Modeling and Simulation

Dr. Belkacem KHALDI
b.khaldi@esi-sba.dz

ESI-SBA, Algeria

Level: 2nd SC Class
Date:February 26, 2024

Lecture 3:

Probabilities and Random Number Simulation

Part I

Probabilities simulation

Probabilities and Random Number Simulation

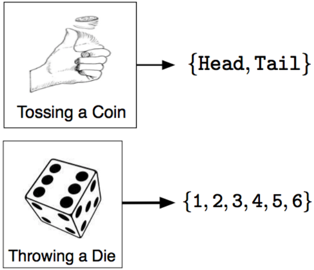
Why Studying Probabilities?

- Probability is the science of uncertainty.
- It is used to study situations whose outcomes are unpredictable.
- Needed mainly to understand how to:
    - model a probabilistic system,
    - validate the simulation model,
    - choose the input probability distributions,
    - generate random samples from these distributions,
    - perform statistical analyses of the simulation output data, and
    - design the simulation experiments.

Probabilities and Random Number Simulation

Random Experiments and Events

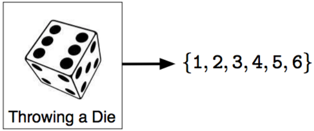
- **A random experiment** is an experiment that can result in a different outcome, even though the experiment is repeated under the same conditions.
- e.g.(1): **Tossing a coin**
  - Two possible outcomes for this experiment: **Head** and **Tail**.
- e.g.(2): **Throwing a die**
  - Six possible outcomes for this experiment: **1,2,3,4,5,6**.
- The set of all possible outcomes is referred to as **the sample space**, denoted by Ω.
  - For the experiment of tossing a coin, the sample space, $\Omega = \{Head, Tail\}$
  - For the experiment of throwing a die, the sample space, $\Omega = \{1, 2, 3, 4, 5, 6\}$



Probabilities and Random Number Simulation

Random Experiments and Events

- **An event** occurs whenever any of its outcomes are observed. It is a set of possible outcomes taken from Ω
- e.g.: The following are some events that can be defined for the random experiment of throwing a die:
  - $E_1 = \{$ One is Observed $\} = \{1\}$,
  - $E_2 = \{$ A number < 4 is observed $\} = \{1, 2, 3\}$,
  - $E_3 = \{$ A number ≥ 5 is observed $\} = \{5, 6\}$.
- The event E_1 occurs whenever the outcome 1 is observed.
- Similarly, the event E_3 occurs if the outcome 5 or 6 is observed. (both outcomes cannot be observed at the same time).



Random Experiments and Events

Basic set operations summarized in terms of events:

- The **union** of events A and B, denoted as $A \cup B$, is a new set that contains all elements from A and all elements from B (or both).

$$A \cup B = \{x : x \in A \text{ OR } x \in B\}$$

- The **intersection** of events A and B, denoted as $A \cap B$, is the set of outcomes that are common in A and B.

$$A \cap B = \{x : x \in A \text{ AND } x \in B\}$$

- The **complement** of event A, denoted as A', is the set of outcomes in Ω that are not in A.

$$A0 = \{x : x \notin A\}$$

Probability

- **Probability** is used to quantify the likelihood, or chance, that an outcome of a random experiment will occur.
- Assigned to each possible outcome in the sample space Ω.
- Probabilities must be assigned to outcomes in such a way that they add up to one.
- The probability of each outcome can be computed as follows: $P(X_i) = \frac{1}{|\Omega_i|}, X_i \in \Omega$. where $\mid \Omega_i \mid$ is the size of the sample space.
- The probability of an event is simply the sum of the probabilities of the individual outcomes making up the event.

$$P(X_i) = \frac{1}{|\Omega_i|}, X_i \in \Omega.$$

- e.g: For the experiment of tossing a coin, $P(X_i = Head) = \frac{1}{2}$
- e.g: For the experiment of throwing a die, $P(X_i = 1) = \frac{1}{6}$

