

# Nocon\_Tomasz

Praca projektowa nr 1

## ZAD1

## CALKA

### Opis użycia funkcji

Funkcja ta wyznacza wartość całki funkcji  $f$  na przedziale zamkniętym  $[a, b]$  za pomocą metody trapezów.

### Użycie

`calka(f, x, a, b, n)`

### Argumenty

- `f` funkcja, której chcemy policzyć całkę.
- `a` liczba rzeczywista.
- `b` liczba rzeczywista, która spełnia relację  $a < b$ .
- `n` dodatnia liczba całkowita, która określi podział odcinka  $[a, b]$  na  $n$  równych części.

### Szczegóły

Jeśli wartość  $n$  nie jest spercyzowana, to funkcja *calka* domyślnie ustawia  $n = 100$ .

### Wartości

Funkcja zwraca wartość rzeczywistą, która jest przybliżeniem.

### Przykłady użycia

```
calka(dnorm, -3, 3, 1000)
## [1] 0.9973262
```

```
calka(function(x) -x^2 + 2, -1, 1)
## [1] 3.352408
```

```
calka(function(x) tan(x) , 20, 5)
## Error in calka(function(x) tan(x), 20, 5) : a < b is not TRUE
```

```
calka(dnorm, -3, 3, -7)
## Error in calka(dnorm, -3, 3, -7) : n > 0 is not TRUE
```

```
calka(function(x) x^3, 4.7, 10.5, 5.5)
## Error in calka(function(x) x^3, 4.7, 10.5, 5.5) :
  n - as.integer(n) == 0 is not TRUE
```

## ZAD2

### SKLEJ

#### Opis użycia funkcji

Funkcja ta łączy ze sobą napisy na miejscu  $i$ -tym z podanych wektorów oddzielane są przy tym podanym znakiem przez użytkownika. Powstaje w ten sposób nowy wektor napisów.

#### Użycie

`sklej(x, sep)`

#### Argumenty

`x`  $k$ -elementowa lista niepustych wektorów, których długości są takie same. Oczywiście  $k$  jest dodatnią liczbą całkowitą.

`sep` separator, który określa czym będą oddzielone elementy w nowym wektorze.

#### Szczegóły

Jeśli wartość `sep` nie została sprecyzowana to domyślnie funkcja przyjmuje `sep = ""`, pusty znak.

#### Wartości

Zwróconą wartością jest wektor długości równej długości listy, w którym  $i$ -ty element nowopowstałego wektora jest złożony z  $i$ -tych elementów wektorów w liście.

#### Przykłady użycia

```
w <- list(c("Ala", "Kasia", "Zosia"), c("ma", "lubi", "zawsze"),
c("kota", "truskawki", "wygrywa"), c(".", ";", "!"))
sklej(w, sep = " ")
### "Ala ma kota ." "Kasia lubi truskawki ;" "Zosia zawsze wygrywa !"

sklej(list(c("a", "b", "c"),
as.character(1:3),
rep("%", 3)), sep = "~")
## "a~1~%" "b~2~%" "c~3~%"

sklej(c("a", "b", "c"), sep = "*")
## Error in sklej(c("a", "b", "c"), sep = "*") : is.list(x) is not TRUE

x <- list(c("jest", "różna"), c("długość"), c("wektorów"))
sklej(x)
##Error in sklej(x) :
min(unlist(lapply(x, length))) == max(unlist(lapply(x, length))) is not TRUE
```

## ZAD3

### REPR\_MACIERZ

#### Opis użycia funkcji

Funkcja ta przekształca macierz liczbową do specjalnej postaci macierzy dostosowanej do oszczędnego reprezentowania w pamięci komputera macierzy rzadkich, czyli takich dla których większość komórek zawiera wartości zerowe.

#### Użycie

`repr_macierz(x, eps)`

#### Argumenty

`x` macierz liczbowa

`eps` wartość, dla której elementy macierzy mniejsze od niej są tożsame z zerem.

#### Szczegóły1

Jesli epsilon nie został podany, to jego domyślna wartość wynosi  $\text{eps} = 1e - 16$ . Wynikowa macierz składa się z trzech kolumn i liczby wierszy równej liczbie niezerowych elementów macierzy wejściowej. Kolejne wiersze macierzy wynikowej zawierają trójki liczb

$$(i, j, a_{ij}),$$

które kodują pozycje i wartości wszystkich niezerowych komórek  $(a_{ij})$  macierzy wejściowej. Kolumny macierzy rzadkiej oznaczone są kolejno przez (**row**, **col**, **value**)

#### Wartości

Zwracaną wartością jest macierz bez zer. Szczegółowy opis wyniku macierzy jest opisany wyżej, patrz Szczegóły1.

#### Przykłady użycia

```
set.seed(1)
values <- ifelse(sample(c(T,F), 35, replace = TRUE, prob = c(0.1,0.9)), rnorm(20), 0)
x <- matrix(values, ncol = 7)
repr_macierz(x)
##      row col      val
[1,]    4   1 0.37739565
[2,]    2   2 -0.05710677
[3,]    3   4 1.15191175
[4,]    1   5 0.43568330
```

```
y <- matrix(0, ncol = 5, nrow = 7)
y[c(1,2,5,6),1] <- rnorm(4)
repr_macierz(y)
##      row col      val
[1,]    1   1 1.2383041
[2,]    2   1 -0.2793463
[3,]    5   1 1.7579031
[4,]    6   1 0.5607461
```

```

y <- matrix(0, ncol = 5, nrow = 7)
y[1,c(1,3,4)] <- rnorm(3)
y[6,2] <- runif(1)
repr_macierz(y)
##      row col      val
[1,]   1   1 0.3411197
[2,]   6   2 0.9761707
[3,]   1   3 -1.1293631
[4,]   1   4 1.4330237

repr_macierz(list(c(1,2,3), c(2,3,5), c(1,4,7)))
##Error in repr_macierz(list(c(1, 2, 3), c(2, 3, 5), c(1, 4, 7))) :
  is.matrix(x) is not TRUE

```

## ZAD4

### LOGIDERLE

#### Opis użycia funkcji

Funkcja ta generuje z dwóch równolicznych zadanych wektorów jeden wektor logiczny o zadanej długości ##  
Użycie `logiderle(i,j,n)`

#### Argumenty

- `i`            wektor całkowitoliczbowy.
- `j`            wektor całkowitoliczbowy, którego długość jest równa wektowi `i`.
- `n`            całkowita wartość dodatnia.

#### Szczegóły

Wektory `i` oraz `j` spełniają zależność, że dla każdego możliwego `l` zachodzi

$$1 \leq i_l \leq j_l \leq n \text{ oraz } i_l > j_{l-1}$$

Drugi z warunków daje też to, że każdy z wektorów jest posortowany rosnąco, w przeciwnym wypadku otrzymalibyśmy sprzeczność.

#### Wartości

Zwróconą wartością jest  $n$ -elementowy wektor logiczny `w` taki, że  $w_l == TRUE$  wtedy i tylko wtedy gdy  $\exists_p l \in [i_p; j_p]$ .

#### Przykłady użycia

```
logiderle(i =c(1,4), j = c(1,6), n = 7)
## [1] TRUE FALSE FALSE TRUE TRUE TRUE FALSE
```

```
logiderle(i =c(1,4,7), j = c(1,6,8), n = 10)
## [1] TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE FALSE FALSE
```

```
logiderle(i =c(2,5,7,12,14), j = c(4,6,10,13,15), n = 17)
##[1] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE
[13] TRUE TRUE TRUE FALSE FALSE
```