

# Podstawy pakietu R

## Ćwiczenia

Marek Panfilow

2024/2025

## 1 Środowisko R

1. Uruchom R
2. W konsoli wpisz:
  - a. `x <- 12`
  - b. `x`
  - c. `x * 7`
  - d. `y <- x*7`
  - e. `y`
  - f. `pi`
3. Oblicz **sin 165** stopni
4. Napisz skrypt przypisujący do **a** wartość **12**, do **b** wartość **56** i wyświetlający  $(a+b)^2$
5. Wykonaj skrypt

## 2 R Studio (Posit)

1. Utwórz nowy skrypt
2. W oknie skryptu wpisz: `pff <- 42`
3. Wyślij kod do konsoli
4. Obejrzyj prawe górne okno RStudio
5. W konsoli wpisz `pff`
6. Wpisz w skrypcie `pff/3`
7. Wykonaj skrypt
8. W konsoli wpisz `z <- 32; z`
9. Przenieś ostatni wykonany kod do skryptu korzystając z historii
10. Stwórz nowy projekt **p1**
11. Utwórz skrypt i wpisz w niego `x <- log(20); x`
12. Wykonaj skrypt
13. Stwórz nowy projekt **p2**
14. Utwórz skrypt i wpisz:
  - a. `p <- 34;p`
  - b. `z = p > 3`
15. Wykonaj skrypt
16. Sprawdź wartość **z**
17. Przełącz się między projektami
18. Zainstaluj pakiet **validate**

19. Zaimportuj plik **t1.csv**
20. Zaimportuj plik **t2.xlsx**
21. Wyświetl wartości z pliku **t2** na konsoli

## 3 Vectors

### 3.1 Część A

1. Na trzy różne sposoby utwórz vector o wartościach **od -2 do 4** (**-2, -1, 0, 1, 2, 3, 4**). Przypisz je do obiektów **v1, v2, v3**
2. Sprawdź czy vectory **v1, v2, v3** są identyczne. Użyj: **identical()**
3. Utwórz 13 elementowy vector **d** o wartościach z zakresu od -2.7 do 9.3
4. Policz średnią z wartości **d** (? **mean**)
5. Wyświetl liczbę elementów **d**
6. Utwórz vector **d1** jak sumę **d** i **v1**
7. Wyświetl liczbę elementów **d1**
8. Utwórz vector **d2** składający się z elementów **d1** i **v2**
9. Utwórz vector składający się z 3 powtórzeń **d**.
10. Utwórz vector z zwiłokrotniając każdy element **d** dwukrotnie.
11. Sprawdź typ **d**
12. Utwórz vector **f** składający się z liczb całkowitych **od 5 do 9** przy użyciu funkcji **c()** i poprzez „:”
13. Dodaj vectory **v3** i **f**.
14. W **d** zwiększ wartości w indeksach parzystych o **3** a w nieparzystych o **5**
15. W **d** zwiększ co trzecią wartość o **15**

### 3.2 Część B

1. Utwórz vector składający się z **13** pierwszych małych liter alfabetu i nazwij go **l** (? **letters**)
2. Z **l** wybierz **3** pierwsze elementy
3. Z **l** wybierz 3, 5, i 9 element
4. Z **l** wybierz elementy o indeksach **parzystych**
5. Wybierz z **l** elementy powyżej „g”
6. Utwórz vector **q** o długości **20** elementów składający się z wartości **od -5 do 5**
7. Utwórz logiczny vector z wartością **true** dla elementów **q** większych od **2**
8. Z **q** wybierz elementy większe od **2**
9. Policz ile jest takich elementów
10. Z **q** wybierz co **3** element
11. Z **q** wybierz elementy których kwadrat jest większy od **7**
12. Utwórz vectory: **a1** z wartościami **od 4 do 7** i **a2** z wartościami **od 0 do 9**
13. Oblicz sumę elementów **a1** (? **sum**)
14. Posortuj vector **d** (? **order**)

## 4 Macierze

1. Utwórz macierz z **3** wierszy i **8** kolumn z wartościami **1:24**
2. Utwórz macierz z wartościami **1:24** z **4** kolumnami
3. Używając funkcji **dim()** utwórz macierz **m1** z vectora **1:10** o wymiarach **2** wiersze **5** kolumn
4. Używając funkcji **dim()** utwórz macierz **m2** z vectora **-4:3** o wymiarach **2** wiersze **4** kolumn
5. Połącz macierze **m1** i **m2**
6. Wyświetl wymiar otrzymanego wyniku

