Laborator 6

Elemente de estimare punctuală

Obiectivul acestui laborator este de a ilustra noțiunea de consistență a unui estimator precum și de a compara mai mulți estimatori.

1 Proprietăți ale estimatorilor

1.1 Exemplu de comparare a trei estimatori

Fie $X_1, X_2, \dots, X_n$ un eșantion de talie n dintr-o populație normală de medie μ și variantă σ^2. Atunci

$$\hat{\mu}_1 = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \hat{\mu}_2 = M_n \,(\text{mediana}\,), \quad \hat{\mu}_3 = \frac{X_{(1)} + X_{(n)}}{2}$$

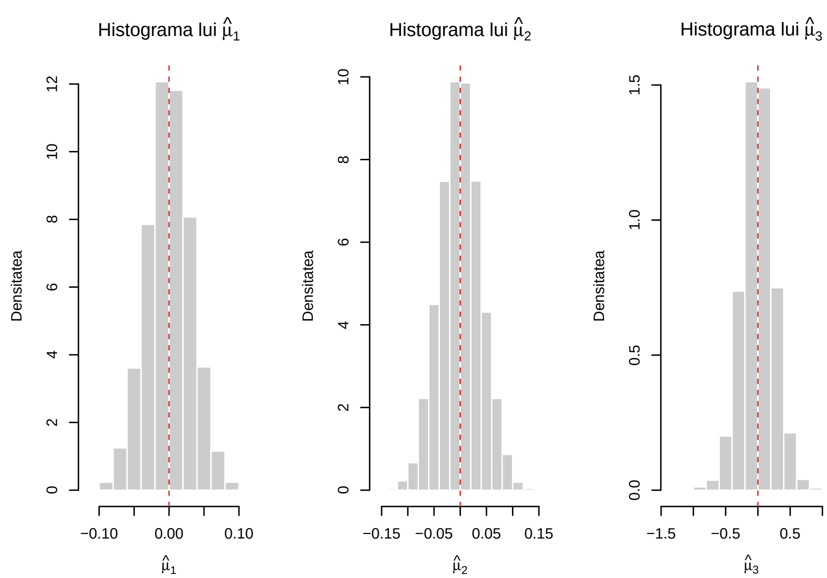
sunt trei estimatori punctuali pentru μ. Creați o funcție care să ilustreze cum sunt repartizați cei trei estimatori. Începeți cu $n = 10$, $\mu = 0$ și $\sigma^2 = 1$ și trasați histogramele pentru a-i compara. Ce se întâmplă dacă schimbați n, μ sau σ^2 ?

Vom crea o funcție numită `norm_estimators` care va construi repartițiile celor trei estimatori:

```
norm_estimators = function(n, mu, sigma, S){
  # Initializam
  mu1 = numeric(S)
  mu2 = numeric(S)
  mu3 = numeric(S)

  # repetam experimentul de S ori
  for (i in 1:S){
    x = rnorm(n, mean = mu, sd = sigma)

    # calculam estimatorii
    mu1[i] = mean(x)
    mu2[i] = median(x)
    mu3[i] = (min(x)+max(x))/2
  }

  # afisam variantele estimatorilor
  print(cbind(var_mu1 = var(mu1), var_mu2 = var(mu2), var_mu3 = var(mu3)))

 return(cbind(mu1 = mu1, mu2 = mu2, mu3 = mu3))

}
```

Pentru a ilustra grafic histogramele celor trei estimatori, considerăm $\mu = 0$ și $\sigma^2 = 1$ și avem:

```
mu = 0
sigma = 1

n = 1000
S = 10000

a = norm_estimators(n, mu, sigma, S)
         var_mu1     var_mu2    var_mu3
[1,] 0.0009916917 0.001546501 0.06069873

par(mfrow = c(1,3))
hist(a[,1], freq=FALSE, xlab=expression(hat(mu)[1]),
     col="gray80", border="white",
     main = expression(paste("Histograma lui ", hat(mu)[1])),
     ylab = "Densitatea")
abline(v=mu, col = "brown3", lty = 2)

hist(a[,2], freq=FALSE, xlab=expression(hat(mu)[2]),
     col="gray80", border="white",
     main = expression(paste("Histograma lui ", hat(mu)[2])),
     ylab = "Densitatea")
abline(v=mu, col = "brown3", lty = 2)

hist(a[,3], freq=FALSE, xlab=expression(hat(mu)[3]),
     col="gray80", border="white",
     main = expression(paste("Histograma lui ", hat(mu)[3])),
     ylab = "Densitatea")
abline(v=mu, col = "brown3", lty = 2)
```



1.2 Ilustrarea consistenței unui estimator

Fie $X_1, X_2, \dots, X_n$ un eșantion de talie n dintr-o populație $Pois(\theta)$. Ilustrați grafic consistența estimatorului $\hat{\theta}_n = S_n^2$ trasând histograma repartiției lui $\hat{\theta}_n$ pentru $n \in \{10, 25, 50, 100\}$. Ce observați?

