Sławacka Weronika ćwiczenia 3

2023-10-27

zadanie 1

```
# Utwórz data frame planets.
nazwy_kolumn <- c("name", "type", "diameter", "rotation", "rings");
name <- c("Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune");
type <- c(rep("Terrestrial planet", 4), rep("Gas giant", 4));
diameter <- c(0.382, 0.949, 1.000, 0.532, 11.209, 9.449, 4.007, 3.883);
rotation <- c(58.64, -243.02, 1.00, 1.03, 0.41, 0.43, -0.72, 0.67);
rings <- c(rep(FALSE, 4), rep(TRUE, 4));
planets <- data.frame(name, type, diameter, rotation, rings, stringsAsFactors = FALSE);
colnames(planets) <- nazwy_kolumn;

planets;
```

```
##      name               type diameter rotation rings
## 1 Mercury Terrestrial planet    0.382    58.64 FALSE
## 2   Venus Terrestrial planet    0.949  -243.02 FALSE
## 3   Earth Terrestrial planet    1.000     1.00 FALSE
## 4    Mars Terrestrial planet    0.532     1.03 FALSE
## 5 Jupiter          Gas giant   11.209     0.41  TRUE
## 6  Saturn          Gas giant    9.449     0.43  TRUE
## 7  Uranus          Gas giant    4.007    -0.72  TRUE
## 8 Neptune          Gas giant    3.883     0.67  TRUE
```

```
# Wybierz średnicę dla Marsa.
planets$diameter[which(planets$name=="Mars")];
```

```
## [1] 0.532
```

```
# Wybierz całość danych dla Uranu.
subset(planets, planets$name=="Uranus");
```

```
##     name      type diameter rotation rings
## 7 Uranus Gas giant    4.007    -0.72  TRUE
```

```
# Sprawdź strukturę utworzonej data frame.
str(planets);
```

```
## 'data.frame':    8 obs. of  5 variables:
##  $ name    : chr  "Mercury" "Venus" "Earth" "Mars" ...
```

```
##  $ type    : chr  "Terrestrial planet" "Terrestrial planet" "Terrestrial planet" "Terrestrial planet
##  $ diameter: num  0.382 0.949 1 0.532 11.209 ...
##  $ rotation: num  58.64 -243.02 1 1.03 0.41 ...
##  $ rings   : logi  FALSE FALSE FALSE FALSE TRUE TRUE ...
```

```
# Wyświetl informacje o pierścieniach.
planets$rings;
```

```
## [1] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
```

```
# Utwórz zmienną planets_rings w której będą tylko te planety które mają pierścienie.
planets_rings=planets$name[which(planets$rings==TRUE)];
planets_rings;
```

```
## [1] "Jupiter" "Saturn"  "Uranus"  "Neptune"
```

```
# Utwórz podzbiór zawierający planety o średnicy mniejszej niż 1.
planety_o_srednicy_mniejszej_niz_1 <- subset(planets, planets$diameter<1);
planety_o_srednicy_mniejszej_niz_1;
```

```
##      name               type diameter rotation rings
## 1 Mercury Terrestrial planet    0.382    58.64 FALSE
## 2   Venus Terrestrial planet    0.949  -243.02 FALSE
## 4    Mars Terrestrial planet    0.532     1.03 FALSE
```

