OntarioTech Engineering & Applied Science

SOFE 4820U: Modelling and Simulation Winter 2024

Dr. Anwar Abdalbari

Week 2 : Randomness and Generation

Chapter outlines

- Definition of Bernoulli process
- Stochastic processes
- Random number simulation
- The pseudorandom number generator
- Exploring generic methods for random distributions
- Random number generation using Python

Stochastic vs Deterministic Simulation Policies

Simulation-based player:

- Simulations need to be stochastic and randomize moves.
- Simulations need to explore different movement sequences.

- The opposite of stochastic: deterministic
  - Deterministic policy: all simulations from the same start state play the same sequence, and have the same result.
  - This is useless!
  - All moves would have either a 0% or 100% win rate.

Why Use Randomized Simulation Policies?

- Having variety in simulations is very important.
- It gives us more information about the huge state space.
- This is the main idea of sampling.
- We hope that errors in simulation “average out” through randomness.
- This is true if simulations have no bias.
- The variance can be reduced by getting more samples.
- Contrast with deterministic policy:
    - it repeats the same errors in each try

Why Use Randomized Simulation Policies? Cont’d

- Imagine a large table with a selection of different drinks.
- There are more of some drinks than others Random experiment:
- The waiter randomly grabs one drink
- Given the probability of selecting each drink
- drinks= [("Coffee",0.3),("Tea",0.2), ("OJ",0.4),...]
- Repeat random experiment many times
- Measure drink selection frequency empirically.

Bernoulli Trial

A Bernoulli trial is a random experiment or process that satisfies the following conditions:

- **Two Outcomes**: There are only two possible outcomes for each trial, commonly referred to as "success" and "failure."
- **Constant Probability**: The probability of success (denoted as p) remains constant for each trial. Similarly, the probability of failure (denoted as q, where $q=1-p$) also remains constant.
- **Independence:** The outcome of one trial is independent of the outcomes of previous and future trials. The outcome of one trial does not affect the outcome of another.

Bernoulli Trial (2)

- Bernoulli trials are named after Jacob Bernoulli, a Swiss mathematician, who introduced the concept in the 18th century.

- The probability distribution that arises from a sequence of Bernoulli trials is called the Bernoulli distribution.

Bernoulli’s trials

- Two events are said to be **complementary when** the occurrence of the first excludes the occurrence of the second but one of the two will certainly occur.
- X can take integer values 0 or 1
- Where E[x] is the expected value of x. To get the mean, we multiply each value by its probability.
- μ = 0 *q + p * 1 = p

$$P[\text{success}] = p \qquad 0 \le p \le 1$$

$$P[\text{failure}] = 1 - p$$

$$X \sim birnoulli(p)$$
$$X = num.of success$$
$$X = 0, 1$$
$$P(X = 0) = \mathbf{q} = 1 - p$$
$$P(X = 1) = p$$
$$\mu = \mathbb{E}[X] = p$$
$$\sigma^2 = var(X) = p.q, \sigma = \sqrt{pq}$$

Bernoulli Distribution: Introduction

- One of the simplest probability distributions
- Random variable X with two different values
    - 0 (loss) or 1 (win)
- Bernoulli trials are used in many fields including thermodynamics, biology, physics, chemistry, and economics where it is used to model fluctuations in the stock market.

Bernoulli Distribution (2)

- Examples:
  - Sequence of lottery wins/losses.
  - Sequence of ups and downs of the Dow Jones.
  - Arrivals (each second) to a bank.
  - coin flip

Bernoulli Distribution (3)

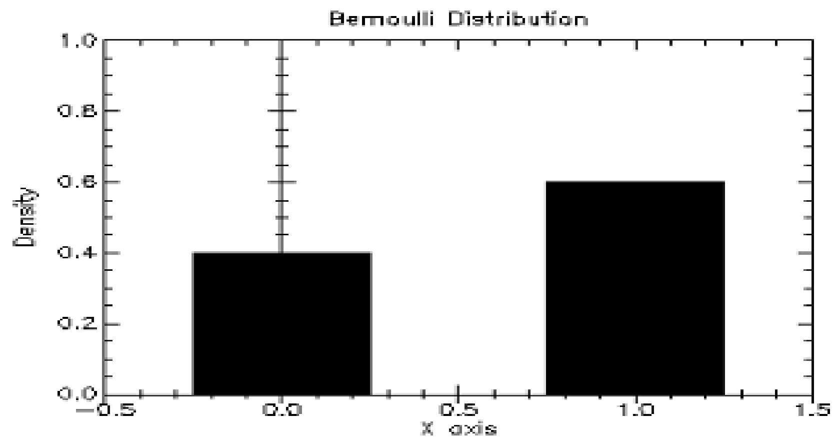


Image source:

http://planet.racket-lang.org/package-source/williams/science.plt/3/1/planet-docs/science/random-distributions.html

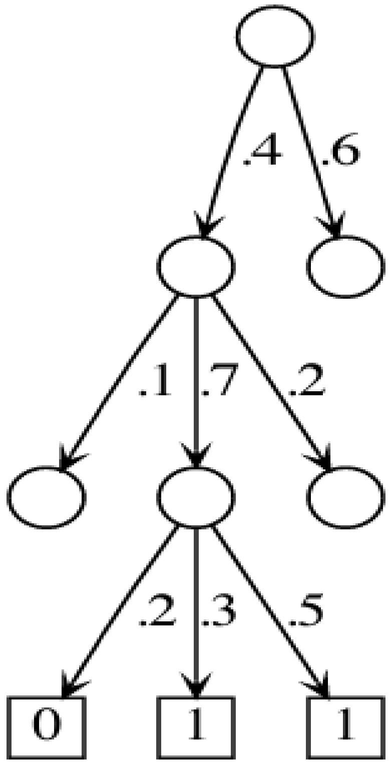
- Given fixed probability p with $0 \leq p \leq 1$
- Probabilities for outcomes 1 and 0
  - $Pr(X=1)=p$
  - $Pr(X=0)=1-p$
- Sometimes, q is written for $1-p$
- Example: $p=0.6, q=1-p=0.4$

Bernoulli Experiment

Image source: `https://en.wikipedia.org/wiki/Coin_flipping`

- Random experiment, typically repeated many times, same fixed p
- Every single experiment draws from Bernoulli distribution for p
- Example: coin flip with a fair coin, $p=q=0.5$
- Implementation: bernoulli.py

Simulation is a Bernoulli Experiment

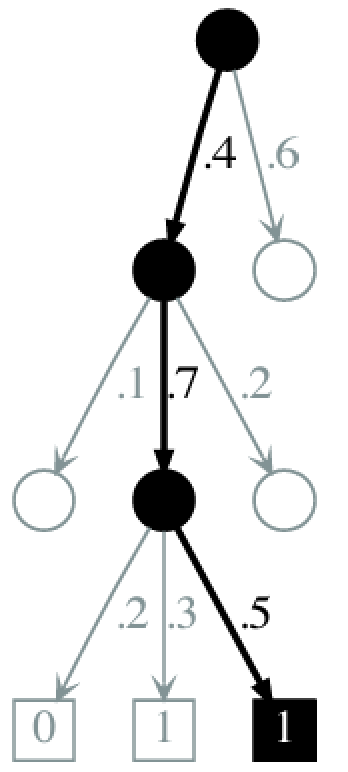


Why is a random sampling from a game tree a Bernoulli experiment?

Proof sketch

- Finite tree, finitely many leaves, finitely many paths to leaves.
- Each leaf has a fixed value 0 or 1
- In each node, the simulation policy has a fixed distribution over its children
- We can compute the probability of choosing each path as the product of the probabilities of choosing each move on the path Proof sketch

Simulation is a Bernoulli Experiment #2



.4 * .7 * .5 = .014

- The probability of choosing some specific path to a leaf is a(small) constant
- The winning probability at the root is just the probability of choosing a path leading to a win This is a sum of(a huge number of) constants, so is a constant p
- Each simulation is like a Bernoulli experiment with parameter p
- We win by choosing a winning path, which happens with probability p

Bernoulli Examples (1)

- We flip a coin 12 times and count the number of heads. Here n=12. Each flip is a trial.
- It is reasonable to assume the trials are independent. Each trial has two outcomes heads(success) and tails (failure). The probability of success on each trial is p=1/2 and the probability of failure is q=1−1/2=1/2.
- We are interested in the variable X which counts the number of successes in 12 trials.
- This is an example of a Bernoulli Experiment with 12 trials.

Bernoulli Examples (2)

- A basketball player takes four independent free throws with a probability of 0.7 of getting a basket on each shot.
- The number of baskets made is recorded.
- Here each free throw is a trial and trials are assumed to be independent.
- Each trial has two outcomes basket(success) or no basket(failure).
- The probability of success is $p=0.7$ and the probability of failure is

  $q=1-p = 0.3$.
- We are interested in the variable X which counts the number of successes in 4 trials.
- This is an example of a Bernoulli experiment with 4 trials.

Bernoulli Examples (3)

- An urn contains 6 red marbles and 4 blue marbles.
- Five marbles are drawn from the urn without replacement and the number of red marbles is observed.
- We might let a trial here consist of drawing a marble from the urn and let success be getting red.
- However, this is not a Bernoulli experiment since the trials are not independent (because the mix of reds and blues changes on each trial since we do not replace the marble) and the probability of success and failure varies from trial to trial.

Binomial distribution

- The binomial distribution is just *n* independent individual (Bernoulli) trials added up.
- It is the number of “successes” in *n* trials.
- The sum of the probabilities of *all* the independent trials totals 1.
- We can define a ‘success’ as a ‘1’, and a failure as a ‘0’.

