

**Marek Walesiak**

Akademia Ekonomiczna we Wrocławiu

## **WYBRANE ZAGADNIENIA KLASYFIKACJI OBIEKTÓW Z WYKORZYSTANIEM PROGRAMU KOMPUTEROWEGO CLUSTER`SIM` DLA ŚRODOWISKA R**

### **1. Wstęp**

Pakiet `clusterSim`, napisany w programie R dostępnym na stronie <http://cran.r-project.org>, składa się z funkcji podstawowej `cluster.Sim` oraz z szesnastu funkcji pomocniczych. Funkcja podstawowa służy poszukiwaniu optymalnej procedury klasyfikacyjnej (spośród różnych kombinacji formuł normalizacyjnych, miar odległości i metod klasyfikacji) dla danego typu danych z punktu widzenia wybranego indeksu jakości klasyfikacji.

W artykule zaprezentowana zostanie szczegółowa charakterystyka wybranych funkcji pomocniczych pakietu `clusterSim`. Szczegółowy opis pakietu `clusterSim` znajduje się w pracach: [Walesiak, Dudek 2006a; 2006b; 2007]. Program dostępny jest na stronie <http://www.ae.jgora.pl/keii/clusterSim>. Funkcjonujące na rynku pakiety statystyczne (np. SPSS, Statistica, S-Plus, SAS) nie udostępniają takich możliwości, jakie niesie z sobą pakiet `clusterSim` oraz środowisko programistyczne R. W pakiecie `clusterSim` dostępnych jest m.in. jedenaście formuł normalizacyjnych, cztery miary odległości oraz siedem indeksów jakości klasyfikacji. Inne miary odległości oraz metody klasyfikacji (m.in. hierarchiczne metody aglomeracyjne, metoda  $k$ -średnich i metoda  $k$ -medoids) dostępne są w pakietach `stats` i `cluster`. W zasadniczej części artykułu zaproponowane zostaną przykładowe składnie poleceń (procedury) z wykorzystaniem wybranych funkcji z pakietu `clusterSim`, w tym w szczególności `dist.GDM`. Procedury te, mające zastosowanie nie tylko w odniesieniu do miary GDM, ułatwią potencjal-

nemu użytkownikowi realizację wielu zagadnień klasyfikacyjnych niedostępnych w podstawowych pakietach statystycznych.

## 2. Charakterystyka funkcji pakietu **clusterSim**

Pakiet `clusterSim` w wersji 0.30-2 składa się z funkcji podstawowej `cluster.Sim` oraz z szesnastu funkcji pomocniczych: `data.Normalization`, `dist.BC`, `dist.GDM`, `dist.SM`, `index.G1`, `index.G2`, `index.G3`, `index.S`, `index.KL`, `index.H`, `index.Gap`, `cluster.Description`, `initial.Centers`, `HINoV.Mod`, `HINoV.Symbolic`, `replication.Mod`. Szczegółowo scharakteryzowane zostaną funkcje niezbędne z punktu widzenia dalszych rozważań:

- `data.Normalization (x, type="n0")` – funkcja dokonująca normalizacji danych według jednej z formuł `n0`–`n11` ( $x$  – macierz danych, `type` – typ formuły z tab. 1).
- `dist.GDM (x, method="GDM1")` – funkcja obliczająca macierz odległości według miary GDM (`GDM1` – dla danych metrycznych, a `GDM2` dla danych porządkowych) – zob. [Walesiak 2006].
- `index.G1 (x, cl)` – funkcja obliczająca wartości indeksu Calińskiego-Harabasa dla macierzy danych  $x$  i ustalonego podziału zbioru obiektów na klasy `cl`.
- `index.G2 (d, cl)` – funkcja obliczająca wartości indeksu Bakera-Huberta dla macierzy odległości  $d$  i ustalonego podziału zbioru obiektów na klasy `cl`.
- `index.G3 (d, cl)` – funkcja obliczająca wartości indeksu Huberta-Levine’a dla macierzy odległości  $d$  i ustalonego podziału zbioru obiektów na klasy `cl`.
- `index.S (d, cl)` – funkcja obliczająca wartości indeksu Silhouette Rousseuwa dla macierzy odległości  $d$  i ustalonego podziału zbioru obiektów na klasy `cl`.
- `index.KL (x, clall)` – funkcja obliczająca wartości indeksu Krzanowskiego-Lai dla macierzy danych  $x$  oraz trzech wektorów liczb całkowitych informujących o przynależności poszczególnych obiektów do klas w podziale na  $u-1$ ,  $u$  i  $u+1$  klas (`clall`).
- `index.H (x, clall)` – funkcja obliczająca wartości indeksu Hartigana dla macierzy danych  $x$  oraz dwóch wektorów liczb całkowitych informujących o przynależności poszczególnych obiektów do klas w podziale na  $u$  i  $u+1$  klas (`clall`).
- `index.Gap (x, clall, reference.distribution="unif", B=10, method="pam")` – funkcja obliczająca wartości indeksu Gap dla

