Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Artur Kaczyński, Adam Kupiński
Nr albumu: 305150, 305208

Pakiet ClustOfVar i jego zastosowania w analizie danych medycznych

Praca licencjacka
na kierunku MATEMATYKA

Praca wykonana pod kierunkiem
dra Błażeja Miasojedowa
Instytut Matematyki Stosowanej i Mechaniki UW

Czerwiec 2013

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

Klastrowanie zmiennych polega na szukaniu powiązań wśród pewnych cech i podzieleniu ich na grupy, w których każda cecha niesie podobną informację. W niniejszej pracy omówimy sposób klastrowania zmiennych w pakiecie ClustOfVar programu R. W tym celu przedstawimy wprowadzenie teoretyczne, które precyzuje sposób wyboru podziału i omówimy wybrane funkcje pakietu. Następnie pokażemy przykład jego zastosowania na rzeczywistych danych medycznych.

Słowa kluczowe

klastrowanie zmiennych, klaster, zmienna centralna klastra, jednorodność klastra, jednorodność podziału, hclustvar, cutreevar, kmeansvar

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.2 Statystyka

Klasyfikacja tematyczna

62H30 Classification and discrimination; cluster analysis

Tytuł pracy w języku angielskim

ClustOfVar package and its application in the analysis of medical data

Udział w przygotowaniu pracy

Artur Kaczyński jest autorem:

- 1.1 Jednorodność podziału,
- 1.3 Stabilność,
- 2.3 Funkcja kmeansvar,
- 2.4 Funkcja stability,
- dodatek A.

Adam Kupiński jest autorem:

- 1.2.1 Grupowanie hierarchiczne,
- 2.1 Funkcja hclustvar,
- 2.2 Funkcja cutreevar,
- 3 Analiza danych,
- dodatek B.

Pozostałe części są napisane wspólnie.

Spis treści

Spis rysunków

Spis tabel

Rozdział 1

Teoria

1.1. Jednorodność podziału

Klastrowanie zmiennych polega na podziale pewnych zmiennych (cech) na rozłączne zbiory nazywane klastrami w ten sposób, aby zmienne znajdujące się w danym klastrze były jak najbardziej do siebie podobne. Intuicyjnie zmienne znajdujące się w tym samym klastrze powinny nieść tę samą informację. Wtedy każda z nich daje prawie tyle samo wiadomości, co wszystkie razem.

Na początku zdefiniujemy pojęcie partycji zmiennych na klastry. Następnie sformalizujemy podobieństwo zmiennych w klastrze, na którym opierają się algorytmy użyte w pakiecie ClustOfVar, czyli wyjaśnimy, w jaki sposób dokonuje się podziału zmiennych na klastry i na podstawie jakich kryteriów podejmuje się decyzję o wyborze jednej z wielu możliwych partycji. Zanim jednak zdefiniujemy jednorodność podziału, wprowadzimy kilka oznaczeń i definicji.

Definicja 1.1.1 *(Partycja)*
Partycją zmiennych $z_1, z_2, ..., z_k$ nazywamy każdy ich podział na niepuste i rozłączne klastry.

Partycję będziemy oznaczać literą P, z dodanym indeksem dolnym, który ma oznaczać liczbę klastrów, z których składa się partycja. Ponieważ będzie wiadomo, jakie zmienne mamy na myśli, nie będziemy pisać zależności partycji od tych zmiennych, tzn. podział zmiennych $z_1, z_2, ..., z_k$ na l klastrów oznaczamy przez P_l.

Niech $\{x_1, ..., x_{p_1}\}$ będzie zbiorem p_1 zmiennych ilościowych o wartościach liczbowych, a $\{y_1, ..., y_{p_2}\}$ będzie zbiorem p_2 zmiennych jakościowych. Dla zmiennej jakościowej y_j niech M_j oznacza $1 \times m_j$ wymiarowy wektor jej kategorii, gdzie m_j jest ilością kategorii y_j. Na przykład określmy wektor y, który opisuje kolor oczu osób w danej grupie. Wtedy można przyjąć $M = \{$*"zielony", "niebieski", "brązowy", "szary"*$\}$.

Dla dwóch zmiennych ilościowych u i v o rozmiarze n oznaczmy przez

$$r^2(u,v) := \frac{\left(\sum_{i=1}^{n}(u_i - \overline{u})(v_i - \overline{v}\right)^2}{\sum_{i=1}^{n}(u_i - \overline{u})^2 \sum_{i=1}^{n}(v_i - \overline{v})^2} \tag{1.1}$$

kwadrat ich korelacji.

