Probabilități și Statistică
Proiect

Nedelcu Ionuț-Daniel

Rusu Ana-Maria

Văcaru Marta-Patricia

An Universitar 2024-2025

Grupa 242

CUPRINS

Problema 1 4

1. Justificați teoretic că putem simula un vector (cuplu) aleator repartizat uniform pe pătratul $[-1, 1]^2$ plecând de la două v.a. independente repartizate uniform pe segmentul $[-1,1]$. 4

2. Prin metoda acceptării si respingerii, simulați N = 1000 de puncte independente repartizate uniform pe discul unitate D(1). Reprezentați grafic punctele (Xi,Yi) din interiorul discului unitate cu albastru și pe celelalte cu roșu. 6

3. Calculați media aritmetică a distanței care separă cele N puncte de origine. Comparați rezultatul cu media teoretică a variabilei corespunzătoare. 9

O a doua metodă de simulare a unui punct (X,Y) repartizat uniform pe D(1) consta in folosirea schimbarii de variabila in coordonate polare: X = Rcos(Θ) si Y = Rsin(Θ). 4. Plecând de la densitatea cuplului (X,Y), găsiți densitatea v.a. R si Θ. 13

5. Simulați N = 1000 de puncte prin această metodă si ilustrati grafic aceste puncte (inclusiv conturul cercului). 14

Problema 2 16

Aplicațiile Shiny 16

Exemplul de bază 17

Cod Sursă 18

User Interface 24

Server 25

Apel final 28

Modul de funcționare al aplicației 28

Alegerea tipului de variabilă aleatoare în meniu 33

Problema 3 35

a) Pe un plan sunt trasate liniile y = n (cu n = 0, ±1, ±2, . . .) și un ac de lungime 1 este aruncat aleator pe acest plan. Arătați că probabilitatea ca acul să intersecteze vreo linie este egală cu 2/π. 35

b) Planul este secționat ca mai sus, de liniile y = n (unde n = 0, ±1, ±2, . . .), iar pe plan aruncăm o cruce formată prin unirea mijloacelor a două ace perpendiculare, de lungime 1. Notăm cu Z numărul de intersecții ale crucii cu liniile de pe plan. Arătați că E[Z/2] = 2/π, iar Var(Z/2) = $3 - 2\pi - 4\pi 2$. Dacă ați avea de ales între a folosi acul sau crucea (deci unul din acești doi algoritmi aleatori) pentru a estima valoarea lui π, ce ați alege? De ce? 41

c1) Considerați acum o valoare d fixată și un plan pe care sunt trasate liniile y = n · d (unde n = 0, ±1, ±2, . . .). Un ac de lungime L(< d) este aruncat la întâmplare pe acest plan. Arătați că probabilitatea ca acul să intersecteze vreo linie din plan este egală cu (2L)/(πd). 48

c2) Mai departe, fixați acum poziția acului și considerați un cerc C de diametru d, centrat în mijlocul acului. Fie λ o linie ale cărei direcție și distanță față de centrul lui C sunt independente și uniform distribuite pe [0, 2π] și, respectiv, [0, d/2]. Arătați că probabilitatea ca acul să se intersecteze cu linia (aleatoare) λ este egală (tot cu) (2L/ πd). 52

d) Pe un plan se consideră următorul grid format cu 2 seturi de linii paralele suprapuse: primul set conține linii la distanța d1 unele de altele, iar al doilea set conține linii la distanța d2 unele de altele și perpendiculare pe cele din primul set. Un ac de lungime L < min{d1, d2} este aruncat la întâmplare pe acest plan. Arătați că probabilitatea ca acesta să intersecteze planul este egală cu L*(2*d1+2*d2−L)/(π*d1*d2). 57

e) În cadrul simulărilor pentru această problemă ce tip de strategie aleatoare a fost implementată: Las Vegas sau Monte Carlo? Justificați sumar și dați un exemplu de algoritm aleator (cu implementare) din cealalta categorie. 65

Bibliografie 68

Observație: liderul echipei – Nedelcu Ionuț-Daniel.

Problema 1

Obiectivul acestui exercițiu este de a simula un vector aleator (X1,X2) repartizat uniform pe discul unitate D(1) (discul de centru (0,0) si de raza 1). Densitatea acestuia este:

$$f(X1,X2)(x1, x2) = 1/\pi\cdot 1D(1)(x1,x2).$$

Pentru aceasta, vom folosi două metode. O primă metodă este metoda de simulare prin acceptare si respingere. Această metodă este des utilizată pentru generarea unei v.a. repartizate uniform pe o mulțime oarecare E. Metoda constă în generarea unei v.a. X repartizată uniform pe o mulțime F ⊇ E, mai simplă decât E, apoi testăm daca X se află in E sau nu. În caz afirmativ, păstrăm X, altfel generăm o nouă realizare a lui X pe F.

1. Justificați teoretic că putem simula un vector (cuplu) aleator repartizat uniform pe pătratul $[-1, 1]^2$ plecând de la două v.a. independente repartizate uniform pe segmentul $[-1,1]$.

Fie X , Y două variabile aleatoare continue independente, astfel încât X ~ U ([-1,1]) și Y ~ U ([-1,1]).

Dacă X, Y sunt variabile aleatoare continue independente atunci densitatea comună este produsul densităților marginale ale celor două variabile. Notăm cu (1) propozițtia enunțată anterior.

Considerăm g funția densitate marginală. Atunci:

$$g_X(x) = \frac{1}{1-(-1)} = \frac{1}{2}, \text{ dacă } x \in [-1,1]$$
$$g_Y(y) = \frac{1}{1-(-1)} = \frac{1}{2}, \text{ dacă } y \in [-1,1]$$

Astfel, densitatea este:

$$g_X(x) = \begin{cases} \frac{1}{2}, \forall\, x \in [-1,1] \\ 0, altfel \end{cases}$$

$$g_Y(y) = \begin{cases} \frac{1}{2}, \forall\, y \in [-1,1] \\ 0, altfel \end{cases}$$

Conform propoziției (1) ⇒ $g_{X,Y}(x,y) = g_X(x) \cdot g_Y(y) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$, pentru $x \in [-1,1]$ și $y \in [-1,1]$.

Astfel,

$$g_{X,Y}(x,y) = \begin{cases} \frac{1}{4}, \forall\ x \in [-1,1]\ și\ y \in [-1,1] \\ 0, altfel \end{cases}$$

O variabilă este repartizată uniform pe un interval, dacă densitatea acelei variabile este o funcție continuă. Pentru că funcția g este constantă, înseamnă ca este și continuă. Astfel, (X,Y) ~ U($[-1,1]^2$).

Așadar, am demonstrat că putem simula un vector aleator repartizat uniform pe pătratul $[-1,1]^2$ plecând de la două variabile alatoare independente repartizate uniform pe segmentul $[-1,1]$.

Explicație: Pentru rezolvarea primei cerințe a problemei prezente, adică pentru a genera un vector aleator (X1,X2) repartizat uniform pe pătratul [−1,1] x [-1,1], putem folosi două variabile aleatoare independente, fiecare având o distribuție uniformă pe intervalul [−1,1]. Deoarece aceste variabile sunt independente, densitatea lor comună este produsul densităților marginale, rezultând o valoare constantă de 1/4 pe întregul pătrat. Fiind constantă, deci continua, această densitate confirmă că distribuția vectorului este uniformă pe [−1,1]x[-1,1], demonstrând astfel posibilitatea generării dorite.

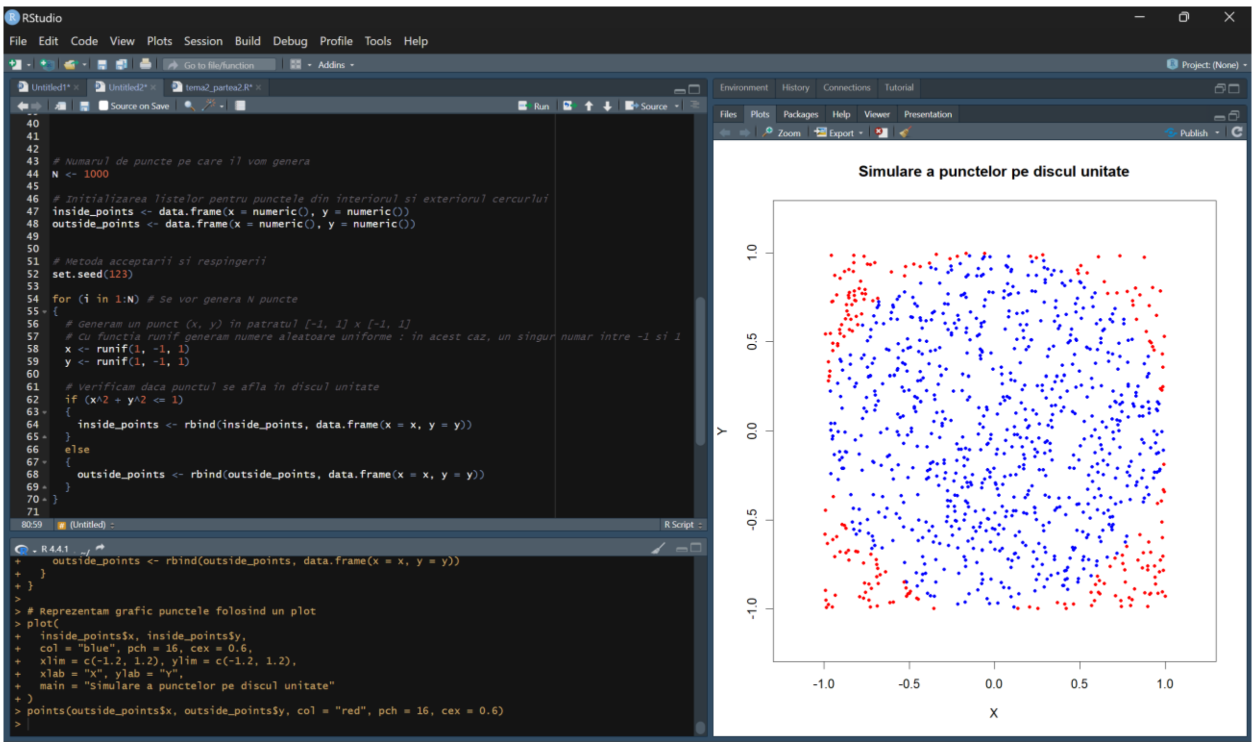
2. Prin metoda acceptării si respingerii, simulați N = 1000 de puncte independente repartizate uniform pe discul unitate D(1). Reprezentați grafic punctele (Xi,Yi) din interiorul discului unitate cu albastru și pe celelalte cu roșu.

```r
# Numarul de puncte pe care il vom genera
N <- 1000

# Initializarea listelor pentru punctele din interiorul si exteriorul cercurlui
inside_points <- data.frame(x = numeric(), y = numeric())
outside_points <- data.frame(x = numeric(), y = numeric())

# Metoda acceptarii si respingerii
set.seed(123)

for (i in 1:N) # Se vor genera N puncte
{
  # Generam un punct (x, y) în patratul [-1, 1] x [-1, 1]
  # Cu functia runif generam numere aleatoare uniforme : in acest caz, un singur
numar intre -1 si 1
  x <- runif(1, -1, 1)
  y <- runif(1, -1, 1)

  # Verificam daca punctul se afla în discul unitate
  if (x^2 + y^2 <= 1)
  {
    inside_points <- rbind(inside_points, data.frame(x = x, y = y))
  }
  else
  {
    outside_points <- rbind(outside_points, data.frame(x = x, y = y))
  }
}

# Reprezentam grafic punctele folosind un plot
plot(
  inside_points$x, inside_points$y,
  col = "blue", pch = 16, cex = 0.6,
  xlim = c(-1.2, 1.2), ylim = c(-1.2, 1.2),
  xlab = "X", ylab = "Y",
  main = "Simulare a punctelor pe discul unitate"
)
points(outside_points$x, outside_points$y, col = "red", pch = 16, cex = 0.6)
```

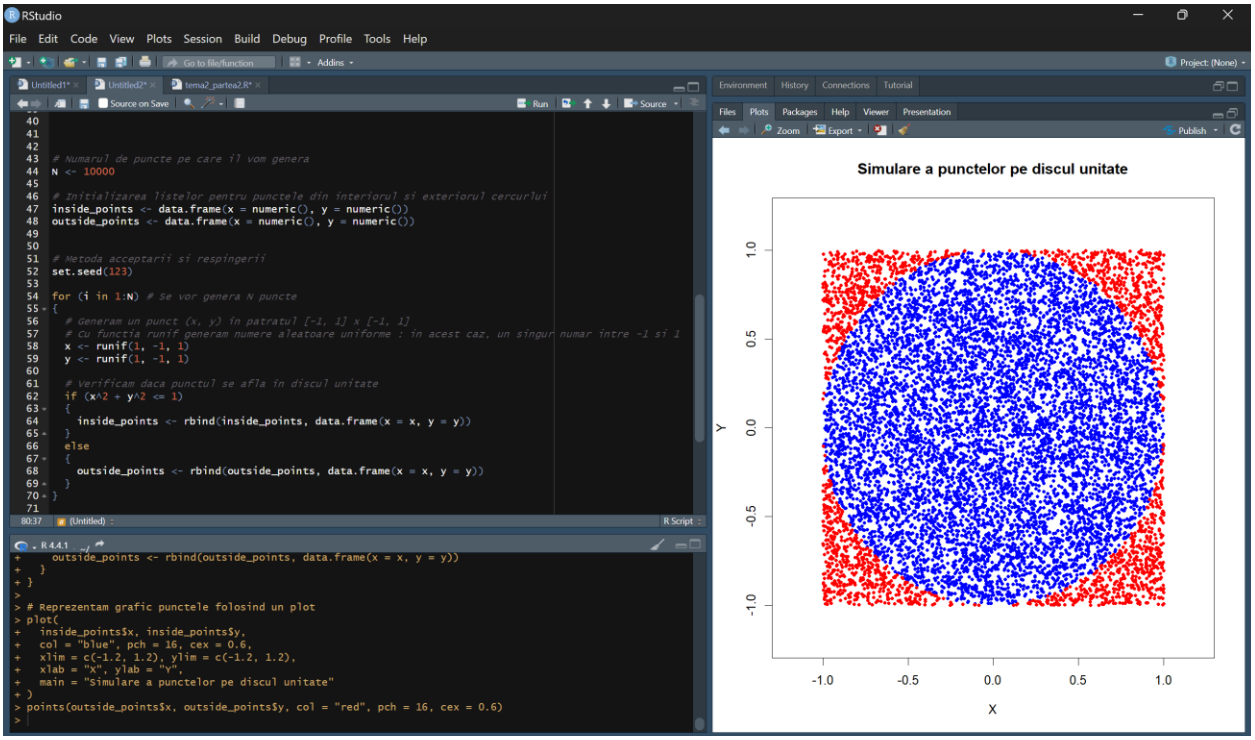
Explicație: Programul R descris mai sus, demonstrează metoda acceptării și respingerii prin generarea aleatorie a unor puncte într-un pătrat și clasificarea acestora în funcție de poziția față de un discul unitate. În acest context, mulțimea F este chiar patratul [−1,1] x [-1,1], iar mulțimea E discul unitate. Fiecare punct generat este evaluat pentru a verifica dacă se află în interiorul cercului, utilizând ecuația $x^2+y^2 \leq 1$. Punctele care îndeplinesc această condiție sunt adăugate într-o listă pentru punctele interioare, în timp ce celelalte sunt salvate ca puncte exterioare. După ce toate punctele sunt generate, ele sunt reprezentate grafic folosind un plot.

Ilustrare:

Pentru N=1 000 vom obține:



Un alt exemplu pentru N=10 000 pentru a se contura mai clar atât cercul cât și pătratul:



3. Calculați media aritmetică a distanței care separă cele N puncte de origine. Comparați rezultatul cu media teoretică a variabilei corespunzătoare.

Media teoretică:

Fie h : $R^2 \rightarrow R^2$ h(x,y) = $\sqrt{x^2+y^2}$, funcția ce calculează distanța de la origine la punctul de coordonate (x,y) în planul cartezian.

Pentru că h este integrabilă atunci $E[h(x,y)] = \iint_D f(x,y)g(x,y)\,dxdy$, unde D = {(x,y) ∈ R^2 | x, y ∈ [-1,1], $x^2 + y^2 \leq 1$} . Astfel, vom obține :

$$E[h(x,y)] = \iint\limits_D f(x,y)\,g(x,y)\,dxdy = \iint\limits_D \frac{1}{\pi}\sqrt{x^2+y^2}\,dxdy = \frac{1}{\pi}\iint\limits_D \sqrt{x^2+y^2}\,dxdy$$

Pentru a obține valoarea integralei, (și pentru a ușura calculele) se va folosi schimbarea la coordonate polare:

Fie $x = \mathrm{R}\cos\Theta$, $y = \mathrm{R}\sin\Theta$, unde R ≥ 0 (pentru ca este vorba despre raza) $si\ \Theta \in [0,2\pi]$ $(pentru\ ca\ este\ vorba\ despre\ unghi)$.

Vom afla noul domeniu D în coordonate polare, înlocuind variabilele x și y cu definițiile corespunzătoare. Se va lua pe rând fiecare relație din D:

$$x^2 + y^2 \leq 1 \Leftrightarrow R^2cos^2\Theta + R^2sin^2\Theta \leq 1 \Leftrightarrow R^2(cos^2\Theta + sin^2\Theta) \leq 1 \Leftrightarrow R^2 \leq 1$$
$$\Leftrightarrow -1 \leq R \leq 1 .$$

Cum R $\in [0,\infty]$ ⇒ R $\in [0,1]$:

$$-1 \leq \mathrm{x} \leq 1 \Leftrightarrow -1 \leq \mathrm{R}\cos\Theta \leq 1 . \mathrm{Cum\ R} \in [0,1] , putem\ spune\ că\ \Theta \in [0,2\pi].$$

$$-1 \leq \mathrm{y} \leq 1 \Leftrightarrow -1 \leq \mathrm{R}\sin\Theta \leq 1 . \mathrm{Cum\ R} \in [0,1] , putem\ spune\ că\ \Theta \in [0,2\pi].$$

Din cele 3 relații de mai sus, concluzionăm că că $\Theta \in [0,2\pi]$ și R $\in [0,1]$.