macierzy danych  $x$ , dwóch wektorów liczb całkowitych informujących o przynależności poszczególnych obiektów do klas w podziale na  $u$  i  $u+1$  klas (`clall`), sposobu generowania obserwacji na zmiennych z rozkładu jednostajnego (`unif`, `pc`), liczby generowanych zbiorów obserwacji ( $B=10$ ) oraz przyjętej metody klasyfikacji (`ward`, `single`, `complete`, `average`, `mcquitty`, `median`, `centroid`, `pam`, `k-means`).

Tabela 1. Formuły normalizacyjne dla danych metrycznych

| Nr  | Nazwa formuły                       | Formuła                                                       | Skala pomiaru zmiennych  |                 |
|-----|-------------------------------------|---------------------------------------------------------------|--------------------------|-----------------|
|     |                                     |                                                               | przed normalizacją       | po normalizacji |
| n0  | bez normalizacji                    | –                                                             | ilorazowa / przedziałowa | –               |
| n1  | standaryzacja                       | $z_{ij} = (x_{ij} - \bar{x}_j) / s_j$                         | ilorazowa / przedziałowa | przedziałowa    |
| n2  | standaryzacja Webera                | $z_{ij} = (x_{ij} - Me_j) / 1,4826 \cdot MAD_j$               | ilorazowa / przedziałowa | przedziałowa    |
| n3  | unitaryzacja                        | $z_{ij} = (x_{ij} - \bar{x}_j) / r_j$                         | ilorazowa / przedziałowa | przedziałowa    |
| n4  | unitaryzacja zerowa                 | $z_{ij} = [x_{ij} - \min_i \{x_{ij}\}] / r_j$                 | ilorazowa / przedziałowa | przedziałowa    |
| n5  | normalizacja w przedziale $[-1; 1]$ | $z_{ij} = (x_{ij} - \bar{x}_j) / \max_i  x_{ij} - \bar{x}_j $ | ilorazowa / przedziałowa | przedziałowa    |
| n6  | przekształcenia ilorazowe           | $z_{ij} = x_{ij} / s_j$                                       | ilorazowa                | ilorazowa       |
| n7  |                                     | $z_{ij} = x_{ij} / r_j$                                       | ilorazowa                | ilorazowa       |
| n8  |                                     | $z_{ij} = x_{ij} / \max_i \{x_{ij}\}$                         | ilorazowa                | ilorazowa       |
| n9  |                                     | $z_{ij} = x_{ij} / \bar{x}_j$                                 | ilorazowa                | ilorazowa       |
| n10 |                                     | $z_{ij} = x_{ij} / \sum_{i=1}^n x_{ij}$                       | ilorazowa                | ilorazowa       |
| n11 |                                     | $z_{ij} = x_{ij} / \sqrt{\sum_{i=1}^n x_{ij}^2}$              | ilorazowa                | ilorazowa       |

$x_{ij}(z_{ij})$  – wartość (znormalizowana wartość)  $j$ -tej zmiennej dla  $i$ -tego obiektu,

$\bar{x}_j$  ( $s_j$ ,  $r_j$ ) – średnia (odchylenie standardowe, rozstęp) dla  $j$ -tej zmiennej,

$Me_j$  ( $MAD_j$ ) – mediana Webera (medianowe odchylenie bezwzględne dla  $j$ -tej zmiennej).

Źródło: opracowanie własne.