Dla zmiennej ilościowej x o rozmiarze n i zmiennej jakościowej y o rozmiarze n określmy miarę podobieństwa

$$\eta^2(x;y) := \frac{\sum\limits_{s\in M} n_s(\overline{x_s} - \overline{x})^2}{\sum\limits_{i=1}^{n}(u_i - \overline{u})^2}, \tag{1.2}$$

gdzie M jest zbiorem kategorii y, n_s oznacza ilość wystąpień kategorii s w y, a $\overline{x_s}$ jest średnią x policzoną na obserwacjach należących do s. Miara η^2 przyjmuje wartości w zbiorze $I = [0, 1]$.

1.1.1. Centralna zmienna klastra

Definicje będące głównymi pojęciami trzech następnych podrozdziałów pochodzą z [1].

Definicja 1.1.2 *(Centralna zmienna klastra)*
Centralną zmienną klastra C_k nazywamy wektor $c_k \in R^n$ spełniający warunek:

$$c_k = arg\max_{u\in R^n}\{\sum_{x_i\in C_k} r^2(u, x_i) + \sum_{y_i\in C_k}\eta^2(u; y_i)\}.$$

Centralna zmienna klastra wraz ze zdefiniowanymi przez wzory (1.1) i (1.2) miarami podobieństwa są podstawowymi pojęciami wykorzystywanymi w algorytmach grupowania zmiennych w pakiecie ClustOfVar.

Przedstawimy teraz sposób znajdowania zmiennej centralnej, jaki został zaproponowany w [1] i [3] oraz udowodnimy jego poprawność. Opiera się on na pewnym przekodowaniu zmiennych, zapisaniu ich w jednej macierzy i zastosowaniu wobec niej rozkładu SVD (singular value decomposition).

Określmy dla zmiennej jakościowej x_j i zmiennej ilościowej y_j:

X_i wektor wymiaru $n \times 1$ uzyskany z wektora x_i przez odjęcie $\overline{x_i} := \frac{1}{n}\sum\limits_{j=1}^{n} x_{i,j}$ od każdej współrzędnej i podzielenie wektora przez $\rho_{x_i} := \sqrt{\sum\limits_{j=1}^{n}(x_{i,j} - \overline{x_i})^2}$.

G_j macierz charakterystyczna zmiennej y_j o wymiarze $n \times m_j$, gdzie m_j jest ilością kategorii y_j. Każda kolumna G_j odpowiada za jedną kategorię y_j i ma wartość 1 na $i - tej$ pozycji jeśli $i - ta$ współrzędna y_j odpowiada tej samej kategorii oraz 0 w przeciwnym przypadku, tzn. $G_j M_j = y_j$.

D_j macierz diagonalna wymiaru $m_j \times m_j$, gdzie d_{ii} jest ilością wystąpień i - tej kategorii w y_j, $\sum\limits_{i=1}^{m_j} d_{ii} = n$.

W pracy będziemy stosować oznaczenie $(X|Y)$, które dla macierzy lub wektorów kolumnowych X i Y o tej samej liczbie wierszy ma oznaczać macierz powstałą przez dopisanie do X, z jej prawej strony, macierzy Y.

Dla klastra C składającego się z $\{x_1, x_2, ..., x_{r_1}\}$ zmiennych jakościowych i $\{y_1, y_2, ..., y_{r_2}\}$ zmiennych jakościowych określmy (używając macierzy opisanych powyżej):

J $= Id_n - \frac{1}{n}\cdot\mathbb{1}_n$, gdzie Id_n jest macierzą identyczności, a $\mathbb{1}_n$ kwadratową macierzą jedynek, obie wymiaru $n \times n$

X $= (X_1|...|X_j|...|X_{r_1})$ - macierz wymiaru $n \times r_1$

G $= (G_1|...|G_j|...|G_{r_2})$ - macierz wymiaru $n \times m$, gdzie $m = m_1 + ... + m_{r_2}$ oraz m_j jest ilością kategorii y_j

D $= diag(D_1, ..., D_{r_2})$ - macierz wymiaru $m \times m$

Y $= JGD^{-\frac{1}{2}}$

Powołując się na Twierdzenie 3.2. z [4] mówimy, że dla każdej macierzy $B \in R^{n\times p}$ istnieją macierze $U \in R^{n\times n}$, $V \in R^{p\times p}$ oraz $\Lambda \in R^{n\times p}$ t. że: $B = U\Lambda V^T$, U i V są ortonormalne oraz Λ jest diagonalna z uporządkowanymi nierosnąco wyrazami na diagonali.

