

Sprawozdanie

Zajęcia: Analiza procesów uczenia

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 1

Data: 01.03.2024

Temat: Podstawy języka R

Wariant: 1

Agnieszka Białecka
Informatyka II stopień,
stacjonarne,
1 semestr,
Gr.1a

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z językiem R poprzez rozwiązywanie zadań polegających na tworzeniu i wyświetlaniu ramki danych.

2. Wstęp teoretyczny

R - język i środowisko do obliczeń statystycznych i grafiki. Jest to projekt GNU podobny do języka i środowiska S, który został opracowany w Bell Laboratories (dawniej AT&T, obecnie Lucent Technologies) przez Johna Chambersa i współpracowników. R można uznać za inną implementację S. Istnieje kilka istotnych różnic, ale większość kodu napisanego dla S działa w niezmienionym stanie w R.

R zapewnia szeroką gamę technik statystycznych (modelowanie liniowe i nieliniowe, klasyczne testy statystyczne, analiza szeregów czasowych, klasyfikacja, grupowanie) i technik graficznych.

3. Przebieg ćwiczenia

- a) Do zmiennej *a* podstaw wartość wyrażenia $2 \cdot \exp(5)$. Do zmiennej *b* podstaw podwojoną wartość zmiennej *a*. Wywołaj funkcję sprawdzającą, która z wartości zmiennych jest większa.

```
> a <- 2 * exp(5)
> b <- 2 * a
>
> # funkcja sprawdzająca, która zmienna jest większa
> check <- function(a,b) {
+   if (a > b) {
+     print("wartość zmiennej a jest większa.")
+   } else if (b > a) {
+     print("wartość zmiennej b jest większa.")
+   } else {
+     print("obie zmienne są równe.")
+   }
+ }
>
> check(a,b)
[1] "wartość zmiennej b jest większa."
```

- b) Uruchom i poczytaj dokumentację dla funkcji `sum()`.

```
> ?sum
```

Sum of Vector Elements

Description

`sum` returns the sum of all the values present in its arguments.

Usage

```
sum(..., na.rm = FALSE)
```

Arguments

`...` numeric or complex or logical vectors.

`na.rm` logical. Should missing values (including `NaN`) be removed?

Details

This is a generic function: methods can be defined for it directly or via the [Summary](#) group generic. For this to work properly, the arguments `...` should be unnamed, and dispatch is on the first argument.

If `na.rm` is `FALSE` an `NA` or `NaN` value in any of the arguments will cause a value of `NA` or `NaN` to be returned, otherwise `NA` and `NaN` values are ignored.

- c) Stwórz wektor `a` zawierający liczby od 15 do 25. Oblicz sumę liczb zawartych w wektorze.

```
> a <- 15:25
> suma_a <- sum(a)
> print(paste("Suma liczb w wektorze a:", suma_a))
[1] "Suma liczb w wektorze a: 220"
```

values	
a	int [1:11] 15 16 17 18 19 20 21 22 23 24 ...
b	593.652636410306
suma_a	220L

- d) Wyświetl wszystkie funkcje zawierające frazę “sum” w swojej nazwie.

```

> apropos("sum")
[1] ".colSums"                ".rowSums"
[3] ".rs.callSummary"         ".rs.summarizeDir"
[5] ".rs.tutorial.onResume"   ".tryResumeInterrupt"
[7] "colSums"                 "contr.sum"
[9] "cumsum"                  "format.summaryDefault"
[11] "marginSums"              "print.summary.table"
[13] "print.summary.warnings"  "print.summaryDefault"
[15] "rowsum"                  "rowsum.data.frame"
[17] "rowsum.default"          "rowSums"
[19] "sum"                     "suma_a"
[21] "summary"                 "Summary"
[23] "summary.aov"             "summary.connection"
[25] "summary.data.frame"      "Summary.data.frame"
[27] "summary.Date"            "Summary.Date"
[29] "summary.default"         "Summary.difftime"
[31] "summary.factor"          "Summary.factor"
[33] "summary.glm"             "summary.lm"
[35] "summary.manova"          "summary.matrix"
[37] "Summary.numeric_version" "Summary.ordered"
[39] "summary.POSIXct"         "Summary.POSIXct"
[41] "summary.POSIXlt"         "Summary.POSIXlt"
[43] "summary.proc_time"       "summary.srcfile"
[45] "summary.srcref"          "summary.stepfun"
[47] "summary.table"           "summary.warnings"
[49] "summaryRprof"