Charakterystykę siedmiu indeksów jakości klasyfikacji znajdujących zastosowanie do ustalenia liczby klas zarówno w przypadku metod optymalizacyjnych ( $k$ -średnich,  $k$ -medoids), jak i hierarchicznych zawiera tab. 2.

- `cluster.Description(x, cl, sdType="sample")` – funkcja obliczająca osobno dla każdej klasy i zmiennej z ustalonego podziału zbioru obiektów na klasy `cl` następujące statystyki opisowe: średnią arytmetyczną, odchylenie standardowe, medianę, medianowe odchylenie bezwzględne, dominantę (dla zmiennych nominalnych i porządkowych, jeśli występuje więcej wartości o maksymalnej częstości występowania, to zwracana jest wartość „N.A.”).

Tabela 2. Indeksy oceny jakości klasyfikacji

| Lp. | Nazwa indeksu             | Formuła                                                                                                           | Kryterium wyboru liczby klas                    |
|-----|---------------------------|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| 1   | Calińskiego<br>i Harabasz | $G1(u) = \frac{B_u/(u-1)}{W_u/(n-u)}, G1(u) \in R_+$                                                              | $\hat{u} = \arg \max_u \{G1(u)\}$               |
| 2   | Bakera<br>i Huberta       | $G2(u) = \frac{s(+) - s(-)}{s(+) + s(-)}, G2(u) \in [-1, 1]$                                                      | $\hat{u} = \arg \max_u \{G2(u)\}$               |
| 3   | Huberta<br>i Levine       | $G3(u) = \frac{D(u) - r \cdot D_{\min}}{r \cdot D_{\max} - r \cdot D_{\min}}, G3(u) \in (0, 1)$                   | $\hat{u} = \arg \min_u \{G3(u)\}$               |
| 4   | Silhouette                | $S(u) = \frac{1}{n} \sum_{i=1}^n \frac{b(i) - a(i)}{\max\{a(i); b(i)\}}, S(u) \in [-1, 1]$                        | $\hat{u} = \arg \max_u \{S(u)\}$                |
| 5   | Krzanowskiego<br>i Lai    | $KL(u) = \left  \frac{DIFF_u}{DIFF_{u+1}} \right , KL(u) \in R_+$<br>$DIFF_u = (u-1)^{2/m} W_{u-1} - u^{2/m} W_u$ | $\hat{u} = \arg \max_u \{KL(u)\}$               |
| 6   | Hartigana                 | $H(u) = \left( \frac{W_u}{W_{u+1}} - 1 \right) (n - u - 1), H(u) \in R_+$                                         | najmniejsze $u$ , dla którego<br>$H(u) \leq 10$ |
| 7   | Gap                       | $Gap(u) = \frac{1}{B} \sum_{b=1}^B \log W_{ub} - \log W_u, Gap(u) \in R$                                          | najmniejsze $u$ , dla którego<br>$diffu \geq 0$ |

$B_u$  – macierz kowariancji międzyklasowej,  $W_u$  – macierz kowariancji wewnątrzklasowej,  $tr$  – ślad macierzy,  $B_u(W_u) = tr(B_u)(tr W_u)$ ,  $r, s = 1, \dots, u$  – numer klasy,  $u$  – liczba klas,  $i, k = 1, \dots, n$  – numer obiektu,  $n$  – liczba obiektów,  $m$  – liczba zmiennych,  $s(+)$  – liczba par odległości zgodnych,  $s(-)$  – liczba par odległości niezgodnych,  $D(u)$  – suma wszystkich odległości wewnątrzklasowych,  $r$  – liczba odległości wewnątrzklasowych,  $D_{\min}$  ( $D_{\max}$ ) – najmniejsza (największa) odległość wewnątrzklasowa,  $a(i) = \sum_{k \in \{P_s, \forall i\}} d_{ik} / (n_r - 1)$  – średnia odległość obiektu  $i$  od pozostałych obiektów należących do klasy  $P_r$ ;  $b(i) = \min_{s \neq r} \{d_{iP_s}\}$ ,  $d_{iP_s} = \sum_{k \in P_s} d_{ik} / n_s$  – średnia odległość obiektu  $i$  od obiektów należących do klasy  $P_s$ ,  $B$  – liczba generowanych zbiorów obserwacji,  $s_u = sd_u \sqrt{1 + 1/B}$ ,  $sd_u$  – odchylenie standardowe z wartości  $\{\log W_{ub}\}$ ;  $diffu = Gap(u) - Gap(u+1) + s_{u+1}$ .