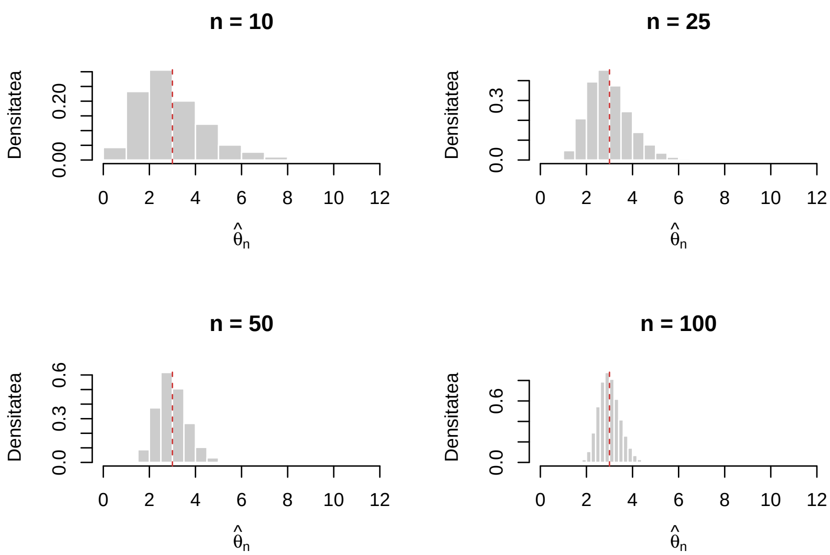
Considerăm funcția `pois_est` care pentru θ fixat simulează repartiția estimatorului $\hat{\theta}_n$:

```
pois_est1 = function(n, theta, S){
  # initializare
  sigma1 = numeric(S)

  for (i in 1:S){
    x = rpois(n, theta)
    sigma1[i] = var(x)
  }
  # afisam varianta estimatorului
  print(paste0("Pentru n = ", n," varianta estimatorului este ", var(sigma1)))
  return(sigma1)
}
```

Considerând $\theta = 3$ și $n \in \{10, 25, 50, 100\}$ avem:

```
theta = 3

par(mfrow=c(2,2))
a1 = pois_est1(10, theta, 50000)
[1] "Pentru n = 10 varianta estimatorului este 2.30244438915425"
a2 = pois_est1(25, theta, 50000)
[1] "Pentru n = 25 varianta estimatorului este 0.870906680642053"
a3 = pois_est1(50, theta, 50000)
[1] "Pentru n = 50 varianta estimatorului este 0.428041768952562"
a4 = pois_est1(100, theta, 50000)
[1] "Pentru n = 100 varianta estimatorului este 0.212316756387859"


hist(a1, freq=FALSE, xlab=expression(hat(theta)[n]),
     col="gray80", border="white", main = "n = 10", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)

hist(a2, freq=FALSE, xlab=expression(hat(theta)[n]),
     col="gray80", border="white", main = "n = 25", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)

hist(a3, freq=FALSE, xlab=expression(hat(theta)[n]),
     col="gray80", border="white", main = "n = 50", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)

hist(a4, freq=FALSE, xlab=expression(hat(theta)[n]),
     col="gray80", border="white", main = "n = 100", xlim = c(0,12),
     ylab = "Densitatea")
```

```r
abline(v=theta, col = "brown3", lty = 2)
```

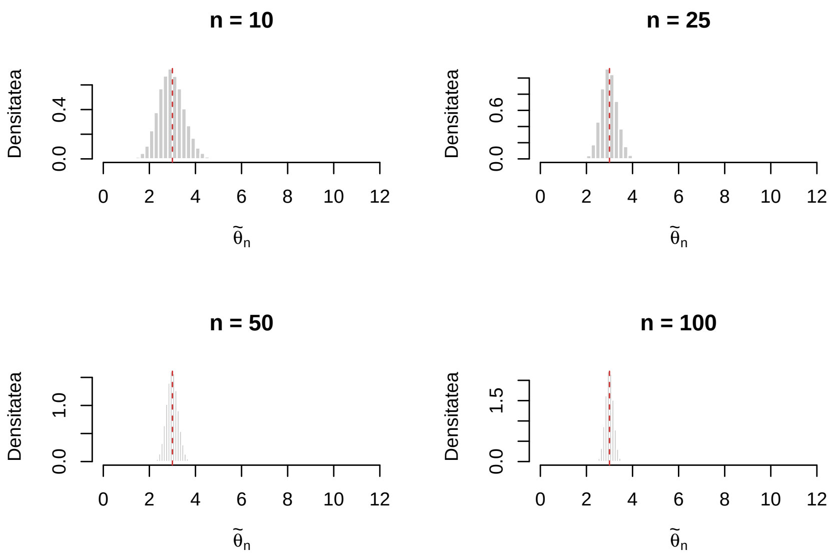


Ce se întâmplă dacă în loc de $\hat{\theta}_n$ considerăm estimatorul $\tilde{\theta}_n = \bar{X}_n$ sau estimatorul $\dot{\theta}_n = \sqrt{\bar{X}_n S_n^2}$?

Pentru $\tilde{\theta}_n$ avem

```r
pois_est2 = function(n, theta, S){
  # initializare
  sigma2 = numeric(S)

  for (i in 1:S){
    x = rpois(n, theta)
    sigma2[i] = mean(x)
  }
  # afisam varianta estimatorului
  print(paste0("Pentru n = ", n," varianta estimatorului este ", var(sigma2)))
  return(sigma2)
}

theta = 3

par(mfrow=c(2,2))
a1 = pois_est2(10, theta, 50000)
[1] "Pentru n = 10 varianta estimatorului este 0.299715966123322"
a2 = pois_est2(25, theta, 50000)
[1] "Pentru n = 25 varianta estimatorului este 0.121187195338147"
a3 = pois_est2(50, theta, 50000)
[1] "Pentru n = 50 varianta estimatorului este 0.0600944942898858"
a4 = pois_est2(100, theta, 50000)
[1] "Pentru n = 100 varianta estimatorului este 0.0300141779183984"


hist(a1, freq=FALSE, xlab=expression(tilde(theta)[n]),
     col="gray80", border="white", main = "n = 10", xlim = c(0,12),
     ylab = "Densitatea")
```

