

Sławacka_Weronika_6

2023-11-25

#zadanie1

```
# a) Zapisz jako datę w odpowiednim formacie
# 30/Lipiec/2020
d1 <- "30/Lipiec/2020";
date1 <- as.Date(d1, format="%d/%B/%Y");
date1;
```

```
## [1] "2020-07-30"
```

```
# 1/15/2020
d2 <- "1/15/2020";
date2 <- as.Date(d2, format="%m/%d/%Y");
date2;
```

```
## [1] "2020-01-15"
```

```
# 1 Sty 20
d3 <- "1 Sty 20";
date3 <- as.Date(d3, format="%d %b %y");
date3;
```

```
## [1] "2020-01-01"
```

```
# b) Oblicz różnicę w dniach między dwoma datami (sprawdź ?difftime)
# day1: "2020-10-15"
# day5: "2020/02/11" (miesiąc listopad)
day1 <- as.Date("2020-10-15", format="%Y-%m-%d");
day5 <- as.Date("2020/02/11", format="%Y/%d/%m");
difftime(day1, day5, units="days");
```

```
## Time difference of -18 days
```

```
# c) Zapisz urodziny do jednej zmiennej bdays:
# Tukey- 1915-06-16, Fisher- 1890-02-17, Cramer-1893-09-25 i wyświetl,
# w jakim dniu tygodnia się urodzili

names <- c("Tukey", "Fisher", "Cramer");
strdates <- c("1915-06-16", "1890-02-17", "1893-09-25");
dates <- c(as.Date(strdates, format="%Y-%m-%d"));
bdays <- data.frame(names, dates);
```

```

for(i in 1:3) {
  cat(bdays$names[i], weekdays(bdays$dates[i]), "\n");
}

## Tukey środa
## Fisher poniedziałek
## Cramer poniedziałek

# d) Zapisz do zmiennej mydate 2020-4-19 7:01:00 (POSIXct)
mydate <- as.POSIXct("2020-4-19 7:01:00", format="%Y-%m-%d %H:%M:%S");
mydate;

## [1] "2020-04-19 07:01:00 CEST"

# e) Jaka będzie data za 100 dni od 2020-11-10?
data <- as.Date(100, origin="2020-11-10");
data;

## [1] "2021-02-18"

# f) Do zmiennej my_fav_dates zapisz 2005-4-19 8:01:00, 2006-4-19 11:01:00,
# 2008-4-19 7:01:00, sprawdź jakie są średnia i maksymalna różnica pomiędzy
# datami (sprawdź ?diff)
my_fav_dates <- as.POSIXct(c("2005-4-19 8:01:00", "2006-4-19 11:01:00",
"2008-4-19 7:01:00"), format="%Y-%m-%d %H:%M:%S");
roznice <- diff(my_fav_dates);
srednia_roznica <- mean(roznice);
srednia_roznica;

## Time difference of 547.9792 days

max_roznica <- max(roznice);
max_roznica;

## Time difference of 730.8333 days

# g) Oblicz jaka ilość sekund dzieli zmienne zawierające informacje o dacie
# i godzinie: 2000-11-10 12:00:00 i 2020-11-10 14:00:00
info1 <- as.POSIXct("2000-11-10 12:00:00", format="%Y-%m-%d %H:%M:%S");
info2 <- as.POSIXct("2020-11-10 14:00:00", format="%Y-%m-%d %H:%M:%S");
difftime(info1, info2, units="secs");

## Time difference of -631159200 secs

#zadanie2

library(lubridate);

## Warning: pakiet 'lubridate' został zbudowany w wersji R 4.3.2

```

```
##
## Dołączanie pakietu: 'lubridate'

## Następujące obiekty zostały zakryte z 'package:base':
##
##     date, intersect, setdiff, union

# a) Zapisz datę swoich urodzin do b_day, następnie sprawdź informacje
# o zapisanej dacie (dzień, miesiąc, rok, czy był to rok przestępny...)
b_day <- dmy(30012004);
day(b_day);

## [1] 30

month(b_day);

## [1] 1

year(b_day);

## [1] 2004

wday(b_day, label=TRUE);