Źródło: opracowanie własne na podstawie prac: [Caliński, Harabasz 1974; Hubert 1974; Milligan, Cooper 1985; Kauffman, Rousseeuw 1990; Hartigan 1975; Tibshirani, Walther, Hastie 2001].

W odchyleniu standardowym w mianowniku występować będzie  $n-1$  dla próby (`sdType="sample"`) i  $n$  dla populacji (`sdType="population"`).

Każda z funkcji może być wykorzystana w innych procedurach programu R, po uruchomieniu pakietu `clusterSim`.

### 3. Charakterystyka przykładowych składni poleceń z wykorzystaniem funkcji pakietu `clusterSim`

Zwykle macierz danych, stanowiąca punkt wyjścia zastosowania metod klasyfikacji, przygotowywana jest w arkuszu kalkulacyjnym MS Excel. W związku z tym, że nie ma bezpośrednich procedur wczytywania zawartości skoroszytów MS Excel do programu R, najwygodniejsze jest zapisanie danych z arkusza w formacie `csv`. Zapisanie danych w Excelu wymaga wybrania polecenia Plik|Zapisz jako... i wybrania odpowiedniego formatu.

Do wczytania danych w programie R służy instrukcja `read.csv2` (w języku polskim symbolem ułamka dziesiętnego jest przecinek). Podane parametry instrukcji `read.csv2` oznaczają, że dane w pliku `csv` zawierają nagłówek (`header=TRUE`) oraz w pierwszej kolumnie znajdują się nazwy obiektów (`row.names=1`). Zawsze należy podać pełną ścieżkę dostępu do pliku z danymi z separatorem nazwy oznaczonym „/”. Macierz danych w przykładowym pliku `csv` jest następująca (macierz danych zawiera 7 obiektów i 2 zmienne):

```
;x1;x2
1;3;4
2;2,5;5
3;2,5;3,5
4;10;2
5;9;1,5
6;4;11
7;4,5;10.
```

Składnia poleceń pozwalająca na wczytanie danych oraz obliczenie odległości między obiektami (funkcja `dist.GDM`) i zestawienie ich w macierz jest następująca<sup>1</sup>:

```
> library(cluster)
> library(clusterSim)
> x <- read.csv2("C:/Dane_7x2.csv", header=TRUE,
  strip.white = TRUE, row.names=1)
> x <- as.matrix(x)
> z <- data.Normalization (x, type="n1")
> z <- as.data.frame(z)
> d <- dist.GDM (z, method="GDM1").
```

W przedstawionej składni poleceń przyjęto następujące założenia:

- do normalizacji wartości zmiennych zastosowano formułę klasycznej standaryzacji „n1” (inne formuły znajdują się w tab. 1),
- do pomiaru odległości zastosowano miarę `GDM1` – zob. m.in. [Walesiak 2006].

---

<sup>1</sup> Miary odległości dla zmiennych ilorazowych lub przedziałowych dostępne są w procedurze `dist`, a dla zmiennych binarnych – w procedurze `dist.binary`.

Po zastosowaniu tej procedury (dodając na końcu polecenie `print(d)`) otrzymuje się w rezultacie macierz odległości:

|   | 1            | 2            | 3            | 4            | 5            | 6           |
|---|--------------|--------------|--------------|--------------|--------------|-------------|
| 2 | 0.013077019  |              |              |              |              |             |
| 3 | 0.0044437960 | 0.019931539  |              |              |              |             |
| 4 | 0.4681415790 | 0.5618657820 | 0.501412714  |              |              |             |
| 5 | 0.3979434960 | 0.4972044680 | 0.4251273680 | 0.007082568  |              |             |
| 6 | 0.3510609180 | 0.2643031830 | 0.3975055890 | 0.6649804540 | 0.680073183  |             |
| 7 | 0.3158094170 | 0.2413243750 | 0.3667692140 | 0.5931089770 | 0.6059548460 | 0.005287833 |