```

- e) Ustaw dowolny katalog roboczy. Następnie stwórz zmienną `a` zawierającą łańcuch znaków "smartfony Samsung". Zapisz zmienną `a` z obszaru roboczego do pliku w katalogu roboczym. Następnie usuń zmienną `a`. Sprawdź wartość zmiennej `a` (powinno jej brakować). Na końcu wczytaj plik ze zmienną `a` i sprawdź jej wartość.

```

> setwd("C:/Users/student/Documents/Lab1") # ustawienie katalogu roboczego
> a <- "smartfony Samsung"
> write(a, file = "a.txt") # zapis zmiennej a do pliku
> rm(a) # usunięcie zmiennej a
> print(a) # sprawdzenie, czy zmienna a została usunięta
BŁĄD: nie znaleziono obiektu 'a'
> a <- readLines("a.txt") # wczytanie zmiennej a z pliku
> print(a) # wyświetlenie wczytanej zmiennej a
[1] "smartfony Samsung"

```

- f) Zainstaluj i załaduj pakiet `gridExtra`, który umożliwia m.in ładną wizualizację danych tabelarycznych. Następnie przy pomocy dokumentacji pakietu znajdź funkcję do wizualizacji danych tabelarycznych. Użyj jej na pierwszych 10 wierszach zbioru danych `mtcars`.

```

> install.packages("gridExtra")

```

```
> library(gridExtra)
> ?grid.arrange
```

g) Stwórz wektor zawierający ciąg liczb 100, 96, 92, ... 20.

```
> wektor_g <- seq(100, 20, by = -4)
> print(wektor_g)
[1] 100 96 92 88 84 80 76 72 68 64 60 56 52 48 44 40 36 32 28 24 20
```

h) Stwórz wektora a z liczbami od 9 do 5 oraz wektor b z liczbami od 11 do 16. Utwórz nowy wektor d będący połączeniem wektora b i a (w takiej kolejności). Wyświetl go.

```
> a <- 9:5
> b <- 11:16
> d <- c(b, a)
> print(d)
[1] 11 12 13 14 15 16 9 8 7 6 5
```

values	
a	int [1:5] 9 8 7 6 5
b	int [1:6] 11 12 13 14 15 16
d	int [1:11] 11 12 13 14 15 16 9 8 7 6 ...

i) Stwórz wektor nazwa zawierający nazwy 10 smartfonów Samsung z systemem Android 8 i ośmiordzienneowym procesorem. Potem stwórz wektory wyświetlacz, pamięć_RAM i pamięć_wbudowana, aparat_foto, cena, liczba_opinii zawierające kolejno dane 10 smartfonów. Następnie utwórz ramkę danych smartfony złożoną z wektorów nazwa, wyświetlacz, pamięć_RAM, pamięć_wbudowana, aparat_foto, cena oraz liczba_opinii. Wylicz średnią cenę smartfonów.

```
> nazwa <- c("Samsung 1", "Samsung 2", "Samsung 3", "Samsung 4", "Samsung 5", "Samsung 6",
"Samsung 7", "Samsung 8", "Samsung 9 ", "Samsung 10")
> wyświetlacz <- c(6.2, 6.5, 6.1, 6.4, 6.0, 6.3, 6.5, 6.2, 6.4, 6.6)
> pamiec_RAM <- c(4, 8, 6, 6, 4, 6, 8, 6, 8, 6)
> pamiec_wbudowana <- c(64, 128, 64, 128, 64, 128, 256, 128, 256, 128)
> aparat_foto <- c(48, 64, 48, 64, 48, 64, 108, 64, 108, 48)
> cena <- c(800, 1000, 900, 1100, 700, 1200, 1300, 1000, 1400, 900)
> liczba_opinii <- c(200, 150, 180, 220, 250, 190, 170, 230, 160, 210)
> smartfony <- data.frame(nazwa, wyświetlacz, pamiec_RAM, pamiec_wbudowana, aparat_foto,
cena, liczba_opinii)
> srednia_cena <- mean(smartfony$cena)
> print(paste("Średnia cena smartfonów:", srednia_cena))
[1] "Średnia cena smartfonów: 1030"
```