Binomial distribution

- Examples (note – success/failure could be switched!):

Situation	x=1 (success)	x=0 (failure)
Coin toss to get heads	Turns up heads	Turns up tails
Rolling dice to get 1	Lands on 1	Lands on anything but 1
While testing a product, how many are found defective	Product is defective	Product is not defective

Binomial distribution

- Let us associate the events of the Bernoulli trial with a random variable *X* such that when the outcome of the trial is a success. We define *X* = 1 and when the outcome is a failure we define *X* = 0.
- The random variable *X* is called a Bernoulli random variable and its PMF is given by

$$p_X(x) = \begin{cases} 1-p & x=0 \\ p & x=1 \end{cases}$$

Coin Toss Simulation

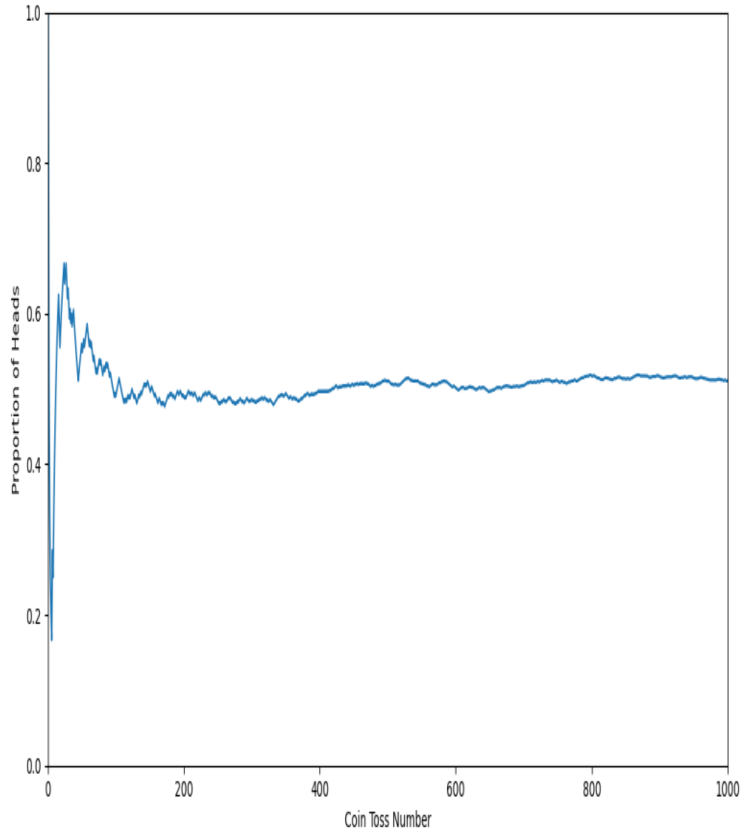
Write code using Python to simulate tossing a fair coin to see how the law of large numbers works.

```python
import numpy as np
import matplotlib.pyplot as plt

rng = np.random.default_rng()

n = 1000
U = rng.random(n)
toss = (U < 0.5)
avg = [sum(toss[:i]) / float(i) for i in range(1, n +
1)]  # Convert to float for accurate division

plt.xlabel("Coin Toss Number")
plt.ylabel("Proportion of Heads")
plt.axis([0, n, 0, 1])
plt.plot(range(1, n + 1), avg)
plt.show()
```



Binomial distribution Example

- Let's figure out a binomial random variable's probability function or formula
- Suppose we are looking at a binomial with n=3 (ex. 3 coin flips; 'heads' is a 'success')
- We will start with 'all tails' P(x=0):
  - Can happen only one way: 000
  - Which is: (1-p)(1-p)(1-p)
  - Simplified: $(1-p)^3$
- Let's figure out a binomial probability function (for n = 3)
- This time we want 1 success plus 2 failures (ex. 1 head + 2 tails, or P(x=1)):
  - This can happen three ways: 100, 010, 001
  - Which is: p(1-p)(1-p)+(1-p)p(1-p)+(1-p)(1-p)p
  - Simplified: $3p(1-p)^2$

Binomial distribution Example

- Let's figure out a binomial probability function (for $n = 3$)
- We want 2 'successes' $P(x=2)$:
  - Can happen three ways: 110, 011, 101, or…
  - pp(1-p)+(1-p)pp+p(1-p)p, which simplifies to..
  - $3p^2(1-p)$
- Let's figure out a binomial probability function ($n = 3$)
- We want all 3 'successes' $P(x=3)$:
  - This can happen only one way: 111
  - Which we represent as: ppp
  - Which simplifies to: p^3

Binomial distribution Example

- Let's figure out a binomial probability function – in summary, for *n* = 3, we have: P(*x*)

$$x = \begin{cases} 0 & (1-p)^3 \\ 1 & 3p(1-p)^2 \\ 2 & 3p^2(1-p) \\ 3 & p^3 \end{cases}$$

The sum of these expressions is the binomial distribution for n=3. The resulting equation is an example of the **Binomial Theorem**.

(Where *x* is the number of successes; ex. # of heads)

Binomial Experiment/Process

- The binomial probability refers to the probability that a binomial experiment results in exactly x successes.
- Properties
  - Repeated trials
  - Each trial has two possible outcomes: **Success** and **Failure**
  - Probability of success is constant from trial to trial (identically distributed)
  - Trials are independent (the outcome of one trial does not affect other trial outcomes)
- Main idea is:
  - You need to know how many trials are performed: **n**
  - Number of desired successes: **x**
  - Probability of an individual success per trial : **p**

Binomial Experiment/Process

- Let p denote the probability of success in a Bernoulli trial. The random variable, x, which counts the number of successes in n trials is called the **binomial random variabl**e of the n and p parameters.

- Which can be written as:

$$P(x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

or $P(x) = nC_x\, p^x(1 - p)^{n-x}$

- The above formula is often called the **general term** of the binomial distribution

Examples

- Suppose a die is tossed 5 times. What is the probability of getting exactly 2 fours?

- *Solution:* This is a binomial experiment in which the number of trials is equal to 5, the number of successes is equal to 2, and the probability of success on a single trial is 1/6 or about 0.167. Therefore, the binomial probability is:

$$b(2; 5, 0.167) = {}_5C_2 * (0.167)^2 * (0.833)^3$$

$$b(2; 5, 0.167) = 0.161$$

Stochastic Models

- Sometimes we are interested in how a random variable changes over time.
- The study of how a random variable evolves over time includes Stochastic processes.
- For example, in manufacturing facilities monitoring revenue, profit, and cost is required. Where those factors are varying over time.
- Stochastic processes help in analyze and predict those systems.

What is a Stochastic Process?

- Suppose we observe some characteristic of a system at discrete points in time.
- Let $\boldsymbol{X_t}$ be the value of the system characteristic at time t, In most situations, $\boldsymbol{X_t}$ is not known with certainty before time *t* and may be viewed as a random variable.
- **A discrete-time- stochastic process** is simply a description of the relation between the random variables $(\boldsymbol{X}_0, \boldsymbol{X}_1, \boldsymbol{X}_2, \dots\dots.)$ at discrete points of time.
- The system state provides information about those variables at specific time *t*.
- For example: system state $\boldsymbol{X_t} = \boldsymbol{Profit_t}$, means at time t, profit is recorded.
- Or we can record more than variable at specific time, $x_t = \begin{bmatrix} profit \\ cost \end{bmatrix}$

Random Processes

- Random Processes: A random process may be thought of as a process where the outcome is probabilistic (also called stochastic) rather than deterministic in nature; that is, where there is uncertainty as to the result.
- Random Process is a collection of random variables (RVs). Each has a separate function or RV for each time-step.

- Examples:
  - Tossing a die – we don’t know in advance what number will come up.
  - Flipping a coin – if you carefully enough devise an apparatus to flip the coin, it will always come up the same way. However, normal flipping by a human being can be considered a random process.
  - Shaking up a collection of balls in a hat and then pulling out one without looking.

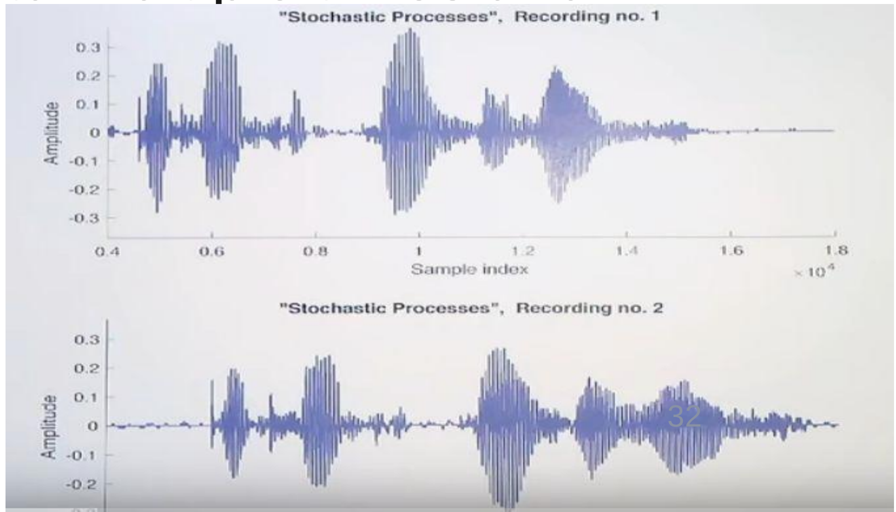
Random Variables

- In most applications, a random variable can be thought of as a variable that depends on a random process.
- Examples:
  - Toss a die and look at what number is on the side that lands up. •
    - Tossing the die is an example of a random process;
    - The number on top is the value of the random variable.
  - Toss two dice and take the sum of the numbers that land up.
    - Tossing the dice is the random process;
    - The sum is the value of the random variable.
  - Toss two dice and take the product of the numbers that land up.
    - Tossing the dice is the random process;
    - The product is the value random variable.

