

Reprezentarea grafică a densității și a funcției de repartiție pentru diferite valori ale parametrilor repartiției. În cazul în care funcția de repartiție nu este dată într-o formă explicită (ex. repartiția normală) se acceptă reprezentarea grafică a unei aproximări a acesteia

Oferim un set de funcții, câte două pentru fiecare repartiție (normală, exponențială, beta, gamma), care, pe baza parametrilor corespunzători, afișează densitatea sau funcția-repartiție. De exemplu, pentru a obține graficul densității în repartiția beta de parametri (2, 5), apelăm `den.beta(2, 5)`. Unele funcții conțin și valori implicite (repartiția normală este standard „by default” – la apel fără argumente). Pentru repartiția beta, am adăugat și o funcție care afișează graficul densității animat: o buclă apelează succesiv `plot`, urmat de un foarte scurt timp de așteptare, pentru cursivitatea animației. Graficul mișcător nu poate fi exportat însă; este funcțional în interiorul mediului RStudio. Detalii despre funcțiile densitate și repartiție în exercițiul 8.

Funcțiile sunt definite astfel:

```

1      # Repartitia normala (implicit standard); m = medie, s = deviatie
      standard
2      den.normala <- function(m = 0, s = 1) {
3          curve(expr = 1 / (s * sqrt(2 * pi)) * exp(1) ^ (-(x - m)^2 / (2 *
4              s^2)),
5              from = m - 3 * s,
6              to   = m + 3 * s,
7              ylab = "densitate",
8              main = "Densitatea in repartitia normala")
9          abline(v = 0, col = "gray")
10         }
11
12     rep.normala <- function(m = 0, s = 1) {
13         curve(expr = pnorm(x, m, s),
14             from = m - 3 * s,
15             to   = m + 3 * s,
16             ylab = "probabilitate",
17             main = "Functia repartitie normala")
18     }
19
20     # Repartitia exponentiala
21
22     den.exponentiala <- function(l = 1) {
23         curve(expr = l * exp(1) ^ (-l * x),
24             from = 0,
25             to   = 20,
26             ylab = "densitate",
27             main = "Densitatea in repartitia exponentiala")
28     }
29
30     rep.exponentiala <- function(l = 1) {
31         curve(expr = 1 - exp(1) ^ (-l * x),
32             from = 0,

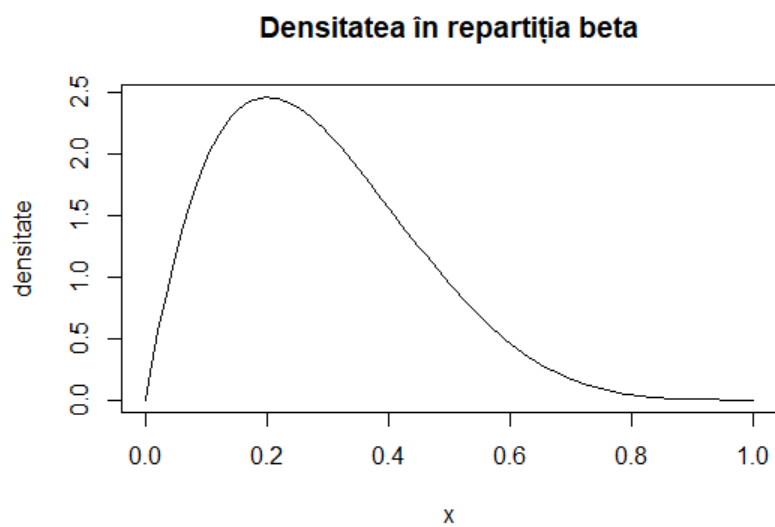
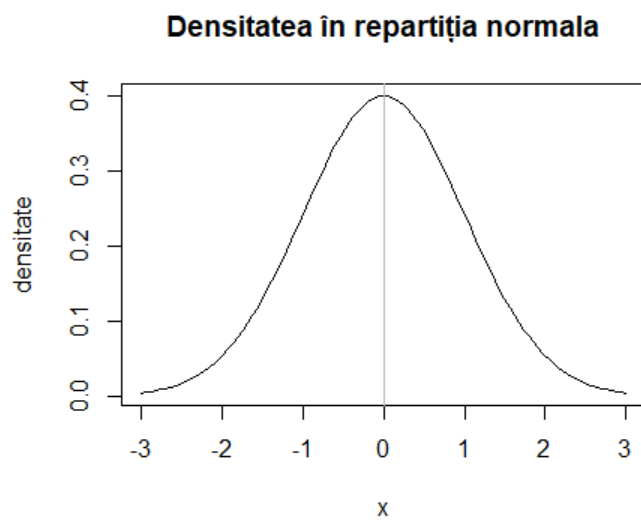
```

```

32     to    = 10,
33     ylab  = "probabilitate",
34     main  = "Functia_repartitie_exponentiala")
35 }
36
37 # Repartitia beta
38
39 den.beta <- function(a, b) {
40     curve(expr = x^(a - 1) * (1 - x) ^ (b - 1) /
41     integrate(function(u) u^(a - 1) * (1 - u)^(b - 1),
42     0, 1, abs.tol = 0)$value,
43     from = 0,
44     to    = 1,
45     ylab  = "densitate",
46     main  = "Densitatea_in_repartitia_beta")
47 }
48
49 # Teste: den.beta(0.5, 0.5)
50 #         den.beta(2, 2)
51 #         den.beta(2, 5)
52
53 # la fel, dar animata; a - fixat, 0 < b <= bmax
54 den.beta_anim <- function(a = 2, bmax = 8) {
55
56     par(bty = "o")
57
58     for (b in seq(0.1, bmax, length.out = 150)) {
59
60         d <- function(x) x^(a - 1) * (1 - x) ^ (b - 1) /
61         integrate(function(u) u^(a - 1) * (1 - u)^(b - 1),
62         0, 1, abs.tol = 0)$value
63
64         x <- seq(0, 1, length.out = 1000)
65         plot(x,
66         d(x),
67         type = "l",
68         xlab = "x",
69         ylab = "densitate",
70         ylim = c(0, 7),
71         main = "Densitatea_in_repartitia_beta")
72
73         legend(0.7, 6.5, legend = sprintf("b=%.2f", b), cex = 0.8)
74
75         Sys.sleep(0.05)
76     }
77
78 }
79
80 # Test: den.beta_anim()
81 #         den.beta_anim(0.8, 15)
82
83 rep.beta <- function(a, b) {
84     curve(expr = pbeta(x, shape1 = a, shape2 = b),
85     from = 0,

