

Sławacka Weronika ćwiczenia 4

2023-11-08

#zadanie1

```
# Utwórz wektor exercf zawierający 200 obserwacji z rozkładu normalnego  
# o wartości średniej wynoszącej 5 i odchyleniu standardowym wynoszącym 10.  
# Ustaw ziarno losowe na 12.
```

```
set.seed(12);  
exercf <- rnorm(200, mean=5, sd=10);
```

```
# Wypisz wszystkie wartości ujemne.  
exercf[which(sign(exercf)==-1)];
```

```
## [1] -9.80567595 -4.56744479 -4.20005248 -14.97642097 -1.28255237  
## [6] -2.77719582 -7.93882298 -2.79566508 -2.03464254 -5.25244839  
## [11] -0.41028649 -5.70492158 -5.04451202 -6.55992890 -10.95625656  
## [16] -4.77053283 -5.50890062 -8.14272797 -16.49260002 -0.25400626  
## [21] -1.33838203 -7.71053787 -7.74059551 -0.41888860 -4.63398332  
## [26] -4.84673797 -1.94737942 -5.07298960 -0.51457761 -6.69235514  
## [31] -0.27675029 -0.24525579 -6.82413972 -8.54708484 -9.30479047  
## [36] -7.99669102 -13.33469910 -2.35378912 -7.62820466 -1.99478364  
## [41] -7.95311285 -3.97059867 -13.84776116 -4.07416653 -11.47561703  
## [46] -0.15263232 -15.50134377 -11.86741024 -12.70192882 -0.60520433  
## [51] -0.08129562 -4.38833088 -1.52816889 -1.11595292 -0.15233958  
## [56] -2.99804142 -2.68470970 -10.43152723 -8.20600907 -1.75821505  
## [61] -0.13872181 -4.59253963 -9.40450924 -3.68897484
```

```
# Policz, ile wartości jest nieujemnych.  
sum(sign(exercf)>=0);
```

```
## [1] 136
```

```
# Wyznacz wartość średnią wszystkich wartości.  
mean(exercf);
```

```
## [1] 4.892623
```

```
# Wyznacz wartość średnią arytmetyczną wartości bezwzględnych elementów.  
mean(abs(exercf));
```

```
## [1] 8.414796
```

```
# Wypisz na ekran wektor powstały w wyniku przekształcenia z exercf
# do przedziału [0,1].
exercf_w_przedziale <- (exercf-min(exercf))/(max(exercf)-min(exercf));
print(exercf_w_przedziale);
```

```
## [1] 0.15617265 0.87030502 0.27851117 0.28709160 0.03541026 0.43836363
## [7] 0.42830871 0.35522961 0.47709362 0.60192074 0.32032231 0.19977287
## [13] 0.31989097 0.50474951 0.46636148 0.33766459 0.77962009 0.58148459
## [19] 0.62036025 0.43345697 0.55418948 0.97073869 0.73830521 0.43131904
## [25] 0.26251296 0.43951064 0.45545719 0.53258177 0.53600965 0.58651815
## [31] 0.65936602 0.98588070 0.37560132 0.25194548 0.41497117 0.38865376
## [37] 0.56613384 0.38996836 0.68835513 0.26736930 0.52647718 0.23197685
## [43] 0.63698113 0.12930092 0.42990737 0.60693064 0.27376807 0.54633205
## [49] 0.67278861 0.38691200 0.49198915 0.47564405 0.60864987 0.97380599
## [55] 0.25652354 0.67353567 0.62789959 0.19501068 