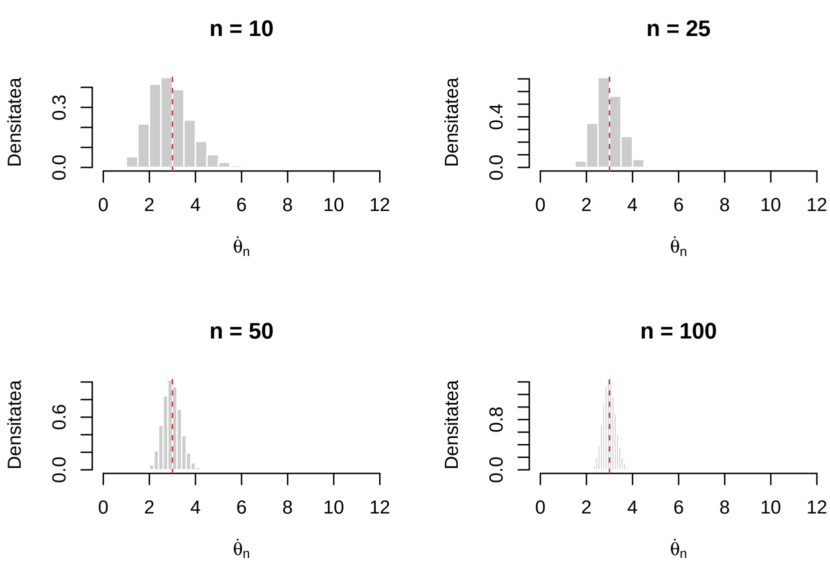
```
abline(v=theta, col = "brown3", lty = 2)

hist(a2, freq=FALSE, xlab=expression(tilde(theta)[n]),
     col="gray80", border="white", main = "n = 25", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)

hist(a3, freq=FALSE, xlab=expression(tilde(theta)[n]),
     col="gray80", border="white", main = "n = 50", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)

hist(a4, freq=FALSE, xlab=expression(tilde(theta)[n]),
     col="gray80", border="white", main = "n = 100", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)
```



iar pentru $\dot{\theta}_n$ avem

```
pois_est3 = function(n, theta, S){
  # initializare
  sigma3 = numeric(S)

  for (i in 1:S){
    x = rpois(n, theta)
    sigma3[i] = sqrt(mean(x)*var(x))
  }
  # afisam varianta estimatorului
  print(paste0("Pentru n = ", n," varianta estimatorului este ", var(sigma3)))
  return(sigma3)
}

theta = 3
```

```
par(mfrow=c(2,2))

a1 = pois_est3(10, theta, 50000)
[1] "Pentru n = 10 varianta estimatorului este 0.774312512559173"
a2 = pois_est3(25, theta, 50000)
[1] "Pentru n = 25 varianta estimatorului este 0.300199856748503"
a3 = pois_est3(50, theta, 50000)
[1] "Pentru n = 50 varianta estimatorului este 0.151213574285"
a4 = pois_est3(100, theta, 50000)
[1] "Pentru n = 100 varianta estimatorului este 0.0752523336106622"


hist(a1, freq=FALSE, xlab=expression(dot(theta)[n]),
     col="gray80", border="white", main = "n = 10", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)

hist(a2, freq=FALSE, xlab=expression(dot(theta)[n]),
     col="gray80", border="white", main = "n = 25", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)

hist(a3, freq=FALSE, xlab=expression(dot(theta)[n]),
     col="gray80", border="white", main = "n = 50", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)

hist(a4, freq=FALSE, xlab=expression(dot(theta)[n]),
     col="gray80", border="white", main = "n = 100", xlim = c(0,12),
     ylab = "Densitatea")
abline(v=theta, col = "brown3", lty = 2)
```



2 Estimare prin metoda verosimilității maxime

2.1 Exemplu: EVM nu este întotdeauna media eșantionului chiar dacă $\mathbb{E}_{\theta}[\hat{\theta}_n] = \theta$

Fie $X_1, X_2, \dots, X_n$ un eșantion de talie n dintr-o populație Laplace $L(\theta, c)$ a cărei densitate este dată de formula

$$f_{\theta,c}(x) = \frac{1}{2c} e^{-\frac{|x-\theta|}{c}}, \quad -\infty < x < \infty$$

a) Ilustrați grafic densitatea și funcția de repartiție a repartiției Laplace pentru diferite valori ale parametrilor θ (de locație) și c (de scală), e.g. $\theta \in \{0, 3\}$ și $c \in \{1, 2, 3, 4\}$.

b) Determinați estimatorul de verosimilitate maximă $\hat{\theta}_n$ pentru θ.

