

Programowanie w języku R. Funkcje w R

Aleksander Denisiuk
Uniwersytet Warmińsko-Mazurski
Olsztyn, ul. Słoneczna 54
denisjuk@matman.uwm.edu.pl

18 marca 2024

Funkcje w R

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

Najnowsza wersja tego dokumentu dostępna jest pod adresem

<http://wmii.uwm.edu.pl/~denisjuk/uwm>

Składowe funkcji

❖ Trzy składowe

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

Składowe funkcji

Trzy składowe

Składowe funkcji

❖ Trzy składowe

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- `body()` — kod funkcji
- `formals()` — lista argumentów
- `environment()` — „kontekst” funkcji

♦ jeżeli nie podany, to globalny

```
f <- function(x) x^2
```

```
f
```

```
formals(f)
```

```
body(f)
```

```
environment(f)
```

- Można użyć `body()`, `formals()`, `environment()` do modyfikacji funkcji

```
body(f) <- expression({x^3})
```

Funkcje prymitywne

Składowe funkcji

❖ Trzy składowe

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- Zaimplementowane w C, nie mają kodu w R
 - ◆ wykorzystują `.Primitive()`

```
sum
formals (sum)
body (sum)
environment (sum)
```

Składowe funkcji

Zasięg

❖ Lokalizacja
zmiennych

❖ Domknięcia

❖ Dynamiczna
lokalizacja



Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

Zasięg

Lokalizacja zmiennych

Składowe funkcji

Zasięg

❖ Lokalizacja zmiennych

❖ Domknięcia

❖ Dynamiczna lokalizacja



Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- Jednostką syntaktyczną jest funkcja

```
f <- function() {  
  x <- 1  
  y <- 2  
  c(x, y)  
}  
f()  
rm(f)
```

Decyduje włożoność

Składowe funkcji

Zasięg

❖ Lokalizacja
zmiennych

❖ Domknięcia

❖ Dynamiczna
lokalizacja



Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

```
● x <- 2
  g <- function() {
    y <- 1
    c(x, y)
  }
  g()

● h <- function() {
  y <- 2
  i <- function() {
    z <- 3
    c(x, y, z)
  }
  i()
}
h()
rm(x, h, g)
```


Inicjalizacja

Składowe funkcji

Zasięg

❖ Lokalizacja
zmiennych

❖ Domknięcia

❖ Dynamiczna
lokalizacja



Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

```
j <- function() {  
  if (!exists("a")) {  
    a <- 1  
  } else {  
    a <- a + 1  
  }  
  a  
}  
j()  
j()  
rm(j)
```

Domknięcia

Składowe funkcji

Zasięg

❖ Lokalizacja
zmiennych

❖ Domknięcia

❖ Dynamiczna
lokalizacja



Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

```
j <- function(x) {  
  y <- 2  
  function() {  
    c(x, y)  
  }  
}  
k <- j(1)  
k()  
rm(j, k)
```

Dynamiczna lokalizacja

Składowe funkcji

Zasięg

❖ Lokalizacja
zmiennych

❖ Domknięcia

❖ **Dynamiczna
lokalizacja**



Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

```
f <- function() x
```

```
x <- 15
```

```
f()
```

```
x <- 20
```

```
f()
```



Składowe funkcji

Zasięg

- ❖ Lokalizacja zmiennych
- ❖ Domknięcia
- ❖ Dynamiczna lokalizacja



Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- R szuka funkcji spośród funkcji, nie zmiennych

```
n <- function(x) x / 2
o <- function() {
  n <- 10
  n(n)
}
o()
```

- Można zmienić definicje standardowych funkcji
 - ◆ nigdy tak nie robić

Składowe funkcji

Zasięg

Operatory

❖ Operatory
a funkcje

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

Operatory

Operatory a funkcje

[Składowe funkcji](#)

[Zasięg](#)

[Operatory](#)

❖ [Operatory
a funkcje](#)

[Argumenty funkcji](#)

[Funkcje specjalne](#)

[Zwracane wartości](#)

[Superpozycja](#)

[Pakiety](#)

To understand computations in R, two slogans are helpful:

- Everything that exists is an object.
- Everything that happens is a function call.

