Documentatia proiectului pentru probabilitati si statistica

Grupa 232

Anul universitar 2024-2025

1. **Telu Mihai-Sebastian (lider)**
2. **Fugulin Victor**
3. **Sarighioleanu Sebastian Laurentiu**
4. **Ungureanu Matei-Stefan**

Cerinta 1: Simularea și analiza timpului total T pentru finalizarea unei activități

1 Introducere

Acest document descrie implementarea unui algoritm care simulează și analizează timpul total T petrecut de o persoană A într-o activitate compusă din n etape. Fiecare etapă este caracterizată printr-un timp de execuție modelat ca o variabilă aleatoare exponențială și printr-o probabilitate de a continua către etapa următoare.

2 Modelul matematic

Pentru fiecare etapă i, timpul de finalizare este o variabilă aleatoare exponențială:

$$T_i \sim \mathrm{Exp}(\lambda_i)$$

unde λ_i este rata exponențială pentru etapa i.

După finalizarea etapei i, persoana A decide să continue cu probabilitatea α_i sau să se oprească cu probabilitatea $1 - \alpha_i$.

Timpul total T este suma timpilor petrecuți în etapele în care persoana A a participat efectiv:

$$T = \sum_{i=1}^{k} T_i, \quad \text{unde } k \text{ este ultima etapă atinsă}$$

3 Implementare și simulare

Se generează 10^6 valori pentru T prin simulare. Fiecare simulare se desfășoară astfel:

- Se începe de la etapa $i = 1$.

- Se generează un timp $T_i \sim \text{Exp}(\lambda_i)$ pentru etapa curentă.
- Cu probabilitatea α_i, se continuă la etapa următoare.
- Dacă persoana se oprește, simularea se termină și se salvează T.

Cod R pentru simulare:

```
simulate_T <- function() {
  T_total <- 0
  i <- 1
  while (i <= n) {
    T_total <- T_total + rexp(1, lambda[i])
    if (runif(1) > alpha[i]) return(list(T_total, FALSE, i))
    i <- i + 1
  }
  return(list(T_total, TRUE, n))
}
```

Extragem informatiile din simulare in urmatorul mod:

```
n_sim <- 10^6
sim_results <- replicate(n_sim, simulate_T(), simplify = FALSE)
t_values <- sapply(sim_results, function(x) x[[1]])
completion_flags <- sapply(sim_results, function(x) x[[2]])
stopping_stages <- sapply(sim_results, function(x) x[[3]])

- sim_results este o lista cu n_sim elemente, fiecare element fiind rezultatul
unei simulari, adica o lista cu 3 componente: timpul total T, un flag de
finalizare si etapa unde s-a oprit activitatea.

- t_values este un vector numeric cu n_sim elemente care contin valorile
simulate ale lui T.

- completion_flags este un vector de valori TRUE/FALSE cu lungimea n_sim.
TRUE atunci cand activitatea s-a terminat fara sa fie intrerupta,
FALSE atunci cand activitatea a fost intrerupta.

- stopping_stages este un vector numeric cu valori intre 1 si n, care retine
etapa in care s-a incheiat fiecare simulare.
```

Reprezentare grafică a distribuției lui T:

```
hist(t_values, breaks = 100, probability = TRUE, col = "skyblue")
lines(density(t_values), col = "red", lwd = 2)
```

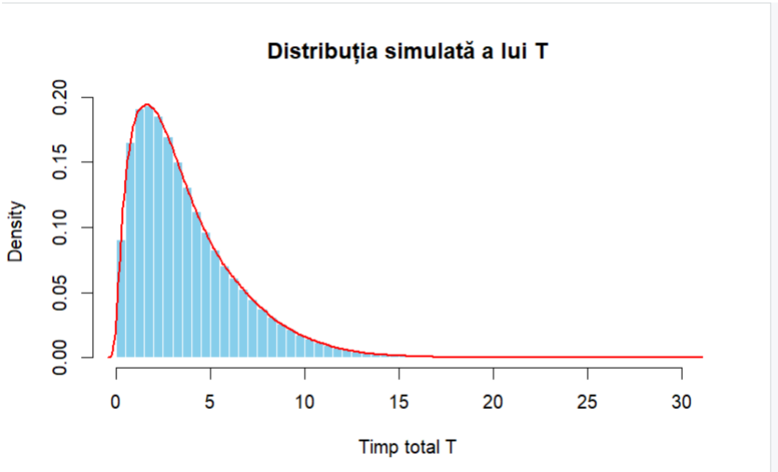


Figure 1: Reprezentarea grafica in R

4 Analiza repartitiei lui T

1. Forma distribuției lui T

Histograma și densitatea estimată arată că distribuția lui T este asimetrică spre dreapta. Acest lucru indică faptul că valorile mici ale lui T sunt mai frecvente, iar valorile mari apar mai rar, un comportament tipic pentru sume de variabile exponențiale trunchiate.

2. Legătura cu distribuția exponențială și gamma

Fiecare etapă are un timp exponențial $T_i \sim \text{Exp}(\lambda_i)$. Dacă procesul ar ajunge întotdeauna la etapa n, atunci suma

$$T = \sum_{i=1}^{n} T_i$$

ar urma aproximativ o distribuție gamma:

$$T \sim \text{Gamma}(k, \theta), \quad \text{unde } k = n, \quad \theta = \frac{1}{\lambda}.$$

Totuși, din cauza opririlor premature cu probabilitate $1 - \alpha_i$, distribuția lui T devine o sumă trunchiată de variabile exponențiale.

3. Confirmare prin simulare

Rezultatele simulării arată că distribuția lui T are o curbă descrescătoare, ceea ce confirmă un comportament asemănător unei variabile gamma trunchiate. Dacă ratele λ_i sunt mari, activitatea se finalizează rapid. Dacă probabilitățile α_i sunt mari, sunt mai multe șanse să se ajungă la etapa finală, ceea ce face ca distribuția să fie mai apropiată de gamma completă.

4. Concluzie

Distribuția lui T este asimetrică spre dreapta și poate fi aproximată printr-o distribuție gamma trunchiată. Forma exactă depinde de parametrii λ_i și α_i, care influențează cât de departe ajunge procesul înainte de oprire.

5 Calculul exact al lui $E(T)$

Matematic, valoarea așteptată a lui T se calculează astfel:

$$E(T) = \sum_{i=1}^{n} \left(\frac{1}{\lambda_i} \prod_{j=1}^{i-1} \alpha_j \right)$$

Cod R pentru calculul exact:

```
E_T_exact <- sum(sapply(1:n, function(i) {
  (1 / lambda[i]) * prod(alpha[1:(i-1)])
}))
```

E_T	3.69975479832001
E_T_exact	3.67599147379786

Figure 2: Observam ca media aproximata si media exacta sunt foarte apropiate. Asta se datoreaza numarului mare de simulari, mai exact 1000000.

Daca numarul de simulari era mai mic, observam o diferenta mai mare intre media aproximata si media exacta.

6 Probabilitatea de finalizare

Probabilitatea ca persoana A să finalizeze întreaga activitate este estimată prin:

$$P(\text{finalizare}) \approx \frac{\text{numărul de simulări care ajung la etapa } n}{\text{numărul total de simulări}}$$

Cod R:

```
P_completion <- mean(completion_flags)
```

7 Probabilitatea ca $T \leq \sigma$

Se estimează probabilitatea ca timpul total să fie sub o valoare σ:

$$P(T \leq \sigma) \approx \frac{\text{numărul de simulări cu } T \leq \sigma}{\text{numărul total de simulări}}$$

Cod R:

```
P_T_leq_sigma <- mean(t_values <= sigma)
```

8 Timp minim și maxim de finalizare

Se determină timpul minim și maxim doar pentru simulările unde activitatea s-a finalizat.

Cod R:

```
t_values_completed <- t_values[completion_flags]
T_min <- min(t_values_completed)
T_max <- max(t_values_completed)
```

Graficul distribuției timpilor de finalizare:

```
hist(t_values_completed, breaks = 100, probability = TRUE, col = "lightgreen")
```

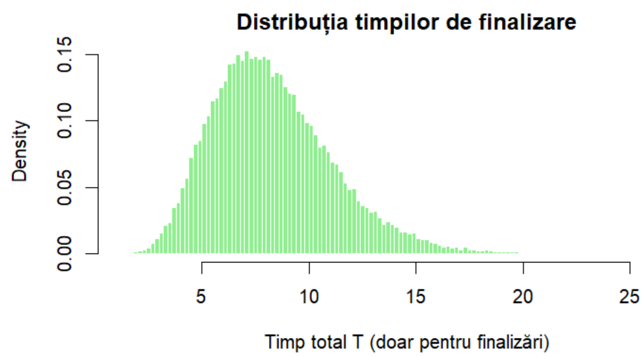


Figure 3: Majoritatea valorilor sunt incadrate intre 5 si 10, unde se observa ca graficul este mai dens.

9 Probabilitatea de oprire înainte de etapa k

Probabilitatea de oprire înainte de etapa k se calculează astfel:

$$P(\text{oprire înainte de } k) \approx \frac{\text{numărul de simulări oprite la } i < k}{\text{numărul total de simulări}}$$

Cod R:

```
stop_probabilities <- sapply(2:n, function(k) mean(stopping_stages < k))
```

Reprezentarea grafică:

```
plot(2:n, stop_probabilities, type = "b", col = "blue", pch = 19, lwd = 2,
     main = "Probabilitatea de oprire înainte de etapa k",
     xlab = "Etapa k", ylab = "Probabilitate de oprire",
     ylim = c(0, 1))
grid()
```

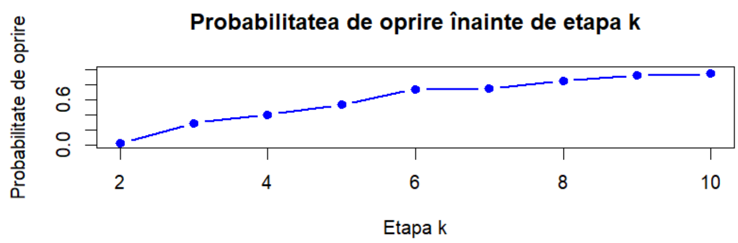


Figure 4: Se observa ca probabilitatea ca activitatea sa se opreasca creste cu cat persoana parcurge din ce in ce mai multe etape.

Afisare probabilitati:

```
cat("Probabilități de oprire înainte de fiecare k:\n")
print(data.frame(k = 2:n, P_stop = stop_probabilities))
```

	k	P_stop
1	2	0.021595
2	3	0.288983
3	4	0.403263
4	5	0.531367
5	6	0.741507
6	7	0.754538
7	8	0.847070
8	9	0.920325
9	10	0.947031

Figure 5: Cand k se apropie de n = 10 in aplicatia noastra, probabilitatea ca activitatea sa se opreasca se apropie de 1. Daca n ar fi mai mare, probabilitatea s-ar apropia si mai mult de 1.