Jeśli do klasyfikacji zbioru obiektów zastosowana zostanie metoda *k-medoids* (*pam*) [Kauffman, Rousseeuw 1990, s. 68-108; Gatnar, Walesiak i in. 2004, s. 330-332], to składnię należy uzupełnić o następujące polecenia:

```
> cl <- pam(d, 3, diss = TRUE)
> print(cl$clustering).
```

W wyniku zastosowania tej procedury otrzymuje się podział zbioru obiektów na 3 klasy:

```
> [1] 1 1 1 2 2 3 3.
```

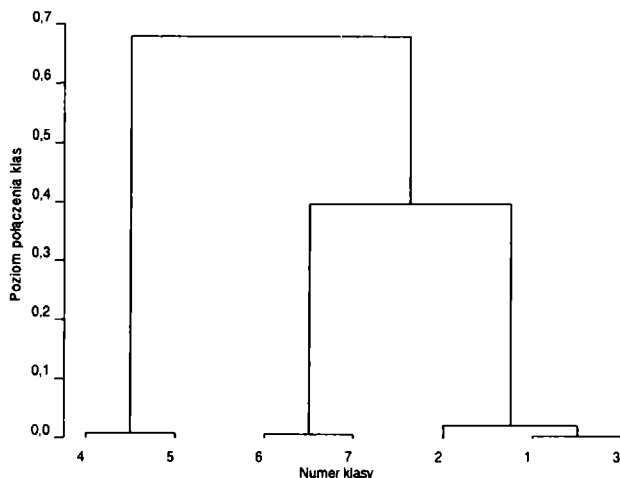
Zastosowanie innych metod klasyfikacji wymaga modyfikacji składni. Dla hierarchicznej metody aglomeracyjnej najdalszego sąsiada składnia poleceń jest następująca:

```
> hc <- hclust(d, method = "complete")
> cl <- cutree(hc, 3)
> print(cl).
```

Inne hierarchiczne metody aglomeracyjne dostępne są w programie *hclust* (pakiet *stats*) i mają następującą składnię: metoda najbliższego sąsiada (*method = "single"*), metoda średniej grupowej (*method = "average"*), metoda ważonej średniej grupowej (*method = "mcquitty"*), metoda Warda (*method = "Ward"*), metoda medianowa (*method = "median"*), metoda centroidalna (*method = "centroid"*).

Dla hierarchicznych metod aglomeracyjnych sporządza się dendrogram (zob. rys. 1), dodając polecenia:

```
> options(OutDec = ",")
> plot(hc, hang = -1, labels = NULL, main = NULL, sub = NULL,
      ann = FALSE)
> title(xlab = "Numer klasy", ylab = "Poziom połączenia
klas").
```



Rys. 1. Dendrogram  
Źródło: opracowanie własne.

Jeśli dodatkowo wprowadzimy polecenia:

```
> options(OutDec = ", ")
> desc <- cluster.Description(x, cl$cluster, "population")
> print(desc),
```

to osobno dla każdej klasy i zmiennej z ustalonego podziału zbioru obiektów na trzy klasy obliczone zostaną następujące statystyki opisowe: średnia arytmetyczna (1), odchylenie standardowe (2), mediana (3), medianowe odchylenie bezwzględne (4), dominanta (5).

Dla danego podziału zbioru obiektów na klasy możliwe jest obliczenie wybranych statystyk opisowych. Zmienić należy polecenie z ostatniej linii:

```
> print(desc[, , 3]).
```

Pierwszy wymiar w poleceniu `desc[, , 3]` oznacza numer klasy, drugi wymiar – numer zmiennej, a trzeci wymiar – numer statystyki opisowej. Znak „,” niepoprzedzony liczbą oznacza, że dotyczy wszystkich rozpatrywanych wariantów. Zatem polecenie `desc[, , 3]` oznacza, że dla wszystkich klas i zmiennych obliczone zostaną mediany.