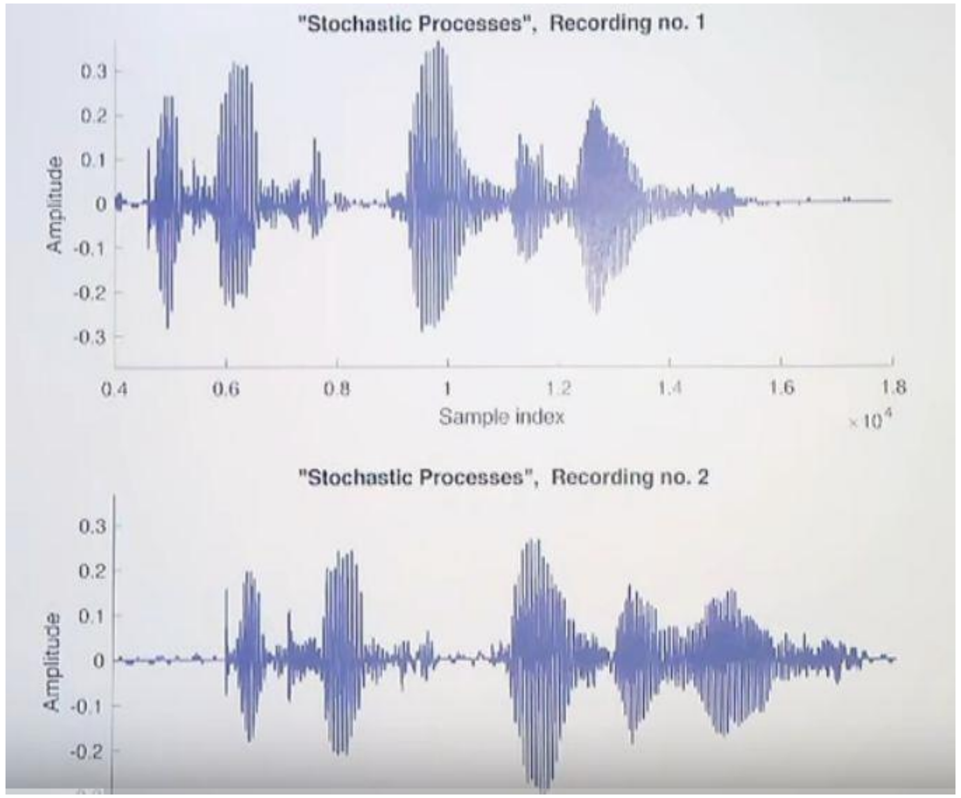
Why do we need a Random Process?

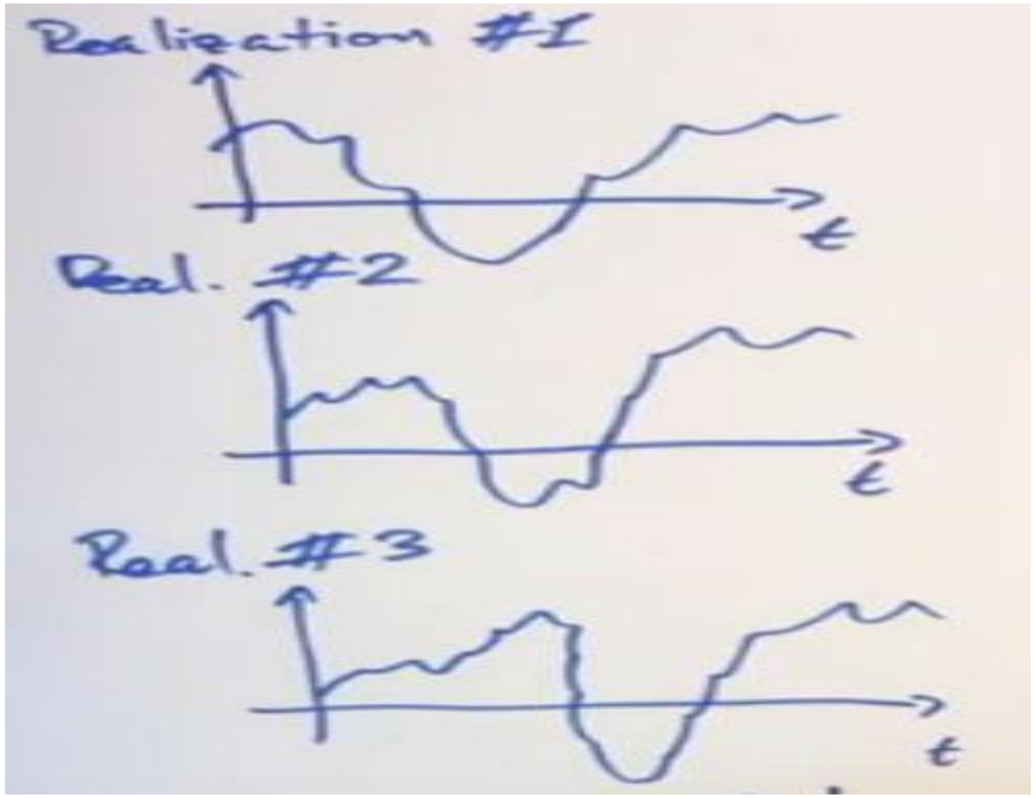
- The problem really arises when we want to model random signals e.g.
  - Speech signal,
  - received power in radio transmission
  - Accelerometer data
  - Noise Signal
  - Sound compression
  - Denoise the signal
  - Speak recognition
- For instance let's say you record some process data multiple times and wonder how can I model this behavior?
- Can I generate a formula that recreates

  something similar?



Why do we need a Random Process?





Stochastic Processes

Stochastic process	Time	Characteristics	Examples
Poisson process	continuous	events that happen independently and with a small probability per unit of time	mutations in lineage of individuals, coalescent process
Markov chain	discrete	switches between different states, with probabilities depending on previous state	nucleotide substitutions in DNA sequence evolution
Branching process	discrete or continuous	population model where each individual’s offspring number is drawn from same distribution	colonization of new habitat, spread of new disease
Wiener process	continuous	random changes in variable but with mean zero	movement of individuals in space
Random walk	discrete	describes the position taken at time t by a moving point	fluid dynamics, and quantum mechanics

Poisson Process

- You have a fixed time interval, and you ask how many times does an event occur in this interval?
- There is no bound on the occurrences or values X can take on**.**
- What you need initially is a fixed interval and average occurrences of an event during that interval.
- The Poisson distribution is used as a model in cases where the events or realizations of a process, distributed randomly in space or time, are counts, that is, discrete variables
- **Difference between Binomial vs. Poisson**
  - Binomial number of trials **n** is fixed and number of successes **x** cannot exceed **n**
  - Poisson, both number of trials and successes can be infinite

Poisson Formula

- Suppose we conduct a Poisson experiment, in which the average number of successes within a given region is **λ**. Then, the Poisson probability is:

$$X \sim Poisson(\lambda)$$
$$\mu = \lambda$$
$$\sigma^2 = \lambda, \sigma = \sqrt{\lambda}$$
$$P(X = x) = e^{-\lambda}.\frac{\lambda^x}{x!}$$

- where **λ** is the actual number of successes that result from the experiment, x is a Poisson random variable, and *e* is approximately equal to 2.71828.
- The Poisson distribution has the following properties:
  - The mean of the distribution is equal to μ.
  - The variance is also equal to μ.

Poisson Example

- **Question:** As only 3 students came to attend the class today, find the probability for exactly 4 students to attend the classes tomorrow.

- **Solution:**
  - Average rate of value(λ)= 3
  - Poisson random variable(x) = 4

$$\text{Poisson distribution} = P(X = x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

$$P(X = 4) = \frac{e^{-3}.3^{4}}{4!}$$

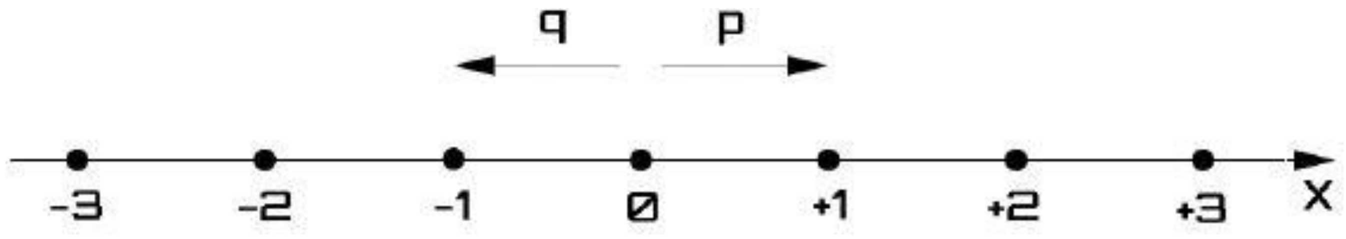
$$P(X = 4) = 0.16803135574154$$

Random walk Process

- Is a discrete parameter stochastic process in which Xt, where X represents a random variable, describes the position taken at time t by a moving point.
- This kind of simulation is extremely important for a physicist and has applications in statistical mechanics, fluid dynamics, and quantum mechanics.
- Random walks find applications in chemistry, biology, and physics, but also in other fields such as economics, sociology, and information technology
- This model can assume a variable number of degrees of freedom, depending on the system we want to describe.

