Faculty of Information Technology

Fall 2020

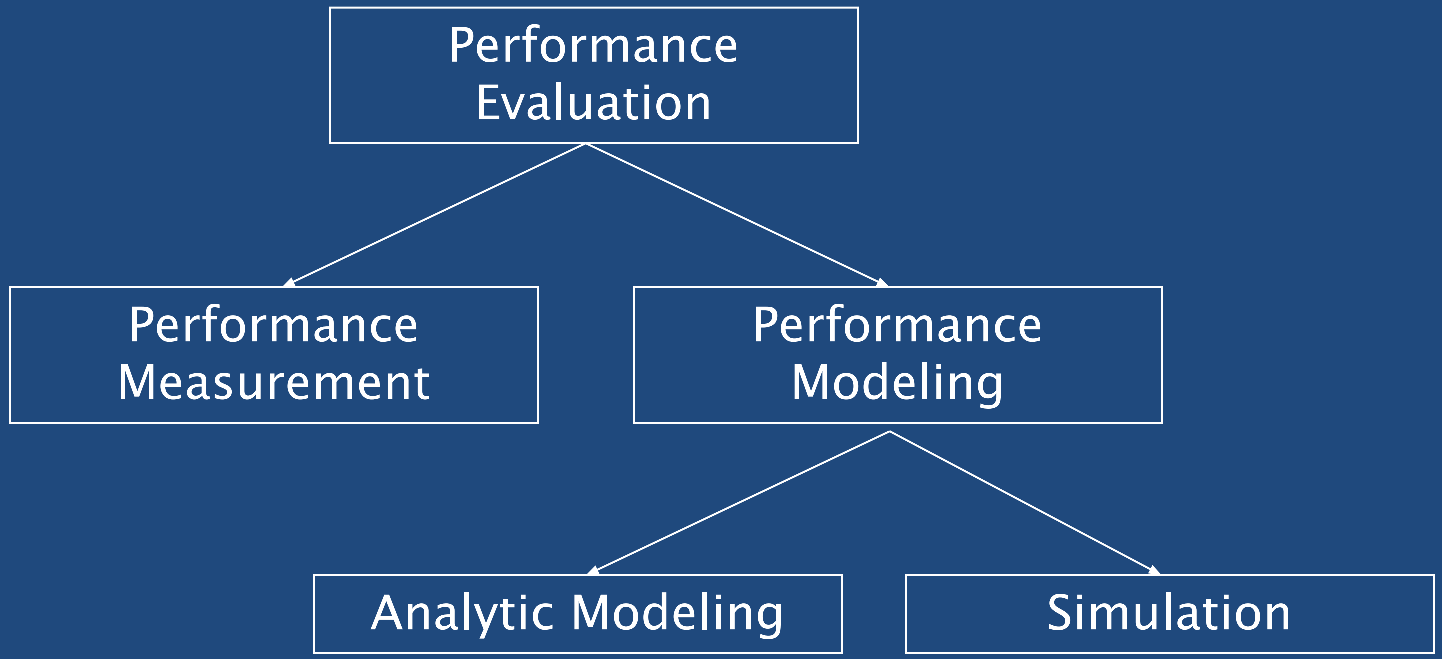
Modelling and Simulation

IS 331

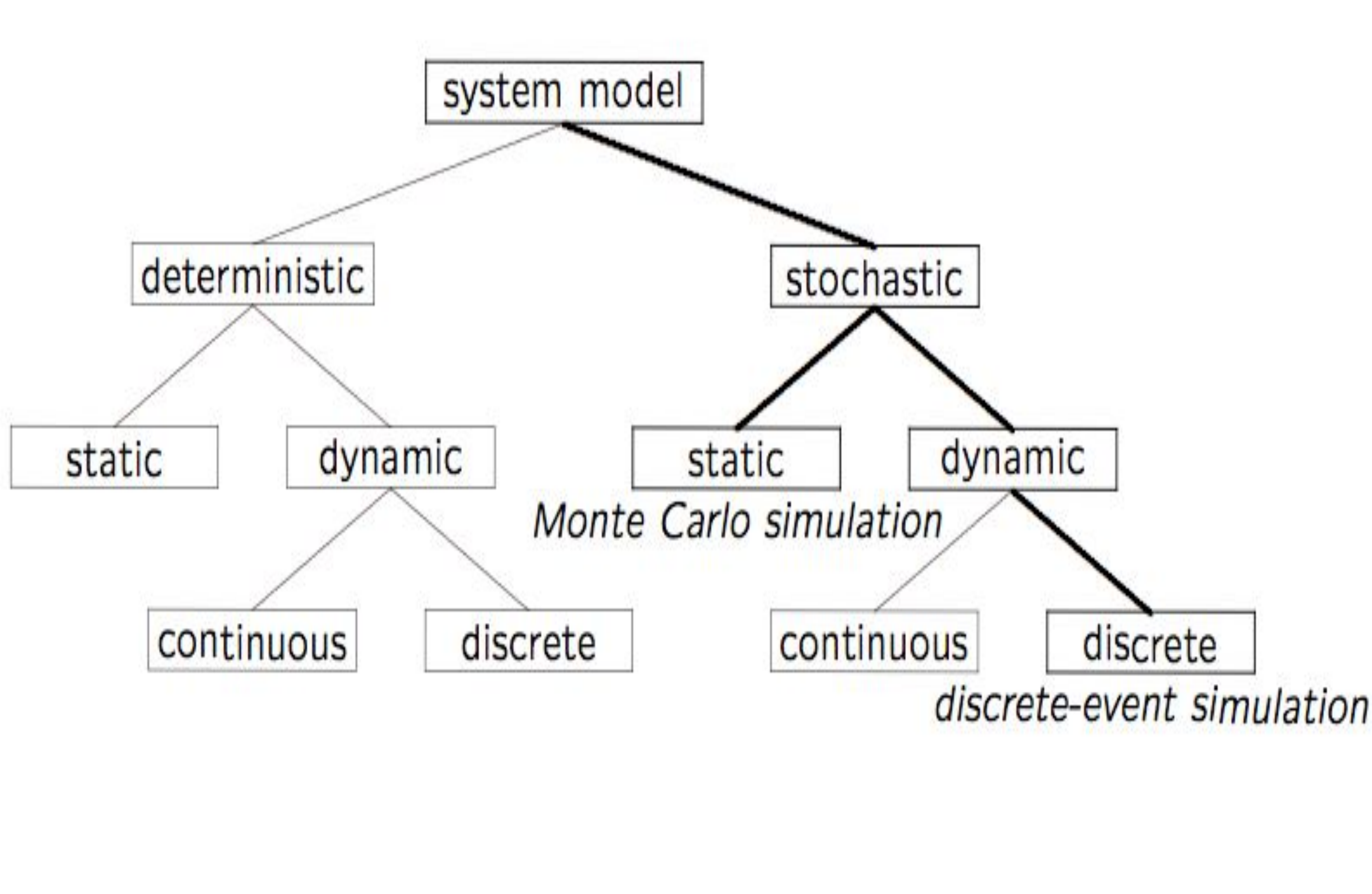
Lec (2)

