

Lista 1 - raport

Erwin Jasic

3 listopada 2020

Zadanie 1

```
#zad1
n <- 50
u <- runif(n)
grid <- 0:n/n
w <- dnorm(qnorm(grid[-(n + 1)])) - dnorm(qnorm(grid[-1]))
estymator <- function(x, u, w)
{
  th1a <- mean(x)
  th2a <- median(x)
  th3a <- (u/sum(u))%%x
  th4a <- w%%sort(x)
  tha <- c(th1a, th2a, th3a, th4a)
  return(tha)
}

M = 10000
A = matrix(0, M, 4)
B = matrix(0, M, 4)
C = matrix(0, M, 4)
paramA = matrix(0, 4, 3)
paramB = matrix(0, 4, 3)
paramC = matrix(0, 4, 3)

for(i in 1:M) {
  a <- rnorm(n, 1, 1)
  b <- rnorm(n, 4, 1)
  c <- rnorm(n, 1, 2)
  A[i, ] <- estymator(a, u, w)
  B[i, ] <- estymator(b, u, w)
  C[i, ] <- estymator(c, u, w)
}

for(j in 1:4) {
  paramA[j, 1] <- var(A[, j])
  paramA[j, 3] <- mean(A[, j]) - 1
  paramA[j, 2] <- paramA[j, 1] + paramA[j, 3] ^ 2
  paramB[j, 1] <- var(B[, j])
  paramB[j, 3] <- mean(B[, j]) - 1
  paramB[j, 2] <- paramB[j, 1] + paramB[j, 3] ^ 2
  paramC[j, 1] <- var(C[, j])
```

```

paramC[j, 3] <- mean(C[, j]) - 1
paramC[j, 2] <- paramC[j, 1] + paramC[j, 3] ^ 2
}
row.names(paramA) <- c("th1", "th2", "th3", "th4")
kable(paramA, col.names = c("wariancja", "obciazenie", "blad sr kwadrat"), caption = "Norm(50, 1, 1)")

```

Table 1: Norm(50, 1, 1)

	wariancja	obciazenie	blad sr kwadrat
th1	0.0200241	0.0200250	-0.0009855
th2	0.0309718	0.0309742	-0.0015545
th3	0.0255790	0.0255791	-0.0003019
th4	0.0097202	0.0105038	-0.0279941

```

row.names(paramB) <- c("th1", "th2", "th3", "th4")
kable(paramB, col.names = c("wariancja", "obciazenie", "blad sr kwadrat"), caption = "Norm(50, 4, 1)")

```

Table 2: Norm(50, 4, 1)

	wariancja	obciazenie	blad sr kwadrat
th1	0.0197338	9.0242139	3.0007466
th2	0.0308146	9.0339229	3.0005180
th3	0.0250917	9.0335027	3.0014015
th4	0.0099269	0.0106475	-0.0268451

```

row.names(paramC) <- c("th1", "th2", "th3", "th4")
kable(paramC, col.names = c("wariancja", "obciazenie", "blad sr kwadrat"), caption = "Norm(50, 1, 2)")

```

Table 3: Norm(50, 1, 2)

	wariancja	obciazenie	blad sr kwadrat
th1	0.0802587	0.0802587	0.0000451
th2	0.1216241	0.1216250	0.0009543
th3	0.1009997	0.1009999	0.0005194
th4	0.0385206	0.9226376	0.9402750

W zadaniu pierwszym okreslamy ktory estymator jest najlepszy, jest to taki, ktory cechuje sie najmniejsza wariancja oraz najmniejszym obciazeniem. Zatem dla podpunktu a) najlepszym estymatorem jest th4, dla b) jest to rowniez th4, a dla c) jest to th1, poniewaz th4 ma wzglednie duze obciazenie.

Zadanie 5

#zad5

```

lprim <- function(theta, x, sigma) {
  n <- length(x)/sigma
  n - (2/sigma) * sum(exp((-x+theta)/sigma)/(1+exp(-(x-theta)/sigma)))
}