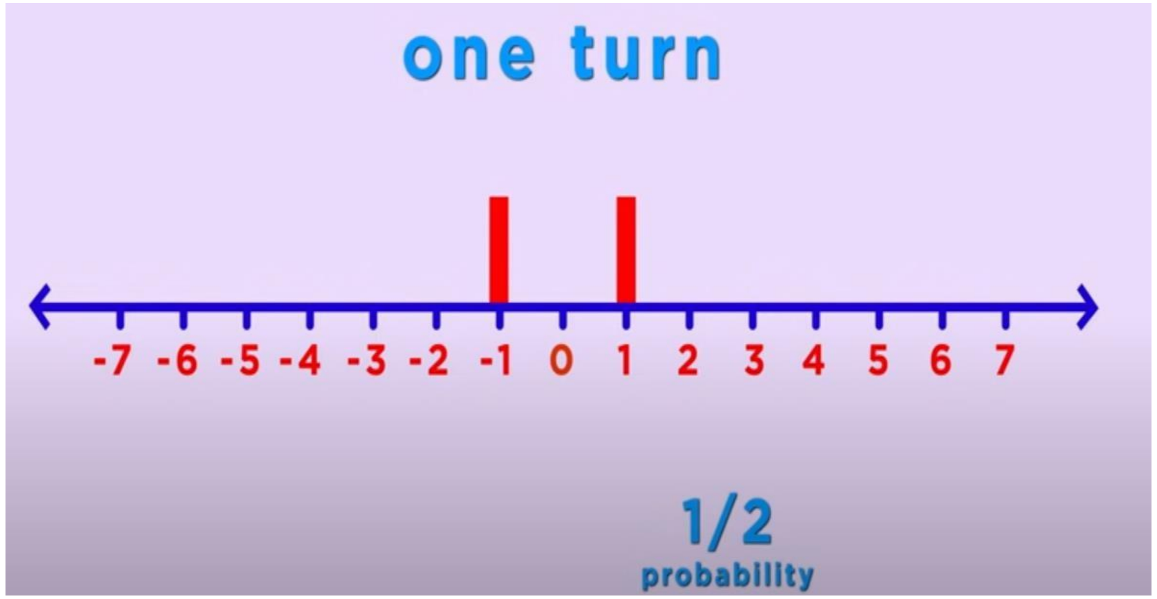
Random walk Process

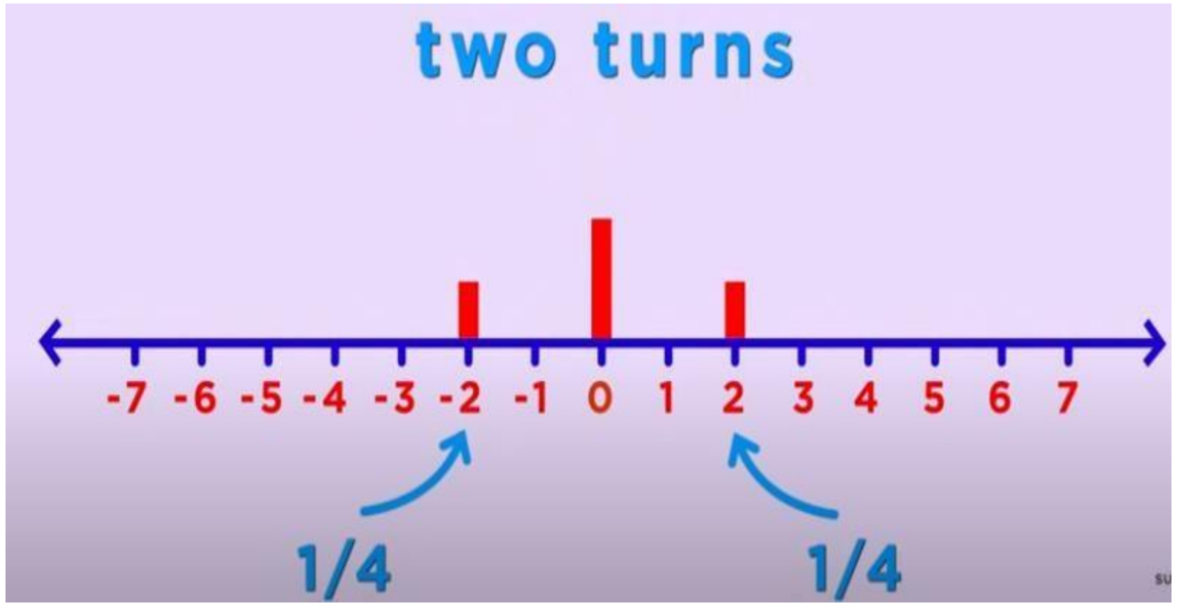
- E.g. Random one-dimensional walking is a model that is used to simulate the movement of a particle moving along a straight line. Point can move either to right with probability **p and to left with probability q** (where p + q = 1).
- Each step is constant step and independent of other steps.

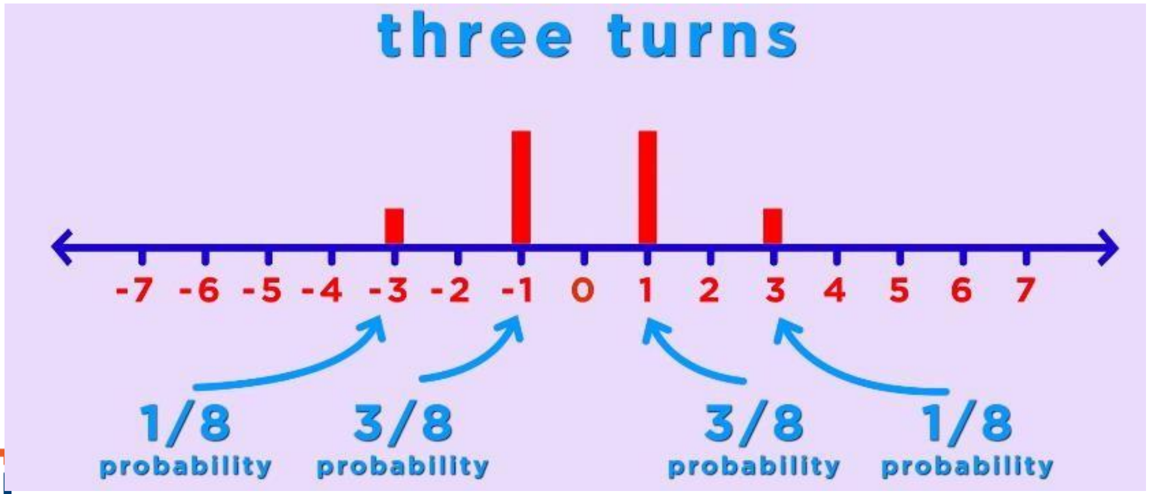


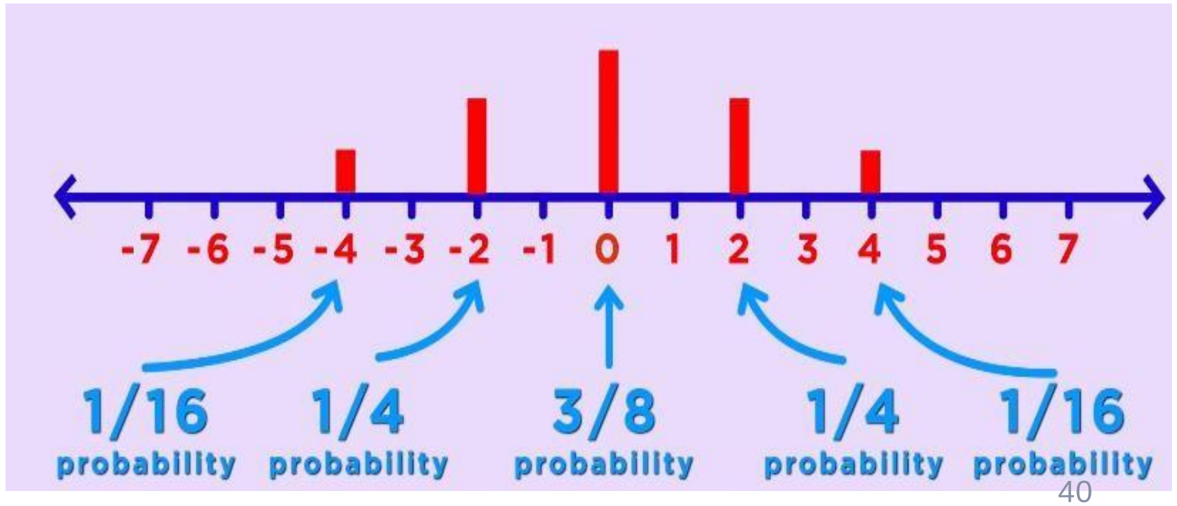
- The aim is to calculate the probability of passing from the starting point after n movements. Obviously, nothing assures us that the point will return to the starting position. The variable, X(n), returns the abscissa of the particle after n steps. It is a discrete random variable with a binomial distribution.

