# Exercitiul 1

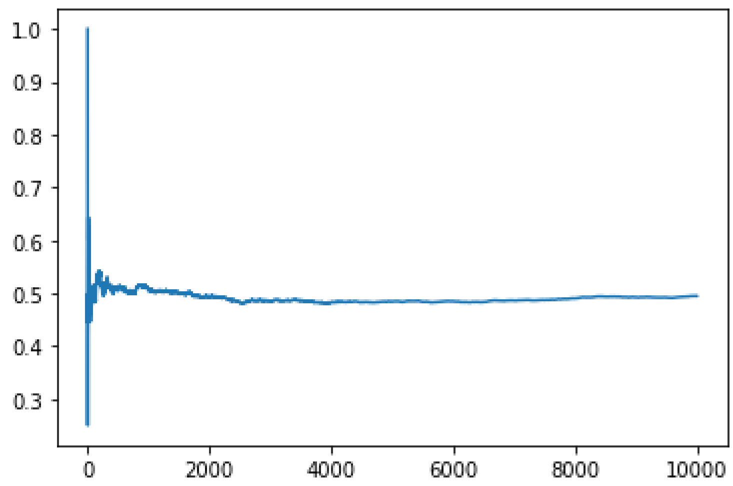
Dorim sa simulam aruncarea cu o moneda echilibrata. Consideram $H \equiv 1, T \equiv 0$.

1. Folosind functia numpy.random.randinit, generati un sir de $N = 10000$ de experimente;
2. Ilustrati faptul ca pe masura ce efectuam mai multe experimente,
   $P =$ nr de aparitii ale lui H/nr de simulari $\longrightarrow 1/2$.

***Solutie:***

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

def coins(N):
  a = np.random.randint(2, size = N)
  P_H = np.cumsum(a)
  P = np.divide(P_H, range(1, N+1))
  plt.plot(P)

def main():
  N = int(input("Enter a number: "))
  coins(N)

if __name__=="__main__":
  main()
```

Pentru $N = 10000$ aruncari, avem urmatorul output:

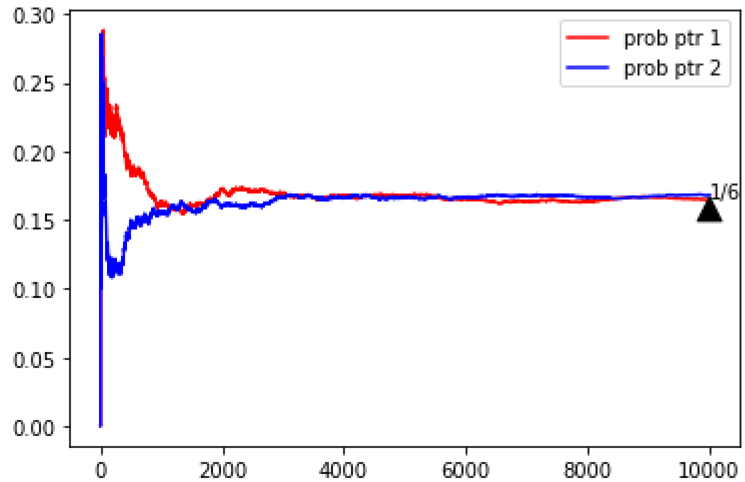


# Exercitiul 2

La fel ca la Exercitiul 4, pentru aruncarea cu zarul.
***Solutie:***
Vrem sa vedem ca pentru orice fata $i \in \{1, 2, 3, 4, 5, 6\}$, $\frac{\#\text{ nr de i din vector}}{\text{nr total de aruncari}} \sim \frac{1}{6} \sim 1.66$.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

def dice(N):
  a = np.random.randint(1, 7, size = N)

  P_1 = [list(a[0:i]).count(1) for i in range(1, len(a) + 1)]
  P_1 = np.divide(P_1, range(1, N + 1))
  P_2 = [list(a[0:i]).count(2) for i in range(1, len(a) + 1)]
  P_2 = np.divide(P_2, range(1, N + 1))

  fig, ax = plt.subplots()
  figure(figsize = (6, 4), dpi=80)
  ax.annotate('1/6',
           xy=(10000, 1/6),
           xytext=(10000, 1/6),
           arrowprops = dict(facecolor='black', shrink=0.05))
  ax.plot(range(N), P_1, color = 'red', label = 'prob ptr 1')
  ax.plot(range(N), P_2, color = 'blue', label = 'prob ptr 2')

  ax.legend()

  def main():
  N = int(input("Enter a number: "))
  dice(N)
  print(1/6)

if __name__=="__main__":
  main()
```