— *John Chambers*

Przykłady

Składowe funkcji

Zasięg

Operatory

❖ Operatory
a funkcje

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

```
x <- 10; y <- 5
```

```
x + y
```

```
`+`(x, y)
```

```
for (i in 1:2) print(i)
```

```
`for`(i, 1:2, print(i))
```

```
if (i == 1) print("yes!") else print("no.")
```

```
`if`(i == 1, print("yes!"), print("no."))
```

```
x[3]
```

```
`[`(x, 3)
```

```
{ print(1); print(2); print(3) }
```

```
`{`(print(1), print(2), print(3))
```

Operatory zamiast funkcji

Składowe funkcji

Zasięg

Operatory

❖ Operatory
a funkcje

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

```
add <- function(x, y) x + y
sapply(1:10, add, 3)
sapply(1:5, `+`, 3)
sapply(1:5, "+", 3)
```

- `sapply()` może mieć jako argument funkcję albo jej nazwę
 - ◆ zobaczyć w kodzie

```
x <- list(1:3, 4:9, 10:12)
sapply(x, function(x) x[2])
sapply(x, "[", 2)
```




Redefinicja operatorów

Składowe funkcji

Zasięg

Operatory

❖ Operatory
a funkcje

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- Można zmienić definicję operatorów
 - ✦ nigdy tego nie robić

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

- ❖ Przekazywanie argumentów
- ❖ Lista argumentów
- ❖ Wartości domyślne
- ❖ Wartości pominięte
- ❖ Wartościowanie
- ❖ Argument . . .
- ❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

Argumenty funkcji

Przekazywanie argumentów

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument ...

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- Według pełnej nazwy
- Według skrótu nazwy
- Według pozycji

```
f <- function(abcdef, bcde1, bcde2) {  
  list(a = abcdef, b1 = bcde1, b2 = bcde2)  
}
```

```
str(f(1, 2, 3))
```

```
str(f(2, 3, abcdef = 1))
```

```
str(f(2, 3, a = 1))
```

```
str(f(1, 3, b = 1))
```

Reguły dobrego stylu

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument . . .

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- Według pozycji tylko dla pierwszego (pierwszych dwóch) argumentów
 - ◆ oczywistych, powszechnie znanych
- Jeżeli skróty, to czytelne
- Jeżeli to moduł dla CRAN, to tylko pełna nazwa
- Argumenty po . . . tylko pełna nazwa

Lista argumentów

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument . . .

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

```
args <- list(1:10, na.rm = TRUE)
do.call(mean, args)
```

```
mean(1:10, na.rm = TRUE)
```

Wartości domyślne

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument ...

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

● Wartości domyślne

```
f <- function(a = 1, b = 2) {  
  c(a, b)  
}  
f()
```

● Zależne od poprzednich

```
g <- function(a = 1, b = a * 2) {  
  c(a, b)  
}  
g()  
g(10)
```

Wartości pominięte

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument ...

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

● Funkcja `missing()`

```
i <- function(a, b) {  
  c(missing(a), missing(b))  
}  
i()  
i(b=2)  
i(2, 4)
```

● Inne podejście

```
g <- function(a = NULL, b = NULL) {  
  c(is.null(a), is.null(b))  
}  
g()  
g(b=2)  
g(1, 2)
```

Wartościowanie leniwe

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument ...

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- Argumenty są obliczane w momencie, gdy są potrzebne

```
f <- function(x) {  
  10  
}  
f(stop("This is an error!"))
```

- Funkcja `force()`

```
f <- function(x) {  
  force(x)  
  10  
}  
f(stop("This is an error!"))
```


Argumenty domyśle

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument . . .

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- są obliczane wewnątrz funkcji
 - ✦ wynik może być inny, niż w globalnym kontekście

```
f <- function(x = ls()) {  
  a <- 1  
  x  
}
```

```
f()
```

```
f(ls())
```

Wartościowanie leniwe w wyrażeniach

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument . . .

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

```
if (!is.null(x) && x > 0) {
```

```
!is.null(a) || stop("a is null")
```

Argument . . .

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument . . .

❖ Pytania

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- . . . pasuje do argumentów niedopasowanych w inny sposób
 - ◆ może zostać przekazany do innych funkcji
 - ◆ na przykład, `plot(x, y, . . .)`
 - ◆ literówki nie powodują błędu:
`sum(1, 2, NA, na.mr = TRUE)`
- Odwołać się do . . . w programie można za pomocą listy:

```
f <- function(. . .) {  
  names(list(. . .))  
}  
f(a = 1, b = 2)
```

Pytania

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

❖ Przekazywanie argumentów

❖ Lista argumentów

❖ Wartości domyślne

❖ Wartości pominięte

❖ Wartościowanie

❖ Argument ...

❖ **Pytania**

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

- Jaki będzie wynik obliczenia:

```
f1 <- function(x = {y <- 1; 2}, y = 0) {  
  x + y  
}  
f1()
```

- Jaki będzie wynik obliczenia:

```
f2 <- function(x = z) {  
  z <- 100  
  x  
}  
f2()
```

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

❖ Notacja infiksowa

❖ Funcje
zastępujące

Zwracane wartości

Superpozycja

Pakiety

Fukcje specjalne

Notacja infiksowa

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

❖ Notacja infiksowa

❖ Funcje zastępujące

Zwracane wartości

Superpozycja

Pakiety

- Operatory standardowe: `% %`, `% * %`, `% / %`, `% in %`, `% o %`, `% x %`,
`:`, `::`, `:::`, `$`, `@`, `^`, `*`, `/`, `+`, `-`, `>`, `>=`, `<`,
`<=`, `==`, `!=`, `!`, `&`, `&&`, `|`, `||`, `~`, `<-`, `<<-`
- Operatory użytkownika powinny się rozpoczynać od `%`
``%+%` <- function(a, b) paste0(a, b)`
`"new" %+% " string"`
- W nazwach operatorów można używać znaków specjalnych
``% %` <- function(a, b) paste(a, b)`
``%'%` <- function(a, b) paste(a, b)`
``%/\\%` <- function(a, b) paste(a, b)`
`"a" % % "b"`
`"a" %' % "b"`
`"a" %/\\ % "b"`
- Operatory obliczane są od lewej strony

Funcje zastępujące

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

❖ Notacja infiksowa

❖ Funcje
zastępujące

Zwracane wartości

Superpozycja

Pakiety

- Modyfikują argumenty

- ✦ mają postać `xxx<-`

```
`second<-` <- function(x, value) {  
  x[2] <- value  
  x  
}  
x <- 1:10  
second(x) <- 5L  
`second<-` (x, 5L)
```

- Jest tworzona kopia danych

- ✦ funkcje prymitywne nie kopiują danych

- `second(x) <- 5L` **vs** `x[2] <- 5L`

- ✦ `tracemem(x)`

Dodatkowe argumenty

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

❖ Notacja infiksowa

❖ Funcje
zastępujące

Zwracane wartości

Superpozycja

Pakiety

- Między `x` `value`

```
`modify<-` <- function(x, position, value) {  
  x[position] <- value  
  x  
}
```

```
modify(x, 1) <- 10
```

```
`modify<-`(x, 1, 10)
```


Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

- ❖ Wynik funkcji
- ❖ `return`
- ❖ Efekty uboczne
- ❖ Funkcje niewidoczne
- ❖ `on.exit()`

Superpozycja

Pakiety

Zwracane wartości

Wynik funkcji

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

❖ Wynik funkcji

❖ `return`

❖ Efekty uboczne

❖ Funkcje niewidoczne

❖ `on.exit()`

Superpozycja

Pakiety

- Ostatnie obliczane wyrażenie

```
f <- function(x) {  
  if (x < 10) {  
    0  
  } else {  
    10  
  }  
}  
f(5)  
f(15)
```