By Dr. Alaa Zaghloul

Recap: Performance Evaluation



Simulation Model Taxonomy (preview)



Terminology (1 of 2)

- A *system* is defined as a group of objects that interact with each other to accomplish some purpose
  - A computer system: CPU, memory, disk, bus, NIC
  - An automobile factory: Machines, components parts and workers operate jointly along assembly line
- A system is often affected by changes occurring outside the system: system environment
  - Hair salon: arrival of customers
  - Warehouse: arrival of shipments, fulfilling of orders
    - Effect of supply on demand: relationship between factory output from supplier and consumption by customers

Terminology (2 of 2)

- Entity
  - An object of interest in the system: Machines in factory
- Attribute
  - The property of an entity: speed, capacity, failure rate
- State
  - A collection of variables that describe the system in any time: status of machine (busy, idle, down,…)
- Event
  - An instantaneous occurrence that might change the state of the system: breakdown

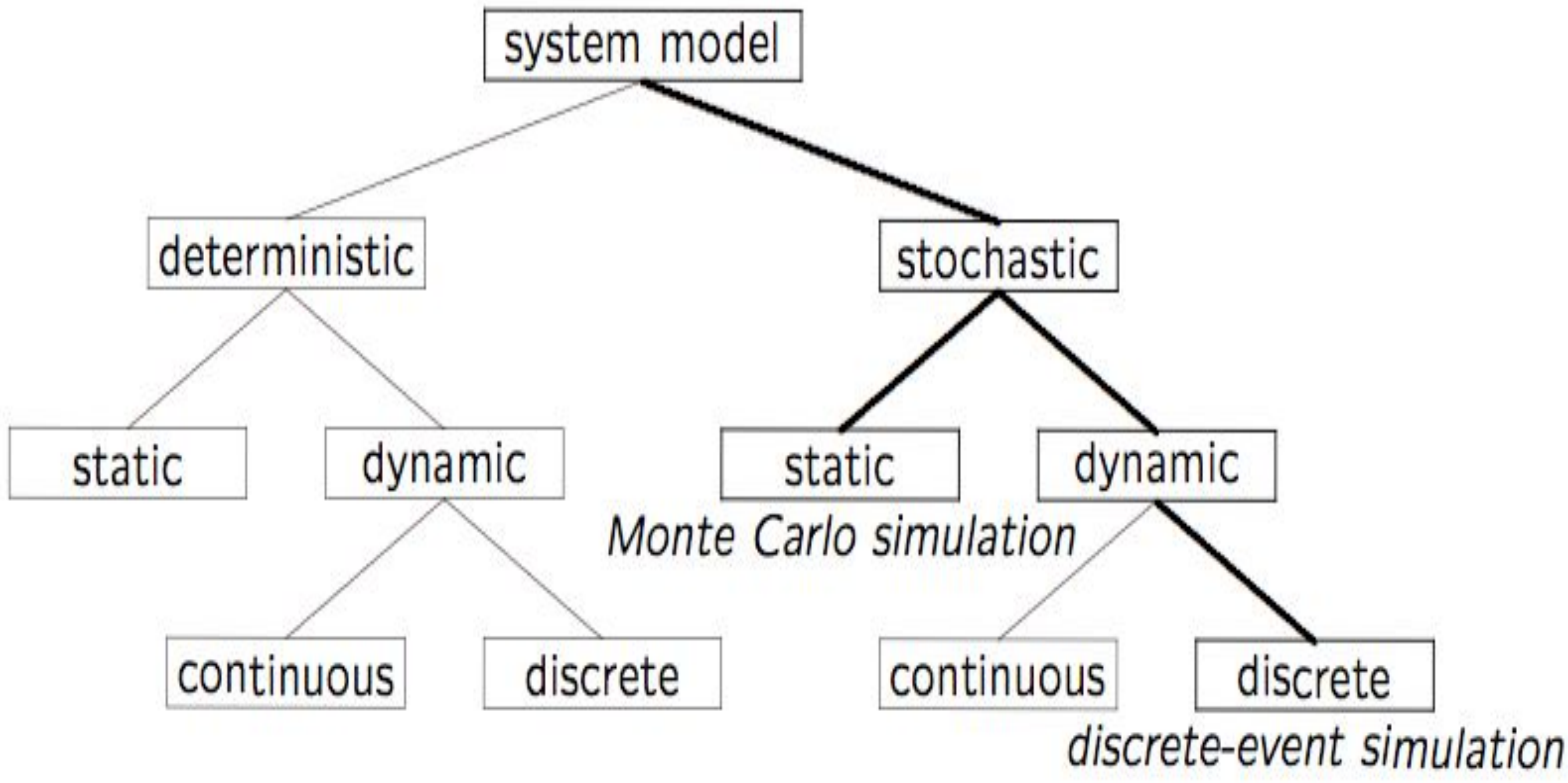
When Simulation Is Not Appropriate

- When the problem can be solved by common sense
- When the problem can be solved analytically
- When it is easier to perform direct experiments
- When cost of simulations exceeds (expected) savings for the real system
- When system behavior is too complex (e.g., humans)

Types of Simulations

- Monte Carlo simulation
- Time-stepped simulation
- Trace-driven simulation
- Discrete-event simulation
- Continuous simulation

Simulation Model Taxonomy



Simulation Examples

- Monte Carlo simulation
  - Estimating π
  - Craps (dice game)
- Time-stepped simulation
  - Mortgage scenarios
- Trace-driven simulation
  - Single-server queue (ssq1.c)
- Discrete-event simulation
  - Witchcraft hair salon

Simulation Examples

- Monte Carlo simulation
  - Estimating π
  - Craps (dice game)

Monte Carlo Simulation

Named after Count Montgomery de Carlo, who was a famous Italian gambler and random-number generator (1792-1838).