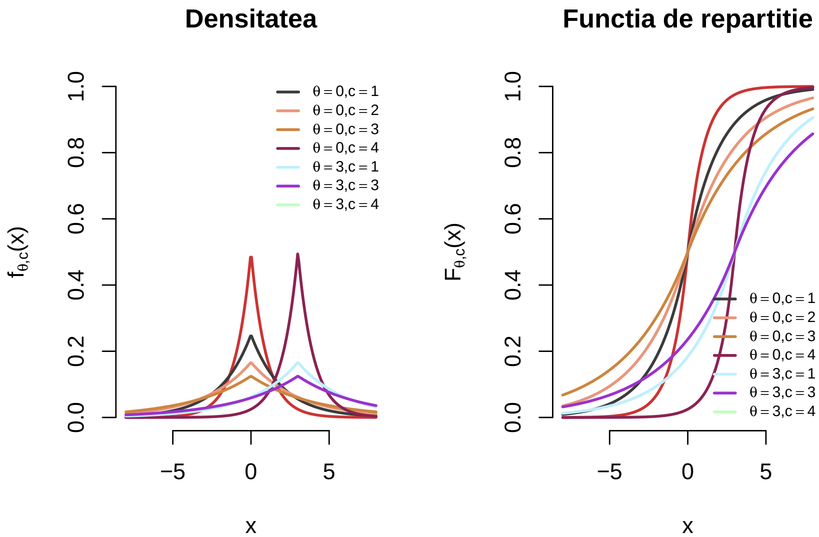
a) Se poate arăta cu ușurință că funcția de repartiție a repartiției Laplace $L(\theta, c)$ este

$$F_{\theta,c}(x) = \frac{1}{2} + \frac{1}{2}\operatorname{sgn}(x-\theta)\left(1 - e^{-\frac{|x-\theta|}{c}}\right) = \begin{cases} \frac{1}{2}e^{-\frac{|x-\theta|}{c}}, & x < \theta \\ 1 - \frac{1}{2}e^{-\frac{|x-\theta|}{c}}, & x \geq \theta \end{cases}$$

Ilustrarea grafică a densității și a funcției de repartiție pentru repartiția Laplace:

```r
# Cream functia de densitate si de repartitie

dLaplace = function (x, mu = 0, b = 1)
{
    d <- exp(-abs(x - mu)/b)/(2 * b)
    return(d)
}

pLaplace = function (q, mu = 0, b = 1)
{
    x <- q - mu
    0.5 + 0.5 * sign(x) * (1 - exp(-abs(x)/b))
}

# Generam graficele
pars = matrix(c(0, 1, 0, 2, 0, 3, 0, 4, 3, 1, 3, 3, 3, 4),
              ncol = 2, byrow = TRUE)

x = seq(-8, 8, length.out = 250)

set.seed(1234)
cols = sample(colors(), nrow(pars))

par(mfrow = c(1, 2))
# densitatile
plot(x, dLaplace(x, mu = pars[1, 1], b = pars[1, 2]),
     xlab = "x",
     ylab = TeX("$f_{\\theta, c}(x)$"),
     ylim = c(0,1),
     col = "brown3",
     lwd = 2, type = "l",
```

```r
     bty = "n",
     main = "Densitatea")

for (i in seq(nrow(pars)-1)){
  mu = pars[i+1, 1]
  b = pars[i+1, 2]

  y = dLaplace(x, mu = mu, b = b)

  lines(x, y, lwd = 2,
        col = cols[i])
}

legend("topright",
       legend = TeX(paste0("$\\theta = ", pars[,1], ", \\c = ", pars[,2], "$")),
       col = cols,
       lwd = rep(2, nrow(pars)),
       bty = "n",
       cex = 0.7,
       seg.len = 1.5)

# functiile de repartitie
plot(x, pLaplace(x, mu = pars[1, 1], b = pars[1, 2]),
     xlab = "x",
     ylab = TeX("$F_{\\theta, c}(x)$"),
     ylim = c(0,1),
     col = "brown3",
     lwd = 2, type = "l",
     bty = "n",
     main = "Functia de repartitie")

for (i in seq(nrow(pars)-1)){
  mu = pars[i+1, 1]
  b = pars[i+1, 2]

  y = pLaplace(x, mu = mu, b = b)

  lines(x, y, lwd = 2,
        col = cols[i])
}

legend("bottomright",
       legend = TeX(paste0("$\\theta = ", pars[,1], ", \\c = ", pars[,2], "$")),
       col = cols,
       lwd = rep(2, nrow(pars)),
       bty = "n",
       cex = 0.7,
       seg.len = 1.5)
```



b) Pentru a determina estimatorul de verosimilitate maximă să observăm că funcția de verosimilitate este

$$L(\theta|\mathbf{X}) = \prod_{i=1}^{n}\left(\frac{1}{2c}e^{-\frac{|X_i-\theta|}{c}}\right) = \frac{1}{(2c)^n}e^{-\sum_{i=1}^{n}\frac{|X_i-\theta|}{c}}$$

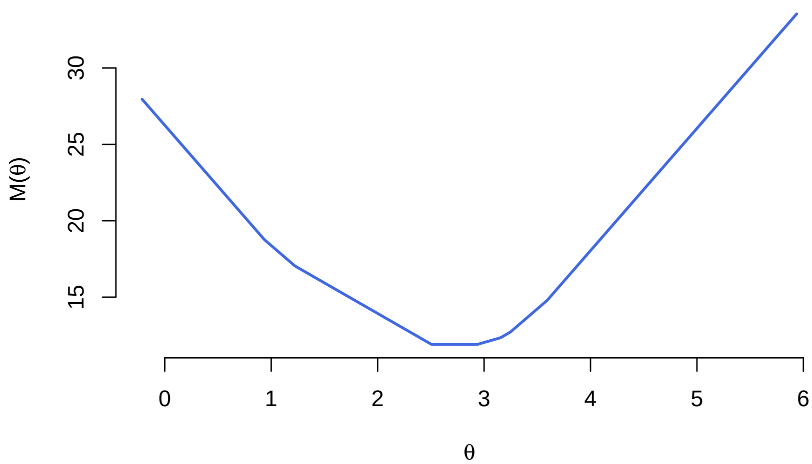
și acesta ia valoarea maximă pentru toate valorile lui θ care minimizează funcția de la exponent