```

```
86     to    = 1,
87     ylab  = "probabilitate",
88     main  = "Functia_repartitie_beta")
89   }
90
91   # Teste: rep.beta(5, 1)
92   #         rep.beta(2, 2)
93   #         rep.beta(2, 5)
94
95
96   # Repartitia gamma; k - "shape", t - "scale"
97
98   den.gamma <- function(k, t) {
99     curve(expr = dgamma(x, shape = k, scale = t),
100     from = 0,
101     to    = 20,
102     ylab  = "densitate",
103     main  = "Densitatea_in_repartitia_gamma")
104   }
105
106   # Teste: den.gamma(1, 2)
107   #         den.gamma(2, 2)
108   #         den.gamma(7.5, 1)
109
110   rep.gamma <- function(k, t) {
111     curve(expr = pgamma(x, shape = k, scale = t),
112     from = 0,
113     to    = 20,
114     ylab  = "probabilitate",
115     main  = "Functia_repartitie_gamma")
116   }
117
118   # Teste: rep.gamma(0.5, 1)
119   #         rep.gamma(7.5, 1)
```

Figure 1: `den.beta(2, 5)`Figure 2: `den.normala(?, ?)`

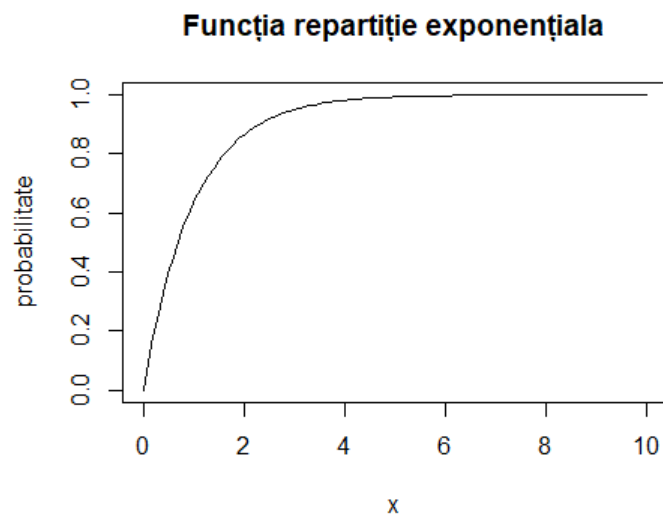


Figure 3: `rep.exponentiala(?, ?)`

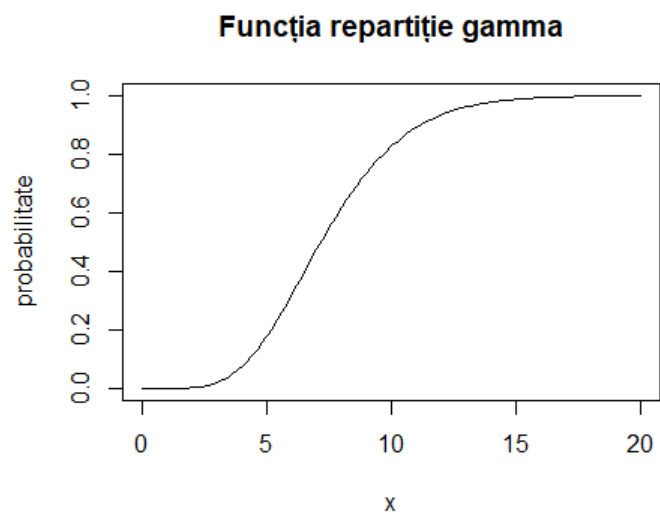


Figure 4: `rep.gamma(?, ?)`