- Static simulation (no time dependency)
- To model probabilistic phenomenon
- Can be used for evaluating non-probabilistic expressions using probabilistic methods
- Can be used for estimating quantities that are “hard” to determine analytically or experimentally

Classic Example

Find the value of π ?

Use the reject and accept method

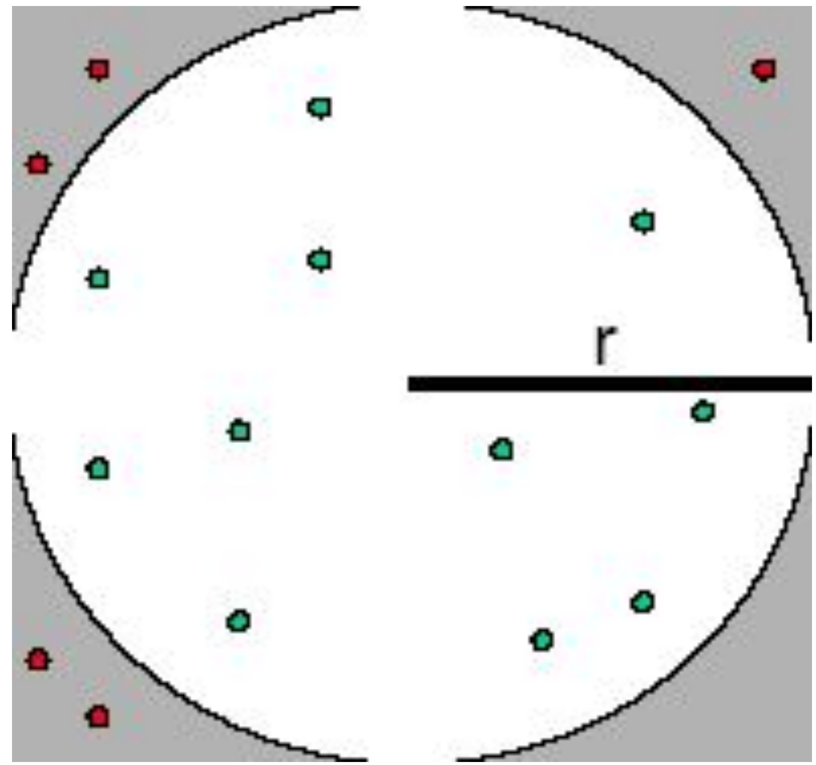
Or hit and miss method

The area of square=$(2r)^2$

The area of circle = πr^2

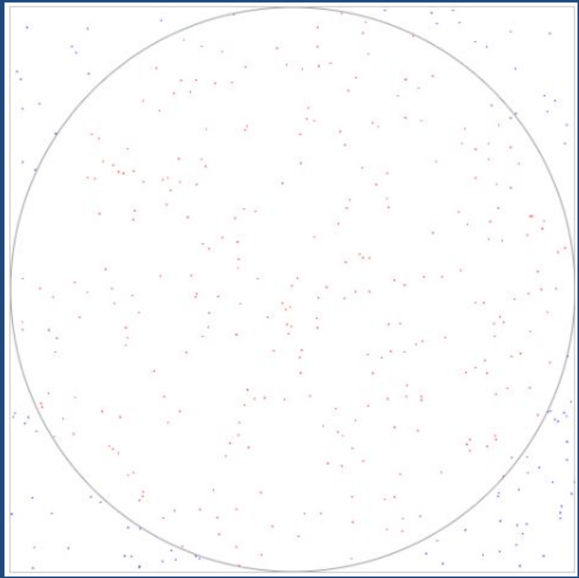
$$\frac{area \cdot of \cdot square}{area \cdot of \cdot circle} = \frac{4r^2}{\pi r^2} = \frac{4}{\pi}$$

$$\pi = 4 * \frac{area \cdot of \cdot circle}{area \cdot of \cdot square}$$

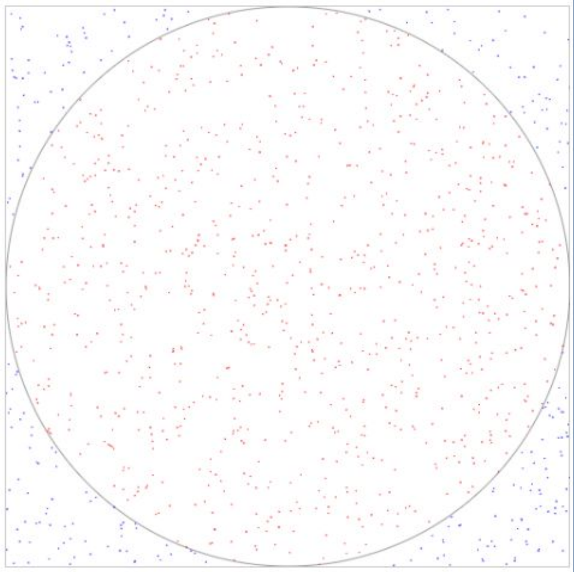


Estimating Pi using the Monte Carlo Method

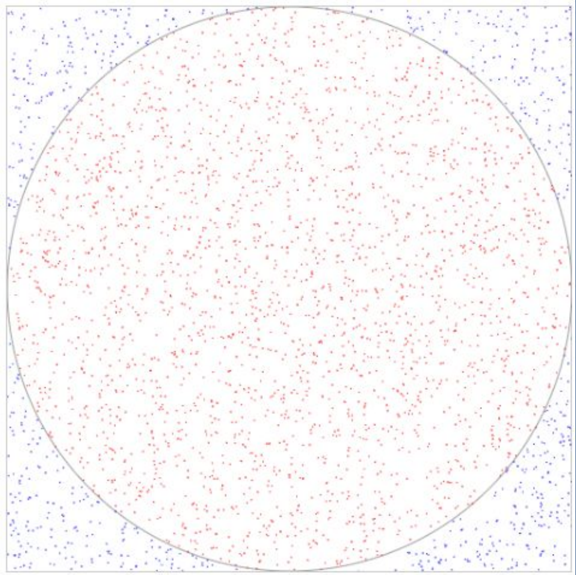
$$\frac{\pi}{4} \approx \frac{N_{inner}}{N_{total}} \qquad \pi \approx 4\frac{N_{inner}}{N_{total}}$$