Assigning Probabilities to Outcomes and Events

The following conditions must be satisfied in order to have a valid probability assignment:

- For each outcome $X_i \in \Omega$,

$$P(X_i) \in [0, 1], \qquad (1)$$

- For all outcomes $X \in \Omega$,

$$\sum_i P(X_i) = 1, \qquad (2)$$

- For each event $E_j \subseteq \Omega$,

$$P(E_j) = \sum_i P(X_i), \qquad (3)$$

  where $X_i \in E_j$,

- For all possible disjoint events $E_j \subseteq \Omega$,

$$P(\bigcup_j E_j) = \sum_j P(E_j). \qquad (4)$$

Complementary events

Complementary events are two events – usually referred to as E and E' – that are mutually exclusive.

- For example in the rolling some dice experiment:
  - Given the event $E = \{$ number 5 comes out$\}$.
  - The complementary event will be $E' = \{$ number 5 does not come out$\}$.
- E and E' are mutually exclusive because the two events cannot happen simultaneously.
- They are exhaustive because the sum of their probabilities is 1.

- $P(E) = \frac{1}{6}$
- $P(E') = \frac{5}{6}$
- $P(E)+P(E') = \frac{1}{6}+\frac{5}{6} = 1$

Bayes' theorem

Bayes' theorem finds the probability of an event occurring given the probability of another event that has already occurred.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{5}$$

where A and B are events and $P(B) \neq 0$

- $P(A \mid B)$: the likelihood of event A occurring given that B is true.
- $P(B \mid A)$: the likelihood of event B occurring given that A is true.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B respectively.

Example:

- dangerous fires are rare (1%)
- but smoke is fairly common (10%) due to barbecues,
- and 90% of dangerous fires make smoke

Therefore, **the probability of dangerous Fire when there is Smoke:**

$P(Fire|Smoke) = \frac{P(Fire)*P(Smoke|Fire)}{P(Smoke)}$
$P(Fire|Smoke) = \frac{0.01*0.9}{0.1}$
$P(Fire|Smoke) = 0.09$

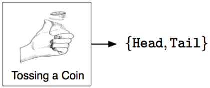
Relative frequency and probability

The relative frequency $f(E)$ of an event subjected to n experiments, all carried out under the same conditions, is the ratio between the number v of the times the event occurred and the number n of tests carried out:

$$f(E) = \frac{v}{n} \tag{6}$$

- If we consider the toss of a coin and the event $E = \{$ Head up$\}$:
  - Theoretical probability gives us: $P(E) = \frac{1}{2}$
- If we perform many throws, we will see that the number of times the coin landed heads up is almost equal to the number of times a cross occurs. That is, the relative frequency of the event E approaches the theoretical value:

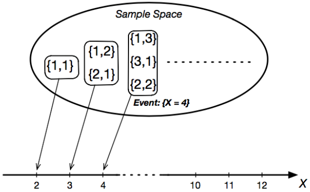
$$f(E) \cong P(E) = \frac{1}{2} \tag{7}$$



Random Variables

A **random variable** is a variable whose possible values are numerical outcomes of a random event.

- e.g.: Throwing two fair dice, one possible event is that the two dice show up $\{\mathbf{1{,}1}\}$.
- Given $\mathbf{X}$ the random variable corresponding to the sum of the two dice, then this event is represented as $\{\mathbf{X = 2}\}$.
- The range of a random variable is a probability (e.g., $P[X = 2] = \frac{1}{36}$)
- The Sample space for this experiment is: $\Omega = \{(1,1),(1,2),...,(6,6)\}$
- The outcome of the experiment is a random variable $X \in \{2, 3, ..., 12\}$.

