

Funkcja dokonująca porządkowania za pomocą metody sum

Wstęp

Na początku użytkownik przed skorzystaniem z funkcji musi zaimportować dane, które chce poddać porządkowaniu. W tym celu może użyć np. takiego wywołania z podaniem swojej ścieżki pliku formatu xlsx, który chce poddać porządkowaniu. Jako przykład zaimportowałam tabele zawierającą 8 obiektów, będących ofertami sprzedaży aut.

```
library(readxl)
zbior_danych <- read_excel("datasets/8_Rozniacych_sie_obiektow.xlsx",
  sheet = "Arkusz1", col_types = c("numeric","text",
    "text", "text", "text", "text",
    "numeric", "numeric", "text", "numeric",
    "text", "text", "text", "text","text",
    "text", "numeric", "text", "text",
    "text", "numeric", "numeric",
    "numeric", "numeric", "numeric"))
```

`col_type = "blank"` deprecated. Use "skip" instead.

Podgląd danych:

```
head(zbior_danych)
```

```
## # A tibble: 6 x 29
##   Nr MARKA  MODEL WERSJA TYP   WOJE~ `CENA~ `MOC~ `POJE~ ROK.~ `PRZE~
##   <dbl> <chr>  <chr>  <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1.00 Mazda  3      II   kompa~ zach~ 25200 105 1598 2009 1.36e5
## 2  2.00 Jaguar XF      X260 kombi doln~ 323600 240 1999 2017 1.00e3
## 3  3.00 Subaru B9 Tr~ <NA>  suv   malo~ 38900 245 3000 2007 1.33e5
## 4  4.00 Volks~ Golf  VII   kombi lodz~ 113900 150 1395 2017 1.00e0
## 5  5.00 Peuge~ 508    <NA> kombi slas~ 42500 115 1560 2014 1.57e5
## 6  6.00 Opel  Antara <NA>  suv   lodz~ 24000 150 1991 2007 1.42e5
## # ... with 18 more variables: KOLOR <chr>, L.DZRZWI <dbl>,
## #   RODZAJ.PALIWA <chr>, SKRZYNIA.BIEGOW <chr>, NAPED <chr>,
## #   KRAJ.AKTUALNEJ.REJESTRACJI <chr>, KRAJ.POCHODZENIA <chr>,
## #   STATUS.POJAZDU.SPROWADZONEGO <chr>, PIERWSZY.WLASCICIEL <dbl>,
## #   KTO.SPRZEDAJE <chr>, STAN <chr>, SERWISOWANY <chr>, ABS <dbl>,
## #   KOMPUTER.POKLADOWY <dbl>, ESP <dbl>, KLIMATYZAJCA <dbl>,
## #   BEZWYPADKOWY <dbl>, USZKODZONY <dbl>
```

Podzbiór danych

W kolejnym, kroku po przyjrzeniu się zbiorowi danych, użytkownik musi zdecydować na których danych ilościowych chce pracować - ważna jest znajomość danych. Dodatkowo pierwszą kolumną musi być kolumna zawierająca numery indeksów obiektów, ze względu na to, że w wyniku zastosowania funkcji odpowiedzialnej za porządkowanie, zostaną zwrócone w kolejności malejącej numery indeksów, mówiące o kolejności uporządkowania. W związku z tym, za pomocą poniższej procedury użytkownik tworzy podzbiór zaimportowanego

zbioru, gdzie w miejsce “” wpisuje nazwy kolumn zawierających zmienne ilościowe, wybrane do porządkowania (przyjmijmy założenie, że podzbiór będzie nazywał się dane_porzadkowanie - będzie to pomocne w dalszej części programu). U mnie wybranymi kolumnami są: cena, moc, pojemność, rok produkcji, przebieg.

```
dane_porzadkowanie<-zbior_danych[c("Nr", "CENA.BRUTTO_[pln]", "MOC_[km]",  
                                "POJEMNOSC.SKOKOWA_[cm3]",  
                                "ROK.PRODUKCJI", "PRZEBIEG_[km]")]
```

