CDS 230
Modeling and Simulation I

Module 9

Monte Carlo Simulation

Dr. Hamdi Kavak
http://www.hamdikavak.com
hkavak@gmu.edu

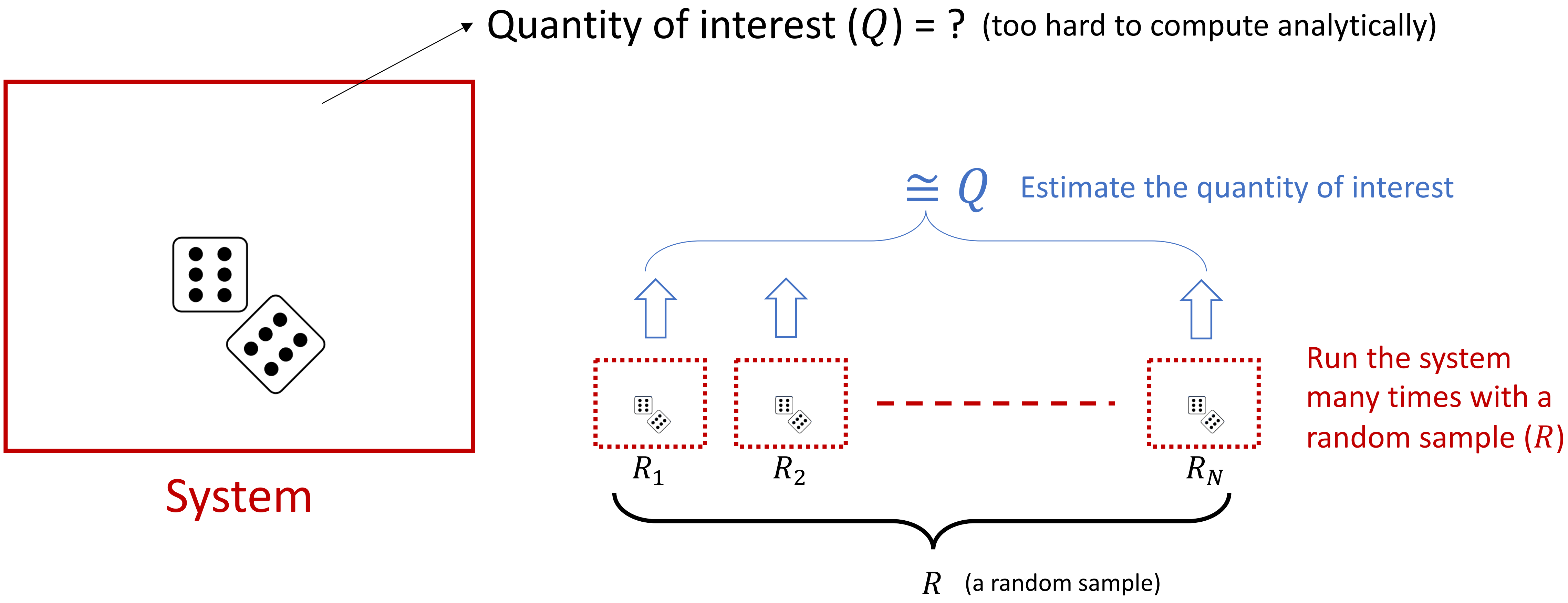
Monte Carlo simulation

- Named after the famous gambling complex in Monaco, France.

- Dates back to 1940s
- Very popular method used in many domains
  - Obviously highly used in gambling,
  - … but also in physical sciences, chemistry, computational biology, finance, business,… .
- Heavily used in data science and optimization

