Proiect Probabilități și Statistică

ContinuousRandomVars

Documentație

Ducal Nicolae (Lider)
Tender Laura - Maria
Avram Daniel

Grupa 234

1 Februarie, 2021

Cuprins

Descrierea problemei

În cadrul acestui proiect, am ales să realizăm un pachet R denumit "ContinuousRandomVars". Acesta corespunde primului proiect dintre cele propuse, construirea unui pachet R pentru lucrul cu variabile aleatoare continue. Problema constă în a crea un pachet R care să permită obținerea direct și cu usurință a informațiilor despre variabile aleatoare continue, un "instrument" de lucru asemenea pachetului R discreteRV dedicat lucrului cu variabile aleatoare discrete.

În cadrul proiectului, am implementat următoarele cerințe:

- **normalizing_constant** (funcție, limita inferioară, limita superioară)
  (Cerința 1)
  Determinarea unei constante de normalizare k pentru o funcție introdusă de utilizator, iar în cazul în care nu există, afișarea unui mesaj corespunzător către utilizator.
  De asemenea există funcția **normalize_function** care primește ca parametrii o funcție, limita inferioară și cea superioară și returnează funcția normalizată.

- **check_if_pdf** (funcție, limita inferioară, limita superioară)
  (Cerința 2)
  Să verificăm dacă o funcție introdusă de utilizator este densitate de probabilitate.

- **ContRV** (funcția densitate de probabilitate, limita inferioară, limita superioară)
  (Cerința 3)
  Crearea unui obiect de tip variabilă aleatoare continuă primind o densitate de probabilitate introdusă de utilizator.

- **plot_RV** (obiect de tip variabilă aleatoare continuă, intervalul de valori a lui x in grafic, intervalul de valori a lui y in grafic, culoare)

plot_distribution (numele repatiției, intervalul de valori a lui x in grafic, intervalul de valori a lui y in grafic, ...)
(Cerința 4)
Reprezentarea grafică a densității de probabilitate și a funcției de repartiție pentru diferite valori ale parametrilor repartiției. Pentru funcțiile de repartiție care nu sunt date într-o formă explicită (ex. repartiția normală) sunt reprezentate grafic aproximări ale acestora.

- **mean** (obiect de tip variabilă aleatoare continuă) Media unei variabile aleatoare continue
  Var (obiect de tip variabilă aleatoare continuă) Dispersia
  moment_centrat (obiect de tip variabilă aleatoare continuă, numarul de ordine al momentului centrat)
  moment_initial (obiect de tip variabilă aleatoare continuă, numarul de ordine al momentului inițial)
  (Cerința 5)
  Calculul mediei, dispersiei și a momentelor inițiale și centrate până la ordinul 4, dacă acestea există. În caz contrar, este afișat un mesaj corespunzător către utilizator.

- **mean_crv_func** (obiect de tip variabilă aleatoare continuă, functia aplicată)
  var_crv_func (obiect de tip variabilă aleatoare continuă, functia aplicată)
  (Cerința 6)
  Calculul mediei și dispersiei unei variabile aleatoare $g(X)$, unde X are o repartiție continuă cunoscută, iar g este o funcție continuă precizată de utilizator.

- **uniform, exponential, normal, logNormal, weibull, repgamma, student, repbeta, chi, repf**
  (Cerința 8)
  Crearea unor "fișe de sinteză" pentru câteva repartiții cunoscute. Acestea conțin informații (utilizarea, seminificația parametrilor, media, dispersia etc.) despre respectivele repartiții. Utilizatorul alege repartiția despre care dorește să afle informații.

- **generate_crv** (obiect de tip variabilă aleatoare continuă, numarul de valori generate)
  (Cerința 9)
  Generarea unui număr de valor ales de utilizator dintr-o repartiție de variabile aleatoare continue

- **Cov** (obiect de tip variabilă aleatoare continuă, obiect de tip variabilă aleatoare cotinuă, densitatea comună)
  Cor (obiect de tip variabilă aleatoare continuă, obiect de tip variabilă aleatoare cotinuă, densitatea comună)
  (Cerința 10)
  Calculul covarianței și coeficientului de corelație pentru două variabile aleatoare continue