Po zastosowaniu tej procedury otrzymuje się:

```
>      [,1] [,2]
> [1,]  2,50 4,00
> [2,]  9,50 1,75
> [3,]  4,25 10,50.
```

Zdecydowana większość funkcji programu R zwraca złożone obiekty lub listy zawierające wiele informacji wygenerowanych przez wykonywany algorytm. Na

przykład obiekt „hc” opisujący wyniki działania dowolnej hierarchicznej metody aglomeracyjnej wywołany poleceniem `names(hc)` zawiera:

- `merge`                    - numery łączonych klas w klasyfikacji hierarchicznej (wartości ujemne oznaczają dołączenie klasy jednoelementowej, a dodatnie – dołączenie klasy co najmniej dwuelementowej),
- `height`                   - wartości malejące poziomów połączenia klas,
- `order`                    - wektor zawierający numery obiektów w kolejności występującej w dendrogramie,
- `labels`                   - etykiety nadane klasyfikowanym obiektom,
- `method`                   - nazwa zastosowanej hierarchicznej metody aglomeracyjnej,
- `call`                    - składnia zastosowanego polecenia `hclust`,
- `dist.method`           - nazwa miary odległości (występuje, jeśli wykorzystano w składni polecenie `dist` z atrybutem „`method`”).

Wydanie polecenia np. `print(hc$height)` spowoduje wypisanie poziomów połączenia klas w klasyfikacji hierarchicznej. Zapisanie poziomów połączenia klas w osobnym pliku na dysku, zgodnie z formatem `csv`, wymaga dodania do składni polecenia:

```
> write.table(hc$height, file="C:/hc_height.csv",  
  sep=";", dec=".", col.names=FALSE).
```

Gdy jesteśmy zainteresowani przeprowadzeniem podziału metodą *k-medoids* zbioru siedmiu obiektów na dwie do sześciu klas, a następnie wybraniu tego podziału, dla którego indeks Calińskiego i Harabasa (zob. [Gatnar, Walesiak i in. 2004, s. 338]) *G1* przyjmuje wartość maksymalną, to pierwotną składnię należy uzupełnić o polecenia:

```
> min_liczba_klas=2  
> max_liczba_klas=6  
> max<- -1  
> wyniki<-array(0,c(max_liczba_klas-min_liczba_klas+1,  
  2))  
> wyniki[,1]<- min_liczba_klas:max_liczba_klas  
for (liczba_klas in min_liczba_klas:max_liczba_klas)  
> {  
>   cl2 <- pam(d, liczba_klas, diss = TRUE)  
>   wyniki[liczba_klas - min_liczba_klas+1,2]<- G1 <- index.G1(z, cl2$cluster)  
>   if (max<G1){  
>     max<- G1  
>     clmax<- cl2$cluster  
>     lk<- liczba_klas  
>   }
```

```

> }
> print(paste("max G1 dla", lk, "klas =", max))
> print("klasyfikacja dla max G1")
> print(clmax)
> write.table(wyniki, file="G1_results.csv", sep=";",
dec=".", row.names=TRUE, col.names=FALSE)
> plot(wyniki, type="p", xlab="Liczba klas", ylab="G1",
xaxt="n")
> axis(1, c(min_liczba_klas:max_liczba_klas)).

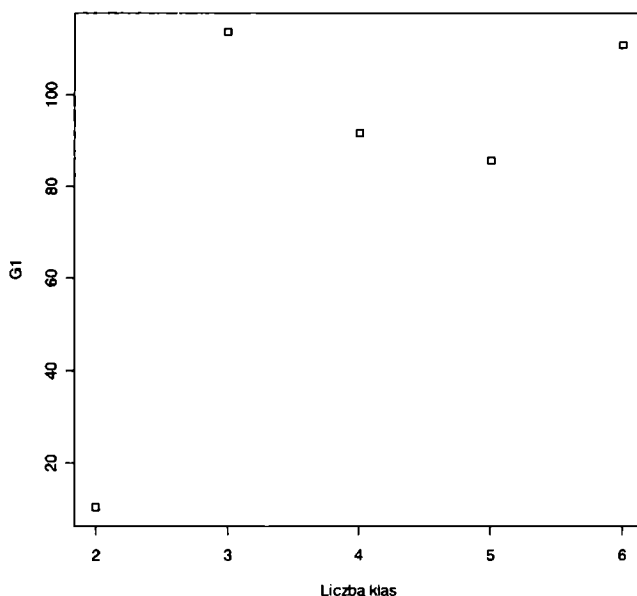
```

W wyniku zastosowania tej procedury otrzymuje się:

```

[1] "max G1 dla 3 klas = 113,675361661312"
[1] "klasyfikacja dla max G1"
[1] 1 1 1 2 2 3 3

```



Rys. 2. Graficzna prezentacja wartości indeksu G1

Źródło: opracowanie własne.