Source: Public Domain, https://commons.wikimedia.org/w/index.php?curid=334551

Monte Carlo simulation



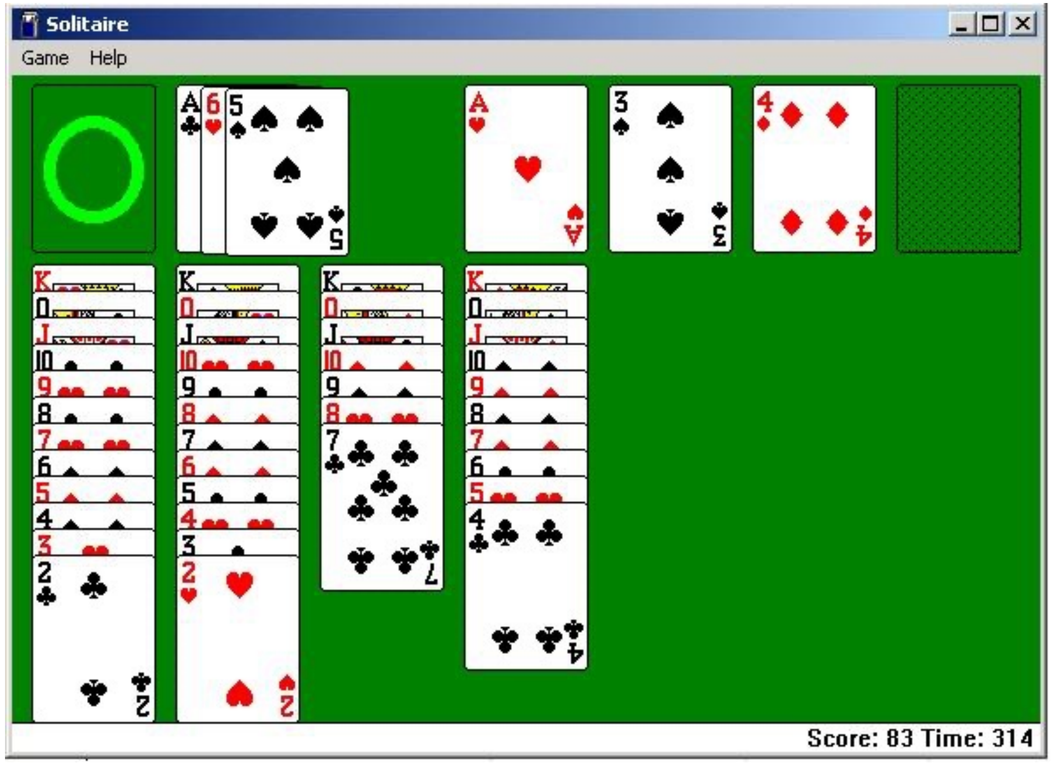
Monte Carlo simulation, in other words

- Is a computational method to estimate the value of a **quantity of interest** using a set of randomly generated numbers (random sample).
- Assume that the quantity of interest is very complex to compute analytically (e.g., via mathematical representation)
- It relies on the fact that a **sample of a population** behaves close enough to the one observed in the entire population.

Do you like Solitaire?

What is the chances to win the solitaire even before seeing the cards?