$$M(\theta) = \sum_{i=1}^{n}|X_i - \theta| = \sum_{i=1}^{n}|X_{(i)} - \theta|,$$

unde $x_{(i)}$ este statistica de ordine de rang i. Se poate vedea că funcția $M(\theta)$ este continuă și afină pe porțiuni din figura de mai jos (pentru un eșantion de talie 10 dintr-o populație $L(3, 1)$ - creați o funcție care vă permite să generați observații repartizate Laplace).

```
rLaplace = function (n, mu = 0, b = 1)
{
    u <- runif(n) - 0.5
    x <- mu - b * sign(u) * log(1 - 2 * abs(u))
    return(x)
}

theta0 = 3
b = 1

set.seed(333)
x = rLaplace(10, mu = theta0, b = b)

M_theta = function(x, theta){
  sapply(theta, function(t){sum(abs(x-t))})
}

theta = seq(min(x), max(x), length.out = 500)
M = M_theta(x, theta)
```

```r
plot(theta, M, type = "l",
     xlab = TeX("$\\theta$"),
     ylab = TeX("$M(\\theta)$"),
     bty = "n", lwd = 2,
     col = "royalblue")
```



Observăm că dacă θ se află între statistica de ordine de rang m și cea de rang $m+1$, i.e. $X_{(m)} \leq \theta \leq X_{(m+1)}$, atunci am avea că $X_{(i)} \leq X_{(m)} \leq \theta$ dacă $i \leq m$ și $\theta \leq X_{(m+1)} \leq X_{(i)}$ dacă $m+1 \leq i \leq n$, prin urmare

$$M(\theta) = \sum_{i=1}^{n} |X_{(i)} - \theta| = \sum_{i=1}^{m} (\theta - X_{(i)}) + \sum_{i=m+1}^{n} (X_{(i)} - \theta)$$

deci dacă $X_{(m)} < \theta < X_{(m+1)}$ atunci

$$\frac{d}{d\theta} M(\theta) = m - (n-m) = 2m - n.$$

Astfel, $M'(\theta) < 0$ (și $M(\theta)$ este descrescătoare) dacă $m < \frac{n}{2}$ și $M'(\theta) > 0$ (și $M(\theta)$ este crescătoare) dacă $m > \frac{n}{2}$. Dacă $n = 2k+1$ este impar, atunci $\frac{n}{2} = k + \frac{1}{2}$ iar $M(\theta)$ este strict descrescătoare dacă $\theta < X_{(k+1)}$ și strict crescătoare dacă $\theta > X_{(k+1)}$ de unde deducem că minimul se atinge pentru $\theta = X_{(k+1)}$.

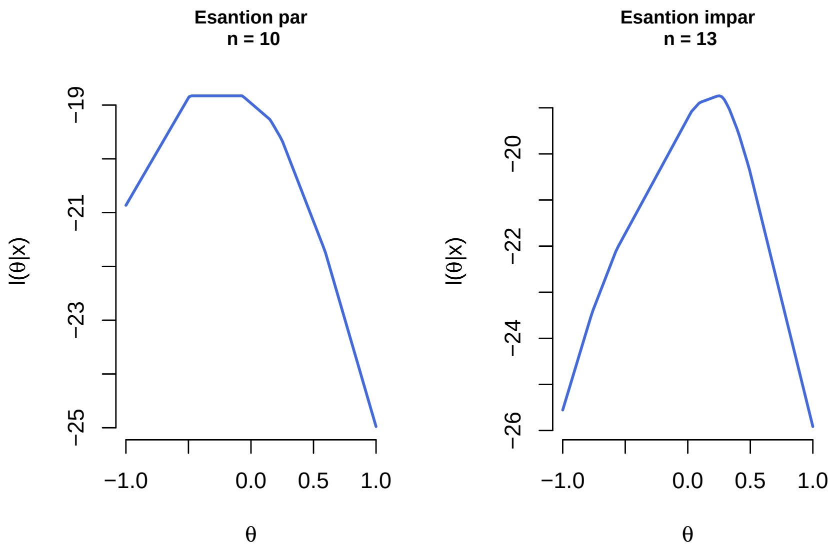
Dacă $n = 2k$ este par atunci, raționând asemănător, deducem că $M(\theta)$ este minimizată pentru orice punct din intervalul $(X_{(k)}, X_{(k+1)})$, deci orice punct din acest interval va maximiza și funcția de verosimilitate. Prin convenție alegem estimatorul de verosimilitate maximă să fie mijlocul acestui interval, i.e. $\theta = \frac{X_{(k)} + X_{(k+1)}}{2}$.

