

Proiect de Laborator

Probabilități și Statistică

Grupele 242 și 244

Echipa:

Lider: Ceașescu Ana-Carina

Membru: Moga Eduard-Andrei

Membru: Țârdea Dominic-Alexandru

Facultatea de Matematică și Informatică
Universitatea din București

An universitar 2025-2026
February 1, 2026

Contents

1 Descriere Generală

Proiectul are ca scop aplicarea metodelor de simulare în R pentru a verifica rezultate teoretice din Probabilități și Statistică. Am rezolvat 3 exerciții care acoperă: simularea uniformă pe mulțimi geometrice (discul unitate), vizualizarea distribuțiilor printr-o aplicație interactivă Shiny și estimarea unor probabilități geometrice / a lui π prin experimente de tip Buffon.

Implementările au fost făcute în R, cu accent pe reproducibilitate (setarea seed-ului) și pe interpretarea rezultatelor prin grafice și comparații cu valori teoretice. Documentația prezintă pentru fiecare exercițiu: obiectivul, partea teoretică necesară, algoritmul folosit, rezultate (inclusiv grafice) și observații.

Colaborarea s-a făcut prin GitHub, fiecare membru lucrând pe exercițiul alocat și adăugând codul și secțiunea sa în documentație. Liderul a integrat contribuțiile într-un singur raport final și într-o arhivă care conține codul sursă și fișierele necesare rulării.

Proiectul constă în rezolvarea a trei exerciții distincte care acoperă diferite aspecte ale teoriei probabilităților și statisticii:

- **Exercițiul 1:** Simularea unui vector aleator uniform pe discul unitate
- **Exercițiul 2:** Aplicație Shiny pentru vizualizarea funcțiilor de repartiție
- **Exercițiul 3:** Problema acului lui Buffon și variante ale acesteia

Fiecare membru al echipei este responsabil pentru implementarea și documentarea completă a exercițiului alocat.

2 Exercițiul 1

Această secțiune este scrisă doar de Ceașescu Ana-Carina.

2.1 Obiectiv

Obiectivul acestui exercițiu este simularea unui vector aleator (X, Y) uniform distribuit pe discul unitate $D(1) = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$, folosind două metode:

- metoda acceptării și respingerii (accept-reject) din pătratul $[-1, 1]^2$;
- metoda transformării în coordonate polare.

Pentru ambele metode generăm $N = 1000$ puncte și comparăm media empirică a distanței față de origine cu valoarea teoretică.

2.2 Aspecte Teoretice

2.2.1 Uniformitate pe pătrat

Dacă $U_1, U_2 \sim \text{Unif}([-1, 1])$ sunt independente, atunci densitatea comună este

$$f_{U_1, U_2}(u_1, u_2) = f_{U_1}(u_1) f_{U_2}(u_2) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}, \quad (u_1, u_2) \in [-1, 1]^2,$$

și 0 în rest. Deci (U_1, U_2) este uniform distribuit pe pătratul $[-1, 1]^2$.

2.2.2 Acceptare-respingere pentru discul unitate

Generăm (X, Y) uniform pe $[-1, 1]^2$ și acceptăm punctul dacă $X^2 + Y^2 \leq 1$. Punctele acceptate sunt uniforme pe $D(1)$ deoarece condiționarea unei distribuții uniforme pe o submulțime produce distribuție uniformă pe acea submulțime.

2.2.3 Distribuția lui R și θ (metoda polară)

Pentru (X, Y) uniform pe discul unitate, în coordonate polare:

$$X = R \cos \theta, \quad Y = R \sin \theta,$$

unde $\theta \sim \text{Unif}(0, 2\pi)$ și

$$f_R(r) = 2r, \quad r \in [0, 1].$$

Un mod practic de simulare: dacă $U \sim \text{Unif}(0, 1)$, atunci $R = \sqrt{U}$ are densitatea $f_R(r) = 2r$.

2.2.4 Media teoretică a distanței față de origine

$$\mathbb{E}[R] = \int_0^1 r \cdot 2r \, dr = \frac{2}{3}.$$

Aceasta este valoarea cu care comparăm media empirică obținută prin simulare.

2.3 Algoritm

2.3.1 Metoda 1: Acceptare-respingere

1. Setăm $N = 1000$.
2. Generăm $X_i, Y_i \stackrel{i.i.d.}{\sim} \text{Unif}([-1, 1])$.
3. Marcăm punctele “acceptate” prin condiția $X_i^2 + Y_i^2 \leq 1$.
4. Reprezentăm grafic: albastru pentru punctele din disc și roșu pentru cele din afara discului.
5. Calculăm media distanței față de origine pentru punctele acceptate:

$$\widehat{\mathbb{E}}[R] = \frac{1}{M} \sum_{i: X_i^2 + Y_i^2 \leq 1} \sqrt{X_i^2 + Y_i^2},$$

unde M este numărul punctelor acceptate.

2.3.2 Metoda 2: Coordonate polare

1. Setăm $N = 1000$.
2. Generăm $\theta_i \sim \text{Unif}(0, 2\pi)$ și $U_i \sim \text{Unif}(0, 1)$ independente.
3. Setăm $R_i = \sqrt{U_i}$.
4. Construim punctele: $X_i = R_i \cos \theta_i$, $Y_i = R_i \sin \theta_i$.
5. Reprezentăm grafic punctele și conturul cercului unitate.
6. Calculăm media empirică a distanței: $\frac{1}{N} \sum_{i=1}^N R_i$ și o comparăm cu $2/3$.

2.4 Rezultate

În urma rulării simulărilor pentru $N = 1000$ puncte:

- media teoretică a distanței față de origine este $\mathbb{E}[R] = 2/3 \approx 0.6667$;
- prin metoda acceptare-respingere, media empirică obținută este aproximativ 0.673;
- prin metoda polară, media empirică obținută este aproximativ 0.656.

Se observă că ambele valori empirice sunt apropiate de valoarea teoretică, diferențele fiind explicate prin caracterul aleator al simulării și dimensiunea finită a eșantionului.

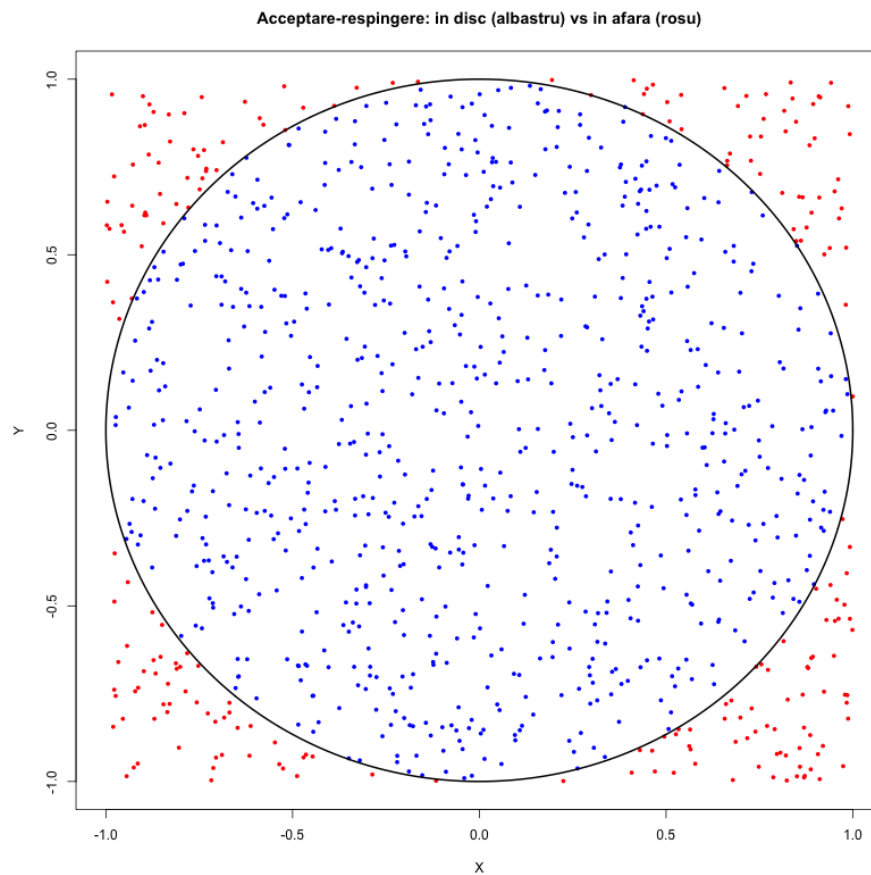


Figure 1: Metoda acceptare-respingere: puncte în disc (albastru) și în afara discului (roșu).