Total Number of points: 384
Points within circle: 295
Pi estimation: 3.07292



Total Number of points: 1152
Points within circle: 903
Pi estimation: 3.13542

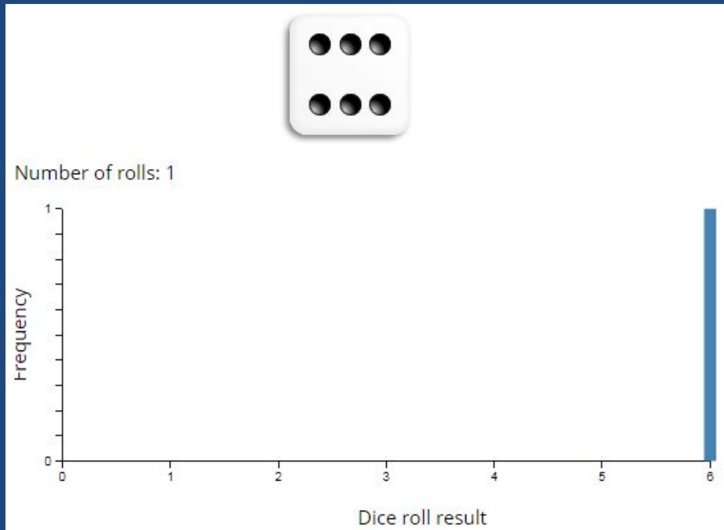


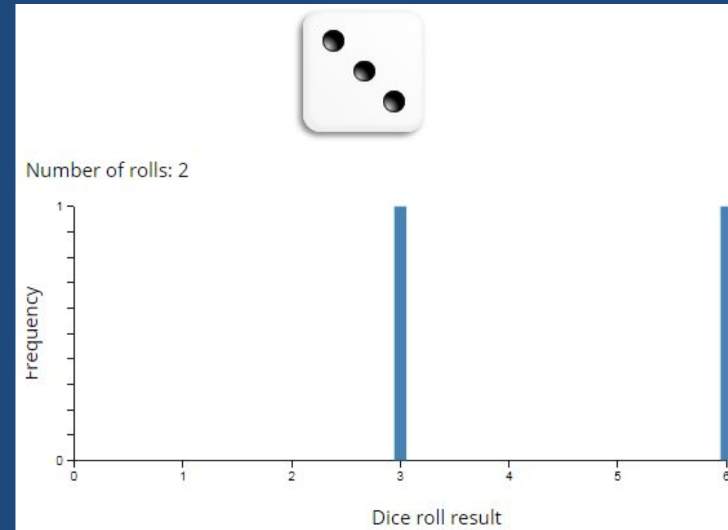
Total Number of points: 3336
Points within circle: 2617
Pi estimation: 3.13789

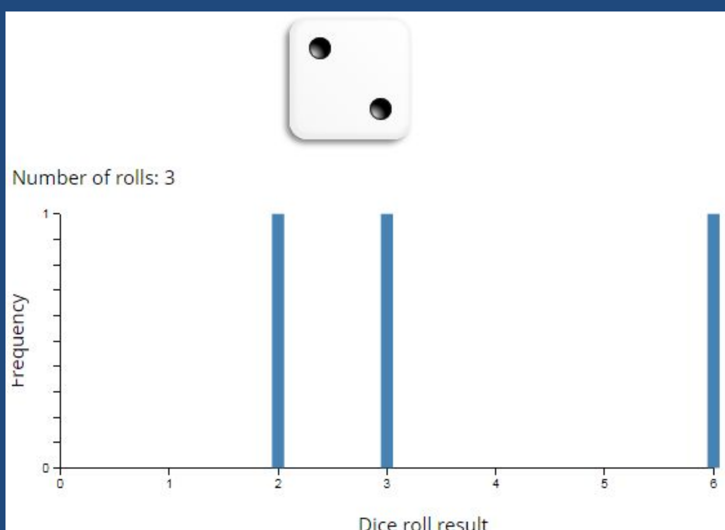
Statistics of rolling dice

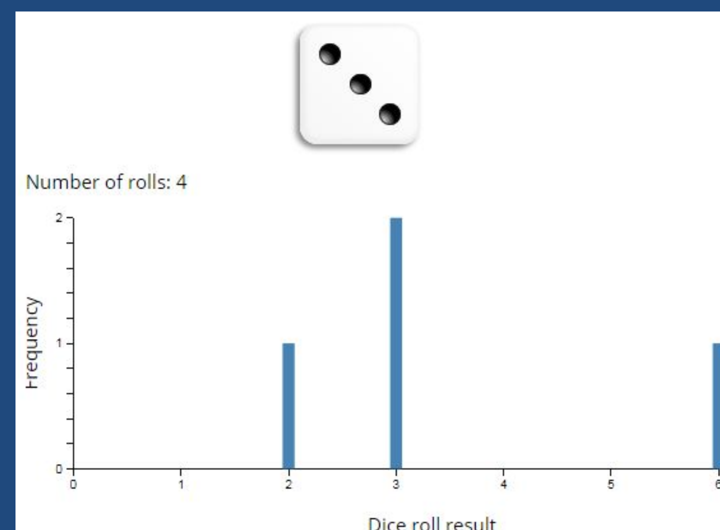
If you roll a fair, 6-sided die, there is an equal probability that the die will land on any given side. That probability is 1/6. This means that if you roll the die 600 times, each face would be expected to appear 100 times. You can simulate this experiment by ticking the "roll automatically" button above.

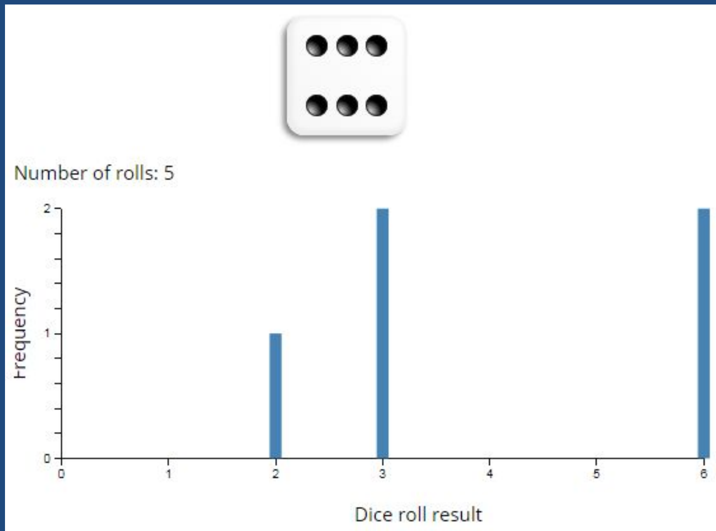
Now imagine you have two dice. The combined result from a 2-dice roll can range from 2 (1+1) to 12 (6+6). However, the probability of rolling a particular result is no longer equal. This is because there are multiple ways to obtain certain results. Let's use 7 as an example. There are 6 different ways: 1+6, 2+5, 3+4, 4+3, 5+2, 6+1, whereas the result 2 can only be obtained in a single way, 1+1. This means you are 6 times more likely to achieve a 7 than you are to achieve a 2.