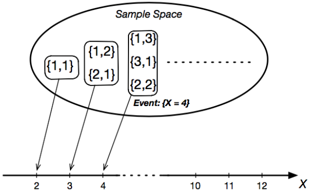


Probabilities and Random Number Simulation

Random Variables (continue)

A **Random Variable** can be either **Discrete** or **Continuous**:

- A **Discrete variable** can only take certain values (such as $1, 2, 3, 4, 5$)
- A **Continuous variable** can take any value within a range (such as a person's height)



Probability Distributions

- A **Probability Distribution** of a random variable X is a description of the probabilities associated with the possible values of X.
- Formally, let X be a random variable and let x be a possible value of X. Then, we have two types of **probability distributions**:
  - **Discrete Probability Distributions:**
    - The **probability mass function** of X specifies $P(x) \equiv P(X = x)$ for all possible values of x
  - **Continuous Probability Distributions:**
    - The **probability density function** of X is a function $f(x)$ that is such that $f(x) \approx P(x < X \leq x + h)$ for small positive h.
- **Basic Concept:** The **probability mass function** specifies the actual probability, while the **probability density function** specifies the probability rate.

Probabilities and Random Number Simulation

Discrete Probability Distributions

A **probability mass function** must satisfy the following two requirements:

1. $0 \leq P(x) \leq 1$ for all x
2. $\sum_{\text{all } x} P(x) = 1$

Empirical data can be used to estimate the probability mass function. Consider, for example, the number of TVs in a household.

No. of TVs	**No. of Households**	x	$P(x)$
0	1,218	0	0.012
1	32,379	1	0.319
2	37,961	2	0.374
3	19,387	3	0.191
4	7,714	4	0.076
5	2,842	5	0.028
	101,501		1.000

For $x = 0$, the probability 0.012 comes from $1,218/101,501$. Other probabilities are estimated similarly.

Discrete Probability Distributions — Mean and Variance

The population **mean or expected value** of a discrete random variable X, denoted as μ or $E(X)$, is

$$\mu = E(X) = \sum_{\text{all } x} x.f(x) = \sum_{\text{all } x} x.P(x)$$

Example: the number of TVs in a household

$$\mu = 0x0.012 + 1x0.319 + ...5x0.028 = 2.084$$

No. of TVs	No. of Households	x	$P(x)$
0	1,218	0	0.012
1	32,379	1	0.319
2	37,961	2	0.374
3	19,387	3	0.191
4	7,714	4	0.076
5	2,842	5	0.028
	101,501		1.000

Discrete Probability Distributions — Mean and Variance

The population **variance** is calculated similarly. It is the weighted average of the squared deviations from the mean. Formally,

$$\sigma^2 = \sum_{\text{all } x} (x-\mu)^2 . P(x)$$

Example: the number of TVs in a household

$\sigma^2 = (0 - 2.084)^2 x 0.012 + ... + (5 - 2.084)^2 = 1.107$

No. of TVs	No. of Households	x	$P(x)$
0	1,218	0	0.012
1	32,379	1	0.319
2	37,961	2	0.374
3	19,387	3	0.191
4	7,714	4	0.076
5	2,842	5	0.028
	101,501		1.000

Continuous Probability Distributions

- For a **continuous random variable** X, the range of X includes all values in an interval of real numbers. This could be an infinite interval such as $(-\infty, \infty)$.
- Formally we describe the probability distribution with a smooth curve called a **probability density function** $f(x)$.
- **Example:** How likely is it that X falls between 0.18 and 0.22 in Fig.1?
- If $f(x)$ is a known function then we could answer this question through integration:

$$P(0.18 \leq x \leq 0.22) = \int_{0.18}^{0.22} f(x)\, dx$$

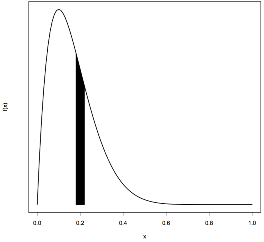


Figure 1: Probability Density Function

Continuous Probability Distributions

Definition (Probability Density Function): For a continuous random variable X, a probability density function is a function such that:

1. $f(x) \geq 0$
2. $\int_{-\infty}^{\infty} f(x)\, dx = 1$
3. $P(a \leq x \leq b) = \int_{a}^{b} f(x)\, dx$

Discrete Probability Distributions — Mean and Variance

The **mean or expected value** of a continuous random variable X, denoted as μ or $E(X)$, is:

$$\mu = E(X) = \int_{-\infty}^{\infty} xf(x)\,dx$$

The **variance** of X, denoted as $V(X)$ or σ^2, is:

$$\sigma^2 = V(X) = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)\,dx$$

Common Special Discrete Probability Distributions

- **The Binomial Distribution:** describes the number of successes in t independent trials with prob. p of success on each trial.