Graficul este pentru probabilitatile fetelor 1 si 2. Pentru restul se procedeaza identic.




# Exercitiul 3

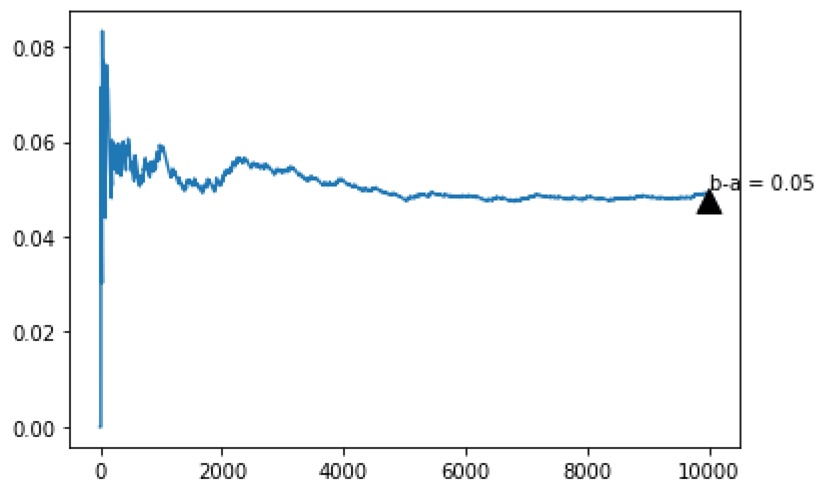
Un sir de numere $x = [x_1, x_2, ..., x_n], 0 \leq x_i \leq 1$ se numeste distribuit uniform pe $[0, 1]$ daca $\frac{\#x_i \in (a,b)}{n} \approx b - a, \forall (a, b) \subseteq [0, 1]$. Desigur, $b - a$ trebuie sa fie rezonabil in raport cu $n$ (numarul de sample-uri). Instructiunea random.uniform(0, 1, size=n) genereaza astfel de sir in Python. Pentru diverse valori ale lui $n$ si ale lui $b - a$, verificati ca sirul este intr-adevar uniform;