0.44356181 0.57534045
## [61] 0.59690688 0.73420443 0.70182231 0.54799751 0.69681422 0.69972544
## [67] 0.95833811 0.00000000 0.72876265 0.76938652 0.37925123 0.56042027
## [73] 0.40167057 0.45933085 0.47783010 0.35392571 0.20510445 0.41228684
## [79] 0.62264614 0.46039375 0.50295265 0.20440246 0.45475545 0.77391838
## [85] 0.49649794 0.71148852 0.46068431 0.76206417 0.37540042 0.27695717
## [91] 0.58987745 0.27198830 0.71158256 0.53214736 0.74337880 0.42201676
## [97] 0.60758816 0.33970262 0.44613673 0.26670421 0.95194203 0.51396971
## [103] 0.92246926 0.37316561 0.52696171 0.68625756 0.44491252 0.59168315
## [109] 0.22888404 0.85654966 0.51900289 0.37872005 0.37945560 0.67639634
## [115] 0.22580622 0.18556696 0.45237905 0.48105984 0.56198795 0.78321854
## [121] 0.46131335 0.66315784 0.59850387 0.16787079 0.19842137 0.82970957
## [127] 0.45072909 1.00000000 0.07375256 0.47614872 0.68078931 0.57751246
## [133] 0.33021095 0.76953963 0.43697602 0.74878274 0.58779093 0.69989463
## [139] 0.20702734 0.62532749 0.75638839 0.33859550 0.83383786 0.83804070
## [145] 0.19943913 0.29245047 0.06177003 0.50838030 0.29003165 0.11717129
## [151] 0.56089202 0.65197024 0.42093150 0.38161881 0.59495026 0.72466081
## [157] 0.56272279 0.02315072 0.56921355 0.58649029 0.10802099 0.79323383
## [163] 0.08853086 0.55400135 0.37104902 0.38328488 0.28269437 0.88396296
## [169] 0.34949325 0.69795594 0.35912053 0.41514394 0.38162565 0.42213836
## [175] 0.66168694 0.68841051 0.31516448 0.55558817 0.89748513 0.32248232
## [181] 0.14155593 0.19353276 0.34412054 0.41624432 0.38194369 0.27792509
## [187] 0.47354372 0.65748734 0.79126034 0.77846673 0.85189947 0.43443806
## [193] 0.95123400 0.62305641 0.61326567 0.57263108 0.69375651 0.59364480
## [199] 0.16554187 0.29902778
```

```
# Zapisz exercf jako nowy wektor liczb całkowitych exercfint.
exercfint <- as.integer(exercf);
exercfint;
```

```
## [1] -9 20 -4 -4 -14 2 1 -1 3 9 -2 -7 -2 5 3 -2 16 8
## [19] 10 2 7 25 15 1 -5 2 3 6 6 8 11 25 0 -5 1 0
## [37] 7 0 12 -5 6 -6 10 -10 1 9 -4 6 12 0 4 3 9 25
## [55] -5 12 10 -8 2 8 9 14 13 6 13 13 24 -16 14 16 0 7
## [73] 0 3 3 -1 -7 1 10 3 5 -7 2 16 4 13 3 16 0 -4
## [91] 8 -4 13 6 15 1 9 -1 2 -5 24 5 23 0 6 12 2 8
## [109] -6 20 5 0 0 12 -6 -8 2 4 7 17 3 11 9 -9 -7 19
## [127] 2 26 -13 3 12 8 -2 16 2 15 8 13 -7 10 15 -1 19 19
## [145] -7 -3 -13 5 -4 -11 7 11 1 0 8 14 7 -15 7 8 -11 17
```

```
## [163] -12  7  0  0 -4 21 -1 13 -1  1  0  1 11 12 -2  7 21 -2
## [181] -10 -8 -1  1  0 -4  3 11 17 16 19  2 24 10  9  8 13  8
## [199] -9 -3
```