Mass Function:	$p(x)=\begin{cases} \binom{t}{x} p^t (1-p)^{t-x}, & \text{if } x \in \{0,1,..t\} \\ 0, & \text{otherwise} \end{cases}$
Mean (μ) and Variance: (σ^2)	$\mu = tp$, $\sigma^2 = tp(1-p)$

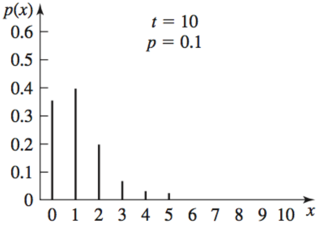


Figure 2: Binomial Mass Function

Probabilities and Random Number Simulation

Common Special Discrete Probability Distributions

- **The Geometric Distribution:** defines number of failures before the 1st success in a sequence of independent trials with probability p of success on each trial.

Mass Function:	$p(x) = \begin{cases} p(1-p)^{x-1}, & \text{if } x \in \{1, ..\} \\ 0, & \text{otherwise} \end{cases}$
Mean (μ) and Variance: (σ^2)	$\mu = \frac{1-p}{p}$, $\sigma^2 = \frac{1-p}{p^2}$

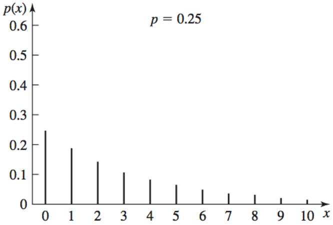


Figure 3: Geometric Mass Function with $p = 0.25$

Common Special Discrete Probability Distributions

- **The Poisson Distribution:** defines number of events that occur in an interval of time when the events are occurring at a constant rate.

Mass Function:	$p(x) = \begin{cases} \frac{e^{-\lambda}\lambda^x}{x!}, & \text{if } x \in \{0, 1, ..\},\ e = 2.72 \\ 0, & \text{otherwise} \end{cases}$
Mean (μ) and Variance: (σ^2)	$\mu = \lambda$, $\sigma^2 = \lambda$

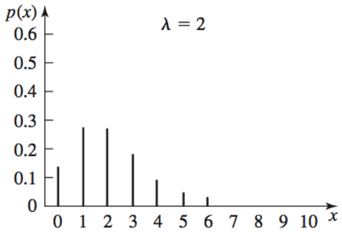


Figure 4: Poisson Mass Function with $\lambda = 2$

Common Special Continuous Probability Distributions

- **The Uniform Distribution:** Used to model a quantity that is felt to be randomly varying between a and b.

Density Function:	$f(x) = \begin{cases} \frac{1}{b-a}, & \text{if } a \leq x \leq b \\ 0, & \text{otherwise} \end{cases}$
Mean (μ) and Variance: (σ^2)	$\mu = \frac{a+b}{2}$, $\sigma^2 = \frac{(b-a)^2}{12}$

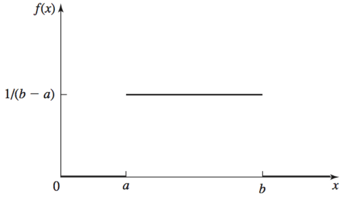


Figure 5: Uniform density Function

Common Special Continuous Probability Distributions

- **The Exponential Distribution:** Used to model the time between the occurrences of two consecutive events.