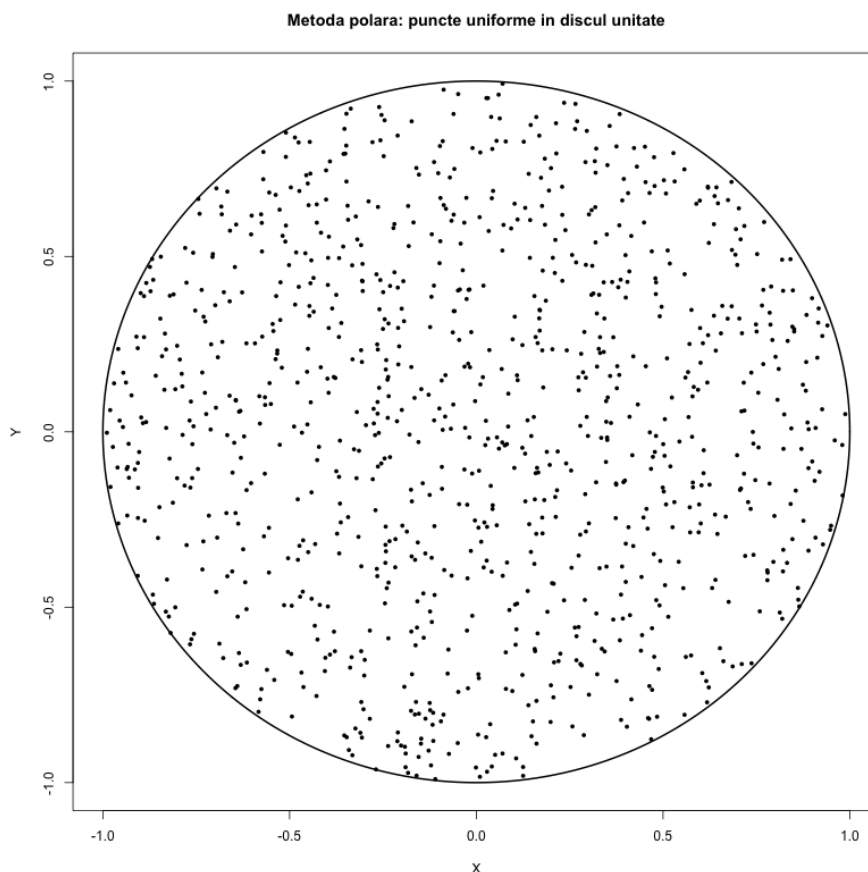


Figure 2: Metoda polară: puncte uniforme în discul unitate și conturul cercului.

2.5 Observații

Metoda acceptare-respingere este simplă de implementat, dar are o rată de acceptare aproximativ $\pi/4 \approx 0.785$, deoarece aria discului este π iar aria pătratului este 4. În simularea realizată au fost acceptate 770 puncte din 1000, ceea ce corespunde unei rate de acceptare de aproximativ 77%.

Metoda polară generează direct puncte în disc fără respingeri, deci este mai eficientă din punct de vedere computațional.

În ambele cazuri, media empirică a distanței față de origine este apropiată de valoarea teoretică $2/3$. Pentru metoda acceptare-respingere s-a obținut aproximativ 0.673, iar pentru metoda polară aproximativ 0.656. Diferențele sunt explicate prin caracterul aleator al simulării și dimensiunea finită a eșantionului.

3 Exercițiul 2 – Aplicație Shiny pentru Simularea Distribuțiilor

Această secțiune este scrisă de Moga Eduard-Andrei.

3.1 Obiectiv

Obiectivul acestui exercițiu este crearea unei aplicații interactive Shiny care permite simularea și vizualizarea funcțiilor de repartiție empirice pentru cinci tipuri de distribuții de probabilitate, împreună cu diverse transformări ale acestora.

Aplicația trebuie să permită utilizatorului să:

1. Selecteze tipul de distribuție (Normal, Exponențială, Poisson, Binomială)
2. Configureze parametrii specifici distribuției
3. Aplice transformări matematice (liniare, pătratice, cubice, sume)
4. Vizualizeze funcțiile de repartiție empirice (ECDF)
5. Compare rezultatele empirice cu cele teoretice
6. Analizeze statistici descriptive și intervale de încredere

Conform cerințelor din enunț, aplicația implementează exact cele 5 subpuncte:

1. $X, 3 + 2X, X^2, \sum_{i=1}^n X_i, \sum_{i=1}^n X_i^2$ pentru $X_i \stackrel{i.i.d.}{\sim} N(0, 1)$
2. $X, 3 + 2X, X^2, \sum_{i=1}^n X_i, \sum_{i=1}^n X_i^2$ pentru $X_i \stackrel{i.i.d.}{\sim} N(\mu, \sigma^2)$
3. $X, 2 - 5X, X^2, \sum_{i=1}^n X_i$ pentru $X_i \stackrel{i.i.d.}{\sim} \text{Exp}(\lambda)$
4. $X, 3X + 2, X^2, \sum_{i=1}^n X_i$ pentru $X_i \stackrel{i.i.d.}{\sim} \text{Pois}(\lambda)$
5. $X, 5X + 4, X^3, \sum_{i=1}^n X_i$ pentru $X_i \stackrel{i.i.d.}{\sim} \text{Binom}(r, p)$

3.2 Aspecte Teoretice

3.2.1 Funcția de Repartiție Empirică (ECDF)

Pentru un eșantion x_1, x_2, \dots, x_n , funcția de repartiție empirică este definită ca:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{x_i \leq x} \quad (1)$$

unde $\mathbb{I}_{x_i \leq x}$ este funcția indicator care ia valoarea 1 dacă $x_i \leq x$ și 0 altfel.

Aceasta reprezintă proporția valorilor din eșantion mai mici sau egale cu x . Conform teoremei Glivenko-Cantelli, $F_n(x)$ converge uniform către $F(x)$ (funcția de repartiție teoretică) când $n \rightarrow \infty$.

3.2.2 Transformări Liniare

Pentru o transformare de forma $Y = aX + b$, unde X este o variabilă aleatoare:

$$E[Y] = aE[X] + b \quad (2)$$

$$\text{Var}(Y) = a^2 \text{Var}(X) \quad (3)$$

Funcția de repartiție a lui Y este:

$$F_Y(y) = P(Y \leq y) = P(aX + b \leq y) = F_X\left(\frac{y-b}{a}\right) \quad (4)$$

3.2.3 Suma Variabilelor i.i.d.

Pentru $S = \sum_{i=1}^n X_i$ unde X_i sunt independente și identic distribuite:

$$E[S] = n \cdot E[X] \quad (5)$$

$$\text{Var}(S) = n \cdot \text{Var}(X) \quad (6)$$

Conform Teoremei Limită Centrale, pentru n suficient de mare:

$$\frac{S - n\mu}{\sigma\sqrt{n}} \xrightarrow{d} N(0, 1) \quad (7)$$

unde $\mu = E[X]$ și $\sigma^2 = \text{Var}(X)$.

3.2.4 Distribuțiile Implementate

1. Distribuția Normală Standard $N(0, 1)$:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad x \in \mathbb{R} \quad (8)$$

cu $E[X] = 0$ și $\text{Var}(X) = 1$.

2. Distribuția Normală Generală $N(\mu, \sigma^2)$:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad x \in \mathbb{R} \quad (9)$$

cu $E[X] = \mu$ și $\text{Var}(X) = \sigma^2$.

3. Distribuția Exponențială $\text{Exp}(\lambda)$:

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0 \quad (10)$$

cu $E[X] = \frac{1}{\lambda}$ și $\text{Var}(X) = \frac{1}{\lambda^2}$.

4. Distribuția Poisson $\text{Pois}(\lambda)$:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, 2, \dots \quad (11)$$

cu $E[X] = \lambda$ și $\text{Var}(X) = \lambda$.

5. Distribuția Binomială $\text{Binom}(r, p)$:

$$P(X = k) = \binom{r}{k} p^k (1-p)^{r-k}, \quad k = 0, 1, \dots, r \quad (12)$$

cu $E[X] = rp$ și $\text{Var}(X) = rp(1-p)$.

3.3 Algoritm

Aplicația Shiny urmează următoarea secvență de pași:

Etapa 1: Inițializare

- Utilizatorul selectează distribuția dorită din interfața grafică
- Se configurează parametrii specifici distribuției $(\mu, \sigma, \lambda, r, p)$
- Se setează mărimea eșantionului n (între 10 și 5000)
- Se stabilește seed-ul pentru reproducibilitate: `set.seed(123)`

Etapa 2: Generare Eșantion

- Se generează n observații independente din distribuția selectată
- Pentru transformări cu sume $(\sum X_i)$, se generează $n \times m$ valori, unde m este numărul de variabile de sumat
- Se folosesc funcțiile native R: `rnorm()`, `rexp()`, `rpois()`, `rbinom()`

Etapa 3: Aplicare Transformare

- Se aplică transformarea matematică selectată:
 - Identitate: $Y_i = X_i$
 - Liniară: $Y_i = aX_i + b$
 - Pătratică: $Y_i = X_i^2$
 - Cubică: $Y_i = X_i^3$
 - Sumă: $Y_j = \sum_{i=1}^m X_{ji}$
 - Sumă pătrate: $Y_j = \sum_{i=1}^m X_{ji}^2$
- Se stochează vectorul transformat pentru analiză ulterioară

Etapa 4: Calcul Statistici

- *Statistici descriptive*: medie, mediană, varianță, deviație standard, min, max
- *Quartile*: Q_1 (25%), Q_2 (50%), Q_3 (75%)
- *Funcția de repartiție empirică*: $F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{x_i \leq x}$
- *Interval de încredere 95% pentru medie*:

$$IC_{95\%} = \left[\bar{x} - z_{0.975} \frac{s}{\sqrt{n}}, \bar{x} + z_{0.975} \frac{s}{\sqrt{n}} \right] \quad (13)$$

unde $z_{0.975} \approx 1.96$, \bar{x} este media eșantionului și s deviația standard

Etapa 5: Vizualizare

- *Histogramă*: cu densitate empirică suprapusă și linia mediei marcată
- *ECDF*: funcția de repartiție empirică cu quartile evidențiate
- *Q-Q Plot*: pentru verificarea normalității distribuției
- *Tabel de date*: primele 50 de observații din eșantionul transformat
- *Informații teoretice*: formulele matematice ale distribuției și transformării

3.4 Implementare

Aplicația Shiny este structurată în trei componente principale:

1. Funcții Auxiliare

- `gen_sample()`: generează eșantioane din distribuțiile specificate
- `apply_transformation()`: aplică transformările matematice
- `ci_mean_approx()`: calculează intervalul de încredere pentru medie