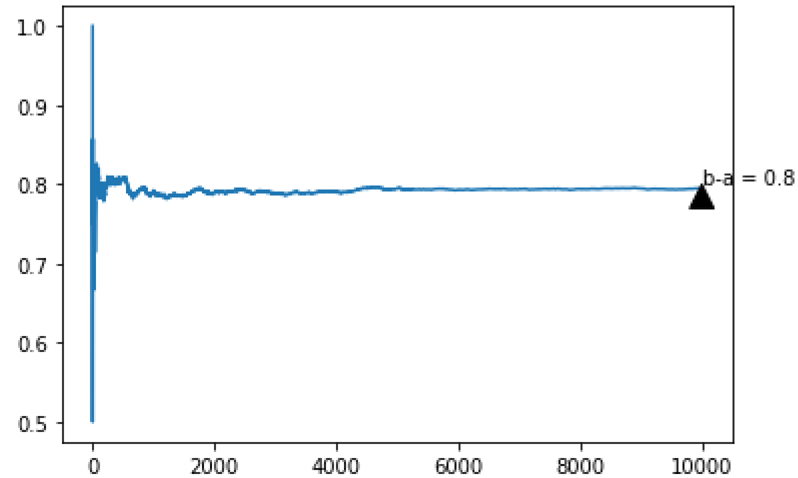
***Solutie:***

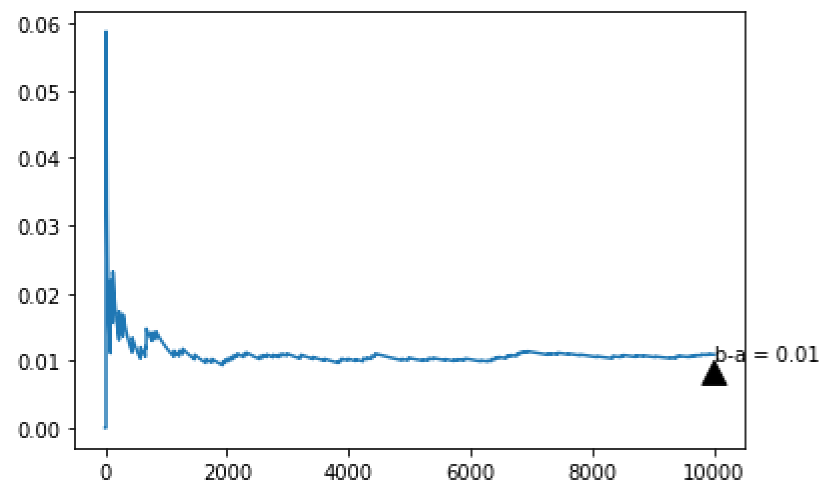
```
import numpy as np
import matplotlib.pyplot as plt

def count(a, b, v):
   ctr = 0
   for i in v:
      if i > a and i < b:
         ctr += 1
   return ctr

def coins(N, l, r):
    P = []
    a = np.random.uniform(0, 1, size = N)
    for i in range(1, N + 1):
      P.append(count(l, r, a[0:i])/i)
    fig, ax = plt.subplots()
    ax.annotate('b-a = ' + str(round(r-l, 2)),
            xy=(10000, r - l),
            xytext=(10000, r - l),
```

```
            arrowprops = dict(facecolor='black', shrink=0.05))
    ax.plot(range(N), P)

def main():
    N = 10000
    l = 0.1
    r = 0.15
    coins(N, l, r)
    l = 0.1
    r = 0.9
    coins(N, l, r)
    l = 0.5
    r = 0.51
    coins(N, l, r)
if __name__ == "__main__":
    main()
```

Outputul:






# Exercitiul 4

Am vazut ca random.uniform genereaza un numar aleator uniform in $[0,1]$. De exemplu daca generam sirul $x = \text{random.uniform}(0,1,N)$ si $N$ e mare, atunci $P(x \in (a,b)) = \frac{\#(a<x_i<b)}{N} \approx b-a$. Folositi random.uniform pentru a simula $N$ aruncari cu o moneda masluita. De exemplu, $P(H) = p = 0.7, P(T) = q = 0.3$, si atunci vom considera "moneda masluita" astfel: $H \equiv 1$ pica cand $x_N < 0.7$ si $T \equiv 0$ pica cand $x_N \geq 0.7$.

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

def coins(N, p):
    P = []
```

```
    P_B = []
    B = np.random.uniform(0, 1, size = N)
    for i in range(N):
        B[i] = 1 * (B[i] < p)
    P_B = np.cumsum(B)
    P = np.divide(P_B, range(1, N + 1))
    plt.plot(P)
    plt.show()