10 Concluzii

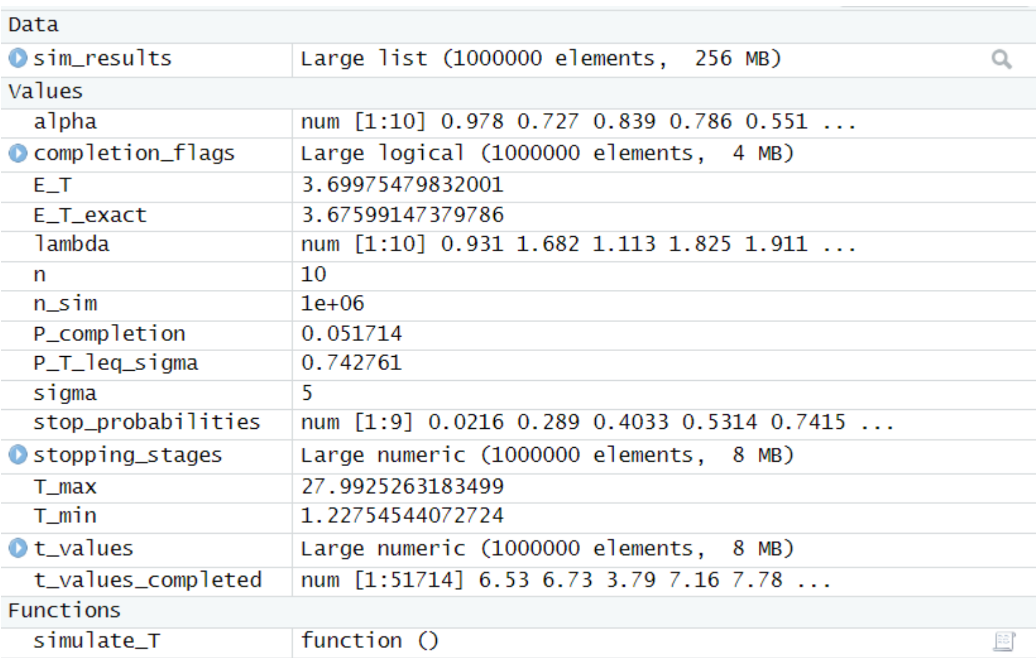
1. Distribuția lui T pare să urmeze o distribuție asemănătoare exponențialei, dar este influențată de opririle aleatorii.
2. Estimarea lui $E(T)$ prin simulare este foarte apropiată de valoarea exactă.
3. Probabilitatea de finalizare poate fi mică dacă α_i sunt scăzute.
4. Timpul minim și maxim de finalizare arată o variabilitate semnificativă.
5. Probabilitatea de oprire înainte de k scade treptat, indicând că cei care ajung în etapele superioare sunt mai predispuși să finalizeze activitatea.

11 Cod cu comentarii + Valorile variabilelor

```
# -------------------------------------------1------------------------------

set.seed(123)  # Asigură reproducibilitatea

# Parametrii problemei
n <- 10  # Numărul de etape
lambda <- runif(n, 0.5, 2)  # Ratele exponențiale pentru fiecare etapă
alpha <- runif(n, 0.5, 1)  # Probabilitățile de a continua

# Funcția de simulare a lui T
simulate_T <- function() {
  T_total <- 0
  i <- 1
  while (i <= n) {
    T_total <- T_total + rexp(1, lambda[i])  # Simulează timpul pentru etapa i
    if (runif(1) > alpha[i]) return(list(T_total, FALSE, i))  # Probabilitatea d
    i <- i + 1
  }
  return(list(T_total, TRUE, n))  # Finalizare activitate
}

# Simulăm 10^6 valori
n_sim <- 10^6
sim_results <- replicate(n_sim, simulate_T(), simplify = FALSE)
t_values <- sapply(sim_results, function(x) x[[1]])
completion_flags <- sapply(sim_results, function(x) x[[2]])
stopping_stages <- sapply(sim_results, function(x) x[[3]])

# Aproximarea lui E(T)
E_T <- mean(t_values)
cat("Estimarea lui E(T):", E_T, "\n")

# Reprezentare grafică
hist(t_values, breaks = 100, probability = TRUE,
     main = "Distribuția simulată a lui T",
     xlab = "Timp total T", col = "skyblue", border = "white")

# Suprapunere densitate estimată
lines(density(t_values), col = "red", lwd = 2)
```

```r
# ------------------------------------------2-----------------------------------

# Calcul exact al lui E(T)
E_T_exact <- sum(sapply(1:n, function(i) {
  (1 / lambda[i]) * prod(alpha[1:(i-1)])
}))
cat("Valoarea exactă a lui E(T):", E_T_exact, "\n")

#-------------------------------------------3-----------------------------------

# Aproximare probabilitate finalizare activitate
P_completion <- mean(completion_flags)
cat("Probabilitatea ca persoana A finalizează activitatea:", P_completion, "\n")

#-------------------------------------------4-----------------------------------

# Aproximare P(T <= sigma)
sigma <- 5  # Definim o valoare pentru sigma
P_T_leq_sigma <- mean(t_values <= sigma)
cat("Probabilitatea ca T <=", sigma, "este", P_T_leq_sigma, "\n")

#-------------------------------------------5-----------------------------------

# Timp minim și maxim pentru finalizările reușite
t_values_completed <- t_values[completion_flags]
T_min <- min(t_values_completed)
T_max <- max(t_values_completed)
cat("Timpul minim de finalizare:", T_min, "\n")
cat("Timpul maxim de finalizare:", T_max, "\n")

# Reprezentare grafică a timpilor de finalizare
hist(t_values_completed, breaks = 100, probability = TRUE,
     main = "Distribuția timpilor de finalizare",
     xlab = "Timp total T (doar pentru finalizări)", col = "lightgreen", border

#------------------------------------------6------------------------------------

# Probabilitatea de oprire înainte de etapa k
stop_probabilities <- sapply(2:n, function(k) mean(stopping_stages < k))
```

```
# Afișare probabilități
cat("Probabilități de oprire înainte de fiecare k:\n")
print(data.frame(k = 2:n, P_stop = stop_probabilities))

# Reprezentare grafică
plot(2:n, stop_probabilities, type = "b", col = "blue", pch = 19, lwd = 2,
     main = "Probabilitatea de oprire înainte de etapa k",
     xlab = "Etapa k", ylab = "Probabilitate de oprire",
     ylim = c(0, 1))
grid()
```



Data	
sim_results	Large list (1000000 elements, 256 MB)
Values	
alpha	num [1:10] 0.978 0.727 0.839 0.786 0.551 ...
completion_flags	Large logical (1000000 elements, 4 MB)
E_T	3.69975479832001
E_T_exact	3.67599147379786
lambda	num [1:10] 0.931 1.682 1.113 1.825 1.911 ...
n	10
n_sim	1e+06
P_completion	0.051714
P_T_leq_sigma	0.742761
sigma	5
stop_probabilities	num [1:9] 0.0216 0.289 0.4033 0.5314 0.7415 ...
stopping_stages	Large numeric (1000000 elements, 8 MB)
T_max	27.9925263183499
T_min	1.22754544072724
t_values	Large numeric (1000000 elements, 8 MB)
t_values_completed	num [1:51714] 6.53 6.73 3.79 7.16 7.78 ...
Functions	
simulate_T	function ()

Figure 6: Valorile datelor

Cerinta II

Descrierea problemei:

A. Se dau 5 repartitii (normala standardizata, normala nestandardizata, exponentiala, possion, binomiala) si 4-5 variabile aleatoare pentru fiecare repartitie. Se cere sa se reprezinte grafic functiile de repartitie pentru fiecare variabila aleatoare.
B. Se dau 7 functii de densitate. Se cere sa se reprezinte grafic functiile si sa se obtina media si varianta pentru o variabila aleatoare definita de fiecare functie.

Pachete software folosite:

Am folosit shiny: install.packages("shiny")