- **marginal_pdf_1** (obiect de tip variabilă aleatoare continuă, obiect de tip variabilă aleatoare continuă, densitatea comună) Densitatea marginală a primei variabile în funcție de a doua
  marginal_pdf_2 (obiect de tip variabilă aleatoare continuă, obiect de tip variabilă aleatoare continuă, densitatea comună) Densitatea marginală a celei de-a doua variabile în funcție de prima
  conditional_density_Y_X (obiect de tip variabilă aleatoare continuă, obiect de tip variabilă aleatoare continuă, densitatea comună) Densitatea lui Y condiționată de X
  conditional_density_X_Y (obiect de tip variabilă aleatoare continuă, obiect de tip variabilă aleatoare continuă, densitatea comună) Densitatea lui X condiționată de Y
  (Cerința 11)
  Construirea densităților marginale și a densităților condiționate a două variabile aleatore continue folosind densitatea lor comună

- "+" "-"
  (Cerința 12)
  Construirea sumei și diferenței a două variabile aleatoare continue independente, folosind formula de convoluție

Aspecte teoretice folosite în rezolvarea problemei care depășesc nivelul cursului

- **Repartițiile: Weibull, Student, Fisher, Lognormală, Beta, Gamma, Chi** (în cadrul cerințelor numărul 4 și 8)
  Repartiția log-normală este o distrubuție de probabilitate a unei variabile aleatoare continue a cărei logaritm are o distrubuție normală. Această se utilizează în diverse domenii, precum: simulări care se comportă ca și repartițiile normale, însă nu pot lua valori negative, durata unui joc de șah urmează o repartiție log-normală. Sursă: Log-normal distribution.

  Repartiția Weibull este determinată de trei parametri: β - parametrul de formă, γ - parametrul de scară, η - parametrul de poziție (sau a poziției în timp). Poate fi transformată ușor în repartiție normală și exponențială, reprezentând un caz mai general pentru acestea. Sursă: Weibull distribution, LEGI DE REPARTIŢIE UTILIZATE ÎN STUDIUL FIABILITĂŢII.

  Repartiția Gamma este o distrubuție ce depinde de 2 parametri. Repartiția exponențială și Chi-squared sunt cazuri particulare ale acesteia. Se aplică în diverse domenii: neuroștiințe, genetică, biologie, oncologie. Sursă: Gamma distribution.

  Repartiția Beta este o familie de distribuții de variabile aleatoare continue, definite pe intervalul [0, 1], parametrizată de două valori pozitive: α și β, care apar ca exponenți în repartiție și controlează forma acesteia. Generalizare cu mai multe variabile se numește *distribuție Dirichlet*. Sursă: Beta distribution.

  Repartiția Student-t se utilizează pentru estimarea mediei unei populații distribuite normal în situații în care deviația standard este

necunoscută și mărimea spațiului probelor este mic. Are un singur parametru - ν, ce reprezintă gradul de libertate al repartiției. Graficul t-distribuției are formă de clopot asemănătoare cu repartiția normală, însă are capetele mai pronunțate, deci e mult mai folositoare pentru modelarea distribuțiilor a căror valori pot cădea mai departe de medie. Sursă: Student's t-distribution.

Repartiția Chi poate reprezenta distribuția distanțelor Euleriene a unei mulțimi de puncte distribuite după repartiția normală. Este legată de repartiția *chi-squared* prin descrierea distribuției radicalilor pozitivi ai unei variabile care are distribuție *chi-squared*. Sursă: Chi distribution

Repartiția Fisher-Snedecor (F) este o distribuție continuă de probabilitate, deseori întâlnită ca distribuția nulă a unui test statistic. Aceasta se mai utilizează și la verificarea ipotezei egalităţii dispersiilor de sondaj, obţinute în două eşantioane independente. Sursă: F-distribution.

- **Simularea variabilelor aleatore continue** (în cadrul cerinței numărul 9)
  Pentru această cerință, am folosit ca sursă de informare materialele suplimentare puse la dispoziție în cadrul laboratorului. Astfel, în rezolvare am folosit metoda inversă care ne spune că pentru o variabilă aleatore continuă X cu funcția de repartiție F, $X = F(U)^{-1}$, unde $U =$ Uniform $(0, 1)$. Algoritmul implemetat constă în aflarea inversei funcției de repartitie F, generarea variabilei uniforme U pe $[0, 1]$ și aplicarea funcției inverse obținute la U.

- **Densitățile condiționate a două variabile aleatoare continue** (în cadrul cerinței numărul 11)
  Densitatea condiționate a variabilei aleatoare continue X, în raport cu variabila aleatoare continuă Y, $h(y|x)$ este egală cu $\frac{f(x,y)}{f_X(x)}$, unde $f(x, y)$ este densitatea comună a lui X și Y, iar $f_X(x)$ este densitatea marginală a lui X.
  Sursa din care am preluat informația este:
  https://online.stat.psu.edu/stat414/lesson/20/20.2.