The deal 3 cards and 3 times around the deck option

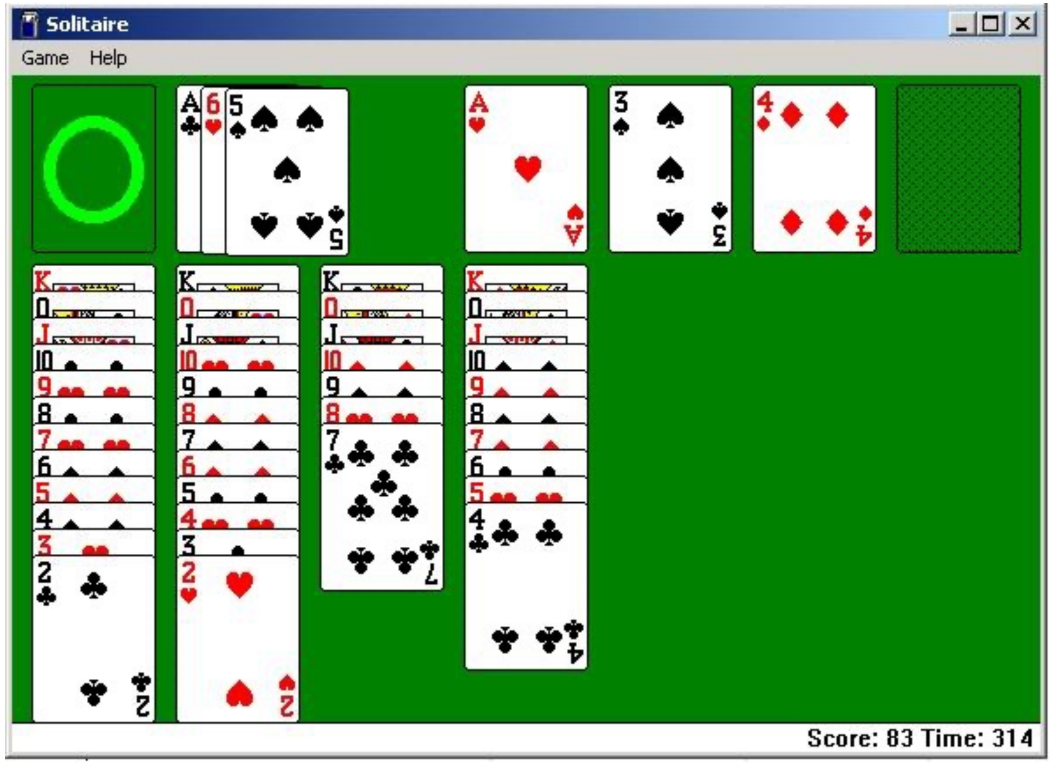


Source: https://www.flickr.com/photos/bobb/43826727

Do you like Solitaire?

What is the chances to win the solitaire even before seeing the cards?

The deal 3 cards and 3 times around the deck option



Source: https://www.flickr.com/photos/bobb/43826727

After playing 100,000,000 games, it is found that only 8.7% of cases win*.

*According to “Bill’s Solitare Tester” using the deal 3 cards and 3 times around the deck option

Coin flipping example

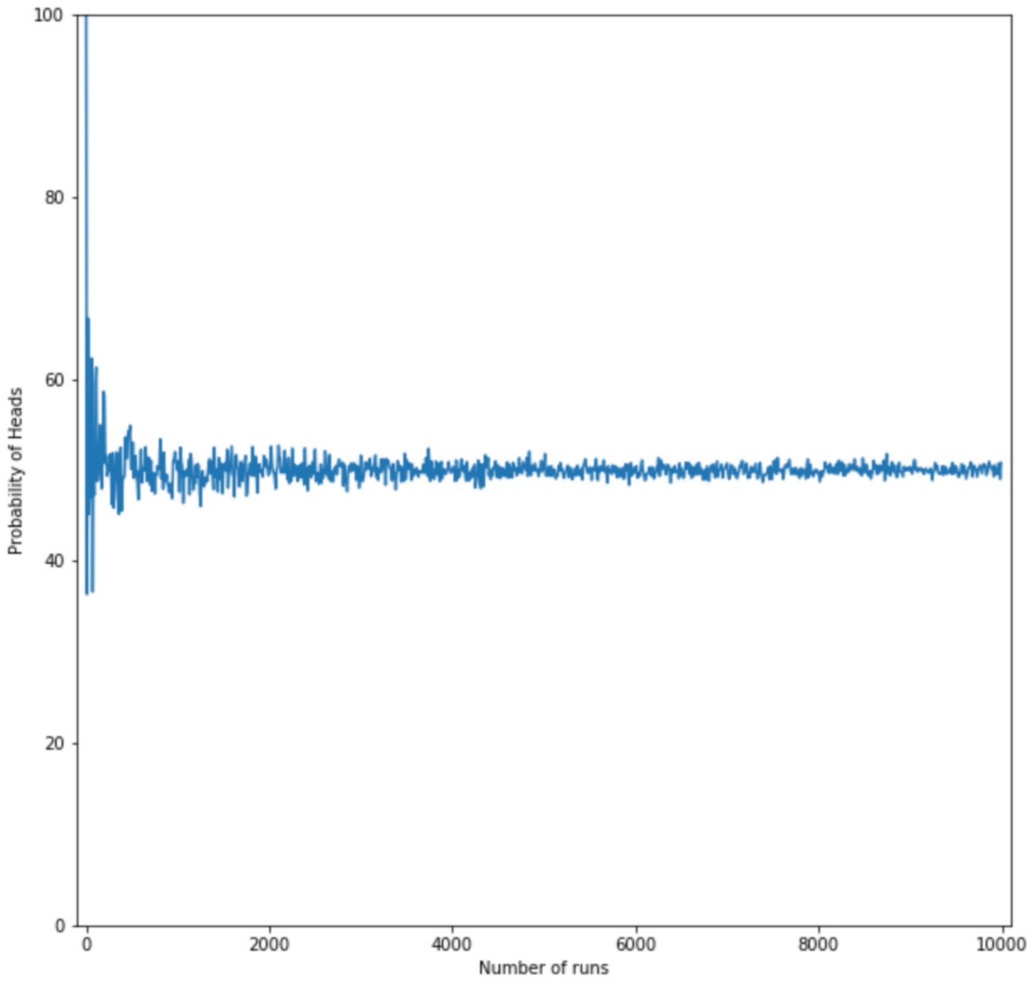
- What is the odds of seeing **heads** up if you flip a fair coin one time?
- Fact: **50% heads** and **50% tails**.
- Assume that we don't know this fact or suspect that the coin is fixed. Can we estimate the odds using the Monte Carlo simulation method?
- Navigate to google.com and search: flip a coin
  - Repeat this 10 times
  - What did you observe?