Cod in R:

```
library(shiny)

# Definirea interfetei grafice a aplicatiei
ui <- fluidPage(
  titlePanel("Aplicatie Shiny: Functii de Repartitie si Densitati"),
  sidebarLayout(
    sidebarPanel(
      tabsetPanel(
        id = "main_tabs",
        # Tab-ul pentru functiile de repartitie (A)
        tabPanel("Functii de Repartitie (A)", value = "A",
                 selectInput("distribution", "Alege distributia:",
                             choices = c("Normala (N(0,1))" = "normal",
                                         "Normala (N(μ,σ^2))" = "normal_custom",
                                         "Exponentiala" = "exponential",
                                         "Poisson" = "poisson",
                                         "Binomiala" = "binomial")),
                 # Afiseaza input-uri suplimentare in functie de distributia selectata
                 conditionalPanel(
```

```
  condition = "input.distribution == 'normal'",
  selectInput("variable", "Alege variabila:",
        choices = c("X" = "X",
              "3-2X" = "3-2X",
              "X^2" = "X^2",
              "Suma de la 1 la n din Xi" = "Suma X",
              "Suma de la 1 la n din (Xi)^2" = "Suma X^2"
        ))
),
conditionalPanel(
  condition = "input.distribution == 'normal_custom'",
  selectInput("variable", "Alege variabila:",
        choices = c("X" = "X",
              "3-2X" = "3-2X",
              "X^2" = "X^2",
              "Suma de la 1 la n din Xi" = "Suma X",
              "Suma de la 1 la n din (Xi)^2" = "Suma X^2"
        )),
  numericInput("mean", "Media (μ):", value = 0),
  numericInput("sd", "Deviatia standard (σ):", value = 1, min = 0.01)
),
conditionalPanel(
  condition = "input.distribution == 'exponential'",
  selectInput("variable", "Alege variabila:",
        choices = c("X" = "X",
              "2+5X" = "2+5X",
              "X^2" = "X^2",
              "Suma de la 1 la n din Xi" = "Suma X"
        )),
  numericInput("lambda_exp", "Lambda (λ):", value = 1, min = 0.01)
),
conditionalPanel(
  condition = "input.distribution == 'poisson'",
  selectInput("variable", "Alege variabila:",
        choices = c("X" = "X",
              "3X-2" = "3X-2",
              "X^2" = "X^2",
              "Suma de la 1 la n din Xi" = "Suma X"
        )),
  numericInput("lambda_pois", "Lambda (λ):", value = 1, min = 0.01)
),
conditionalPanel(
  condition = "input.distribution == 'binomial'",
  selectInput("variable", "Alege variabila:",
```

```
                    choices = c("X" = "X",
                                "5X-4" = "5X-4",
                                "X^3" = "X^3",
                                "Suma de la 1 la n din Xi" = "Suma X"
                    )),
          numericInput("n", "Numar de incercari (n):", value = 10, min = 1),
          numericInput("p", "Probabilitate (p):", value = 0.5, min = 0, max = 1)
        ),
        # Selecteaza numarul de variabile generate si genereaza graficul
        numericInput("nr", "Numar de variabile aleatoare (nr):", value = 10, min = 1),
        actionButton("generateA", "Genereaza graficul")
    ),
    # Tab-ul pentru functiile de densitate (B)
    tabPanel("Functii de Densitate (B)", value = "B",
        selectInput("functionB", "Alege functia:",
                    choices = c("f(x) = cx^4" = "fx1",
                                "f(x) = ax + bx^2" = "fx2",
                                "f(x) = 4 / (x(x+1)(x+2))" = "fx3",
                                "f(x) = log(x/(x+1))" = "fx4",
                                "f(x) = (θ^2 / (1+θ))(1+x)e^(-θx))" = "fx5",
                                "f(x) = 1/3 * e^x pentru x<0, 1/3 pentru 0<=x<1, 1/3 * e^(-(x-1)) pentru x>=1" =
"fx6",
                                "f(x) = 1 / (π(1+x^2))" = "fx7")),
        # Input-uri suplimentare pentru functiile care au parametri
        conditionalPanel(
          condition = "input.functionB == 'fx2'",
          numericInput("a", "Parametrul a:", value = 1)
        ),
        conditionalPanel(
          condition = "input.functionB == 'fx5'",
          numericInput("theta", "Parametrul θ:", value = 1, min = 0.01)
        ),
        # Buton pentru generarea functiei si calcularea mediei si variantei
        actionButton("generateB", "Genereaza functia si calculeaza")
    )
   )
  ),
  # Panoul principal
  mainPanel(
   conditionalPanel(
    condition = "input.main_tabs === 'A'",
    plotOutput("plotA")
   ),
   conditionalPanel(
```

```
        condition = "input.main_tabs === 'B'",
        plotOutput("plotB"),
        verbatimTextOutput("statsB")
      )
    )
  )
)

# Definirea serverului
server <- function(input, output) {
  # Generarea functiilor de repartitie
  observeEvent(input$generateA, {
    output$plotA <- renderPlot({
      nr <- input$nr
      X <- NULL

      # Generarea variabilelor in functie de distributia selectata
      if (input$distribution == "normal") {
        X <- rnorm(nr, mean = 0, sd = 1)
      } else if (input$distribution == "normal_custom") {
        X <- rnorm(nr, mean = input$mean, sd = input$sd)
      } else if (input$distribution == "exponential") {
        X <- rexp(nr, rate = input$lambda_exp)
      } else if (input$distribution == "poisson") {
        X <- rpois(nr, lambda = input$lambda_pois)
      } else if (input$distribution == "binomial") {
        X <- rbinom(nr, size = input$n, prob = input$p)
      }

      # Transformarea variabilei
      if (input$variable == "X") {
        Y <- X
      } else if (input$variable == "3-2X") {
        Y <- 3 - 2 * X
      } else if (input$variable == "2+5X") {
        Y <- 2 + 5 * X
      } else if (input$variable == "3X-2") {
        Y <- 3 * X - 2
      } else if (input$variable == "5X-4") {
        Y <- 5 * X - 4
      } else if (input$variable == "X^2") {
        Y <- X^2
      } else if (input$variable == "X^3") {
        Y <- X^3
```

```
    } else if (input$variable == "Suma X") {
      Y <- cumsum(X)
    } else if (input$variable == "Suma X^2") {
      Y <- cumsum(X^2)
    }

    # Calculul si afisarea functiei de repartitie empirica
    F <- ecdf(Y)
    plot(F, main = "Functia de Repartitie Empirica", xlab = "Y", ylab = "F(Y)", col = "blue", lwd = 2)
  })
})

# Generarea functiilor de densitate
observeEvent(input$generateB, {
  output$plotB <- renderPlot({
    # Initializarea lui x si normalizarea lui f
    if (input$functionB == "fx1") {
      x <- seq(0, 2, length.out = 1000)
      c <- 5 / 32
      f <- c * x^4
    } else if (input$functionB == "fx2") {
      x <- seq(0, 1, length.out = 1000)
      b <- 3 * (1 - input$a / 2)
      f <- (input$a * x + b * x^2)
    } else if (input$functionB == "fx3") {
      x <- seq(1, 100)
      f <- 4 / (x * (x + 1) * (x + 2))
      f <- f / sum(f)
    } else if (input$functionB == "fx4") {
      x <- seq(1, 9)
      f <- log(x / (x + 1))
      f <- f / sum(f)
    } else if (input$functionB == "fx5") {
      x <- seq(0, 10, length.out = 1000)
      f <- (input$theta^2 / (1 + input$theta)) * (1 + x) * exp(-input$theta * x)
      norm_factor <- 1 / integrate(function(x) (input$theta^2 / (1 + input$theta)) * (1 + x) * exp(-
input$theta * x), 0, Inf)$value
      f <- norm_factor * f
    } else if (input$functionB == "fx6") {
      x <- seq(-5, 5, length.out = 1000)
      norm_factor <- 1 / (
        integrate(function(x) (1/3) * exp(x), -Inf, 0)$value +
        integrate(function(x) (1/3) * x^0, 0, 1)$value +
        integrate(function(x) (1/3) * exp(-(x - 1)), 1, Inf)$value)
```

```
    f <- ifelse(x < 0, norm_factor * (1/3) * exp(x),
              ifelse(x >= 0 & x < 1, norm_factor * (1/3),
                     norm_factor * (1/3) * exp(-(x - 1))))
  } else if (input$functionB == "fx7") {
    x <- seq(-5, 5, length.out = 1000)
    f <- 1 / (pi * (1 + x^2))
  }

  # Afisarea graficului
  plot(x, f, type = "l", main = "Graficul functiei", xlab = "x", ylab = "f(x)", col = "red", lwd = 2)
})

output$statsB <- renderText({
  # Calculul mediei si variantei
  if (input$functionB == "fx1") {
    x <- seq(0, 2, length.out = 1000)
    c <- 5 / 32
    mean_val <- integrate(function(x) x * c * x^4, 0, 2)$value
    var_val <- integrate(function(x) (x - mean_val)^2 * c * x^4, 0, 2)$value
  } else if (input$functionB == "fx2") {
    x <- seq(0, 1, length.out = 1000)
    b <- 3 * (1 - input$a / 2)
    mean_val <- integrate(function(x) x * (input$a * x + b * x^2), 0, 1)$value
    var_val <- integrate(function(x) (x - mean_val)^2 * (input$a * x + b * x^2), 0, 1)$value
  } else if (input$functionB == "fx3") {
    x <- seq(1, 100)
    f <- 4 / (x * (x + 1) * (x + 2))
    f <- f / sum(f)
    mean_val <- sum(x * f)
    var_val <- sum((x - mean_val)^2 * f)
  } else if (input$functionB == "fx4") {
    x <- seq(1, 9)
    f <- log(x / (x + 1))
    f <- f / sum(f)
    mean_val <- sum(x * f)
    var_val <- sum((x - mean_val)^2 * f)
  } else if (input$functionB == "fx5") {
    x <- seq(1, 10, length.out = 1000)
    density_function <- function(x, theta) {
      (theta^2 / (1 + theta)) * (1 + x) * exp(-theta * x)
    }
    norm_factor <- 1 / integrate(function(x) density_function(x, input$theta), 0, Inf)$value
    normalized_density <- function(x, theta) {
      density_function(x, theta) * norm_factor
```

```
      }
      mean_val <- integrate(function(x) x * normalized_density(x, input$theta), 0, Inf)$value
      var_val <- integrate(function(x) (x - mean_val)^2 * normalized_density(x, input$theta), 0, Inf)$value
     } else if (input$functionB == "fx6") {
      x <- seq(-5, 5, length.out = 1000)
      norm_factor <- 1 / sum(
        integrate(function(x) (1/3) * exp(x), -Inf, 0)$value,
        integrate(function(x) (1/3) * x^0, 0, 1)$value,
        integrate(function(x) (1/3) * exp(-(x - 1)), 1, Inf)$value
      )
      mean_val <- (
        integrate(function(x) x * (1/3) * exp(x) * norm_factor, -Inf, 0)$value +
        integrate(function(x) x * (1/3) * norm_factor, 0, 1)$value +
        integrate(function(x) x * (1/3) * exp(-(x - 1)) * norm_factor, 1, Inf)$value
      )
      mean_x2_val <- (
        integrate(function(x) x^2 * (1/3) * exp(x) * norm_factor, -Inf, 0)$value +
        integrate(function(x) x^2 * (1/3) * norm_factor, 0, 1)$value +
        integrate(function(x) x^2 * (1/3) * exp(-(x - 1)) * norm_factor, 1, Inf)$value
      )
      var_val <- mean_x2_val - (mean_val)^2
     } else if (input$functionB == "fx7") {
      mean_val <- "nedefinita (Cauchy)"
      var_val <- "nedefinita (Cauchy)"
     }
     # Afisarea mediei si variantei
     paste("Media: ", mean_val, "\nVarianta: ", var_val)
    })
  })
}

# Initializarea aplicatiei
shinyApp(ui = ui, server = server)
```

Aplicatia pune la dispozitie 2 optiuni: Functii de repartitie (A) si Functii de densitate (B). Pentru functia de repartitie avem de ales repartitia din cele 5, apoi variabila aleatoare din cele 4-5 corespunzatoare repartitiei selectate. Pentru repartitia normala (nestandardizata) putem alege media si deviatia standard, pentru repartitia exponentiala si repartitia poisson putem alege lambda, iar pentru repartitia binomiala putem alege numarul de incercari si probabilitatea. Pentru toate repartitiile putem alege numarul de de variabile aleatoare. Pe partea de server creez repartitia in functie de selectia utilizatorului, apoi fac transformarea variabilei aleatoare selectate, calculez functia de repartitie empirica si fac graficul. Pentru functia de densitate avem de ales functia din cele 7, si parametri pe care poate sa ii modifice utilizatorul, unde e cazul. Pe partea de server generez valori in domeniul functiei, o normalizez, ii afisez graficul, si ii calculez media si varianta.

Aspecte teoretice care depasesc nivelul cursului:

Nu am folosit aspecte teoretice care depasesc nivelul cursului.

Dificultati intampinate in realizarea cerintelor:

O problema a fost la a sasea functie de la B, cea pe ramuri. Pe langa faptul ca a fost mai complex calculul fata de celelalte functii, am intampinat probleme la integrarea ramurii din mijloc in R. Dupa (mult) debugging am ajuns la concluzia ca in R o functie care are dat ca parametru x nu poate fi o constanta, asa ca am inmultit constanta cu x^0 si asa am rezolvat problema.