Pentru a schimba variabilele la coordonate polare, calculăm Jacobianul (noțiune din cursul de Calcul Diferențial și Integral, anul 1, semestrul 1):

$$\Delta = \left\|\begin{matrix} \frac{\partial \mathrm{x}}{\partial \mathrm{R}} & \frac{\partial \mathrm{x}}{\partial \Theta} \\ \frac{\partial \mathrm{y}}{\partial \mathrm{R}} & \frac{\partial \mathrm{y}}{\partial \Theta} \end{matrix}\right\| = \left\|\begin{matrix} \cos\Theta & -R\sin\Theta \\ \sin\Theta & \mathrm{R}\cos\Theta \end{matrix}\right\| = |\ \mathrm{R}(cos^2\Theta + sin^2\Theta\ )| = |\mathrm{R}|.\ \mathrm{Cum\ R} \in [0,1] \Rightarrow \Delta = \mathrm{R}.$$

$$E[h(x,y)] = \frac{1}{\pi}\int_{-1}^{1}\int_{-1}^{1}\sqrt{x^2+y^2}\,dxdy = \frac{1}{\pi}\int_{0}^{2\pi}\int_{0}^{1}\sqrt{R^2cos^2\Theta + R^2sin^2\Theta}\,\Delta\,dRd\Theta$$

$$= \frac{1}{\pi}\int_{0}^{2\pi}\int_{0}^{1}\sqrt{R^2(cos^2\Theta + sin^2\Theta)}\,R\,dRd\Theta = \frac{1}{\pi}\int_{0}^{2\pi}\int_{0}^{1}\sqrt{R^2}\,R\,dRd\Theta = \frac{1}{\pi}\int_{0}^{2\pi}\int_{0}^{1}|R|\,R\,dRd\Theta =$$

$$= \frac{1}{\pi}\int_{0}^{2\pi}\int_{0}^{1}R^2\,dRd\Theta == \frac{1}{\pi}\int_{0}^{2\pi}\frac{R^3}{3}\Big|_{0}^{1}\;d\Theta = \frac{1}{\pi}\int_{0}^{2\pi}\frac{1}{3}\,d\Theta = \frac{1}{3\pi}\int_{0}^{2\pi}d\Theta = \frac{1}{3\pi}\,\Theta\,\Big|_{0}^{2\pi} = \frac{2\pi}{3\pi}$$

$$= \frac{2}{3}$$

Deci, media teoretică $E[h(x,y)] = \frac{2}{3}$

Calculul mediei în R:

```
set.seed(123)
N <- 100000  # Numarul de puncte
suma <- 0  # Variabila pentru suma distantelor ce ne va ajuta la calculul mediei
aritmetice
count <- 0  # Contor pentru punctele care sunt in interiorul discului unitate

# Generarea punctelor si calcularea distantei de la fiecare punct la origine
for (i in 1:N)
{
  # Generam un punct (X, Y) în patratul [-1, 1] x [-1, 1]
  # Cu functia runif generam numere aleatoare uniforme : in acest caz, un singur
numar intre -1 si 1
  X <- runif(1, -1, 1)
  Y <- runif(1, -1, 1)

  # Verificam dacă punctul se afla în interiorul discului
  if (X^2 + Y^2 <= 1)
  {
    # Calculam distanța față de origine
    distanta <- sqrt(X^2 + Y^2)
    suma <- suma + distanta  # Adunam distanta la suma
    count <- count + 1  # Incrementăm contorul
  }
}
```

```r
# Calculul mediei aritmetice a distantelor
media <- suma / count

# Afisarea rezultatului
cat("Media aritmetica a distantelor:", media, "\n")
```

Explicație: Programul R descris mai sus, generează puncte aleatorii în interiorul unui pătrat [-1,1] x [-1,1]. Apoi, verifică dacă punctele generate se află în interiorul unui disc unitate (cu raza 1 și centrul în origine). Pentru punctele care se află în interiorul discului, se calculează distanța față de origine și această distanță este adăugată la o sumă totală. În final, media aritmetică a distanțelor este calculată împărțind suma totală a distanțelor la numărul de puncte care se află în interiorul discului. Astfel, se oferă o estimare a mediei distanțelor pentru punctele aflate într-un disc unitate în raport cu centrul său.

Pentru N = 100 media obtinută a fost 0.6570258 < 6.6666667 (media teoretică)

Pentru N = 1000 media obtinută a fost 0.6628566 < 6.6666667 (media teoretică)

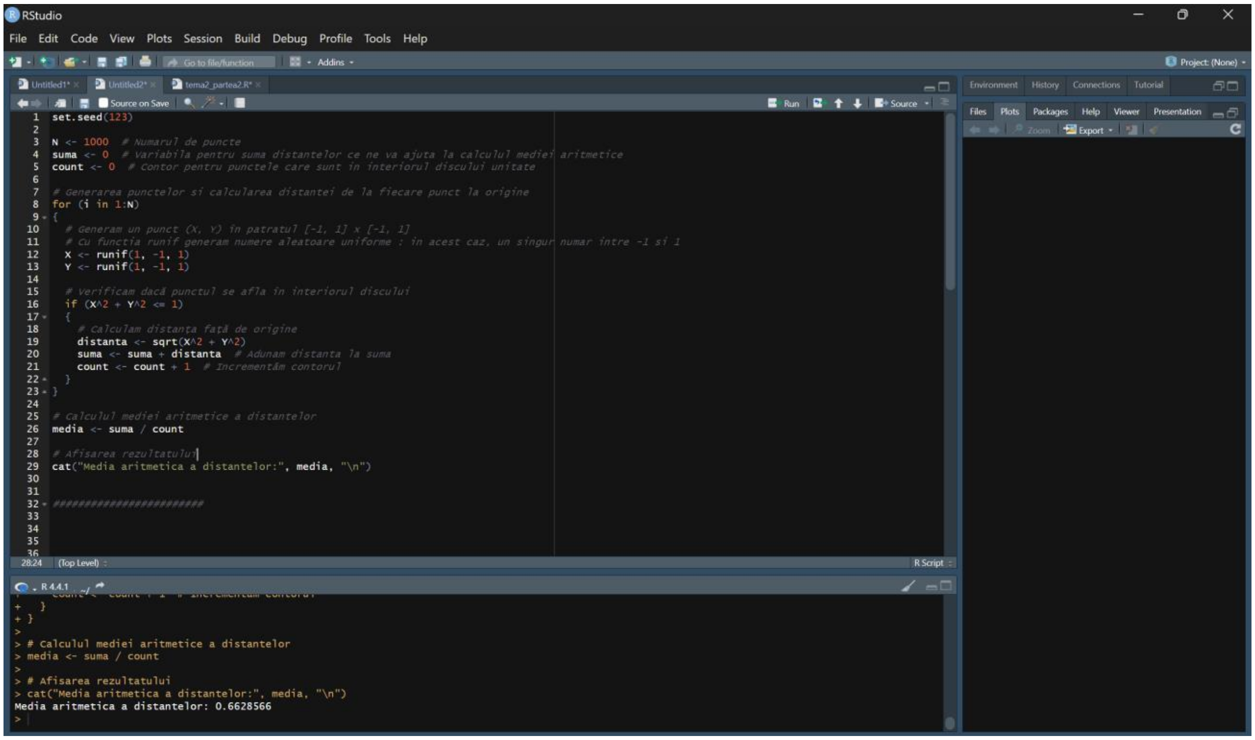
Pentru N = 10000 media obtinută a fost 0.6630516 < 6.6666667 (media teoretică)

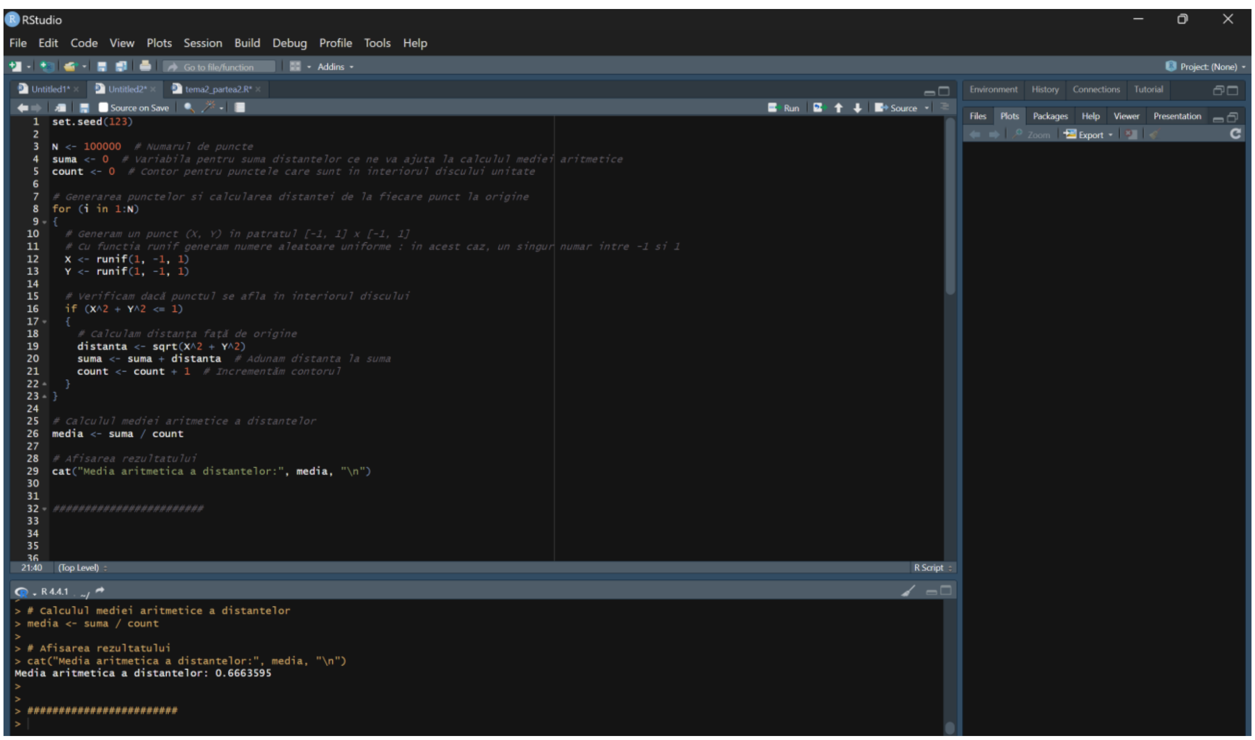
Pentru N = 100000 media obținuta a fost 0.6663595 < 6.6666667 (media teoretică)

Pentru N = 1000000 media obtinută a fost 0.6664819 < 6.6666667 (media teoretică)

Se observă că pe măsură ce N crește, media distanțelor se apropie tot mai mult de valoarea teoretică, 6.6666667 sau 2/3.

Ilustrare:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
set.seed(123)
N <- 1000 # Numarul de puncte
suma <- 0 # Variabila pentru suma distantelor ce ne va ajuta la calculul mediei aritmetice
count <- 0 # Contor pentru punctele care sunt in interiorul discului unitate
# Generarea punctelor si calcularea distantei de la fiecare punct la origine
for (i in 1:N)
{
# Generam un punct (X, Y) in patratul [-1, 1] x [-1, 1]
# Cu functia runif generam numere aleatoare uniforme : in acest caz, un singur numar intre -1 si 1
X <- runif(1, -1, 1)
Y <- runif(1, -1, 1)
# Verificam dacă punctul se afla in interiorul discului
if (X^2 + Y^2 <= 1)
{
# Calculam distanta față de origine
distanta <- sqrt(X^2 + Y^2)
suma <- suma + distanta # Adunam distanta la suma
count <- count + 1 # Incrementăm contorul
}
}
# Calculul mediei aritmetice a distantelor
media <- suma / count
# Afisarea rezultatului
cat("Media aritmetica a distantelor:", media, "\n")
########################
> # Calculul mediei aritmetice a distantelor
> media <- suma / count
> # Afisarea rezultatului
> cat("Media aritmetica a distantelor:", media, "\n")
Media aritmetica a distantelor: 0.6628566


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
set.seed(123)
N <- 100000 # Numarul de puncte
suma <- 0 # Variabila pentru suma distantelor ce ne va ajuta la calculul mediei aritmetice
count <- 0 # Contor pentru punctele care sunt in interiorul discului unitate
# Generarea punctelor si calcularea distantei de la fiecare punct la origine
for (i in 1:N)
{
# Generam un punct (X, Y) in patratul [-1, 1] x [-1, 1]
# Cu functia runif generam numere aleatoare uniforme : in acest caz, un singur numar intre -1 si 1
X <- runif(1, -1, 1)
Y <- runif(1, -1, 1)
# Verificam dacă punctul se afla in interiorul discului
if (X^2 + Y^2 <= 1)
{
# Calculam distanta față de origine
distanta <- sqrt(X^2 + Y^2)
suma <- suma + distanta # Adunam distanta la suma
count <- count + 1 # Incrementăm contorul
}
}
# Calculul mediei aritmetice a distantelor
media <- suma / count
# Afisarea rezultatului
cat("Media aritmetica a distantelor:", media, "\n")
########################
> # Calculul mediei aritmetice a distantelor
> media <- suma / count
> # Afisarea rezultatului
> cat("Media aritmetica a distantelor:", media, "\n")
Media aritmetica a distantelor: 0.6663595
> ########################

O a doua metodă de simulare a unui punct (X,Y) repartizat uniform pe D(1) consta in folosirea schimbarii de variabila in coordonate polare: X = Rcos(Θ) si Y = Rsin(Θ). 4. Plecând de la densitatea cuplului (X,Y), găsiți densitatea v.a. R si Θ.

Pentru a găsi denistatea cuplului (X, Y) , exprimată în funcție de R și Θ, vom folosi valoarea Jacobianului, ce a fost calculată la punctul anterior (Problema 1, cerința 3).

Astfel , Δ = R.

$$f_{R,\Theta}(r,\theta) = f_{X,Y}(x,y)\,\Delta = \frac{1}{\pi}\,1_{D(1)} \cdot r = \frac{r}{\pi}\,1_{D(1)}$$

Mai departe, se vor calcula densitățile variabilelor R și Θ:

$$f_\Theta(\theta) = \int_0^1 f(r,\theta)dr = \int_0^1 \frac{r}{\pi}\,dr = \frac{1}{\pi} * \frac{r^2}{2}\,\Big|_0^1 = \frac{1}{\pi} * \frac{1}{2} = \frac{1}{2\pi}\,, \forall\,\theta \in [0,2\pi]$$

$$f_R(r) = \int_0^{2\pi} f(r,\theta)d\theta = \int_0^{2\pi} \frac{r}{\pi}\,d\theta = \frac{r}{\pi} * \theta\,\Big|_0^{2\pi} = \frac{r}{\pi} * 2\pi = 2r\,, \forall\, r \in [0,1]$$

Astfel, densitatea variabilelor aleatoare R si θ este:

$$f_\Theta(\theta) = \begin{cases} \frac{1}{2\pi}, \forall\,\theta \in [0,2\pi] \\ 0, altfel \end{cases}$$

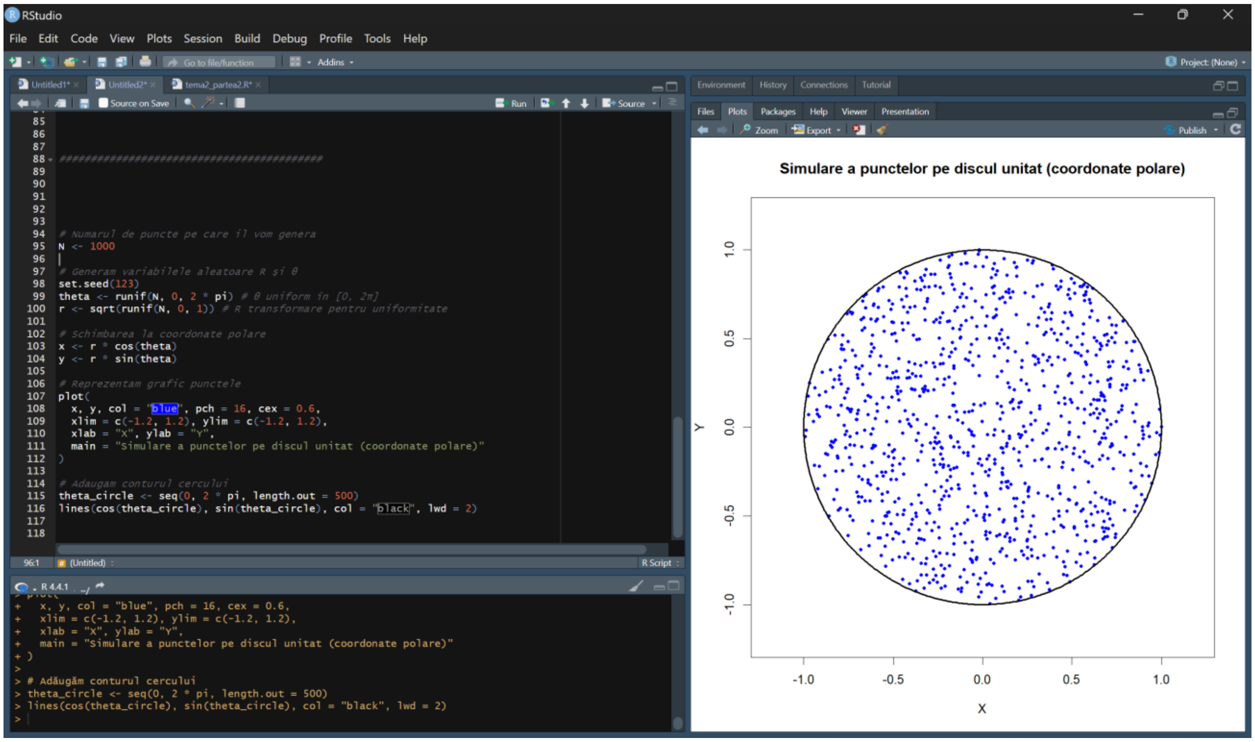
$$f_R(r) = \begin{cases} 2r\,, \forall\, r \in [0,1] \\ 0, altfel \end{cases}$$

5. Simulați N = 1000 de puncte prin această metodă si ilustrati grafic aceste puncte (inclusiv conturul cercului).

```
# Numarul de puncte pe care il vom genera
N <- 1000

# Generam variabilele aleatoare R și θ
set.seed(123)
theta <- runif(N, 0, 2 * pi) # θ uniform în [0, 2π]
r <- sqrt(runif(N, 0, 1)) # R transformare pentru uniformitate

# Schimbarea la coordonate polare
x <- r * cos(theta)
y <- r * sin(theta)

# Reprezentam grafic punctele
plot(
  x, y, col = "blue", pch = 16, cex = 0.6,
  xlim = c(-1.2, 1.2), ylim = c(-1.2, 1.2),
  xlab = "X", ylab = "Y",
  main = "Simulare a punctelor pe discul unitat (coordonate polare)"
)

# Adaugam conturul cercului
theta_circle <- seq(0, 2 * pi, length.out = 500)
lines(cos(theta_circle), sin(theta_circle), col = "black", lwd = 2)
```

Explicație: Codul simulează N=1000 puncte uniform distribuite pe discul unitate folosind coordonate polare, unde poziția fiecărui punct este definită printr-un unghi θ și o rază R. Unghiul θ este generat uniform în intervalul $[0,2\pi]$ pentru a asigura o distribuție egală pe circumferința discului, iar raza R este obținută prin transformarea $\sqrt{U}$ unde U este generat uniform în [0,1] astfel încât densitatea punctelor să fie uniformă pe suprafața discului. Punctele sunt apoi convertite în coordonate carteziene conform relațiilor date. Grafic, punctele sunt reprezentate în plan, folosind un plot, iar conturul discului este ilustrat desenând un cerc unitate utilizând puncte calculate de-a lungul marginii acestuia.