Number of rolls: 1
1
0
Frequency
0 1 2 3 4 5 6
Dice roll result

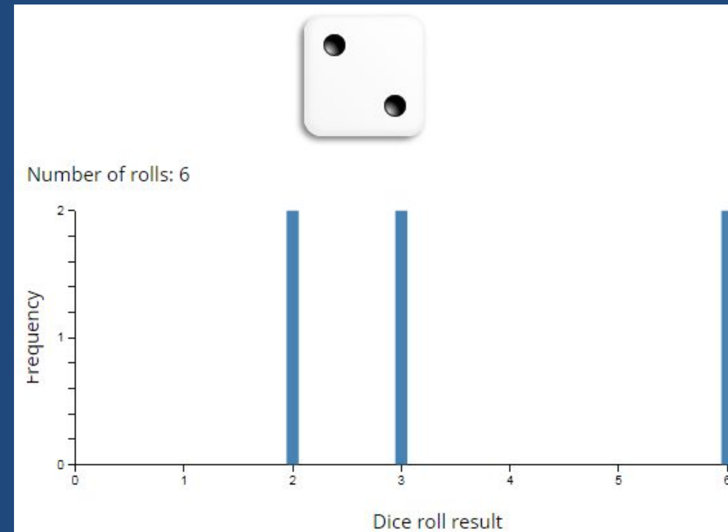

Number of rolls: 2
1
0
Frequency
0 1 2 3 4 5 6
Dice roll result


Number of rolls: 3
1
0
Frequency
0 1 2 3 4 5 6
Dice roll result

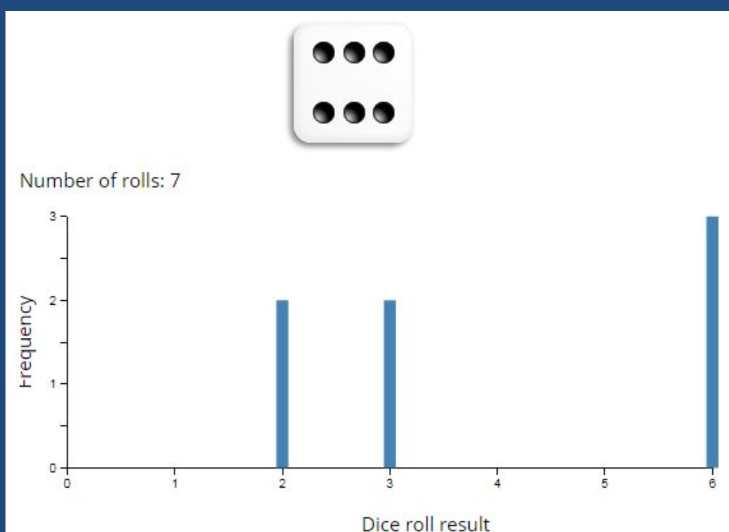

Number of rolls: 4
2
1
0
Frequency
0 1 2 3 4 5 6
Dice roll result



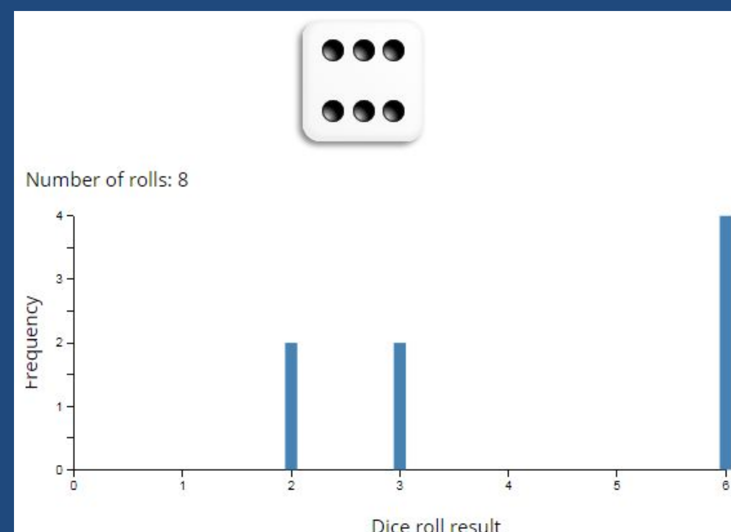
Number of rolls: 5
Frequency
0 1 2
0 1 2 3 4 5 6
Dice roll result



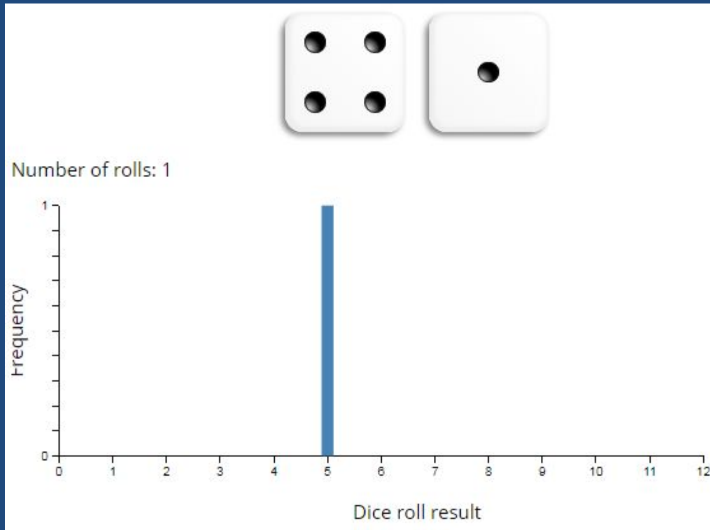
Number of rolls: 6
Frequency
0 1 2
0 1 2 3 4 5 6
Dice roll result