## [1] pt\\.
## Levels: niedz\\. < pon\\. < wt\\. < śr\\. < czw\\. < pt\\. < sob\\.

leap_year(b_day);

## [1] TRUE

# b) Utwórz wektor tekstowy zawierający daty: 30-2020-01, 15/2020/02
daty <- c("30-2020-01", "15/2020/02");

# c) Następnie utwórz z niego datę z pakietem lubridate i zapisz
# do zmiennej date4.
date4 <- dym(daty);

# d) Zapisz obecną godzinę i datę zajęć z uwzględnieniem strefy czasowej
# do zmiennej class_d_h

# zapisuję datę i godzinę wykonania ćwiczenia (poza zajęciami)
class_d_h <- now();

# e) Zmień strefę czasową na Australię i zapisz wynik do class_d_h_a
class_d_h_a <- with_tz(class_d_h, tzone="Australia/Canberra");

# f) Zmień strefę czasową, bez zmiany godziny class_d_h_f
class_d_h_f <- force_tz(class_d_h, tzone="Australia/Canberra");

# g) Dodaj do class_d_h 2 godziny (poszukaj funkcji w dokumentacji)
class_d_h + hours(2);
```

```
## [1] "2023-11-27 23:14:58 CET"
```

```
# h) Zaokrąglaj class_d_h do godzin  
round_date(class_d_h, unit="hour");
```

```
## [1] "2023-11-27 21:00:00 CET"
```

```
# i) Utwórz zmienne date_Auckland i date_Arizona: 2020-10-10 10:10:00  
# z właściwymi strefami czasowymi i sprawdź różnicę godzin między zmiennymi.  
data <- ymd_hms("2020-10-10 10:10:00");  
date_Auckland <- with_tz(data, tzone="Pacific/Auckland");  
date_Auckland;
```

```
## [1] "2020-10-10 23:10:00 NZDT"
```

```
date_Arizona <- with_tz(data, tzone="US/Arizona");  
date_Arizona;
```

```
## [1] "2020-10-10 03:10:00 MST"
```

```
difftime(date_Auckland, date_Arizona, units="hours")
```

```
## Time difference of 0 hours
```

```
# j) Sprawdź jaka jest różnica dni między datą Twoich urodzin (zmienna może  
# wymagać modyfikacji), a class_d_h  
difftime(b_day, class_d_h, units="days");
```

```
## Time difference of -7241.844 days
```

```
# k) Zapisz datę 2021-11-10 na kilka sposobów z wykorzystaniem pakietu lubridate  
s1 <- ymd("2021-11-10");  
s2 <- dmy("10/11/2021");  
s3 <- mdy("11102021");
```

```
#zadanie3
```

```
library(readr);
```

```
## Warning: pakiet 'readr' został zbudowany w wersji R 4.3.2
```

```
library(readxl);
```

```
## Warning: pakiet 'readxl' został zbudowany w wersji R 4.3.2
```

```
# Wczytaj plik hotdogs.txt  
hotdogs <- read.delim("hotdogs.txt", sep=";", header=FALSE, fill=TRUE);  
  
# Zobacz podsumowanie danych.  
str(hotdogs);
```

```
## 'data.frame': 54 obs. of 3 variables:
## $ V1: chr "Beef" "Beef" "Beef" "Beef" ...
## $ V2: int 186 181 176 149 184 190 158 139 175 148 ...
## $ V3: int 495 477 425 322 482 587 370 322 479 375 ...
```

```
# Użyj funkcji which.min i which.max aby wyszukać hot-dogi z najmniejszą liczbą
# kalorii oraz najwyższą zawartościę sodu
colnames(hotdogs) <- c("rodzaj_miesa", "zawartosc_sodu", "ilosc_kalorii");
hotdog_najmniej_kalorii=hotdogs[which.min(hotdogs$ilosc_kalorii),];
hotdog_najmniej_kalorii;
```

```
##      rodzaj_miesa zawartosc_sodu ilosc_kalorii
## 33      Meat          107          144
```

```
hotdog_najwiecej_sodu=hotdogs[which.max(hotdogs$zawartosc_sodu),];
hotdog_najwiecej_sodu;
```

```
##      rodzaj_miesa zawartosc_sodu ilosc_kalorii
## 34      Meat          195          511
```

```
#zadanie4
```

