PROIECT

PROBABILITĂȚI ȘI STATISTICĂ

PARTEA II

Studenți:

Gheorghe Andrei-Bogdan

Marian Rareș-Ștefan

Tudorache Andrei-Theodor

Boț George

**Problema I**

set.seed(6)

N <- 10^6

#a)

f\_a <- function(x) {

  1/((x^5\*(1-x))^(1/6))

}

x\_a <- runif(N, 0, 1)

valori\_a <- f\_a(x\_a)

integrala\_a <- mean(valori\_a)

val\_teoretica\_a <- integrate(f\_a, 0, 1)

print(integrala\_a)

print(val\_teoretica\_a)

#b)

f\_b <- function(x) {

  sin(x)^3\*cos(x)^5

}

x\_b <- runif(N, 0, pi/2)

valori\_b <- f\_b(x\_b)

integrala\_b <- mean(valori\_b)\*(pi/2)

val\_teoretica\_b <- integrate(f\_b, 0, pi/2)

print(integrala\_b)

print(val\_teoretica\_b)

#c)

f\_c <- function(x) {

  x^2/(1+x^4)

}

transformare <- function(t) {

  t/(1-t)

}

jacobian <- function(t) {

  1/(1-t)^2

}

x\_c <- runif(N, 0, 1)

x\_c2 <- transformare(x\_c)

valori\_c <- f\_c(x\_c2)\*jacobian(x\_c)

integrala\_c <- mean(valori\_c)

val\_teoretica\_c <- integrate(f\_c, 0, Inf)

print(integrala\_c)

print(val\_teoretica\_c)

#d)

f\_d <- function(x) {

  exp(-x^2+2\*x-4)

}

x\_d <- runif(N, 0, 1)

x\_d2 <- 1/x\_d

jacobian\_d <- 1/x\_d^2

valori\_d <- f\_d(x\_d2)\*jacobian\_d

integrala\_d <- mean(valori\_d)

val\_teoretica\_d <- integrate(f\_d, 0, Inf)

print(integrala\_d)

print(val\_teoretica\_d)

#e)

f\_e <- function(x, y) {

  exp(sqrt(x^2+y^2))

}

regiune\_e <- function(x, y) {

  return(x^2+y^2>=1 & x^2+y^2<=4 & x>=0 & x<=y)

}

x\_e <- runif(N, -2, 2)

y\_e <- runif(N, 0, 2)

regiune <- regiune\_e(x\_e, y\_e)

valori\_e <- f\_e(x\_e[regiune], y\_e[regiune])

integrala\_e <- mean(valori\_e)\*4\*pi

print(integrala\_e)

#f)

f\_f <- function(x, y) {

  sqrt(x^2+y^2)

}

regiune\_f <- function(x, y) {

  return(x^2+y^2>=4 & x^2+y^2<=9 & y>=0)

}

x\_f <- runif(N, -3, 3)

y\_f <- runif(N, 0, 3)

regiune <- regiune\_f(x\_f, y\_f)

valori\_f <- f\_f(x\_f[regiune], y\_f[regiune])

integrala\_f <- mean(valori\_f)\*5\*pi

print(integrala\_f)

#g)

f\_g <- function(x, y) {

  exp(-x-y)

}

x\_g <- runif(N, 0, pi/2)

y\_g <- runif(N, 0, pi/2)

valori\_g <- f\_g(x\_g, y\_g)

integrala\_g <- mean(valori\_g)\*(pi/2)^2

print(integrala\_g)

#h)

U <- runif(N, 0, 1)

x\_h <- U

y\_h <- sqrt(1-U^2)

corelatie <- cor(x\_h, y\_h)

print(corelatie)

#i)

x\_i <- U^2

y\_i <- sqrt(1-U^2)

corelatie <- cor(x\_i, y\_i)

print(corelatie)

**Rezultate:**

**a)**

> print(integrala\_a)

[1] 6.276871

> print(val\_teoretica\_a)

6.283185 with absolute error < 1.2e-05

**b)**

> print(integrala\_b)

[1] 0.04168018

> print(val\_teoretica\_b)