Aplicabilitate

Aplicatia dezvoltata poate fi utilizata în urmatoarele domenii practice:

-Finante/Asigurari: Modelarea riscului financiar prin distributii normale sau exponentiale (exemplu: o banca are un portofoliu de 100 de imprumuturi)

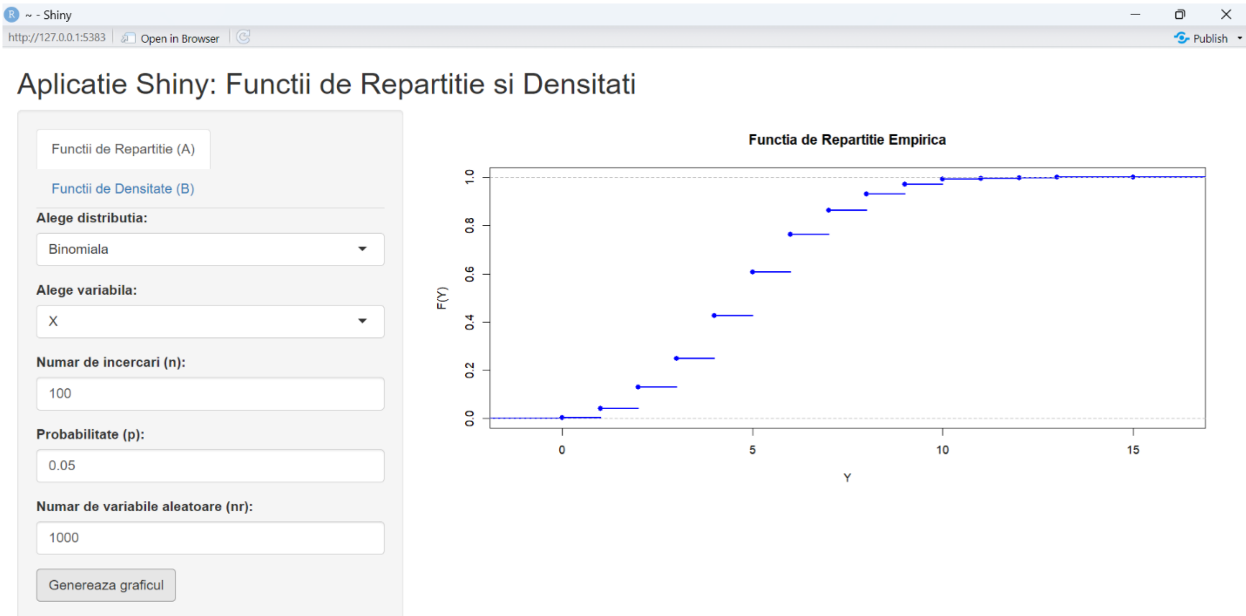
-Inginerie/Control calitate: Analiza ratei defectelor în productie cu distributii binomiale

Exemplu:

Un producator de becuri testeaza loturi de 100 de becuri pentru a identifica defectele. Probabilitatea ca un bec sa fie defect este de 5%. Dorim sa estimam probabilitatea de a avea un anumit numar de becuri defecte intr-un lot de 100.

Distributia: Distributie **Binomiala**.
Variabila: **X**
Numar de încercări (n): **100** (pentru un lot de 100 becuri).

Probabilitatea (p): **0,05** (probabilitatea ca un bec să fie defect).
Numar de variabile aleatoare (nr): **1000** (simulam 1000 de loturi de 100 de becuri).



Interpretarea graficului:

1. Valoarea 0.4 pe axa F(y) cand Y este aproximativ 5 ne spune ca in aproximativ 40% din loturi vom gasi 5 sau mai putine becuri defecte.
2. Valoarea 0.8 pe axa F(y) cand Y este aproximativ 8 ne spune ca in aproximativ 80% din loturi vom găsi 8 sau mai putine becuri defecte.
3. Media teoretica asteptata este np = 100 × 0.05 = 5 becuri defecte per lot

Implicatii practice pentru producator:

- Este foarte rar sa gasim 0 becuri defecte intr-un lot (probabilitate aproape de 0).
- Este la fel de rar sa gasim mai mult de 10 becuri defecte.
- Cel mai probabil vom gasi între 3 si 7 becuri defecte per lot.
- Dacă intr-un lot gasim peste 10 becuri defecte, ar trebui sa investigam procesul de productie, deoarece acest rezultat este neobisnuit.

Potentiale imbunatatiri:

-Integrarea cu date reale:

Importul de fisiere CSV pentru compararea distributiilor empirice cu cele teoretice.

Exportul rezultatelor (grafice si valori numerice) in formate standard (PNG, CSV).

-Functionalitati avansate:

Intervale de incredere pentru medie si varianta.

Concluzii:

Aceasta aplicatie interactiva in Shiny analizeaza distributii probabilistice si functii de densitate, aplicand concepte din cursul de probabilitati si statistica. In contextul industriei are aplicatii practice semnificative, cum ar fi in **FinTech**, unde poate modela riscuri financiare, si in **inginerie**, pentru analiza calitatii si controlul proceselor de productie.

Prin implementarea unor imbunatatiri precum integrarea datelor reale si funcționalitati avansate precum intervale de incredere, aplicatia poate evolua la o unealta profesionala folosita in diverse sectoare economice, devenind astfel un sprijin in luarea deciziilor bazate pe analize statistice complexe.

Cerința III

1 Descrierea problemei:

Se dă o variabilă aleatoare X definită pentru anumite densități/funcții de masă și un eșantion pentru ea. Se cere să se estimeze parametrul „teta” prin metoda verosimilității maxime și prin metoda momentelor analitic/teoretic („pe foaie”). Ulterior, va trebui să se construiască o funcție în R care să preia eșantionul și să întoarcă estimațiile realizate în baza celor doi estimatori. În final, se va compara estimația dată de metoda verosimilității maxime cu valoarea lui teta dată de o metodă numerică.

2 Descrierea soluției și a altor amănunte teoretice:

Întâi de toate, voi explica ce este aceea o metodă de estimare și voi descrie succint ambele metode de estimare propuse de problemă: ce fac în esență și pașii lor generali.

Fiind dată o repartiție cu anumiți parametri necunoscuți, o metodă de estimare este o modalitate prin care încercăm să „ne apropiem cât de mult” de valorile reale ale parametrilor necunoscuți, prin utilizarea unuia sau a mai multor seturi de date/observații din acea repartiție, seturi care poartă denumirea de eșantioane.

Metoda momentelor:

Face presupunerea ca **media empirică**, media valorilor observate (care coincide chiar cu media lor aritmetica: $\bar{X} = \frac{1}{n}\sum_{i=1}^{n}$ Xi, unde X_1, X_2, X_3,...,X_n este eșantionul în formă virtuală; noi vom lucra cu valori concrete: x_i, $\bar{x}$): coincide cu **media repartiției**. Din egalitatea aceasta se va scoate parametrul necunoscut/parametrii necunoscuți (θ în $cazul\ nostru$) în funcție de $\bar{x}$. Spunem, în acest punct, ca am obținut un **estimator ($\widehat{\boldsymbol{\theta}}$)** al parametrului necunoscut (în funcție de eșantion) și, de fapt, chiar acest estimator va reprezenta funcția noastră de implementat în R. Metoda nu admite o comparare cu vreo metodă numerică, întrucât ar fi necesare mai multe eșantioane/simulări.

Metoda verosimilității maxime:

Așa cum îi spune și numele, ea va încerca să maximizeze, să ne ofere o valoare a parametrului necunoscut θ pentru care observațiile din eșantionul dat sunt cele mai probabile. Astfel, se va considera o funcție numită **funcția de verosimilitate:** $L(\theta|x_1, x_2,..,x_n) = \prod_{i=1}^{n} f(x_i)$ și se va căuta valoarea lui θ pentru care este maximă. Știind că logaritmul cu baza mai mare decât 1 păstrează monotonia și că f(x) este pozitivă, f nu este funcția nulă, fiind funcția specifică unei repartiții, putem **logaritma** și căuta **punctul de maxim** acolo, în noua expresie. Punctul de maxim se găsește, bineînțeles, prin identificarea unui punct critic și probarea faptului că **derivata a doua este negativ definită** acolo (atenție când e 0)/cu semnul primei derivate. Valoarea găsită pentru θ va fi exprimată în funcție de $\bar{x}$ adesea (ori altă funcție ce depinde de eșantion). În acest moment, putem spune că am găsit un estimator pentru θ (funcția din R). Compararea estimării cu o estimarea generată de o metodă numerică reprezintă compararea valorii lui θ obținută cu metoda descrisă (**maximul teoretic**) cu valoarea lui θ obținută din expresia logaritmată, de o funcție din R (**optimize()**) care ne dă punctul de maxim pe un interval - **maximul efectiv/practic**.

3 Rezolvarea problemei:

a)

a) $f_\theta(x) = e^{-2\theta} \cdot \frac{(2\theta)^x}{x!}, \; x \in \mathbb{N}, \; \theta \in \mathbb{R}$

Eșantion:

8 12 6 14 9 12 15 7 15 7 10 10 14 9 12 15 11 6 8 6 8 8 9 12 13 10 11 11 13 15 10 8 7
8 13 9 9 13 12 9 10 6 10 8 10 11 12 11 9 10 7 8 8 16 7 15 10 10 8 14 13 4 11 13 6 9
13 10 10 12 11 5 6 4 9 6 9 7 13 9 11 5 5 9 15 10 11 10 14 7 11 9 14 10 5 10 8 12 13
11

a) I. Metoda momentelor:

$E(X) = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, unde $X_1, X_2, \ldots, X_n$ - eșantion (1)

f_θ - discretă ($\Leftrightarrow X$ - discretă) $\Rightarrow$ În acest caz, f_θ - pmf $\Rightarrow$

$\Rightarrow E(X) = \sum_{x \in \mathbb{N}} x \cdot e^{-2\theta} \cdot \frac{(2\theta)^x}{x!} = e^{-2\theta}\left[\sum_{x \in \mathbb{N}^*} \frac{(2\theta)^x}{(x-1)!} + \underbrace{0}_{\text{"când } x=0\text{"}}\right] =$

$= e^{-2\theta} \cdot \sum_{x \in \mathbb{N}} \frac{(2\theta)^{x+1}}{x!} = e^{-2\theta} \cdot 2\theta \cdot \underbrace{\left(\sum_{x \in \mathbb{N}} \frac{(2\theta)^x}{x!}\right)}_{\to e^{2\theta}} = \cancel{e^{-2\theta}} \cdot 2\theta \cdot \cancel{e^{2\theta}} =$

$= 2\theta \quad (X \sim \text{Poisson}(2\theta))$. (2)

Din (1) și (2) $\Rightarrow 2\theta = \overline{x} \Leftrightarrow \theta = \frac{\overline{x}}{2} \Rightarrow \hat{\theta} = \frac{\overline{X}}{2}$.

II. Metoda verosimilității maxime:

$L(\theta \mid x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f_\theta(x_i)$, $x_i \in \mathbb{N}$, $\theta \in \mathbb{R}$ $\Rightarrow$

$\Rightarrow L(\theta) = (e^{-2\theta})^n \cdot \left(\prod_{i=1}^{n} \frac{1}{x_i!}\right) \cdot (2\theta)^{\sum_{i=1}^{n} x_i}$, $\sum_{i=1}^{n} x_i = n\overline{x}$ $\Rightarrow$

$\Rightarrow \ln L(\theta) = \boxed{-2\theta n} + \ln\left(\prod_{i=1}^{n} \frac{1}{x_i!}\right) + n\overline{x}\ln 2 + \boxed{n\overline{x}\ln\theta} \Rightarrow$

$\Rightarrow (\ln \circ L)'(\theta) = -2\cancel{n} + \frac{\cancel{n}\,\overline{x}}{\theta} = 0 \Leftrightarrow$

$\Leftrightarrow \boxed{\overline{x} = 2\theta} \Leftrightarrow \theta = \frac{\overline{x}}{2} \Rightarrow \hat{\theta} = \frac{\overline{X}}{2}$.

Obs: Cum f_θ nu este fc. nulă și nu este negativă $\Rightarrow \theta > 0$.