Ilustrare:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Untitled2*
tema2_partea2.R*
Source on Save
Run
Source
###
# Numarul de puncte pe care il vom genera
N <- 1000
# Generam variabilele aleatoare R și θ
set.seed(123)
theta <- runif(N, 0, 2 * pi) # θ uniform in [0, 2π]
r <- sqrt(runif(N, 0, 1)) # R transformare pentru uniformitate
# Schimbarea la coordonate polare
x <- r * cos(theta)
y <- r * sin(theta)
# Reprezentam grafic punctele
plot(
x, y, col = "blue", pch = 16, cex = 0.6,
xlim = c(-1.2, 1.2), ylim = c(-1.2, 1.2),
xlab = "X", ylab = "Y",
main = "Simulare a punctelor pe discul unitat (coordonate polare)"
)
# Adaugam conturul cercului
theta_circle <- seq(0, 2 * pi, length.out = 500)
lines(cos(theta_circle), sin(theta_circle), col = "black", lwd = 2)
96:1
(Untitled)
R Script
R 4.4.1
+ x, y, col = "blue", pch = 16, cex = 0.6,
+ xlim = c(-1.2, 1.2), ylim = c(-1.2, 1.2),
+ xlab = "X", ylab = "Y",
+ main = "Simulare a punctelor pe discul unitat (coordonate polare)"
+)
>
> # Adăugăm conturul cercului
> theta_circle <- seq(0, 2 * pi, length.out = 500)
> lines(cos(theta_circle), sin(theta_circle), col = "black", lwd = 2)
>
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation
Zoom
Export
Publish
Simulare a punctelor pe discul unitat (coordonate polare)
Y
-1.0
-0.5
0.0
0.5
1.0
X

Problema 2

Construiți o aplicație Shiny în care să reprezetați grafic funcțiile de repartiție pentru următoarele variabile aleatoare:

1. $X, 3+2X, X^2, \sum_{i=1}^{n} X_i^2$, unde $X_1, X_2, \dots, X_n \sim i.i.d.\ N(0,1), iar\ n\ este\ fixat, n \in \mathbb{N}$.
2. $X, 3+2X, X^2, \sum_{i=1}^{n} X_i^2$, unde $X_1, X_2, \dots, X_n \sim i.i.d.\ N(\mu, \sigma^2), cu\ \mu \in \mathbb{R}, \sigma > 0,$ $iar\ n\ este\ fixat, n \in \mathbb{N}$.
3. $X, 2-5X, X^2, \sum_{i=1}^{n} X_i$, unde $X_1, X_2, \dots, X_n \sim i.i.d.\ Exp(\lambda), cu\ \lambda > 0,$ $iar\ n\ este\ fixat, n \in \mathbb{N}$.
4. $X, 3X+2, X^2, \sum_{i=1}^{n} X_i$, unde $X_1, X_2, \dots, X_n \sim i.i.d.\ Pois(\lambda), cu\ \lambda > 0,$ $iar\ n\ este\ fixat, n \in \mathbb{N}$.
5. $X, 5X+4, X^3, \sum_{i=1}^{n} X_i$, unde $X_1, X_2, \dots, X_n \sim i.i.d.\ Binom(r, p), cu\ r \in \mathbb{N},\ p \in (0,1),$ $iar\ n\ este\ fixat, n \in \mathbb{N}$.

Aplicațiile Shiny

Așa cum este prezentat și pe prima pagină a site-ului oficial, „*Shiny* is package that makes it easy to build interactive web apps straight from R & Python” (i.e. „*Shiny* este un pachet care facilitează construirea unei aplicații web interactive direct din R & Python”).

În R, *Shiny* reprezintă un pachet care permite implementarea dezvoltarea de aplicații web interactive. Acesta este folosit cu precădere pentru vizualizarea de date, în special când vine vorba despre statistică, având un format accesibil. *Shiny* nu necesită cunoștințe de dezvoltare web (HTML, CSS, JavaScript, framework-uri sau arhitecturi).

Structura unei aplicații *Shiny* constă în 2 elemente (interfața și serverul) și un pseudo-element (reactivitatea):

- Interfața utilizatorului se referă strict la aspectul aplicației, *frontend* (UI – User Interface) și elemente de control/vizuale: slidere, butoane, grafice, tabele.

- Serverul se ocupă de partea logică, *backend*, și gestionează acțiunile utilizatorului. Pentru o aplicație *Shiny*, nu este necesar să se realizeze în mod explicit conexiunea la un server, bază de date, rularea pe un port, și alte astfel de aspecte aplicate în industria de dezvoltare web. „*Shiny* makes it easy” (i.e. „*Shiny* face să fie ușor”), deci, toate aceste aspecte sunt preluate de pachet în sine. (Mențiune: dacă se dorește, acestea pot fi suprascrise, dar această suprascriere nu este obligatorie)

- Reactivitatea este considerat un pseudo-element întrucât nu este de sine stătătoare. Aceasta face legătura între UI și server, astfel că preia informația de la interfață la server și invers, actualizând constant și dinamic întreaga aplicație. Putem considera că aceasta funcționează, ca principii de dezvoltare web, pe baza conceptelor de formulare (completare de date) și prelucrare de evenimente (schimbarea interfeței la selectarea unor elemente, apăsarea de butoane). După ce au fost preluate din *frontend*, informațiile sunt duse în backend și prelucrate sincron (în aceeași sesiune, iar utilizatorul vede modificările fără a deschide într-o altă sesiune aplicația web). Acest mod de prelucrare aduce atât beneficii (aplicația este una dinamică, se observă modificările realizate asupra interfeței, și implicit prelucrarea datelor), cât și dezavantaje (întrucât se așteaptă prelucrarea datelor, pot apărea situații în care aceasta durează mult, iar utilizatorul trebuie să aștepte pentru a vedea actualizările apărute în urma modificărilor).

Cele două elemente ale aplicației sunt stocate în două variabile omonime numelor: ui, server. În final, se realizează un apel către funcția shinyApp cu cei 2 parametri, urmând să fie lansată aplicația

Instalarea pachetului *Shiny* se realizează prin comanda: install.packages(”shiny”)

Site-ul oficial al pachetului oferă îndrumare pentru înțelegerea conceptelor de funcționare, oferind exemple și lecții/exerciții. Cu toate acestea, în rezolvarea problemei am plecat de la exemplul lor de bază, după care am citit despre funcții prin intermediul documentației, nu prin parcurgerea lecțiilor de pe site.

Exemplul de bază

* extras din site-ul oficial *Shiny*

```r
library(shiny)

library(bslib)

# Define UI for app that draws a histogram ----
ui <- page_sidebar(
  # App title ----
  title = "Hello Shiny!",
  # Sidebar panel for inputs ----
  sidebar = sidebar(
    # Input: Slider for the number of bins ----
    sliderInput(
      inputId = "bins",
      label = "Number of bins:",
      min = 1,
      max = 50,
      value = 30
    )
  ),
  # Output: Histogram ----
  plotOutput(outputId = "distPlot")
)
```

```
# Define server logic required to draw a histogram ----
server <- function(input, output) {

  # Histogram of the Old Faithful Geyser Data ----
  # with requested number of bins
  # This expression that generates a histogram is wrapped in a call
  # to renderPlot to indicate that:
  #
  # 1. It is "reactive" and therefore should be automatically
  #    re-executed when inputs (input$bins) change
  # 2. Its output type is a plot
  output$distPlot <- renderPlot({

    x    <- faithful$waiting
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    hist(x, breaks = bins, col = "#007bc2", border = "white",
         xlab = "Waiting time to next eruption (in mins)",
         main = "Histogram of waiting times")

    })

}

shinyApp(ui = ui, server = server)
```

Cod Sursă

```
library(shiny)
library(bslib)

ui <- fluidPage(
  titlePanel("Reprezentarea grafica a functiilor de repartitie"),

  sidebarLayout(
    sidebarPanel(
      selectInput(
        inputId = "tipVA",
        label = "Tipul variabilei aleatoare",
        choices = list(
          "Normala" = "normala",
          "Normala parametrizata" = "normalaParametrizata",
          "Exponentiala" = "exponentiala",
          "Poisson" = "poisson",
          "Binomiala" = "binomiala"
        )
      ),
      numericInput(
```

```
      inputId = "n",
      label = "Numarul n:",
      value = 5,
      min = 1
    ),
    conditionalPanel(
      condition = "input.tipVA == 'normalaParametrizata'",
      numericInput(
        inputId = "miu",
        label = "Media:",
        value = 0
      ),
      numericInput(
        inputId = "sigma",
        label = "Deviatia standard",
        value = 1,
        min = 0.00001
      )
    ),
    conditionalPanel(
      condition = "input.tipVA == 'poisson' || input.tipVA == 'exponentiala'",
      numericInput(
        inputId = "lambda",
        label = "Lambda:",
        value = 1,
        min = 0.00001
      )
    ),
    conditionalPanel(
      condition = "input.tipVA == 'binomiala'",
      numericInput(
        inputId = "r",
        label = "Numarul de repetari:",
        value = 5,
        min = 1
      ),
      numericInput(
        inputId = "p",
        label = "Probabilitatea realizarii evenimentului:",
        value = 0.5,
        min = 0,
        max = 1
      )
    )
  ),
```

```r
    mainPanel(
      plotOutput("repartitie")
    )
  )
)

server <- function(input, output) {

  output$repartitie <- renderPlot({
    n <- input$n
    tipVariabila <- input$tipVA

    if (tipVariabila == "binomiala") {
      r <- input$r
      p <- input$p
      x1 <- rbinom(n = n, size = r, prob = p)
      x2 <- 5 * x1 + 4
      x3 <- x1^3
      x4 <- cumsum(x1)

      par(mfrow = c(2, 2))

      plot(
        ecdf(x1),
        main = "Functia de repartitie pentru X",
        xlab = "X",
        ylab = "F(X)",
        col = "red"
      )
      plot(
        ecdf(x2),
        main = "Functia de repartitie pentru 5X+4",
        xlab = "X",
        ylab = "F(X)",
        col = "purple"
      )
      plot(
        ecdf(x3),
        main = "Functia de repartitie pentru X^3",
        xlab = "X",
        ylab = "F(X)",
        col = "violet"
      )
      plot(
```

```
    ecdf(x4),
    main = "Functia de repartitie pentru Σ(X)",
    xlab = "X",
    ylab = "F(X)",
    col = "blue"
  )

} else if (tipVariabila == "poisson") {
  lambda <- input$lambda
  x1 <- rpois(n = n, lambda = lambda)
  x2 <- 3 * x1 + 2
  x3 <- x1^2
  x4 <- cumsum(x1)

  par(mfrow = c(2, 2))

  plot(
    ecdf(x1),
    main = "Functia de repartitie pentru X",
    xlab = "X",
    ylab = "F(X)",
    col = "red"
  )
  plot(
    ecdf(x2),
    main = "Functia de repartitie pentru 3X+2",
    xlab = "X",
    ylab = "F(X)",
    col = "purple"
  )
  plot(
    ecdf(x3),
    main = "Functia de repartitie pentru X^2",
    xlab = "X",
    ylab = "F(X)",
    col = "violet"
  )
  plot(
    ecdf(x4),
    main = "Functia de repartitie pentru Σ(X)",
    xlab = "X",
    ylab = "F(X)",
    col = "blue"
  )
```

```r
  } else if (tipVariabila == "exponentiala") {
    lambda <- input$lambda
    x1 <- rexp(n = n, rate = lambda)
    x2 <- 2 - 5 * x1
    x3 <- x1^2
    x4 <- cumsum(x1)

    par(mfrow = c(2, 2))

    plot(
      ecdf(x1),
      main = "Functia de repartitie pentru X",
      xlab = "X",
      ylab = "F(X)",
      col = "red"
    )
    plot(
      ecdf(x2),
      main = "Functia de repartitie pentru 2-5X",
      xlab = "X",
      ylab = "F(X)",
      col = "purple"
    )
    plot(
      ecdf(x3),
      main = "Functia de repartitie pentru X^2",
      xlab = "X",
      ylab = "F(X)",
      col = "violet"
    )
    plot(
      ecdf(x4),
      main = "Functia de repartitie pentru Σ(X)",
      xlab = "X",
      ylab = "F(X)",
      col = "blue"
    )

  } else if (tipVariabila == "normala") {
    x1 <- rnorm(n = n, mean = 0, sd = 1)
    x2 <- 3 + 2 * x1
    x3 <- x1^2
    x4 <- cumsum(x1^2)

    par(mfrow = c(2, 2))
```

```
    plot(
      ecdf(x1),
      main = "Functia de repartitie pentru X",
      xlab = "X",
      ylab = "F(X)",
      col = "red"
    )
    plot(
      ecdf(x2),
      main = "Functia de repartitie pentru 3+2X",
      xlab = "X",
      ylab = "F(X)",
      col = "purple"
    )
    plot(
      ecdf(x3),
      main = "Functia de repartitie pentru X^2",
      xlab = "X",
      ylab = "F(X)",
      col = "violet"
    )
    plot(
      ecdf(x4),
      main = "Functia de repartitie pentru Σ(X^2)",
      xlab = "X",
      ylab = "F(X)",
      col = "blue"
    )

  } else if (tipVariabila == "normalaParametrizata") {
    media <- input$miu
    sigma <- input$sigma
    x1 <- rnorm(n = n, mean = media, sd = sigma)
    x2 <- 3 + 2 * x1
    x3 <- x1^2
    x4 <- cumsum(x1^2)

    par(mfrow = c(2, 2))

    plot(
      ecdf(x1),
      main = "Functia de repartitie pentru X",
      xlab = "X",
      ylab = "F(X)",
```

```
        col = "red"
      )
      plot(
        ecdf(x2),
        main = "Functia de repartitie pentru 3+2X",
        xlab = "X",
        ylab = "F(X)",
        col = "purple"
      )
      plot(
        ecdf(x3),
        main = "Functia de repartitie pentru X^2 sau X^3",
        xlab = "X",
        ylab = "F(X)",
        col = "violet"
      )
      plot(
        ecdf(x4),
        main = "Functia de repartitie pentru Σ(X^2)",
        xlab = "X",
        ylab = "F(X)",
        col = "blue"
      )
    }
  })
}

shinyApp(ui = ui, server = server)
```

User Interface

Parametrul User Interface (ui) conține o pagină fluidă (fluidPage). La rândul ei, aceasta conține un panou de tip bară lateral (poziționat în stânga - sidebarPanel) și un panou principal (poziționat în stânga - mainPanel).

Interfața utilizatorului stochează un dicționar de tip cheie-valoare (unde cheia este setată în fiecare input prin parametrul inputId). Accesarea câmpurilor acestui dicționar se face astfel: în cadrul UI: input.nume_câmp; în cadrul serverului: input$nume_câmp.

Panoul lateral începe printr-un input de tip select, ce reprezintă tipul variabilei aleatoare pe care dorim să o modelăm, cu id-ul tipVA, lista de variante dintre care putem alege fiind: Normală (valoare: normala), Normală Parametrizată (valoare: normalaParametrizata), Exponențială (valoare: exponentiala), Poisson (valoare: poisson), Binomială (valoare: binomiala). În urma selectării variantei, la cheia „tipVA” se stochează una dintre valorile din lista anterioară.

Urmează un input numeric, unde se introduce numărul n, cu valoare standard 5 și valoare minimă 1. După introducere, la cheia „n” se stochează numărul introdus.

Următoarele elemente sunt panouri condiționate (i.e panouri care apar pe ecran doar dacă o anumită condiție este îndeplinită):

a) Repartizarea normală parametrizată.

Dacă în primul input a fost selectată repartizarea normală parametrizată, vor apărea două inputuri numerice: media (cu valoare standard 0 și inputId = „miu”) și deviația standard (valoare standard 1, inputId = „sigma”, minim = 0.00001). După introducere, la cheile „miu” și „sigma” se stochează media, respectiv deviația standard.

b) Repartizarea Poisson sau exponențială.

Atât repartizarea poisson, cât și cea exponențială, funcționează cu ajutorul parametrului lambda. Dacă în primul input a fost selectată repartizarea Poisson sau repartizarea exponențială, va apărea un input numeric: lambda (cu valoare standard 1, inputId = „lambda”, minim = 0.00001).

c) Repartizarea binomială.

Dacă în primul input a fost selectată repartizarea binomială, vor apărea două inputuri numerice: numărul de repetări (cu valoare standard 5, inputId = „r”, minim = 1) și probabilitatea realizării evenimentului (valoare standard 0.5, inputId = „p”, minim = 0, maxim = 1). După introducere, la cheile „r” și „p” se stochează numărul de repetări, respectiv probabilitatea realizării evenimentului.

Panoul principal rulează outputul intors prin cheia „repartitie” prin funcția plotOutput.

Server

Parametrul server conține o funcție cu 2 parametri (input și output). Prin input primește informația de la ui, iar prin output intoarce in ui un obiect. În cazul nostru, toată informația este stocată în câmpul „repartitie” al obiectului „output”.

Se începe prin extragerea valorii numărului n (care e introdus necondiționat de către utilizator; cheia „n”) și a tipului de variabilă aleatoare selectat (cheia „tipVA”) din obiectul „input”. Acestea sunt stocate in variabilele „n” și „tipVariabila”.

Se modelează apoi ploturile pentru variabilele aleatoare, în funcție de input. Ordinea acestora în cerință era: Normală, Normală parametrizată, Exponențială, Poisson, Binomială. În cod sunt acoperite în ordinea: Binomială, Poisson, Exponențială, Normală, Normală parametrizată.

a) Repartizarea binomială.

Pentru a ajunge în acest cadru, se verifică dacă tipul variabilei aleatoare „tipVariabila” este „binomiala” (valoare selectată în UI).

În variabilele „r” și „p” sunt stocate valorile din panoul condițional asociat, de la cheile „r” și „p”.

Se generează apoi X (notat x1), prin „rbinom(n = n, size = r, prob = p)”. Urmează apoi a fi generate variabilele aleatoare listate în funcție de aceasta: x2 <- 5*x1+4, x3 <- x1^3, x4 <- cumsum(x1).

Se setează dispunerea elementelor grafice în 2 linii și 2 coloane (pentru a afișa toate cele 4 grafice) prin „par(mfrow = c(2, 2))”.