0.04166667 with absolute error < 2.9e-15

**c)**

> print(integrala\_c)

[1] 1.109151

> print(val\_teoretica\_c)

1.110721 with absolute error < 1.6e-05

**d)**

> print(integrala\_d)

[1] 0.04405332

> print(val\_teoretica\_d)

0.08130482 with absolute error < 3e-05

**e)**

> print(integrala\_e)

[1] 61.84328

**f)**

> print(integrala\_f)

[1] 39.78935

**g)**

> print(integrala\_g)

[1] 0.6274565

**h)**

> print(corelatie)

[1] -0.9211792

**i)**

> print(corelatie)

[1] -0.9835109

**Comentarii:**

Pentru subpunctele unde am folosit metoda de simulare Monte Carlo, dar am calculat și valoarea teoretică, am observat că valorile sunt foarte apropiate, eroarea fiind nesemnificativă.

**Dificultăți:**

Am avut dificultăți în a calcula valoarea teoretică pentru subpunctele e) -> i).

**Problema II**

#a)

esantion\_a <- c(8, 12, 6, 14, 9, 12, 15, 7, 10, 10, 14, 9, 12, 15, 11, 8, 6,

              9, 13, 12, 10, 8, 11, 13, 15, 13, 15, 10, 8, 7, 13, 8, 9, 11,

              14, 7, 12, 16, 15, 10, 9, 10, 12, 8, 13, 9, 11)

#Estimatorul metodei verosimilitatii maxime

teta\_ver\_a <- mean(esantion\_a)/2

# Estimatorul metodei momentelor

teta\_mom\_a <- mean(esantion\_a)/2

print(teta\_mom\_a)

print(teta\_ver\_a)

#b)

esantion\_b <- c(3, 2, 1, 4, 2, 3, 4, 1, 3, 2, 2, 4, 2, 1, 7, 5, 4, 5, 5, 2,

              3, 4, 3, 1, 2, 4, 1, 1, 2, 3, 1, 3, 1, 4, 1, 3, 1, 6, 1, 3,

              3, 4, 3, 1, 3, 2, 2, 3, 2, 4, 1, 1, 2, 6, 3, 1, 3, 6, 1, 2,

              3, 6, 3, 2, 2, 2, 4, 2, 1, 3, 3, 4, 2, 3, 4, 1, 4, 4, 6, 3,

              3, 5, 2, 2, 2, 3, 1, 3, 1, 3, 3, 5, 3, 4, 3, 2, 4, 2, 3, 3)

#Estimatorul metodei verosimilitatii maxime

teta\_ver\_b <- mean(esantion\_b)

print(teta\_ver\_b)

#d)

esantion\_d <- c(6, 3, 24, 24, 4, 56, 10, 13, 2, 28, 24, 2, 22, 11, 2, 8, 118,

                2, 14, 19, 7, 9, 8, 189, 2, 9, 21, 6, 6, 2, 3, 2, 3, 18, 3, 2,

                21, 1, 5, 9, 11, 13, 19, 76, 1, 5, 9, 4, 57, 1, 2, 16, 5, 2, 20,

                8, 1, 40, 6, 4, 19, 6, 3, 2, 4, 9, 1, 5, 10, 12, 6, 525, 19, 6,

                17, 2, 5, 159, 5, 62, 6, 3, 45, 21, 23, 3, 17, 2, 1, 1, 474, 15,

                3, 3, 7, 7, 13, 4, 38, 4)

#Estimatorul metodei verosimilitatii maxime

teta\_ver\_d <- mean(esantion\_d)

# Estimatorul metodei momentelor

teta\_mom\_d <- mean(esantion\_d)

print(teta\_mom\_d)

print(teta\_ver\_d)

#e)

esantion\_e <- c(3.5930579, 2.1027540, 1.7820777, 9.6550388, 6.8803846, 0.7388358, 2.9194654, 3.1178660,

                1.2323236, 2.9776820, 1.1172078, 2.4184586, 3.3258971, 1.9498871, 2.6088612, 3.9535062,

                3.0389107, 4.4226628, 3.9366318, 2.4551569, 5.2814487, 5.6778622, 4.7683935, 1.1581498,