- **Convoluția a două variabile aleatoare continue** (în cadrul cerinței numărul 12)
  Convoluția a două variabile aleatoare continue reprezintă funcția:
  $$f_Z(z) = \int_{-\infty}^{\infty} f(z-y)g(y)dy = \int_{-\infty}^{\infty} g(z-x)f(x)dx,$$
  unde f(x) și g(y) sunt funcțiile densitate de probabilate corespunzătoare variabilelor aleatoare continue X și Y. Dacă cele două variabile sunt

și independente, atunci convoluția lor reprezintă suma acestora: $Z = X + Y$. Sursă: 7.2: Sums of Continuous Random Variables.

Diferența a două variabile aleatoare continue este dată de formula:

$$f_{X-Y}(z) = \int_{-\infty}^{\infty} f(x)g(x-z)dx.$$

Sursa: Convolution - Difference of two random variables.

Precizări privind pachete software folosite și surse de inspirație

În cadrul proiectului, am folosit **Roxygen2**, un instrument pentru generarea documentației pachetelor din R prin generarea automată a fișierelor .Rd (care conțin documentația propriu-zisă) și a fișierului 'NAMESPACE'. Acesta permite adăugarea documentației înaintea funcțiilor, variabilelor, utilizând taguri speciale și generează automat fișierele cu documentația. De asemena, importă funcții din alte pachete și exportă funcțiile. Importarea pachetelor se poate face atât prin adăugarea denumirii pachetului în fișierul 'DESCRIPTION', secțiunea Imports, cât și cu ajutorul Roxygen2 prin tagurile: @importFrom <nume_pachet> <nume_functie>(pentru a importa o funcție dintr-un pachet specificat) sau @import <nume_pachet>. Sursele din care ne-am inspirat sunt următoarele: Object documentation, Introduction to Roxygen2.

În cadrul cerinței numărul 3, pentru a crea un obiect de tip variabilă aleatoare continuă am construit o clasa "ContRV" ce înglobează funcția densitate de probabilitate, limitele inferioară și superioară. Atunci când dorim să creăm un astfel de obiect, funcția "validity" verifică dacă densitatea de probabilitate este validă.

```
ContRV <- setClass(
  "ContRV",
  slots = list(
    pdf = "function",
    lowerBound = "numeric",
    upperBound = "numeric"),

  validity = function(object) {
    if(check_if_pdf(object@pdf, object@lowerBound,
                    object@upperBound))
      TRUE
```

```
    else {
      stop("Given function is not a valid pdf")
    }
  }
)
```

Sursa de informare a fost codul de mai jos, articolul complet poate fi găsit aici: Objects and Classes in R.

```
setClass("employee", slots=list(name="character",
                    id="numeric", contact="character"))
```

Sursa de inspirație pentru adăugarea validării a fost următorul cod, care poate fi găsit la acest link.

```
setClass(
  "test", slots=c(a="integer"),
   validity=function(object) if(length(object@a) != 1L)
                    "not scalar" else TRUE
)
```

Alte surse de informare au fost: setClass - RDocumentation, OO field guide. Pentru a construi clasa împreună cu metodele sale am importat pachetul **methods**.

În cadrul cerinței numărul 9, pentru a obține funcția inversă celei de repartiție am folosit funcția **inverse** din pachetul **GoFKernel**, sursa de inspirație fiind Inverse CDF Function, R Documentation. Pentru generarea variabilei uniforme pe intervalul $[0, 1]$, am folosit funcția **runif** din pachetul **stats**. Sursa de inspirație este The Uniform Distribution, R Documentation.

În cadrul cerinței numărul 10, pentru a calcula covarianța avem nevoie de calculul integralelor duble. Pentru acest lucru am folosit funcția **integral2** din pachetul **pracma**, despre care am citit aici: integral2, RDocumentation.

Comentarea codului și a soluției prezentate

(Cerința 1)

Pentru prima cerință, trebuia să determinăm constansta de normalizare k a unei funcții sau să afișăm un mesaj corespunzător dacă nu există. Știm că

$$k \cdot \int_{-\infty}^{\infty} f(x)dx = 1, k = \frac{1}{\int_{-\infty}^{\infty} f(x)dx}$$

Dacă valoarea integralei aparține $\mathbf{R}^*$ atunci obținem soluție. În R, într-un bloc tryCatch, calculăm această valoarea cu ajutorul funcției integrate (care returnează eroare pentru valorile $\{-\infty, \infty\}$). Apoi verificăm, cu marja de eroare, dacă valoarea integralei este 0. Altfel, returnăm valoarea lui k. Codul poate fi găsit la secțiunea următoare.