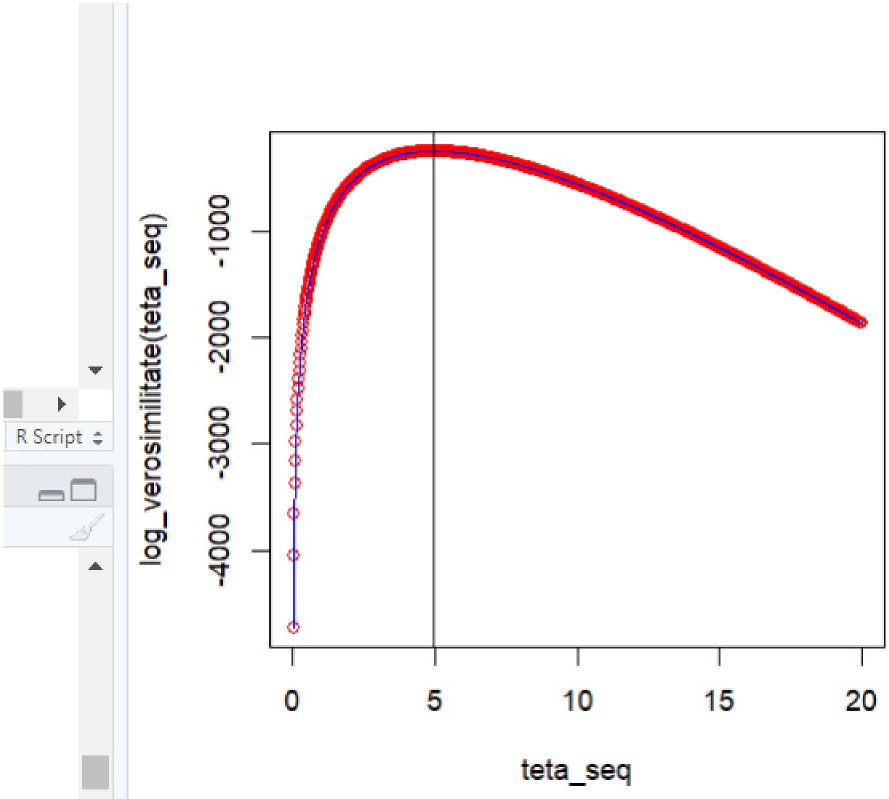
$(\ln \circ L)''(\theta) = -\frac{n\,\overline{x}}{\theta^2} < 0 \Rightarrow \theta = \frac{\overline{x}}{2}$ chiar este punct de maxim.

Se remarcă că obținem aceeași estimație teoretica prin ambele metode.

```
###a)
```

```r
esantion <- c(8, 12,6, 14,  9, 12, 15,  7, 15,  7, 10, 10, 14,  9, 12, 15,
11,  6,  8,  6,  8,  8,  9, 12, 13, 10, 11, 11, 13, 15, 10,  8,  7,
8, 13,  9,  9, 13, 12,  9, 10,  6, 10,  8, 10, 11, 12, 11,  9, 10,  7,  8,  8,
16,  7, 15, 10, 10,  8, 14, 13,  4, 11, 13,  6,  9,
13, 10, 10, 12, 11,  5,  6,  4,  9,  6,  9,  7, 13,  9, 11,  5,  5,  9, 15,
10, 11, 10, 14,  7, 11,  9, 14, 10,  5, 10,  8, 12, 13,
11 )#Retin esantionul intr-un vector.

x_bar_calculat <- mean(esantion)#Fac media esantionului.


estimare_efectiva <- function(x_bar){
cat("MME: ", x_bar/2, "\n", "MVM: ", x_bar/2)
}

estimare_efectiva(x_bar_calculat)


n <- length(esantion)#Fac lungimea esantionului.
const1 <- sum(-1*log(factorial(esantion)))#Fac constantele corespunzatoare
formei logaritmate.

#Compararea cu o metoda numerica:
log_verosimilitate <- function(teta){
   -2*teta*n+const1+n*x_bar_calculat*log(2)+n*x_bar_calculat*log(teta)
}

maxim_teoretic <- x_bar_calculat/2
o <- optimize(log_verosimilitate, c(0, 20) ,maximum=T)
maxim_efectiv <- o$maximum

teta_seq <- seq(0, 20, length.out = 1000)
plot(teta_seq,log_verosimilitate(teta_seq),col="red")
lines(teta_seq,log_verosimilitate(teta_seq), col="blue")
abline(v=maxim_teoretic,col="green")
abline(v=maxim_efectiv,col="black")
```

maxim_efectiv	4.97000700503495
maxim_teoretic	4.97



b)

b) $f_{\theta}(x) = C_n^x \cdot \theta^x (1-\theta)^{1-x}, \;\; x \in \{1, 2, 3 \ldots n\}, \;\; \theta \in (0,1), \;\; n \;\; fixat$

Eșantion pentru n=14:

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

(a) (I):

$E(X) = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, X_i, $i=\overline{1,n}$ - eșantion (1)

Dar, $E(X) \overset{f.\ discretă}{=} \underbrace{\sum_{i=1}^{n} i\, C_n^i \cdot \theta^i (1-\theta)^{1-i}}_{=f_\theta(i)} = \sum_{i=1}^{n}\left[\cancel{i} \cdot \frac{n!}{\cancel{i}(i-1)!\,(n-1-(i-1))!} \cdot \theta^i (1-\theta)^{1-i}\right] =$

$= n \cdot \theta \cdot \frac{1}{(1-\theta)^{n-1}} \cdot \sum_{i=1}^{n} \left[\frac{(n-1)!}{(i-1)!\,(n-1-(i-1))!} \cdot \theta^{i-1} \cdot (1-\theta)^{n-1-(i-1)}\right] =$

$= \frac{n\theta}{(1-\theta)^{n-1}} \cdot \sum_{i=1}^{n} \left[C_{n-1}^{i-1} \cdot \theta^{i-1} \cdot (1-\theta)^{n-1-(i-1)}\right] =$

$= \frac{n\theta}{(1-\theta)^{n-1}} \cdot \sum_{j=0}^{n-1} \left[C_{n-1}^{j} \cdot \theta^{j} \cdot (1-\theta)^{(n-1)-j}\right] = \frac{n\theta}{(1-\theta)^{n-1}} \cdot (\cancel{\theta}+1-\cancel{\theta})^{n-1} =$

$= \frac{n\theta}{(1-\theta)^{n-1}}$, $\theta \in (0,1)$, $n \in \mathbb{N}^*$ - fixat. (2)

Din (1) și (2) $\Rightarrow \overline{x} = \frac{n\theta}{(1-\theta)^{n-1}} \Rightarrow$ Se obține o ecuație de grad "n-1" - imposibil de rezolvat!

Presupun că e vorba de o greșeală și că am de-a face cu distribuția binomială (dar de la 1) $\Rightarrow E(X) = \overline{x} =$
$= n\theta \Leftrightarrow \theta = \frac{\overline{x}}{n} \Rightarrow \hat{\theta} = \frac{\overline{X}}{n}$

(II): $L(\theta \mid x_1, \dots, x_n) = \prod_{i=1}^{n} C_n^{x_i} \cdot \theta^{x_i} \cdot (1-\theta)^{n-x_i}$, $\theta \in (0,1)$, $n \in \mathbb{N}^*$ fixat

$\Rightarrow L(\theta) = \theta^{\sum_{i=1}^{n} x_i} \cdot (1-\theta)^{\sum_{i=1}^{n} n - x_i} \cdot \prod_{i=1}^{n} C_n^{x_i} = \theta^{n\overline{x}} \cdot (1-\theta)^{n(n-\overline{x})} \cdot \prod_{i=1}^{n} C_n^{x_i}$

$\Rightarrow \ln L(\theta) = n \cdot \overline{x} \cdot \ln\theta + n(n-\overline{x}) \cdot \ln(1-\theta) + \ln \prod_{i=1}^{n} C_n^{x_i} \Rightarrow$

$\Rightarrow (\ln L)'(\theta) = \frac{\cancel{n} \cdot \overline{x}}{\theta} + \frac{\cancel{n}\overline{x} - \cancel{n} \cdot n}{1-\theta} = 0 \overset{\cdot\,\theta(1-\theta) \geq 0}{\Leftrightarrow} \overline{x} - \cancel{\overline{x}\theta} + \cancel{\overline{x}\theta} - n\theta = 0$

$\Leftrightarrow \theta = \frac{\overline{x}}{n} \Rightarrow \hat{\theta} = \frac{\overline{X}}{n}$.

$\frac{\overline{x}}{n} \in [0,1]$

Verificare:

θ	0		$\frac{\overline{x}}{n}$		1
$(\ln L)'$		+ + + + + +	0	- - - - -	
$\ln L$		$\nearrow$		$\searrow$	

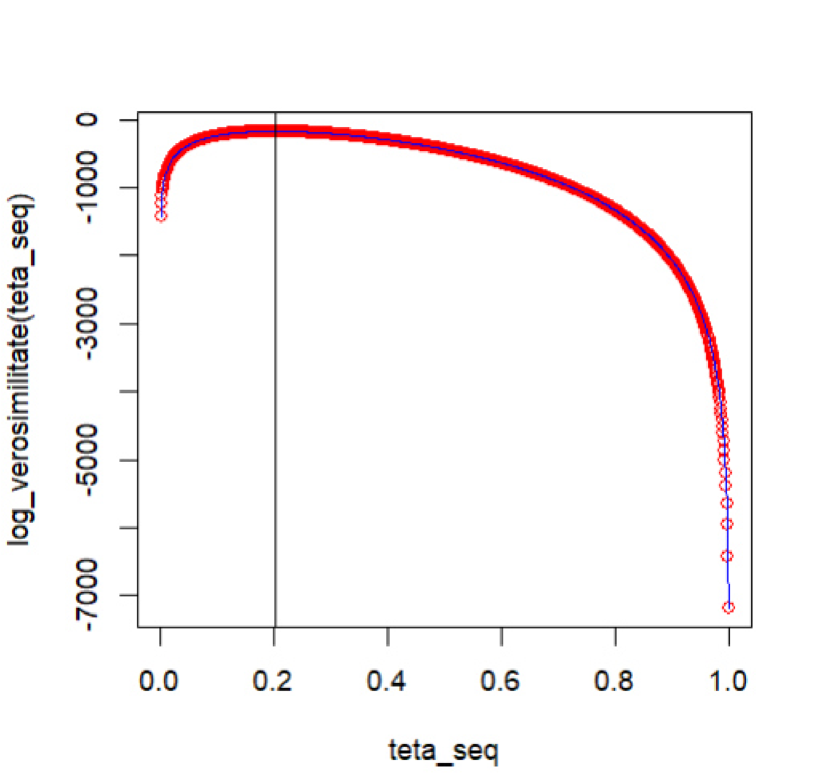
$\Rightarrow \hat{\theta} = \frac{\overline{x}}{n}$ este punct de maxim (chiar este pct. de maxim)