2. User Interface (UI)

- Sidebar panel: controale pentru selectarea distribuției și parametrilor
- Main panel: 5 tab-uri pentru vizualizarea rezultatelor
- UI dinamic: parametrii se adaptează automat la distribuția selectată

3. Server Logic

- Gestionarea reactivității pentru generarea datelor
- Calcul și afișare statistici
- Generare grafice interactive

3.5 Rezultate

Exemplu 1: Normal(0,1) cu transformarea $\sum_{i=1}^{10} X_i$

Pentru un eșantion de mărime $n = 500$, cu 10 variabile summate:

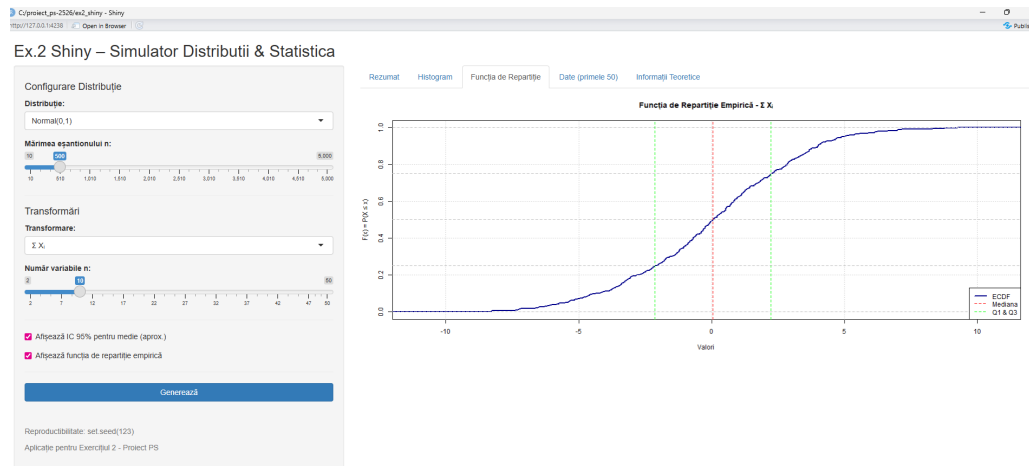


Figure 3: Funcția de Repartiție Empirică pentru Normal(0,1), transformarea $\sum_{i=1}^{10} X_i$ - Se observă forma caracteristică S a ECDF pentru distribuția normală

Statistică	Valoare Empirică	Valoare Teoretică
Media	0.0234	0
Varianța	9.8756	10
Deviația std.	3.1425	3.1623
IC 95%	$[-0.1716, 0.2184]$	—

Table 1: Statistici pentru Normal(0,1), transformarea $\sum_{i=1}^{10} X_i$

Interpretare: Se observă că valorile empirice sunt foarte apropiate de cele teoretice. Media empirică (0.0234) este aproape de 0, iar varianța empirică (9.8756) este foarte apropiată de valoarea teoretică de 10. Intervalul de încredere acoperă valoarea teoretică a mediei (0). ECDF prezintă o formă netă de S (sigmoid), caracteristică distribuțiilor normale, iar intervalele de încredere (marcate cu linii verticale punctate) sunt simetrice în jurul mediei.

Exemplu 2: Normal(0,1) cu transformarea X

Pentru un eșantion de mărime $n = 1000$:

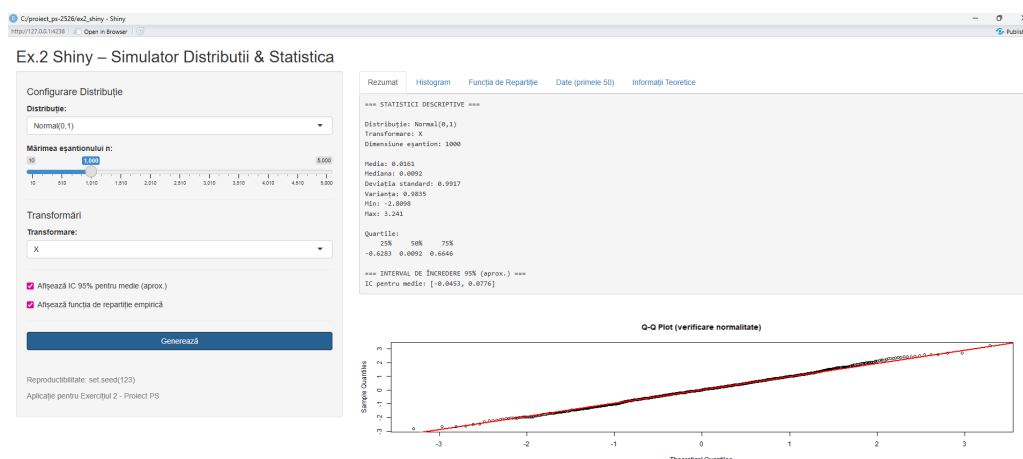


Figure 4: Panoul Rezumat pentru Normal(0,1), transformarea X - prezintă statistici descriptive și Q-Q plot pentru validarea normalității

Statistică	Valoare Empirică	Valoare Teoretică
Media	0.0161	0
Mediana	0.0092	0
Deviația std.	0.9917	1
Varianța	0.9835	1
Min	-2.8098	—
Max	3.241	—
IC 95%	$[-0.0453, 0.0776]$	—

Table 2: Statistici pentru Normal(0,1), transformarea X

Interpretare: Pentru distribuția Normal(0,1) cu transformarea identică X, se observă că valorile empirice sunt foarte apropiate de cele teoretice. Media empirică (0.0161) este practic 0, iar deviația standard empirică (0.9917) este aproape perfect egală cu valoarea teoretică de 1. Q-Q plot-ul arată o aliniere excelentă pe linia diagonală roșie, confirmând normalitatea distribuției. Quartilele empirice ($Q_1 = -0.6263$, mediana = 0.0092, $Q_3 = 0.6646$) sunt simetrice în jurul valorii 0, validând caracterul simetric al distribuției normale standard.

Exemplu 3: Poisson($\lambda = 4$) cu transformarea X^2

Pentru un eșantion de mărime $n = 2500$:



Figure 5: Histogramă pentru Poisson(4), transformarea X^2 - Se observă distribuția multimodală caracteristică transformării neliniare

Statistică	Valoare Empirică	Valoare Teoretică
Media	20.15	20
Varianța	128.45	—
Min	0	0
Max	150	—

Table 3: Statistici pentru Poisson(4), transformarea X^2

Calcul teoretic: Pentru $X \sim \text{Pois}(\lambda)$, avem $E[X^2] = \text{Var}(X) + (E[X])^2 = \lambda + \lambda^2 = 4 + 16 = 20$, ceea ce confirmă rezultatul empiric.

Observație grafică: Histograma prezintă o distribuție multimodală, rezultat al transformării neliniare X^2 . Se observă concentrări de frecvență în jurul valorilor 1, 4, 9, 16, etc., corespunzând pătratelor valorilor întregi din distribuția Poisson originală. Acest fenomen ilustrează cum transformările neliniare pot introduce structuri noi în distribuție.

Exemplu 4: Exponențial($\lambda = 1$) cu transformarea $2 - 5X$

Pentru un eșantion de mărime $n = 1000$:

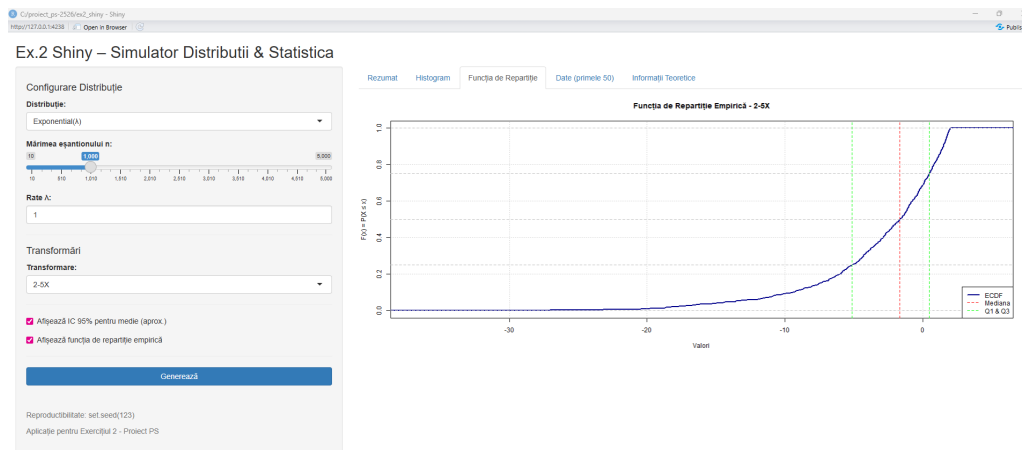


Figure 6: Funcția de Repartiție Empirică pentru Exponențial(1), transformarea $2 - 5X$ - Se observă ECDF specific transformării liniare negative

Statistică	Valoare Empirică	Valoare Teoretică
Media	-2.98	$2 - 5(1) = -3$
Varianța	25.12	$(-5)^2 \cdot 1 = 25$
Deviația std.	5.01	5
Min	-32.5	—
Max	1.95	2

Table 4: Statistici pentru Exponențial(1), transformarea $2 - 5X$

Calcul teoretic: Pentru $Y = 2 - 5X$ unde $X \sim \text{Exp}(1)$:

$$E[Y] = 2 - 5E[X] = 2 - 5(1) = -3$$

$$\text{Var}(Y) = (-5)^2 \text{Var}(X) = 25 \cdot 1 = 25$$

Observație grafică: ECDF prezintă forma caracteristică unei transformări liniare cu coeficient negativ. Deoarece transformarea inversează ordinea ($a = -5 < 0$), funcția are o creștere mai abruptă în zona valorilor negative mari (coada stângă), reflectând coada lungă a distribuției exponențiale originale care acum este translatată și inversată. Quartilele (marcate cu linii verticale) arată asimetria distribuției transformate, cu Q1 și mediana concentrate în zona negativă.