Example


one turn
-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
1/2
probability

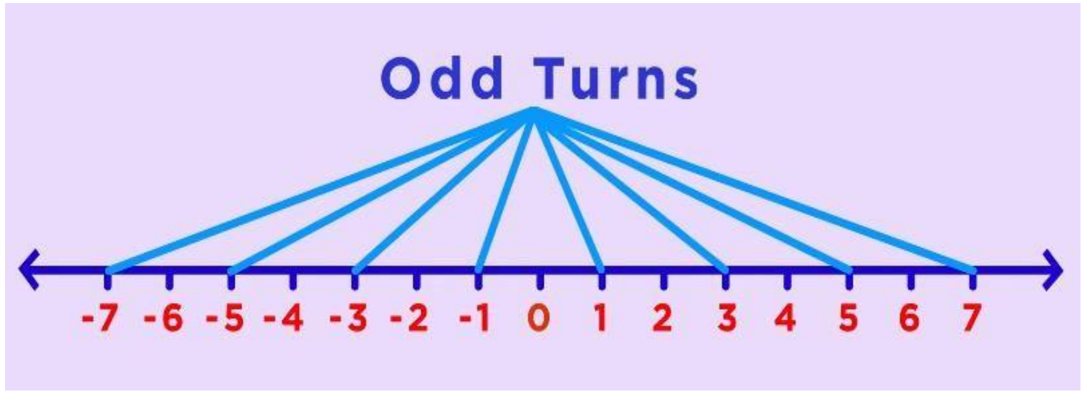

two turns
-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
1/4
1/4

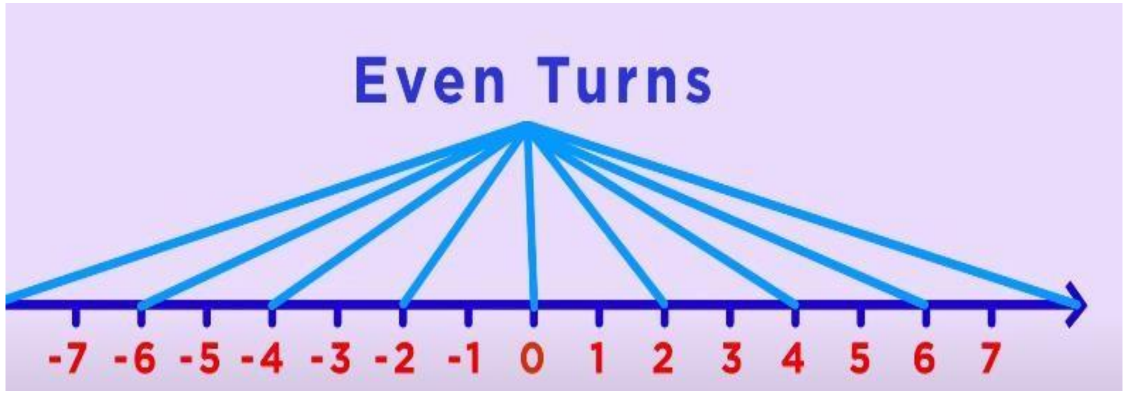

three turns
-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
1/8
probability
3/8
probability
3/8
probability
1/8
probability


-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
1/16
probability
1/4
probability
3/8
probability
1/4
probability
1/16
probability

Experiment observation

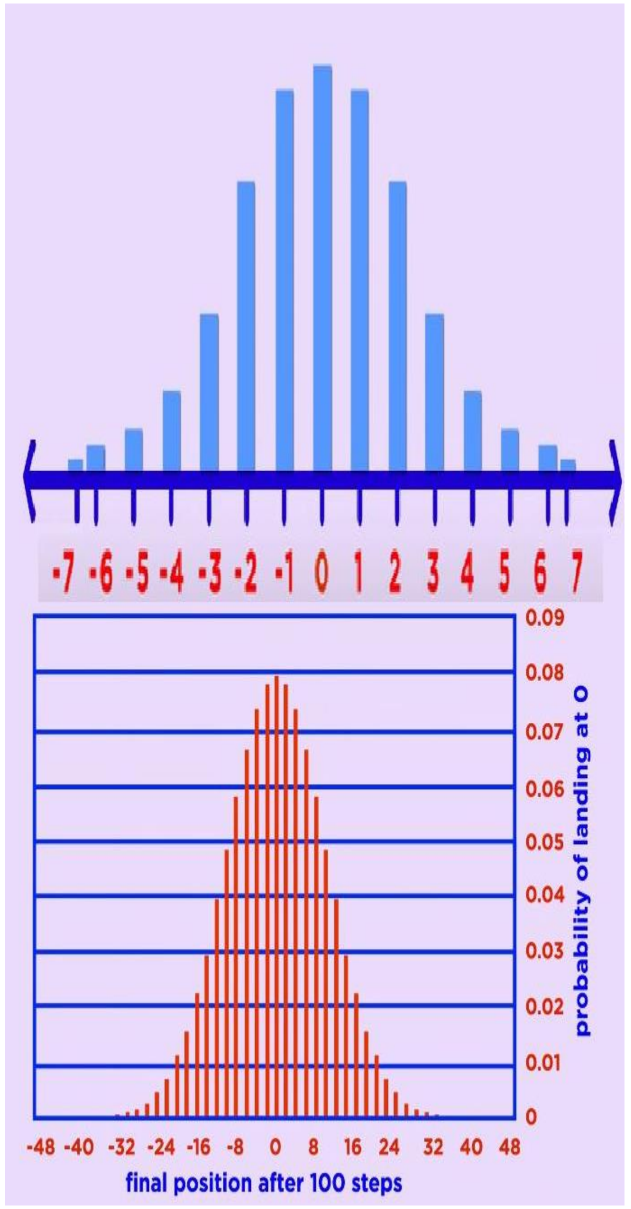
- On the odd turns the random walk can be on odd numbers.
- On the even turns the random walk can be on even numbers.



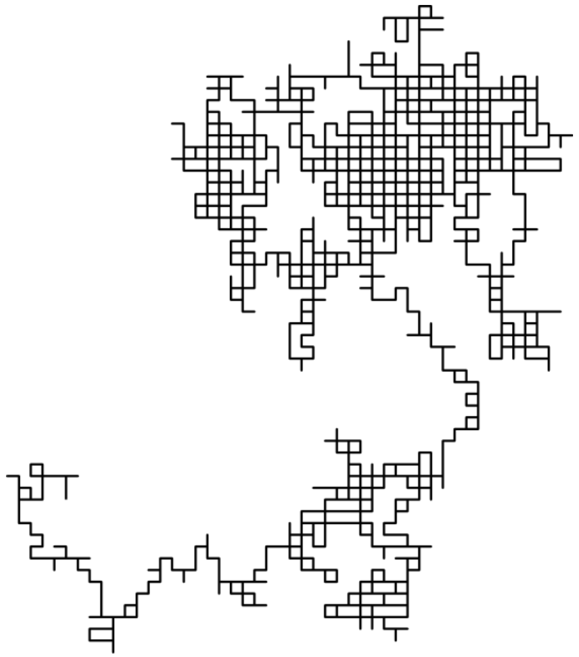


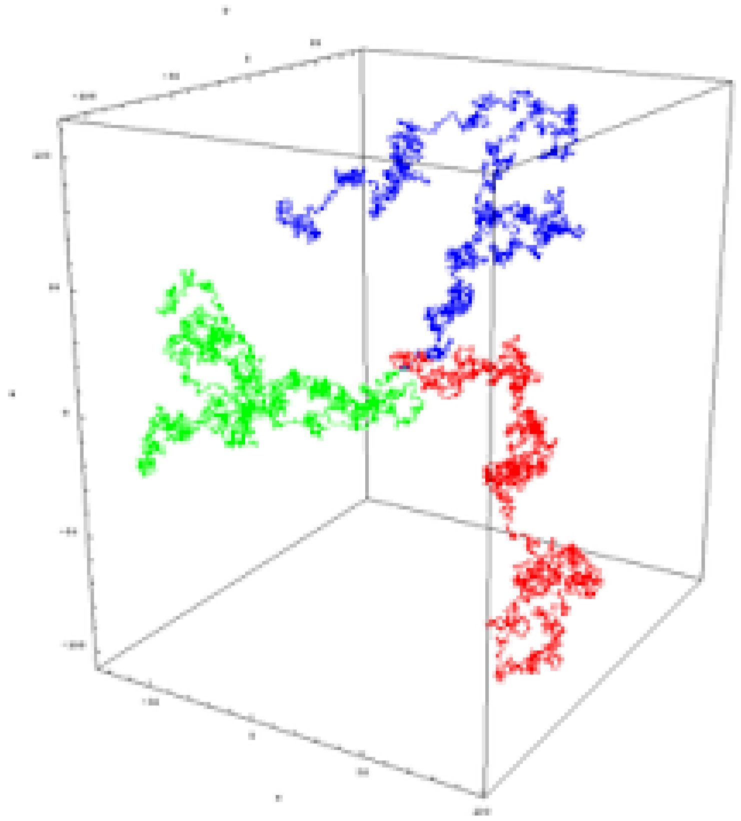
Experiment observation

- It mostly hovers around the middle, the number 0, and the longer you flip the coin for the more it can spread out.
- The probabilities of a random walk being in a specific location to form a bell curve.
- The most general version: After N coin flips , you will usually between $S_n = +\sqrt{n}\ and\ -\sqrt{n}$