Se realizează plot-area celor 4 variabile aleatoare cu ajutorul funcției plot, astfel: „plot (ecdf(variabila), main = „Titlul graficului”, xlab = „X”, ylab = „F(x)”, col= „nume_culoare”)”. Funcția ecdf (Emipirical Cumulative Distribution Function) returnează funcția de distribuție cumulativă empirică (empirică – obținută prin observați asupra datelor) pentru un set de date.

b) Repartizarea Poisson.

Pentru a ajunge în acest cadru, se verifică dacă tipul variabilei aleatoare „tipVariabila” este „poisson” (valoare selectată în UI).

În variabila „lambda” este stocată valoarea din panoul condițional asociat, de la cheia „lambda”.

Se generează apoi X (notat x1), prin „rpois(n = n, lambda = lambda)”. Urmează apoi a fi generate variabilele aleatoare listate în funcție de aceasta: x2 <- 3*x1+2, x3 <- x1^2, x4 <- cumsum(x1).

Se setează dispunerea elementelor grafice în 2 linii și 2 coloane (pentru a afișa toate cele 4 grafice) prin „par(mfrow = c(2, 2))”.

Se realizează plot-area celor 4 variabile aleatoare cu ajutorul funcției plot, astfel: „plot (ecdf(variabila), main = „Titlul graficului”, xlab = „X”, ylab = „F(x)”, col= „nume_culoare”)”. Funcția ecdf (Emipirical Cumulative Distribution Function) returnează funcția de distribuție cumulativă empirică (empirică – obținută prin observați asupra datelor) pentru un set de date.

c) Repartizarea Exponențială.

Pentru a ajunge în acest cadru, se verifică dacă tipul variabilei aleatoare „tipVariabila” este „exponentiala” (valoare selectată în UI).

În variabila „lambda” este stocată valoarea din panoul condițional asociat, de la cheia „lambda”.

Se generează apoi X (notat x1), prin „rexp(n = n, rate = lambda)”. Urmează apoi a fi generate variabilele aleatoare listate în funcție de aceasta: x2 <- 2-5*x1, x3 <- x1^2, x4 <- cumsum(x1).

Se setează dispunerea elementelor grafice în 2 linii și 2 coloane (pentru a afișa toate cele 4 grafice) prin „par(mfrow = c(2, 2))”.

Se realizează plot-area celor 4 variabile aleatoare cu ajutorul funcției plot, astfel: „plot (ecdf(variabila), main = „Titlul graficului”, xlab = „X”, ylab = „F(x)”, col= „nume_culoare”)”. Funcția ecdf (Emipirical Cumulative Distribution Function) returnează funcția de distribuție cumulativă empirică (empirică – obținută prin observați asupra datelor) pentru un set de date.

d) Repartizarea Normală.

Pentru a ajunge în acest cadru, se verifică dacă tipul variabilei aleatoare „tipVariabila” este „normala” (valoare selectată în UI).

Se generează X (notat x1), prin „rnorm(n = n, mean = 0, sd = 1)”. Urmează apoi a fi generate variabilele aleatoare listate în funcție de aceasta: x2 <- 3+2*x1, x3 <- x1^2, x4 <- cumsum(x1^2).

Se setează dispunerea elementelor grafice în 2 linii și 2 coloane (pentru a afișa toate cele 4 grafice) prin „par(mfrow = c(2, 2))”.

Se realizează plot-area celor 4 variabile aleatoare cu ajutorul funcției plot, astfel: „plot (ecdf(variabila), main = „Titlul graficului”, xlab = „X”, ylab = „F(x)”, col= „nume_culoare”)”. Funcția ecdf (Emipirical Cumulative Distribution Function) returnează funcția de distribuție cumulativă empirică (empirică – obținută prin observați asupra datelor) pentru un set de date.

e) Repartizarea Normală Parametrizată.

Pentru a ajunge în acest cadru, se verifică dacă tipul variabilei aleatoare „tipVariabila” este „normalaParametrizata” (valoare selectată în UI).

În variabilele „media” și „sigma” sunt stocate valorile din panoul condițional asociat, de la cheile „miu” și „sigma”.

Se generează X (notat x1), prin „rnorm(n = n, mean = media, sd = sigma)”. Urmează apoi a fi generate variabilele aleatoare listate în funcție de aceasta: x2 <- 3+2*x1, x3 <- x1^2, x4 <- cumsum(x1^2).

Se setează dispunerea elementelor grafice în 2 linii și 2 coloane (pentru a afișa toate cele 4 grafice) prin „par(mfrow = c(2, 2))”.

Se realizează plot-area celor 4 variabile aleatoare cu ajutorul funcției plot, astfel: „plot (ecdf(variabila), main = „Titlul graficului”, xlab = „X”, ylab = „F(x)”, col= „nume_culoare”)”. Funcția ecdf (Emipirical Cumulative Distribution Function) returnează funcția de distribuție cumulativă empirică (empirică – obținută prin observați asupra datelor) pentru un set de date.

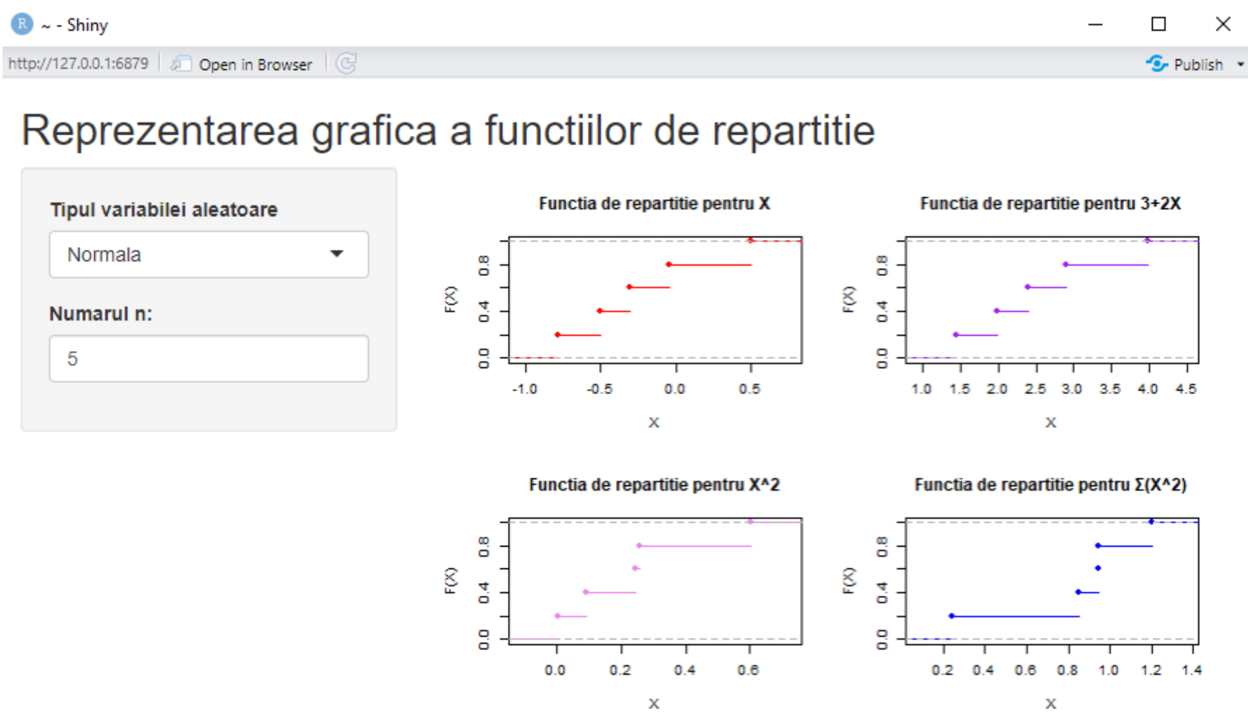
Apel final

La finalul aplicației se apelează funcția shinyApp cu parametrii ui și server, anterior construiți: „shinyApp(ui = ui, server = server)”.

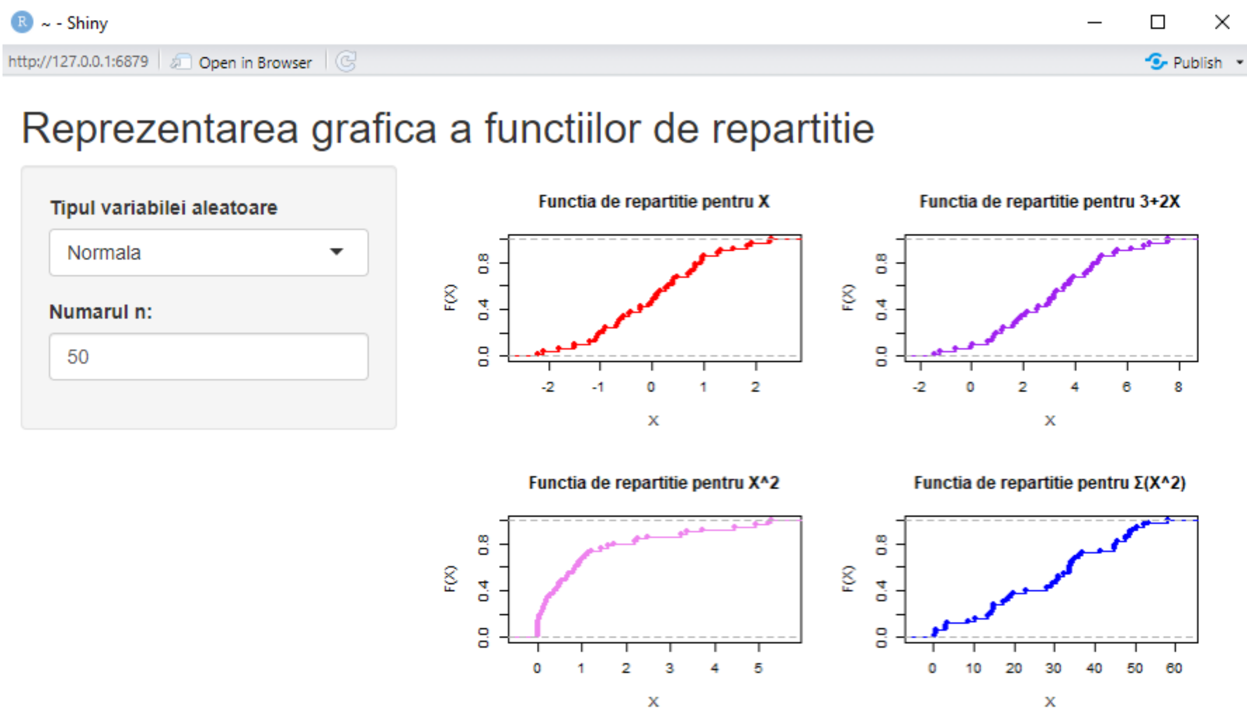
Modul de funcționare al aplicației

1. Repartiția Normală

a) Inițial;

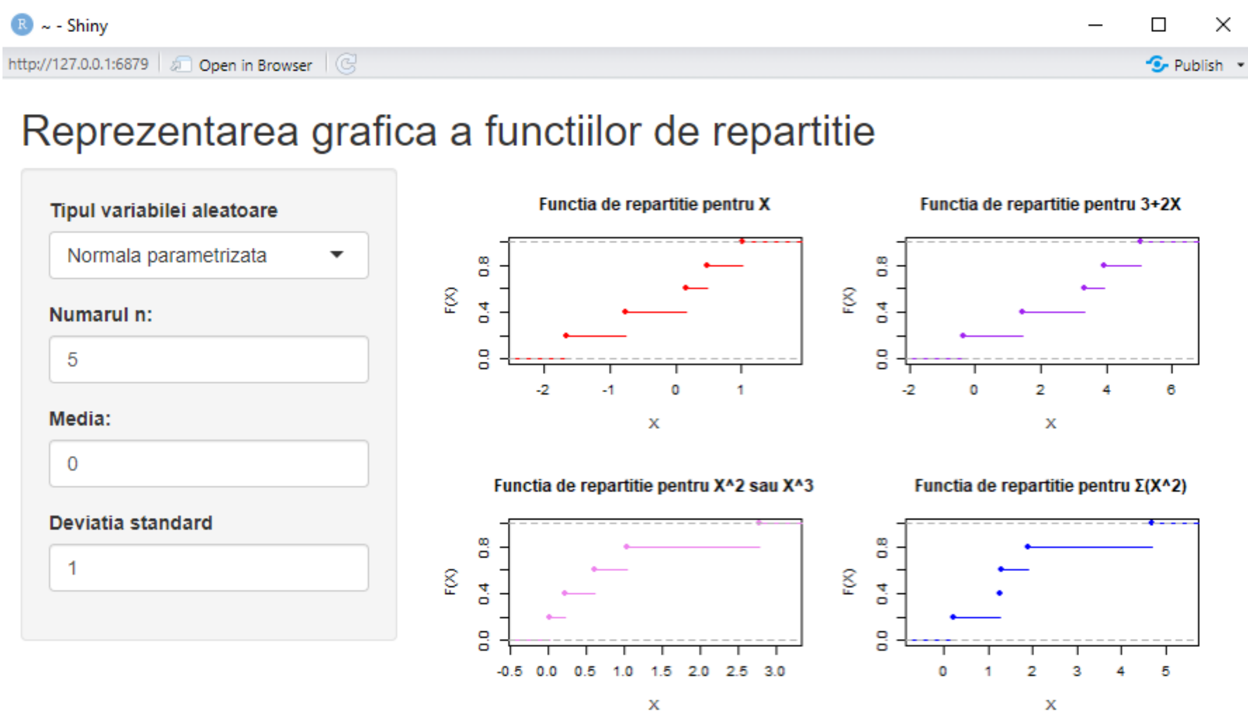


b) După modificarea parametrului n.

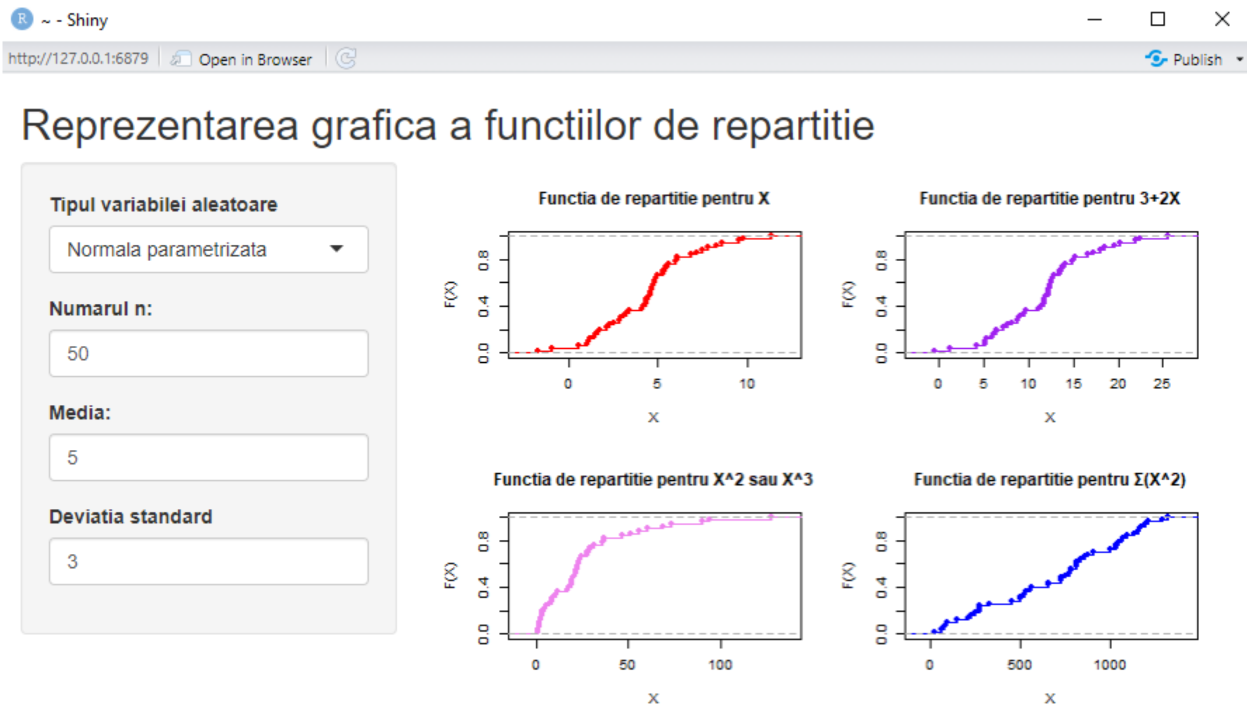


2. Repartiția Normală Parametrizată

a) Inițial;

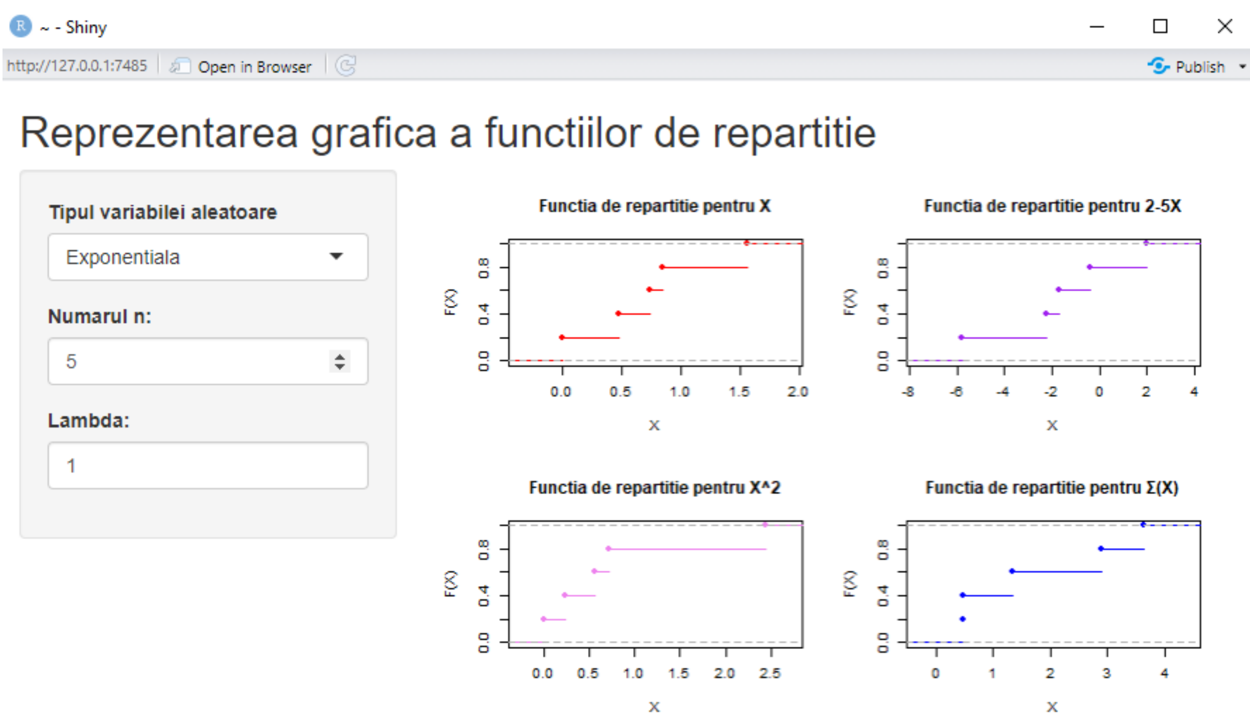


b) După modificarea parametrilor.