```
# Wykonaj podzbiór danych składający się z 50 obserwacji, zastosuj metodę
# ze zwracaniem.
```

```
podzbior <- sample(exercf, 50, replace=TRUE);
podzbior;
```

```
## [1] 13.97559402 -0.08129562  4.57315088 -9.40450924 -0.15263232
## [6]  7.52340039 25.07201457 13.34325038 11.42314265 -10.43152723
## [11] 13.34325038  1.27543268 25.72035768 -12.70192882  1.57710726
## [16]  0.14858645  7.52340039  3.96689534  8.14204596  8.67514437
## [21]  1.57710726 -2.68470970  7.87970718 12.98105326  1.97540755
## [26] -10.95625656 12.31453357 16.88879156  9.28014802  1.91496344
## [31] 24.23687218 -9.30479047 -9.80567595  3.23275545 12.98105326
## [36]  6.89997859  9.13384582 21.35649384  1.16049606  3.78336328
## [41] -0.15263232 -4.77053283 -13.84776116  8.14204596  8.02603747
## [46]  1.28283029 -5.25244839  6.89997859  6.97128917 -4.38833088
```

```
# Utwórz wektor składający się z mediany, minimum, odchylenia standardowego
# i maksimum.
```

```
wektor <- c(median(exercf), min(exercf), sd(exercf), max(exercf));
wektor;
```

```
## [1]  4.036040 -16.492600  9.323812 26.324911
```

```
#zadanie2
```

```
# Napisz funkcję zmieniającą odległość podaną w metrach na cale.
```

```
meters_to_inches <- function(m){return (m*39.3701)};
```

```
meters_to_inches(2);
```

```
## [1] 78.7402
```

```
meters_to_inches(50);
```

```
## [1] 1968.505
```

```
# Napisz funkcję odejmującą od wektora liczb wartość średnią i wyświetlającą
# nowo powstałą zmienną wraz ze zmienną pierwotną.
```

```
funkcja <- function(x){y <- (x-mean(x)); cat("pierwotny wektor: ", x, "\nwektor po odjeciu sredniej: ",
funkcja(c(1, 2, 3, 4, 5, 6)));
```

```
## pierwotny wektor:  1 2 3 4 5 6
```

```
## wektor po odjeciu sredniej: -2.5 -1.5 -0.5 0.5 1.5 2.5
```

```
#zadanie3
```

```
# Sprawdź, jak wyglądają dane (wczytaj pierwsze 10 wierszy dla flights).
library(nycflights13);
```

```
## Warning: pakiet 'nycflights13' został zbudowany w wersji R 4.3.2
```

```
flights[1:10,];
```

```
## # A tibble: 10 x 19
##   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>   <int>         <int>         <dbl>   <int>         <int>
## 1  2013     1     1     517           515           2     830           819
## 2  2013     1     1     533           529           4     850           830
## 3  2013     1     1     542           540           2     923           850
## 4  2013     1     1     544           545          -1    1004          1022
## 5  2013     1     1     554           600          -6     812           837
## 6  2013     1     1     554           558          -4     740           728
## 7  2013     1     1     555           600          -5     913           854
## 8  2013     1     1     557           600          -3     709           723
## 9  2013     1     1     557           600          -3     838           846
## 10 2013     1     1     558           600          -2     753           745
## # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dtm>
```

```
# Wczytaj dane z pliku cars.txt.
setwd("C:/Users/huawei/Desktop");
cars_dane <- read.delim("cars.txt", sep=";", header=FALSE);

# Dodaj nazwy kolumn i ponumeruj wiersze.
nazwy_kolumn <- c("mpg", "cylinders", "cubicinches", "hp", "weightlbs", "time-to-60", "year", "brand");
colnames(cars_dane) <- nazwy_kolumn;
rownames(cars_dane) <- c(1:nrow(cars_dane));

cars_dane;
```

```
##   mpg cylinders cubicinches  hp weightlbs time-to-60 year   brand
## 1 23.5         4         150 95    2700      14.5 1978   Toyota
## 2 20.5         6         200 85    2965      15.8 1978    Ford
## 3 22.3         4         140 88    2720      15.4 1978 Chevrolet
## 4 20.2         6         232 90    3265      18.2 1978    AMC
## 5 15.8         8         350 145   4082      13.7 1978 Pontiac
## 6 19.0         6         250 110   3158      15.8 1978   Dodge
```