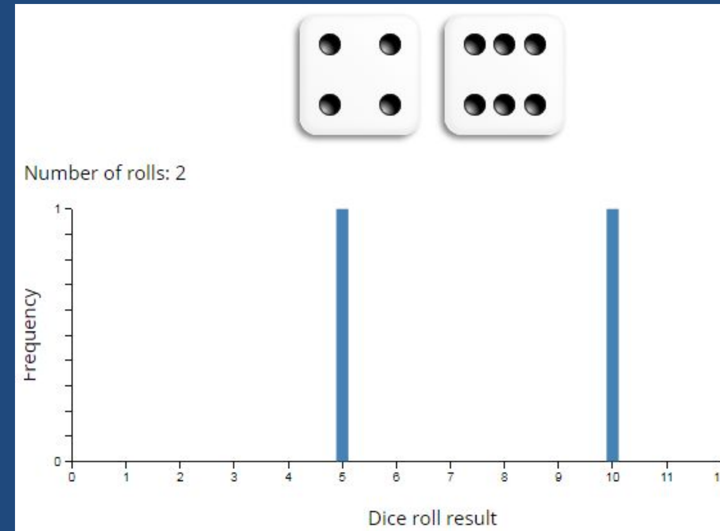


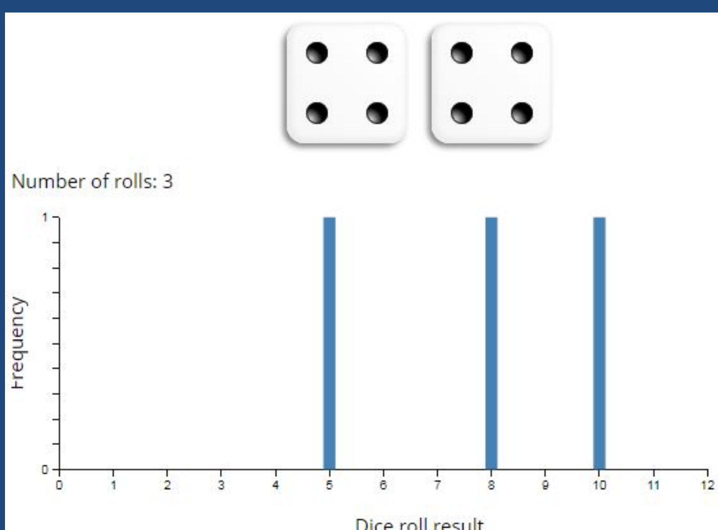
Number of rolls: 7
Frequency
0 1 2 3
0 1 2 3 4 5 6
Dice roll result

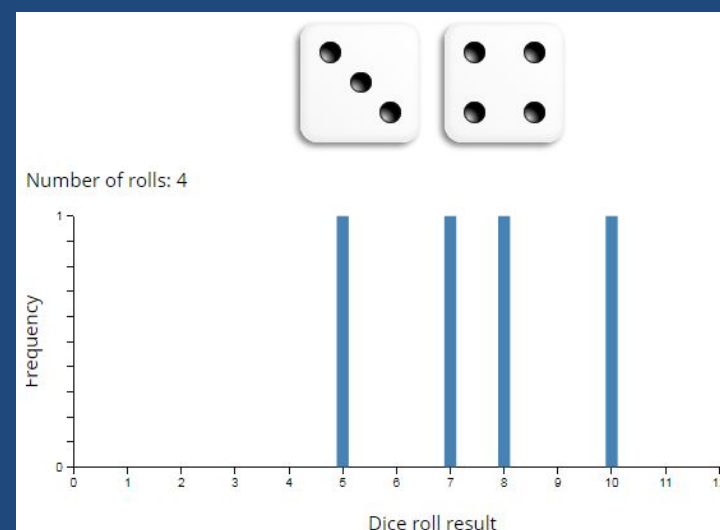


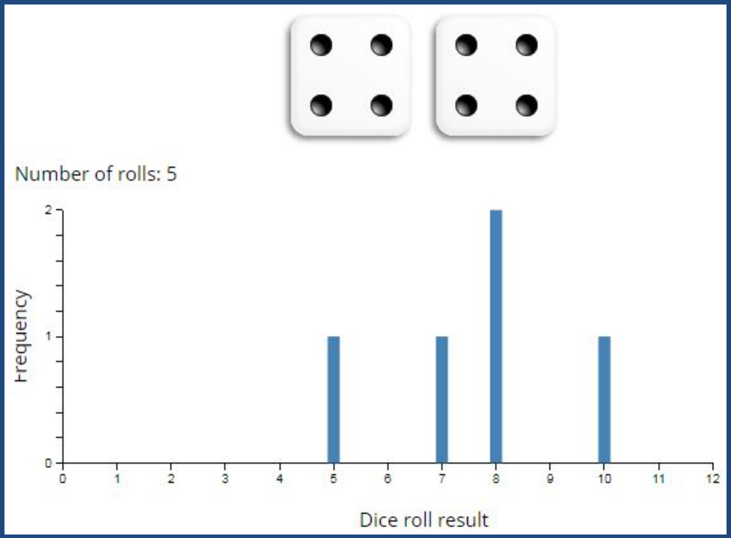
Number of rolls: 8
Frequency
0 1 2 3 4
0 1 2 3 4 5 6
Dice roll result