                3.1270783, 4.1473311, 7.4830426, 1.1342893, 1.7773392, 7.7510826, 1.3919927, 2.3613291,

                2.6234826, 1.6562602, 1.4992235, 2.3455062, 3.8458809, 5.8333841, 3.3834034, 1.5202546,

                3.1248186, 5.3029567, 3.6225571, 4.8309931, 3.1579595, 3.2640258, 3.9538891, 4.0796841,

                4.0991772, 3.2779944, 2.5002127, 3.0654695, 1.6996010, 3.2175175, 1.9033087, 4.4052061,

                2.3158379, 2.4778345, 5.4382190, 4.9141207, 6.0978745, 1.1428936, 3.5639106, 7.4541937,

                7.7778289, 3.2859563, 0.7432908, 1.4442696, 3.6619932, 2.8361371, 4.3180773, 1.6763585,

                4.4464154, 2.5049617, 0.4448735, 5.0518839, 3.4151834, 1.6823650, 5.4517583, 2.8212788,

                2.1566837, 2.9893287, 1.6925123, 6.5197938, 4.2165408, 1.6728425, 2.7650830, 2.6742755,

                2.9622047, 0.7809781, 1.3913415, 5.3430751, 2.4859925, 3.7329465, 6.3129236, 0.6635228,

                3.7640343, 2.1850174, 4.3773328, 5.0931544)

#Estimatorul metodei verosimilitatii maxime

teta\_ver\_e <- mean(esantion\_e)

# Estimatorul metodei momentelor

teta\_mom\_e <- (gamma(4)/mean(esantion\_e))^(1/4)

print(teta\_mom\_e)

print(teta\_ver\_e)

**Rezultate:**

**a)**

> print(teta\_mom\_a)

[1] 5.414894

> print(teta\_ver\_a)

[1] 5.414894

**b)**

> print(teta\_ver\_b)

[1] 2.82

**d)**

> print(teta\_mom\_d)

[1] 25.85

> print(teta\_ver\_d)

[1] 25.85

**e)**

> print(teta\_mom\_e)