Values	
aparat_foto	num [1:10] 48 64 48 64 48 64 108 64 108 48
cena	num [1:10] 800 1000 900 1100 700 1200 1300 1000 1400 900
liczba_opinii	num [1:10] 200 150 180 220 250 190 170 230 160 210
nazwa	chr [1:10] "Samsung 1" "Samsung 2" "Samsung 3" "Samsung...
pamiec_RAM	num [1:10] 4 8 6 6 4 6 8 6 8 6
pamiec_wbudowana	num [1:10] 64 128 64 128 64 128 256 128 256 128
srednia_cena	1030
suma_a	220L
wyświetlacz	num [1:10] 6.2 6.5 6.1 6.4 6 6.3 6.5 6.2 6.4 6.6

- j) Do utworzonej w poprzednim zadaniu ramki danych smartfonów dodaj wpis zawierający dane nowego smartfonu. Wylicz średnią ceny ponownie.

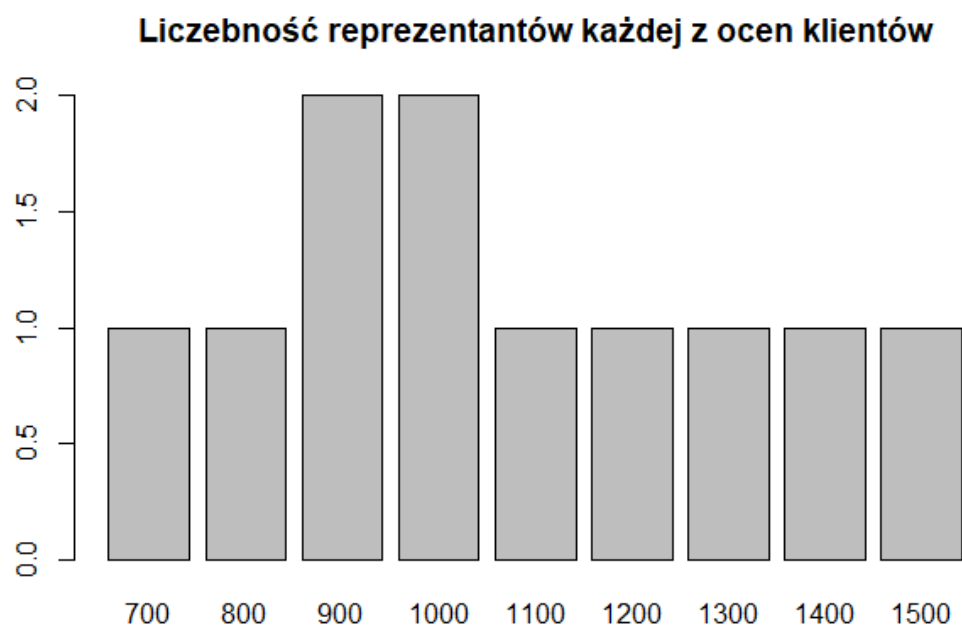
```
> nowy_smartfon <- data.frame("Samsung 11", 6.7, 8, 256, 108, 1500, 18)
> names(nowy_smartfon) <- c("nazwa", "wyświetlacz", "pamiec_RAM", "pamiec_wbudowana",
"aparat_foto", "cena", "liczba_opinii")
> smartfony <- rbind(smartfony, nowy_smartfon)
> nowa_srednia_cena <- mean(smartfony$cena)
> print(paste("Nowa średnia cena smartfonów po dodaniu nowego:", nowa_srednia_cena))
[1] "Nowa średnia cena smartfonów po dodaniu nowego: 1072.727272727"
```