Deoarece se ajunge la rezolvarea unei ecuații de grad n-1, s-a considerat enunțul greșit și s-a lucrat cu repartiția binomială de la 1 (care are aceeași medie cu cea standard). In eșantion NU sunt toate valorile disponibile pentru funcția de repartiție, ci doar o parte din ele, iar unele se pot repeta. PENTRU A NU EXISTA CONFUZIE, S-A CONSIDERAT CA ESANTIONUL ESTE: $X_1, X_2, ..., X_m$ (n nu este neapărat egal cu m, iar x_i cu i). Se remarcă că obținem aceeași estimație teoretica prin ambele metode.

```
###b)
n <- 14##NU este lungimea esantionului/ ci un parametru fixat.
esantion <- 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)#Retin esantionul
intr-un vector.

x_bar_calculat <- mean(esantion)#Fac media esantionului.


estimare_efectiva <- function(x_bar){
  cat("MME: ", x_bar/n, "\n", "MVM: ", x_bar/n)
}

estimare_efectiva(x_bar_calculat)


m <- length(esantion)#Fac lungimea esantionului.
const1 <- sum(log(choose(n,esantion)))#Fac constantele corespunzatoare formei
logaritmate.
#Am folosit choose(n, k) pentru combinari.

#Compararea cu o metoda numerica:
log_verosimilitate <- function(teta){
  m*x_bar_calculat*log(teta) + m*(n - x_bar_calculat)*log(1-teta)+const1
}

maxim_teoretic <- x_bar_calculat/n
o <- optimize(log_verosimilitate, c(0,1) ,maximum=T)
```

```r
maxim_efectiv <- o$maximum

teta_seq <- seq(0, 1, length.out = 1000)
plot(teta_seq,log_verosimilitate(teta_seq),col="red")
lines(teta_seq,log_verosimilitate(teta_seq), col="blue")
abline(v=maxim_teoretic,col="green")
abline(v=maxim_efectiv,col="black")
```

m	100L
maxim_efectiv	0.201444561413453
maxim_teoretic	0.201428571428571
n	14



c)

c) $f_\theta(x) = e^{-\frac{x}{\theta}} \cdot \frac{x^{\alpha-1}}{\Gamma(\alpha)\cdot\theta^{\alpha}}$, $x \in (0,\infty)$, $\theta \in (0,\infty)$, $\alpha \in (0,\infty)$ *fixat*

Eșantion pentru $\alpha = 7$:

6.269128 25.204245 13.994878 13.391437 11.458827 10.565065 11.706398 10.625808
7.485952 16.353358 9.277565 8.566438 14.788638 6.830955 9.542004 20.272463
36.562137 12.244005 16.084879 11.454008 15.592298 6.332908 13.106441 6.198981
15.726780 7.883712 35.124934 11.856011 13.766200 16.534869 16.803648 11.196542
19.785629 26.300717 21.270154 7.192149 5.882948 15.812796 10.963237 24.963600

13.802383 15.281262 10.310398 20.940469 23.992540 15.869985 12.041726 12.521264
10.869006 15.386514 14.636832 18.104562 17.029779 4.506616 20.941222 12.050877
9.757833 20.070802 12.472900 6.474476 15.059776 13.157344 9.124414 13.768482
24.354934 12.363936 11.110749 9.092514 17.856801 14.757801 13.898665 9.119410
11.430184 11.958829 13.516191 10.701083 14.713596 10.121266 16.945351 13.524070
14.742403 19.165805 10.338392 12.327837 19.619227 7.328246 14.894399 19.631003
7.622796 12.343832 13.138183 10.061520 17.674638 9.675168 12.115561 15.182861
13.292479 17.888244 16.695139 2.952334

c) (I): $E(X) = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, unde $X_1, X_2, \dots, X_n$ - eșantion (1)

Dar, $E(X) \overset{f_\theta\text{-continuă}}{=} \int_0^{\infty} x f_\theta(x)\,dx = \int_0^{\infty} x \cdot e^{-\frac{x}{\theta}} \cdot \frac{x^{\alpha-1}}{\Gamma(\alpha)\cdot\theta^{\alpha}}\,dx =$

$= \int_0^{\infty} e^{-\frac{x}{\theta}} \cdot \frac{x^{\alpha}}{\Gamma(\alpha)\cdot\theta^{\alpha}}\,dx = \Gamma(\alpha)^{-1}\cdot\theta^{-\alpha}\cdot\int_0^{\infty} e^{-\frac{x}{\theta}}\cdot x^{\alpha}\,dx$

Facem schimbare de variabilă:

$\boxed{\frac{x}{\theta} = t} \Leftrightarrow x = \theta\cdot t \Rightarrow dx = \boxed{x'}\,dt = \theta\cdot dt$

$x \to 0 \Rightarrow t \to 0$

$x \to +\infty \Rightarrow t \to +\infty$ $\Rightarrow$

$\Rightarrow E(X) = \Gamma(\alpha)^{-1}\cdot\theta^{-\alpha}\cdot\int_0^{\infty} e^{-t}\cdot(\theta t)^{\alpha}\cdot\theta\,dt =$

$= \Gamma(\alpha)^{-1}\cdot\cancel{\theta^{-\alpha}}\cdot\theta^{\cancel{\alpha}+1}\cdot\int_0^{\infty} e^{-t}\cdot t^{\alpha}\,dt =$

$= \cancel{\Gamma(\alpha)^{-1}}\cdot\theta\cdot\Gamma(\alpha+1) = \alpha\cdot\theta$. (2)

$= \cancel{\Gamma(\alpha)}\cdot\alpha$

Din (1) și (2) $\Rightarrow$

$\Rightarrow \hat{\theta} = \frac{\overline{X}}{\alpha}$

(II): $L(\theta|x_1,\dots,x_n) = \prod_{i=1}^{n} f_\theta(x_i) = \prod_{i=1}^{n} e^{-\frac{x_i}{\theta}}\cdot\frac{x_i^{\alpha-1}}{\Gamma(\alpha)\cdot\theta^{\alpha}}$, $\theta > 0 \Rightarrow$

$\Rightarrow L(\theta) = e^{-\frac{\sum_{i=1}^{n} x_i}{\theta}}\cdot\frac{1}{\Gamma(\alpha)^n\cdot\theta^{n\alpha}}\cdot\left(\prod_{i=1}^{n} x_i^{\alpha-1}\right) \Rightarrow$

$\Rightarrow \ln L(\theta) = -\frac{n\cdot\overline{x}}{\theta} - n\cdot\ln(\Gamma(\alpha)) - n\cdot\alpha\cdot\ln(\theta) + \ln\left(\prod_{i=1}^{n} x_i^{\alpha-1}\right) \Rightarrow$

$\Rightarrow (\ln L)'(\theta) = \frac{n\cdot\overline{x}}{\theta^2} - \frac{n\cdot\alpha}{\theta} = 0 \Leftrightarrow \overline{x} = \theta\cdot\alpha \Leftrightarrow \theta = \frac{\overline{x}}{\alpha} \Rightarrow$

$\Rightarrow \hat{\theta} = \frac{\overline{X}}{\alpha}$ (care convine, căci: $(\ln L)''(\hat{\theta}) =$

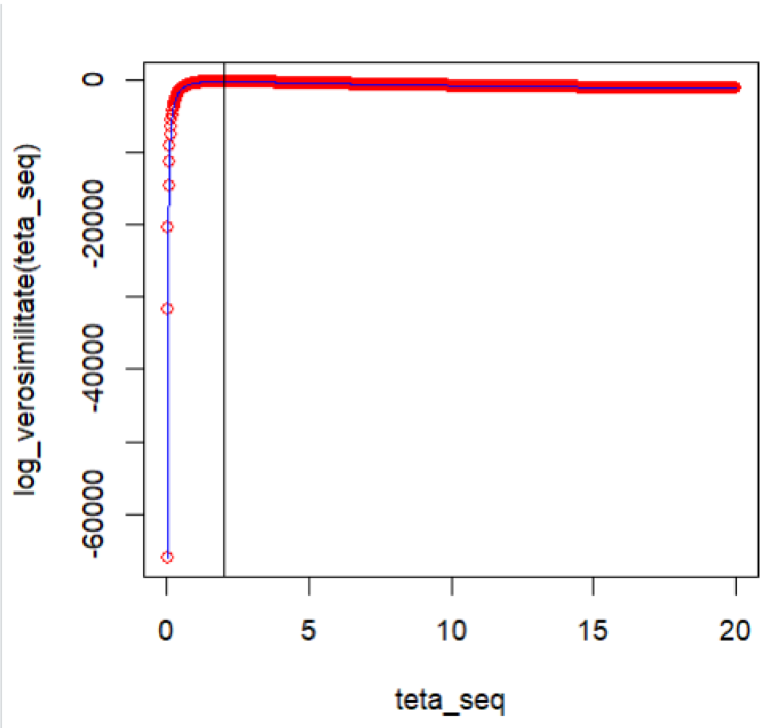
$= -2\cdot\frac{n\cdot\overline{x}}{\hat{\theta}^3} + \frac{n\cdot\alpha}{\hat{\theta}^2} = -2\cdot\frac{n\cdot\alpha}{\hat{\theta}^2} + \frac{n\cdot\alpha}{\hat{\theta}^2} = -\frac{n\cdot\alpha}{\hat{\theta}^2} < 0$).

Se remarcă că obținem aceeași estimație teoretica prin ambele metode.

```
###c)
alfa <- 7
esantion <- c(6.269128, 25.204245, 13.994878, 13.391437, 11.458827, 10.565065,
11.706398, 10.625808, 7.485952, 16.353358,
              9.277565, 8.566438, 14.788638, 6.830955, 9.542004, 20.272463,
36.562137, 12.244005, 16.084879, 11.454008,
              15.592298, 6.332908, 13.106441, 6.198981, 15.726780, 7.883712,
35.124934, 11.856011, 13.766200, 16.534869,
              16.803648, 11.196542, 19.785629, 26.300717, 21.270154, 7.192149,
5.882948, 15.812796, 10.963237, 24.963600,
              13.802383, 15.281262, 10.310398, 20.940469, 23.992540,
15.869985, 12.041726, 12.521264, 10.869006, 15.386514,
              14.636832, 18.104562, 17.029779, 4.506616, 20.941222, 12.050877,
9.757833, 20.070802, 12.472900, 6.474476, 15.059776,
              13.157344, 9.124414, 13.768482, 24.354934, 12.363936, 11.110749,
9.092514, 17.856801, 14.757801, 13.898665, 9.119410,
              11.430184, 11.958829, 13.516191, 10.701083, 14.713596,
10.121266, 16.945351, 13.524070, 14.742403, 19.165805,
              10.338392, 12.327837, 19.619227, 7.328246, 14.894399, 19.631003,
7.622796, 12.343832, 13.138183, 10.061520, 17.674638,
              9.675168, 12.115561, 15.182861, 13.292479, 17.888244, 16.695139,
2.952334)#Retin esantionul intr-un vector.

x_bar_calculat <- mean(esantion)#Fac media esantionului.


estimare_efectiva <- function(x_bar){
  cat("MME: ", x_bar/alfa, "\n", "MVM: ", x_bar/alfa)
}

estimare_efectiva(x_bar_calculat)


n <- length(esantion)#Fac lungimea esantionului.
const1 <- sum((alfa-1)*log(esantion))#Fac constantele corespunzatoare formei
logaritmate.

#Compararea cu o metoda numerica:
log_verosimilitate <- function(teta){
  (-1*n*x_bar_calculat)/teta-n*log(gamma(alfa))-n*alfa*log(teta)+const1
}
#Folosesc functia gamma() din R pentru gamma din exercitiu.
```

```r
maxim_teoretic <- x_bar_calculat/alfa
o <- optimize(log_verosimilitate, c(0, 20) ,maximum=T)
maxim_efectiv <- o$maximum

teta_seq <- seq(0, 20, length.out = 1000)
plot(teta_seq,log_verosimilitate(teta_seq),col="red")
lines(teta_seq,log_verosimilitate(teta_seq), col="blue")
abline(v=maxim_teoretic,col="green")
abline(v=maxim_efectiv,col="black")
```

maxim_efectiv	1.99329499067741
maxim_teoretic	1.99328521571429



d)

d) $f_\theta(x) = \dfrac{\theta^x}{(1+\theta)^{1+x}}, \quad x \in \mathbb{N}, \quad \theta \in (0,\infty)$