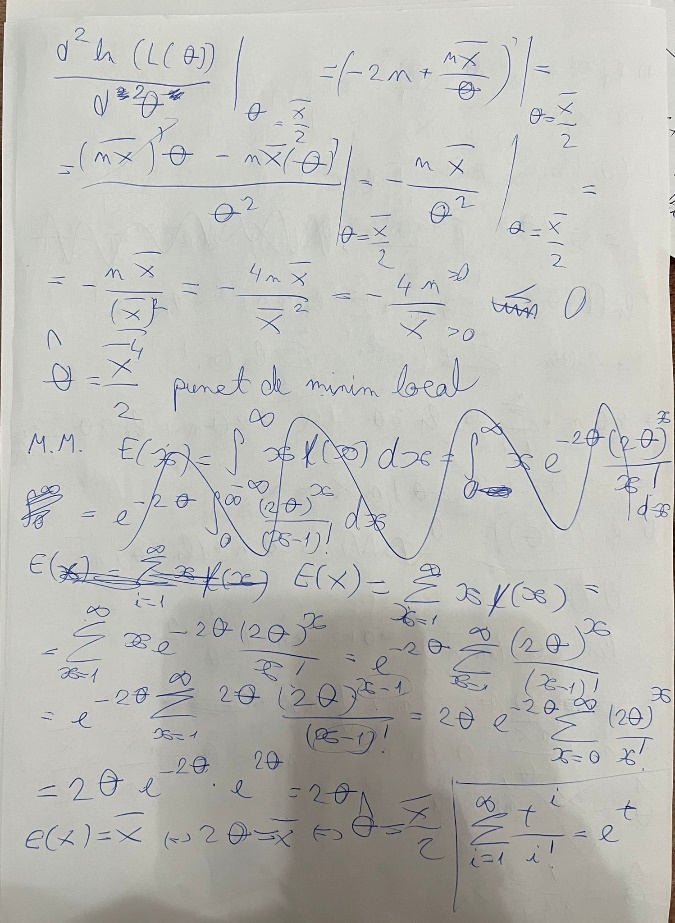
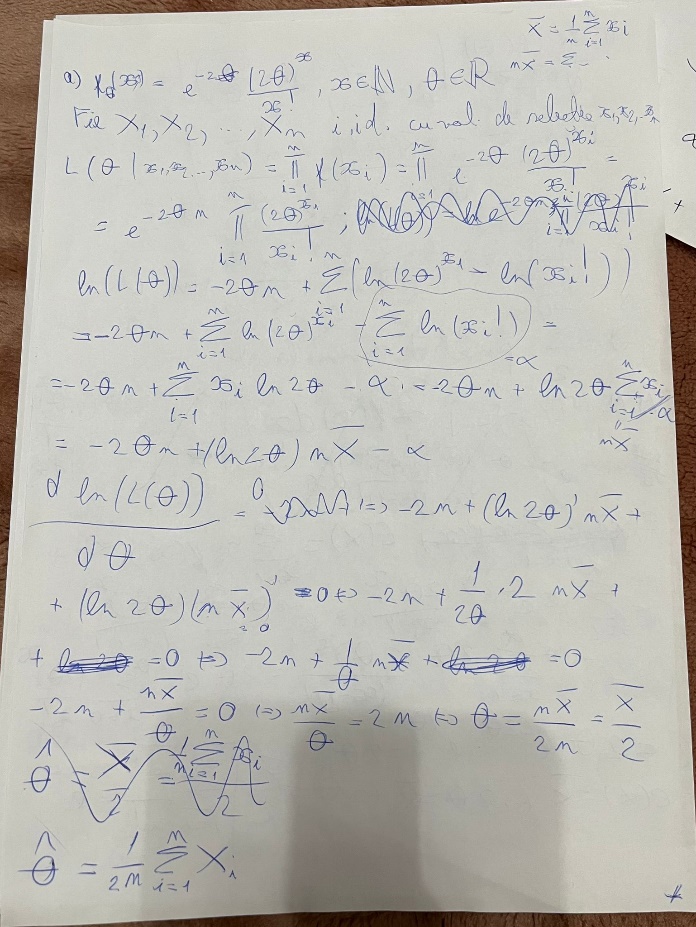
[1] 1.154918