(Cerința 4)

La această cerință, am implementat două funcții pentru afișarea variabilelor aleatoare continue: plot_RV și plot_distribution. Prima este similară cu funcția plot.RV din pachetul discreteRV, studiat la laborator, doar că afișează nu doar funcția densitate de probabilitate, ci și funcția de repartiție a variabilei. A doua funcție este un caz particular al funcției plot_RV, în care utilizatorul poate specifica numele repartiției pe care vrea să o afișeze dintre cele disponibile. Pentru ambele funcții pot fi specificate intervale pentru x și y și pentru prima funcție - culoarea graficului.

(Cerința 5)

Pentru a calcula media unei variabile aleatore continue X știm că

$$E(X) = \int_{-\infty}^{\infty} xf(x)dx,$$

unde $f(x)$ este densitatea de probabilitate. Folisind această formulă am implementat astfel (x este un obiect de tip variabilă aleatoare continuă):

```
setMethod(
  f = "mean",
  signature = c("ContRV"),
  definition = function(x) {
    x_func = function(new_x){
      new_x * x@pdf(new_x)
    }
    return(integrate(x_func, x@lowerBound, x@upperBound) $value)
  }
)
```

Pentru a calcula dispersia unei variabile aleatore continue X am folosit următoarea formulă:

$$Var(X) = \int_{-\infty}^{\infty} (x - E(X))^2 f(x) dx.$$

Pentru a calcula momentul centrat de ordin r unei variabile aleatore continue X am folosit următoarea formulă:

$$\mu_r = \int_{-\infty}^{\infty} (x - E(X))^r f(x) dx.$$

Pentru a calcula momentul inițial de ordin r unei variabile aleatore continue X am folosit următoarea formulă:

$$m_r = \int_{-\infty}^{\infty} x^r f(x) dx.$$

În cadrul funcțiilor verificăm validitatea lui r și dacă momentul există.

(Cerința 6)

Pentru a calcula media unei variabile aleatoare continue $Y = g(X), X$ variabilă aleatoare continuă și g funcție continuă, am folosit formula:

$$E(Y) = \int_{-\infty}^{\infty} g(x) f_X(x) dx.$$

Pentru a calcula dispersia, am folosit formula:

$$Var(Y) = E(Y^2) - E(Y)^2.$$

Implementarea este următoarea:

```
setMethod(
  f = "var_crv_func",
  signature = c("ContRV", "function"),
  definition = function(crv, g) {
    g_2 = function(x){
      g(x) * g(x)
    }
    return(mean_crv_func(crv, g_2)- (mean_crv_func(crv, g))^2)
  }
)
```

(Cerința 8)

În cadrul acestui exercițiu am implementat un set de 10 repartiții de variabile aleatoare continue cunoscute. Funcțiile pentru fiecare din ele se aseamănă cu repartițiile de bază din R, utilizând componentele create în cadrul exercițiilor anterioare. Pentru fiecare distribuție am adăugat câte o pagină de documentație în Roxygen2, afișând pentru fiecare date precum: modul de utilizare, pdf, cdf, media, dispersia, aplicațiile acestora în viața reală. Pentru vizualizarea documentației, în consolă rulează comanda: ?<numele_repartiției>.

(Cerința 10)

Pentru a calcula covarianța a două variabile aleatore continue X și Y, am folosit formula:

$$Cov(X,Y) = \int_c^d \int_a^b xyf(x,y)dxdy - \mu_X\mu_Y,$$

unde $f(x,y)$ este densitatea comună.

Pentru a calcula corelația a două variabile aleatore continue X și Y, am folosit formula:

$$Cor(X,Y) = \frac{Cov(X,Y)}{\sigma_X\sigma_Y} = \frac{Cov(X,Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}}$$

unde $f(x,y)$ este densitatea comună.

(Cerința 11)

Pentru a calcula pmf-urile marginale, pentru densitatea comună continuă $f(x,y)$ cu domeniul $[a,b] \times [c,d]$, am folosit formula:

$$f_X(x) = \int_c^d f(x,y)dy, f_Y(y) = \int_a^b f(x,y)dx$$

.

(Cerința 12)

Pentru acest punct am implementat două metode pentru clasa ContRV, supraîncărcând operatorii '+' și '-'. Metodele utilizează în implementare convoluția dintre cele două variabile, reprezentând suma lor, dacă acestea sunt independente și returnează în final o variabilă aleatoare continuă, ce are pdf-ul convoluția lor.