```
# a) Zapisz do zmiennej potatoes_frag plik potatoes.csv, funkcję read_csv(),
# pomiń pierwsze 6 wierszy, pobierz następne 50, nadaj nazwy zmiennych podane
# w przykładach.
properties <- c("area", "temp", "size", "storage", "method",
"texture", "flavor", "moistness");
potatoes_frag <- read_csv("potatoes.csv", col_names=properties, skip=6,
n_max=50);
```

```
## Rows: 50 Columns: 8
## -- Column specification -----
## Delimiter: ","
## dbl (8): area, temp, size, storage, method, texture, flavor, moistness
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
potatoes_frag;
```

```
## # A tibble: 50 x 8
##       area temp size storage method texture flavor moistness
##   <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1     1     1     1     2       1     1.8     3     1.7
## 2     1     1     1     2       2     2.6     3.1   2.4
## 3     1     1     1     2       3     3       3     2.9
## 4     1     1     1     2       4     2.2     3.2   2.5
## 5     1     1     1     2       5     2       2.8   1.9
## 6     1     1     1     3       1     1.8     2.6   1.5
## 7     1     1     1     3       2     2       2.8   1.9
## 8     1     1     1     3       3     2.6     2.6   2.6
```

```
## 9      1      1      1      3      4      2.1      3.2      2.1
## 10     1      1      1      3      5      2.5      3      2.1
## # i 40 more rows
```

```
# b) Zapisz do zmiennej potatoes_char plik potatoes.txt, ustal wszystkie kolumny
# jako character (c)
potatoes_char <- read_tsv("potatoes.txt", col_types = "cccccccc");

# c) Zapisz zmienną potatoes_frag do pliku csv Funkcja write_csv()
write_csv(potatoes_frag, "potatoes_frag.csv");

# d) Zainstaluj i uruchom bibliotekę readxl
library(readxl);
```

#zadanie5

```
# Wykorzystaj funkcję excel_sheets() do podglądu skoroszytów w pliku
# urbanpop.xlsx, a następnie z wykorzystaniem funkcji read_excel() wczytaj
# skoroszyty do zmiennych pop1, pop2, pop3 i złącz je w jednej liście. Wykonaj
# podgląd danych.
excel_sheets("urbanpop.xlsx");
```

```
## [1] "1960-1966" "1967-1974" "1975-2011"
```

```
pop1 <- read_excel("urbanpop.xlsx", sheet="1960-1966");
pop2 <- read_excel("urbanpop.xlsx", sheet="1967-1974");
pop3 <- read_excel("urbanpop.xlsx", sheet="1975-2011");
lista <- list(pop1=pop1, pop2=pop2, pop3=pop3);
print(lista);
```

```
## $pop1
## # A tibble: 209 x 8
##   country      '1960'      '1961'      '1962'      '1963'      '1964'      '1965'      '1966'
##   <chr>         <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Afghanistan  769308    814923.    858522.    9.04e5    9.51e5    1.00e6    1.06e6
## 2 Albania      494443    511803.    529439.    5.47e5    5.66e5    5.84e5    6.03e5
## 3 Algeria      3293999   3515148.   3739963.   3.97e6    4.22e6    4.49e6    4.65e6
## 4 American Samoa      NA      13660.    14166.    1.48e4    1.54e4    1.60e4    1.67e4
## 5 Andorra       NA      8724.    9700.    