def main():
    coins(10000, 0.15)
if __name__ == "__main__":
    main()
```

Observam ca probabilitatea calculata frecventionist este aproximativ egala cu cea teoretica;

# Exercitiul 5

Un cod scris de echipa de informatica de la firma PS contine un bug in 5 din 100 de cazuri. Mihai are rolul de a verifica daca codul contine vreun bug. Performanta lui Mihai este urmatoarea:

- Din 100 de coduri cu bug, pe 95 le identifica corect;
- Din 100 de coduri fara bug, in 98 de cazuri identifica corect;

Mihai testeaza un cod nou si decide ca nu are bug. Care e probabilitatea sa greseasca?
2 solutii:

1. Teoretica
2. Practica:
    - De simulat evenimentul in care testam un cod si obtinem sau nu un bug; (simulam "moneda masluita" cu probabilitatea de reusita 5/100);
    - In functie de acesta, simulam evenimentul in care Mihai decide asupra codului; (la fel, pentru probabilitatea de reusita de 95/100);
    - Numaram in cate cazuri Mihai a prezis ca codul are un bug si cate din acestea erau prezise corect;

***Solutie:***

1. Notam cu $B$ evenimentul "codul are bug", $B^C$ este "codul nu are bug", $M_B$ este "Mihai spune ca are bug", $M_B^C$ este "Mihai spune ca nu are bug";
   Spatiul experimentelor este $\Omega = \mathbb{N}$ (pentru ca avem o multime de coduri, nu stim daca este finita si de aici presupunerea ca $\Omega = \mathbb{N}$ . Pe $\Omega$ definim $\sigma -$ algebra $\mathcal{F}$ generata de multimile $B, M_B$; Pe $\mathcal{F}$ avem o probabilitate $\mathbb{P}$ care indeplineste:

   - $\mathbb{P}(B) = 5/100$;
   - $\mathbb{P}(M_B|B) = 95/100$;
   - $\mathbb{P}(M_B^C|B^C) = 98/100$;

   Vrem sa determinam $\mathbb{P}(B|M_B^C)$;
   Stim din relatia lui Bayes ca $\mathbb{P}(B|M_B^C) = \mathbb{P}(M_B^C|B) \cdot \frac{\mathbb{P}(B)}{\mathbb{P}(M_B^C)}$.
   $\mathbb{P}(M_B^C) = \mathbb{P}(M_B^C|B) \cdot \mathbb{P}(B) + \mathbb{P}(M_B^C|B^C) \cdot \mathbb{P}(B^C) = (1 - \mathbb{P}(M_B|B)) \cdot \mathbb{P}(B) + \mathbb{P}(M_B^C|B^C) \cdot (1 - \mathbb{P}(B)) = \frac{5}{100} \cdot \frac{5}{100} + \frac{98}{100} \cdot \frac{95}{100} = \frac{9335}{10000}$.
   Deci, $\mathbb{P}(B|M_B^C) = \frac{5}{100} \cdot \frac{5}{100} \cdot \frac{10000}{9335} = \frac{25}{9335} \sim 0,0026$.

2. Aici consideram $\Omega = \{1, \ldots 100000\}$. Deci fiecare element din $\Omega$ este un cod care are sau nu un bug, probabilitatea de a avea unul fiind de 5/100; Practic, avem aproximativ 500 de elemente care reprezinta un cod cu bug; De asemenea, stim ca din cele care au bug, Mihai considera ca aproximativ 95/100 din ele au un bug si din cele care nu au, considera ca 98/100 nu au bug; Astfel:

   - Generam 100000 de valori 0 sau 1; probabilitatea sa avem 1 este 5/100;