Eșantion:

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

(a) (I):

$E(X) = \overline{X}$, pt. un eșantion cu n elemente și presupus (1)

Dar, $E(X) \overset{\text{"discretă"}}{=} \sum_{x \in \mathbb{N}} x \cdot f_\theta(x) = \sum_{x \in \mathbb{N}} \frac{x \cdot \theta^x}{(1+\theta)^{1+x}} =$

$= \sum_{n \geq 0} \frac{n \cdot \theta^n}{(1+\theta)^{n+1}} = \frac{\theta}{(1+\theta)^2} \cdot \sum_{n \geq 1} n \cdot \left(\frac{\theta}{1+\theta}\right)^{n-1}$; $\Rightarrow \boxed{\frac{\theta}{1+\theta} = t} \in (0,1)$

$\underbrace{}_{= 0 + \sum_{n \geq 1} \frac{n \cdot \theta^n}{(1+\theta)^{n+1}}}$

$\Rightarrow \sum_{n \geq 1} n \left(\frac{\theta}{1+\theta}\right)^{n-1} = \sum_{n \geq 1} n \cdot t^{n-1} \overset{\boxed{n-1=m}}{=} \sum_{m \geq 0} (m+1) \cdot t^m$, $t \in (0,1)$

A = mulțimea de convergență = "intervalul de convergență" = $(0,1)$ $(\subseteq (-1,1))$

Fie $g : A \to \mathbb{R}$, $g(t) = \sum_{m \geq 0}^{\infty} (m+1) t^m$, suma seriei.

$\int g(t)\,dt = \left(\sum_{m \geq 0}^{\infty} t^{m+1}\right) + C = t \cdot \frac{\lim_{m \to \infty} t^m - 1}{t - 1} + C =$

$= \frac{t}{1-t} + C \Rightarrow g(t) = \left(\frac{t}{1-t} + C\right)' =$

$= \left(\frac{t}{1-t}\right)' = \frac{1 \cdot (1-t) - t \cdot (-1)}{(1-t)^2} = \frac{1 - \cancel{t} + \cancel{t}}{(1-t)^2} = \frac{1}{(1-t)^2} \Rightarrow$

$\Rightarrow E(X) = \frac{\theta}{(1+\theta)^2} \cdot \frac{1}{\left(1 - \frac{\theta}{1+\theta}\right)^2} = \frac{\theta}{(1+\theta)^2} \cdot \frac{1}{\left(\frac{1+\theta-\theta}{1+\theta}\right)^2} = \frac{\theta}{\cancel{(1+\theta)^2}} \cdot \cancel{(1+\theta)^2}$

$= \theta$. (2)

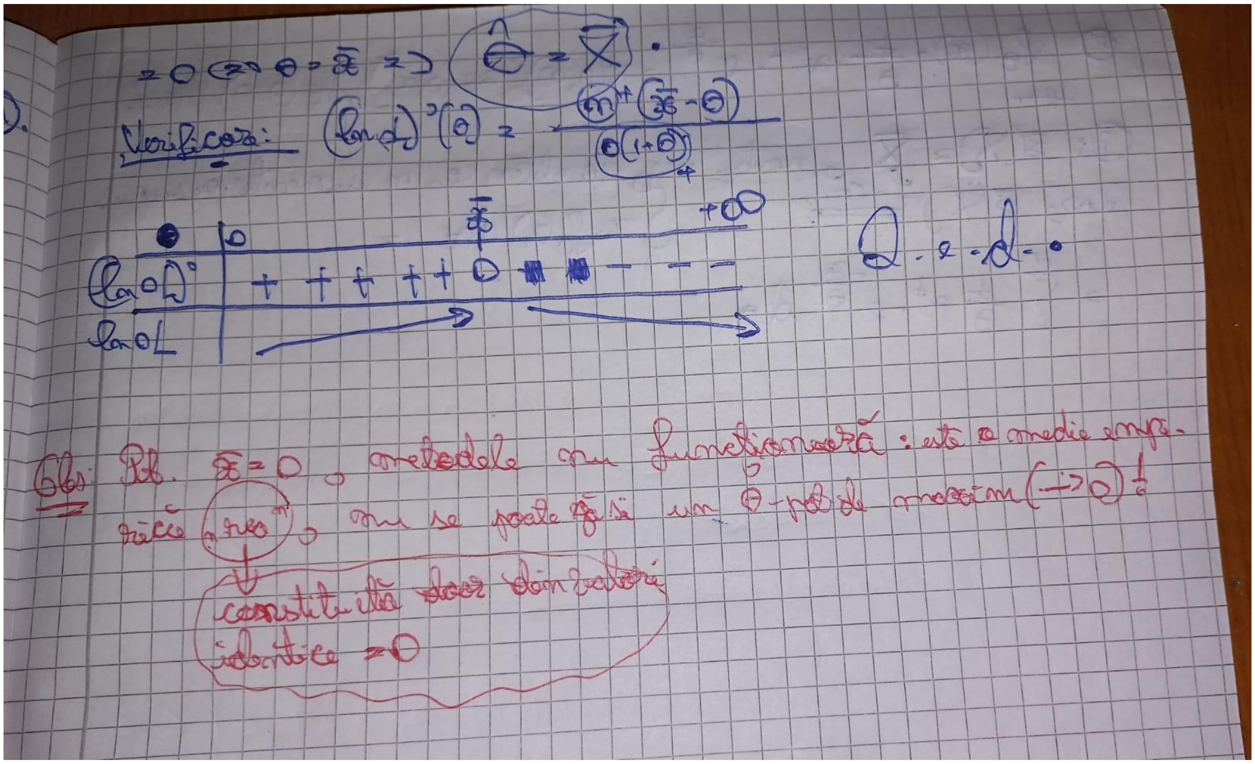
Din (1) și (2) $\Rightarrow \bar{x} = \theta \Rightarrow \hat{\theta} = \overline{X}$.

(II): $L(\theta | x_1, \dots, x_n) = \prod_{i=1}^{n} f_\theta(x_i)$, $\theta > 0$, pt. eșantionul anterior $\Rightarrow L(\theta) = \prod_{i=1}^{n} \frac{\theta^{x_i}}{(1+\theta)^{1+x_i}} = \frac{\theta^{\sum_{i=1}^n x_i}}{(1+\theta)^{\sum_{i=1}^n (1+x_i)}} =$

$\frac{1}{n}\sum_{i=1}^n x_i = \bar{x} \Rightarrow n\bar{x} = \sum_{i=1}^n x_i$

$\frac{\theta^{n\bar{x}}}{(1+\theta)^{n+n\bar{x}}} \Rightarrow \ln L(\theta) = n\bar{x} \cdot \ln(\theta) - (n + n\bar{x}) \ln(1+\theta)$

$\Rightarrow (\ln L)'(\theta) = \frac{n\bar{x}}{\theta} - \frac{n + n\bar{x}}{1+\theta} = 0 \Leftrightarrow \cancel{n\bar{x}} + \theta \cdot n\bar{x} - \theta n - \theta n\bar{x}$

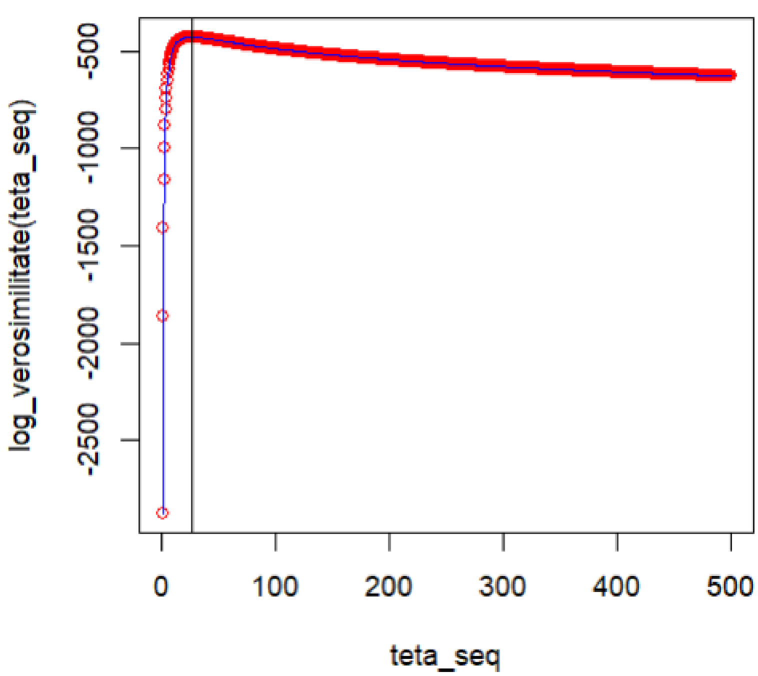


Se remarcă că obținem aceeași estimație teoretica prin ambele metode.

```
####d)
esantion <- 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 )#Retin esantionul intr-un
vector.

x_bar_calculat <- mean(esantion)#Fac media esantionului.


estimare_efectiva <- function(x_bar){
  cat("MME: ", x_bar, "\n", "MVM: ", x_bar)
}

estimare_efectiva(x_bar_calculat)
```

```
n <- length(esantion)#Fac lungimea esantionului.

#Compararea cu o metoda numerica:
log_verosimilitate <- function(teta){
  n*x_bar_calculat*log(teta)-(n+n*x_bar_calculat)*log(1+teta)
}

maxim_teoretic <- x_bar_calculat
o <- optimize(log_verosimilitate, c(min(esantion),max(esantion)), maximum=T)
maxim_efectiv <- o$maximum

teta_seq <- seq(0, 500, length.out = 1000)
plot(teta_seq,log_verosimilitate(teta_seq),col="red")
lines(teta_seq,log_verosimilitate(teta_seq), col="blue")
abline(v=maxim_teoretic,col="green")
abline(v=maxim_efectiv,col="black")
```