7. Utwórz macierz, której wierszami będą wektory: **1:5, 5:9, (1,5,3,2,6)**
8. Utwórz **3 wektory** od długości **10 elementów** o wartościach z zakresu **od -3 do 3**. Nazwij je **v1, v2, v3**
9. Utwórz macierz, której kolumnami będą wektory **v1, v2, v3**
10. Utwórz macierz **mt** o **3 kolumnach** i **20 wierszach**. W pierwszej kolumnie powinny pojawić się wartości kolejno **od 1 do 20**, w drugiej na przemian wartości **-5 i 6**, w trzeciej wartości od **40:59**.
11. Z macierzy **mt** wybierz pierwszy, trzeci i ostatni element
12. Z macierzy **mt** wybierz kolumny parzyste
13. Z **mt** wybierz wszystkie wiersze bez **3 i 17**
14. Dodaj do **mt** wiersz z liczbami z zakresu **od -600 do 300**
15. Z macierzy **mt** wybierz te kolumny, które w **4 wierszu** mają wartości z zakresu **od -10 do 280**
16. Z macierzy wybierz elementy **większe od 20**
17. Do kolumny **2** wiersz **1** wpisz wartość **42**
18. Do kolumny **3** wpisz wartość **5**
19. (\*) Dla wierszy **2 i 3** dla wartości **parzystych** wpisz wartość **2**. *Podpowiedź:* operator modulo (reszta z dzielnia): `%% (? '%%')`
20. (\*) Z macierzy **mt** wybierz te elementy z **3 wiersza** gdzie w wierszu pierwszym wartość jest podzielna przez **3**

## 5 Listy

1. Utwórz listę **l1** z **3 elementów** **a1**: vector numeryczny 5 elementowy, **a2**: 3 elementowy vector logiczny, **a3**: 13 elementowy vector znakowy
2. Wybierz element **a3** listy
3. Sprawdź typ wybranego elementu
4. Wybierz element **a1** jako vector
5. Wybierz pierwsze 3 elementy z **a3**
6. Wyświetl nazwy elementów listy
7. Dodaj do listy element „moj” w postaci macierzy **3 kolumny 4 wiersze** o wartościach z zakresu **1:12**
8. Wpisz wartość **4** do **3 elementu** wektora **a1**
9. Stwórz listę **l2** złożoną z **3 wektorów** liczbowych **1:43, 3:56, -5:8**
10. Połącz listę **l1** i **l2** w listę **l3**
11. (\*) Dodaj listę **l1** jako element **l2**
12. (\*) Wybierz z macierzy „moj” 2 wiersz i kolumny o wartościach powyżej średniej wartości elementów z **a1**

## 6 Factors

1. Utwórz vector **vf1** o długości **17** z elementami („mały”, „średni”, „duży”)
2. Przekształć **vf1** w factor **ff1** (bez kolejności)
3. Utwórz vector **vf2** o długości **14** z wartościami całkowitymi z przedziału **od 1 do 6**.
4. Z **vf2** utwórz factor **ff2** tak aby poziomy miały nazwy **od „F” do „A”**
5. Wyświetl poziomy **ff2**
6. Policz wystąpienia „C” w **ff2**
7. Z **vf2** utwórz ordered factor
8. Za pomocą `table()` wyświetl wynik dla **vf2**
9. Z **ff2** wybierz elementy **1, 5 i 12**
10. Z **ff2** wybierz elementy o wartościach „A” i „D”

## 7 Data Frame

1. Utwórz Data Frame **df** ze zmiennymi:
  - a. **C1** – vector 8 elementowy z imionami osób `c("Ala", "Beata", "Marek", "Igor", "Franek", "Adam", "Tomek", "Argh")`
  - b. **S1** – vector 8 elementowy z płcią poszczególnych osób `c("k", "k", "m", "m", "m", "m", "m", "m")`
  - c. **Hi** – vector z wzrostem w cm poszczególnych osób `sample(100:220, 8)`
2. Wyświetl strukturę **df**
3. Wyświetl elementy **df**
4. Wyświetl wymiar **df**
5. Z **df** wybierz **2 i 5** linie
6. Z **df** wybierz zmienną **Hi**
7. Wyświetl nazwy zmiennych z **df**
8. Z **df** wybierz linie dla których w **Hi** znajduje się wartość **większa od 150**