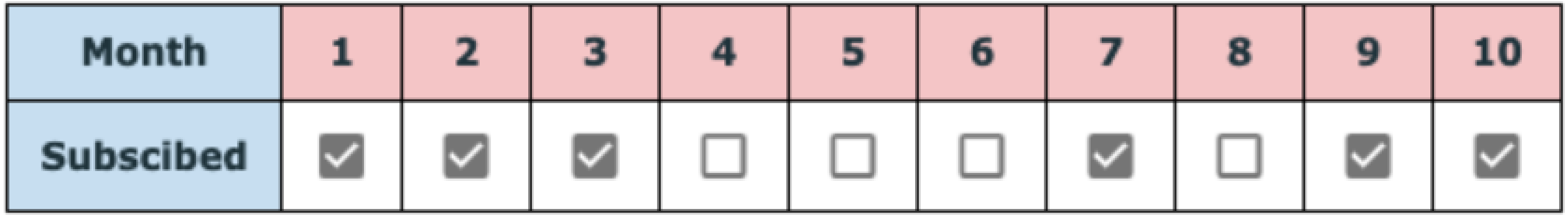
Random Walk in 2D and 3D





Markov Chain Process

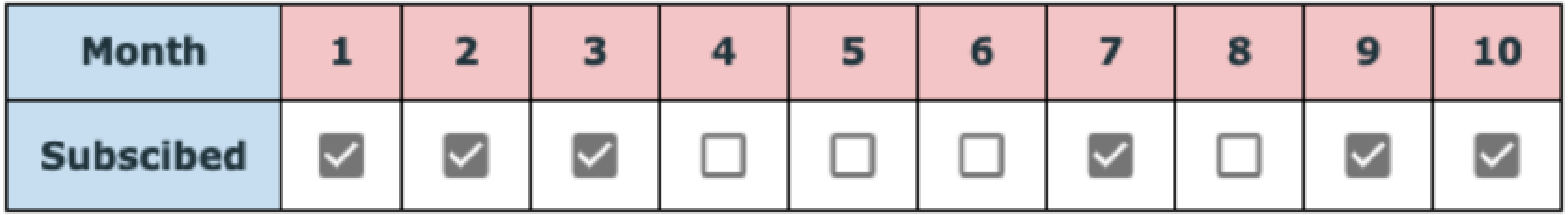
- Markov Chain is used to model a series of events.
- Each sequence usually is composed by various events and the order and the length of the sequence can vary drastically different from sample to sample.
- This chain of events are usually very hard to describe with deterministic statistics.
- For example, in a monthly subscription service, a user can choose whether they want to subscribe or not each month. One user’s subscription history may look like something like this:



Month	1	2	3	4	5	6	7	8	9	10
Subscibed	✓	✓	✓	☐	☐	☐	✓	☐	✓	✓

Markov Chain Process

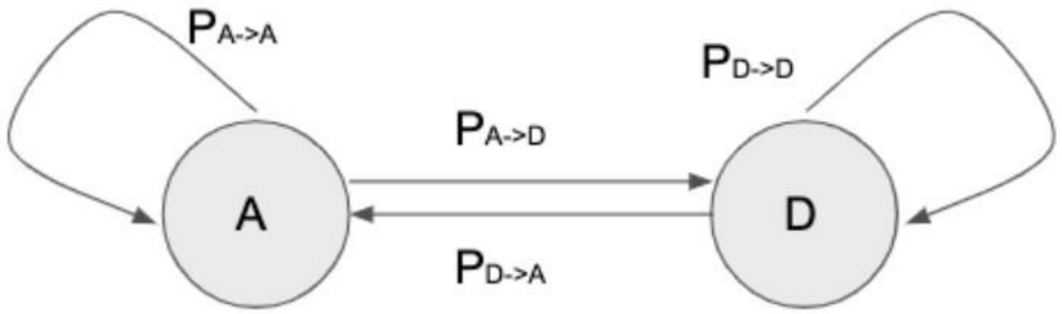
- In this example, a user have two types of events: subscribed and not subscribed.
- It is easy to see that out of 10 months, the member is active for 6 months.
- However, it is hard to describe the fact that the user has switched on and off several times, and here goes the Markov Chain Model.



Month	1	2	3	4	5	6	7	8	9	10
Subscibed	✓	✓	✓	☐	☐	☐	✓	☐	✓	✓

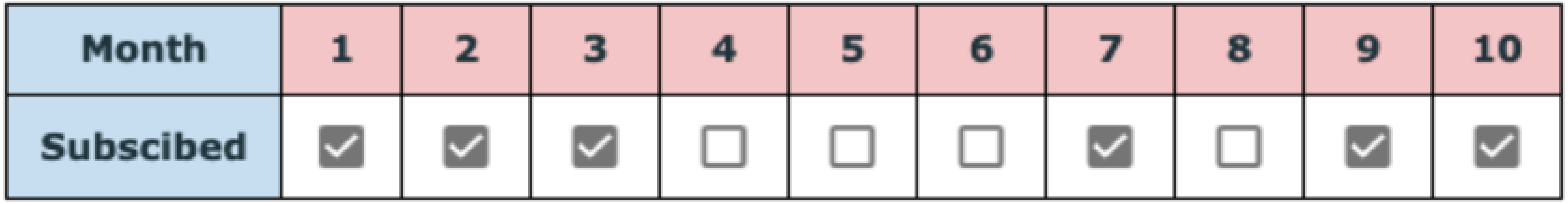
States of the Markov Chain Model

- Let us put this data into a Markov model, and it has two states:
  - Active (A) and Disabled(D).
- The probability of transition from one state to the other can be described in the following diagram.
- Specifically, if a user is at Active State, the probability of transitioning into Disabled State is P(A->D)
- and the probability of staying at Active is P(A->A).
- Since only these two options are available, the summation of P(A->D) and P(A->A) should equal to 1.

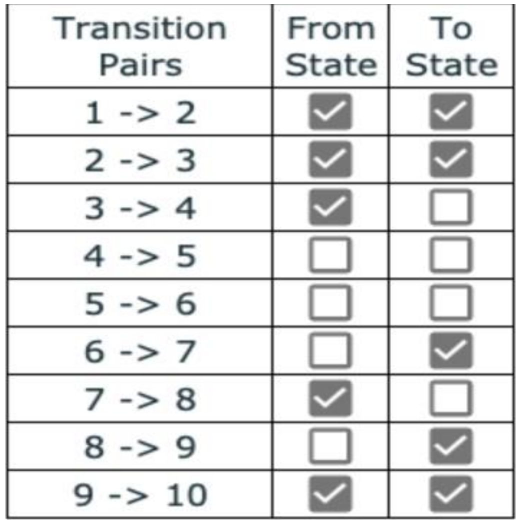


Calculate the Transition Probabilities

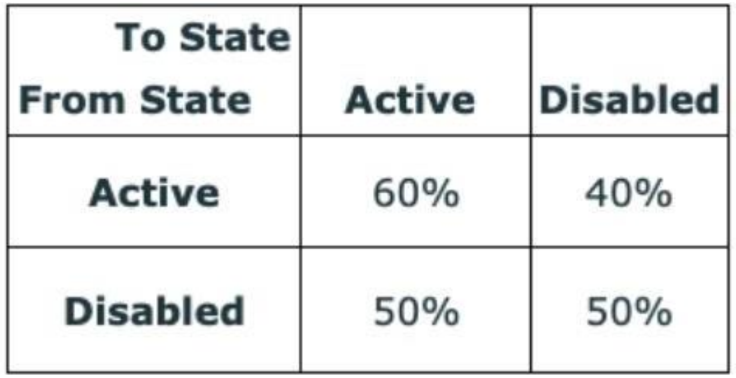
- Now let us use the data to calculate these transition probabilities. At each month, the user needs to make a decision of whether he or she wants to move from one state to the other
- Let’s transform the subscription sequence into the transition pairs:



Month	1	2	3	4	5	6	7	8	9	10
Subscibed	☑	☑	☑	☐	☐	☐	☑	☐	☑	☑



Transition Pairs	From State	To State
1 -> 2	☑	☑
2 -> 3	☑	☑
3 -> 4	☑	☐
4 -> 5	☐	☐
5 -> 6	☐	☐
6 -> 7	☐	☑
7 -> 8	☑	☐
8 -> 9	☐	☑
9 -> 10	☑	☑



To State From State	Active	Disabled
Active	60%	40%
Disabled	50%	50%

Interpretation and Prediction

- Another way of using the Markov Model is to make predictions. The questions we can answer are:
  - *For a user at the Active State right now, what is the probability of staying active in 1 month and 5 months?*
  - The calculation entails multiplication of the vector of current state ([1, 0]) and the transition matrix:

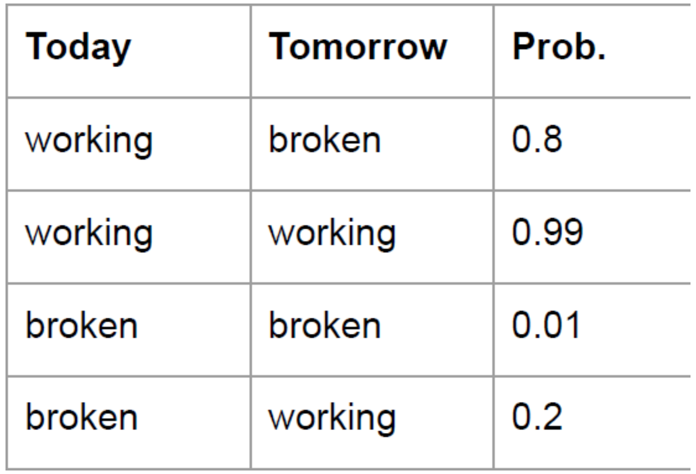
$$\text{One month from now: } \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.6 & 0.4 \end{bmatrix}$$

$$\text{Five months from now: } \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix}^5 = \begin{bmatrix} 0.56 & 0.44 \end{bmatrix}$$

  - Therefore, the probability of this member staying alive is 0.6 in one month and 0.56 in five months.

