Simulation of Probabilities

Sagar Mukherjee

Université du Luxembourg

December 8, 2019

Outline

Random Number Generation Problem

- Uniform Random Variable?
- Pseudorandom Number Generator

The Solution

- Linear Congruential Generator
- Python Implementation: LCG
- Random Variables from distribution laws

Examples

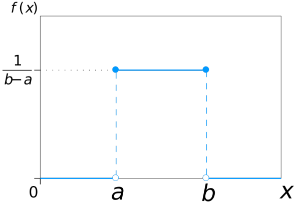
Plots

Further Improvements

Random Number Generation Problem

Uniform Random Variable?

Challenge for computer-generated random variable with uniform distribution:

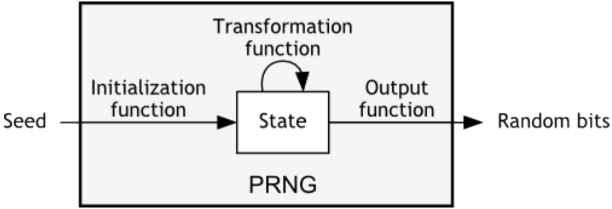


Implications:

1. Random variables from other distributions!
2. Simulate unpredictability/randomness!

Pseudorandom Number Generator

Algorithmic Flow:



The Solution

Linear Congruential Generator:

$$X_{n+1} = aX_n \mod m,$$

where X is the *sequence* of pseudorandom values, and
$m, 0 < m$ - the "modulus"
$a, 0 \leq a < m$ - the "multiplier"
$X_0, 0 \leq X_0 < m$ - the "seed" or "start value"

Uniform Random Variable Proxy:

$$X_n/m$$

Python Implementation: LCG

Random Digit

```
int(time.perf_counter_ns() / 100) % 10
```

Linear Congruential Generator

```
def RNG():
        ‘‘‘ Uniform Random Variable: [0, 1)’’’
        X = state
        state = X = (a * X) % m
        yield X/m
```

Random Variables from distribution laws

Random Variables from the following distributions have been generated:

Discrete distributions:

1. Uniform (discrete)
2. Bernoulli
3. Binomial
4. Geometric
5. Negative Binomial
6. Poisson

Continuous distributions:

1. Uniform
2. Exponential
3. Gamma
4. Normal (Gaussian)
5. Log-normal
6. Bi-variate Normal

Examples

Bernoulli Random Variable

Python: Bernoulli

```
def bernoulli(p):
        ‘‘‘Bernoulli Random Variable, parameter: p’’’
        U = next(RNG())
        if U <= p:
                X = 1
        else:
                X = 0
        return X
```

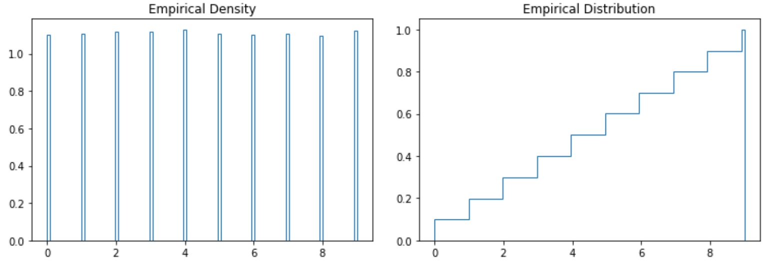
Binomial Random Variable

Python: Binomial

```
def binomial(p):
        ‘‘‘Binomial R.V. with parameters: n, p’’’
        X = 0
        for i in range(n):
                U = next(RNG())
                if U <= p:
                        X += 1
        return X
```

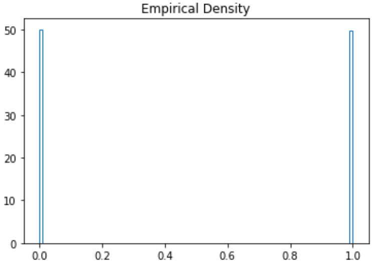
Plots

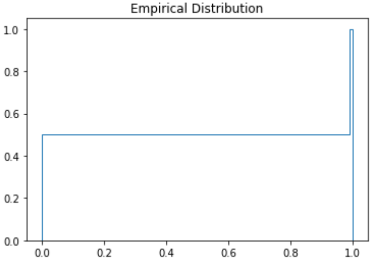
Discrete Uniform Random Variable:



Plots

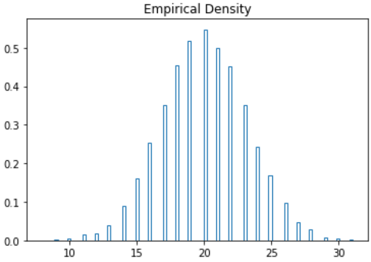
Bernoulli Random Variable:

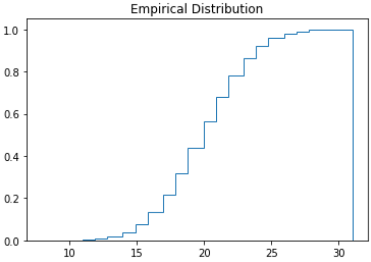




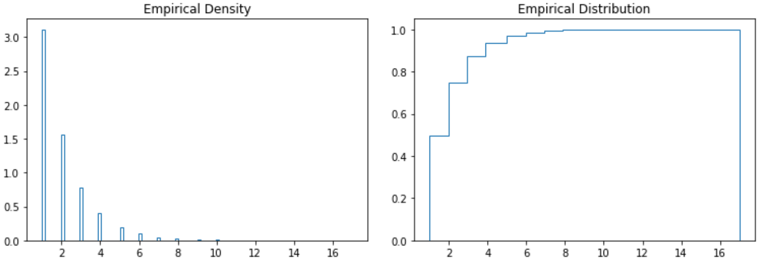
Plots

Binomial Random Variable:



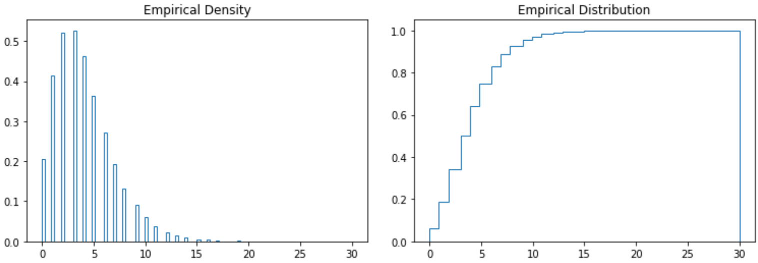


Geometric Random Variable:



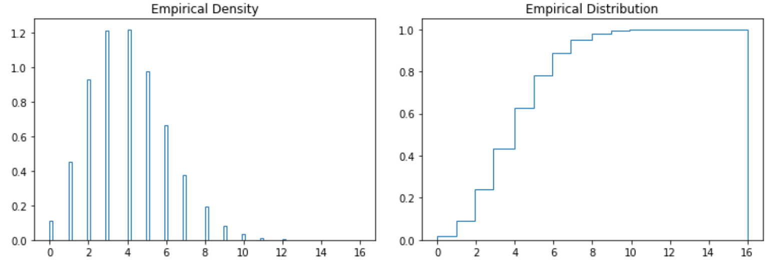
Plots

Negative-binomial Random Variable:



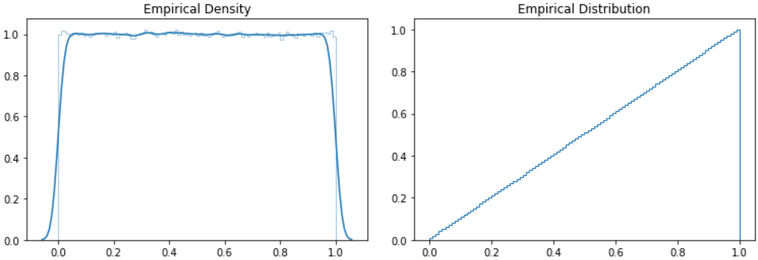
Plots

Poisson Random Variable:



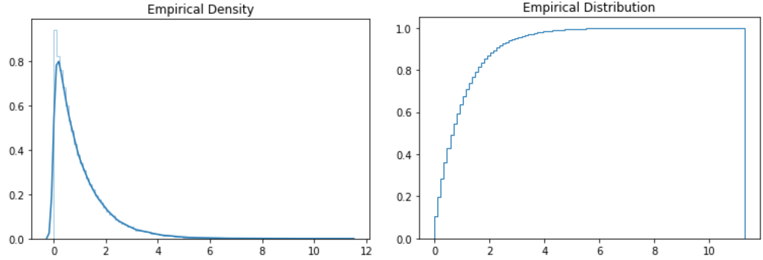
Plots

Uniform Random Variable:



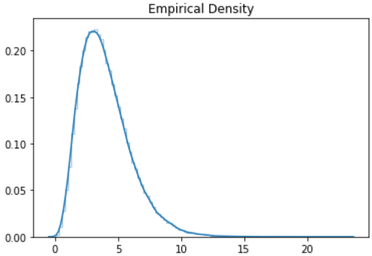
Plots

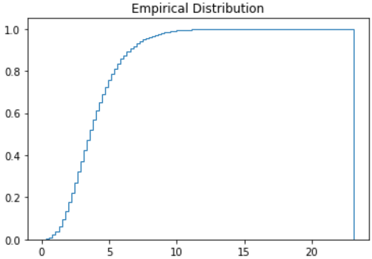
Exponential Random Variable:



Plots

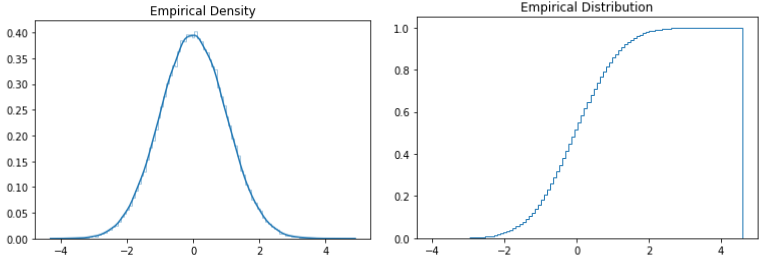
Gamma Random Variable:





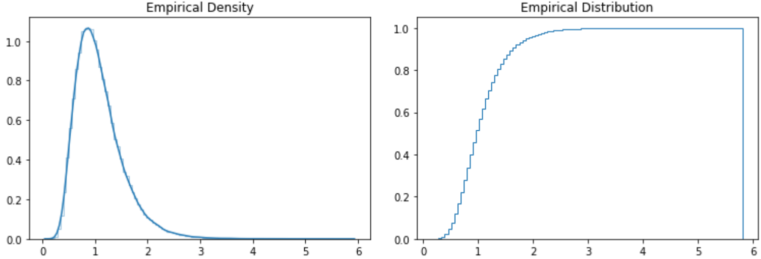
Plots

Normal/Gaussian Random Variable:



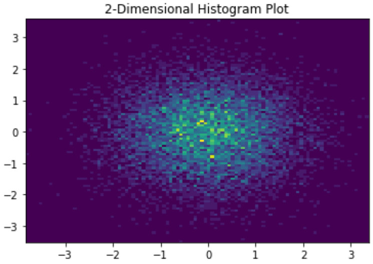
Plots

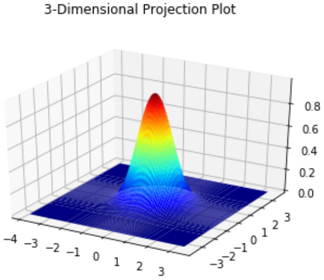
Log-normal Random Variable:



Plots

Bi-variate Normal Random Variable:





Further Improvements

Further Improvements

Some of the ways this project can be improved/developed further:

1. Random Number Generator Algorithm
2. More Distributions
3. Faster implementations

Thanks!