## 8 Import/Eksport

1. Utwórz wektor `v <- -100:100`, wektor logiczny `l <- c(T,F,T,T,T,F)`, wektor `c <- c("A ja jaj", "Oj", "Pff")`
2. Zapisz wektor **v** do pliku **vfile.RData**
3. Zapisz wektory **l** i **c** do plik **lc.RData**
4. Przypisz wektor **c** do **v**
5. Załaduj dane z pliku **vfile.RData**
6. Obejrzyj plik **c1.csv** i odczytaj z niego dane
7. Zapisz **v** do pliku **vfile.csv**
8. Załaduj dane z pliku **x1.xlsx**
9. Odczytaj dane z pliku **r.dbf** (użyj `read.dbf` z pakietu `foreign`)

## 9 Wartości specjalne

1. Utwórz **100** elementowy wektor **t** z wartościami z zakresu **od 20 do 90**
2. W **t** wpisz w wartość **NA** dla wartości z przedziału **od 60 do 80**
3. Użyj `is.na()` na wektorze **t**
4. Utwórz **20** elementowy vector **v** o wartościach z przedziału **od -7 do 3**
5. Policz pierwiastek z vectora **v**
6. Wypisz wartości dla których **pierwiastek** daje wartość **NaN**
7. Korzystając ze swojej wiedzy matematycznej utwórz vector **vnii** z wartościami **NaN, -Inf, Inf**
8. Sprawdź, które z wartości w tym wektorze są **Inf** i wpisz w nie wartość **42**.

## 10 Podstawowe funkcje matematyczne

1. Utwórz vector **a** jako **-12:12**
2. Wylicz wartości bezwzględne elementów **a**
3. Wylicz pierwiastek dla liczb z zakresu **od -3 do 10**
4. Dla każdego elementu **a** oblicz **exp** wynik przypisz do **ve**
5. Oblicz **logarytm** z liczby **3**
6. Użyj `sinpi()` do obliczenia sinusa **30 stopni**

7. Zaokrąglaj wartości **ve** do liczb całkowitych
8. Zaokrąglaj wartości **ve** do 3 miejsc po przecinku
9. Zaokrąglaj wartości **ve** do pełnych 100
10. Na wartościach **ve** użyj metody `floor()`, `ceiling()` i `trunc()`

## 11 Podstawowe funkcje znakowe

1. Utwórz wektor znakowy **s** o zawartości „DUŻE”, „małe”, „te inne 123456789”
2. Oblicz liczbę znaków elementów **s**
3. Utwórz **s1** jako połączenie elementów **s** oddzielonych spacją
4. Wyświetl **s1** w postaci małych liter
5. Utwórz **300 elementowy** wektor **a** o zawartości: „a.1”, „a.2”, ..., „a.300”
6. Znajdź indeks elementu „a.9”
7. W napisie „Ala ma kota. Ala ma rybki” zmień „kota” na „psa”

## 12 Podstawowe funkcji losujące/funkcje analityczne

1. Używając **rozkładu normalnego** wylosuj **30 liczb** i zapisz je w wektorze **v**
2. Oblicz **średnią** elementów **v**
3. Oblicz **medianę** elementów **v**
4. Oblicz **kwantyl** rzędu 0.1 wektora **v**
5. Oblicz odchylenie standardowe **v**
6. Utwórz wektor **7 elementowy p1** o zakresie wartości **od -10 do 10**
7. Utwórz wektor **-3:4** i nazwij go **p2**
8. Oblicz minimum z wartości **p1**
9. Wylicz zakres wartości wektora **v**
10. Oblicz minimalne wartości między wektorami **p1** i **p2**

## 13 Instrukcje sterujące

1. Utwórz wektor **v** o **10 elementach** z zakresu od -100:100
2. Napisz instrukcję warunkową tak by wyświetlić „**TAK**” jeśli średnia z elementów **v** jest większa niż **20** i „**nie**” w przeciwnym wypadku
3. Korzystając z pętli **for** napisz skrypt wyliczający sumę elementów **v** i wyświetlający ją na konsoli

## 14 Własne funkcje

1. Napisz funkcję **foo** wyświetlającą na konsolę wartość podanego parametru
2. Napisz funkcję **hgtg** zwracającą liczbę 42
3. Napisz funkcję **goo** zwracającą podany element wektora **-100:100**
4. Zmodyfikuj **goo** tak by obsługiwać przypadki szczególne