Transformacje zmiennych

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("~/Praca licencjacka/R studio/repozytorium/datasets/8_Rozniacych_sie_obiektow
```

Podgląd danych:

```
head(zbior_danych)
```

```
## # A tibble: 6 x 30
##      Nr MARKA     MODEL    WERSJA TYP    WOJEWODZTWO     `CENA.NETTO_[pln~
##   <dbl> <chr>     <chr>    <chr>  <chr>  <chr>           <lgl>
## 1  1.00 Mazda     3        II     kompa~ zachodniopomor~ NA
## 2  2.00 Jaguar    XF       X260   kombi  dolnoslaskie    NA
## 3  3.00 Subaru    B9 Trib~ <NA>   suv    malopolskie     NA
## 4  4.00 Volkswag~ Golf     VII    kombi  lodzkie         NA
## 5  5.00 Peugeot   508      <NA>   kombi  slaskie         NA
## 6  6.00 Opel      Antara   <NA>   suv    lodzkie         NA
## # ... with 23 more variables: `CENA.BRUTTO_[pln]` <dbl>, `MOC_[km]` <dbl>,
## #   `POJEMNOSC.SKOKOWA_[cm3]` <dbl>, ROK.PRODUKCJI <dbl>, `PRZEBIEG_[km]`
## #   <dbl>, 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 <dbl>, ABS <dbl>, KOMPUTER.POKLADOWY <dbl>, ESP <dbl>,
## #   KLIMATYZAJCA <dbl>, BEZWYPADKOWY <dbl>, USZKODZONY <lgl>
```

Podzbiór danych

W kolejnym, kroku po przyjrzeniu się zbiorowi danych, użytkownik musi zadecydować 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]")]
```

Stymulacja zmiennych

Poniżej zostaną przedstawione metody stymulacji zmiennych, ograniczam się do przypadku, że dana zmienna jest destymulantą i należy przeprowadzić dla niej stymulacje. W tym celu stworzyłam dwie funkcje: stymulacja_przeksztalcenie_ilorazowe(x,y) oraz stymulacja_przeksztalcenie_roznicowe(x,y). W miejscu argumentu x należy wpisać nazwę zbioru na którym dokonywane jest porządkowanie, z kolei w miejscu argumentu y należy podać nazwę kolumny poddanej stymulacji, z tym że nazwa kolumny musi zostać podana w "".

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

```
stymulacja_przeksztalcenie_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)
}
```

UWAGA! Stymulacja zmiennych dokonywana jest pojedynczo, tj. jeżeli w naszym zbiorze jest wiele zmienych mających charakter destymulant, dla każdej z nich musimy użyć funkcji a na sam koniec nadpisać nasz zbiór, tym nowym wystumulowanym, dzięki czemu przekształcenia zostaną zapisane.

```
#charakter destymulanty ma zmienna: "PRZEBIEG_[km]", w zwiazku z czym to ona zostanie poddana stymulacj

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

Transformacje normalizacyjne

W celu uzyskania porównywalności między zmiennymi, zostały one poddane transofrmacji normalizacyjnej - standaryzacji, unitaryzacji lub przekształceniu ilorazowemu. W zależności od charakteru zmiennych, użytkownik musi wybrać jedną z tych metod dla zmiennych, które będą wykorzystywane w porządkowaniu.

Standaryzacja

```
standaryzacja<-function(x){
  suma=0
  srednia=0

  odchylenie=0
  for (j in 2:ncol(x)){
    suma[j]=sum(x[j])
    srednia[j]=suma[j]/nrow(x)
    suma_kwadratow=0
    kwadrat=0
    for(i in 1:nrow(x)){
      kwadrat=(x[i,j]-srednia[j])^2
      suma_kwadratow=suma_kwadratow+kwadrat
```

```
    }

    odchylenie[j]=sqrt(suma_kwadratow/nrow(x))

    for (i in 1:nrow(x)){
      x[i,j]=(x[i,j]-srednia[j])/odchylenie[j]
    }
  }
  return(x)
}
```

Unitaryzacja

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

Przekształcenie ilorazowe

```
przeksztalcenie_ilorazowe<-function(x){
    suma=0
    srednia=0
    for (j in 2:ncol(x)){
      suma[j]=sum(x[j])
      srednia[j]=suma[j]/nrow(x)

    for(i in 1:nrow(x)){
      x[i,j]=x[i,j]/srednia[j]
      }
    }
  return(x)
}
```

Wywoływanie funkcji

Aby wywołać funkcję należy podać jej nazwę, a następnie w miejsce argumentu wpisać nazwę zbioru na którym użytkownik pracuje. Np. dla wywołania funckji przeksztalcenie_ilorazowe:

```
przeksztalcenie_ilorazowe(dane_porzadkowanie)
```

```
## # A tibble: 8 x 6
```

```
##      Nr `CENA.BRUTTO_[pln~ `MOC_[km]` `POJEMNOSC.SKOKOWA_[c~ ROK.PRODUKCJI
##   <dbl>              <dbl>      <dbl>                  <dbl>         <dbl>
## 1  1.00             0.347      0.730                  0.910         1.000
## 2  2.00             4.45       1.67                   1.14          1.00
## 3  3.00             0.535      1.70                   1.71          0.999
## 4  4.00             1.57       1.04                   0.795         1.00
## 5  5.00             0.584      0.800                  0.889         1.00
## 6  6.00             0.330      1.04                   1.13          0.999
## 7  7.00             0.0316     0.278                  0.513         0.996
## 8  8.00             0.155      0.730                  0.910         0.997
## # ... with 1 more variable: `PRZEBIEG_[km]` <dbl>
```

Oczywiście po wybraniu odpowiedniego sposobu transformacji normalizacyjnej, należy nadpisać wystymulowany zbiór, który będzie dalej poddany porządkowaniu.

```
dane_porzadkowanie<-przeksztalcenie_ilorazowe(dane_porzadkowanie)
```