Coin flipping example in Python

- Python code to simulate coin flipping
  - Simulate 1 time, 10 times, 100 times, 1,000 times, and 10,000 times
- Simple coin flipping code

```
np.random.choice([0,1])
```

0: Heads
1: Tails



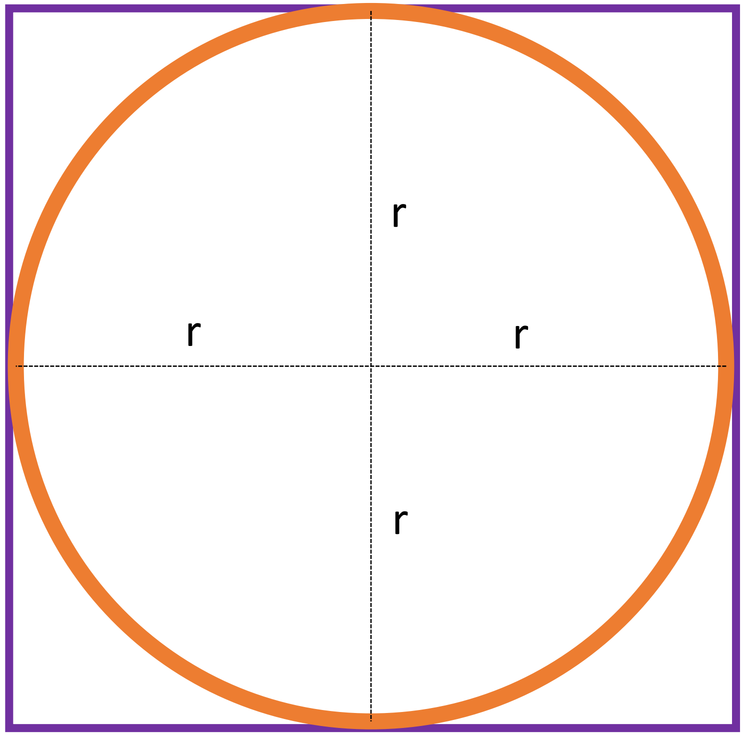
More coin flipping examples

- Now the goal is to estimate the odds of the certain sequential combinations
- Sequences to estimate
  - Flip three times and check if the sequence is **Heads, Heads, Heads**
  - Flip three times and check if the sequence is **Heads, Tails, Heads**
  - Flip five times and check if the sequence is **Heads, Heads, Tails, Tails, Tails**