Density Function:	$f(x)=\begin{cases} \frac{1}{\beta}e^{-x/\beta}, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$
Mean (μ) and Variance: (σ^2)	$\mu = \beta$, $\sigma^2 = \beta^2$

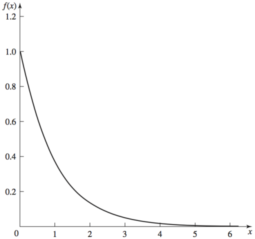


Figure 6: Exponential density Function with $\beta = 1$

Probabilities and Random Number Simulation

Common Special Continuous Probability Distributions

- **The Gaussian (Normal) Distribution:** The most used distribution in data science that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean.

Density Function:	$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$, for all real numbers x
Mean (μ) and Variance: (σ^2)	$\mu = \mu$, $\sigma^2 = \sigma^2$

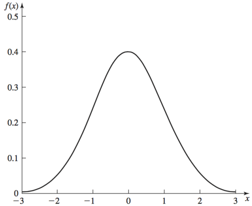


Figure 7: Exponential density Function with $\mu = 0$ and $\sigma = 1$

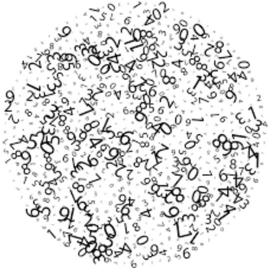
Part II

Random number simulation

Probabilities and Random Number Simulation

Random Number Simulation

- A **random number**: a number drawn in a random process from a finite set of numbers.
- **Random numbers** are useful for a variety of purposes such as:
  - Generating data encryption keys,
  - Simulating and modeling complex phenomena
  - Selecting random samples from larger data sets.
  - Aesthetically, for example in literature and music,
  - Popular for games and gambling.



Random Number Generation Using Python

- In Python, there is a specific module for the generation of random numbers: this is the **random module**.
- The **random module** implements an algorithm called **PRNGs** for various distributions.
- It is important to note that random numbers are generated using repeatable and predictable deterministic algorithms.
- They begin with a certain **seed** value and, every time we ask for a new number, we get one based on the current seed.
- The **seed** is an attribute of the generator. If we invoke the generator twice with the same **seed**, the sequence of numbers that will be generated starting from that **seed** will always be the same.

Random Number Generation Using Python

The random.random() function

- The *random.random*() function returns the next nearest floating-point value from the generated sequence.
- All return values are enclosed between 0 and 1.0.
- Example:

```
import random
for i in range(20):
    print('%05.4f' % random.random(), end=' ')
```

```
0.7603 0.3264 0.1094 0.2749 0.9305 0.1285 0.8310 0.4916 0.2794 0.6723 0.1805 0.0530 0.1874 0.6098 0.2236 0.0847 0.298
7 0.0709 0.0291 0.1021
```

As you can see, the numbers are uniformly distributed in the range of $[0, 1]$.
By running the code repeatedly, you get sequences of different numbers:

```
0.6602 0.7143 0.2588 0.8199 0.0540 0.1624 0.7515 0.0484 0.5843 0.4615 0.7889 0.4589 0.3354 0.3569 0.1200 0.4766 0.904
4 0.6849 0.6609 0.8250
```

Random Number Generation Using Python

The random.seed() function

- In sometimes it is useful to have the same set of data available to be processed in different ways. To do this, we can use the *random.seed*() function.
- The *random.seed*() function The seed setting is particularly useful when you want to make the simulation repeatable.
- Example:

```
import random

random.seed(1)
for i in range(20):
    print('%05.4f' % random.random(), end=' ')
```

```
0.1344 0.8474 0.7638 0.2551 0.4954 0.4495 0.6516 0.7887 0.0939 0.0283 0.8358 0.4328 0.7623 0.0021 0.4454 0.7215 0.228
8 0.9453 0.9014 0.0306
```

Let's see what happens if we launch this piece of code again:

```
0.1344 0.8474 0.7638 0.2551 0.4954 0.4495 0.6516 0.7887 0.0939 0.0283 0.8358 0.4328 0.7623 0.0021 0.4454 0.7215 0.228
8 0.9453 0.9014 0.0306
```

The result is similar.

