

# 1 Cerința 1

1) Fiind dată o funcție  $f$ , introdusă de utilizator, determinarea unei constante de normalizare  $k$ . În cazul în care o asemenea constantă nu există, afișarea unui mesaj corespunzător către utilizator

Scurtă descriere a antetului funcției:

Parametrul	Tipul	Descriere
Func	function	Funcția data ca parametru
sup	listă de liste	Suportul funcției

```

1 Nor_constant <- function(Func, sup) {
2
3   sum <- 0
4
5   for (i in sup) {
6     tryCatch(
7       sum <- sum + integrate(Vectorize(Func), i[1], i[2], abs.tol = 0)$
        value,
8       error= function(err) {
9         stop("Integrala_e_divergenta_sau_functia_nu_e_integrabila") #
          daca integrala nu poate fi calculata returnez un mesaj de eroare
10      }
11    )
12
13    if (i[1] == -Inf && i[2] == Inf) {
14      i[1] <- -1000
15      i[2] <- 1000
16    }
17    else if (i[1] == -Inf)
18      i[1] <- i[2] - 1000
19    else if (i[2] == Inf)
20      i[2] <- i[1] + 1000
21
22    if(any(sapply(seq(i[1], i[2], length.out = 1000), Func) < 0))
23      stop("Functie_negativa") # daca functia are valori negative nu pot
          calcula constanta de normalizare
24
25  }
26
27  if(sum == 0)
28    stop("Nu_exista_constanta_de_normalizare_pentru_functia_data") # daca
          integrala = 0 inseamna ca nu exista constanta de normalizare
29
30  const <- 1 / sum
31  return (const)
32
33 }
```

Va fi parcurs suportul funcției dată ca parametru, și vom calcula suma pe fiecare interval din suport. În același timp, verificăm dacă funcția e pozitivă pe fiecare interval din suport, folosind `any(sapply(seq(i[1], i[2], length.out = 1000), Func) < 0)`, pentru a alege 1000 de valori echidistante.

La final, dacă integrala este 0 înseamnă că nu există constantă de normalizare și afișează un mesaj corespunzător. În caz contrar, calculează constanta și o returnează.