Estimating the Pi (π) number example

Area of this circle is… πr^2

Area of the enclosing square is … $2r * 2r = 4r^2$

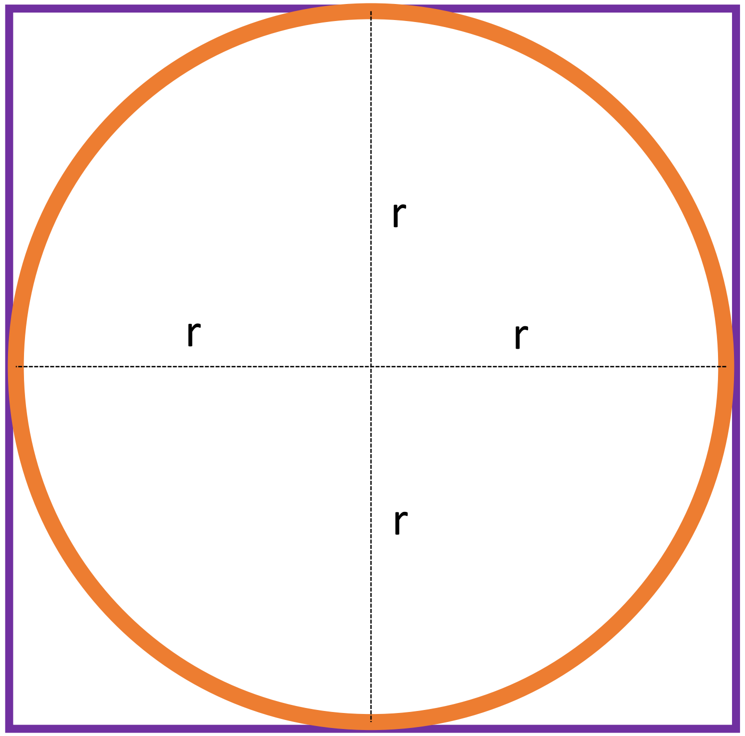


If we throw a dart randomly to this square, the odds of falling in the circle is… $\frac{Area\ of\ circle}{Area\ of\ square}$

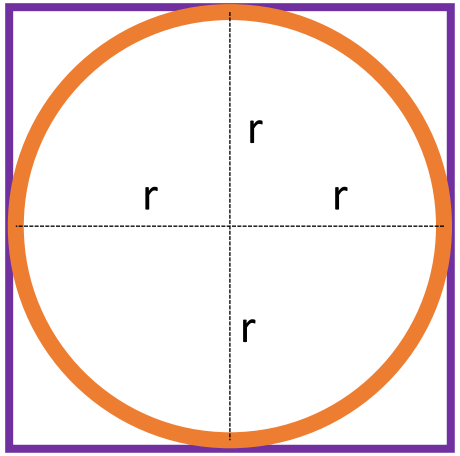
$or \quad \frac{\pi r^2}{4r^2} = \frac{\pi}{4}$

Estimating the Pi (π) number example

- Write Python code to throw dart at the square randomly.
- Record number of times that dart falls within the circle and total number of darts thrown.
- $\frac{darts\ within\ circle}{total\ number\ of\ darts} \cong \frac{\pi}{4}$
- Try it for 10, 100, 1,000, 10,000 and 100,000 darts.