Random Number Generation Using Python

The random.uniform() function

- The *random.uniform*() function generates numbers within a defined numeric range.
- It can be used when requesting random numbers in well-defined intervals.
- Example:

```
: import random
  for i in range(20):
      print('%6.4f' % random.uniform(1, 100), end=' ')
```

```
3.5191 54.5998 93.9758 38.7392 22.4433 42.7895 3.8750 22.9475 44.3509 50.0854 24.0754 23.8558 22.6593 46.5007 29.6884
3.1275 83.9202 56.0890 64.5871 19.4047
```

We asked it to generate 20 random numbers in the range of $[1, 100)$.

Random Number Generation Using Python

The random.randint() function

- The *random.randint*() function generates random integers. The arguments for *randint*() are the values of the range, including the extremes.
- The numbers may be negative or positive, but the 1st value should be less than the second.
- Example:

```
import random
for i in range(20):
    print(random.randint(-100, 100), end=' ')
```

```
85 -25 -70 90 -15 84 82 28 8 29 71 -52 -23 -28 50 27 29 0 50 -92
```

We asked it to generate 20 random integer numbers from the range of $[-100, 100]$.

Random Number Generation Using Python

The random.choice() function

- The *random.choice*() function selects a random element from a sequence of an enumerated values.
- This function is suitable to use in extracting values from a predetermined list.
- Example:

```
import random
CitiesList = ['Rome','New York','Algiers','London','Berlin','Moskov',
'Los Angeles','Paris','Madrid','Tokio','Toronto']
for i in range(5):
    CitiesItem = random.choice(CitiesList)
    print ("Randomly selected item from Cities list is - ",CitiesItem)
```

```
Randomly selected item from Cities list is -  Tokio
Randomly selected item from Cities list is -  Tokio
Randomly selected item from Cities list is -  Los Angeles
Randomly selected item from Cities list is -  Toronto
Randomly selected item from Cities list is -  Algiers
```

We asked it to select 5 random elements from the sequence list CitiesList.
At each iteration of the cycle, a new element is extracted from the list containing the names of the cities.

Random Number Generation Using Python

The random.sample() function

- The *random.sample*() function generates samples without repeating the values and without changing the input sequence.
- This function is used for simulations that require random samples from a population of input values.
- Example:

```
import random
DataList = range(10,100,10)
print("Initial Data List = ",DataList)
DataSample = random.sample(DataList,k=5)
print("Sample Data List = ",DataSample)
```

```
Initial Data List =  range(10, 100, 10)
Sample Data List =  [30, 40, 10, 20, 50]
```

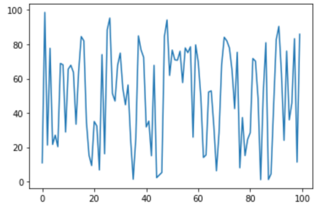
Only 5 elements of the initial list were selected, and this selection was completely random.
At each iteration of the cycle, a new elements is extracted from the initial DataList.

Exploring Probability Distributions With Python

Uniform Distribution

- This distribution generates random values uniformly distributed over the half-open interval $[a, b)$.
- Any value within the given interval is equally likely to be drawn by uniform distribution:

```python
import numpy as np
import matplotlib.pyplot as plt
#Initializing uniform distribution parameters
a=1
b=100
N=100
#Randomly generate 100 numbers between a and b
#using uniform distribution
X1=np.random.uniform(a,b,N)
#Ploting X1 using matplotlib.pyplot library
plt.plot(X1)
plt.show()
```

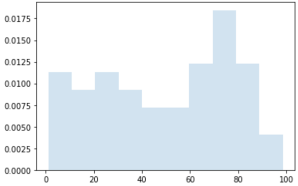


Exploring Probability Distributions With Python

Uniform Distribution (Continue)

- Lets now analyze how the generated values are distributed in the interval considered.
- We draw a graph of the probability density function:

```
plt.figure()
plt.hist(X1, density=True, histtype='stepfilled',
alpha=0.2)
plt.show()
```



- Here, we can see that the generated values are distributed almost evenly throughout the range.
- What happens if we increase the number of generated values?

