

LABORATOR #5

EX#1 Se aruncă două zaruri corecte cu fețele 1,2,3,4,5,6. Considerăm următoarele evenimente:

- (i) A = primul este 1;
- (ii) B = al doilea este 6;
- (iii) C = suma celor două este 7.

Creați un fișier în Python[®] prin care să se genereze N aruncări cu cele două zaruri și să se verifice numeric (frecvenționist) dacă:

- (a) A este independent de C ;
- (b) B este independent de C ;
- (c) A , B și C sunt independente.

EX#2 Se aruncă două zaruri corecte cu fețele 1,2,3,4,5,6. Considerăm următoarele evenimente:

- (i) A = primul este mai mic sau egal cu 2;
- (ii) B = suma celor două este 7;
- (iii) C = al doilea este par.

Creați un fișier în Python[®] prin care să se genereze N aruncări cu cele două zaruri și să se verifice numeric (frecvenționist) dacă:

- (a) A este independent de C ;
- (b) A este independent de B ;
- (c) B este independent de C ;
- (d) A , B și C sunt independente.

EX#3 Fie X un număr generat aleator uniform în intervalul $[-1, 1]$.

- (a) Fie Y un număr generat aleator uniform în intervalul $[-1, 1]$;
- (b) Fie $Y = -X$.

Creați un fișier în Python[®] prin care să se verifice numeric (frecvenționist) dacă X și Y sunt independente pentru fiecare dintre cazurile (a), respectiv (b).

EX#4 Propuneți 3 evenimente realiste A , B și C . Creați un fișier în Python[®] prin care să se verifice numeric (frecvenționist) dacă:

- (a) A este independent de B ; A este independent de C ;
- (b) B este independent de C ;
- (c) A , B și C sunt independente.

EX#5 Considerăm următorul joc pentru o singură persoană. Un jucător aruncă o monedă corectă. Dacă pică H , primește 1 leu, dacă pică T , pierde 1 leu. La începutul jocului, jucătorul are suma de $m > 0$ lei. Jocul se termină atunci când jucătorul nu mai are bani sau atinge o sumă dorită de M lei, $M > m$. Creați un fișier în Python® prin care:

- (a) să se simuleze jocul;
- (b) să se estimeze numeric (frecvenționist) probabilitatea ca jucătorul să atingă suma de $M > 0$ lei, știind că la începutul jocului a avut m lei, $0 < m < M$;
- (c) să se afișeze histograma corespunzătoare lungimii (duratei) jocurilor simulate.

EX#6 Un jucător are m lei, $m > 0$, pentru un joc. Propuneți un joc pentru un singur jucător astfel încât, la fiecare pas al jocului, jucătorul poate să câștige o sumă de bani sau pierde (eventual altă) sumă de bani. Jocul se termină atunci când jucătorul nu mai are bani sau atinge suma dorită de M lei, $M > m$. Creați un fișier în Python® prin care:

- (a) să se simuleze jocul propus;
- (b) să se estimeze numeric (frecvenționist) probabilitatea ca jucătorul să atingă suma de $M > 0$ lei, știind că la începutul jocului a avut m lei, $0 < m < M$;
- (c) să se afișeze histograma corespunzătoare lungimii (duratei) jocurilor simulate.

Indicații Python®: `numpy`, `numpy.random`, `matplotlib.pyplot`, `matplotlib.pyplot.hist`