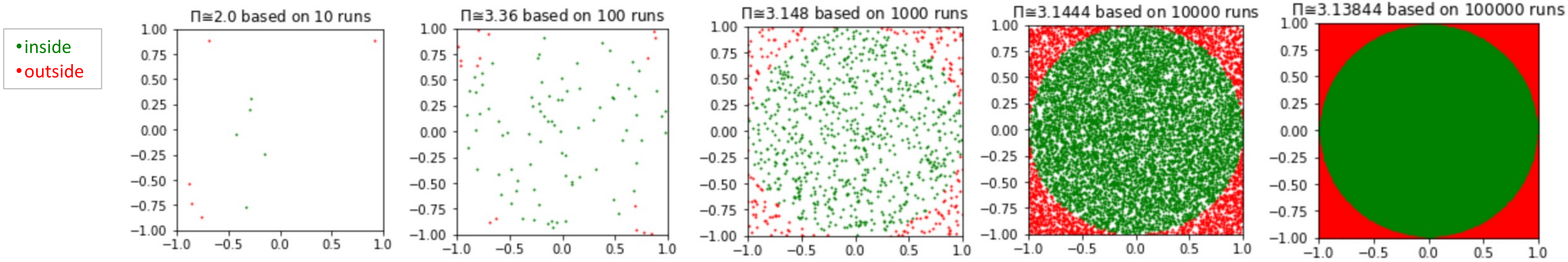


Pi number example visualized



If we throw a dart randomly, the odds of falling in the circle is

$$\frac{Area\ of\ circle}{Area\ of\ square} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4}$$

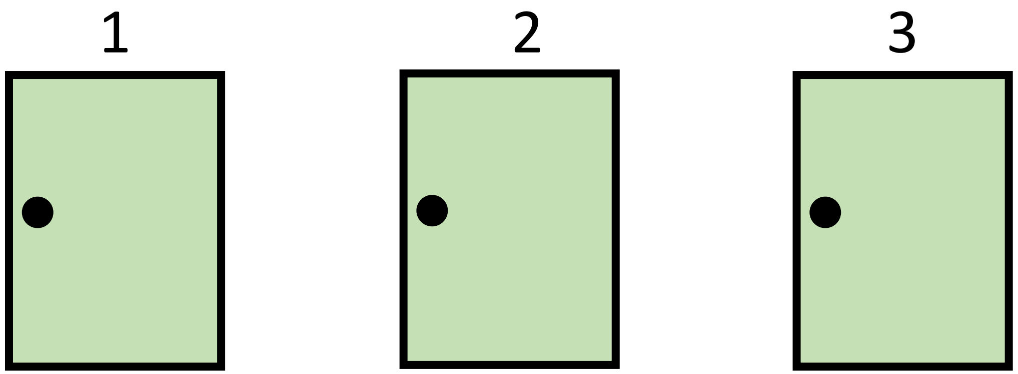


The Monty Hall problem

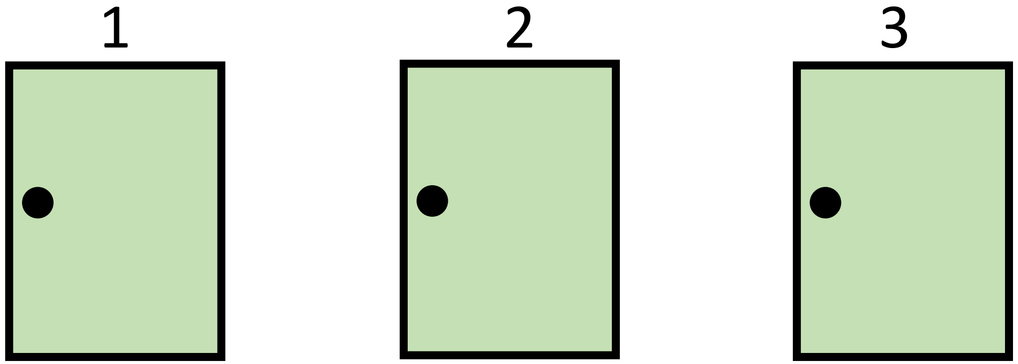
- Based on a Television program called Let’s Make a Deal hosted by Monty Hall.

Source: vos Savant, Marilyn (9 September 1990a). "Ask Marilyn". Parade Magazine: 16.