zadanie 2

```
# a)

# Utwórz speedfactor.
speed <- c("medium", "slow", "slow", "medium", "fast", "very_fast", "slow", "fast", "fast");
speed_values <- c("slow", "medium", "fast", "very_fast");
speedfactor <- factor(speed, levels=speed_values, ordered=TRUE);

speedfactor;
```

```
## [1] medium    slow      slow      medium    fast      very_fast slow
## [8] fast      fast
## Levels: slow < medium < fast < very_fast
```

```
# Wyświetl podsumowanie dla speedfactor.
summary(speedfactor);
```

```
##      slow    medium      fast very_fast
##         3         2         3         1
```

```r
# Zapisz 2. element faktora do zmiennej sf2.
sf2 <- speedfactor[2];

# Zapisz 6. element faktora do zmiennej sf6.
sf6 <- speedfactor[6];

# Sprawdź, który z tych elementów jest większy.
if (sf2 > sf6) {print("2. element jest większy")} else {print("6. element jest większy")};
```

```
## [1] "6. element jest większy"
```

```r
# b)

# Utwórz listę o nazwie film.
all_reviews <- c("Good", "OK", "Good", "Perfect", "Bad", "Perfect", "Good");
reviews_values <- c("Bad", "OK", "Good", "Perfect");
film <- list(title="The Shining", actors=c("Jack Nicholson", "Shelley Duvall", "Danny Lloyd", "Scatman C

film;
```

```
## $title
## [1] "The Shining"
##
## $actors
## [1] "Jack Nicholson"   "Shelley Duvall"   "Danny Lloyd"      "Scatman Crothers"
## [5] "Barry Nelson"
##
## $reviews
## [1] Good    OK      Good    Perfect Bad     Perfect Good
## Levels: Bad < OK < Good < Perfect
```

```r
# Wyświetl wszystkie opinie o filmie.
film$reviews;
```

```
## [1] Good    OK      Good    Perfect Bad     Perfect Good
## Levels: Bad < OK < Good < Perfect
```

```r
# Pokaż aktora, który znajduje się na drugiej pozycji na liście aktorów.
film$actors[2];
```

```
## [1] "Shelley Duvall"
```

```r
# Rozszerz listę film o własną opinię na temat filmu. Twoja opinia powinna być wyrażona w
# skali od 1 do 5 i zapisana jako liczba zmiennoprzecinkowa (typ double). Umieść tę opinię
# na końcu listy pod nazwą my_review.
film$my_review <- as.double(4);

film;
```

```
## $title
```

```
## [1] "The Shining"
##
## $actors
## [1] "Jack Nicholson"   "Shelley Duvall"   "Danny Lloyd"      "Scatman Crothers"
## [5] "Barry Nelson"
##
## $reviews
## [1] Good    OK      Good    Perfect Bad     Perfect Good
## Levels: Bad < OK < Good < Perfect
##
## $my_review
## [1] 4
```

```
typeof(film$my_review);
```

```
## [1] "double"
```

zadanie 3

```
# Wygeneruj 3 wektory (chisq, tstud, unif), składające się z 100 obserwacji z
# rozkładów chi kwadrat, t-studenta oraz równomiernego. Dobierz dowolnie
# argumenty funkcji. Skorzystaj z help("Distributions"). Zastosuj ziarno losowe
# (set.seed()). Sprawdź typ wygenerowanych wektorów.

set.seed(1);
chisq <- rchisq(100, 10);
typeof(chisq);
```

```
## [1] "double"
```

```
set.seed(2);
tstud <- rt(100, 10);
typeof(tstud);
```

```
## [1] "double"
```

```
set.seed(3);
unif <- runif(100, min=0, max=1);
typeof(unif);
```

```
## [1] "double"
```

```
# Utwórz wektory. Spróbuj zgadnąć, jakiego typu atomowego będzie każdy z nich.
# Zapisz odpowiedź w komentarzu. Następnie potwierdź swoją odpowiedź z
# wykorzystaniem dedykowanych funkcji.

vec_a<-c(TRUE, 2, 5); # typ double
typeof(vec_a);
```

```
## [1] "double"
```

```
vec_b<-c(3, "cztery", FALSE); # typ character
typeof(vec_b);
```

```
## [1] "character"
```

```
vec_c<-c(FALSE, 1L, 2.0, 3.0i); # typ complex
typeof(vec_c);
```

```
## [1] "complex"
```

```
vec_d<-c(FALSE, 1L, T); # typ integer
typeof(vec_d);
```

```
## [1] "integer"
```

```
# Ile wynosi suma elementów większych od 15 dla zmiennej numbers o
# wartościach: 20, 5, 18, 19, 8.5, 3, 4, 101, -2, 24, -30?
numbers <- c(20, 5, 18, 19, 8.5, 3, 4, 101, -2, 24, -30);
suma_elementow_wiekszych_od_15 <- sum(numbers[which(numbers>15)]);
suma_elementow_wiekszych_od_15;
```

```
## [1] 182
```

```
# Sprawdź jak wygląda wektor letters (wbudowany w R). Następnie uzupełnij go o
# polskie litery diakrytyzowane i przypisz do zmiennej letters_pl. Jaką długość ma
# nowo utworzony wektor?

letters;
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
letters_pl <- c(letters, "ą", "ć", "ę", "ł", "ń", "ó", "ś", "ź", "ż");
dlugosc_wektora <- length(letters_pl);
dlugosc_wektora;
```

```
## [1] 35
```