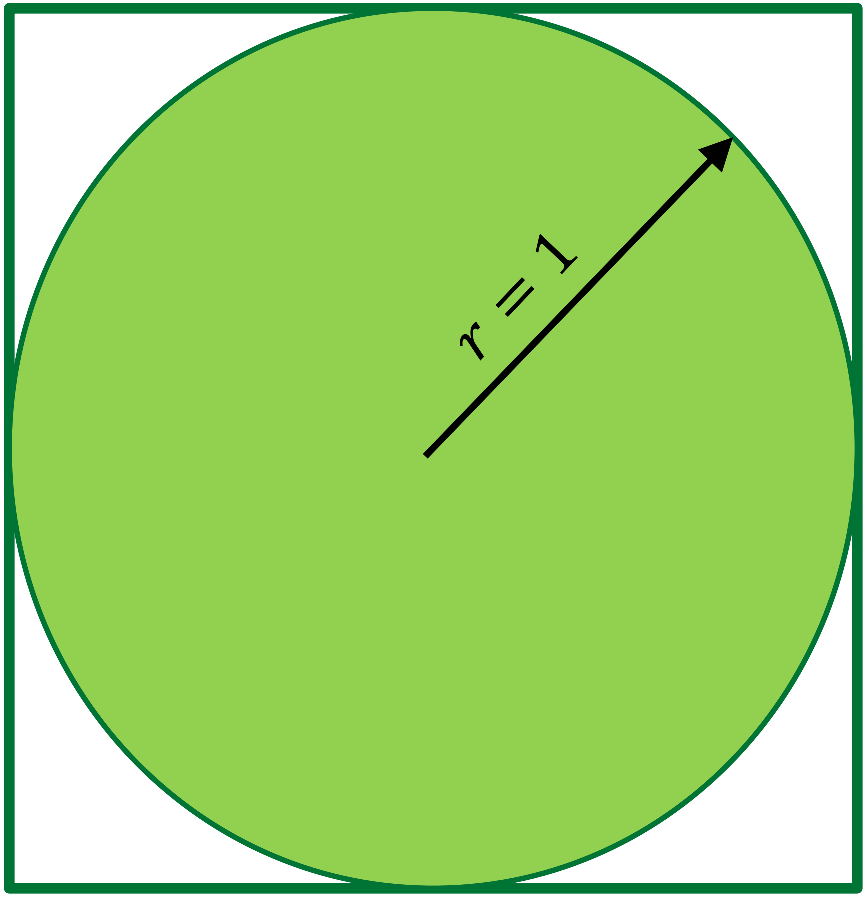
10,000 games

Monte Carlo methods

- You just implemented an example of a Monte Carlo method!

- Algorithm that compute APPROXIMATIONS of desired quantities based on randomized sampling

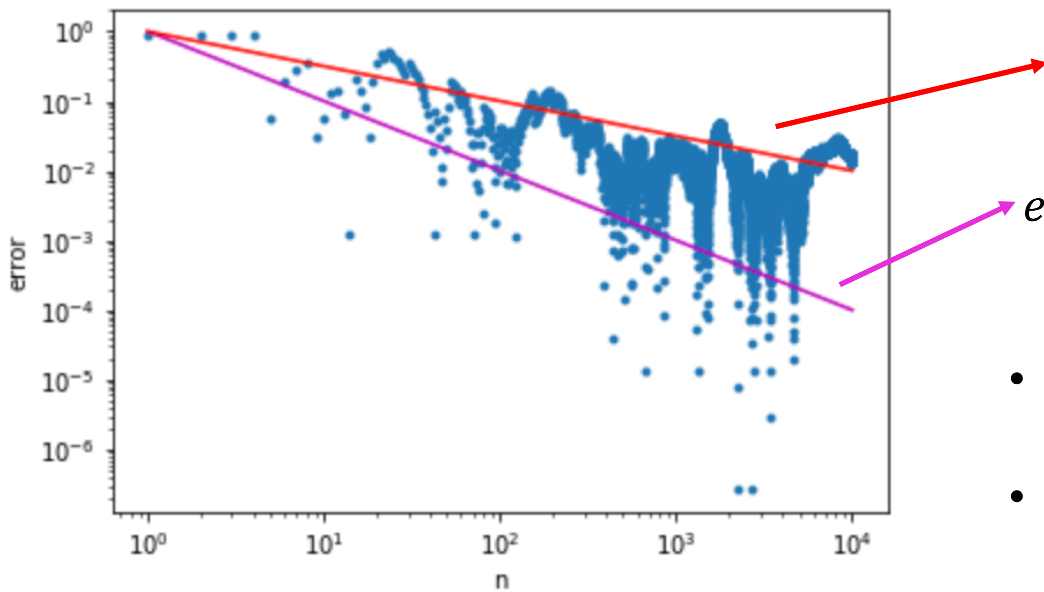
Example: Approximate the number π



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?



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

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

- 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)