   - Cand rezulta valoarea 1, vrem sa vedem daca Mihai considera daca aceasta este un bug sau nu; astfel, generam 0 sau 1, cu probabilitatea lui 1 fiind 95/100, bit care reprezinta clasificarea lui Mihai;
   - Cand intalnim 0, generam 0 sau 1, cu probabilitatea lui 0 fiind 98/100;
   - Ne intereseaza sa vedem $\frac{\text{\# perechi (1, 0)}}{\text{\#zerouri generate de "Mihai"}}$.
   - Cu cat $N$ este mai mare, cu atat acest raport trebuie sa se apropie de probabilitatea calculata la primul punct.

# Exercitiul 6

Aratati ca $P(A|B_1) = P(A|B_1, B_2) \cdot P(B_2|B_1) + P(A|B_1, B_2^C) \cdot P(B_2^C|B_1)$ ($\mathbb{P}(A|B_1, B_2)$ reprezinta $\mathbb{P}(A|(B_1 \cap B_2))$).

***Solutie:***
$\mathbb{P}(A|B_1, B_2) = \frac{\mathbb{P}(A \cap B_1 \cap B_2)}{\mathbb{P}(B_1 \cap B_2)} \Rightarrow \mathbb{P}(A|B_1, B_2) \cdot \mathbb{P}(B_2|B_1) = \frac{\mathbb{P}(A \cap B_1 \cap B_2)}{\mathbb{P}(B_1)}$;

$\mathbb{P}(A|B_1, B_2^C) = \frac{\mathbb{P}(A\cap B_1 \cap B_2^C)}{\mathbb{P}(B_1 \cap B_2^C)} \Rightarrow \mathbb{P}(A|B_1, B_2^C) \cdot \mathbb{P}(B_2^C, B_1) = \frac{\mathbb{P}(A\cap B_1 \cap B_2^C)}{\mathbb{P}(B_1)}$;
Deci, membrul drept este egal cu $\frac{\mathbb{P}(A\cap B_1 \cap B_2)+\mathbb{P}(A\cap B_1 \cap B_2^C)}{\mathbb{P}(B_1)} = \frac{\mathbb{P}(A\cap B_1)}{\mathbb{P}(B_1)} = \mathbb{P}(A|B_1)$.

# Exercitiul 7

Un program e format din 2 module independente.

1. Primul modul produce erori in 20% din rulari.
2. Cel de-al doilea modul produce erori in 40% din rulari.
3. Daca doar primul modul are eroare, atunci programul crapa in 50% din cazuri.
4. Daca doar al doilea modul are eroare, atunci programul crapa in 80% din cazuri.
5. Daca ambele module au eroare, atunci programul crapa in 90% din cazuri.

Daca programul a crapat, care e probabilitatea ca ambele module sa fi produs erori?

***Solutie:***
$E_1 =$ modulul 1 produce eroare, $E_2 =$ modulul 2 produce eroare, $C =$ programul crapa.
$\Omega = \mathbb{N}$ (pentru ca lucram pe "rulari", nu stim cate sunt si fiecarei rulari i se atribuie un numar natural), $\sigma -$ algebra pe care lucram este $\mathcal{F} = \sigma(E_1, E_2, C)$. Probabilitatea definita pe $\mathcal{F}$ are urmatoarele proprietati:

- $\mathbb{P}(E_1) = 2/10, \mathbb{P}(E_2) = 4/10$.
- $\mathbb{P}(C|E_1 \setminus E_2) = 5/10, \mathbb{P}(C|E_2 \setminus E_1) = 8/10, \mathbb{P}(C|E_1 \cap E_2) = 9/10$.
- Vrem sa calculam $\mathbb{P}(E_1 \cap E_2|C)$.

Observam ca $(E_1 \setminus E_2) \cup (E_2 \setminus E_1) \cup (E_1 \cap E_2) \cup \Omega \setminus (E_1 \cup E_2) = \Omega$ si aceste multimi sunt disjuncte 2 cate 2;
Stim ca $\mathbb{P}(A) = \mathbb{P}(A|A_1) \cdot \mathbb{P}(A_1) + \dots + \mathbb{P}(A|A_n) \cdot \mathbb{P}(A_n)$ unde $A_1, \dots A_n$ reprezinta o partitie disjuncta a lui $\Omega$. Deci, putem afla $\mathbb{P}(C) = \mathbb{P}(C|E_1 \setminus E_2) \cdot \mathbb{P}(E_1 \setminus E_2) +$
