**Proiect pentru materia „Probabilități și statistică”**

**PROIECT 3**

**„Cȃteva aplicații cu variabile aleatoare ȋn R Studio”**

Realizat de : Gherasim Rareș, Mihai Mihail, Mihai Alexandru, Bumbăcea Diana

Facultatea de Matematică și Informatică, specializarea Informatică

Anul II, Grupa 243

Liderul Echipei: Gherasim Rareș.

**Exercitul 1:**

*a) Functia frepcomgen(m,n)*

Functia cconstruieste initial 2 matrici x si y cu dimensiuni 2\*m si respectiv 2\*n, cu toate elementele initializate cu 0. Pentru fiecare matrice actualizam elementele in felul urmator:

Pentru fiecare dintre elementele de pe prima linia se genereaza o valoare aleatorie diferita folosind functia sample(), iar pentru elemetele de pe a doua linie o valoare aleatorie in intervalul 0 si valoarea aflata in variabila sum\_prob care ne arata suma probabilitatilor ramase de pus pe linie.

Pentru a construi tabelul cu repartitia comuna a lui x si y, construim o matrice n\*m cu toate elementele 0. Pentru fiecare linie, generam un numar random intre 1 si m stocat in variabila pozitie\_libera, astfel aceea pozitie nu va fi actualizata. Pentru fiecare pozitie (i,j), generam un numar random in intervalul 0 si minimul dintre x[2,i] si y[2,j], dupa care actualizam valorile lui x[2,i] si y[2,j] scazand valoarea aleasa pentru pozitia (i,j) din matrice.

*b) Functia fcomplrepcom(matrice)*

Functia are ca paramentru matricea returnata de functia frepcomgen. Primul while se executa pana cand a fost gasita matricea finala. Al doilea while trateaza cazul pentru matricea patratica si numara pentru fiecare linie si pentru fiecare coloana, numarul de zerouri. Urmatoarele 2 while-uri verifică elementele rămase în cazul în care matricea nu este pătratica. Daca pe o linie sau pe coloana se afla o singura valoare de 0, se inlocuie acea valoare cu diferenta dintre suma totala a liniei\coloanei si suma celorlalte elemente de pe aceea linie\coloana (x[2, i] - suma elementelor de pe linia i ,respectiv y[2, j] suma elementelor de pe coloana j). In cazul in care matricea nu este patratica, urmatoarele 2 while -uri verifica liniile si coloanele ramase. La final, se mai face o verificare dacă sunt zero-uri de înlocuit, iar dacă nu sunt, se returneaza matricea.

c)

*1) Cov(5X+9,-3Y-2)*

In variabilele f\_x si f\_y calculam si stocam 5X+9 si -3Y-2. Dupa care calculam mediile pentru variabilele aleatoare f\_x, f\_y, f\_x\*f\_y si le stocam in med\_x, med\_y si med\_xy. Calculam covarianta dupa formula cov\_xy = med\_xy - med\_x \* med\_y

*2) P(0<X<0.8/Y>0.3) = P(0<X<0.8) intersectatat P(Y>0.3) / P(Y>0.3) = P(0<X<0.8) \* P(Y>0.3) / P(Y>0.3) = P(0<X<0.8)*

Probabilitatea se obtine facand suma probabilitatilor pentru care x se afla in intervalul (0,0.8) Calculam aceasta suma adunand toate elementele de pe prima linie a matricei x care respecta conditia si o stocam in sum\_x

*3) P(X>0.2,Y<1.7) = P(X>0.2) intersectat P(Y<1.7) = P(X>0.2) \* P(Y<1.7).*

Calculam P(X>0.2) si P(Y<1.7) prin acelasi procedeu ca la punctul anterior si facem produsul lor pentru a obtine raspunsul final

d) Functia fverind verifica daca variabilele aleatoare x si y sunt independente. Pentru a fi idependente trebuie ca produsul x[2, i] \* y[2, j] sa fie egal cu elementul de pe pozitia (i,j) din matricea comuna aflata la punctul b) pentru toate elementele din matricea comuna. Aceasta functie returneaza mesajul "Sunt independente" in caz afirmativ sau mesajul "Nu sunt independente" altfel.

Functia fvrnecor verifica daca variabilele aleatoare x si y sunt necorelate. Pentru a fi necorelate trebuie ca covalenta lui X si Y sa fie egala cu 0, altfel nu sunt corelate. Calculam mediile pentru variabilele aleatoare x, y, x\*y si le stocam in med\_x, med\_y si med\_xy. Calculam covarianta dupa formula cov\_xy = med\_xy - med\_x \* med\_y.

In functie de rezultatul obtinut, functia returneaza mesajul "Sunt necorelate" in cazul in care cov\_xy = 0 si mesajul "Sunt corelate" altfel.