- Tylko jeden obiekt

- ◆ może być lista

return

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

❖ Wynik funkcji

❖ **return**

❖ Efekty uboczne

❖ Funkcje niewidoczne

❖ `on.exit()`

Superpozycja

Pakiety

- Funkcja `return()` zazwyczaj w przypadku błędów, bądź wcześniejszego końca obliczeń

```
f <- function(x, y) {  
  if (!x) return(y)  
  .....  
}
```

Efekty uboczne

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

❖ Wynik funkcji

❖ `return`

❖ **Efekty uboczne**

❖ Funkcje niewidoczne

❖ `on.exit()`

Superpozycja

Pakiety

- Większość funkcji nie ma efektów ubocznych
 - ✦ zawsze taki sam wynik dla tych samych danych
- Modyfikacja argumentów funkcji nie jest możliwa

```
f <- function(x) {  
  x$a <- 2  
  x  
}  
  
x <- list(a = 1)  
f(x)  
  
x$a
```

- ✦ wyjątek: kontekst funkcji oraz klasy referencyjne

Funkcje mające efekty uboczne

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

❖ Wynik funkcji

❖ `return`

❖ **Efekty uboczne**

❖ Funkcje niewidoczne

❖ `on.exit()`

Superpozycja

Pakiety

- `library()`
- `setwd()`, `Sys.setenv()`, `Sys.setlocale()`
- `plot()`
- `write()`, `write.csv()`, `saveRDS()`
- `options()` oraz `par()`
- Niektóre funkcje klas S4
- Generator liczb pseudolosowych

Funkcje niewidoczne

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

❖ Wynik funkcji

❖ `return`

❖ Efekty uboczne

❖ Funkcje
niewidoczne

❖ `on.exit()`

Superpozycja

Pakiety

- Funkcja, która zwraca `invisible()` nie wyświetla wyniku

```
f1 <- function() 1
f2 <- function() invisible(1)

f1()
f2()

f1() == 1
f2() == 1

(f2())
a <- 2
(a <- 2)

a <- b <- c <- d <- 2
(a <- (b <- (c <- (d <- 2))))
```

on.exit()

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

❖ Wynik funkcji

❖ `return`

❖ Efekty uboczne

❖ Funkcje niewidoczne

❖ `on.exit()`

Superpozycja

Pakiety

● Działania po zakończeniu funkcji

```
in_dir <- function(dir, code) {  
  old <- setwd(dir)  
  on.exit(setwd(old))  
  
  force(code)  
}  
getwd()
```

- ◆ w razie wielu `on.exit()` dodać parameter `add = TRUE`

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

❖ Superpozycja funkcji

❖ Superpozycja

❖ Kolejne wywołania

❖ Pipe

Pakiety

Superpozycja

Superpozycja funkcji

[Składowe funkcji](#)

[Zasięg](#)

[Operatory](#)

[Argumenty funkcji](#)

[Funkcje specjalne](#)

[Zwracane wartości](#)

[Superpozycja](#)

❖ Superpozycja funkcji

❖ Superpozycja

❖ Kolejne wywołania

❖ Pipe

[Pakiety](#)

- Odchylenie standardowe

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^m (x_i - \bar{x})^2}$$

- Implementacja z wykorzystaniem `mean()`, `sqrt()` oraz dodatkowych dwóch funkcji:

```
square <- function(x) x^2  
deviation <- function(x) x - mean(x)
```

Superpozycja

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

❖ Superpozycja
funkcji

❖ Superpozycja

❖ Kolejne wywołania

❖ Pipe

Pakiety

```
x <- runif(100)
sqrt(mean(square(deviation(x))))
```

- Krótkie wyrażenia, czyta się z prawej, kanapka Dagwooda



Kolejne wywołania

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

❖ Superpozycja
funkcji

❖ Superpozycja

❖ Kolejne wywołania

❖ Pipe

Pakiety

```
out <- deviation(x)
out <- square(out)
out <- mean(out)
out <- sqrt(out)
out
```