- k) Korzystając z ramki danych smartfony dodaj nową kolumnę określając ocenę klientów. Wpisz do kolumny odpowiednio oceny w skali od 0 do 5 krok 0.5. Dodana kolumna powinna się automatycznie przekonwertować do cech jakościowych (tzw. factors). Wylicz średnią ceny każdej oceny.

```
> smartfony$ocena_klientow <- seq(0, 5, by = 0.5)
> srednie_ceny_ocen <- tapply(smartfony$cena, smartfony$ocena_klientow, mean)
> print(srednie_ceny_ocen)
 0  0.5  1  1.5  2  2.5  3  3.5  4  4.5  5
800 1000 900 1100 700 1200 1300 1000 1400 900 1500
```

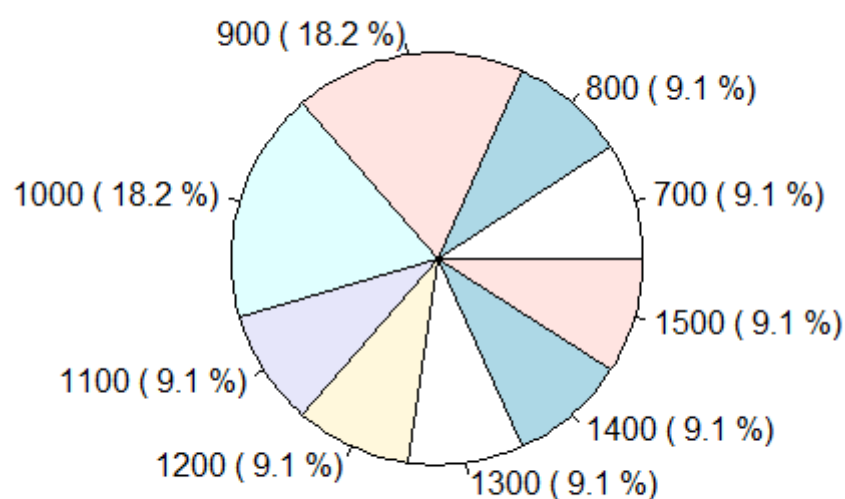
- l) Do ramki danych smartfony dodaj kolejne 4 smartfony. Narysuj na wykresie słupkowym liczebność reprezentantów każdej z ocen klientów.

```
> barplot(table(srednie_ceny_ocen), main = "Liczebność reprezentantów każdej z ocen klientów")
```



m) Wykorzystując ramkę danych smartfony pokaż procentowy udział każdej oceny przy pomocy wykresu kołowego oraz wachlarzowego.

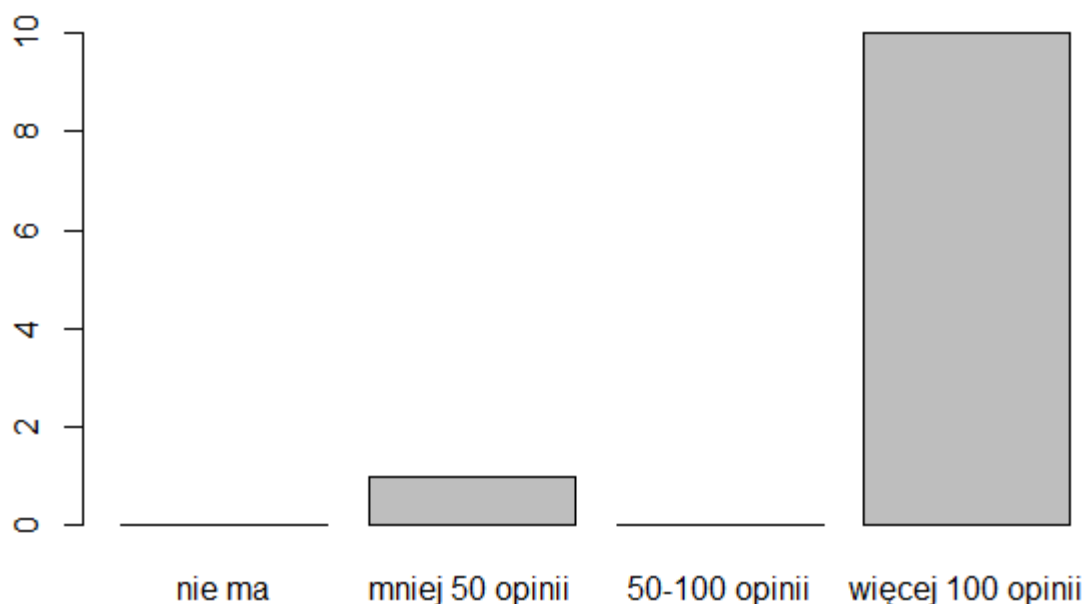
```
> pct <- prop.table(table(srednie_ceny_ocen)) * 100
> pie(pct, labels = paste(names(pct), "(", round(pct, 1), "%)"))
```