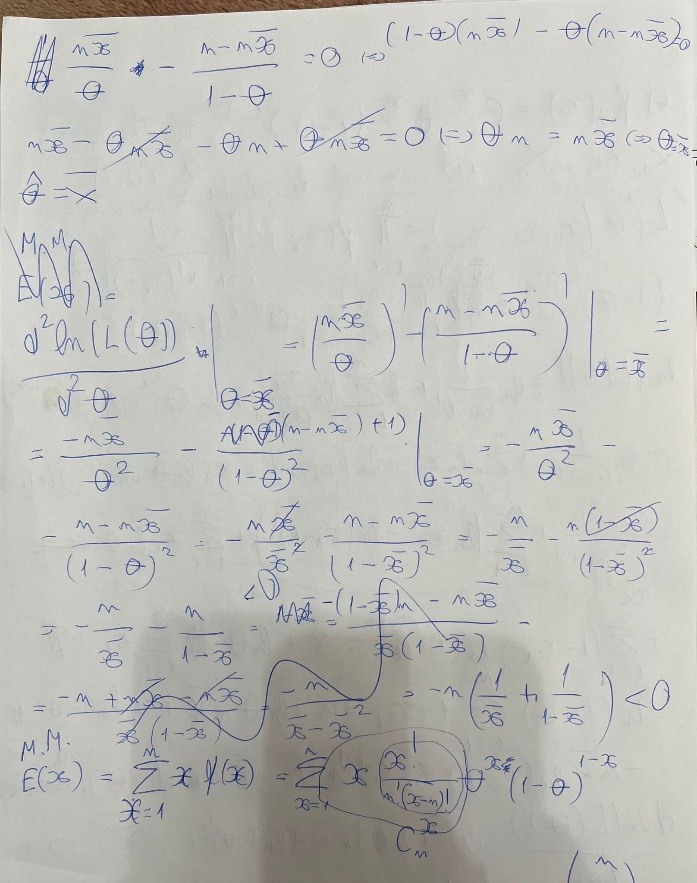
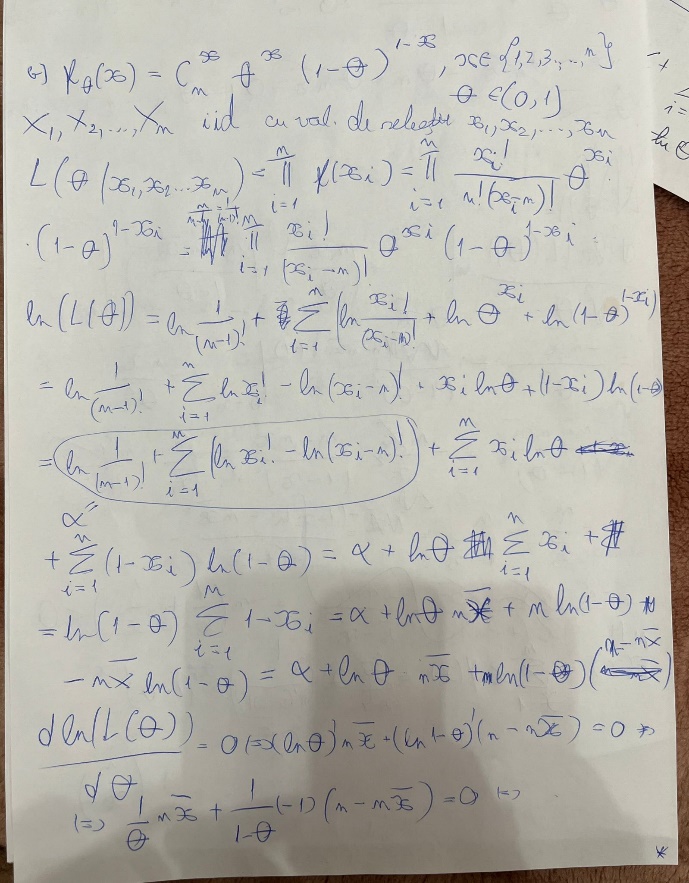
> print(teta\_ver\_e)

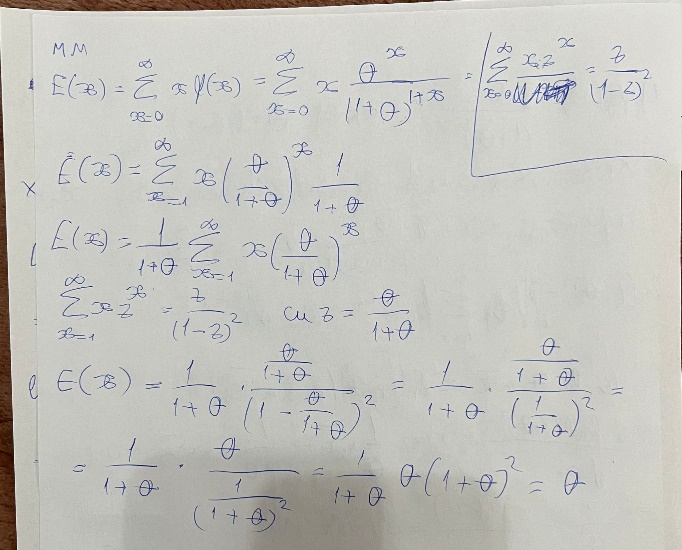
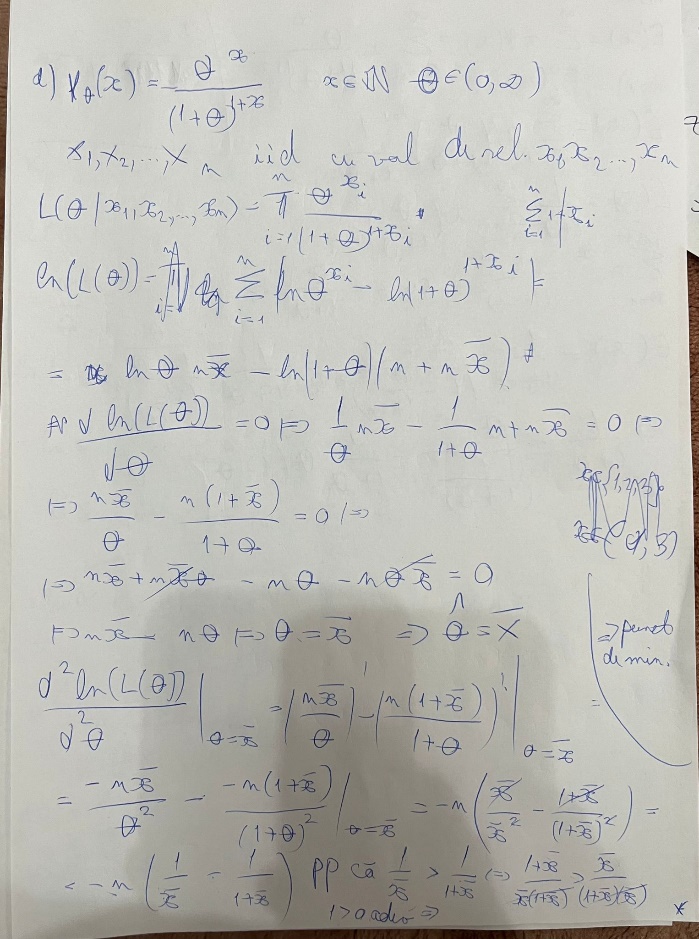
[1] 3.372459