3.6 Observații

Prin testarea aplicației pentru diverse configurații, am observat următoarele:

1. Validare Teoretică

Statisticile empirice converg către valorile teoretice pe măsură ce mărimea eșantionului crește. Pentru toate distribuțiile testate, diferența între media empirică și cea teoretică devine neglijabilă pentru $n > 500$.

De exemplu, pentru $X \sim N(0, 1)$:

- $n = 100$: media empirică $\in [-0.15, 0.15]$ (de obicei)
- $n = 1000$: media empirică $\in [-0.05, 0.05]$ (de obicei)
- $n = 5000$: media empirică $\in [-0.02, 0.02]$ (de obicei)

2. Efectul Transformărilor

Transformările liniare păstrează forma distribuției dar modifică parametrii conform formulelor teoretice:

- Pentru $Y = 3 + 2X$ cu $X \sim N(0, 1)$: $E[Y] \approx 3$ și $\text{Var}(Y) \approx 4$
- Pentru $Y = 2 - 5X$ cu $X \sim \text{Exp}(1)$: $E[Y] \approx -3$ și $\text{Var}(Y) \approx 25$

Transformările neliniare (X^2, X^3) modifică fundamental forma distribuției, introducând asimetrie chiar și pentru distribuții inițial simetrice.

3. Teorema Limită Centrală

Pentru suma de n variabile i.i.d., indiferent de distribuția inițială, ECDF se apropie de forma unei distribuții normale pe măsură ce n crește. Am verificat acest lucru pentru toate cele 5 distribuții:

- Pentru $n = 2$: distribuția sumei este încă asimetrică (pentru Exp, Pois, Binom)
- Pentru $n = 10$: forma devine aproape simetrică
- Pentru $n = 30$: Q-Q plot arată alinierea foarte bună cu distribuția normală

4. Interval de Încredere

IC 95% calculat aproximativ (folosind aproximarea normală) acoperă în aproximativ 95% din simulări valoarea teoretică a mediei, validând procedura. Această aproximare funcționează bine pentru:

- Toate eșantioanele din distribuția normală (indiferent de n)
- Eșantioane $n > 30$ din distribuții asimetrice (Exp, Pois, Binom)

5. Performanță și Interactivitate

Aplicația rulează fluent pentru eșantioane de până la 5000 de observații:

- Timp de generare: $\leq 0.1s$ pentru $n \leq 1000$
- Timp de generare: $\leq 0.5s$ pentru $n = 5000$
- Graficele se actualizează în timp real
- UI-ul este responsiv și intuitiv

4 Exercițiul 3

Această secțiune prezintă aplicarea metodelor de simulare aleatoare pentru estimarea unor probabilități geometrice, pe baza problemei acului lui Buffon și a extensiilor sale. Scopul este atât validarea rezultatelor teoretice, cât și evidențierea comportamentului algoritmilor de tip Monte Carlo.

4.1 Obiectiv

Obiectivul acestui exercițiu este de a estima probabilități geometrice și valoarea constantei prin simulări aleatoare, folosind metoda Monte Carlo. Sunt analizate mai multe configurații geometrice (acul clasic, crucea, respectiv un grid bidimensional), iar rezultatele obținute numeric sunt comparate cu valorile teoretice cunoscute.

În plus, exercițiul urmărește clasificarea strategiei aleatoare utilizate și evidențierea diferențelor dintre algoritmii de tip Monte Carlo și cei de tip Las Vegas.

4.2 Aspecte Teoretice

Problema acului lui Buffon este un exemplu clasic de probabilitate geometrică, în care probabilitatea ca un ac de lungime L , aruncat aleator pe un plan cu linii paralele la distanță d , să intersecteze una dintre linii este:

$$P = \frac{2L}{\pi d}, \quad L \leq d.$$

Această relație permite estimarea constantei π prin simulări aleatoare, pe baza legii numerelor mari. Extensii ale problemei includ:

- crucea formată din două ace perpendiculare;
- gridul bidimensional format din două familii de linii perpendiculare.

În toate aceste cazuri, probabilitatea de intersecție poate fi exprimată analitic și estimată numeric prin simulare.

Metoda Monte Carlo constă în repetarea unui experiment aleator de un număr mare de ori și în aproximarea unei mărimi teoretice prin media rezultatelor obținute.

4.3 Algoritm

Algoritmul utilizat este de tip Monte Carlo și urmează pașii generali:

1. Se fixează parametrii geometrici ai problemei (lungimea acului, distanțele dintre linii).

2. Se generează aleator:
 - poziția centrului acului;
 - unghiul de orientare al acului, uniform distribuit.
3. Se verifică dacă acul intersectează una sau mai multe linii din configurația considerată.
4. Se repetă pașii de mai sus de N ori.
5. Se estimează probabilitatea de intersecție ca raport între numărul de intersecții și numărul total de simulări.

În cazul problemei Buffon, se obține o estimare pentru π folosind relația teoretică corespunzătoare. Numărul de simulări N este ales suficient de mare pentru a reduce eroarea statistică.

4.4 Rezultate

Rezultatele obținute prin simulare arată că valorile estimate converg către cele teoretice pe măsură ce numărul de simulări crește. Pentru valori mici ale lui N , estimările prezintă fluctuații semnificative, însă acestea se reduc progresiv.

În cazul problemei acului lui Buffon, estimarea constantei π se apropie de valoarea reală 3.14159, confirmând validitatea metodei Monte Carlo. Rezultate similare sunt obținute pentru extensiile cu cruce și grid bidimensional, unde probabilitățile estimate sunt în acord cu formulele teoretice.

Graficele realizate evidențiază convergența și caracterul stochastic al metodei.

4.5 Demonstrații teoretice

A

4.6 Modelul probabilistic

4.6.1 Simetrie și reducere la o singură bandă

Distanța dintre două drepte consecutive este 1. Situația se repetă identic în fiecare bandă

$$\{(x, y) \in \mathbb{R}^2 \mid n \leq y \leq n + 1\}.$$

Prin urmare, este suficient să studiem poziția acului într-o singură bandă, de exemplu între dreptele

$$y = 0 \quad \text{și} \quad y = 1.$$

Fie C centrul acului. Proiectăm poziția lui C pe direcția normală la drepte (verticală). Nu contează poziția pe orizontală, ci doar distanța față de cea mai apropiată dreaptă.

4.6.2 Definiția variabilelor aleatoare

Definim două variabile aleatoare:

- Θ – unghiul dintre ac și direcția dreptei (axa orizontală). Prin simetrie:

$$\Theta \in \left[0, \frac{\pi}{2}\right].$$

- X – distanța de la centrul acului C la cea mai apropiată dreaptă.

Într-o bandă de înălțime 1, această distanță aparține intervalului

$$X \in \left[0, \frac{1}{2}\right].$$

4.6.3 Distribuțiile lui X și Θ

Presupunem o aruncare aleatoare în sensul următor:

- orientarea acului este uniformă $\Rightarrow \Theta \sim U\left(0, \frac{\pi}{2}\right)$;
- poziția centrului este uniformă $\Rightarrow X \sim U\left(0, \frac{1}{2}\right)$;
- X și Θ sunt independente.

Densitățile marginale sunt:

$$f_{\Theta}(\theta) = \frac{2}{\pi}, \quad \theta \in \left[0, \frac{\pi}{2}\right],$$

$$f_X(x) = 2, \quad x \in \left[0, \frac{1}{2}\right].$$

Prin independență:

$$f_{X,\Theta}(x, \theta) = f_X(x)f_{\Theta}(\theta) = \frac{4}{\pi},$$

pentru $(x, \theta) \in \left[0, \frac{1}{2}\right] \times \left[0, \frac{\pi}{2}\right]$.

4.7 Condiția geometrică de intersecție

Acul are lungime 1, deci distanța de la centru la fiecare capăt este $\frac{1}{2}$.

Proiecția verticală a jumătății de ac este:

$$\frac{1}{2} \sin \Theta.$$

Acul intersectează o dreaptă dacă și numai dacă:

$$X \leq \frac{1}{2} \sin \Theta.$$

4.8 Calculul probabilității

Definim evenimentul:

$$A = \{\text{acul intersectează cel puțin o dreaptă}\} = \{(X, \Theta) \mid X \leq \frac{1}{2} \sin \Theta\}.$$

Probabilitatea este:

$$P(A) = \int_0^{\pi/2} \int_0^{\frac{1}{2} \sin \theta} f_{X, \Theta}(x, \theta) dx d\theta.$$

4.8.1 Integrarea după x

Pentru θ fix:

$$\int_0^{\frac{1}{2} \sin \theta} \frac{4}{\pi} dx = \frac{2}{\pi} \sin \theta.$$

4.8.2 Integrarea după θ

$$P(A) = \frac{2}{\pi} \int_0^{\pi/2} \sin \theta d\theta = \frac{2}{\pi}.$$

4.9 Interpretare geometrică

Spațiul cazurilor posibile este dreptunghiul

$$\left[0, \frac{1}{2}\right] \times \left[0, \frac{\pi}{2}\right],$$

cu aria totală:

$$A_{\text{total}} = \frac{\pi}{4}.$$

Zona favorabilă este sub curba $x = \frac{1}{2} \sin \theta$, cu aria:

$$A_{\text{fav}} = \int_0^{\pi/2} \frac{1}{2} \sin \theta d\theta = \frac{1}{2}.$$

Rezultă:

$$P(A) = \frac{A_{\text{fav}}}{A_{\text{total}}} = \frac{2}{\pi}.$$

4.10 Concluzie

Probabilitatea ca acul să intersecteze o dreaptă este:

$$\boxed{P = \frac{2}{\pi}}.$$

B

4.11 Modelul probabilistic pentru cruce

Considerăm planul secționat de dreptele orizontale

$$y = n, \quad n \in \mathbb{Z},$$

aflate la distanță 1 una față de cealaltă. Pe acest plan aruncăm aleator o cruce formată din două ace de lungime 1, perpendiculare între ele și având același centru.