- Trzeba mianować pośrednie wyniki

Potok (pipe)

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

❖ Superpozycja
funkcji

❖ Superpozycja

❖ Kolejne wywołania

❖ Pipe

Pakiety

```
library(magrittr)
```

```
x %>%
```

```
deviation() %>%
```

```
square() %>%
```

```
mean() %>%
```

```
sqrt()
```

- $x \%>\% f(y) \iff f(x, y)$
- Zewnętrzny moduł, tylko liniowe potoki

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ `devtools`

❖ **Prawdziwy pakiet**

❖ `DESCRIPTION`

❖ `NAMESPACE`

❖ Dokumentacja

❖ Licencja

❖ weryfikacja

Pakiety

Własne pakiety

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ **Własne pakiety**

❖ **Minimalny pakiet**

❖ Kompilacja

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ Dokumentacja

❖ Licencja

❖ weryfikacja

- Ponowne wykorzystanie swoich funkcji
- Rozpowszechnienie danych oraz oprogramowania
 - ❖ CRAN
 - ❖ kod używanych w eksperymentach dotyczących artykułu (pracy magisterskiej)
- Pakiet nie musi być duży
- Nie ma obowiązku rozpowszechniania
- Roxygen2 znacznie ułatwia napisanie dokumentacji

Minimalny pakiet

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ Dokumentacja

❖ Licencja

❖ weryfikacja

● Struktura katalogów:

```
adscore
├── R
│   ├── confusionmatrix.r
│   └── f1score.r
└── DESCRIPTION
```

● Plik DESCRIPTION:

Package: `adscore`

Version: `0.1`

Kompilacja

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ **Kompilacja**

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ Dokumentacja

❖ Licencja

❖ weryfikacja

- W katalogu nadrzędnym do `adscore`:
`R CMD build adscore`
- Tworzony jest domyślny plik `NAMESPACE`
- Pakiet zapisywany jest do pliku `adscore_0.1.tar.gz`

Wykorzystanie devtools

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ **devtools**

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ Dokumentacja

❖ Licencja

❖ weryfikacja

- Można skorzystać z pakietu devtools
- Ustawić bieżący katalog na adscore

```
install.packages('devtools')  
library(devtools)  
build()  
install()
```

- Trzeba jednak mieć plik NAMESPACE

Prawdziwy pakiet

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ `devtools`

❖ **Prawdziwy pakiet**

❖ `DESCRIPTION`

❖ `NAMESPACE`

❖ Dokumentacja

❖ Licencja

❖ weryfikacja

Dodatkowe czynności

- Uzupełnić plik `DESCRIPTION`
- Utworzyć plik `NAMESPACE`
- Dodać dokumentację
- Dodać licencję

Uzupełnienie *DESCRIPTION*

<u>Składowe funkcji</u>
<u>Zasięg</u>
<u>Operatory</u>
<u>Argumenty funkcji</u>
<u>Fukcje specjalne</u>
<u>Zwracane wartości</u>
<u>Superpozycja</u>
<u>Pakiety</u>
❖ Własne pakiety
❖ Minimalny pakiet
❖ Kompilacja
❖ devtools
❖ Prawdziwy pakiet
❖ DESCRIPTION
❖ NAMESPACE
❖ Dokumentacja
❖ Licencja
❖ weryfikacja

- Dodać co najmniej taką informację

Package: `adscore`

Version: `0.1`

Date: `2024-03-11`

Title: `A. D. Score`

Description: `Functions that Aleksander Denisiuk
uses for computing F1score.`

Author: `Aleksander Denisiuk <denisiuk@matman.uwm.edu.pl>`

Maintainer: `Aleksander Denisiuk <denisiuk@matman.uwm.edu.pl>`

Encoding: `UTF-8`

- Tytuł ma być krótki
- Opisanie może być na kilka linii
- Dla autora email jest opcjonalny
- Kodowanie — dla Roxygen2