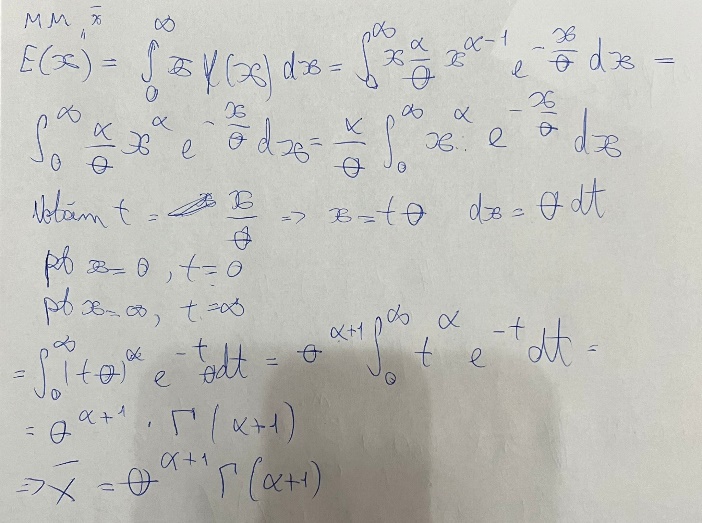
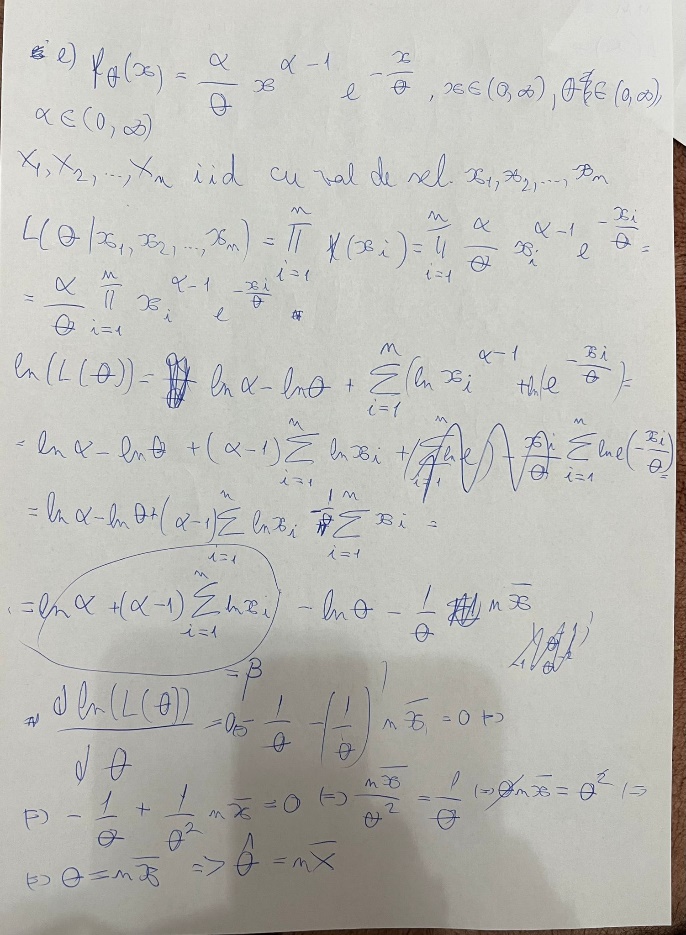
**Comentarii:**