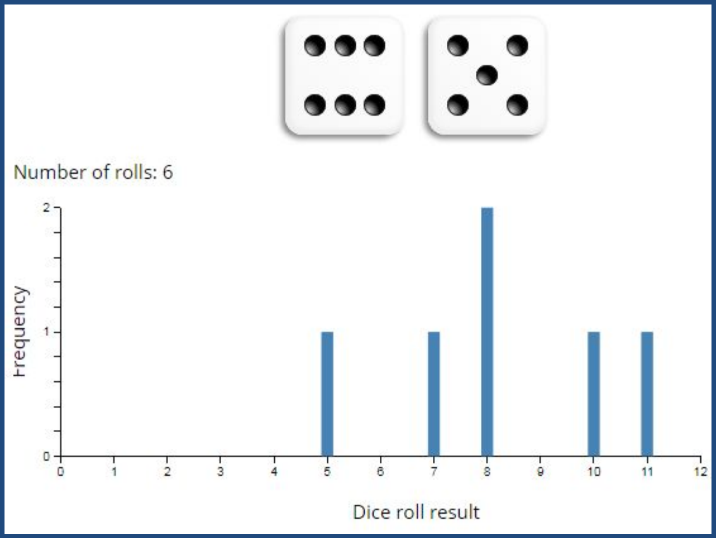

Number of rolls: 1
1
0
Frequency
0 1 2 3 4 5 6 7 8 9 10 11 12
Dice roll result

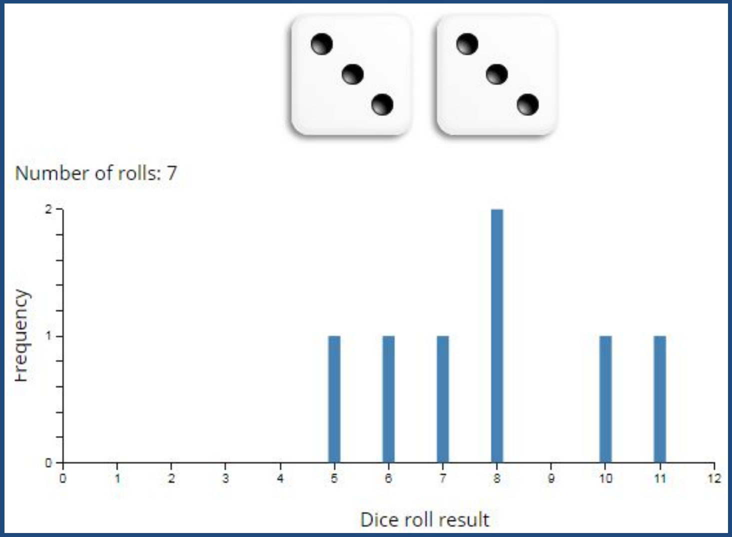

Number of rolls: 2
1
0
Frequency
0 1 2 3 4 5 6 7 8 9 10 11 12
Dice roll result

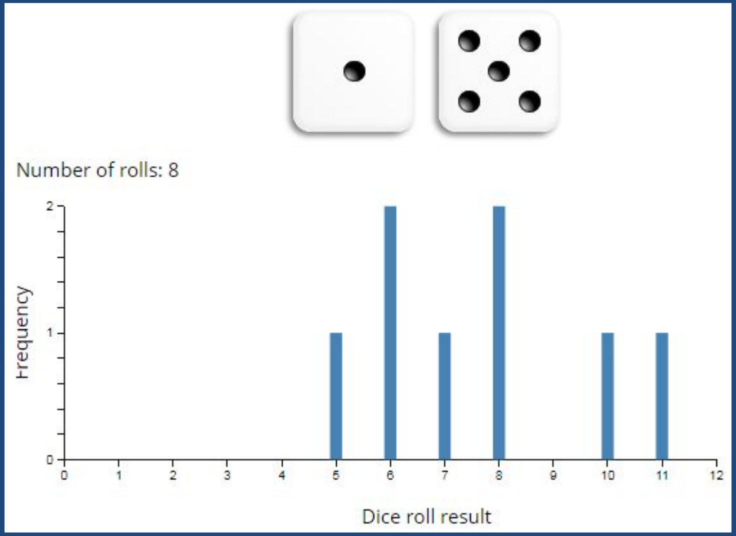

Number of rolls: 3
1
0
Frequency
0 1 2 3 4 5 6 7 8 9 10 11 12
Dice roll result


Number of rolls: 4
1
0
Frequency
0 1 2 3 4 5 6 7 8 9 10 11 12
Dice roll result


Number of rolls: 5
Frequency
Dice roll result


Number of rolls: 6
Frequency
Dice roll result


Number of rolls: 7
Frequency
Dice roll result


Number of rolls: 8
Frequency
Dice roll result

Simulation Examples

- Monte Carlo simulation
  - Estimating π
  - Craps (dice game)