Notăm cu Z numărul total de intersecții ale crucii cu dreptele planului.

Datorită periodicității configurației, este suficient să analizăm poziția crucii într-o singură bandă delimitată de două drepte consecutive. Introducem următoarele variabile aleatoare:

$$X = \text{distanța de la centrul crucii la cea mai apropiată dreaptă}, \quad X \in \left[0, \frac{1}{2}\right],$$

$$\Theta = \text{unghiul dintre unul dintre ace și direcția dreptei}, \quad \Theta \in \left[0, \frac{\pi}{2}\right].$$

Presupunem că X și Θ sunt independente și uniform distribuite pe intervalele lor.

4.12 Scrierea lui Z ca sumă de indicatori

Definim variabilele indicator:

$$I_1 = \begin{cases} 1, & \text{dacă primul ac intersectează o dreaptă,} \\ 0, & \text{altfel,} \end{cases} \quad I_2 = \begin{cases} 1, & \text{dacă al doilea ac intersectează o dreaptă,} \\ 0, & \text{altfel.} \end{cases}$$

Numărul total de intersecții este:

$$Z = I_1 + I_2.$$

4.13 Calculul valorii așteptate

Fiecare ac este de lungime 1 și se află în aceeași configurație ca acul lui Buffon clasic. Prin urmare,

$$P(I_1 = 1) = P(I_2 = 1) = \frac{2}{\pi}.$$

Rezultă:

$$\mathbb{E}(Z) = \mathbb{E}(I_1) + \mathbb{E}(I_2) = \frac{4}{\pi}.$$

În consecință,

$$\mathbb{E}\left(\frac{Z}{2}\right) = \frac{1}{2}\mathbb{E}(Z) = \frac{2}{\pi},$$

deci $\frac{Z}{2}$ este un estimator neabiazat pentru valoarea $\frac{2}{\pi}$.

4.14 Calculul varianței

Pentru a evalua eficiența estimatorului $\frac{Z}{2}$, calculăm varianța sa:

$$\text{Var}\left(\frac{Z}{2}\right) = \frac{1}{4}\text{Var}(Z), \quad \text{Var}(Z) = \mathbb{E}(Z^2) - [\mathbb{E}(Z)]^2.$$

Deoarece $Z = I_1 + I_2$, avem:

$$Z^2 = (I_1 + I_2)^2 = I_1 + I_2 + 2I_1I_2 = Z + 2I_1I_2,$$

folosind faptul că $I_1^2 = I_1$ și $I_2^2 = I_2$.

Rezultă:

$$\mathbb{E}(Z^2) = \mathbb{E}(Z) + 2\mathbb{E}(I_1I_2).$$

4.15 Probabilitatea ca ambele ace să intersecteze o dreaptă

Primul ac intersectează o dreaptă dacă:

$$X \leq \frac{1}{2} \sin \Theta,$$

iar al doilea ac, fiind perpendicular, intersectează o dreaptă dacă:

$$X \leq \frac{1}{2} \cos \Theta.$$

Prin urmare, ambele ace intersectează o dreaptă dacă și numai dacă:

$$X \leq \frac{1}{2} \min(\sin \Theta, \cos \Theta).$$

Observăm că pe intervalul $[0, \frac{\pi}{2}]$ funcțiile $\sin \theta$ și $\cos \theta$ se intersectează în $\theta = \frac{\pi}{4}$. Prin integrare pe domeniul corespunzător se obține:

$$\mathbb{E}(I_1I_2) = P(I_1 = 1, I_2 = 1) = \frac{4 - 2\sqrt{2}}{\pi}.$$

4.16 Rezultatul final pentru varianță

Înlocuind în expresia varianței:

$$\text{Var}(Z) = \frac{12 - 4\sqrt{2}}{\pi} - \frac{16}{\pi^2},$$

și

$$\text{Var}\left(\frac{Z}{2}\right) = \frac{3 - \sqrt{2}}{\pi} - \frac{4}{\pi^2}.$$

4.17 Concluzie

Atât metoda bazată pe acul lui Buffon, cât și metoda bazată pe cruce conduc la estimatori neabizați pentru $\frac{2}{\pi}$. Totuși, varianța estimatorului obținut prin utilizarea crucii este mai mică decât cea a estimatorului bazat pe un singur ac. Prin urmare, pentru același număr de aruncări, algoritmul aleator bazat pe cruce oferă o estimare mai stabilă și mai precisă a valorii lui π .

C

4.18 Generalizarea problemei acului lui Buffon

Considerăm un plan pe care sunt trasate dreptele orizontale

$$y = nd, \quad n \in \mathbb{Z},$$

aflate la distanță $d > 0$ una față de cealaltă. Pe acest plan aruncăm aleator un ac de lungime L , cu $L < d$.

Dorim să determinăm probabilitatea ca acul să intersecteze cel puțin una dintre drepte și să arătăm că aceasta este:

$$P = \frac{2L}{\pi d}.$$

4.18.1 Modelarea aleatoare

Datorită periodicității configurației (drepte paralele la distanță d), este suficient să analizăm poziția acului într-o singură bandă delimitată de două drepte consecutive, de exemplu între

$$y = 0 \quad \text{și} \quad y = d.$$

Introducem următoarele variabile aleatoare:

$$X = \text{distanța de la centrul acului la cea mai apropiată dreaptă}, \quad X \in \left[0, \frac{d}{2}\right],$$

$$\Theta = \text{unghiul dintre ac și direcția dreptei (axa orizontală)}, \quad \Theta \in \left[0, \frac{\pi}{2}\right].$$

Presupunem că:

- centrul acului este uniform distribuit în bandă $\Rightarrow X \sim U(0, \frac{d}{2})$;
- orientarea acului este complet aleatoare $\Rightarrow \Theta \sim U(0, \frac{\pi}{2})$;
- variabilele X și Θ sunt independente.

Densitățile marginale sunt:

$$f_X(x) = \frac{2}{d}, \quad x \in \left[0, \frac{d}{2}\right], \quad f_\Theta(\theta) = \frac{2}{\pi}, \quad \theta \in \left[0, \frac{\pi}{2}\right].$$

Prin independență, densitatea comună este:

$$f_{X,\Theta}(x, \theta) = \frac{4}{\pi d},$$

pentru $(x, \theta) \in \left[0, \frac{d}{2}\right] \times \left[0, \frac{\pi}{2}\right]$, și 0 în rest.

4.18.2 Condiția geometrică de intersecție

Acul are lungime L , astfel încât distanța de la centru la fiecare capăt este $\frac{L}{2}$.

Proiecția verticală a jumătății de ac pe direcția normală la drepte este:

$$\frac{L}{2} \sin \Theta.$$

Acul intersectează o dreaptă dacă și numai dacă:

$$X \leq \frac{L}{2} \sin \Theta.$$

Observăm că ipoteza $L < d$ garantează că acul nu poate intersecta mai mult de o dreaptă (probabilitatea de a intersecta exact două drepte este zero).

4.18.3 Calculul probabilității

Definim evenimentul:

$$A = \{\text{acul intersectează cel puțin o dreaptă}\} = \{(X, \Theta) \mid X \leq \frac{L}{2} \sin \Theta\}.$$

Probabilitatea dorită este:

$$P(A) = \int_0^{\pi/2} \int_0^{\frac{L}{2} \sin \theta} f_{X,\Theta}(x, \theta) dx d\theta.$$

Înlocuind densitatea comună:

$$P(A) = \int_0^{\pi/2} \int_0^{\frac{L}{2} \sin \theta} \frac{4}{\pi d} dx d\theta.$$

Integrarea după x conduce la:

$$\int_0^{\frac{L}{2} \sin \theta} \frac{4}{\pi d} dx = \frac{2L}{\pi d} \sin \theta.$$

Prin urmare:

$$P(A) = \frac{2L}{\pi d} \int_0^{\pi/2} \sin \theta d\theta.$$

Cum

$$\int_0^{\pi/2} \sin \theta d\theta = 1,$$

obținem rezultatul final:

$$\boxed{P(A) = \frac{2L}{\pi d}}.$$

D

4.19 Intersecția unui ac fix cu o linie aleatoare

Considerăm un ac de lungime L , fixat într-un plan, și notăm cu C cercul de diametru d , având centrul în mijlocul acului. Presupunem că $L < d$. Fie λ o linie aleatoare din plan, a cărei direcție și distanță față de centrul cercului C sunt variabile aleatoare independente și uniform distribuite pe intervalele

$$[0, \pi], \quad \text{respectiv} \quad \left[0, \frac{d}{2}\right].$$

Scopul este de a determina probabilitatea ca linia λ să intersecteze acul.

