

Eksperyment 2

Wstęp

Zostanie tutaj zaprezentowane porównanie wyników porządkowania liniowego całego zbioru dla par: metoda rang i metoda sum, metoda rang i metoda Hellwiga, metoda sum i metoda Hellwiga. Na początku użytkownik 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.

```
library(readxl)
zbior_danych <- read_excel("datasets/zbior_danych.xlsx",
                           sheet = "DANE_INNA_WERSJA")
```

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>                <dbl>
## 1  1.00 Hyundai i20    II    kompakt malopolskie      NA
## 2  2.00 Hyundai i20    I     kompakt mazowieckie     NA
## 3  3.00 Subaru  Legacy V      kombi  mazowieckie      NA
## 4  4.00 Ford    Mondeo Mk4    sedan  dolnoslaskie     NA
## 5  5.00 Opel    Astra  G      kompakt slaskie        NA
## 6  6.00 Mazda   Premacy <NA> minivan dolnoslaskie     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 <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

Przed zastosowaniem metod, należy dokonać ich transformacji. W pierwszym kroku należy dokonać stymulacji destymulant - dla metody rang i metody Hellwiga zostanie użyte przekształcenie ilorazowe, dla metody sum przekształcenie ilorazowe. Po stymulacji można przejść do transformacji normalizacyjnej - metoda sum i metoda rang za pomocą normalizacji, natomiast metoda Hellwiga za pomocą standaryzacji. Do powyższych operacji, zostaną wykorzystane wcześniej zbudowane ogólne funkcje odpowiedzialne za to.

```
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)
}
```

```
stymulacja_przekształcenie_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)
}
```

```
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)
}
```

```
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)
}

```

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_rang<-stymulacja_przekształcenie_ilorazowe(dane_porzadkowanie,"PRZEBIEG_[km]")
dane_hellwig<-stymulacja_przekształcenie_ilorazowe(dane_porzadkowanie,"PRZEBIEG_[km]")
dane_sum<-stymulacja_przekształcenie_roznicowe(dane_porzadkowanie,"PRZEBIEG_[km]")

```

Teraz dokonamy przekształcenia normalizacyjnego wystymulowanych podzbiorów.

```

dane_sum<-unitaryzacja(dane_sum)
dane_rang<-unitaryzacja(dane_rang)
dane_hellwig<-standaryzacja(dane_hellwig)

```

Porządkowanie

W tej części wystymulowane podzbiory zostaną poddane porządkowaniu. W tym celu zostaną użyte poniższe funkcje odpowiedzialne za porządkowanie metodą sum, rang oraz Hellwiga. Wszystkie funkcje zwracają numery indeksów uporządkowanych obiektów, wg tendencji malejącej w hierarchii, ze względu na wartość zmiennej syntetycznej.

```

funkcja_porzadkowanie_metoda_rang<-function(x){
  y<-x #dzięki temu nie bede sztywno odwoływac sie do 1kolumny rang
  for (i in 2:ncol(x)){
    x[ncol(x)+1]=rank(-x[i])
  }
  #ostania kolumna to zmienna_syntetyczna - za pomoca metody sredniej arytmetycznej
  x[, "zmienna_syntetyczna"] <-0
  for(i in 1:nrow(x)){
    for(j in (ncol(y)+1):(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)-7)
  }
  x<-x[order(x$zmienna_syntetyczna),]
  print("Numery indeksów obiektów po uporządkowaniu: ")
  return(x[1])
}

```

```

funkcja_porzadkowanie_metoda_sum<-function(x){
  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]
    }
    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])
}

funkcja_porzadkowanie_metoda_Hellwiga<-function(x){
  #wyznaczenie obiektu wzorcowego, zmienne maja charakter stymulant wiec wspolrzedne
  obiekt_wz=0
  for (j in 2:ncol(x)){ #od 2kolumny bo 1 kolumna to Nr - index
    obiekt_wz[j]=max(x[j])
  }
  odleg<- x[c("Nr" )]
  for (i in 1:nrow(x)){
    SUMKA=0
    for (j in 2:ncol(x)){
      SUMKA=SUMKA+(x[i,j]-obiekt_wz[j])^2
    }
    odleg[i,2]=sqrt(SUMKA) #kolumna zawierajaca odleglosci
  }
  d_0=0
  suma=0
  srednia=0
  odchylenie=0
  for (j in 2:ncol(odleg)){
    suma[j]=sum(odleg[j])
    srednia[j]=suma[j]/nrow(odleg)
    suma_kwadratow=0
    kwadrat=0
    for(i in 1:nrow(odleg)){
      kwadrat=(odleg[i,j]-srednia[j])^2
      suma_kwadratow=suma_kwadratow+kwadrat
    }

    odchylenie[j]=sqrt(suma_kwadratow/nrow(odleg))
    d_0=srednia[j]+2*odchylenie[j] #d_0 to po prostu wartosc
  }
  #ostatnia kolumna to jak zawsze zmienna syntetyczna
  x[, "zmienna_syntetyczna"] <-0
  for (i in 1:nrow(x)){
    x[i,ncol(x)]=1-(odleg[i,2]/d_0)
  }
}

```

```
x<-x[order(-x$zmienna_syntetyczna),]
return(x[1])
}
```

Zastosowanie funkcji do znormalizowanych podzbiorów.