```

```

lbis <- function(theta, x, sigma) {
  (-2/sigma^2) * sum(exp(-(x-theta)/sigma)/(1+exp(-(x-theta)/sigma))^2)
}

result <- function(x, theta0, eps, sigma) {
  vector_th <- c(0, theta0)
  theta1 <- theta0
  i = 1
  while(abs(vector_th[i + 1] - vector_th[i]) > eps) {
    theta1 <- vector_th[i + 1] - lprim(vector_th[i + 1], x, sigma)/lbis(vector_th[i + 1], x, sigma)
    vector_th <- c(vector_th, theta1)
    i = i + 1
  }
  c(theta1, i)
}

x <- rlogis(50, 1, 1)
y <- rlogis(50, 4, 1)
z <- rlogis(50, 1, 2)

# sprawdzam jak wybor epsilon wpływa na theta1 oraz na liczbe krokow
result(x, median(x), 0.1, 1)

## [1] 1.444751 3.000000
result(x, median(x), 0.01, 1)

## [1] 1.444751 3.000000
result(x, median(x), 0.00001, 1)

## [1] 1.444751 4.000000
result(y, median(y), 0.1, 1)

## [1] 3.773168 3.000000
result(y, median(y), 0.01, 1)

## [1] 3.773168 3.000000
result(y, median(y), 0.00001, 1)

## [1] 3.773168 4.000000
result(z, median(z), 0.1, 2)

## [1] 0.6906926 2.0000000
result(z, median(z), 0.01, 2)

## [1] 0.6906806 3.0000000
result(z, median(z), 0.00001, 2)

## [1] 0.6906806 4.0000000
vector_x <- c()
vector_y <- c()

```

```

vector_z <- c()
for(i in 1:10000) {
  x <- rlogis(50, 1, 1)
  y <- rlogis(50, 4, 1)
  z <- rlogis(50, 1, 2)
  vector_x <- c(vector_x, result(x, median(x), 0.01, 1)[1])
  vector_y <- c(vector_y, result(y, median(y), 0.01, 1)[1])
  vector_z <- c(vector_z, result(z, median(z), 0.01, 2)[1])
}
wariancjax <- var(vector_x)
obciazeniex <- mean(vector_x) - 1
blad_sr_kwadratx <- wariancjax + obciazeniex ^ 2

wariancjay <- var(vector_y)
obciazeniey <- mean(vector_y) - 4
blad_sr_kwadraty <- wariancjay + obciazeniey ^ 2

wariancjaz <- var(vector_z)
obciazeniez <- mean(vector_z) - 1
blad_sr_kwadratx <- wariancjaz + obciazeniez ^ 2

wektor5x <- c(wariancjax, obciazeniex, blad_sr_kwadratx)
wektor5y <- c(wariancjay, obciazeniey, blad_sr_kwadratx)
wektor5z <- c(wariancjaz, obciazeniez, blad_sr_kwadratx)
macierz5 <- rbind(wektor5x, wektor5y, wektor5z)
row.names(macierz5) <- c("(50,1,1)", "(50,4,1)", "(50,1,2)")
kable(macierz5, col.names = c("wariancja", "obciazenie", "blad sr kwadrat"), caption = "Logis")

```

Table 4: Logis

	wariancja	obciazenie	blad sr kwadrat
(50,1,1)	0.0607065	-0.0013162	0.0607082
(50,4,1)	0.0611347	0.0028217	0.0611427
(50,1,2)	0.2433807	-0.0024452	0.2433867

Podsumowujac zadanie 5, widzimy, ze dobor epsilon wplywa na theta1 oraz ilosc krokow. Im mniejszy epsilon tym wieksza ilosc krokow w algorytmie, a co za tym idzie lepsze oszacowanie estymatora. Dla powtorzenia eksperymentu 10000 razy przyjalam epsilon = 0.01, poniewaz przy takim epsilon, estymator jest juz dosc dobrze oszacowany. Z tabelki mozemy odczytac, ze jesli sigma wynosi 1 to wariancja, obciazenie oraz blad srednio-kwadratowy sa niewielkie w porownaniu do rozkladu logistycznego z sigma wynoszaca 2 (ostatni wiersz tabeli).

Zadanie 6

```

#zad6

lprim1 <- function(theta, x, sigma) {
  2 * sum((x - theta)/(sigma^2 + (x - theta)^2))
}

lbis1 <- function(theta, x, sigma) {
  2 * sum(((x - theta)^2 - sigma^2)/((sigma^2 + (x - theta)^2)^2))
}