$+ \mathbb{P}(C|E_2 \setminus E_1) \cdot \mathbb{P}(E_2 \setminus E_1) + \mathbb{P}(C|E_1 \cap E_2) \cdot \mathbb{P}(E_1 \cap E_2) + \mathbb{P}(C|\Omega \setminus (E_1 \cup E_2)) \cdot \mathbb{P}(\Omega \setminus (E_1 \cup E_2))$.
Calculam:

- $\mathbb{P}(E_1 \setminus E_2) = \mathbb{P}(E_1 \cap E_2^C)$. Din faptul ca $E_1$ si $E_2$ sunt independente, rezulta ca $\mathbb{P}(E_1 \setminus E_2) = \mathbb{P}(E_1) \cdot \mathbb{P}(E_2^C) = \mathbb{P}(E_1) \cdot (1 - \mathbb{P}(E_2)) = 2/10 \cdot 6/10 = 12/100$.
- Analog, $\mathbb{P}(E_2 \setminus E_1) = \mathbb{P}(E_2) \cdot (1 - \mathbb{P}(E_1)) = 4/10 \cdot 8/10 = 32/100$.

- $\mathbb{P}(E_1 \cap E_2) = \mathbb{P}(E_1) \cdot \mathbb{P}(E_2) = 8/100$.
- Ultimul termen e 0 deoarece $\mathbb{P}(C|\Omega \setminus (E_1 \cup E_2)) = \frac{\mathbb{P}(C \cap E_1^C \cap E_2^C)}{\mathbb{P}(E_1^C \cap E_2^C)} = 0$ deoarece numaratorul e 0 (nu e posibil ca programul sa crape si in acelasi timp niciunul dintre modulele din care este format sa nu dea eroare).

Deci, $\mathbb{P}(C) = 5/10 \cdot 12/100 + 8/10 \cdot 32/100 + 8/100 \cdot 9/10 = 388/1000$.
Acum putem calcula $\mathbb{P}(E_1 \cap E_2|C)$:
$\mathbb{P}(E_1 \cap E_2|C) = \frac{\mathbb{P}(C|E_1 \cap E_2)}{\mathbb{P}(C)} \cdot \mathbb{P}(E_1 \cap E_2) = \frac{9/10}{388/1000} \cdot 8/100 = 72/388 \sim 0,1881$.

# Exercitiul 8

Fie $(\Omega, \mathcal{F})$ un spatiu de probabilitate, cu $\mathcal{F}$ o $\sigma -$ algebra. Aratati ca $X : \Omega \to \{0, 1\}$, $X \sim$ Bernoulli $\Longleftrightarrow \exists A \in \mathcal{F},$ asa incat $X = 1_A$, unde $1_A : \Omega \longrightarrow \{0,1\}, 1_A(\omega) = 1,$ pentru $\omega \in A, 1_A(\omega) = 0,$ pentru $\omega \notin A$.

***Solutie:***

- Ptr implicatia: "$\Rightarrow$" alegem $A = X^{-1}(\{1\}) = [X = 1]$; Atunci $X = 1_A$, evident si $A \in \mathcal{F}$ pentru ca $X$ este v.a. si $\{1\} \in \mathcal{B}(\mathbb{R})$.
- Pentru "$\Leftarrow$", demonstram ceva mai general, anume: $A \in \mathcal{F} \Leftrightarrow 1_A$ este v.a.
  Daca $A \in \mathcal{F}$, vrem sa demonstram ca pentru orice $a \in \mathbb{R}$ avem $1_A^{-1}((-\infty, a]) \in \mathcal{F}$.
  Observam ca $1_A^{-1}((-\infty, a])$ este:

  $$\begin{cases} \varnothing \text{ daca } a < 0 \\ A^C \text{ daca } 0 \le a < 1 \\ \Omega \text{ daca } a \ge 1 \end{cases}$$

  Toate aceste multimi fac parte din $\mathcal{F}$ deci $1_A$ este o v.a. si ptr ca ia doar valorile 0 si 1 inseamna ca este Bernoulli;
  Daca $1_A$ este o v.a., atunci $A = 1_A^{-1}(\{1\}) \in \mathbb{F}$ pentru ca $\{1\} \in \mathcal{B}(\mathbb{R})$.

# Exercitiul 9

Aratati ca orice variabila aleatoare discreta se poate scrie ca o combinatie liniara de variabile aleatoare Bernoulli.
*Reminder* O variabila aleatoare discreta este o v.a. $X : \Omega \longrightarrow \{x_1, x_2, \dots x_n \dots\} \subset \mathbb{R}$.