- n) Do ramki danych smartfony dodaj nową kolumnę status opinii z wartościami: "nie ma", "mniej 50 opinii", "50-100 opinii", "więcej 100 opinii" w zależności od liczby opinii. Zamień dodaną kolumnę na cechy jakościowe. Następnie przy pomocy wykresu kołowego wyrysuj procentowy udział smartfonów o konkretnym statusie opinii.

```
> smartfony$status_opinii <- cut(smartfony$liczba_opinii, breaks = c(-Inf, 0, 50, 100, Inf),
labels = c("nie ma", "mniej 50 opinii", "50-100 opinii", "więcej 100 opinii"))
> smartfony$status_opinii <- as.factor(smartfony$status_opinii)
> barplot(table(smartfony$status_opinii), main = "Procentowy udział smartfonów w zależności
od statusu opinii")
```

Procentowy udział smartfonów w zależności od statusu opinii



- o) Wykorzystując ramkę danych smartfony stwórz zdanie o każdym z smartfonów postaci: nazwa + " ma ocenę klientów " + ocena_klientów + " bo ma liczbę opinii" + liczba_opinii. Plus oznacza konkatencję łańcuchów i wartości.

```
> zdania <- paste(smartfony$`nazwa`, "ma ocenę klientów", smartfony$`ocena_klientow`,
"bo ma liczbę opinii", smartfony$`liczba_opinii`)
> print(zdania)
[1] "Samsung 1 ma ocenę klientów 0 bo ma liczbę opinii 200"
[2] "Samsung 2 ma ocenę klientów 0.5 bo ma liczbę opinii 150"
[3] "Samsung 3 ma ocenę klientów 1 bo ma liczbę opinii 180"
[4] "Samsung 4 ma ocenę klientów 1.5 bo ma liczbę opinii 220"
[5] "Samsung 5 ma ocenę klientów 2 bo ma liczbę opinii 250"
[6] "Samsung 6 ma ocenę klientów 2.5 bo ma liczbę opinii 190"
[7] "Samsung 7 ma ocenę klientów 3 bo ma liczbę opinii 170"
[8] "Samsung 8 ma ocenę klientów 3.5 bo ma liczbę opinii 230"
[9] "Samsung 9 ma ocenę klientów 4 bo ma liczbę opinii 160"
[10] "Samsung 10 ma ocenę klientów 4.5 bo ma liczbę opinii 210"
[11] "Samsung 11 ma ocenę klientów 5 bo ma liczbę opinii 18"
```


- p) Zachować ramkę danych w pliku .csv. Załadować ramkę danych z pliku .csv Dane (15 smartfonów) pobrać ze strony <http://www.euro.com.pl>

```
> write.csv(smartfony, file = "smartfony.csv", row.names = FALSE)
> smartfony <- read.csv("smartfony.csv")
```

Repozytorium GitHub:

<https://github.com/Delisolara/APU>

4. Podsumowanie

Przeprowadzone ćwiczenia umożliwiło zapoznanie się z językiem R oraz jego szerokimi możliwościami w analizie danych i statystyce. W jego ramach została przedstawiona składnia języka R, różne rodzaje obiektów danych, takie jak wektory, ramki danych i listy.