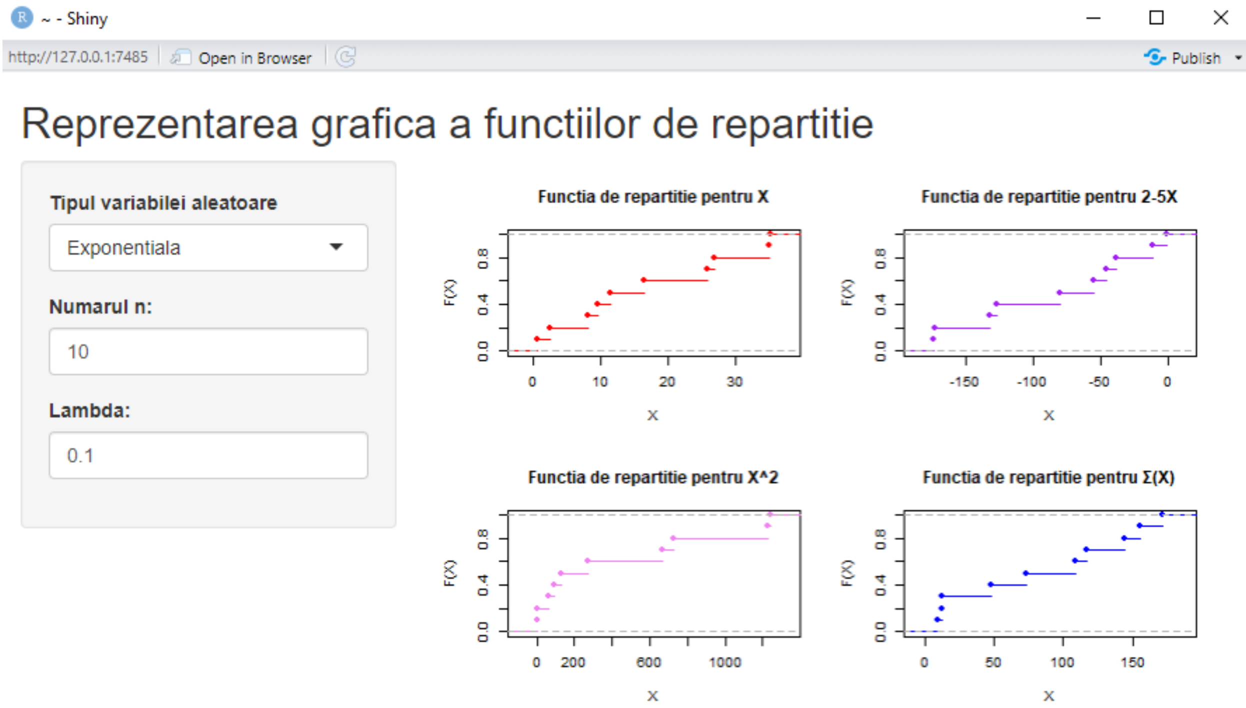


3. Exponențială

a) Inițial;

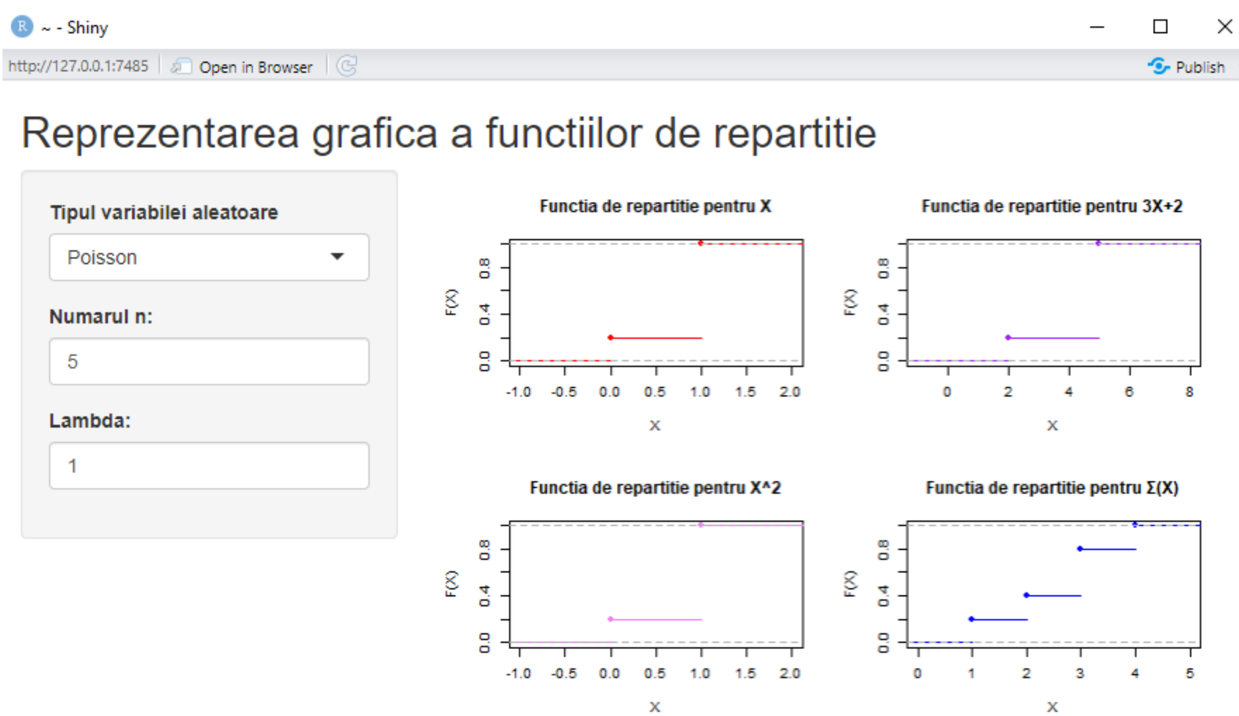


b) După modificarea parametrilor.

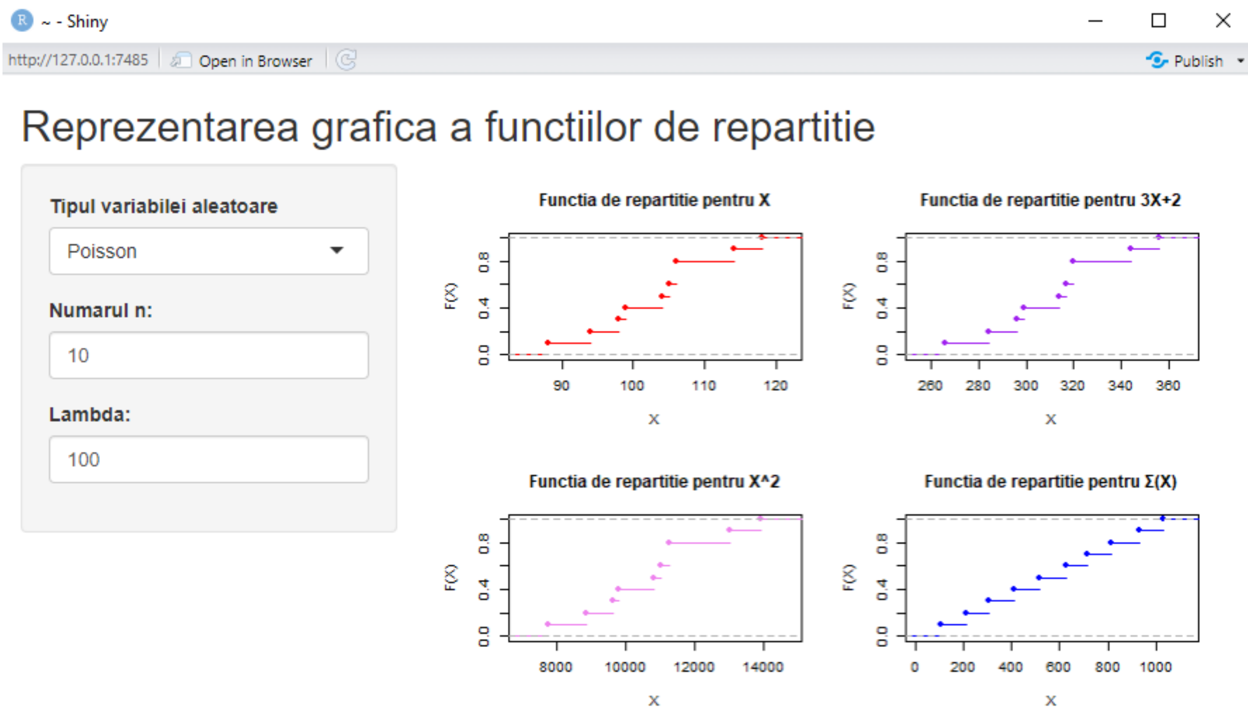


4. Poisson

a) Inițial;

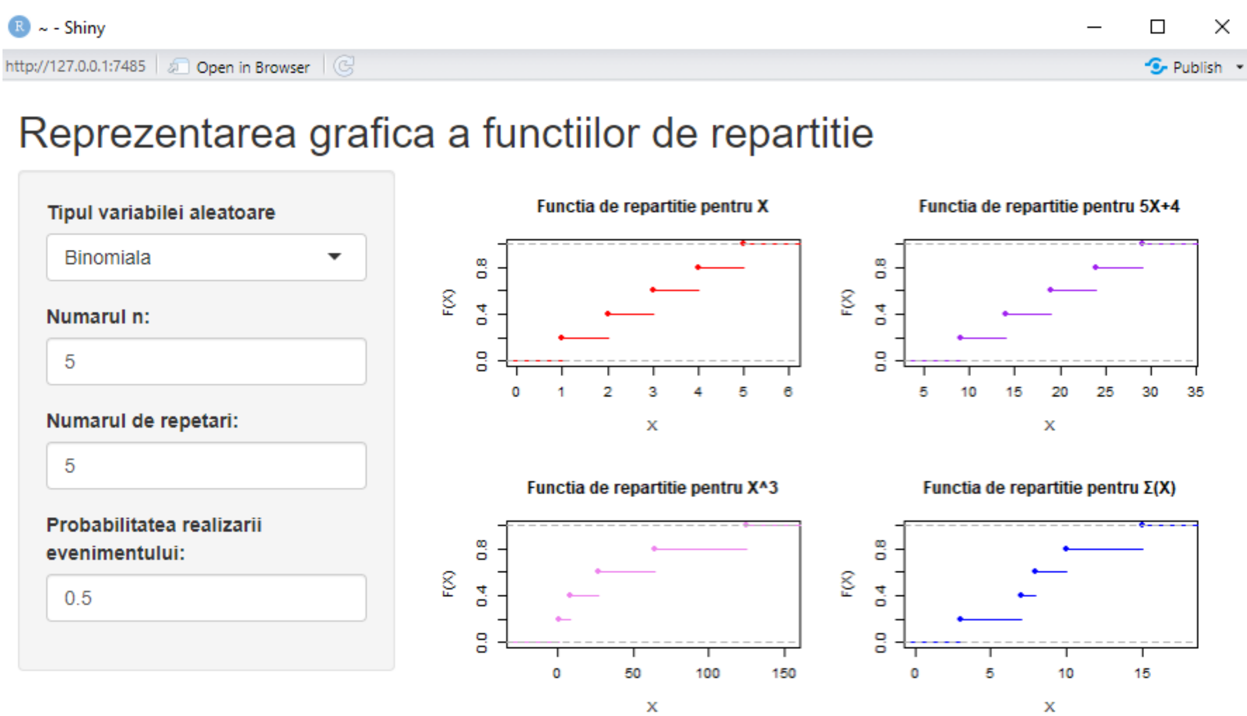


b) După modificarea parametrilor.

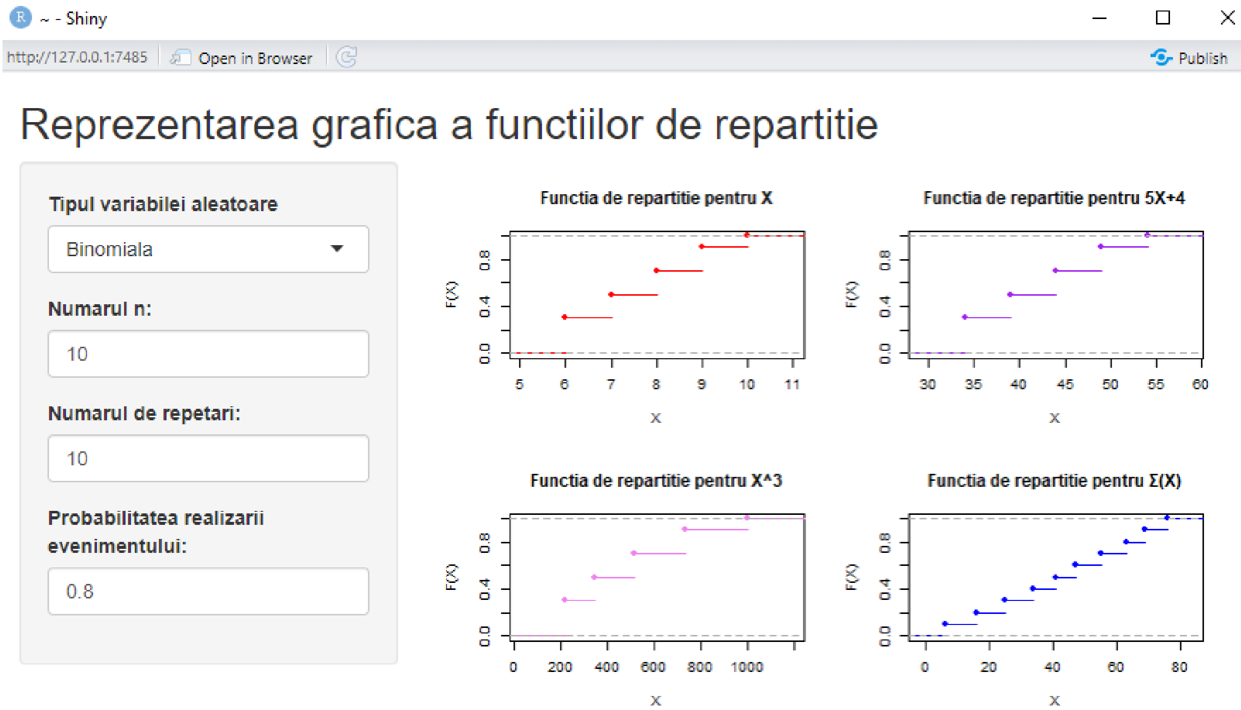


5. Binomială

a) Inițial;

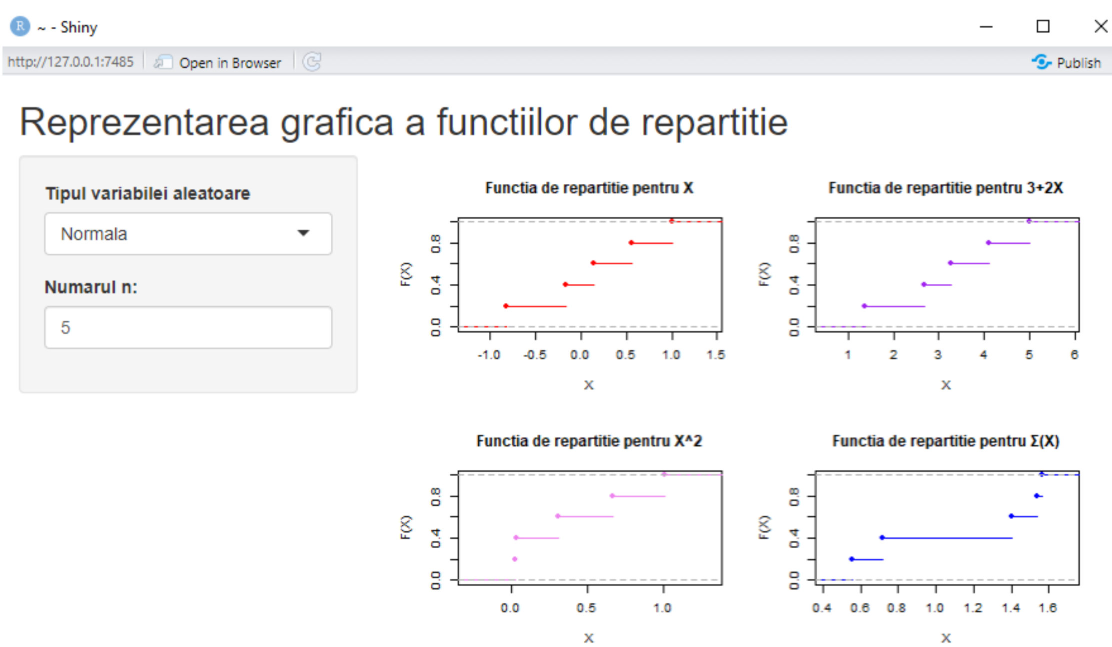


b) După modificarea parametrilor.