Prin urmare am găsit că estimatorul de verosimilitate maximă este mediana eșantionului

$$\hat{\theta}_n = \begin{cases} X_{\left(\frac{n+1}{2}\right)}, & n \text{ impar} \\ \frac{X_{\left(\frac{n}{2}\right)} + X_{\left(\frac{n}{2}+1\right)}}{2}, & n \text{ par} \end{cases}$$

Mai jos avem ilustrat logaritmul funcției de verosimilitate pentru un eșantion de volum par (stânga) și unul de volum impar (dreapta):

```
theta0 = 0
b = 1

logLikelihoodLaplace = function(x, theta, b){
  sapply(theta, function(t){
    -length(x)*log(2*b)-sum(abs(x-t))})
}

par(mfrow = c(1,2))

# Esantion par
n = 10

set.seed(333)
x = rLaplace(n, mu = theta0, b = b)

theta = seq(-1,1, length.out = 100)

y = logLikelihoodLaplace(x, theta, b)

plot(theta, y, type = "l",
     bty = "n", lwd = 2,
     col = "royalblue",
     xlab = TeX("$\\theta$"),
     ylab = TeX("$l(\\theta | x)$"),
     main = paste0("Esantion par\n n = ", n),
     cex.main = 0.8)

# Esantion impar
n = 13

set.seed(1234)
x = rLaplace(n, mu = theta0, b = b)

theta = seq(-1,1, length.out = 100)

y = logLikelihoodLaplace(x, theta, b)

plot(theta, y, type = "l",
     bty = "n", lwd = 2,
     col = "royalblue",
     xlab = TeX("$\\theta$"),
     ylab = TeX("$l(\\theta | x)$"),
     main = paste0("Esantion impar\n n = ", n),
     cex.main = 0.8)
```



2.2 Exemplu de EVM determinat prin soluții numerice

Fie $X_1, X_2, \ldots, X_n$ un eșantion de talie n dintr-o populație logistică a cărei densitate este dată de formula

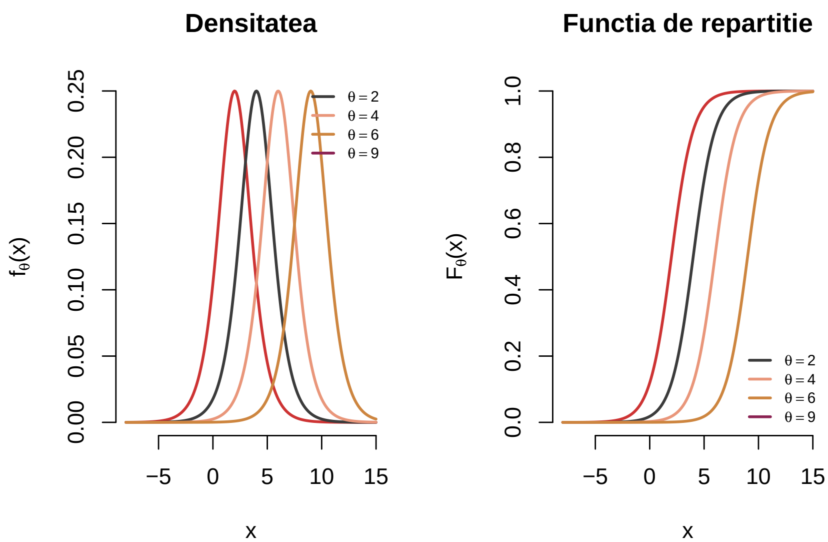
$$f_\theta(x) = \frac{e^{-(x-\theta)}}{\left(1 + e^{-(x-\theta)}\right)^2}, \quad x \in \mathbb{R},\, \theta \in \mathbb{R}$$

Determinați estimatorul de verosimilitate maximă $\hat{\theta}_n$ pentru θ.

Densitatea de repartiție și funcția de repartiție a repartiției logistice sunt ilustrate mai jos (în R se folosesc funcțiile: `rlogis`, `dlogis`, `plogis` și respectiv `qlogis`):

```
# Generam graficele
pars = c(2, 4, 6, 9)

x = seq(-8, 15, length.out = 250)

set.seed(1234)
cols = sample(colors(), length(pars))

par(mfrow = c(1, 2))
# densitatile
plot(x, dlogis(x, location = pars[1]),
     xlab = "x",
     ylab = TeX("$f_{\\theta}(x)$"),
     # ylim = c(0,1),
     col = "brown3",
     lwd = 2, type = "l",
     bty = "n",
     main = "Densitatea")
```

```
for (i in seq(length(pars)-1)){
  location = pars[i+1]

  y = dlogis(x, location = location)

  lines(x, y, lwd = 2,
        col = cols[i])
}

legend("topright",
       legend = TeX(paste0("$\\theta = ", pars, "$")),
       col = cols,
       lwd = rep(2, length(pars)),
       bty = "n",
       cex = 0.7,
       seg.len = 1.5)

# functiile de repartitie
plot(x, plogis(x, location = pars[1]),
     xlab = "x",
     ylab = TeX("$F_{\\theta}(x)$"),
     ylim = c(0,1),
     col = "brown3",
     lwd = 2, type = "l",
     bty = "n",
     main = "Functia de repartitie")

for (i in seq(length(pars)-1)){
  location = pars[i+1]

  y = plogis(x, location = location)

  lines(x, y, lwd = 2,
        col = cols[i])
}

legend("bottomright",
       legend = TeX(paste0("$\\theta = ", pars, "$")),
       col = cols,
       lwd = rep(2, length(pars)),
       bty = "n",
       cex = 0.7,
       seg.len = 1.5)
```



Observăm că funcția de verosimilitate este dată de

$$L(\theta|\mathbf{x}) = \prod_{i=1}^{n} f_\theta(x_i) = \prod_{i=1}^{n} \frac{e^{-(x_i-\theta)}}{\left(1+e^{-(x_i-\theta)}\right)^2}$$

iar logaritmul funcției de verosimilitate este

$$l(\theta|\mathbf{x}) = \sum_{i=1}^{n} \log f_\theta(x_i) = n\theta - n\bar{x}_n - 2\sum_{i=1}^{n} \log\left(1+e^{-(x_i-\theta)}\right).$$