```
dane_sum<-funkcja_porzadzowanie_metoda_sum(dane_sum)
dane_rang<-funkcja_porzadzowanie_metoda_rang(dane_rang)
```

```
## [1] "Numery indeksów obiektów po uporządkowaniu: "
dane_hellwig<-funkcja_porzadzowanie_metoda_Hellwiga(dane_hellwig)
```

Porównanie wyników uporządkowania

W celu porównania wyników uporządkowania zostały stworzone trzy tabele pomocnicze. Każda tabela zawiera trzy kolumny, w dwóch pierwszych kolumnach znajdują się indeksy obiektów po uporządkowaniu. Z kolei w kolumnie trzeciej-“porównanie” znajduje się jedna z dwóch wartości: 0 lub 1. Odpowiednio wartość 1 jest przypisywana tym rekordom, dla których zgadza się kolejność uporządkowania przy zastosowaniu dwóch różnych metod porządkowania. Dodatkowo zostały również stworzone trzy tabele o nazwie “podsumowanie”. W nich

Para 1: metoda rang i metoda sum

```
tabela_porownawcza=data.frame(dane_rang,dane_sum)
names(tabela_porownawcza)<-c("dane_rang", "dane_sum")
tabela_porownawcza$porownanie=0

#inwersja
for(i in 1:nrow(tabela_porownawcza)){
  if(tabela_porownawcza$dane_sum[i]==tabela_porownawcza$dane_rang[i]){
    tabela_porownawcza$porownanie[i]=1
  }
}
head(tabela_porownawcza,15)
```

```
##      dane_rang dane_sum porownanie
## 1          49        49          1
## 2          50        50          1
## 3          53        53          1
## 4          48        16          0
## 5          52        48          0
## 6          16        52          0
## 7          57        38          0
## 8          38        57          0
## 9          37        37          1
## 10         58        58          1
## 11         44        44          1
## 12         17        46          0
## 13         56        17          0
## 14         20         1          0
## 15         46        41          0
```

```
#podsumowanie
podsumowanie=as.data.frame(table(tabela_porownawcza$porownanie))
names(podsumowanie)<-c("wartość","ilosc wystąpień")
podsumowanie

##    wartość ilosc wystąpień
## 1         0          51
## 2         1          10
```

Para 2: metoda rang i metoda Hellwiga

```
tabela_porownawcza=data.frame(dane_rang,dane_hellwig)
names(tabela_porownawcza)<-c("dane_rang","dane_hellwig")
tabela_porownawcza$porownanie=0

#inwersja
for(i in 1:nrow(tabela_porownawcza)){
  if(tabela_porownawcza$dane_hellwig[i]==tabela_porownawcza$dane_rang[i]){
    tabela_porownawcza$porownanie[i]=1
  }
}
head(tabela_porownawcza,15)
```

```
##    dane_rang dane_hellwig porownanie
## 1         49          53           0
## 2         50          49           0
## 3         53          50           0
## 4         48          16           0
## 5         52          57           0
## 6         16          48           0
## 7         57          52           0
## 8         38          58           0
## 9         37          38           0
## 10        58          37           0
## 11        44          46           0
## 12        17          54           0
## 13        56          56           1
## 14        20          36           0
## 15        46           7           0
```

```
#podsumowanie
podsumowanie=as.data.frame(table(tabela_porownawcza$porownanie))
names(podsumowanie)<-c("wartość","ilosc wystąpień")
podsumowanie
```

```
##    wartość ilosc wystąpień
## 1         0          56
## 2         1           5
```

Para 3: metoda sum i metoda hellwiga

```
tabela_porownawcza=data.frame(dane_sum,dane_hellwig)
names(tabela_porownawcza)<-c("dane_sum","dane_hellwig")
tabela_porownawcza$porownanie=0

#inwersja
for(i in 1:nrow(tabela_porownawcza)){
  if(tabela_porownawcza$dane_hellwig[i]==tabela_porownawcza$dane_sum[i]){
    tabela_porownawcza$porownanie[i]=1
  }
}
head(tabela_porownawcza,15)
```

```
##   dane_sum dane_hellwig porownanie
## 1      49          53           0
## 2      50          49           0
## 3      53          50           0
## 4      16          16           1
## 5      48          57           0
## 6      52          48           0
## 7      38          52           0
## 8      57          58           0
## 9      37          38           0
## 10     58          37           0
## 11     44          46           0
## 12     46          54           0
## 13     17          56           0
## 14      1          36           0
## 15     41           7           0
```

```
#podsumowanie
podsumowanie=as.data.frame(table(tabela_porownawcza$porownanie))
names(podsumowanie)<-c("wartość","ilosc wystąpień")
podsumowanie
```

```
##   wartość ilosc wystąpień
## 1      0           59
## 2      1            2
```

Podsumowanie

Na podstawie powyższych wyników zauważamy, że najwięcej zgodności wyniku porządkowania jest widoczne dla pary pierwszej - metody rang oraz metody sum. W przypadku kolejnych par, zgodność ta jest już niewielka, ze względu na to że w wyniku uporządkowania obiekty zajmują np. o miejsce niżej lub wyżej w stosunku do drugiej metody.