Aici sunt poze cu calculele de la fiecare subpunct, făcute pe foaie:









**Dificultăți:**

Am întâmpinat dificultăți la calculele subpunctului c) și la Metoda Momentelor de la subpunctul b).

**Problema III**

#a)

# Functie pentru calculul constantei de normalizare

calc\_const <- function() {

  int <- integrate(function(x) x^4,0,2)$value

  c <- 1/int

  return(c)

}

# Functie pentru calculul mediei

calc\_media <- function(c) {

  media <- integrate(function(x) c\*x^5,0,2)$value

  return(media)

}

# Functie pentru calculul variantei

calc\_var <- function(c, media) {

  media\_sq <- integrate(function(x) c\*x^6,0,2)$value

  var <- media\_sq - media^2

  return(var)

}

c <- calc\_c()

print(c)

media <- calc\_med(c)

print(media)

var <- calc\_var(c, media)

print(var)

#b)

# Functie pentru calculul constantei de normalizare

calc\_const <- function() {

  a <- 1

  b <- 6\*(1-a/2)/3

  return(c(a,b))

}

# Functie pentru calculul mediei

calc\_media <- function(a, b) {

  media <- integrate(function(x) x\*(a\*x+b\*x^2),0,1)$value

  return(media)

}

# Functie pentru calculul variantei

calc\_var <- function(a, b, media) {

  media\_sq <- integrate(function(x) x^2\*(a\*x+b\*x^2),0,1)$value

  var <- media\_sq - media^2

  return(var)

}

const <- calc\_const()

a <- const[1]

b <- const[2]

print(a,b)

media <- calc\_media(a, b)

print(media)

var <- calc\_var(a, b, media)

print(var)

#c)

# Functie pentru calculul mediei

calc\_media <- function() {

  media <- sum(sapply(1:1000, function(x) x\*4/(x\*(x+1)\*(x+2))))

  return(media)

}

# Functie pentru calculul variantei

calc\_var <- function(media) {

  media\_sq <- sum(sapply(1:1000, function(x) x^2\*4/(x\*(x+1)\*(x+2))))

  var <- media\_sq - media^2

  return(var)

}

media <- calc\_media()

print(media)

var <- calc\_var(media)

print(var)

#d)

# Functie pentru calculul densitatilor de probabilitate

calc\_dens <- function(x) {

  return(log10(x/(x+1)))

}

# Functie pentru calculul mediei

calc\_media <- function() {

  x\_values <- 1:9

  densitati <- sapply(x\_values, calc\_dens)

  media <- sum(x\_values \* densitati)

  return(media)

}

# Functie pentru calculul variantei

calc\_var <- function(media) {

  x\_values <- 1:9

  densitati <- sapply(x\_values, calc\_dens)

  media\_sq <- sum((x\_values^2) \* densitati)

  var <- media\_sq - media^2

  return(var)

}

media <- calc\_media()

print(media)

var <- calc\_var(media)

print(var)

#e)

# Functie pentru calculul densitatilor de probabilitate

calc\_dens <- function(x, teta) {

  if (x > 0) {

    return((teta^2 / (1 + teta)) \* (1 + x) \* exp(-teta \* x))

  } else {

    return(0)

  }

}

# Functie pentru calculul mediei

calc\_media <- function(teta) {

  x\_values <- seq(0, 100, by=0.01)

  densitati <- sapply(x\_values, calc\_dens, teta)

  media <- sum(x\_values \* densitati) \* 0.01

  return(media)