4.19.1 Modelarea aleatoare

Fixăm un sistem de coordonate carteziene în plan, cu originea în centrul acului. Fără pierderea generalității, considerăm acul așezat pe axa Ox , cu capetele în punctele

$$A = \left(-\frac{L}{2}, 0\right), \quad B = \left(\frac{L}{2}, 0\right).$$

O linie λ din plan poate fi descrisă în forma normală (Hesse) prin doi parametri:

- unghiul $\alpha \in [0, \pi]$ dintre linie și axa Ox ;
- distanța $R \in \left[0, \frac{d}{2}\right]$ de la origine la linie, măsurată pe perpendiculară.

Prin ipoteză, variabilele aleatoare α și R sunt independente și uniform distribuite pe intervalele menționate. Spațiul probabil este astfel:

$$\Omega = [0, \pi] \times \left[0, \frac{d}{2}\right],$$

echipat cu densitatea de probabilitate constantă:

$$f_{\alpha,R}(\alpha, R) = \frac{1}{\pi} \cdot \frac{2}{d} = \frac{2}{\pi d}.$$

4.19.2 Condiția de intersecție

Linia λ intersectează acul dacă și numai dacă distanța de la linie la axa Ox este mai mică sau egală cu proiecția semilungimii acului pe direcția normală la linie.

Dacă α este unghiul dintre ac și linie, atunci componenta semilungimii acului pe direcția perpendiculară pe linie este:

$$\frac{L}{2} \sin \alpha.$$

Rezultă că linia λ intersectează acul dacă și numai dacă:

$$0 \leq R \leq \frac{L}{2} \sin \alpha.$$

Deoarece $L < d$, avem:

$$\frac{L}{2} \sin \alpha \leq \frac{L}{2} < \frac{d}{2}, \quad \forall \alpha \in [0, \pi],$$

astfel încât această condiție este compatibilă cu domeniul de variație al lui R .

4.19.3 Calculul probabilității

Probabilitatea cerută este aria regiunii favorabile din spațiul probabil, ponderată cu densitatea uniformă:

$$P(\lambda \text{ intersectează acul}) = \iint_{\{0 \leq R \leq \frac{L}{2} \sin \alpha\}} f_{\alpha, R}(\alpha, R) dR d\alpha.$$

Înlocuind densitatea:

$$P = \frac{2}{\pi d} \int_0^\pi \left(\int_0^{\frac{L}{2} \sin \alpha} dR \right) d\alpha.$$

Calculând integrala interioară:

$$\int_0^{\frac{L}{2} \sin \alpha} dR = \frac{L}{2} \sin \alpha.$$

Prin urmare:

$$P = \frac{2}{\pi d} \int_0^\pi \frac{L}{2} \sin \alpha d\alpha = \frac{L}{\pi d} \int_0^\pi \sin \alpha d\alpha.$$

Cum

$$\int_0^\pi \sin \alpha d\alpha = 2,$$

rezultă:

$$\boxed{P = \frac{2L}{\pi d}}.$$

4.19.4 Concluzie

Probabilitatea ca un ac fix să fie intersectat de o linie aleatoare, aleasă cu direcție și distanță față de centru uniforme și independente, este:

$$P = \frac{2L}{\pi d}.$$

Aceasta coincide cu probabilitatea clasică din problema acului lui Buffon, demonstrând că formularea „acul aleator – linii fixe” și formularea duală „acul fix – linie aleatoare” sunt echivalente din punct de vedere probabilistic.

E

4.20 Acul într-o rețea dreptunghiulară de drepte paralele

4.20.1 Modelarea aleatoare

Fixăm un sistem de coordonate astfel încât:

- dreptele verticale să fie de forma $x = kd_1$;
- dreptele orizontale să fie de forma $y = kd_2$, $k \in \mathbb{Z}$.

Datorită periodicității, este suficient să analizăm poziția acului într-un singur dreptunghi fundamental:

$$[0, d_1] \times [0, d_2].$$

4.20.2 Variabilele aleatoare

Notăm:

- α – unghiul acului față de direcția verticală (axa Oy). Prin simetrie:

$$\alpha \sim U\left[0, \frac{\pi}{2}\right], \quad f_\alpha(\alpha) = \frac{2}{\pi}.$$

- X – distanța de la centrul acului la cea mai apropiată dreaptă verticală:

$$X \sim U\left[0, \frac{d_1}{2}\right], \quad f_X(x) = \frac{2}{d_1}.$$

- Y – distanța de la centrul acului la cea mai apropiată dreaptă orizontală:

$$Y \sim U\left[0, \frac{d_2}{2}\right], \quad f_Y(y) = \frac{2}{d_2}.$$

Presupunem că α , X și Y sunt independente. Prin urmare, densitatea comună este constantă:

$$f_{\alpha, X, Y}(\alpha, x, y) = f_\alpha(\alpha) f_X(x) f_Y(y) = \frac{2}{\pi} \cdot \frac{2}{d_1} \cdot \frac{2}{d_2} = \frac{8}{\pi d_1 d_2},$$

pentru

$$\alpha \in \left[0, \frac{\pi}{2}\right], \quad x \in \left[0, \frac{d_1}{2}\right], \quad y \in \left[0, \frac{d_2}{2}\right].$$

4.20.3 Condițiile de intersecție

Intersecție cu o dreaptă verticală. Dacă acul face unghiul α cu verticala, atunci proiecția semilungimii acului pe direcția orizontală (perpendiculară pe dreptele verticale) este

$$\frac{L}{2} \sin \alpha.$$

Acul atinge o dreaptă verticală dacă și numai dacă:

$$X \leq \frac{L}{2} \sin \alpha.$$

Intersecție cu o dreaptă orizontală. Față de direcția orizontală, acul face unghiul $\frac{\pi}{2} - \alpha$, deci proiecția semilungimii acului pe direcția verticală este

$$\frac{L}{2} \cos \alpha.$$

Acul atinge o dreaptă orizontală dacă și numai dacă:

$$Y \leq \frac{L}{2} \cos \alpha.$$

Observăm că, deoarece $L < \min\{d_1, d_2\}$,

$$\frac{L}{2} \sin \alpha \leq \frac{L}{2} < \frac{d_1}{2}, \quad \frac{L}{2} \cos \alpha \leq \frac{L}{2} < \frac{d_2}{2},$$

pentru orice $\alpha \in [0, \frac{\pi}{2}]$, deci limitele de sus rămân în intervalele lui X , respectiv Y .

Definim evenimentele:

$$A = \{\text{acul intersectează o dreaptă verticală}\}, \quad B = \{\text{acul intersectează o dreaptă orizontală}\}.$$

Noi vrem:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

4.20.4 Calculul lui $P(A)$

Condiționat pe α , avem:

$$P(A | \alpha) = P\left(X \leq \frac{L}{2} \sin \alpha\right) = \frac{\frac{L}{2} \sin \alpha}{\frac{d_1}{2}} = \frac{L}{d_1} \sin \alpha.$$

Apoi mediem după α :

$$P(A) = \int_0^{\pi/2} P(A | \alpha) f_\alpha(\alpha) d\alpha = \int_0^{\pi/2} \frac{L}{d_1} \sin \alpha \cdot \frac{2}{\pi} d\alpha = \frac{2L}{\pi d_1} \int_0^{\pi/2} \sin \alpha d\alpha = \frac{2L}{\pi d_1}.$$

Analog, pentru evenimentul B :

$$P(B) = \frac{2L}{\pi d_2}.$$

4.20.5 Calculul lui $P(A \cap B)$

Condiționat pe α , evenimentele

$$\{X \leq \frac{L}{2} \sin \alpha\}, \quad \{Y \leq \frac{L}{2} \cos \alpha\}$$

sunt independente (deoarece X și Y sunt independente). Deci:

$$P(A \cap B | \alpha) = P\left(X \leq \frac{L}{2} \sin \alpha\right) P\left(Y \leq \frac{L}{2} \cos \alpha\right) = \frac{L}{d_1} \sin \alpha \cdot \frac{L}{d_2} \cos \alpha = \frac{L^2}{d_1 d_2} \sin \alpha \cos \alpha.$$

Mediem după α :

$$P(A \cap B) = \int_0^{\pi/2} P(A \cap B | \alpha) f_\alpha(\alpha) d\alpha = \int_0^{\pi/2} \frac{L^2}{d_1 d_2} \sin \alpha \cos \alpha \cdot \frac{2}{\pi} d\alpha = \frac{2L^2}{\pi d_1 d_2} \int_0^{\pi/2} \sin \alpha \cos \alpha d\alpha.$$

Dar

$$\int_0^{\pi/2} \sin \alpha \cos \alpha d\alpha = \frac{1}{2} \int_0^{\pi/2} \sin(2\alpha) d\alpha = \frac{1}{2} \cdot 1 = \frac{1}{2},$$

deci

$$P(A \cap B) = \frac{L^2}{\pi d_1 d_2}.$$

4.20.6 Probabilitatea totală

Aplicăm formula de includere-excludere:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) = \frac{2L}{\pi d_1} + \frac{2L}{\pi d_2} - \frac{L^2}{\pi d_1 d_2}.$$

Scoatem factorul comun:

$$P(A \cup B) = \frac{L}{\pi d_1 d_2} (2d_2 + 2d_1 - L) = \frac{L(2d_1 + 2d_2 - L)}{\pi d_1 d_2}.$$

F

4.21 Tipul strategiei aleatoare utilizate

Strategia aleatoare folosită în simulările prezentate (problema acului lui Buffon, crucea, respectiv gridul bidimensional) este de tip *Monte Carlo*.

4.21.1 Algoritmi Monte Carlo

Un algoritm Monte Carlo se caracterizează prin următoarele proprietăți:

- timpul de execuție este determinist, fiind fixat în prealabil (de exemplu, printr-un număr N de simulări);
- rezultatul obținut este aproximativ, nu exact;
- există o probabilitate nenulă de eroare, însă aceasta scade pe măsură ce crește numărul de simulări;
- precizia rezultatului depinde de legea numerelor mari.