Dificultăți și soluții abordate

În această documentație, la secțiunea "Aspecte teoretice folosite în rezolvarea problemei care depășesc nivelul cursului", am descris câteva dificultăți tehnice întâmpinate, împreună cu informațiile folosite pentru a le depăși precizând și sursa acestora. În această secțiune, voi descrie dificultățile de implementare întâmpinate, cât și soluțiile găsite.

O problemă pe care am întâmpinat-o în cadrul proiectului este tratarea erorilor. Acest link ne-a fost de ajutor. Un exemplu pentru modul în care am rezolvat această problemă, din cerința 1, este următorul:

```
normalizing_constant <- function(func, lowerBound, upperBound) {
  k_norm <- tryCatch({
    value <- integrate(Vectorize(func), lowerBound,
                       upperBound)$value
    if (abs(value) <= EPSILON) {
      stop("valoarea integralei este 0\n")
    }
    return(1/value)
  },
  error = function(err_msg) {
message("Eroare: Constanta de normalizare nu poate fi calculata")
message(err_msg)
  }
  )
}
```

O altă problemă pe care am întâmpinat-o sunt rezultatele inexacte. Pentru a rezolva acest aspect, am folosit și o marjă de eroare EPSILON = 10^{-7}.

La cerința numărul 2, verificăm dacă o funcție este densitate de probabilitate. Funcția trebuie să fie pozitivă pe întreg intervalul cuprins între capetele primite ca parametru. Cu alte cuvinte, valoarea minima a functiei pe acest interval trebuie sa fie >= 0. De asemenea, integrala funcției de la primul la al doilea capăt trebuie să fie egală cu 1. Pentru a afla valoarea minimă a unei funcții am folosit *optimize*. Unele probleme puteau apărea la valorile capetelor intervalului în care determinam valoarea minimă a funcției. Inițial,

am ales ca aceastea să fie limita numerică admisibilă din limbajul R, dar în anumite cazuri (când funcția avea gradul 2 sau mai mare, era exponențială, etc.) obțineam erori, pentru că nu puteau fi calculate unele valori din acel interval. Din acest motiv, am restrâns întervalul la $[-10^{20}, 10^{20}]$, pe care l-am considerat un interval numeric suficient pentru majoritatea funcțiilor pe care le-ar folosi utilizatorul.

O altă dificultate întâmpinată a fost lucrul cu funcțiile constante, întrucât la apelarea funcției constante pe un vector de valori, nu obțineam același număr de valori constante. Din cauza aceasta, apăreau diverse erori la determinarea integralei unei funcții constante, afișarea graficului. Pentru a soluționa această problemă, am folosit funcția predefinită *Vectorize*, care primește ca parametru o funcție și întoarce aceeași funcție, dar, cea din urmă atunci când este apelată cu un vector de valori, returnează de asemenea un vector de valori. Sursa de inspirație poate fi găsită aici.

Le cerința numărul 4, funcția plot_distribution(primește ca argument denumirea distribuției, intervalele de valori pentru x și y și parametrii pentru distribuțiile respective) trebuie să primească un număr variabil de argumente. Acesta diferă în funcție de repartiția pe care dorim să o reprezentăm grafic. Pentru a rezolva această problemă am folosit operatorul '...', ce reprezintă număr variabil de argumente (similar cu *args din Python). Astfel, pentru fiecare distribuție specificată trebuia să introducem numărul respectiv de parametri, în caz contrar, obținem o eroare.

Le cerința numărul 11, pentru a calcula densitățile marginale a două variabile aleatoare continue, am avut nevoie sa integrăm parțial. Pentru acest calcul am folosit funcțiile sapply și integrate astfel:

```
setMethod(
  f = "marginal_pdf_1",
  signature = c("ContRV", "ContRV", "function"),
  definition = function(crv1, crv2, dens) {
    f <- function(x) {
      sapply(x, function(x) {
        integrate(function(y) dens(x, y), crv2@lowerBound,
                  crv2@upperBound)$value
      })}
  }
)
```

Acest cod a fost inspirat de Calculate a double integral.

Concluzii

În concluzie, sperăm că acest pachet este un instrument util pentru lucrul cu variabile aleatoare continue. În cadrul acestui proiect, am avut ocazia să aplicăm și să aprofundăm cunoștințele dobândite la laborator și la curs. Mare parte din informațiile teoretice au fost extrase din cursurile 4-8 și materialele de la laborator. Pentru implementare, ne-am folosit de codurile sursă scrise la laborator și de link-urile ce pot fi găsite în documentație.