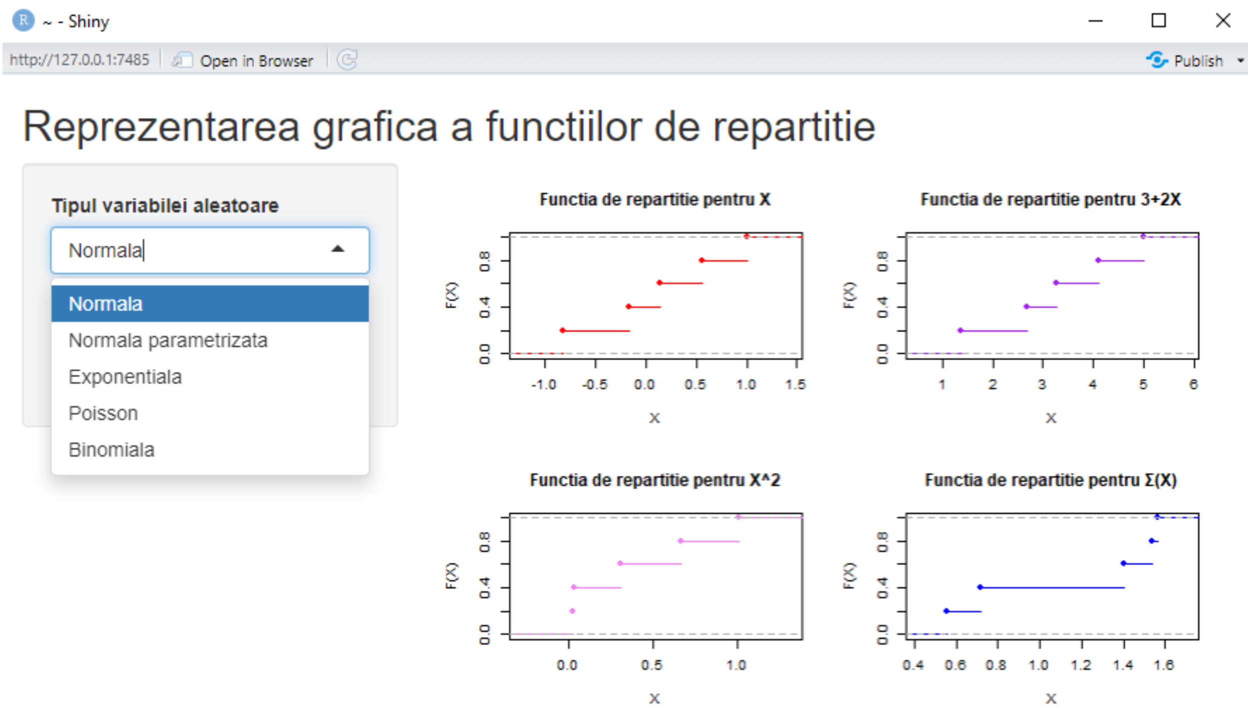


Alegerea tipului de variabilă aleatoare în meniu.

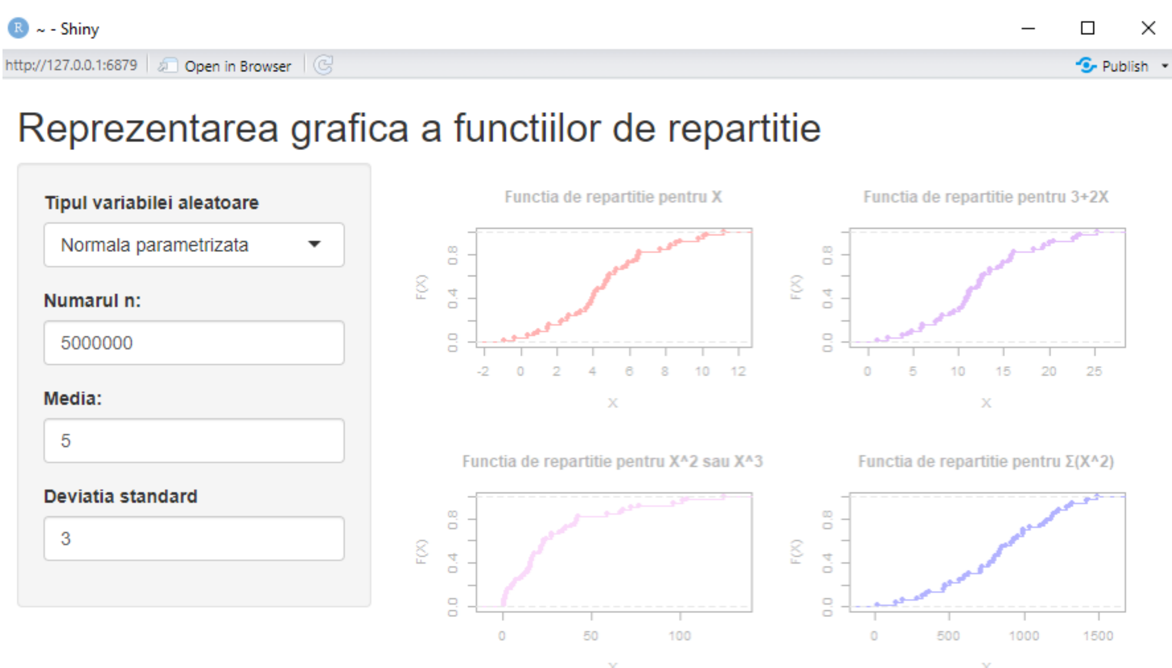
Inițial, aplicatia se deschide cu acest meniu:



Selectarea tipului de distribuție al variabilei aleatoare:



Alegerea tipului de distribuție determină modificarea panoului din stânga, după cum s-a putut observa în punctele 1-5 din rularea aplicației. După cum am amintit inițial, aplicatiile Shiny sunt reactive și dinamice – modificări e ecranul utilizatorului. Prelucrarea sincronă a datelor și afișarea datelor cu întârziere când seturi de date mari determină rularea acestora cu delay, care e sugerat utilizatorului prin scăderea opacității elementelor ce sunt în proces de modificare.



Problema 3

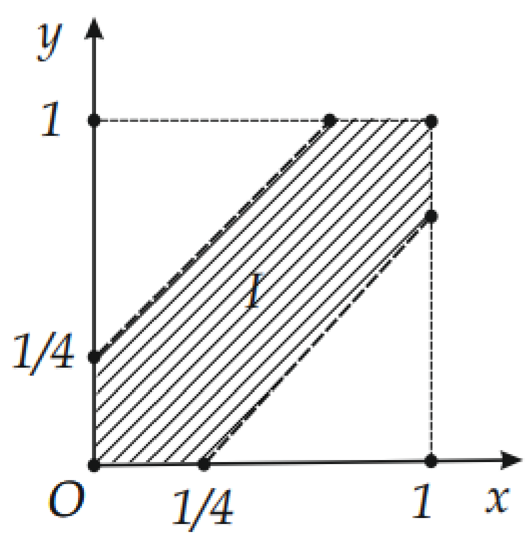
Pentru următoarele subpuncte rezolvările trebuie să fie abordate din două pespective, adică atât dpdv teoretic, cât și prin intermediul simulărilor. Have fun!

a) Pe un plan sunt trasate liniile $y = n$ (cu $n = 0, \pm1, \pm2, \ldots$) și un ac de lungime 1 este aruncat aleator pe acest plan. Arătați că probabilitatea ca acul să intersecteze vreo linie este egală cu $2/\pi$.

Planurile și liniile: Pe un plan, sunt trasate liniile orizontale de forma $y=n$,unde n este orice număr întreg ($n = 0, \pm1, \pm2, \ldots$). Astfel, distanța dintre două linii consecutive este de o unitate.

Aruncarea acului: Un ac de lungime $L=1$ este aruncat aleator pe acest plan.

Obiectiv: Determinarea probabilității ca acul să intersecteze cel puțin una dintre liniile trasate.



Presupunem o suprafață plană, pe care vom trasa linii orizontale, paralele, la distanță unitară. Dacă aruncăm un ac de lungime 1, aleator, pe această suprafață, putem observa că acul intersectează cel puțin una din linii.

Alegerea variabilelor aleatoare:

Pentru a descrie poziția acului, avem nevoie de două informații:

i) Distanța de la mijlocul acului la cea mai apropiată linie

Notăm cu X această distanță. Deoarece liniile sunt la distanță 1 una de alta (distanța dintre y=n și y=n+1 este 1), se poate arăta că X poate fi considerat *uniform* în intervalul [0,1/2].

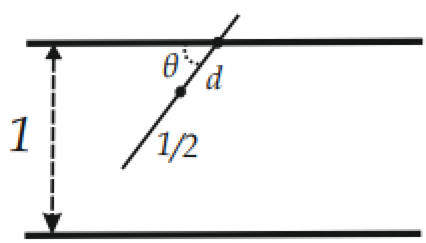
Intuitiv, dacă plasăm mijlocul acului oriunde între două linii consecutive (la distanță 1), „modul” față de cele două linii extreme este echidistribuit pe [0,1]. Pentru a simplifica, vom lua doar distanța până la linia cea mai apropiată, care va fi în [0,1/2].

ii) Unghiul acului cu orizontala

Notăm cu θ unghiul dintre ac și direcția orizontală (liniile y=constanta sunt orizontale). Se presupune că acul este aruncat la întâmplare, deci θ este uniform în [0,π].

De obicei, se ia unghiul în [0,2π), dar acul este un segment neorientat (capetele sunt indistincte), astfel încât unghiurile θ și θ+π reprezintă aceeași poziție relativă. Prin urmare, e suficient să considerăm θ∈[0,π].

Așadar, toate pozițiile posibile ale acului pot fi reprezentate prin perechile (X,θ) cu X∈[0,1/2], θ∈[0,π].



Se observă că $0 \le X \le 1/2$ și $0 \le \theta \le \pi/2$.

Condiția de intersecție:

Acul va intersecta o linie daca jumătate din lungimea sa intra în partea de deasupra unei linii și cealaltă jumătate în partea de dedesubtul aceleiași linii. Matematic, condiția pentru intersecție este:

$$\frac{X}{\sin\theta} \le \frac{L}{2}$$

Având în vedere că L=1, condiția devine:

$$X \leq \frac{1}{2} sin\theta$$

Acul intersectează cea mai apropiată linie dacă distanța de la centrul sau la linia intersectată este:

$$\frac{X}{sin\theta} \leq \frac{1}{2}$$

Vom presupune, în continuare, că atunci când aruncam acul, perechea (θ, d) este aleasă la întâmplare în dreptunghiul:

$$D = \{ (\theta, X) \in R\,\hat{}\,2 \mid 0 \leq \theta \leq \pi/ 2 ,\ \ 0 \leq X \leq 1 /2 \}$$

$$\text{aria(D)}=\frac{\pi}{4}$$

Domeniul în care acul intersectează o linie este submulțimea din D unde

$$0 \leq X \leq 1/2 \sin\theta.$$

Pentru o valoare fixă a lui θ, X poate varia între 0 și 1/2 sinθ. Așadar, „aria" domeniului favorabil se calculează prin:

$$\int_0^{\pi} \left(\int_0^{\frac{1}{2}\sin\theta} \mathrm{d}X \right) \mathrm{d}\theta \;=\; \int_0^{\pi} [\,X]_{X=0}^{X=\frac{1}{2}\sin\theta} \,\mathrm{d}\theta \;=\; \int_0^{\pi} \tfrac{1}{2}\sin\theta \,\mathrm{d}\theta.$$

Calculăm această integrală:

$$\int_0^{\pi} \tfrac{1}{2}\sin\theta \,\mathrm{d}\theta \;=\; \tfrac{1}{2}\,[-\cos\theta]_0^{\pi} \;=\; \tfrac{1}{2}\big((-\cos\pi) - (-\cos 0)\big) \;=\; \tfrac{1}{2}\big((-(-1)) - (-(1))\big) \;=\; \tfrac{1}{2}(1+1) \;=\; 1.$$

Prin urmare, aria domeniului favorabil este 1.

Probabilitatea se obține ca raport dintre „aria favorabilă" și „aria totală", adică aria dreptunghiului D si submulțimea din D:

P = aria favorabilă/aria totală = 2/ π

Probabilitatea producerii evenimentului E, adică intersectarea acului cu o linie este

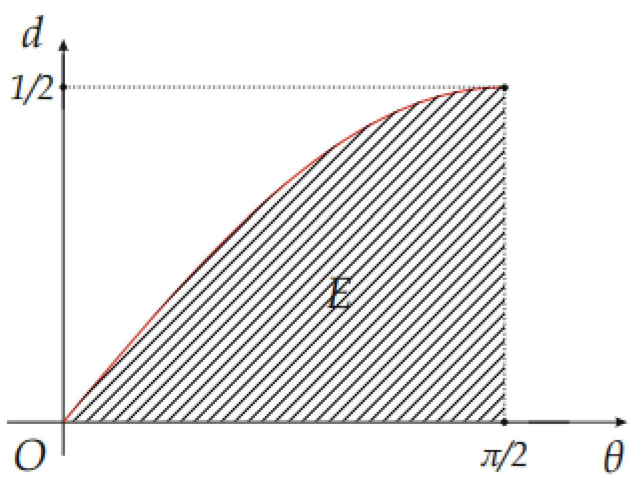
$$P(E) = \text{aria}(E) / \text{aria}(D),$$

unde E este domeniul $E = \{ (\theta, d) \in D \mid d \leq 1/2 \ \sin \theta \}$,si reprezintă o fracțiune din suprafața dreptunghiului, în interiorul căruia putem delimita suprafața E.

Avem:

$$\text{aria}(E) = \int_0^{\frac{\pi}{2}} \frac{1}{2} \sin \theta \, d\theta = \frac{1}{2}$$

$$\text{aria}(D) = \frac{\pi}{4}$$



Așadar, probabilitatea ca acul să întâlnească o linie este $P(E) = \frac{2}{\pi}$.

Simulare în R:

```r
# Exercitiul 3 subpunctul a

# Setam numarul de simulari
N <- 1e6

# 1. Generam distanta X la cea mai apropiata linie, uniform in [0, 0.5]
X <- runif(N, min = 0, max = 0.5)

# 2. Generam unghiul theta, uniform in [0, π]
theta <- runif(N, min = 0, max = pi)

# 3. Verificam conditia de intersectie: X <= 0.5 * sin(theta)
intersecteaza <- (X <= 0.5 * sin(theta))

# 4. Probabilitatea estimata (raportul dintre numarul de intersectii si numarul
total de aruncari)
prob_estimata <- mean(intersecteaza)

# 5. Comparam cu valoarea teoretica 2 / π
prob_teoretica <- 2 / pi

cat("Probabilitatea estimata   =", prob_estimata, "\n")
cat("Probabilitatea teoretica =", prob_teoretica, "\n")
cat("Eroarea (absoluta)       =", abs(prob_estimata - prob_teoretica), "\n")
```

Rezultatul simulării:

```
> cat("Probabilitatea estimata   =", prob_estimata, "\n")
Probabilitatea estimata   = 0.637472
> cat("Probabilitatea teoretica =", prob_teoretica, "\n")
Probabilitatea teoretica = 0.6366198
> cat("Eroarea (absoluta)       =", abs(prob_estimata - prob_teoretica), "\n"
)
Eroarea (absoluta)       = 0.0008522276
```

Explicații:

1. Alegerea lui X
   Distanța de la mijlocul acului la cea mai apropiată linie variază între 0 și jumătatea distanței dintre linii (linia cea mai apropiată este fie y=n fie y=n+1. Deoarece distanța dintre linii este 1, jumătatea este 0.5.

2. Alegerea lui **θ**
   Acul se poate orienta cu orice unghi în [0,π] (e suficient 0 - π, pentru că θ\ și θ+π reprezintă aceeași direcție pentru un segment neorientat).
3. Condiția de intersecție
   Acul intersectează linia dacă proiecția pe verticală a „jumătății” acului este mai mare (sau egală) decât distanța X. Proiecția jumătății acului pe verticală este ½ *sin(θ).

$$X \leq \frac{1}{2}\sin(\theta).$$

4. Estimare și comparație
   Se numară câte ace respectă această condiție și se împarte la `N`. Rezultatul ar trebui să se apropie (pe masură ce `N` crește) de valoarea teoretică 2/π≈0.63662.

b) Planul este secționat ca mai sus, de liniile y = n (unde n = 0, ±1, ±2, . . .), iar pe plan aruncăm o cruce formată prin unirea mijloacelor a două ace perpendiculare, de lungime 1. Notăm cu Z numărul de intersecții ale crucii cu liniile de pe plan. Arătați că E[Z/2] = 2/π, iar Var(Z/2) = $\frac{3-\sqrt{2}}{\pi} - \frac{4}{\pi^2}$. Dacă ați avea de ales între a folosi acul sau crucea (deci unul din acești doi algoritmi aleatori) pentru a estima valoarea lui π, ce ați alege? De ce?

Pe un plan se află liniile orizontale y=n (unde n ∈ Z).

Aruncăm la întamplare: un ac de lungime 1 *sau* o cruce formată din două ace de lungime 1, perpendiculare, cu un cap comun la mijloc (imaginează două segmente de lungime 1, care se intersectează în mijloacele lor şi formează un unghi drept).

Pentru cruce, definim variabila aleatoare Z = numarul total de intersecții ale crucii cu liniile y=n.

Rezultatele cerute sunt:

E [Z/2]=2/π.

Var (Z/2)=$\frac{3-\sqrt{2}}{\pi} - \frac{4}{\pi^2}$

Reamintim acul lui Buffon

Pentru acul simplu de lungime 1, numărul de intersecţii cu liniile este fie 0, fie 1. Mai exact, dacă notăm cu I indicatorul „acul intersectează linia”, atunci

P(I=1) = 2/π, P(I=0)=1−2/ π.

De aici rezultă:

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

Numeric, Var(I)≈0.231…

Crucea

Datele problemei:

„Crucea” este formată din două ace de lungime 1 care se intersectează în mijloacele lor și sunt perpendiculare unul pe celălalt, este aruncată „la întâmplare” pe plan, analog problemei acului lui Buffon.

Fie:

- $X1$ = indicatorul că primul ac (din cruce) intersectează (cel puțin) o linie orizontală.
- $X2$ = indicatorul că al doilea ac (perpendicular pe primul) intersectează (cel puțin) o linie orizontală.

Atunci,

$$Z=X1+X2,$$

unde $X1,X2\in\{0,1\}$.

Problema cere $E[Z/2]$ și $Var(Z/2)$.

Pentru medie:

Din acul lui Buffon:

Pentru un singur ac (lungimea 1, distanța dintre linii = 1), probabilitatea de intersecție este $2/\pi$. Asta înseamnă $E[Xi]=2/\pi$, căci Xi este indicator de eveniment {ac intersecteaza}.

Prin liniaritatea mediei,

$$E[Z] = E[X1+X2] = E[X1]+E[X2].$$

Important: deși $X1$ și $X2$ nu sunt independente, suma lor are media $E[X1]+E[X2]$.

Pentru cruce (două ace de aceeași lungime, orientate perpendicular), orientarea și poziția sunt alese aleator uniform; fiecare ac are aceeași distribuție ca „acul lui Buffon” (cu lungime 1), doar că sunt corelate prin punctul comun de intersecție.

Concluzie:

$E[Z] = E[X1]+E[X2] = 2/\pi + 2/\pi = 4/\pi \Longrightarrow E[Z2]=2/\pi$

Varianța: Var(Z/2):

Var(Z/2)=1/4*Var(Z)=1/4*Var(X1+X2).

Var(X1+X2) = Var(X1) + Var(X2) + 2 Cov(X1,X2)

Varianțele Var(X1) și Var(X2):

Fiecare Xi este indicatorul unui eveniment cu probabilitate 2/π.

Pentru un indicator X cu P(X=1)=p avem Var(X)=p(1−p).