Utworzenie *NAMESPACE*

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ **NAMESPACE**

❖ Dokumentacja

❖ Licencja

❖ weryfikacja

- Minimalny plik

```
# Export all names
exportPattern(".")
```

- Pierwszy wiersz to komentarz
- Już można instalować z devtools

Dokumentacja

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ `devtools`

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ **Dokumentacja**

❖ Licencja

❖ weryfikacja

- W katalogu `man`
- Format `Rd`
- Roxygen2

Format Rd

[Składowe funkcji](#)

[Zasięg](#)

[Operatory](#)

[Argumenty funkcji](#)

[Funkcje specjalne](#)

[Zwracane wartości](#)

[Superpozycja](#)

[Pakiety](#)

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ **Dokumentacja**

❖ Licencja

❖ weryfikacja

```
\name{f1core}
\alias{f1core}
\title{Calculation of the F1 Score}
\usage{
  f1score( precision, recall )
}
\arguments{
\item{precision}{the value of precision}
\item{recall}{the value of recall}
\value{the correspondent f1score value}
\description{
  Calculates the value of F1 Score
}
\examples{
  f1score( precision, recall )
}
```

Format Roxygen2

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ **Dokumentacja**

❖ Licencja

❖ weryfikacja

```
# ' Calculation of the F1 Score
# '
# ' Calculates the value of F1 Score
# '
# ' @param precision the value of precision
# ' @param recall the value of recall
# '
# ' @return the correspondent f1score value
# '
# ' @examples
# ' f1score( precision, recall )
# '
# ' @export

f1score <- function( precision, recall){
  2* (precision*recall) / (precision+recall)
}
```

kompilacja dokumentacji

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ **Dokumentacja**

❖ Licencja

❖ weryfikacja

`document ()`

● Utworzenie pakietu:

```
setwd('adscore')  
library(devtools)  
document()  
build()  
install()
```


Dodatkowe znaczniki Roxygen2

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ **Dokumentacja**

❖ Licencja

❖ weryfikacja

```
#' @author Aleksander Denisiuk,  
#'      \email{denisiuk@matman.uwm.edu.pl}  
# ' @references  
# '      \url{https://en.wikipedia.org/wiki/F-score}  
# ' @seealso \code{\link[MLmetrics]{F1_Score}}  
# ' @keywords flscore  
# ' ...
```

● etc

Licencja

- Składowe funkcji
- Zasięg
- Operatory
- Argumenty funkcji
- Fukcje specjalne
- Zwracane wartości
- Superpozycja
- Pakiety
 - ❖ Własne pakiety
 - ❖ **Minimalny pakiet**
 - ❖ Kompilacja
 - ❖ devtools
 - ❖ **Prawdziwy pakiet**
 - ❖ DESCRIPTION
 - ❖ NAMESPACE
 - ❖ Dokumentacja
 - ❖ **Licencja**
 - ❖ weryfikacja

- W pliku DESCRIPTION

- ◆ GPL-3

- License: GPL-3

- ◆ MIT

- License: MIT + file LICENSE

- W pliku LICENSE

- YEAR: 2014

- COPYRIGHT HOLDER: Aleksander Denisiuk

- ◆ Może być LICENSE

- ◆ Albo nawet inna nazwa

Weryfikacja pakietu

Składowe funkcji

Zasięg

Operatory

Argumenty funkcji

Fukcje specjalne

Zwracane wartości

Superpozycja

Pakiety

❖ Własne pakiety

❖ **Minimalny pakiet**

❖ Kompilacja

❖ devtools

❖ **Prawdziwy pakiet**

❖ DESCRIPTION

❖ NAMESPACE

❖ Dokumentacja

❖ Licencja

❖ **weryfikacja**

- W wierszu poleceń:

```
R CMD check adscore_0.1.tar.gz
```

- ◆ Dodatkowa weryfikacja dla pakietów, które są ładowane na CRAN

```
R CMD check --as-cran adscore_0.1.tar.gz
```

- Z devtools:

```
check()
```