# Podstawy programowania w R - część 1

## Typy danych, podzbiory

1. Stwórz katalog na dysku (pierwsza litera imienia + nazwisko), który będzie Twoim Working Directory.

```
"F:/inazwisko"
```

2. Uruchom RStudio.
3. Wyświetl informacje o funkcji setwd().

```
?setwd
```

4. Ustaw swój Working Directory używając polecenia setwd().

```
setwd("F:/inazwisko")
```

5. Sprawdź, czy Working Directory został prawidłowo ustawiony wykorzystując polecenie getwd().

```
getwd()
```

6. Wykorzystaj operator przypisania – wprowadź wyrażenie, które przypisze wartość „36.6” zmiennej temp.

```
temp <- "36.6"
```

7. Wyświetl wartość zmiennej temp wpisując jej nazwę.

```
temp
```

8. Spróbuj dodać do stworzonej zmiennej wartość 0.1. Jak z pewnością zaobserwujesz operacja jest niemożliwa. Konsola powinna wypisać błąd: Error in "36.6" + 0.1 : non-numeric argument to binary operator.

```
"36.6" + 0.1
```

9. Sprawdź jakiego typu jest stworzona zmienna:

```
class(temp)
```

10. Przekonwertuj zmienną na typ numeric, a następnie sprawdź jej wartość. Zwróć uwagę jak zmienił się sposób jej wyświetlania.

```
temp <- as.numeric(temp)
```

```
temp
```

11. Przekonwertuj zmienną na typ integer, a następnie sprawdź jej wartość. Czy zmienił się jej sposób wyświetlania? Czy wartość została zaokrąglona?

```
temp <- as.integer(temp)
```

```
temp
```

12. Sprawdź typy poniższych wartości.

```
class("36")
class(36)
class(36L)
class(36+0i)
class(TRUE)
```

13. Stwórz wektor używając sekwencji i wypisz jego wartość:

```
v1 <- 3:0
v1
```

14. Sprawdź w jaki sposób wyświetlić wybrane elementy wektora wpisując poniższe wyrażenia:

```
v1[1]
v1[length(v1)]
v1[c(T,T,T,F)]
v1[3:1]
v1[-2]
```

15. Stwórz poniższe wektory używając funkcji c(), a następnie sprawdź ich typ i wartość.

```
v2 <- c(0/0, 1/0, 1/Inf, TRUE, as.numeric("abc"))
v3 <- as.logical(c("T", "False", "abc"))
```

16. Zwróć szczególną uwagę na wartości NA oraz NaN. Co zwróci funkcja is.na() jeśli jako argument przyjmie NaN? Co zwróci funkcja is.nan() jeśli jako argument przyjmie NA?

```
is.na(NaN)
is.nan(NA)
```

17. Stwórz pusty wektor używając funkcji vector(). Przypisz wartości 1,2,3,4,”null”,5,6,7,8,9. Jakiego typu są dane znajdujące się w wektorze po przypisaniu wartości? Przekonwertuj wektor na wektor o wartościach typu numeric, a następnie usuń wartości NA z wektora.

```
v4 <- vector("numeric", length=9)
v4[1:4] <- 1:4
v4[5] <- "null"
v4[6:9] <- 6:9
class(v4)
v4 <- as.numeric(v4)
bad <- is.na(v4)
v4 <- v4[!bad]
```

18. Stwórz wektory v5 oraz v6, a następnie sprawdź wyniki poniższych wyrażeń:

```
v5 <- 1:5

v6 <- 6:10

v5 + v6

v5 - v6

v5 * v6

v5 / v6

v5 == v6

v5 >= 3
```

19. Stwórz macierz wykorzystując funkcje cbind, rbind.

```
m1 <- cbind(v5,v6)

m2 <- rbind(v5,v6)
```

20. Wyświetl macierz m1, a następnie wyświetl element znajdujący się w macierzy m1 na pozycji 1,2. Czy pierwsza wartość określa wiersz, czy kolumnę? Jak wpływa na wynik parametr drop=FALSE?

```
m1

m1[1,2]

m1[1,2, drop=FALSE]
```

21. Wyświetl 2 i 3 wiersz macierzy.

```
m1[2:3,]
```

22. Stwórz macierz o 2 wierszach i 5 kolumnach. Uzupełnij macierz wartościami od 1 do 10. Zwróć uwagę, w jaki sposób zostały wpisane wartości do macierzy. Porównaj wynik z macierzą m1.

```
m3 <- matrix(1:10, nrow=2, ncol=5)
```

23. Stwórz 10 elementowy wektor, a następnie wykorzystaj go do stworzenia macierzy o 5 wierszach i 2 kolumnach. Zwróć uwagę, w jaki sposób zostały wpisane wartości do macierzy. Sprawdź atrybuty macierzy m4. Porównaj wynik z macierzą m2.

```
m4 <- 1:10

dim(m4) <- c(5,2)

attributes(m4)
```

24. Stwórz dwie macierze o wymiarze 2x2, a następnie pomnóż je ze sobą

```
m5 <- matrix(rep(1,4),nrow=2,ncol=2)

m6 <- matrix(rep(2,4),nrow=2,ncol=2)

m5 * m6
```

```
m5 %*% m6
```

25. Stwórz listę używając funkcji list. Sprawdź typ zmiennej oraz poszczególnych elementów. Zwróć uwagę, że lista może posiadać elementy różnego typu.

```
l1 <- list(id=1L, name="Kowalski", temp=36.6)

class(l1)

class(l1[[1]])

class(l1[[2]])
```

26. Sprawdź różne sposoby dostępu do elementów listy. Czy wszystkie poniższe wyrażenia umożliwiają dostęp do danych?

```
l1$name

l1["temp"]

arg <- "id"

l1$arg

l1[arg]

l1["arg"]

l1$i

l1[["i"]]

l1[["i", exact=FALSE]]
```

27. Stwórz dataframe, a następnie sprawdź jego atrybuty oraz liczbę wierszy i kolumn.

```
df <- data.frame(id=1:5, temp=c(36.6, NA, 37.2, 37.1, 36.8))

attributes(df)

nrow(df)

ncol(df)
```

28. Usuń wszystkie wiersze, w których występuje NA:

```
good <- complete.cases(df)

df <- df[good,]
```

29. Przekonwertuj dataframe na macierz:

```
m <- data.matrix(df)
```

30. Stwórz obiekt typu factor. Wypisz podsumowanie tabularyczne oraz zamień obiekt typu factor na wektor.

```
f <- factor(c("male","female","female","male"),
level=c("male","female"))

table(f)

v <- unclass(f)
```

31. Dla wybranych wektorów zmodyfikuj atrybut nazwy.

```
v <- c("Kowalski", "Jan")
names(v) <- c("nazwisko", "imie")
l <- list(nazwisko="Kowalski", imie="Jan")
m <- matrix(nrow=2, ncol=2)
m[1,1] <- "Kowalski"
m[1,2] <- "Jan"
m[2,1] <- "Nowak"
m[2,2] <- "Adam"
dimnames(m) <- list(c("1", "2"), c("nazwisko", "imie"))
```