```

```

}

result1 <- function(x, theta0, eps, sigma) {
  vector_th <- c(0, theta0)
  theta1 <- theta0
  i = 1
  while(abs(vector_th[i + 1] - vector_th[i]) > eps) {
    theta1 <- vector_th[i + 1] - lprim1(vector_th[i + 1], x, sigma)/lbis1(vector_th[i + 1], x, sigma)
    vector_th <- c(vector_th, theta1)
    i = i + 1
  }
  c(theta1, i)
}

x <- rcauchy(50, 1, 1)
y <- rcauchy(50, 4, 1)
z <- rcauchy(50, 1, 2)

# sprawdzam jak wybor epsilon wpływa na theta1 oraz na liczbe krokow
result1(x, median(x), 0.1, 1)

## [1] 1.055855 2.000000
result1(x, median(x), 0.01, 1)

## [1] 1.05501 3.00000
result1(x, median(x), 0.00001, 1)

## [1] 1.05501 4.00000
result1(y, median(y), 0.1, 1)

## [1] 4.167245 2.000000
result1(y, median(y), 0.01, 1)

## [1] 4.167211 3.000000
result1(y, median(y), 0.00001, 1)

## [1] 4.167211 4.000000
result1(z, median(z), 0.1, 2)

## [1] 0.934464 3.000000
result1(z, median(z), 0.01, 2)

## [1] 0.934464 3.000000
result1(z, median(z), 0.00001, 2)

## [1] 0.9344637 4.000000
vector_x <- c()
vector_y <- c()
vector_z <- c()
for(i in 1:10000) {

```

```

x <- rcauchy(50, 1, 1)
y <- rcauchy(50, 4, 1)
z <- rcauchy(50, 1, 2)
vector_x <- c(vector_x, result1(x, median(x), 0.01, 1)[1])
vector_y <- c(vector_y, result1(y, median(y), 0.01, 1)[1])
vector_z <- c(vector_z, result1(z, median(z), 0.01, 2)[1])
}

wariancjax <- var(vector_x)
obciazeniex <- mean(vector_x) - 1
blad_sr_kwadratx <- wariancjax + obciazeniex ^ 2

wariancjay <- var(vector_y)
obciazeniey <- mean(vector_y) - 4
blad_sr_kwadrat_y <- wariancjay + obciazeniey ^ 2

wariancjaz <- var(vector_z)
obciazeniez <- mean(vector_z) - 1
blad_sr_kwadrat_z <- wariancjaz + obciazeniez ^ 2

wektor6x <- c(wariancjax, obciazeniex, blad_sr_kwadratx)
wektor6y <- c(wariancjay, obciazeniey, blad_sr_kwadrat_y)
wektor6z <- c(wariancjaz, obciazeniez, blad_sr_kwadrat_z)
macierz6 <- rbind(wektor6x, wektor6y, wektor6z)
row.names(macierz6) <- c("(50,1,1)", "(50,4,1)", "(50,1,2)")
kable(macierz6, col.names = c("wariancja", "obciazenie", "blad sr kwadrat"), caption = "Cauchy")

```

Table 5: Cauchy

	wariancja	obciazenie	blad sr kwadrat
(50,1,1)	0.0421880	-0.0001204	0.0421880
(50,4,1)	0.0430333	0.0002738	0.0430334
(50,1,2)	0.1678880	0.0027466	0.1678956

Podobnie jak w zadaniu 5, w zadaniu 6 chcemy oszacowac estymator największej wiarygodności, tym razem dla rozkładów Cauchy'ego. Tak samo jak w poprzednim zadaniu, im mniejszy epsilon tym lepsze oszacowanie estymatora oraz zwiększona ilość kroków. Dla powtórzenia eksperymentu 10000 razy przyjąłem epsilon = 0.01, ponieważ przy takim epsilon, estymator jest już dość dobrze oszacowany. Tak samo jak w zadaniu 5, w zadaniu 6 z tabelki możemy odczytać, że jeśli sigma wynosi 1 to wariancja, obciążenie oraz błąd średniokwadratowy są niewielkie w porównaniu do rozkładu Cauchy'ego z sigma wynoszącą 2 (ostatni wiersz tabeli).