***Solutie:***
Pentru orice $x \in \Omega$, avem:

$$X(x) = \begin{cases} x_1 \text{ pentru } x \in X^{-1}(\{x_1\}) \\ x_2 \text{ pentru } x \in X^{-1}(\{x_2\}) \\ \dots \\ x_n \text{ pentru } x \in X^{-1}(\{x_n\}) \\ \dots \end{cases}$$

Deci, $X = x_1 \cdot 1_{X^{-1}(\{x_1\})} + x_2 \cdot 1_{X^{-1}(\{x_2\})} + \dots + x_n \cdot 1_{X^{-1}(\{x_n\})} + \dots$; Conform problemei 3, orice functie de tipul $1_{X^{-1}(\{x_n\})}$ este o v.a. Bernoulli deoarece $X^{-1}(\{x_n\}) \in \mathcal{F}$ ptr ca $X$ e o v.a si $\{x_n\} \in \mathcal{B}(\mathbb{R})$.
**Concluzia:** Ca sa simulam orice v.a discreta este nevoie sa simulam v.a Bernoulli.

**Cum se simuleaza o v.a. Bernoulli?**

### Variabile aleatoare uniforme

Fie $X : \Omega \longrightarrow [0, 1]$. Fie $A \subseteq [0, 1]$. Atunci, pentru $A \subset [0, 1]$, $A$ interval:

$$P(X^{-1}(A)) = P(X \in A) = \begin{cases} b - a, \text{ daca } 0 \le a \le b \le 1 \\ 0, \text{daca } b \le 0 \text{ sau } a \ge 1, \\ b, \text{daca } a \le 0 \le b \le 1, \\ 1 - a, \text{daca } 0 \le a \le 1 \le b \end{cases} \tag{1}$$

Se noteaza cu $X \sim \text{Unif}([0, 1])$

# Exercitiul 10

Fie $X \sim \text{Unif}([0, 1])$ si $p \in [0, 1]$. Aratati ca variabila aleatoare $Z = 1_{[0,p]}(X) = 1_{[0,p]} \circ X$ este Bernoulli($p$).

***Solutie:***
Pentru ca $1_{[0,p]}$ este masurabila si $X$ este o v.a, rezulta $Z$ este o v.a.
Pentru $x \in \Omega$:

$$Z(x) = \begin{cases} 1 \text{ pentru } \in X(x) \in [0,p] \\ 0 \text{ pentru } \in X(x) \in \mathbb{R} \setminus [0,p] \end{cases}$$

care este echivalent cu:

$$Z(x) = \begin{cases} 1 \text{ pentru } \in x \in X^{-1}([0,p]) \\ 0 \text{ pentru } \in x \in \Omega \setminus X^{-1}([0,p]) \end{cases}$$

Deci $Z = 1_{X^{-1}([0,p])}$. Pentru ca $X$ e v.a, avem ca $X^{-1}([0,p]) \in \mathcal{F}$, deci $Z$ este o v.a Bernoulli; $\mathbb{P}(Z = 1) = \mathbb{P}(X \in [0,p]) = p$ pentru ca $X \sim \text{Unif}([0,1])$. Deci, $Z \sim \text{Bernoulli}(p)$.

# Exercitiul 11

1. Simulati o v.a. Bernoulli;
2. Simulati o v.a. discreta;

***Solutie:***

1. Pentru a simula o v.a $X$ luam $\Omega = \{1, \dots N\}$ unde $N \in \mathbb{N}$ este suficient de mare; Pentru a calcula $\mathbb{P}(X = k)$, cu $k \in \{0,1\}$ calculam $\frac{\#\{i \in \Omega | X[i] = k\}}{N}$ si aceasta valoare trebuie sa fie aproximativ egala cu cea teoretica;

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

def Bernoulli(N, p):
    P = []
    Bernoulli = np.random.uniform(0, 1, size = N)
    for i in range(N):
        Bernoulli[i] = 1 * (Bernoulli[i] < p)
    P_B = np.cumsum(Bernoulli)
    P = np.divide(P_B, range(1, N+1))
    plt.plot(P)
    plt.show()

def main():
    Bernoulli(10000, 0.3)
if __name__ == "__main__":
    main()
```