**Exercitiul 2 :**

**2) Suma**

Suma a doua variabile aleatoare continue independente folosind formula de convolutie

Pentru 2 variabile aleatoare continue X si Y, functia va calcula Z = X + Y

Parametrul f este pdf-ul variabilei X, functia f trebuie definita pe (-inf, inf)

Parametrul g este pdf-ul variabilei Y, functia g trebuie definita pe (-inf, inf)

Functia intoarce pdf-ul variabilei Z

Deoarece variabilele X si Y sunt independente, pdf-ul lui Z este convolutia f \* g

**3) Diferenta**

Diferenta a doua variabile aleatoare continue independente folosind formula de convolutie

Pentru 2 variabile aleatoare continue X si Y, functia va calcula Z = X - Y

Parametrul f este pdf-ul variabilei X, functia f trebuie definita pe (-inf, inf)

Parametrul g este pdf-ul variabilei Y, functia g trebuie definita pe (-inf, inf)

Functia intoarce pdf-ul variabilei Z

Pentru a calcula Z ne vom folosi de Z = X + (-Y)

Deoarece variabilele X si Y sunt independente, pdf-ul lui Z este convolutia dintre functia f si pdf-ul variabilei -Y

**Exercitiul 3 :** Construiți o funcție ȋn R care să se comporte ca un generator de numere aleatoare(se generează n valori, unde n este dat de utilizator) avȃnd următoarele specificații: prima valoare este calculata in functie de ora sistemulu modulo 17, prin alegerea unor repartitii diferite in functie de rezultatul modulului. Urmatoarele valori sunt realizate cu ajutorul formulei: xn=a\* xn-1+b unde a este o valoare generată cu funcția rexp din R de parametru 5, iar b este o valoare generată cu funcția rnorm din R de parametri 2 și 1.

Pentru realizarea proiectului a fost nevoie în primul de mai multe tipuri de distrbuții pentru care am căutat informații pe internet, respectiv:

* Rnorm 🡪 Distribuția Normală
* Rpois 🡪 Distribuția Poisson
* Rexp 🡪 Distribuția Exponențială
* Rbinom 🡪 Distribuția Binomială
* Runif 🡪 Distribuția Uniformă
* Rgamma 🡪 Distribuția Gama
* Rhyper 🡪 Distribuția Hyper

Deoarece aceste distribuții au proprietate de a returna mai multe valori, pentru că noi ne dorim la fiecare pas generarea unei singure valori am preluat doar prima valoare generată de fiecare.

Programul realizat este alcătuit din două funcții.

Prima funcție *calcPrimaValoare* are rolul de a ne ajuta să generăm prima valoare din șir. Astfel, pentru început reținem de câte ori s-a încercat calcularea valorii (dacă calcularea eșueaza de 2 ori la rând, se trece pe un caz de baza cu *rnorm(0,1)* ), iar *val* va reține valoare ce trebuie returnată. Condiția repetitivă va rula atâta timp cât nu s-a atins numărul maxim de încercări. Pentru început, în condiția repetivă calculăm *valDeCal* care va reprezenta ora sistemului (11:50:30 🡪 15030; 02:23:11 🡪 22311). După care calculăm restul acestei valori și în funcție de acesta ne decidem dacă problema continuă pe unul dintre cazurile predefinite de problemă sau se intră pe *else*-ul final, unde așteptăm să treacă o secundă, după care recalculăm timpul și reluăm pașii. Se poate observa că *tries* se setează la 3 când intrăm pe un caz predefinit, pentru a opri while-ul și pentru a nu intra pe cazul de dinafara acestuia. Dupa care se returneaza valoarea.

A doua functie generareSirAleator(n) generează sirul cerut și-l returnează, primind ca parametru numărul de elemente pe care le dorim să fie în șir. Întâi generăm primul element cu ajutorul funcției de mai sus și îl adăugăm în șir, după care pe baza formulei din problemă le generăm pe restul până ajungem să avem n numere în șir.

În final realizăm histograma valorilor, care după parerea mea pare să ne arate ca valorile generate prin acest algoritm ies foarte rar din intervalul [-10,10].

**Exercițiul 4:**

*Media* : Am efectuat atat media aritmetica a timpilor cât și media aritmetică a concentrațiilor plasmei. Am făcut la fel și pentru cea mai mare valoare medie

*Varianța:* Varianța este o măsură numerică a modului în care valorile datelor sunt dispersate în jurul valorii medii. În special, varianța eșantionului este definită ca:

          n
s2 =--1--∑  (x - ¯x)2
    n - 1i=1  i


*Quartile*: Există mai multe quartile ale unei variabile de observare. Prima sau cea inferioară este valoarea care taie primele 25%, din date atunci când este sortată în ordine crescătoare. A doua, sau mediana, este valoarea care întrerupe primele 50%. A treia sau cea superioară este valoarea care întrerupe primele 75%. Quartilele sunt numite si cele 4 cuantile.

*Boxplot*: Diagrama Boxplot este un grafic care înglobează cele mai importante caracteristici statistice ale unor distribuţii de frecvenţă, pentru a oferi o mai bună înţelegere şi comparaţie.

*Interpretări*:

* + - Cea mai mare valoare medie pentru durata datasetului e 2, în timp ce pentru concentrația plasmei este 0.34.
    - În timp ce timpul mediu este ~2.88, concentrația medie este ~0.59
    - Varianța în jurul valorii medii este destul de vastă, având valoarea de ~6.07
    - Prima quartilă are valoarea 0,75, cea de-a doua 2.00, iar ultima 5. Ele impart intervalul în 4 părți.
    - Graficul se poate interpret astfel :
* concentratia ia valoare in intervalul (0,2) al unităților graficului. Explicitând prin funcțiile min și max încardrăm concentrația între [0.05,2,72]
* Timpul se incadreaza între (0,8) al unitatilor graficului, fiind explicitat in min[0.25, 8]
* Iar codul subiectelor se incadreaza intre (1,6)al graficulu, fiind explicitat in [1,3]

**Bibliografie:**

https://www.rdocumentation.org/

https://stat.ethz.ch/R-manual

https://statisticsglobe.com/