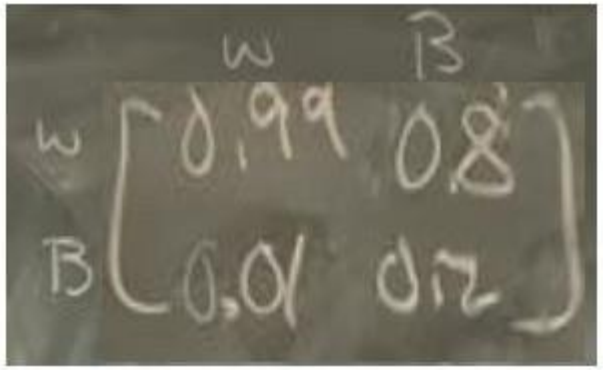
Another Example

- Machine is either broken or working on a given day , this example of a Markov chain. Many examples in practical life involve certain states e.g. here have two states : **working** , **broken** and corresponding transitions.
- So what happens , if we repeat this for a long time lets say for **10 years** , we can get the probability distribution by A^{3650}
- And p, q are prob. Of working and not working after 10 years
- What are the values of **p, q**?

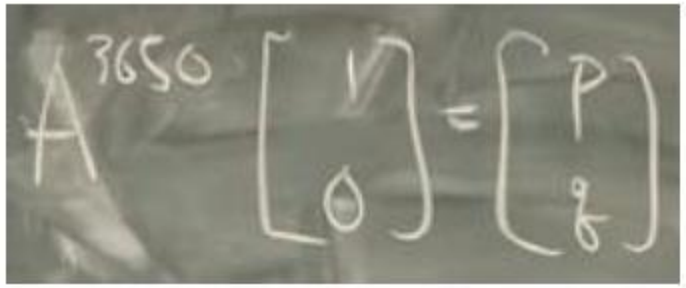


Today	Tomorrow	Prob.
working	broken	0.8
working	working	0.99
broken	broken	0.01
broken	working	0.2

Transition Matrix **A**



$$\begin{array}{c|cc} & W & B \\ \hline W & 0.99 & 0.8 \\ B & 0.01 & 0.2 \end{array}$$



$$A^{3650}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix}$$

Martingale Process

- A **martingale** is model of a fair game. It is a sequence of random variables x_0, x_1, $x_2 \ldots x_n$ with one important property: the conditional expectation of x_{n+1} given x_0, x_1, $x_2 \ldots x_n$ is always just x_n.
- In other words, it is a sequence of random variables such that for any time n:
- $E(|X_n|) < \infty$
  $E(X_{n+1} | X_1 \ldots, X_n) = X_n$
- In other word, $\mathrm{E}[X_n] = [\mathrm{X}_\mathrm{n} + 1]$, n -> 0,,,,,n-1

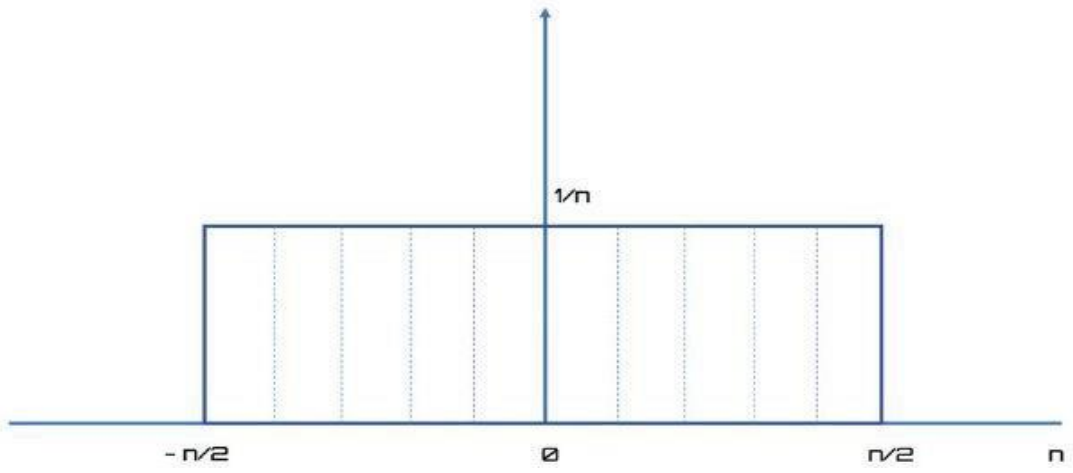
Martingale Process Example

Random Number Generator (RNG) Simulation

- The availability of random numbers is a necessary requirement in many applications.
- In some cases, the quality of the final application strictly depends on the possibility of generating good quality random numbers.
- E.g. video games, cryptography, generating visuals or sound effects, telecommunications, signal processing, optimizations, and simulations like **Monte-Carlo**
- **The concept of random number itself is not absolute**, as any number or sequence of numbers can appear to be random to an observer, but not to another who knows the law with which they are generated.
- Put simply, a random number is defined as a number selected in a random process from a finite set of numbers

Random Number Generator (RNG) Simulation

- **An ideal Random Number generator is one that uses a random variable with Uniform distribution**, such that all outcomes are equi-probable e.g. a dice is a random number generator with 6 outcomes with uniform distribution.
- We desire for Ideal RNG
  - Uniformity
  - Independence



Pseudorandom Number generation (PRNG)

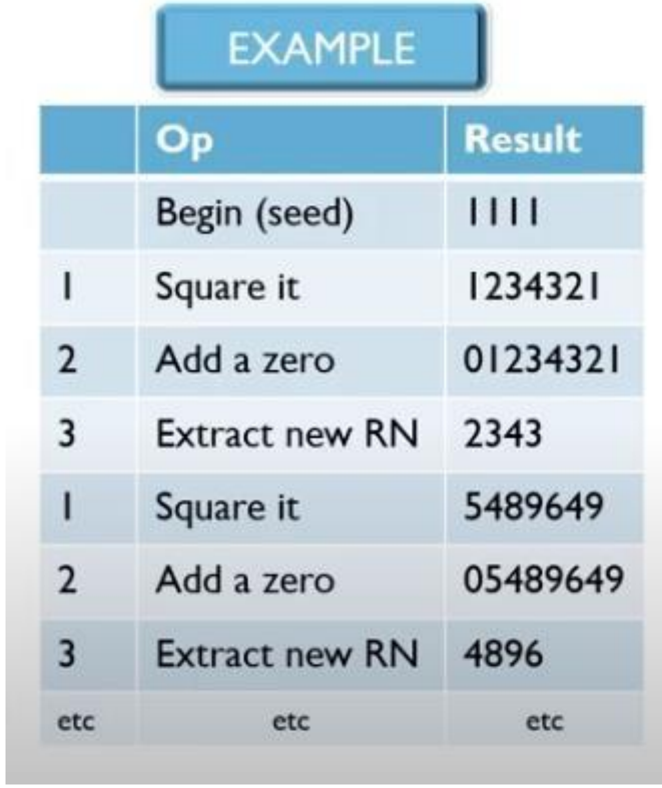
- The generation of real random sequences using deterministic algorithms is impossible: at most,
- pseudorandom sequences can be generated. These are, apparently, random sequences that are actually perfectly predictable and can be repeated after a certain number of extractions.
- A random number generation routine must be the following:
    - Replicable
    - Fast
    - Not have large gaps between two generated numbers
    - Have a sufficiently long running period
    - Generate numbers with statistical properties that are as close as possible to ideal ones

Pseudorandom Number generation (PRNG)

- The most common cons of random number generators are as follows:
  - Numbers not uniformly distributed
  - Discretization of the generated numbers
  - Incorrect mean or variance
  - Presence of cyclical variations

Example of a Simple Pseudorandom Number Generator

- John von Neumann, in 1949 proposed a method called **middle-square**.
- The algorithm is as follows:
  - Get a RN of n digit, where is even.
  - Square it, and add zeros to the left from a sequence of 2n digits.
  - Extract the middle n digits as the new RN.
  - Continue with the next iteration, etc.
- A problem with the middle-square method is that all sequences eventually repeat themselves, some very quickly(such as 0000).


EXAMPLE

	Op	Result
	Begin (seed)	1111
1	Square it	1234321
2	Add a zero	01234321
3	Extract new RN	2343
1	Square it	5489649
2	Add a zero	05489649
3	Extract new RN	4896
etc	etc	etc

Linear congruential generator (LCG)

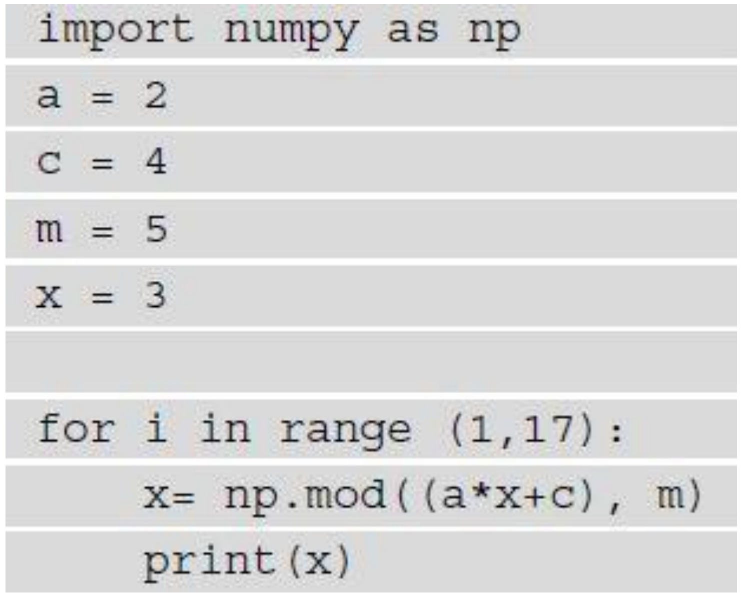
- One of the most common methods for generating random numbers is the Linear Congruence Generator (LCG).