1.07e4    1.19e4    1.31e4    1.42e4
## 6 Angola        521205    548265.    579695.    6.12e5    6.45e5    6.79e5    7.18e5
## 7 Antigua and Barbuda  21699    21635.    21664.    2.17e4    2.18e4    2.19e4    2.20e4
## 8 Argentina    15224096  15545223.  15912120.   1.63e7    1.67e7    1.70e7    1.74e7
## 9 Armenia      957974    1008597.   1061426.   1.12e6    1.17e6    1.23e6    1.28e6
## 10 Aruba        24996     28140.    28533.    2.88e4    2.89e4    2.91e4    2.93e4
## # i 199 more rows
##
## $pop2
## # A tibble: 209 x 8
##   country      '1967'      '1968'      '1969'      '1970'      '1971'      '1972'      '1973'      '1974'
##   <chr>         <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Afghanistan  1.12e6    1.18e6    1.25e6    1.32e6    1.41e6    1.50e6    1.60e6    1.70e6
## 2 Albania      6.21e5    6.40e5    6.59e5    6.78e5    6.99e5    7.20e5    7.42e5    7.63e5
```

```

## 3 Algeria 4.83e6 5.02e6 5.22e6 5.43e6 5.62e6 5.82e6 6.02e6 6.24e6
## 4 American Samoa 1.73e4 1.80e4 1.86e4 1.92e4 1.98e4 2.03e4 2.07e4 2.12e4
## 5 Andorra 1.54e4 1.67e4 1.81e4 1.95e4 2.09e4 2.24e4 2.39e4 2.55e4
## 6 Angola 7.57e5 7.98e5 8.41e5 8.86e5 9.55e5 1.03e6 1.10e6 1.18e6
## 7 Antigua and Barbuda 2.21e4 2.21e4 2.22e4 2.22e4 2.26e4 2.29e4 2.32e4 2.35e4
## 8 Argentina 1.78e7 1.81e7 1.85e7 1.89e7 1.93e7 1.98e7 2.02e7 2.07e7
## 9 Armenia 1.34e6 1.39e6 1.45e6 1.51e6 1.56e6 1.62e6 1.68e6 1.74e6
## 10 Aruba 2.94e4 2.96e4 2.97e4 2.99e4 3.01e4 3.03e4 3.05e4 3.06e4
## # i 199 more rows
##
## $pop3
## # A tibble: 209 x 38
##   country '1975' '1976' '1977' '1978' '1979' '1980' '1981' '1982' '1983' '1984'
##   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Afghan~ 1.79e6 1.91e6 2.02e6 2.14e6 2.27e6 2.40e6 2.49e6 2.59e6 2.69e6 2.80e6
## 2 Albania 7.85e5 8.08e5 8.31e5 8.54e5 8.78e5 9.02e5 9.27e5 9.52e5 9.78e5 1.01e6
## 3 Algeria 6.46e6 6.77e6 7.10e6 7.45e6 7.81e6 8.19e6 8.64e6 9.11e6 9.59e6 1.01e7
## 4 Americ~ 2.16e4 2.20e4 2.25e4 2.29e4 2.35e4 2.42e4 2.52e4 2.63e4 2.77e4 2.91e4
## 5 Andorra 2.70e4 2.84e4 2.97e4 3.10e4 3.26e4 3.44e4 3.64e4 3.86e4 4.10e4 4.32e4
## 6 Angola 1.27e6 1.37e6 1.48e6 1.60e6 1.72e6 1.86e6 2.02e6 2.19e6 2.37e6 2.56e6
## 7 Antigu~ 2.38e4 2.40e4 2.42e4 2.43e4 2.44e4 2.43e4 2.42e4 2.39e4 2.36e4 2.33e4
## 8 Argent~ 2.11e7 2.16e7 2.20e7 2.24e7 2.29e7 2.33e7 2.38e7 2.43e7 2.48e7 2.53e7
## 9 Armenia 1.80e6 1.85e6 1.90e6 1.95e6 2.00e6 2.05e6 2.08e6 2.12e6 2.16e6 2.20e6
## 10 Aruba 3.07e4 3.06e4 3.05e4 3.04e4 3.03e4 3.03e4 3.06e4 3.09e4 3.14e4 3.17e4
## # i 199 more rows
## # i 27 more variables: '1985' <dbl>, '1986' <dbl>, '1987' <dbl>, '1988' <dbl>,
## # '1989' <dbl>, '1990' <dbl>, '1991' <dbl>, '1992' <dbl>, '1993' <dbl>,
## # '1994' <dbl>, '1995' <dbl>, '1996' <dbl>, '1997' <dbl>, '1998' <dbl>,
## # '1999' <dbl>, '2000' <dbl>, '2001' <dbl>, '2002' <dbl>, '2003' <dbl>,
## # '2004' <dbl>, '2005' <dbl>, '2006' <dbl>, '2007' <dbl>, '2008' <dbl>,
## # '2009' <dbl>, '2010' <dbl>, '2011' <dbl>

```