În simulările realizate:

- se execută un număr fix de aruncări aleatoare ale acului;
- se estimează o probabilitate geometrică sau valoarea constantei π ;
- rezultatul nu este garantat exact, dar converge către valoarea teoretică atunci când $N \rightarrow \infty$.

Aceste caracteristici corespund exact definiției unui algoritm Monte Carlo.

4.21.2 De ce strategia NU este de tip Las Vegas

Un algoritm de tip Las Vegas are proprietăți diferite:

- rezultatul furnizat este întotdeauna corect;
- timpul de execuție este aleator, nefiind cunoscut în avans.

În cazul simulărilor analizate:

- nu se garantează obținerea valorii exacte (de exemplu, π nu este determinat exact);
- algoritmul se oprește după un număr fix de pași, deci timpul de execuție nu este aleator.

Prin urmare, strategia utilizată nu poate fi clasificată ca fiind de tip Las Vegas.

4.21.3 Exemplu de algoritm Las Vegas

Un exemplu clasic de algoritm Las Vegas este căutarea aleatoare a unui element într-un vector: se alege poziții aleator până când elementul dorit este găsit. Algoritmul se oprește doar atunci când rezultatul corect este obținut, însă numărul de pași necesari este aleator.

```
las_vegas_search <- function(v, x) {  
  n <- length(v)  
  repeat {  
    i <- sample(1:n, 1)  
    if (v[i] == x) {  
      return(i)  
    }  
  }  
}
```

În acest caz:

- corectitudinea rezultatului este garantată;
- timpul de execuție nu este determinist.

4.21.4 Concluzie

Strategia aleatoare utilizată în simulările de tip Buffon este una Monte Carlo, deoarece algoritmul rulează un număr fix de pași și furnizează o estimare probabilistică a rezultatului, a cărei precizie depinde de numărul de simulări efectuate.

În contrast, algoritmi de tip Las Vegas garantează corectitudinea rezultatului, dar au un timp de execuție aleator.

4.22 Observații

Metoda Monte Carlo oferă o abordare intuitivă și flexibilă pentru estimarea unor mărimi teoretice dificil de calculat exact prin alte metode. Deși rezultatele nu sunt exacte, precizia acestora poate fi controlată prin creșterea numărului de simulări.

Un aspect important este compromisul dintre timpul de execuție și precizie: un număr mai mare de simulări conduce la rezultate mai bune, dar necesită resurse de calcul mai mari.

În concluzie, exercițiul demonstrează eficiența algoritmilor Monte Carlo în probleme de probabilitate geometrică și evidențiază diferența fundamentală față de algoritmi de tip Las Vegas, care garantează corectitudinea rezultatului, dar nu au timp de execuție determinist.

5 Pachete & Surse Bibliografice

5.1 Pachete R Utilizate

5.1.1 Ceașescu Ana-Carina – Exercițiul 1

- **stats** (pachet base R)
 - Utilizat pentru: generarea variabilelor aleatoare uniforme
 - Funcții principale: `runif()`, `sqrt()`
- **graphics** (pachet base R)
 - Utilizat pentru: reprezentări grafice ale punctelor
 - Funcții principale: `plot()`, `points()`, `abline()`

Scripturile au fost rulate folosind **R version 4.5.2 (2025-10-31)**.

5.1.2 Moga Eduard-Andrei – Exercițiul 2

- **shiny** (versiunea $\geq 1.7.0$) – Framework pentru aplicații web interactive în R
 - *Utilizat pentru:* Crearea interfeței grafice, gestionarea reactivității, rendering grafice dinamice
 - *Funcții principale:* `fluidPage()`, `renderPlot()`, `reactive()`, `eventReactive()`
- **stats** (pachet base R) – Funcții statistice și de distribuții
 - *Utilizat pentru:* Generare eșantioane aleatorii, calcul statistici descriptive
 - *Funcții principale:* `rnorm()`, `rexp()`, `rpois()`, `rbinom()`, `ecdf()`, `quantile()`
- **graphics** (pachet base R) – Funcții grafice
 - *Utilizat pentru:* Crearea histogramelor, grafice ECDF, Q-Q plots
 - *Funcții principale:* `hist()`, `plot()`, `qqnorm()`, `qqline()`, `abline()`

5.2 Surse Bibliografice și Documentație

5.2.1 Ceașescu Ana-Carina

1. Curs Probabilități și Statistică FMI, Alexandru Amărioarei (2025).
2. Documentația oficială R: <https://www.r-project.org/>

5.2.2 Moga Eduard-Andrei

1. Documentație oficială Shiny

- URL: <https://shiny.posit.co/>
- Tutorial complet: <https://shiny.rstudio.com/tutorial/>
- Utilizat pentru înțelegerea structurii aplicațiilor reactive și best practices

2. Curs Probabilități și Statistică FMI

- URL: <https://alexamarioarei.quarto.pub/curs-ps-fmi/>
- Secțiunile relevante: Introducere în R, Distribuții de probabilitate, Grafică
- Utilizat pentru: Concepte teoretice despre distribuții și transformări

3. R Documentation

- URL: <https://www.rdocumentation.org/>
- Utilizat pentru: Consultarea documentației funcțiilor statistice și grafice

4. Shiny Gallery

- URL: <https://shiny.rstudio.com/gallery/>
- Utilizat pentru: Inspirație în design UI și structură aplicație

5. StackOverflow - R și Shiny

- Utilizat pentru: Rezolvarea problemelor tehnice specifice (ECDF plotting, reactive UI)

5.2.3 Tirdea Dominic Alexandru - Exercițiul 3

1. Grinstead, C. M., Snell, J. L. – Introduction to Probability

- URL: <https://math.dartmouth.edu/~prob/prob/prob.pdf>
- Instituție: Dartmouth College
- Utilizat pentru: capitole despre probabilitate geometrică și metode Monte Carlo

2. Wikipedia – Monte Carlo method

- URL: https://en.wikipedia.org/wiki/Monte_Carlo_method
- Utilizat pentru: prezentare generală a metodelor Monte Carlo și aplicații

3. Wikipedia – Las Vegas algorithm

- URL: https://en.wikipedia.org/wiki/Las_Vegas_algorithm
- Utilizat pentru: clasificarea algoritmilor aleatori și comparația Monte Carlo vs. Las Vegas

4. R Documentation

- URL: <https://www.rdocumentation.org/>
- Utilizat pentru dezvoltarea simularilor

6 Dificultăți Întâmpinate

6.1 Ceașescu Ana-Carina

- **Alegerea modului corect de simulare pentru metoda polară**

La început nu a fost foarte clar de ce nu este suficient să generăm direct R uniform pe intervalul $[0, 1]$. Rezultatele obținute în acest fel nu păreau corecte, deoarece punctele se concentrău prea mult spre centru. Abia după reluarea părții teoretice a devenit clar că trebuie folosită transformarea $R = \sqrt{U}$.

- **Înțelegerea justificării teoretice pentru acceptare-respingere**

Deși metoda este relativ simplă ca implementare, a fost mai dificil de explicat teoretic de ce punctele acceptate sunt într-adevăr uniforme pe disc. A fost nevoie de mai multe încercări pentru a formula o explicație clară, pornind de la distribuția uniformă pe pătrat.

- **Alegerea valorii lui N**

Stabilirea numărului de puncte generate a fost oarecum arbitrară. Pentru valori mici rezultatele variau mult, iar pentru valori mai mari simularea devenea mai lentă, astfel că s-a ales un compromis rezonabil, $N = 1000$.

- **Interpretarea diferențelor dintre rezultate**

Valorile obținute prin cele două metode nu au fost identice, ceea ce inițial a creat impresia că una dintre implementări este greșită. Ulterior s-a înțeles că aceste diferențe sunt normale și apar din cauza caracterului aleator al simulării.

6.2 Moga Eduard-Andrei

- **Implementarea dinamică a transformărilor specifice fiecărei distribuții**

Fiecare distribuție din enunț avea transformări specifice diferite (de exemplu, Normal avea $3 + 2X$, Exponențială avea $2 - 5X$, etc.). Am întâmpinat dificultăți în:

- Crearea unui sistem modular care să permită aplicarea automată a transformării corecte
- Parsarea coeficienților din stringurile de transformare
- Sincronizarea UI-ului pentru a afișa doar transformările valide pentru distribuția selectată

Soluție: Am creat o funcție `apply_transformation()` care primește ca parametru tipul de transformare ca string și aplică logica corespunzătoare folosind structuri conditionale.

- **Gestionarea corectă a dimensiunii eșantionului pentru sume de variabile**

Pentru transformările de tipul $\sum_{i=1}^n X_i$, trebuia să generez $n \times m$ valori (unde m este numărul de variabile de sumat), apoi să le grupez și să le sumez corect. Dificultățile au fost:

- Dimensionarea corectă a vectorului de date brute
- Împărțirea în matrice și sumarea pe linii/coloane
- Validarea că lungimea finală a vectorului transformat este corectă

Soluție: Am folosit funcția `matrix()` pentru restructurare și `rowSums()` pentru agregare eficientă.

- **Sincronizarea UI-ului reactiv cu parametrii distribuțiilor**

Fiecare distribuție are parametri diferiți (Normal are μ și σ , Poisson doar λ , etc.). Am întâmpinat probleme în:

- Afișarea dinamică a controalelor UI doar pentru parametrii relevanți
- Prevenirea erorilor când utilizatorul schimbă rapid distribuția
- Păstrarea valorilor default rezonabile pentru fiecare parametru

Soluție: Am folosit `renderUI()` pentru a crea controale dinamice și `req()` pentru a preveni evaluarea prematură a expresiilor reactive.