Exploring Probability Distributions With Python

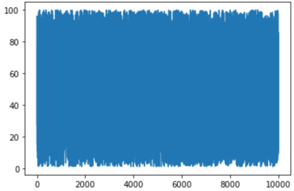
Uniform Distribution (Continue)

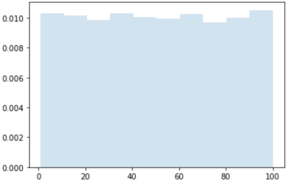
- Then, we repeat the commands to generate up to 10000 random numbers uniformly distributed between 1 and 100.

```
a=1
b=100
N=10000

X2=np.random.uniform(a,b,N)
plt.figure()
plt.plot(X2)
plt.show()
```

```
plt.figure()
plt.hist(X1, density=True, histtype='stepfilled',
alpha=0.2)
plt.show()
```





- We can see that this time, the distribution appears to be flatter

Exploring Probability Distributions With Python

Binomial Distribution

- The binomial distribution is the probability of obtaining x successes in t independent trials:

$$p(x) = \binom{t}{x} p^t (1-p)^{t-x}, 0 \leq x \leq t$$

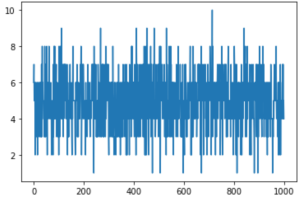
- **Example:** We throw a dice $t = 10$ times and we want to study the binomial variable $x =$ number of times a number $<= 3$ came out. The parameters of the problem are:
  - $t = 10$
  - $p = 3 * \frac{1}{6} = 0.5$
  - $q = 1 - p = 0.5$

Exploring Probability Distributions With Python

Binomial Distribution (Continue)

- We now evaluate the probability density function with Python code as follows:

```
import numpy as np
import matplotlib.pyplot as plt
#Initializing Binomial distribution parameters
N = 1000
t = 10
p = 0.5
#Generate the Binomial probability distribution
#Then Plot
P1 = np.random.binomial(t,p,N)
plt.plot(P1)
plt.show()
```

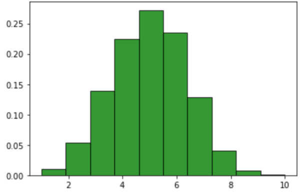


Exploring Probability Distributions With Python

Binomial Distribution (Continue)

- Now, let's see how these samples are distributed in the range considered as follows:

```
plt.figure()
plt.hist(P1, density=True, alpha=0.8, histtype='bar',
color = 'green', ec='black')
plt.show()
```



- All the areas of the binomial distributions, that is, the sum of the rectangles, being the sum of probability, are worth 1.

Exploring Probability Distributions With Python

Normal Distribution

- The most used continuous distribution in statistics.
- Recall that the probability density distribution of the normal distribution is given as follows:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

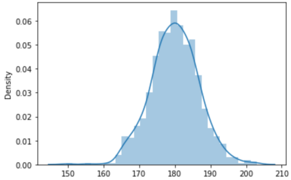
- **Example:** The average height of 18-year old boys is normally distributed with a mean of 180 cm and a standard deviation of 7 cm. The parameters of the problem are:
  - $\mu = 180$
  - $\sigma = 7$

Exploring Probability Distributions With Python

Normal Distribution (Continue)

- Now, let's generate a normal distribution and evaluate the probability density function with Python code as follows:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#Initializing Normal distribution parameters
mu = 180
sigma =7
N=1000
#Generate the Normal probability distribution
P1 = np.random.normal(mu, sigma, N)
#Then Plot
Plot = sns.distplot(P1)
plt.figure()
```



- We can see that the distribution of the 1000 samples populations follow a normal distribution as ploted by *sns.distplot*() function

End of Lecture 3:

Probabilities and Random Number Simulation