Dodanie do składni polecenia:

```

> opis <- cluster.Description(x, clmax, "population")
> print(opis)

```

pozwoli uzupełnić wyniki klasyfikacji o statystyki opisowe obliczone osobno dla każdej klasy i zmiennej podziału zbioru obiektów, dla którego indeks G1 przyjmuje wartość maksymalną.

## 4. Podsumowanie

Funkcjonujące na rynku pakiety statystyczne nie udostępniają takich możliwości, jakie niesie z sobą pakiet `clusterSim` oraz środowisko programistyczne R. W artykule scharakteryzowano wybrane funkcje pomocnicze pakietu `clusterSim`. Następnie zaprezentowano przykładowe składnie poleceń (procedury) z wykorzystaniem m.in. wybranych funkcji z pakietu `clusterSim`, w tym w szczególności `dist.GDM`. Procedury te pozwalają na realizację wielu zagadnień klasyfikacyjnych niedostępnych w podstawowych pakietach statystycznych.

## Literatura

- Caliński R.B., Harabasz J. (1974), *A Dendrite Method for Cluster Analysis*, „Communications in Statistics” vol. 3, s. 1-27.
- Gatnar E., Walesiak M. (red.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych*, AE, Wrocław.
- Hartigan J. (1975), *Clustering Algorithms*, Wiley, New York.
- Hubert L. (1974), *Approximate Evaluation Technique for the Single-Link and Complete-Link Hierarchical Clustering Procedures*, „Journal of the American Statistical Association” vol. 69, no. 347, s. 698-704.
- Kaufman L., Rousseeuw P.J. (1990), *Finding Groups in Data: an Introduction to Cluster Analysis*, Wiley, New York.
- Milligan G.W., Cooper M.C. (1985), *An Examination of Procedures of Determining the Number of Cluster in a Data Set*, „Psychometrika” vol. 50, no. 2, s. 159-179.
- R Development Core Team (2006), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, URL <http://www.R-project.org>.
- Tibshirani R., Walther G., Hastie T. (2001), *Estimating the Number of Clusters in a Data Set via the Gap Statistic*, „Journal of the Royal Statistical Society” ser. B, vol. 63, part 2, s. 411-423.
- Walesiak M. (2006), *Uogólniona miara odległości w statystycznej analizie wielowymiarowej*, wydanie drugie rozszerzone, AE, Wrocław.
- Walesiak M., Dudek A. (2006a), *Symulacyjna optymalizacja wyboru procedury klasyfikacyjnej dla danego typu danych – oprogramowanie komputerowe i wyniki badań*, [w:] K. Jajuga, M. Walesiak (red.), *Klasyfikacja i analiza danych – teoria i zastosowania, Taksonomia 13*, Prace Naukowe AE we Wrocławiu nr 1126, AE, Wrocław, s. 120-129.

Walesiak M., Dudek A. (2006b), *Determination of Optimal Clustering Procedure for a Data Set*, 30<sup>th</sup> Annual Conference of the German Classification Society (GfKI) „Advances in Data Analysis”, Berlin, March 8-10.

Walesiak M., Dudek A. (2007), *Symulacyjna optymalizacja wyboru procedury klasyfikacyjnej dla danego typu danych – charakterystyka problemu*, Zeszyty Naukowe Uniwersytetu Szczecińskiego (w druku).

## **THE SELECTED PROBLEMS IN CLUSTER ANALYSIS WITH APPLICATION OF CLUSTER~~Sim~~ COMPUTER PROGRAM AND R ENVIRONMENT**

### **Summary**

Package clusterSim has been written in R language. It contains one main function cluster.Sim and sixteen auxiliary functions. The article presents selected auxiliary functions of clusterSim program and examples of the syntax (procedures) for solving different clustering problems using among others clusterSim package including especially dist.GDM function. These procedures help to resolve a broad range of classification problems that are not available in statistical packages (e.g. SPSS, Statistica, S-Plus, SAS).