$$x_{k+1} = (a * x_k + c) \, mod \, m$$

- where
  - **a** is the multiplier (non-negative integers)
  - **c** is the increment (non-negative integers)
  - **m** is the mode (non-negative integers)
  - **x0** is the initial value (seed or non-negative integers)
- LCG has following characteristics:
  - It is cyclical with a period that is approximately equal to m
  - The generated numbers are discretized

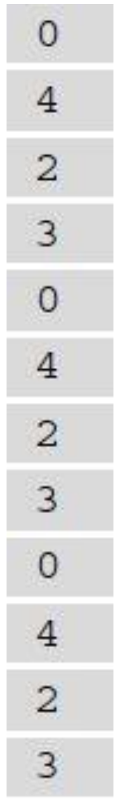
LOG code example

- In this case, the period is equal to 4. It is easy to verify that,
- at most, m distinct integers, Xn, **can be generated in the**
- **interval, [0, m - 1].**
- If **c = 0**, the generator is called multiplicative.



```
import numpy as np
a = 2
c = 4
m = 5
x = 3

for i in range (1,17):
    x= np.mod((a*x+c), m)
    print(x)
```

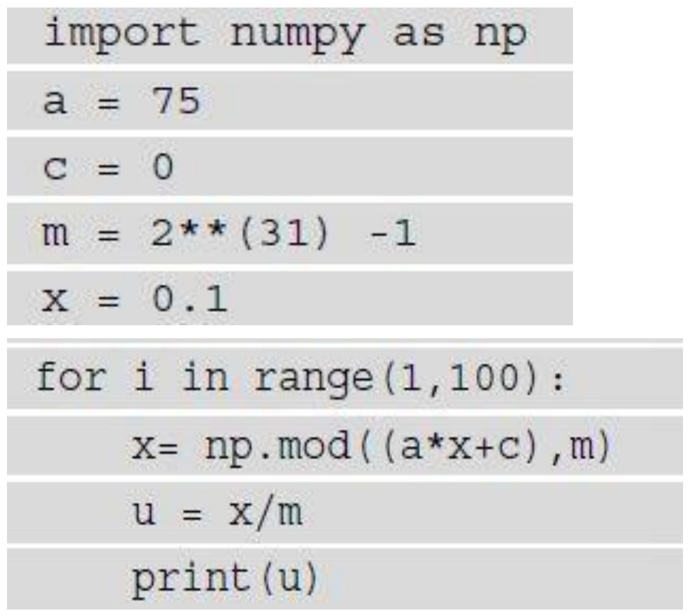


```
0
4
2
3
0
4
2
3
0
4
2
3
```

Random numbers with uniform distribution example

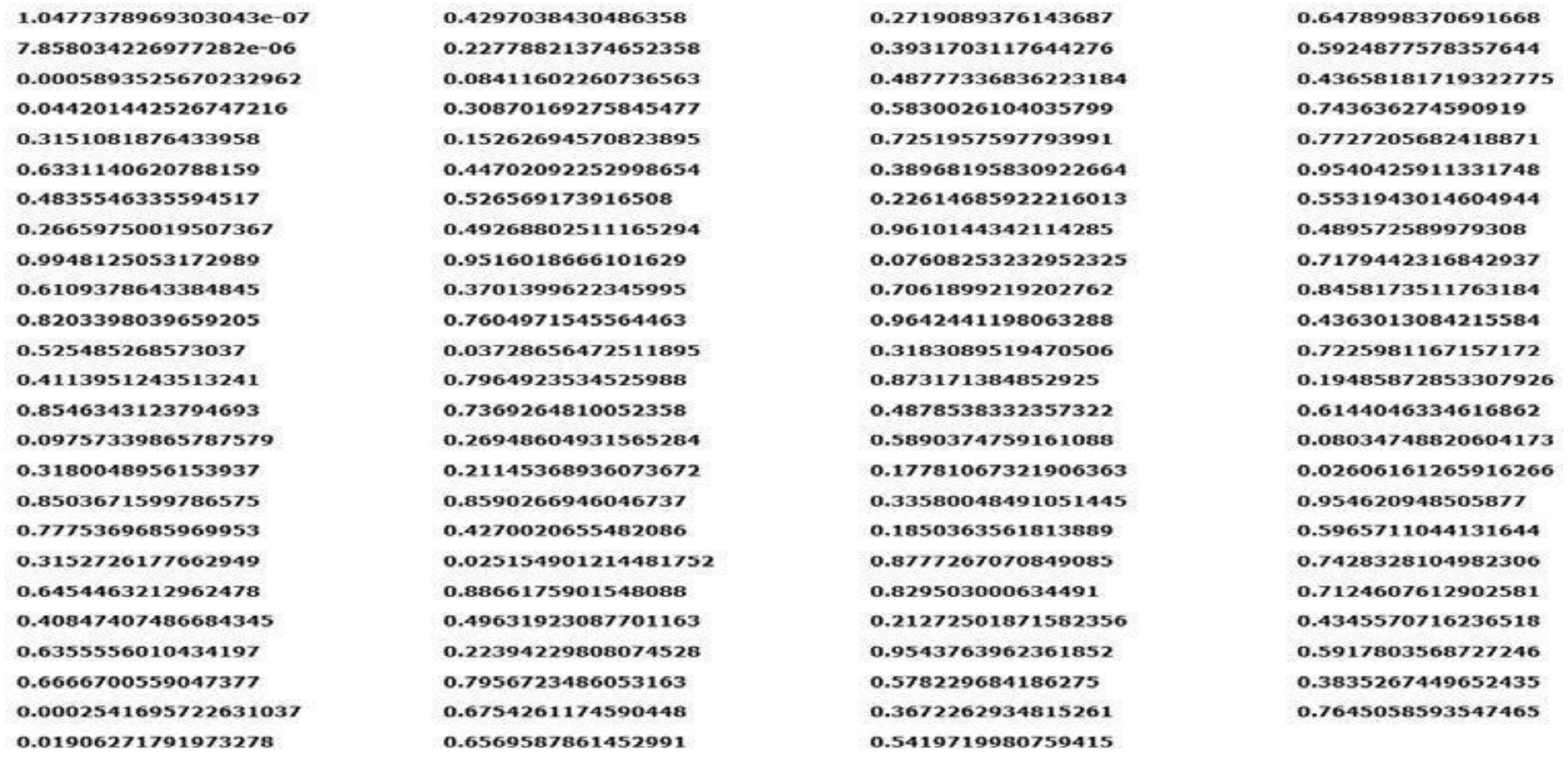
- The difference, in addition to the values of the parameters, lies in the generation of a uniform distribution in the range of

  [0, 1] through the following command: **u = x/m**



```
import numpy as np
a = 75
c = 0
m = 2**(31) -1
x = 0.1

for i in range(1,100):
    x= np.mod((a*x+c),m)
    u = x/m
    print(u)
```



```
1.0477378969303043e-07
7.858034226977282e-06
0.0005893525670232962
0.044201442526747216
0.3151081876433958
0.6331140620788159
0.4835546335594517
0.26659750019507367
0.9948125053172989
0.6109378643384845
0.8203398039659205
0.525485268573037
0.4113951243513241
0.8546343123794693
0.09757339865787579
0.3180048956153937
0.8503671599786575
0.7775369685969953
0.3152726177662949
0.6454463212962478
0.40847407486684345
0.6355556010434197
0.6666700559047377
0.00025416957226310037
0.019062717919732 78
0.4297038430486358
0.22778821374652358
0.08411602260736563
0.30870169275845477
0.15262694570823895
0.44702092252998654
0.526569173916508
0.49268802511165294
0.9516018666101629
0.3701399622345995
0.7604971545564463
0.037286564725 11895
0.7964923534525988
0.7369264810052358
0.26948604931565284
0.21145368936073672
0.8590266946046737
0.4270020655482086
0.025154901214481752
0.8866175901548088
0.49631923087701163
0.22394229808074528
0.7956723486053163
0.6754261174590448
0.6569587861452991
0.2719089376143687
0.3931703117644276
0.48777336836223184
0.5830026104035799
0.7251957597793991
0.38968195830922664
0.22614685922216013
0.9610144342114285
0.07608253232952325
0.7061899219202762
0.9642441198063288
0.3183089519470506
0.873171384852925
0.4878538332357322
0.5890374759161088
0.17781067321906363
0.33580048491051445
0.1850363561813889
0.8777267070849085
0.829503000634491
0.21272501871582356
0.9543763962361852
0.578229684186275
0.3672262934815261
0.5419719980759415
0.6478998370691668
0.5924877578357644
0.43658181719322775
0.743636274590919
0.7727205682418871
0.9540425911331748
0.5531943014604944
0.489572589979308
0.7179442316842937
0.8458173511763184
0.4363013084215584
0.7225981167157172
0.19485872853307926
0.6144046334616862
0.08034748820604173
0.02606161265916266
0.954620948505877
0.5965711044131644
0.7428328104982306
0.7124607612902581
0.4345570716236518
0.5917803568727246
0.3835267449652435
0.7645058593547465
```

Testing

- How do we know that the random numbers generated by a PRNG are random?
    - We know that they are not since they can be predicted with certainty,
    - RNs generated in this way are tested for randomness (and lack of pattern) using a several statistical tests including goodness of fit with the uniform distribution.

References

- Bossel, H., 2013. *Modelling and simulation*. Springer-Verlag.
- Carson, John S. "Introduction to modelling and simulation." *Proceedings of the Winter Simulation Conference, 2005.*. IEEE, 2005.
- https://www.sciencedirect.com/topics/mathematics/bernoulli-trial