Pentru a găsi valoarea lui θ care maximizează logaritmul funcției de verosimilitate și prin urmare a funcției de verosimilitate trebuie să rezolvăm ecuația $l'(\theta|\mathbf{x}) = 0$, unde derivata lui $l(\theta|\mathbf{x})$ este

$$l'(\theta|\mathbf{x}) = n - 2\sum_{i=1}^{n} \frac{e^{-(x_i-\theta)}}{1+e^{-(x_i-\theta)}}$$

ceea ce conduce la ecuația

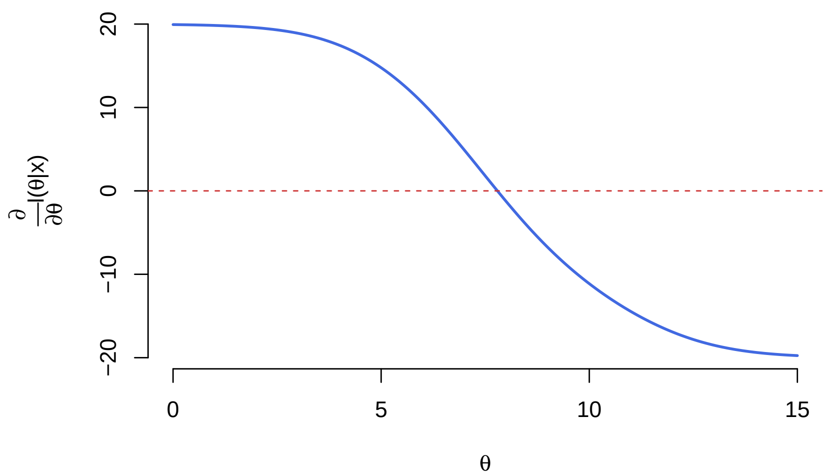
$$\sum_{i=1}^{n} \frac{e^{-(x_i-\theta)}}{1+e^{-(x_i-\theta)}} = \frac{n}{2} \tag{$\star$}$$

Chiar dacă această ecuație nu se simplifică, se poate arăta că această ecuația admite soluție unică. Observăm că derivata parțiala a membrului drept în ($\star$) devine

$$\frac{\partial}{\partial\theta} \sum_{i=1}^{n} \frac{e^{-(x_i-\theta)}}{1+e^{-(x_i-\theta)}} = \sum_{i=1}^{n} \frac{e^{-(x_i-\theta)}}{\left(1+e^{-(x_i-\theta)}\right)^2} > 0$$

ceea ce arată că membrul stâng este o funcție strict crescătoare în θ. Cum membrul stâng în ($\star$) tinde spre 0 atunci când $\theta \to -\infty$ și spre n pentru $\theta \to \infty$ deducem că ecuația ($\star$) admite soluție unică (vezi graficul de mai jos).

```r
set.seed(112)
n = 20
x = rlogis(n, location = 7.5)

# derivata logaritmului functiei de verosimilitate
dLogLogistic = function(n, x, theta){
  sapply(theta, function(t){
    y = exp(-(x - t))
    n - 2*sum(y/(1+y))
  })
}

theta = seq(0, 15, length.out = 250)

mar.default <- c(5,4,4,2) + 0.1
par(mar = mar.default + c(0, 1.2, 0, 0))

plot(theta, dLogLogistic(n, x, theta), type = "l",
     col = "royalblue", lwd = 2,
     bty = "n",
     xlab = TeX("$\\theta$"),
     ylab = TeX("$\\frac{\\partial}{\\partial \\theta} l(\\theta | x)$"))

abline(h = 0, col = "brown3",
       lty = 2)
```



Cum nu putem găsi o soluție a ecuației $l'(\theta|\frown) = 0$ sub formă compactă, este necesar să apelăm la metode numerice. O astfel de metodă numerică este binecunoscuta metodă a lui Newton-Raphson. Metoda presupune să începem cu o valoare (soluție) inițială $\hat{\theta}^{(0)}$ și să alegem, plecând de la aceasta, o nouă valoare $\hat{\theta}^{(1)}$ definită prin

$$\hat{\theta}^{(1)} = \hat{\theta}^{(0)} - \frac{l'\left(\hat{\theta}^{(0)}\right)}{l''\left(\hat{\theta}^{(0)}\right)},$$

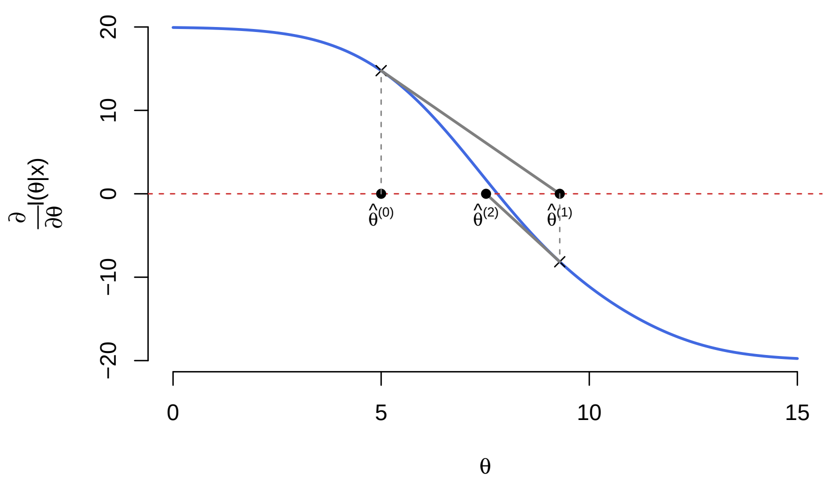
adică $\hat{\theta}^{(1)}$ este intersecția cu axa absciselor a tangentei în punctul $\left(\hat{\theta}^{(0)}, l'\left(\hat{\theta}^{(0)}\right)\right)$ la graficul funcției $l'(\theta)$. Ideea este de a itera procesul până când soluția converge, cu alte cuvinte pornind de la o valoare *rezonabilă* de start $\hat{\theta}^{(0)}$ la pasul $k+1$ avem