maxim_efectiv	25.8500072984973
maxim_teoretic	25.85

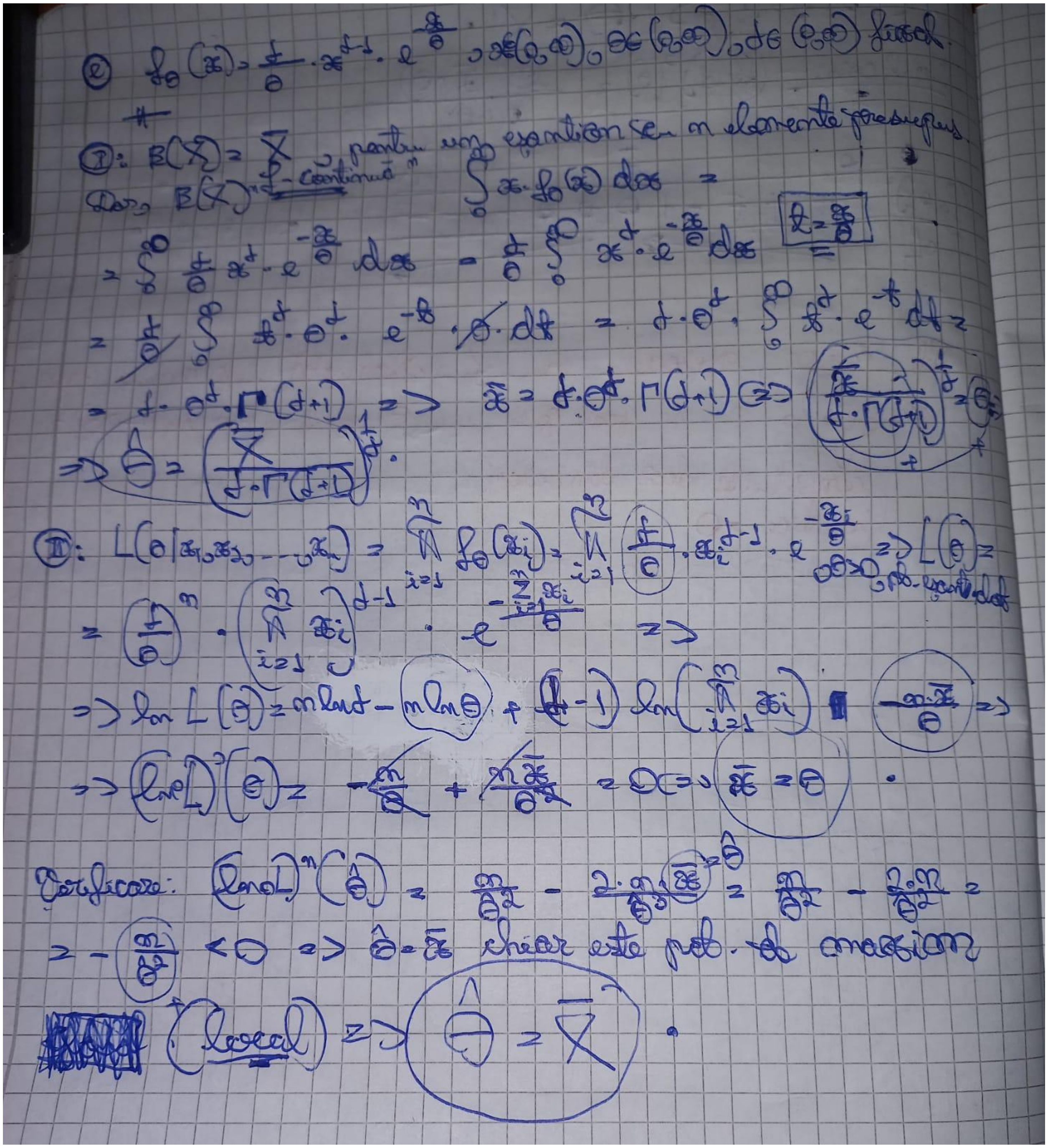


e)

e) $f_\theta(x) = \dfrac{\alpha}{\theta} x^{\alpha-1} \cdot e^{-\frac{x}{\theta}}, \;\; x \in (0,\infty), \;\; \theta \in (0,\infty), \;\; \alpha \in (0,\infty) \;\; fixat$

Eșantion pentru $\alpha = 3$:

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



(e) $f_\theta(x) = \frac{\alpha}{\theta} \cdot x^{\alpha-1} \cdot e^{-\frac{x^\alpha}{\theta}}$, $x \in (0,\infty)$, $\theta \in (0,\infty)$, $\alpha \in (0,\infty)$ fixat.

I: $E(X) = \bar{X}$, pentru un eșantion de n elemente presupus.

Dar, $E(X) \overset{\text{"f-continuă"}}{=} \int_0^\infty x \cdot f_\theta(x)\,dx =$ $\boxed{t = \frac{x^\alpha}{\theta}}$

$= \int_0^\infty \frac{\alpha}{\theta} x^\alpha \cdot e^{-\frac{x^\alpha}{\theta}}\,dx = \frac{\alpha}{\theta}\int_0^\infty x^\alpha \cdot e^{-\frac{x^\alpha}{\theta}}\,dx$

$= \frac{\alpha}{\cancel{\theta}}\int_0^\infty t^\alpha \cdot \theta^\alpha \cdot e^{-t} \cdot \cancel{\theta} \cdot dt = \alpha \cdot \theta^\alpha \cdot \int_0^\infty t^\alpha \cdot e^{-t}\,dt =$

$= \alpha \cdot \theta^\alpha \cdot \Gamma(\alpha+1) \Rightarrow \bar{x} = \alpha \cdot \theta^\alpha \cdot \Gamma(\alpha+1) \Leftrightarrow \left(\frac{\bar{x}}{\alpha \cdot \Gamma(\alpha+1)}\right)^{\frac{1}{\alpha}} = \theta$

$\Rightarrow \hat{\theta} = \left(\frac{\bar{X}}{\alpha \cdot \Gamma(\alpha+1)}\right)^{\frac{1}{\alpha}}$.

II: $L(\theta|x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} f_\theta(x_i) = \prod_{i=1}^{n} \frac{\alpha}{\theta} \cdot x_i^{\alpha-1} \cdot e^{-\frac{x_i}{\theta}} \Rightarrow L(\theta) =$ ($\theta > 0$, α-constant)

$= \left(\frac{\alpha}{\theta}\right)^n \cdot \left(\prod_{i=1}^{n} x_i\right)^{\alpha-1} \cdot e^{-\frac{\sum_{i=1}^{n} x_i}{\theta}} \Rightarrow$

$\Rightarrow \ln L(\theta) = n\ln\alpha - n\ln\theta + (\alpha-1)\ln\left(\prod_{i=1}^{n} x_i\right) - \frac{n \cdot \bar{x}}{\theta} \Rightarrow$

$\Rightarrow (\ln L)'(\theta) = -\frac{n}{\theta} + \frac{n\bar{x}}{\theta^2} = 0 \Leftrightarrow \bar{x} = \theta$.

Verificare: $(\ln L)''(\hat{\theta}) = \frac{n}{\hat{\theta}^2} - \frac{2 \cdot n \cdot \bar{x}}{\hat{\theta}^3} = \frac{n}{\hat{\theta}^2} - \frac{2 \cdot n}{\hat{\theta}^2} =$ ($\bar{x} = \hat{\theta}$)

$= -\frac{n}{\hat{\theta}^2} < 0 \Rightarrow \hat{\theta} = \bar{x}$ chiar este pct. de maxim (local) $\Rightarrow \hat{\theta} = \bar{X}$.

Se remarcă că NU obținem aceeași estimație teoretică prin ambele metode.

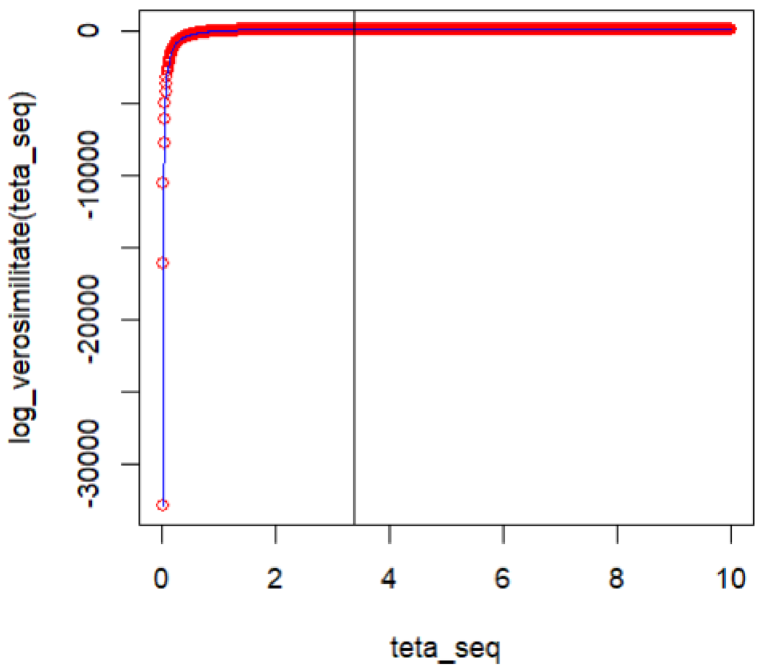
```
####e)
alfa <- 3
esantion <- c(3.5930579, 2.1027540, 1.7820777, 9.6550388, 6.8803846,
0.7388358, 2.9194654, 3.1178660, 1.2323236, 2.9776820,
```

```r
                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)#Retin esantionul intr-
un vector.

x_bar_calculat <- mean(esantion)#Fac media esantionului.


estimare_efectiva <- function(x_bar){
  cat("MME: ", (x_bar/(alfa*gamma(alfa+1)))^(1/alfa), "\n", "MVM: ", x_bar)
}

estimare_efectiva(x_bar_calculat)


n <- length(esantion)#Fac lungimea esantionului.
const1 <- (alfa-1)*sum(log(esantion))#Fac constantele corespunzatoare formei
logaritmate.


#Compararea cu o metoda numerica:
log_verosimilitate <- function(teta){
  n*log(alfa)-n*log(teta)+const1-(n*x_bar_calculat)/teta
}

maxim_teoretic <- x_bar_calculat
o <- optimize(log_verosimilitate, c(min(esantion),max(esantion)), maximum=T)
maxim_efectiv <- o$maximum

teta_seq <- seq(0, 10, length.out = 1000)
```

```
plot(teta_seq,log_verosimilitate(teta_seq),col="red")
lines(teta_seq,log_verosimilitate(teta_seq), col="blue")
abline(v=maxim_teoretic,col="green")
abline(v=maxim_efectiv,col="black")
```

```
> estimare_efectiva(x_bar_calculat)
MME:   0.5722134
 MVM:   3.372459>
>
```

maxim_efectiv	3.37244838689443
maxim_teoretic	3.372458719



4 Concluzii:

Dacă vrem un estimator mai simplu, dar posibil mai puțin precis, MME este o alegere decentă.

MVM oferă un estimator mai bun pentru θ decât MME, deoarece are varianță mai mică (asimptotic/pentru eșantioane mari) și proprietăți mai bune în general, totuși, este adesea mai greu de calculat.