Așadar,

$$\mathrm{Var}(X_i) \;=\; \frac{2}{\pi}\Bigl(1-\frac{2}{\pi}\Bigr) \;=\; \frac{2}{\pi}-\frac{4}{\pi^2}.$$

$$\mathrm{Var}(X_1)+\mathrm{Var}(X_2) \;=\; 2\Bigl(\frac{2}{\pi}-\frac{4}{\pi^2}\Bigr) \;=\; \frac{4}{\pi} \;-\; \frac{8}{\pi^2}.$$

$$\mathrm{Cov}(X_1,X_2) \;=\; \mathbb{E}[X_1X_2] \;-\; \mathbb{E}[X_1]\,\mathbb{E}[X_2].$$

X1*X2=1 <=> ambele ace intersectează cel puțin o linie.

E[X1] E[X2]= 4/π^2

Calculul E[X1*X2]=P(ambele intersectează)

Pentru a intersecta o linie, pentru fiecare ac trebuie ca *distanța* de la centrul comun la linia cea mai apropiată să fie mai mică decât proiecția verticală a jumătății acului respectiv.

Notăm θ = unghiul (aleator uniform) al primului ac cu axa orizontală. Atunci al doilea ac are unghi θ+π/2.

Distanța până la linia orizontală (în mod 1) este X∈[0,1/2] (de asemenea uniform).

Pentru primul ac: intersecție ⟺ X≤1/2·|sin(θ)|

Pentru al doilea ac: intersecție $\iff X \le 1/2 \cdot |\cos(\theta)|$

Prin urmare, *ambele intersectează* $\iff X \le 1/2 * \min(|\sin(\theta)|, |\cos(\theta)|)$

Integrarea

Spațiul (θ, X) il luăm cu θ uniform în $[0,2\pi)$ și X uniform in $[0,1/2]$.

Densitatea comună (X,θ) este 1/aria totală$=1/\pi$ fiindcă intervalul $[0,2\pi)\times[0,1/2]$ are *aria* $2\pi\times 1/2=\pi$.

Așadar,

$$\mathbb{P}(X_1 = 1,\ X_2 = 1) \;=\; \int_0^{2\pi}\int_0^{(1/2)\min(|\sin\theta|,|\cos\theta|)} \underbrace{\frac{1}{\pi}}_{\text{densitatea}} \; dX\, d\theta.$$

$$\mathbb{P}(X_1 = 1,\ X_2 = 1) \;=\; \frac{4-2\sqrt{2}}{\pi}.$$

Prin urmare,

$$\mathbb{E}[X_1 X_2] \;=\; \frac{4-2\sqrt{2}}{\pi}.$$

Concluzia despre covarianță:

$$\mathrm{Cov}(X_1, X_2) \;=\; \frac{4-2\sqrt{2}}{\pi} \;-\; \left(\frac{2}{\pi}\right)\left(\frac{2}{\pi}\right) \;=\; \frac{4-2\sqrt{2}}{\pi} \;-\; \frac{4}{\pi^2}.$$

$$\mathrm{Var}(Z) = \mathrm{Var}(X_1 + X_2) = \underbrace{(\mathrm{Var}(X_1) + \mathrm{Var}(X_2))}_{\frac{4}{\pi} - \frac{8}{\pi^2}} + 2\,\underbrace{\mathrm{Cov}(X_1, X_2)}_{\frac{4-2\sqrt{2}}{\pi} - \frac{4}{\pi^2}}.$$

$$= \left(\frac{4}{\pi} - \frac{8}{\pi^2}\right) + 2\left(\frac{4-2\sqrt{2}}{\pi} - \frac{4}{\pi^2}\right) = \frac{4}{\pi} + \frac{8-4\sqrt{2}}{pi} - \frac{8}{\pi^2} - \frac{8}{\pi^2}.$$

$$= \frac{12-4\sqrt{2}}{\pi} - \frac{16}{\pi^2} = 4\left(\frac{3-\sqrt{2}}{\pi} - \frac{4}{\pi^2}\right).$$

În final:

$$\mathrm{Var}\left(\frac{Z}{2}\right) = \frac{1}{4}\mathrm{Var}(Z) = \frac{1}{4} \cdot 4\left(\frac{3-\sqrt{2}}{\pi} - \frac{4}{\pi^2}\right) = \frac{3-\sqrt{2}}{\pi} - \frac{4}{\pi^2}.$$

Răspuns

Aș alege crucea, deoarece varianța obținută este *mai mică* decât în cazul acului simplu, ceea ce înseamnă că pentru același număr de aruncări putem obține o precizie mai bună în estimarea lui π.

Acul simplu:

Variabila care indică “intersectează sau nu” are probabilitatea de intersecție 2/π.

Drept urmare, *estimatorul* pentru π pe baza acului simplu are varianță mai mare (in jur de 0.23023023 pentru indicator).

Crucea (două ace perpendiculare):

Poate fi privită ca două experimente într-unul singur.

Are *aceeași valoare medie* (indicatorul “număr de intersecții”/2 tot →2/π), deci *același* rezultat “așteptat”.

Varianța este *mai mică* (≈0.10), ceea ce face ca oscilațiile estimatorului sa fie mai reduse.

Concluzie: Într-un algoritm de tip Monte Carlo (unde contează cât de repede se “stabilizează” estimarea), un estimator cu varianță mai mică este *preferabil*. De aceea, „crucea lui Buffon” este o *metodă mai eficientă* pentru estimarea lui π decat un singur ac.

Simulare în R:

```
#Exercitiul 3 subpunctul b

# Numarul de experimente:
N <- 1e6

# 1. Generam distanta X in [0, 0.5]
X <- runif(N, min = 0, max = 0.5)

# 2. Generam unghiul Theta in [0, 2*pi)
Theta <- runif(N, min = 0, max = 2*pi)

# 3. Indicatori de intersectie pentru fiecare ac:
#    Ac 1 intersecteaza daca X <= 0.5 * |sin(Theta)|
X1 <- (X <= 0.5 * abs(sin(Theta)))

#    Ac 2 intersecteaza daca X <= 0.5 * |cos(Theta)|
X2 <- (X <= 0.5 * abs(cos(Theta)))

# 4. Numarul total de intersectii Z = X1 + X2.
Z <- X1 + X2  # poate fi 0, 1, sau 2 pentru o "jumatate" de cruce
# (in total crucea poate intersecta pana la 4 linii,
#  dar aici liniile sunt doar orizontale, deci max 2).

# 5. Media lui Z/2
mean_Z_over_2 <- mean(Z/2)

# 6. Estimam pi cu formula pi_est = 2 / mean(Z/2).
pi_est <- 2 / mean_Z_over_2

# 7. Eventual, vedem varianta empirica a lui Z/2
var_Z_over_2 <- var(Z/2)

cat("Rezultate pentru crucea lui Buffon:\n")
cat("Valoarea medie (Z/2):", mean_Z_over_2, "\n")
cat("Estimare pentru pi   :", pi_est, "\n")
cat("Var(Z/2) empiric     :", var_Z_over_2, "\n")
```

Rezultate:

```
cat("Rezultate pentru crucea lui Buffon:\n")
Rezultate pentru crucea lui Buffon:
> cat("Valoarea medie (Z/2):", mean_Z_over_2, "\n")
Valoarea medie (Z/2): 0.636691
> cat("Estimare pentru pi   :", pi_est, "\n")
Estimare pentru pi   : 3.141241
> cat("Var(Z/2) empiric     :", var_Z_over_2, "\n")
Var(Z/2) empiric     : 0.09956467
```

Explicații:

1. Generăm X ca distanța față de cea mai apropiată linie ∈ [0, 1/2].
2. Generăm Theta uniform în [0, 2π).
3. Pentru fiecare experiment, verificăm dacă acum 1 intersectează (X1), respectiv acum 2 (X2).
4. Z = X1 + X2 e numărul de intersecții, apoi Z/2 este exact variabila aleatoare din problemă.
5. var_z_over_2 se va apropia, cu suficient de multe repetiții, de valoarea teoretică.

c1) Considerați acum o valoare d fixată și un plan pe care sunt trasate liniile $y = n \cdot d$ (unde $n = 0, \pm1, \pm2, \ldots$). Un ac de lungime $L(< d)$ este aruncat la întâmplare pe acest plan. Arătați că probabilitatea ca acul să intersecteze vreo linie din plan este egală cu $(2L)/(\pi d)$.

Definirea variabilelor aleatoare:

X = distanța de la mijlocul acului la cea mai apropiată linie. Deoareec liniile sunt la distanța d, această distanță X va fi uniformă în intervalul [0, d/2].

Θ = unghiul dintre ac și direcția paralelă cu liniile. Considerăm unghiul față de orizontală dacă liniile sunt orizontale; aici liniile $y = n \cdot d$ sunt orizontale, deci Θ este unghiul acului cu orizontala. Se alege uniform în $[0, \pi]$, deoarece acul nu este orientat (capetele sunt indistincte, nu există „vârf” și „gămălie”, deci unghiurile θ și θ+π reprezintă aceeași orientare a acului).

Spațiul de posibilități pentru (X,θ) are, în consecință, aria:

$$\mathrm{Arie}_{\text{totală}} = \left(\frac{d}{2}\right) \times \pi = \frac{\pi d}{2}.$$

Condiția de intersecție a acului cu o linie

Pentru ca acul să atingă vreo linie, jumătatea acului, proiectată pe verticală, trebuie să fie mai mare decât distanța X. Dacă acul face unghiul θ cu orizontala, atunci proiecția verticală a jumătății acului este L/2 * sinθ. Prin urmare, acul intersectează linia dacă segmentul care este proiecția pe direcția perpendiculară liniilor depășește distanța X

$$X \le \frac{L}{2}\sin(\theta).$$

Calculul probabilității

Pentru a găsi P, determinam mai intai *aria* lui Dfavorabil din spațiul {(X,θ)}.

Domeniul total:

$$D_{\text{total}} = \{(X,\theta) \mid 0 \le X \le \tfrac{d}{2},\ 0 \le \theta \le \pi\}, \quad \text{Arie}(D_{\text{total}}) = \frac{\pi d}{2}.$$

Domeniul favorabil:

$$D_{\text{favorabil}} = \{(X,\theta) \mid 0 \le X \le \tfrac{L}{2}\sin(\theta),\ 0 \le \theta \le \pi\}.$$

Pentru o valoare fixă a lui θ, X poate varia de la 0 la L/2*sinθ.

Conditionăm la θ și integrăm în X:

$$\text{Arie}(D_{\text{favorabil}}) = \int_0^{\pi} \int_0^{(L/2)\sin(\theta)} \underbrace{1}_{\text{densitatea de suprafață constantă}} dX\, d\theta.$$

Apoi:

$$\text{Arie}(D_{\text{favorabil}}) = \int_0^{\pi} \frac{L}{2}\sin(\theta)\, d\theta = \frac{L}{2}[-\cos(\theta)]_0^{\pi} = \frac{L}{2}[-\cos(\pi)+\cos(0)] = \frac{L}{2}(+1+1) = L.$$

Astfel, probabilitatea ca un ac de lungime L (L<d) sa intersecteze liniile orizontale desparțite la distanța d este

$$P = \frac{\text{Arie}(D_{\text{favorabil}})}{\text{Arie}(D_{\text{total}})} = \frac{L}{\frac{\pi d}{2}} = \frac{2L}{\pi d}.$$

Metoda 2:

Probabilitatea condiționată, pentru un unghi fix θ, ca acul să intersecteze o linie este:

$$P(\text{intersectare} \mid \theta) = \frac{\text{măsura valorilor lui } x \text{ care satisfac condiția}}{\text{lungimea intervalului de } x} = \frac{\frac{L}{2}\sin\theta}{\frac{d}{2}} = \frac{L\sin\theta}{d}$$

pentru θ astfel încât $L/2 \sin\theta \le d/2$ (ceea ce este adevărat deoarece $L<d$).

Pentru a obține probabilitatea totală, trebuie să integrăm peste toate valorile posibile ale unghiului θ (folosind densitatea unghiului $f(\theta)=2/\pi$ pentru $\theta \in [0,\pi/2]$:

$$P = \int_0^{\pi/2} P(\text{intersectare} \mid \theta)\, f(\theta)\, d\theta = \int_0^{\pi/2} \frac{L\sin\theta}{d} \cdot \frac{2}{\pi}\, d\theta.$$

Scoatem constantele din integrală:

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

Calculăm integrala:

$$\int_0^{\pi/2} \sin\theta\, d\theta = [-\cos\theta]_0^{\pi/2} = (-\cos(\pi/2)) - (-\cos(0)) = (0) - (-1) = 1.$$

Astfel obținem:

$$P = \frac{2L}{\pi d} \cdot 1 = \frac{2L}{\pi d}.$$

Simulare în R:

```
#Exercitiul 3 subpunctul c1

# Setam numarul de simulari
N <- 1e6

# Parametrii:
d <- 2    # distanta dintre linii
L <- 1    # lungimea acului, L < d

# 1. Generam X (distanta mijlocului acului fata de linia cea mai apropiata):
X <- runif(N, min = 0, max = d/2)
```

```r
# 2. Generam unghiul Theta in [0, pi]:
Theta <- runif(N, min = 0, max = pi)

# 3. Verificam intersectia: X <= (L/2) * sin(Theta)
intersecteaza <- (X <= (L/2)*sin(Theta))

# 4. Probabilitatea estimata (frecventa empirica de intersectie):
prob_estimata <- mean(intersecteaza)

# 5. Probabilitatea teoretica:
prob_teoretica <- (2 * L) / (pi * d)

cat("Rezultate Buffon generalizat:\n")
cat(" - Probabilitatea estimata :", prob_estimata, "\n")
cat(" - Probabilitatea teoretica:", prob_teoretica, "\n")
```

Rezultate:

```
> cat("Rezultate Buffon generalizat:\n")
Rezultate Buffon generalizat:
> cat(" - Probabilitatea estimata :", prob_estimata, "\n")
 - Probabilitatea estimata : 0.318558
> cat(" - Probabilitatea teoretica:", prob_teoretica, "\n")
 - Probabilitatea teoretica: 0.3183099
> cat(" - Eroarea absoluta        :", abs(prob_estimata - prob_teoretica), "\n
")
 - Eroarea absoluta        : 0.0002481138
```

Explicații:

1. X <- runif(N, 0, d/2): generăm distanța până la linia cea mai apropiată.
2. Theta <- runif(N, 0, pi): unghiul acului cu orizontala.
3. Condiția $1\{X \le L2\sin(\theta)\}$ determină dacă acul intersectează o linie.
4. mean dă frecvența empirică de intersecție, care, converge la probabilitatea reală pe măsură ce N crește (noțiune curs).
5. Valoarea teoretică este $2 * L/\pi * d$.

c2) Mai departe, fixați acum poziția acului și considerați un cerc C de diametru d, centrat în mijlocul acului. Fie λ o linie ale cărei direcție și distanță față de centrul lui C sunt independente și uniform distribuite pe $[0, 2\pi]$ și, respectiv, $[0, d/2]$. Arătați că probabilitatea ca acul să se intersecteze cu linia (aleatoare) λ este egală (tot cu) $(2L/ \pi d)$.

Descrierea problemei:

Avem un ac de lungime L (cu $L<d$), fixat în plan. Centru un cerc C cu diametru d în mijlocul acului. Alegem o linie λ în mod aleator astfel: Orientarea Θ a liniei este uniformă în $[0,2\pi)$. (Sau $[0,\pi)$, depinde cum definim, dar ideea e că are orientare uniformă.) Distanța ρ de la centrul cercului la linie este uniformă în $[0,d/2]$.

Interpretăm astfel : imaginăm toate liniile care „trec” la distanță cel mult d/2 de centrul cercului (adică intersectează discul de rază d). Se alege uniform atât unghiul, cât și distanța până la centrul C.

Problema ne spune că probabilitatea ca acest ac fix să fie intersectat de linia λ este aceeași cu formula clasică.

Notare:

Notăm cu O centrul cercului C (care e și mijlocul acului).

Linia aleatoare λ se specifică prin doi parametri (ρ,Θ):

$\rho\in[0,d/2]$ = distanța liniei fata de O.

$\Theta\in[0, 2\pi)$= unghiul liniei față de un reper fix (de ex., unghiul fată de axa Ox).

Știm că dacă o linie are ecuația $r=\rho$ cu unghi Θ (în coordonate polar-normale), atunci $0\leq\rho\leq d/2$.

Acul are lungime L și e așezat fix.

α = unghiul acului (față de orizontală sau față de axa Ox) — dar acest unghi e fix, nu intră la aleator.

Poziția acului e fixată cu mijlocul în O.

Când intersectează linia λ?

Distanța de la O la linie este ρ.

Unghiul dintre ac (fix) și normalul la linie este Δ=|Θ−α|.

Din subpunctul a știm că, pentru un ac de lungime L, condiția de a intersecta linia este ca „jumătatea acului, pe direcția perpendiculară la linie” să fie mai mare decât ρ.

Mai explicit:

intersectează ⇔ $\rho \leq L/2 * \sin(\Delta)$.

Aici Δ este unghiul dintre ac și linie (mai exact, dintre ac și perpendiculara pe linie).

Calculul probabilității:

Prin ipoteză, ρ este uniform în [0,d2].

Θ este uniform în [0,2π).

ρ și Θ sunt independente.

Astfel, densitatea comună e:

$$f(\rho,\Theta) = \frac{1}{\text{aria cercului} = \pi\left(\frac{d}{2}\right)^2} = \frac{1}{\frac{\pi d^2}{4}} = \frac{4}{\pi d^2}, \quad 0 \leq \rho \leq \frac{d}{2},\ 0 \leq \Theta < 2\pi.$$

Notă: Este logic să aibă loc un factor constant, important e că întreg domeniul ρ∈[0,d/2], Θ∈[0,2π) are aria d/2×2π=πd.

Domeniul favorabil:

Însă α e fix și Θ e uniform în [0,2π). Prin simetrie, nu contează ce valoare are α: spunem că Θ e ales relativ la α.

Asadar, unghiul Δ=Θ−α va fi și el uniform în [0,2π). Rescriem:

$$\rho \ \leq \ \frac{L}{2}\,|\sin(\Delta)|.$$

Calculul ariei favorabile:

Procedăm ca la Buffon standard, dar cu ρ∈[0,d/2], Δ∈[0,2π). Folosim o integrală dublă:

$$\mathbb{P}(\text{intersectează}) = \int_{\Delta=0}^{2\pi}\int_{\rho=0}^{(L/2)\,|\sin\Delta|} f(\rho,\Delta)\;d\rho\,d\Delta.$$

Densitatea comună f(ρ,Δ) este 1/aria totală. Aici, aria totală a domeniului {(ρ,Δ):0≤ρ≤d/2, 0≤Δ<2π} este πd/2 sau πd după cum parametrii (ρ,Θ) se definesc.