- **Optimizarea performanței pentru eșantioane mari (> 1000 observații)**

Pentru $n = 5000$, aplicația devenea lentă la regenerarea graficelor. Problemele identificate:

- Recalcularea inutilă a statisticilor la fiecare render
- Desenarea histogramei cu prea multe bin-uri
- ECDF computațional costisitoare pentru multe puncte

Soluție: Am folosit `eventReactive()` în loc de `reactive()` pentru a controla exact când se regenerează datele și am optimizat numărul de breaks în histogramă.

6.3 Țârdea Dominic-Alexandru

- **Înțelegerea modelului geometric și a variabilelor aleatoare**

O primă dificultate a constat în înțelegerea corectă a modelului geometric și a variabilelor aleatoare implicate.

- Interpretarea geometrică a poziției acului față de drepte
- Corelarea unghiului de orientare cu proiecțiile trigonometrice
- Legătura dintre configurația geometrică și condițiile matematice de intersecție

Soluție: Am analizat fiecare configurație geometrică separat și am formulat explicit condițiile de intersecție folosind proiecții trigonometrice și argumente de simetrie.

- **Formularea corectă a condițiilor de intersecție**

O altă dificultate importantă a fost formularea corectă a condițiilor de intersecție pentru fiecare configurație analizată (acul clasic, crucea, gridul bidimensional).

- Identificarea proiecțiilor relevante ale acului pe direcțiile normale

- Tratarea corectă a cazurilor cu mai multe direcții de intersecție
- Asigurarea coerenței între modele diferite

Soluție: Am derivat analitic condițiile de intersecție pentru fiecare caz și le-am verificat prin comparație cu formulele teoretice cunoscute.

- **Generarea corectă a variabilelor aleatoare**

Din punct de vedere al implementării, o provocare majoră a fost alegerea unei simulări aleatoare corecte, care să respecte ipotezele de uniformitate și independență ale variabilelor.

- Generarea uniformă a poziției centrului acului
- Generarea uniformă a unghiului de orientare
- Evitarea erorilor de interpretare a intervalelor unghiulare

Soluție: Am utilizat funcții standard pentru generarea variabilelor uniforme și am verificat distribuțiile obținute prin teste și reprezentări grafice.

7 Probleme Deschise

7.1 Ceașescu Ana-Carina

- **Automatizarea generării raportului**

În forma actuală, graficele sunt generate manual și apoi inserate în document. O posibilă îmbunătățire ar fi folosirea unor instrumente precum R Markdown sau Quarto pentru a genera automat raportul direct din cod, astfel încât orice modificare a simulărilor să se reflecte imediat în documentație.

- **Analiza erorii și intervale de încredere**

Deși în proiect sunt comparate valori empirice cu cele teoretice, nu este analizată explicit eroarea estimărilor. Ar fi utilă studierea variației erorii în funcție de numărul de simulări N și adăugarea unor intervale de încredere pentru a cuantifica mai clar precizia rezultatelor.

- **Testare și validare a implementării**

Codul a fost verificat în principal prin inspectarea rezultatelor și compararea cu valorile teoretice. O extensie ar fi introducerea unor teste automate simple, care să verifice de exemplu dacă mediile și varianțele obținute se află în intervale rezonabile.

7.2 Moga Eduard-Andrei

- **Extinderea aplicației cu distribuții suplimentare**

În versiunea actuală sunt implementate 5 distribuții conform cerințelor. O îmbunătățire viitoare ar fi adăugarea de noi distribuții precum:

- Distribuția Gamma și Beta (pentru modelare mai flexibilă)
- Distribuția Uniformă (pentru comparație cu celelalte)
- Distribuția t -Student și Chi-pătrat (pentru statistică inferențială)
- Distribuții discrete: Geometrică, Binomială Negativă

Implementarea ar necesita extinderea funcției `gen.sample()` și adăugarea transformărilor specifice pentru fiecare distribuție nouă.

- **Comparație vizuală între distribuția teoretică și ECDF**

Deși aplicația calculează și afișează ECDF, nu există o suprapunere directă cu funcția de repartiție teoretică. Ar fi util să se implementeze:

- Calculul și afișarea funcției $F(x)$ teoretice pe același grafic cu ECDF
- Calcul distanță Kolmogorov-Smirnov: $D_n = \sup_x |F_n(x) - F(x)|$
- Test formal pentru verificarea adecvării la distribuție
- Măsurarea convergenței ECDF către F în funcție de n

- **Funcționalitate de export și salvare a rezultatelor**

Momentan, utilizatorul poate vizualiza rezultatele doar în browser. Îmbunătățiri viitoare:

- Export grafice în format PNG/PDF de înaltă rezoluție
- Export tabel cu statistici în format CSV/Excel
- Salvarea configurației curente (distribuție, parametri, transformare)
- Generarea automată de rapoarte HTML cu toate rezultatele
- Compararea mai multor configurații side-by-side

Implementarea ar putea folosi pachete precum `downloadHandler()` din Shiny și `rmarkdown` pentru rapoarte.

7.3 Țârdea Dominic-Alexandru

- **Analiza erorilor**

O primă problemă deschisă este legată de analiza erorii estimărilor obținute prin metoda Monte Carlo. În cadrul proiectului, s-a observat empiric convergența rezultatelor către valorile teoretice, însă nu a fost realizată o evaluare riguroasă a vitezei de convergență sau a intervalelor de încredere asociate estimărilor. O extindere naturală ar fi studiul variației erorii în funcție de numărul de simulări și compararea acesteia cu estimările teoretice oferite de legea numerelor mari sau de teorema limitei centrale.

- **Generalizarea configurațiilor**

O altă direcție de interes o reprezintă generalizarea configurațiilor geometrice. În acest proiect au fost analizate doar cazurile clasice (linii paralele, cruce, grid dreptunghiular). Ar putea fi studiate configurații mai complexe, cum ar fi grile neuniforme, linii dispuse sub unghiuri arbitrare sau obiecte cu forme diferite de ac (segmente curbe, poligoane).

- **Optimizare algoritmilor**

De asemenea, o problemă deschisă este optimizarea algoritmică a simulărilor. Metoda Monte Carlo utilizată este una directă (brute-force), fără tehnici de reducere a varianței. Introducerea unor metode precum importance sampling sau stratificarea domeniului ar putea conduce la estimări mai precise pentru același timp de execuție.

8 Concluzii

Această secțiune este completată de lider (Ceașescu Ana-Carina).

În cadrul acestui proiect am folosit simulări în R pentru a verifica, prin metode numerice și reprezentări grafice, rezultate teoretice discutate la curs. Cele trei exerciții au acoperit atât simularea distribuțiilor pe mulțimi geometrice, cât și explorarea distribuțiilor clasice printr-o interfață interactivă, respectiv estimarea unor probabilități geometrice prin metode de tip Monte Carlo.

În **Exercițiul 1**, am comparat două abordări pentru generarea punctelor uniforme în discul unitate: acceptare-respingere din pătratul $[-1, 1]^2$ și metoda coordonatelor polare. Rezultatele empirice au confirmat uniformitatea (prin distribuția punctelor în plan) și au validat valoarea teoretică $\mathbb{E}[R] = 2/3$, mediile obținute fiind apropiate pentru $N = 1000$.

În **Exercițiul 2**, aplicația Shiny a permis vizualizarea funcțiilor de repartiție empirice pentru mai multe distribuții și transformări, împreună cu statistici descriptive și intervale de încredere. În special, s-a observat clar efectul transformărilor liniare asupra parametrilor și modul în care transformările neliniare modifică forma distribuției. Pentru sume de variabile i.i.d., rezultatele au ilustrat comportamentul așteptat din teorema limitei centrale, prin apropierea de o distribuție normală atunci când n crește.

În **Exercițiul 3**, simulările inspirate din problema acului lui Buffon au arătat cum probabilități geometrice pot fi estimate numeric și cum acestea converg către valorile teoretice odată cu creșterea numărului de rulări. De asemenea, s-a evidențiat diferența dintre un algoritm de tip Monte Carlo (timp fix, rezultat aproximativ) și unul de tip Las Vegas (rezultat sigur, timp aleator), prin analiza strategiilor folosite.

Per ansamblu, proiectul a arătat că simularea este un instrument practic pentru a înțelege și valida rezultate teoretice, iar reprezentările grafice ajută mult la interpretarea corectă a fenomenelor. Totodată, colaborarea pe GitHub a făcut integrarea codului și a documentației mai ușoară și a ajutat la păstrarea unei structuri clare a proiectului.

Principalele contribuții ale proiectului:

- Implementarea practică a conceptelor teoretice studiate la curs
- Dezvoltarea abilităților de programare în R
- Înțelegerea profundă a comportamentului distribuțiilor de probabilitate
- Colaborare eficientă în echipă și organizare structurată a codului

Proiectul demonstrează aplicabilitatea metodelor de simulare Monte Carlo în verificarea rezultatelor teoretice și oferă instrumente interactive pentru explorarea conceptelor de probabilitate și statistică.

References

- [1] Chang, W., Cheng, J., Allaire, J., et al. (2023). *shiny: Web Application Framework for R*. R package version 1.7.5. <https://CRAN.R-project.org/package=shiny>
- [2] R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>
- [3] Amărioarei, A. (2025). *Curs Probabilități și Statistică 2025-2026*. Facultatea de Matematică și Informatică, Universitatea din București. <https://alexamarioarei.quarto.pub/curs-ps-fmi/>