2. Pentru a simula o v.a discreta $X$ folosim definitia cu v.a. Bernoulli gasita mai sus; am vazut ca o v.a Bernoulli $X = 1_A$ se simuleaza pentru $A = \mathcal{U}^{-1}([0, p])$,
unde $\mathcal{U} \sim \text{Unif}([0, 1])$ si $p = \mathbb{P}(X = 1)$;
Deci, pentru $X = x_1 \cdot 1_{X^{-1}(\{x_1\})} + x_2 \cdot 1_{X^{-1}(\{x_2\})} + \cdots + x_n \cdot 1_{X^{-1}(\{x_n\})}$, ca sa simulam $1_{X^{-1}(\{x_k\})}$, pentru ca $X^{-1}(\{x_k\}) = \mathcal{U}^{-1}([a_k, b_k])$ cu $b_k - a_k = p_k$ sunt disjuncte si reunite dau $\Omega$, trebuie ca intervalele $[a_k, b_k]$ trebuie sa pastreze aceasta proprietate; astfel, il simulam pe $X$ astfel:

$$X(x) = \begin{cases} x_1 \text{ pentru } x \in \mathcal{U}^{-1}([0, p_1]) \\ x_2 \text{ pentru } x \in \mathcal{U}^{-1}([p_1, p_1 + p_2]) \\ \dots \\ x_k \text{ pentru } x \in \mathcal{U}^{-1}([p_1 + \cdots + p_{k-1}, p_1 + \cdots + p_{k-1} + p_k]) \\ \dots \\ x_n \text{ pentru } x \in \mathcal{U}^{-1}([p_1 + \cdots + p_{n-1}, p_1 + \cdots + p_{n-1} + p_n]) \end{cases}$$

Astfel, pentru a simula o v.a. $X$ discreta care ia valorile $x_1, \ldots, X_n$ cu probabilitatile $p_1, \ldots p_n$ procedam astfel:

(a) Simulam $N$ valori din intervalul $[0, 1]$ (cu random.uniform).

(b) Iteram prin vectorul obtinut; fiecare valoare intalnita o incadram intr-unul dintre intervalele $[0, p_1], [p_1, p_1 + p_2], \ldots [p_1 + \cdots + p_{n-1}, 1]$. In functie de intervalul in care se gaseste, ii atribuim valoarea $x_i$ corespunzatoare (daca e in $[p_1 + \ldots p_{i-1}, \ldots, p_1 + \cdots + p_{i-1} + p_i]$);

(c) Am obtinut astfel un vector de $N$ valori care contine numai valorile $x_1, \ldots, x_n$. Vrem sa vedem ca acestea respecta distributia lui $X$, echivalent cu $x_1$ apare cu probabilitatea $p_1$, ..., $x_n$ apare cu probabilitatea $p_n$.

(d) Verificam acest lucru frecventionist; calculam $\frac{\# x_i}{N}$ si trebuie sa fie apropiat de $p_i$.