Transformacje danych

Chcąc zastosować metodę sum do uporządkowania zbioru, wymagany jest aby zmienne miały charakter stymulant oraz by zostały poddane transformacji normalizacyjnej. Aby funkcja dokonująca porządkowania dawała poprawny wynik, użytkownik musi zająć się transformacją przed jej zastosowaniem. Poniżej podałam tego przykład. Dla zmiennych które stymulantami nie są, należy dokonać stymulacji. Wśród moich zmiennych poddanych porządkowaniu, do stymulant nie należy zmienna zmienna: przebieg - jest destymulantą, w związku z tym, została przekształcona na stymulante, za pomocą przekształcenia różnicowego podanego w postaci funkcji:

```
stymulacja_przekształcenie_roznicowe<-function(x,y){  
  max_wartosc=max(x[which(colnames(x)==y)])  
  for (i in 1:nrow(x)){  
    x[i,which(colnames(x)==y)]=max_wartosc-x[i,which(colnames(x)==y)]  
  }  
  return(x)  
}
```

Gdy użytkownik chce skorzystać z tej funkcji, w miejsce x musi wpisać nazwę zbioru, a w miejsce y nazwę kolumny w “”, którą chce poddać stymulacji.

UWAGA - kolumny wymagające stymulacji, muszą zostać osobno poddane działaniu poniższej funkcji, dodatkowo po każdym zastosowaniu funkcji, należy nadpisać zbiór by zmienny zostały zapisane.

```
dane_porzadkowanie<-stymulacja_przekształcenie_roznicowe(dane_porzadkowanie, "PRZEBIEG_[km]")
```

W celu uzyskania porównywalności między zmiennymi, zostały one poddane unitaryzacji, czyli od wartości zmiennej dla danego obiektu, odejmujemy minimalną wartość danej cechy, a następnie różnica ta jest dzielona przez różnicę między maksymalną i minimalną wartość danej cechy. Poniżej została zaprezentowana funkcja dokonująca unitaryzacji.

```
unitaryzacja<-function(x){  
  maks=0  
  min=0  
  for (j in 2:ncol(x)){  
    maks[j]=max(x[j])  
    min[j]=min(x[j])  
    for (i in 1:nrow(x)){  
      x[i,j]=(x[i,j]-min[j])/(maks[j]-min[j])  
    }  
  }  
  return(x)  
}
```

Funkcja porządkująca metodą sum

W celu dokonania porządkowania na unormowanych danych, należy zastosować poniższą funkcję tj. `funkcja_porzadzowanie_metoda_sum`, zwraca ona numery indeksów obiektów wg kolejności, która uzyskały one po uporządkowaniu. Funkcja ta jest postaci:

```
funkcja_porzadzowanie_metoda_sum<-function(x){
  x<-unitaryzacja(x) #unitaryzacja stymulant

  #sztywne zalozenie___ostania kolumna to zmienna_syntetyczna -za pomoca metody
#sredniej arytmetycznej
  x[, "zmienna_syntetyczna"] <-0
  for(i in 1:nrow(x)){
    for(j in 2:(ncol(x)-1)){
      x[i,ncol(x)]=x[i,ncol(x)]+x[i,j]
      j=j+1
    }
    x[i,ncol(x)]=x[i,ncol(x)]/(ncol(x)-2)
#-2 bo interesuje nas ilosc zmiennych, poza nr indeksu i kolumna zmienna syntetyczna
  }

  #wyeliminowanie ujemnych wartosci zmiennej syntetycznej
  min_zmienna=min(x$zmienna_syntetyczna)
  for(i in 1:nrow(x)){
    x[i,ncol(x)]=x[i,ncol(x)]-min_zmienna
  }

  #ostatnie przekształcenie normalizacja zm. syntetycznej
  max_zmienna=max(x$zmienna_syntetyczna)
  for(i in 1:nrow(x)){
    x[i,ncol(x)]=x[i,ncol(x)]/max_zmienna
  }
  x<-x[order(-x$zmienna_syntetyczna),]
  return(x[1])
}
```

Wywołanie funkcji dla zbioru dane_porzadzowanie - podzbioru wyjściowych danych

```
funkcja_porzadzowanie_metoda_sum(dane_porzadzowanie)
```

```
## # A tibble: 8 x 1
##       Nr
##   <dbl>
## 1  2.00
## 2  4.00
## 3  3.00
## 4  5.00
## 5  6.00
## 6  1.00
## 7  8.00
## 8  7.00
```

Funkcja zwraca nam indeksy uporządkowanych obiektów, tj. 1-wsze miejsce zajął obiekt z numerem indeksu 2, 2-gie miejsce obiekt z numerem indeksu równym 3, z kolei miejsce ostatnie zajął obiekt o numerze indeksu równym 7.