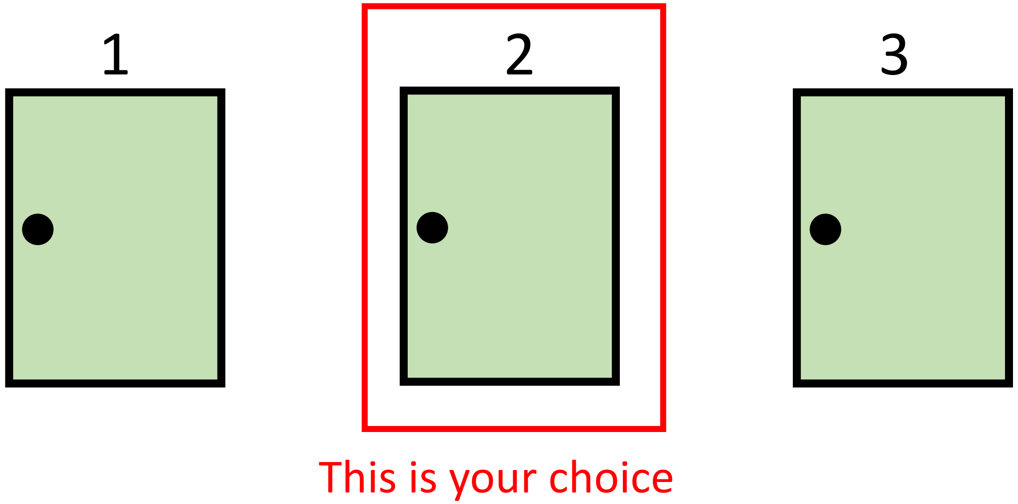
“Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?”