Zadanie 7 dla zadania 1

Table 6: Norm(20, 1, 1)

	wariancja	obciazenie	blad sr kwadrat
th1	0.0498991	0.0499031	0.0019910
th2	0.0734327	0.0734435	0.0032883
th3	0.0715253	0.0715361	0.0032889
th4	0.0236760	0.0281828	-0.0671328

Table 7: Norm(20, 4, 1)

	wariancja	obciazenie	blad sr kwadrat
th1	0.0485821	9.0456518	2.9995116
th2	0.0720976	9.0706251	2.9997546
th3	0.0720624	9.0487144	2.9961061
th4	0.0235655	0.0281469	-0.0676858

Table 8: Norm(20, 1, 2)

	wariancja	obciazenie	blad sr kwadrat
th1	0.2041053	0.2041070	0.0012835
th2	0.2952802	0.2952860	0.0024097
th3	0.2933898	0.2933898	-0.0000415
th4	0.0960702	0.8417658	0.8635367

Table 9: Norm(100, 1, 1)

	wariancja	obciazenie	blad sr kwadrat
th1	0.0100932	0.0100933	0.0002192
th2	0.0156272	0.0156276	-0.0006584
th3	0.0146562	0.0146563	-0.0001920
th4	0.0048928	0.0051198	-0.0150655

Table 10: Norm(100, 4, 1)

	wariancja	obciazenie	blad sr kwadrat
th1	0.0099162	9.0162669	3.0010583
th2	0.0157994	9.0259202	3.0016863
th3	0.0145859	9.0135600	2.9998290
th4	0.0049151	0.0051185	-0.0142631

Table 11: Norm(100, 1, 2)

	wariancja	obciazenie	blad sr kwadrat
th1	0.0399893	0.0399905	-0.0010932
th2	0.0611678	0.0611678	-0.0002261
th3	0.0572105	0.0572146	-0.0020145
th4	0.0196246	0.9579489	0.9686714

Pierwsze trzy tabelki sa dla proby $n = 20$, nastepne 3 dla $n = 100$. Widzimy ze dla pierwszych trzech tabelkek wariancja jest wyraznie wieksza dla kazdej thety, porownujac wyniki z ostatnimi trzema tabelkami. W szczegolnosci dla rozkladu $\text{rnorm}(n, 1, 1)$ obciazenie oraz blad srednio-kwadratowy tez sa wieksze (tabelka 1 oraz 4, porownuje wartosci bezwzgledne). Porownujac 2 i 5 tabelke ($\text{rnorm}(n, 4, 1)$) widzimy ze dla th1, th2, th3 wyniki na obciazenie i blad srednio-kwadratowy sa podobne, natomiast dla th4 wyraznie mniejsze

sa te wartosci w tabelce piatej. Natomiast dla $\text{rnorm}(n, 1, 4)$ widzimy ze tendencja z poprzedniej analizy jest zachowana czyli dla th1 , th2 , th3 obciazenie oraz blad srednio-kwadratowy sa mniejsze w tabelce gdzie $n = 100$, natomiast dla th4 , dla $n = 100$, wartosci obciazenia i bledu srednio-kwadratowego sa lekko wieksze.

Zadanie 7 dla zadania 5

Table 12: Logis

	wariancja	obciazenie	blad sr kwadrat
(20,1,1)	0.1517523	0.0030369	0.1517615
(20,4,1)	0.1511143	0.0049019	0.1511383
(20,1,2)	0.6383895	0.0072884	0.6384426

Table 13: Logis

	wariancja	obciazenie	blad sr kwadrat
(100,1,1)	0.0299572	0.0004926	0.0299575
(100,4,1)	0.0303972	0.0041780	0.0304147
(100,1,2)	0.1242438	-0.0025732	0.1242504

Pierwsza tabelka opisuje probe dla $n = 20$, druga dla $n = 100$. Widzimy, ze dla $n = 100$ wariancje i bledy srednio-kwadratowe sa wyraznie mniejsze niz dla $n = 20$, czyli lepsze estymatory (bo wariancja wyraznie mniejsze oraz obciazenie estymatorów wyraznie mniejsze dla drugiego i trzeciego i wiersza).

Zadanie 7 dla zadania 6

Table 14: Cauchy

	wariancja	obciazenie	blad sr kwadrat
(20,1,1)	0.1083094	0.0330189	0.1093996
(20,4,1)	0.0834240	-0.0049171	0.0834481
(20,1,2)	0.3206295	0.0570932	0.3238891

Table 15: Cauchy

	wariancja	obciazenie	blad sr kwadrat
(100,1,1)	0.0203286	0.0003065	0.0203287
(100,4,1)	0.0205332	-0.0016814	0.0205361
(100,1,2)	0.0816046	-0.0033695	0.0816160

Porownujac te dwie tabelki widzimy, ze zarowno wariancja, obciazenie jak i blad srednio-kwadratowy sa wyraznie mniejsze dla $n = 100$, zatem dla tego parametru estymator bedzie lepszy.