Pentru coerență cu Buffonul clasic, luăm:

“aria totală”=(d/2)×2π=πd

Deci

$$f(\rho,\Delta) \;=\; \frac{1}{\pi d}.$$

Integrarea

$$\int_{\Delta=0}^{2\pi}\int_{\rho=0}^{(L/2)\,|\sin\Delta|}\frac{1}{\pi d}\;d\rho\,d\Delta = \frac{1}{\pi d}\int_{0}^{2\pi}\left(\tfrac{L}{2}\,|\sin\Delta|\right)\,d\Delta.$$

Prin urmare:

$$\mathbb{P}(\text{intersectează}) \;=\; \frac{1}{\pi d}\cdot\frac{L}{2}\cdot\int_{0}^{2\pi}|\sin\Delta|\,d\Delta = \frac{1}{\pi d}\cdot\frac{L}{2}\cdot 4 = \frac{2L}{\pi d}.$$

Astfel, chiar dacă acul e fix şi linia λ este cea aleatoare (distribuită uniform în cercul de diametru d), probabilitatea de intersecţie ramâne aceeaşi.

Simulare în R:

```r
# Exercitiul 3 subpunctul c2

# Numarul de simulari
N <- 1e6

# Parametrii
d <- 2      # diametrul cercului (si distanta maxima pana la linie)
L <- 1      # lungimea acului (L < d)

# 1. Generam unghiul alpha uniform in [0, 2*pi)
alpha <- runif(N, 0, 2*pi)

# 2. Generam distanta rho uniforma in [0, d/2]
rho <- runif(N, 0, d/2)

# 3. Conditia de intersectie: rho <= (L/2)*abs(sin(alpha))
intersecteaza <- (rho <= (L/2) * abs(sin(alpha)))

# 4. Probabilitatea estimata (frecventa empirica):
prob_est <- mean(intersecteaza)

# 5. Valoarea teoretica:
prob_theo <- (2 * L) / (pi * d)

cat("Probabilitatea estimata  :", prob_est, "\n")
cat("Probabilitatea teoretica :", prob_theo, "\n")
cat("Eroarea absoluta         :", abs(prob_est - prob_theo), "\n")
```

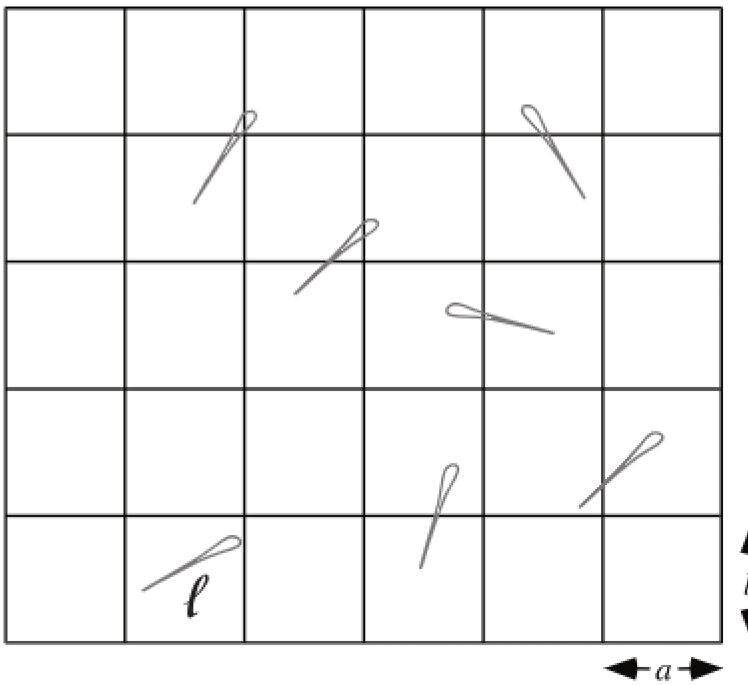
Răspunsuri:

```
> cat("Probabilitatea estimata  :", prob_est, "\n")
Probabilitatea estimata  : 0.318083
> cat("Probabilitatea teoretica :", prob_theo, "\n")
Probabilitatea teoretica : 0.3183099
> cat("Eroarea absoluta         :", abs(prob_est - prob_theo), "\n")
Eroarea absoluta         : 0.0002268862
```

Explicații:

1. alpha <- runif(N, 0, 2*pi): unghiul liniei față de Ox.
2. rho <- runif(N, 0, d/2): distanța de la origine la linie.
3. Condiția rho <= (L/2)*abs(sin(alpha)) este exact “linia intersectează segmentul orizontal de lungime L, centrat în origine”.
4. mean(intersecteaza) dă frecvența empirică a evenimentului de intersecție, care converge către $2L/\pi d$

d) Pe un plan se consideră următorul grid format cu 2 seturi de linii paralele suprapuse: primul set conține linii la distanța d1 unele de altele, iar al doilea set conține linii la distanța d2 unele de altele și perpendiculare pe cele din primul set. Un ac de lungime L < min{d1, d2} este aruncat la întâmplare pe acest plan. Arătați că probabilitatea ca acesta să intersecteze planul este egală cu L*(2*d1+2*d2−L)/(π*d1*d2).



Ideea principală: Incluziune–Excluziune

Fie:

A = evenimentul „acul intersectează o linie verticală” (dintre cele distantate la d2).

B = evenimentul „acul intersectează o linie orizontală” (dintre cele distanțate la d1).

Atunci probabilitatea este:

P(acul intersecteaza grilajul) = P(A∪B) = P(A)+P(B) − P(A∩B)

Probabilitatea de a intersecta *verticale*

Dacă liniile *verticale* sunt la distanță d2, cazul este exact ca în „acul lui Buffon” cu distanță d2. Se știe că, pentru L<d2,

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

Probabilitatea de a intersecta *orizontale*

Analog, liniile *orizontale* sunt la distanță d1, deci

$$P(B) = \frac{2L}{\pi d_1}.$$

Probabilitatea de a intersecta *ambele* tipuri de linii

Notăm A∩B= evenimentul „acul intersectează *o linie verticală* și *o linie orizontală* (simultan)”.

Parametrii X,Y,θ

X∈[0,2d2] = *distanța de la mijlocul acului* la cea mai apropiată linie verticală (aceasta e în mod normal ∣x−m d2∣ “folded” în [0,d2/2].

Y∈[0, d12] = *distanța de la mijlocul acului* la cea mai apropiată linie orizontală.

Motiv: Deoarece liniile verticale sunt la distanță d2, orice punct de pe axa Ox se poate „plia” (mod d2) în intervalul [0,d2]. Mai exact, *pentru mijlocul acului*, luăm doar distanța ≤d2/2 la linia verticală cea mai apropiată. Similar cu Y și liniile orizontale ([0,d1/2]).

θ∈[0,π] = *orientarea* acului față de axa OxOxOx.

Pentru un ac, unghiurile θ și θ+π reprezintă aceeași poziție (capetele nu sunt diferențiate).

Deci θ se ia uniform pe [0,π]

Spațiul tuturor configurațiilor (X,Y,θ):

$$D_{\text{total}} = \left[0, \tfrac{d_2}{2}\right] \times \left[0, \tfrac{d_1}{2}\right] \times [0, \pi].$$

"Volumul" acestui spațiu (în sens 2D × unghi) este:

$$\underbrace{\left(\tfrac{d_2}{2}\right)\left(\tfrac{d_1}{2}\right)}_{\frac{d_1 d_2}{4}} \times \underbrace{\pi}_{\theta\in[0,\pi]} = \frac{\pi\, d_1\, d_2}{4}.$$

Se consideră distribuție *uniform* în această "căsuță pliată", corespunzând faptului că acul e "aruncat la întâmplare" pe tot planul (și unghiul său e uniform).

Condiții de intersectare:

Intersectare cu o verticală:

Acul intersectează (cel puțin) o linie verticală ⟺

$$X \le \frac{L}{2}\,|\cos\theta|.$$

Intersectare cu o orizontală

Analog, acul intersectează (cel puțin) o linie orizontală ⟺

$$Y \le \frac{L}{2}\,|\sin\theta|.$$

Intersectare ambele tipuri (verticală + orizontală):

Prin urmare, evenimentul A∩B = “acul intersectează verticală și orizontală” $\Leftrightarrow$

$$\begin{cases} X \le \frac{L}{2}\,|\cos\theta|, \\ Y \le \frac{L}{2}\,|\sin\theta|. \end{cases}$$

Domeniul favorabil din (X,Y,θ) :

$$D_{\text{fav}} = \left\{(X,Y,\theta) \mid 0 \le X \le \tfrac{d_2}{2},\ 0 \le Y \le \tfrac{d_1}{2},\ 0 \le \theta \le \pi,\ X \le \tfrac{L}{2}\,|\cos\theta|,\ Y \le \tfrac{L}{2}\,|\sin\theta|\right\}.$$

Calculul “volumului” favorabil

Pentru a găsi Vol(D_{fav}), observăm:

Vom considera θ∈[0,π/2] și θ∈[π/2,π].

În [0,π/2], cosθ≥0, sinθ≥0. Deci |cosθ|=cosθ| si |sinθ|=sinθ.

În [π/2,π], cosθ<=0 si sinθ≥0, deci -cosθ>=0.

Simetria unghiurilor face să dublăm rezultatul din [0,π/2].

Domeniul θ∈[0,π/2]

Aici, cosθ≥0, sinθ≥0. Condițiile devin:

$$X \le \tfrac{L}{2}\cos\theta, \quad Y \le \tfrac{L}{2}\sin\theta.$$

Volumul parțial:

$$\int_{\theta=0}^{\pi/2} \int_{X=0}^{(L/2)\cos\theta} \int_{Y=0}^{(L/2)\sin\theta} dY\, dX\, d\theta.$$

Integrarea in Y:

$$\int_0^{(L/2)\sin\theta} dY = \tfrac{L}{2}\sin\theta.$$

Integrarea in X:

$$\int_0^{(L/2)\cos\theta} \left(\tfrac{L}{2}\sin\theta\right) dX = \left(\tfrac{L}{2}\sin\theta\right) \times (L/2)\cos\theta = \frac{L^2}{4}\sin\theta\,\cos\theta.$$

Integrarea în θ pe [0,π/2]:

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

Dar $\int_0^{\pi/2} \sin\theta\,\cos\theta\,d\theta = \frac{1}{2}$. .Deci obținem $\frac{L^2}{4} * \frac{1}{2} = \frac{L^2}{8}$

Domeniul θ∈[π/2,π]

Aici, sinθ≥0 dar cosθ≤0. Se obține aceeași integrală prin simetrie,$\frac{L^2}{8}$.

Prin urmare, întregul interval θ∈[0,π] dă:

$$\mathrm{Vol}(D_{\mathrm{fav}}) = \tfrac{L^2}{8} + \tfrac{L^2}{8} = \frac{L^2}{4}.$$

Probabilitatea

Probabilitatea evenimentului A∩B este raportul dintre “volumul favorabil” și “volumul total”:

$$P(A \cap B) \;=\; \frac{\mathrm{Vol}(D_{\mathrm{fav}})}{\mathrm{Vol}(D_{\mathrm{total}})} \;=\; \frac{\frac{L^2}{4}}{\frac{\pi\, d_1\, d_2}{4}} \;=\; \frac{L^2}{\pi\, d_1\, d_2}.$$

Revenim la P(A∪B)=P(A)+P(B)−P(A∩B)

P(A)=2L/πd2.

P(B)=2L/πd1.

P(A∩B)=L^2/πd1d2.

Rezultă:

$$\begin{aligned} P(A \cup B) &= \frac{2\,L}{\pi d_2} \;+\; \frac{2\,L}{\pi d_1} \;-\; \frac{L^2}{\pi d_1 d_2} \\ &= \frac{1}{\pi\, d_1\, d_2}\Big[\,2\,L\,d_1 + 2\,L\,d_2 - L^2\Big] \\ &= \frac{L}{\pi\, d_1\, d_2}\Big[\,2(d_1 + d_2) - L\Big]. \end{aligned}$$

Simulare în R:

```r
# Exercitiul 3 subpunctul d

# Parametri
d1 <- 2      # distanta intre linii orizontale
d2 <- 3      # distanta intre linii verticale
L  <- 1      # lungimea acului (trebuie L < min(d1,d2))

# Numarul de simulari
N <- 1e6

# 1. Generam X, Y, Theta
X <- runif(N, min = 0, max = d2/2)
Y <- runif(N, min = 0, max = d1/2)
Theta <- runif(N, min = 0, max = pi)

# 2. Conditie: intersecteaza verticala?
intersect_vertical <- (X <= (L/2)*abs(cos(Theta)))

# 3. Conditie: intersecteaza orizontala?
intersect_horizontal <- (Y <= (L/2)*abs(sin(Theta)))

# 4. Acul intersecteaza grilajul (vertical OR horizontal)
intersect_grid <- (intersect_vertical | intersect_horizontal)

# 5. Estimam probabilitatea (frecventa empirica)
prob_est <- mean(intersect_grid)

# 6. Probabilitatea teoretica
prob_theo <- (L * (2*(d1 + d2) - L)) / (pi * d1 * d2)

# Afisam rezultatele
cat("Probabilitate estimata  :", prob_est, "\n")
cat("Probabilitate teoretica :", prob_theo, "\n")
cat("Eroare absoluta         :", abs(prob_est - prob_theo), "\n")
```

Rezultate:

```
> # Afisam rezultatele
> cat("Probabilitate estimata  :", prob_est, "\n")
Probabilitate estimata  : 0.477612
> cat("Probabilitate teoretica :", prob_theo, "\n")
Probabilitate teoretica : 0.4774648
> cat("Eroare absoluta         :", abs(prob_est - prob_theo), "\n")
Eroare absoluta         : 0.0001471707
```

Explicații:

1. intersect_vertical este un vector boolean de mărime N care spune dacă, la fiecare simulare, acul atinge o linie verticală.
2. intersect_horizontal analog, pentru orizontală.
3. Operația | exprimă faptul că acul intersectează {cel puțin o linie} dacă intersectează verticală sau orizontală (ori ambele).
4. mean(intersect_grid) este frecvența empirică a evenimentului “intersectează”,care converge la probabilitatea adevărată, pe măsură ce N crește.

Diferența dintre ele (eroarea) va scădea când N e suficient de mare, confirmând formula Buffon–Laplace pentru probabilitatea că acul (de lungime $L<\min(d1,d2)$) intersectează un grilaj dreptunghiular.

e) În cadrul simulărilor pentru această problemă ce tip de strategie aleatoare a fost implementată: Las Vegas sau Monte Carlo? Justificați sumar și dați un exemplu de algoritm aleator (cu implementare) din cealalta categorie.

Pentru problema anterioară (generarea a N poziții ale acului, calculul frecvenței de intersecție etc.), algoritmul este de tip Monte Carlo.

Explicație:

Monte Carlo este un algoritm care returnează un rezultat cu o anumită aproximație (sau cu un anumit risc de eroare), însă timpul de rulare este (relativ) determinist sau fixat (în sensul că nu așteptăm la nesfârșit un eveniment rar pentru a obține un rezultat exact).

Las Vegas: un algoritm aleator care returnează întotdeauna un rezultat corect (fără eroare), dar durata lui poate fi aleatorie și potențial nelimitată (așteaptă un eveniment care să se întâmple).

În problema Buffon–Laplace, simulăm plasarea acului de N ori (poziție + unghi). După N aruncări, aproximăm probabilitatea dorită (număr de intersecții / N). Rezultatul este aproximat (are o abatere statistică), însă timpul de rulare este clar (de regulă O(N)). Nu garantăm un răspuns exact, ci unul care devine mai bun pe măsură ce creștem N.

Prin urmare, este un algoritm Monte Carlo:

1. Obținem probabil o aproximație a răspunsului corect,
2. Avem control asupra numărului de repetiții (timp de rulare),
3. Putem estima eroarea.

Pentru a exemplifica tipul Las Vegas, am să dau un exemplu știut încă din liceu, puțin modificat: QuickSort randomizat în varianta în care pivotul se alege aleator. Acesta:

1. întotdeauna sortează corect (fără eroare de sortare), deci rezultatul e 100% bun,
2. timpul de execuție (numărul de comparații) este aleator, dar in cazul cel mai bun este O(nlogn).

El se poate considera Las Vegas deoarece produce un rezultat determinist corect, însă durata diferă de la o execuție la alta.

Simulare în R:

```r
# Exercitiul 3 subpunctul e

randomizedQuickSort <- function(A) {
  # Daca lungimea vectorului e 0 sau 1, e deja sortat
  if (length(A) < 2) {
    return(A)
  }

  # Alegem pivotul in mod aleator
  pivot_idx <- sample(1:length(A), 1)
  pivot <- A[pivot_idx]

  # Scoatem pivotul din vector si partitionam
  rest <- A[-pivot_idx]
  left <- rest[rest <= pivot]
  right <- rest[rest > pivot]

  # Apel recursiv
  left_sorted <- randomizedQuickSort(left)
  right_sorted <- randomizedQuickSort(right)

  # Recompunem rezultatul final
  return( c(left_sorted, pivot, right_sorted) )
}

# Exemplu 1: vector de mici dimensiuni
A <- c(3, 1, 4, 1, 5, 9, 2, 6, 5)
sorted_A <- randomizedQuickSort(A)
print(sorted_A)
print(sort(A))
# Verificam daca cele doua rezultate sunt egale, functia facuta de mine si
sortarea generica:
identical(sorted_A, sort(A))

# Vector gol
print(randomizedQuickSort(c()))

# Vector cu un singur element
print(randomizedQuickSort(c(42)))

# Generam un vector de 20 de numere intregi aleatoare intre 1 si 100
set.seed(123)  # Pentru consistenta alegerii numerelor
random_vector <- sample(1:100, 20, replace = TRUE)
```

```
sorted_vector <- randomizedQuickSort(random_vector)
print(random_vector)
print(sorted_vector)
print(sort(random_vector))
```

În final, se afișează vectorul sortat, fără a exista erori.

Bibliografie

Problema 1

Calcul Diferențial și Integral - Lect. Dr. Alexandru Iancu Mihail, Universitatea din București, Facultatea de Matematică și Informatică

Probabilități și Statistică – Lect Dr. Alexandru Amărioarei, Universitatea din București, Facultatea de Matematică și Informatică

Problema 2

Shiny - Posit

Rdocumentation – Shiny Package

Shiny App Education Team – Style Guide

Mastering Shiny

Problema 3

Câmpuri de probabilitate – Universitatea Tehnică „Gheorghe Asachi” din Iași

Buffon’s Needle Problem – Wolfram MathWorld

Buffon’s Needle. An Analysis and Simulation – University of Illinois Urbana-Champaign, Office for Mathematics, Science, and Technology Education

Buffon’s needle problem – Universiteit Leiden, Mathematical Institute, Proofs from the Book (2004, 3rd), Chapter 21

Imagini: Buffon’s Needle Problem – Wolfram MathWorld