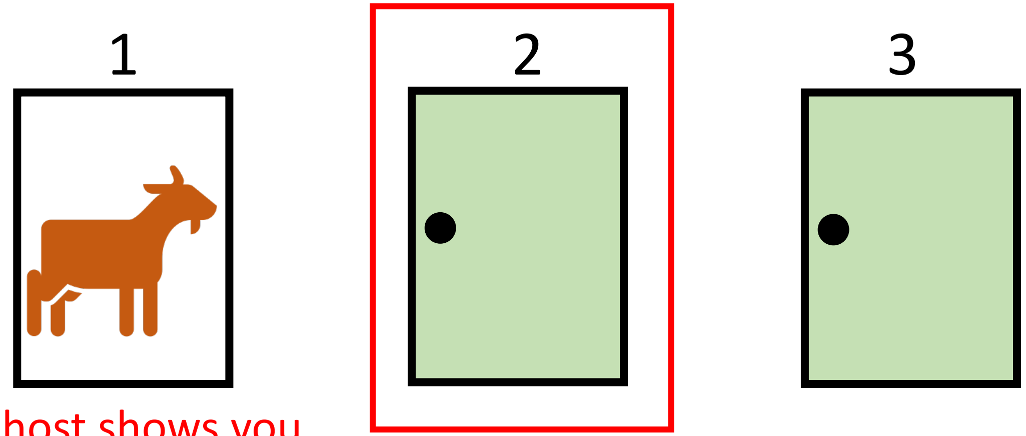
The Monty Hall problem



The Monty Hall problem



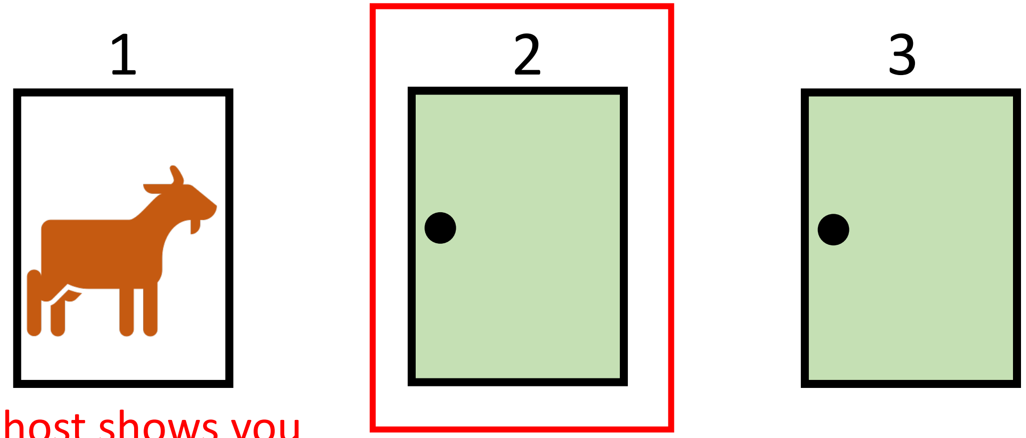
The Monty Hall problem



The host shows you that door #1 has a goat

This is your choice

The Monty Hall problem



The host shows you that door #1 has a goat

This is your choice

<u>Now you have 2 options</u>

Strategy 1: stick with your choice (door #2)

Strategy 2: choose the other door (door #3)

Python coding

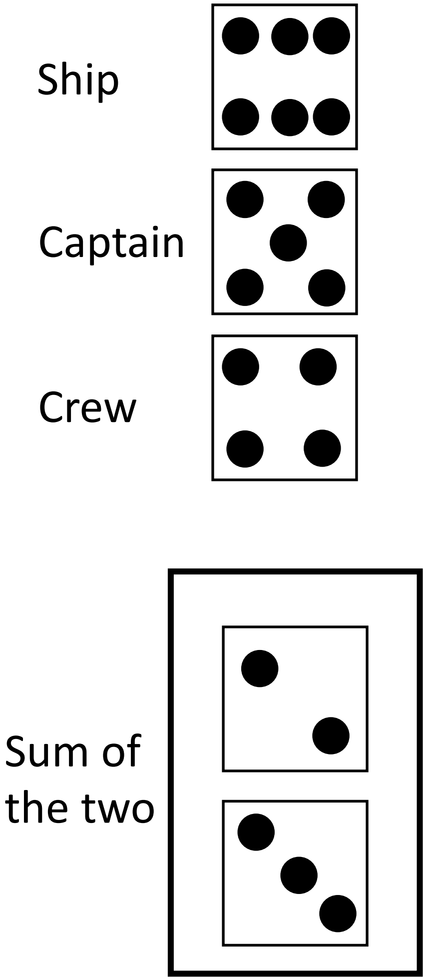
- The Monty Hall problem.
  - Implement the problem in python.
  - Implement Strategy 1 and Strategy 2
    - **Strategy 1:** stick with your choice
    - **Strategy 2:** choose the other door
  - Report which one is a better strategy based on N=1,000 Monte Carlo runs.

How many Monte Carlo runs needed (N) ?

- Usually, higher N is better
  - sometimes runs are too computationally intense so high N may not be feasible
- This is a trial-and-error process.
- Run the system N times and observe the average estimate and standard deviation.
- Are you comfortable with the standard deviation?
  - Yes: N is good. Well done!
  - No: N is not good, increase N and start over.

The bar dice game

- https://www.youtube.com/watch?v=cXNLrM10k_E
- Roll five dice
- You are trying to have a Ship (6), Captain (5), Crew (4),… in the same order.
- Once you have the above three, Your score is the sum of the remaining two dice.
- Like Yahtzee, you have up to three rolls
  - Can keep Ship (6), Captain (5), Crew (4) then roll the rest again.
- If you can’t build the ship, captain, and crew in three rolls, you get 0.



References

- Roger Eckhardt. *Stan ulam, john von neumann, and the monte carlo method*. Los Alamos Science, *15*(131-136), p.30. 1987
- http://www.billturnbull.com/solitaire/solitaire.htm
- Eric Grimson, John Guttag, and Ana Bell. *6.0002 Introduction to Computational Thinking and Data Science.* Fall 2016. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.
- Julian L. Simon. *Resampling: The New Statistics* (Second Edition). 1997.
- Mikael Amelin. *Monte Carlo Simulation in Engineering*. Course Compendium, KTH Royal Institute of Technology, Stockholm. 2015