}

# Functie pentru calculul variantei

calc\_var <- function(teta, media) {

  x\_values <- seq(0, 100, by=0.01)

  densitati <- sapply(x\_values, calc\_dens, teta)

  media\_sq <- sum((x\_values^2) \* densitati) \* 0.01

  var <- media\_sq - media^2

  return(var)

}

teta <- 2 #exemplu

media <- calc\_media(teta)

print(media)

var <- calc\_var(teta, media)

print(var)

#f)

# Functie pentru calculul densitatilor de probabilitate

calc\_dens <- function(x) {

  if (x < 0) {

    return((1/3) \* exp(x))

  } else if (0 <= x && x < 1) {

    return(1/3)

  } else if (x >= 1) {

    return((1/3) \* exp(-(x-1)))

  } else {

    return(0)

  }

}

# Functie pentru calculul mediei

calc\_media <- function() {

  x\_values <- seq(-10, 10, by=0.01)

  densitati <- sapply(x\_values, calc\_dens)

  media <- sum(x\_values \* densitati) \* 0.01

  return(media)

}

# Functie pentru calculul variantei

calc\_var <- function(media) {

  x\_values <- seq(-10, 10, by=0.01)

  densitati <- sapply(x\_values, calc\_dens)

  media\_sq <- sum((x\_values^2) \* densitati) \* 0.01

  var <- media\_sq - media^2

  return(var)

}

media <- calc\_media()

print(media)

var <- calc\_var(media)

print(var)

#g)

# Functie pentru calculul densitatilor de probabilitate

calc\_dens <- function(x) {

  return(1 / (pi \* (1 + x^2)))

}

# Functie pentru calculul mediei

calc\_media <- function() {

  x\_values <- seq(-1000, 1000, by=0.01)

  densitati <- sapply(x\_values, calc\_dens)

  media <- sum(x\_values \* densitati) \* 0.01

  return(media)

}

# Functie pentru calculul variantei

calc\_var <- function(media) {

  x\_values <- seq(-1000, 1000, by=0.01)

  densitati <- sapply(x\_values, calc\_dens)

  media\_sq <- sum((x\_values^2) \* densitati) \* 0.01

  var <- media\_sq - media^2

  return(var)

}

media <- calc\_media()

print(media)

var <- calc\_var(media)

print(var)

**Rezultate:**

**a)**

> c <- calc\_const()

> print(c)

[1] 0.15625

>

> media <- calc\_media(c)

> print(media)

[1] 1.666667

>

> var <- calc\_var(c, media)

> print(var)

[1] 0.07936508

**b)**

> const <- calc\_const()

> a <- const[1]

> b <- const[2]

> print(a,b)

[1] 1

>

> media <- calc\_media(a, b)

> print(media)

[1] 0.5833333

>

> var <- calc\_var(a, b, media)

> print(var)

[1] 0.1097222

**c)**

> media <- calc\_media()

> print(media)

[1] 1.996008

>

> var <- calc\_var(media)

> print(var)

[1] 17.96982

**d)**

> media <- calc\_media()

> print(media)

[1] -3.440237

>

> var <- calc\_var(media)

> print(var)

[1] -29.72697

**e)**

> media <- calc\_media(teta)

> print(media)

[1] 0.6666556

>

> var <- calc\_var(teta, media)

> print(var)

[1] 0.3889037

**f)**

> media <- calc\_media()

> print(media)

[1] 0.499718

>

> var <- calc\_var(media)

> print(var)

[1] 2.187892

**g)**

> media <- calc\_media()

> print(media)

[1] 1.371115e-16

>

> var <- calc\_var(media)

> print(var)

[1] 635.6236

**Comentarii:**

La subpunctul e) am folosit teta=2 ca exemplu.

La subpunctul c) am limitat suma la primii 1000 de termeni pentru a evita eventualele probleme de convergență.

Pentru subpunctele d) -> g), x\_values reprezintă un vector de valori cu pasul 0.01, care este folosit pentru a ne apropia de integrarea numerică.