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} - \frac{l'\left(\hat{\theta}^{(k)}\right)}{l''\left(\hat{\theta}^{(k)}\right)}$$

și oprim procesul atunco când k este suficient de mare și/sau $\left|\hat{\theta}^{(k+1)} - \hat{\theta}^{(k)}\right|$ este suficient de mic. Următorul grafic ilustrează grafic algoritmul lui Newton:

```r
set.seed(112)
n = 20
x = rlogis(n, location = 7.5)

# derivata logaritmului functiei de verosimilitate
dLogLogistic = function(n, x, theta){
  sapply(theta, function(t){
    y = exp(-(x - t))
    n - 2*sum(y/(1+y))
  })
}

theta = seq(0, 15, length.out = 250)

mar.default <- c(5,4,4,2) + 0.1
par(mar = mar.default + c(0, 1.2, 0, 0))

plot(theta, dLogLogistic(n, x, theta), type = "l",
     col = "royalblue", lwd = 2,
     bty = "n",
     xlab = TeX("$\\theta$"),
     ylab = TeX("$\\frac{\\partial}{\\partial \\theta} l(\\theta | x)$"))

abline(h = 0, col = "brown3",
       lty = 2)

# ilustrarea metodei Newton

dl = function(theta) n - 2 * sum(exp(theta - x) / (1 + exp(theta - x)))
ddl = function(theta) {-2 * sum(exp(theta - x) / (1 + exp(theta - x))^2)}

x0 = 5 # punctul de start

points(x0, 0, pch = 16, col = "black")
text(x0, 0, labels = TeX("$\\hat{\\theta}^{(0)}$"), pos = 1, cex = 0.8)
segments(x0, 0, x0, dl(x0), lty = 2, col = "grey50")
points(x0, dl(x0), pch = 4)

x1 = x0 - dl(x0)/ddl(x0)

segments(x0, dl(x0), x1, 0, lty = 1, lwd = 2, col = "grey50")
points(x1, 0, pch = 16, col = "black")
```

```
text(x1, 0, labels = TeX("$\\hat{\\theta}^{(1)}$"), pos = 1, cex = 0.8)
segments(x1, 0, x1, dl(x1), lty = 2, col = "grey50")
points(x1, dl(x1), pch = 4)

x2 = x1 - dl(x1)/ddl(x1)

segments(x1, dl(x1), x2, 0, lty = 1, lwd = 2, col = "grey50")
points(x2, 0, pch = 16, col = "black")
text(x2, 0, labels = TeX("$\\hat{\\theta}^{(2)}$"), pos = 1, cex = 0.8)
```



Obs: Singurul lucru care se schimbă atunci când trecem de la scalar la vector, este funcția $l(\theta)$ care acum este o funcție de $p > 1$ variabile, $\theta = (\theta_1, \theta_2, \dots, \theta_p)^\intercal \in \mathbb{R}^p$. În acest context $l'(\theta)$ este un vector de derivate parțiale iar $l''(\theta)$ este o matrice de derivate parțiale de ordin doi. Prin urmare itarațiile din metoda lui Newton sunt

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} - \left[l''\left(\hat{\theta}^{(k)}\right)\right]^{-1} l'\left(\hat{\theta}^{(k)}\right)$$

unde $[\cdot]^{-1}$ este pseudoinversa unei matrici.

Funcția de mai jos implementează metoada lui Newton pentru cazul multidimensional:

```
# Metoda lui Newton

newton <- function(f, df, x0, eps=1e-08, maxiter=1000, ...) {
  # in caz ca nu e incarcat pachetul sa putem accesa pseudoinversa
  if(!exists("ginv")) library(MASS)

  x <- x0
  k <- 0

  repeat {
    k <- k + 1

    x.new <- x - as.numeric(ginv(df(x, ...)) %*% f(x, ...))
```

```
    if(mean(abs(x.new - x)) < eps | k >= maxiter) {
      if(k >= maxiter) warning("S-a atins numarul maxim de iteratii!")
      break
    }
    x <- x.new
  }
  out <- list(solution = x.new, value = f(x.new, ...), iter = k)

  return(out)
}
```

Să presupunem că am observat următorul eșantion de talie 20 din repartiția logistică:

```
 [1]  6.996304  9.970107 12.304991 11.259549  6.326912  5.378941  4.299639
 [8]  8.484635  5.601117  7.094335  6.324731  6.868456  9.753360  8.042095
[15]  8.227830 10.977982  7.743096  7.722159  8.562884  6.968356
```

```
set.seed(112)
x = rlogis(20, location = 7.5)

n = length(x)
dl = function(theta) n - 2 * sum(exp(theta - x) / (1 + exp(theta - x)))
ddl = function(theta) {-2 * sum(exp(theta - x) / (1 + exp(theta - x))^2)}

logis.newton = newton(dl, ddl, median(x))
```

și aplicând metoda lui Newton găsim estimatorul de verosimilitate maximă $\hat{\theta}_n = 7.7933$ după numai 3 iterații (datele au fost simulate folosin $\theta = 7.5$).