Monte Carlo Simulation of the Craps Dice Game

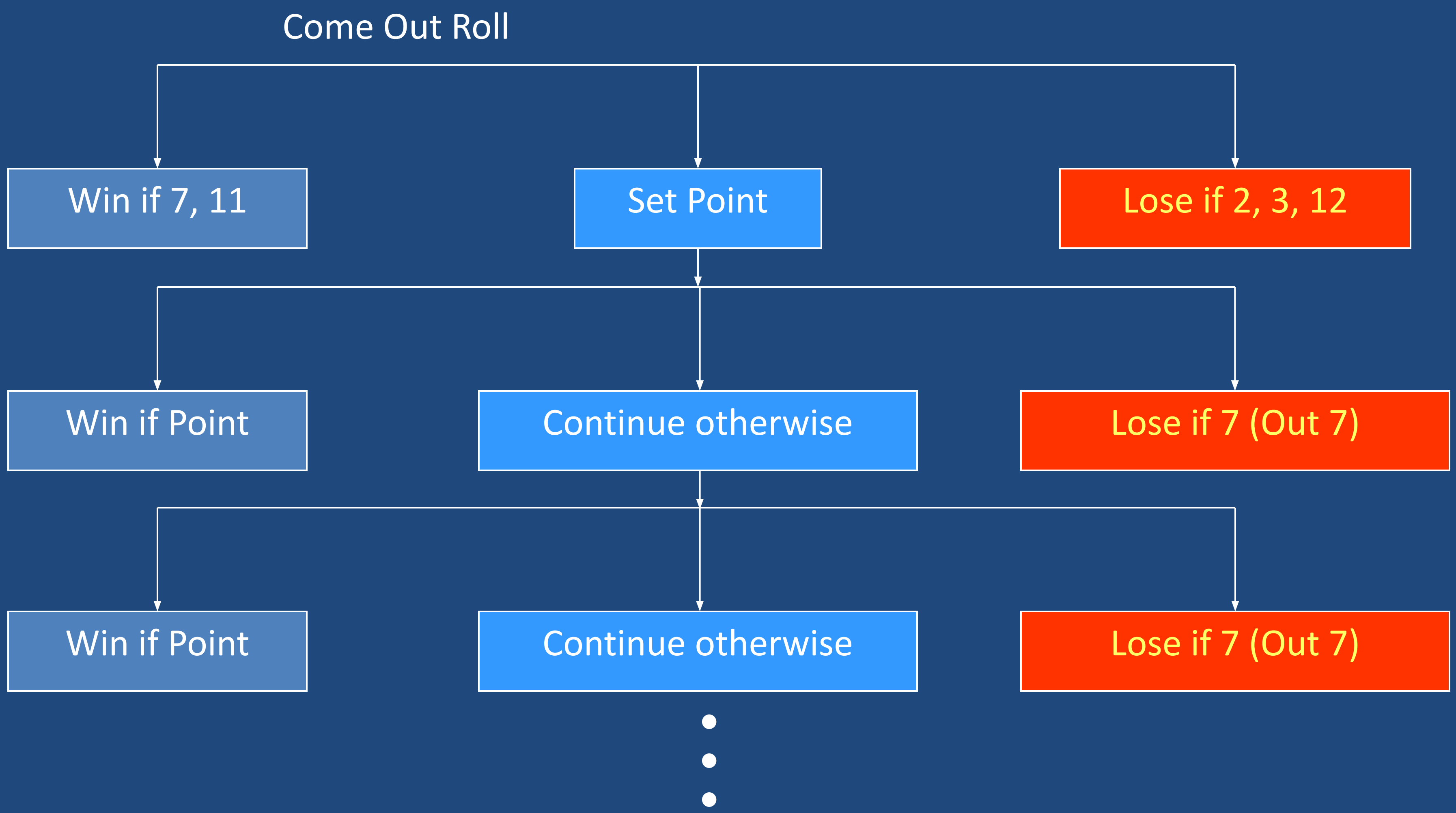
Basics

- The player rolling the dice is the "shooter". Shooters first throw in a round of Craps is called the Come Out roll. If you roll a 7 or 11, you win and the round is over before it started.
- If you roll a 2, 3, or 12 that's a Craps and you lose: again, it's over before it started.
- Any other number becomes the Point. The purpose of the Come Out roll is to set the Point, which can be any of 4, 5, 6, 8, 9 or 10.

Objective

- The basic objective in Craps is for the shooter to win by tossing the Point again before he tosses a 7. That 7 is called Out 7 to differentiate it from the 7 on the Come Out roll.
- If the Point is tossed, the shooter and his fellow bettors win and the round is over. If the shooter tosses Out 7, they lose and the round is over.
- If the toss is neither the Point nor Out 7, the round continues and the dice keep rolling.

Craps Game


Come Out Roll
Win if 7, 11
Set Point
Lose if 2, 3, 12
Win if Point
Continue otherwise
Lose if 7 (Out 7)
Win if Point
Continue otherwise
Lose if 7 (Out 7)

Questions

- What is the probability that the roller wins?
  - Note that this is not a simple problem.
    - The probability of win at later rolls depends on the point value, e.g., if the point is 8, P(8)=5/36 ({2,6},{3,5},{4,4},{5,3},{6,2}) and if the point is 10, P(10)=1/36 ({5,5}).
- How many rolls (on the average) will the game last?

A single game

```
% Roll the dice and find their sum
die1 = randi(6);
die2 = randi(6);
sum  = die1 + die2;

% Check if we have a natural
if (sum == 7 || sum == 11)
    display('We won!');

% Check if we have craps
elseif (sum == 2 || sum == 3 || sum == 12)
    display('We lost...');

% Otherwise, we roll the dice until we get the initial sum
% or a sum of 7
else
    while (true)
        % Roll the dice and find their sum
        die1    = randi(6);
        die2    = randi(6);
        sum_new = die1 + die2;

        if (sum_new == sum)
            display('We won!');

            break;

        elseif (sum_new == 7)
            display('We lost...');

            break;

        end
    end
end
```

Monte Carlo simulation

We can play a single game of craps. To estimate the probability of winning craps using Monte Carlo simulation, we need to play the game multiple times, say $N = 100,000$ times. Recall that a loop is great for repeating something. Since we know how many times, let's use a **for** loop.

```
% Reset the number of wins
numWins = 0;

% Set the number of simulations
N = 10^5;

% Run the simulations
for i = 1 : N
    % Run a single game of craps and increase the number
    % of wins by 1 if we win the game
    [...]
end

% Display the probability of winning craps
display(numWins / N);
```

The probability that I got is 0.49228, slightly less than 0.5. It seems that the game is unfavorable to the player.

Exact

The exact value of probability of win can be calculated by using the theory of Markov Chains

$$0.4921 \le P(\text{win}) \le 0.4937 \quad \text{95\% confidence interval}$$

$$P(\text{win}) = \frac{976}{1980} = 0.4929$$

Examples

<u>Example 1:</u>

The first example we are going to see is the simulation of a tossing of a fair coin. First step is analyzing the problem. The fair coin means that when tossing that coin the probability of head equal the probability of tail equal 50%. So, using a digital computer to simulate this phenomenon we are going to use a uniform Random number generated by the package you are using or you can write its code.

Uniform random number means that you have a set of random number between 0 and 1 all with the same probability. But most languages generate uniform random number integer from a to b with equal probability.

See the following program written in c++

```
#include <iostream>
#include <stdlib.h>
void main(void)
{
int x,nuber_or_trials, head=0, tail=0;
float phead,ptail,error_head,error_tail;
cout<<"enter number of trials"<<endl;
cin>>nuber_or_trials;
for(int i=0;i<nuber_or_trials;i++)
{
    x=random(2);
      if (x==1)  head++;
              else  ++tail;
 }
 phead=head*1.0/nuber_or_trials;
 ptail=tail*1.0/nuber_or_trials;
 error_head = abs(((0.5 - phead)/0.5)*100);
 error_tail    = abs(((0.5 - ptail)/0.5)*100);
 cout<<"probability of head= " <<phead<<"   with    error ="<<error_head<<"%"<<endl;
 cout<<"probability of tai = "<<ptail<<"   with error ="<<error_tail<<"%"<<endl;
 cin>>x;
}
```

The output results are:

Enter number of trials = 1

probability of head= 0   with error =100%

Probability of tail = 1   with error =100%

Enter number of trials   =5

probability of head= 0   with error =100%

Probability of tail = 1   with error =100%

Enter number of trials   = 32767

probability of head= 0.496292   with error =0%

Probability of tail = 0.503708   with error =0%

Example2

Get the average daily demand for a small grocery store selling a fresh bread according to the following table:

Device to generate chance outcomes Number and color of balls	Probability Of demand P(D)	Daily demand D
20 Red	0.20	100
50 Blue	0.50	110
30 Yellow	0.30	120

Note that:

the proportion of balls of a specific color corresponds exactly to the probability of a specific level of daily demand.

To simulate the daily demand(5 days)

- Draw one ball at a time, notice its color and then place it back in the bowl. Then translate the outcomes into unique values of demand.

Simulated Demand	Day of the Week	Color of ball	Sample Number
110	Monday	Blue	1
110	Tuesday	Blue	2
120	Wednesday	Yellow	3
120	Thursday	Yellow	4
100	Friday	Red	5
560	Total		

- Expected value of simulated demand

  = 560/5=112 units / day

- Analytical solution: Expected daily demand

  =100(0.2) + 110(0.5) +120(0.3) = 111 units / day

END