Aby obliczyć centralną zmienną klastra tworzymy $Z = \frac{1}{\sqrt{n}}(X|Y)$ oraz przeprowadzamy jej rozkład SVD:

$$Z = U\Lambda V^T. \tag{1.3}$$

Stwierdzenie 1.1.1 *Dla opisanego powyżej klastra jego centralna zmienna* $c = \sqrt{n}U^{(1)}\lambda_1$, *gdzie* $U^{(1)}$ *jest pierwszą kolumną* U, *a* λ_1 *pierwszym wyrazem* Λ *z równania (1.3) .*

Ze względu na techniczny charakter i dużą objętość dowodu umieściliśmy go w Dodatku A.

1.1.2. Jednorodność klastra

Mając teraz określoną centralną zmienną klastra, możemy przystąpić do opisania jego jednorodności.

Definicja 1.1.3 *(Jednorodność klastra)*
Jednorodność klastra $C_k = \{x_1, ..., x_{r_{k_1}}, y_1, ..., y_{r_{k_2}}\}$ *opisujemy jako*

$$h(C_k) = \sum_{x_j \in C_k} r^2(x_j, c_k) + \sum_{y_j \in C_k} \eta^2(c_k; y_j),$$

gdzie c_k *jest centralną zmienną klastra* C_k.

Jednorodność klastra to suma miar podobieństwa r^2 i η^2 zmiennych należących do klastra ze zmienną centralną. $h(C_k)$ przyjmuje wartości w przedziale $[1, r_{k_1} + r_{k_2}]$, przy czym równość $h(C_k) = r_{k_1} + r_{k_2}$, zachodzi gdy wszystkie zmienne ilościowe są całkowicie skorelowane ze zmienną centralną oraz dla każdej zmiennej jakościowej y_j miara podobieństwa $\eta^2(c_k; y_j) = 1$. Wtedy wszystkie zmienne w klastrze niosą tę samą informację. Intuicyjnie im bardziej stosunek $\frac{h(C_k)}{r_{k_1}+r_{k_2}}$ jest bliższy 1, tym klaster jest lepszy z punktu widzenia poszukiwań podziału na grupy niosące tę samą informację.

Na podstawie Stwierdzenia 1.1.1 możemy zapisać:

Stwierdzenie 1.1.2 *Jednorodność klastra* $h(C_k) = \lambda_1^2$, *gdzie* λ_1 *pierwszym wyrazem* Λ *ze wzoru (1.3).*

Dowód Stwierdzenia 1.1.2 również znajduje się w Dodatku A.

1.1.3. Jednorodność partycji

Definicja 1.1.4 *(Jednorodność partycji)*
Jednorodność partycji $P_l = C_1 \cup ... \cup C_l$ *opisujemy jako*

$$H(P_l) = \sum_{i=1}^{l} h(C_i).$$

Jest to suma jednorodności klastrów składających się na tę partycję. Z jednej strony jednorodność partycji zmiennych na klastry jest maksymalna dla podziału na singletony i równa się wtedy liczbie zmiennych, gdyż $H(C_k) = 1$ dla $C_k = \{x_k\}$ lub $C_k = \{y_k\}$. Jednak z perspektywy poszukiwania grup zmiennych niosących tę samą informację, podział na singletony jest bezużyteczny.

Pakiet ClustOfVar dostarcza dwóch metod grupowania zmiennych w użyteczny sposób. Pierwsza z nich polega na hierarchicznym budowaniu partycji na kolejno $k = p_1 + p_2 - 1$, $p_1 + p_2 - 2, ..., 2$ klastrów, gdzie p_1 i p_2 są liczbą zmiennych ilościowych i jakościowych odpowiednio, w ten sposób, aby różnica $H(P_k) - H(P_{k-1})$, $k = p_1 + p_2, p_1 + p_2 - 1, ..., 3$ była jak najmniejsza. To rozwiązanie zostało zaimplementowane w funkcji `hclustvar`. Inną możliwością jest ustalenie K - porządanej liczby klastrów, a następnie próba znalezienia partycji P_{K_0} takiej, że

$$H(P_{K_0}) = \max\{H(P_k) : \ k = K\}.$$

Ten sposób został użyty w funkcji `kmeansvar`. Oba powyższe algorytmy zostaną dokładnie omówione w kolejnej części pracy.

1.2. Algorytmy podziału

Celem algorytmu podziału jest znalezienie partycji zmiennych ilościowych i jakościowych, tak aby zmienne w każdym klastrze były silnie powiązane ze zmiennymi znajdującymi się w tym samym klastrze. Innymi słowy naszym punktem docelowym jest znalezienie partycji P_K, która maksymalizuje funkcję jednorodności H (def. 1.1.4). W tym celu w pakiecie **ClustOfVar** zostały zaproponowane dwa algorytmy: algorytm grupowania hierarchicznego i algorytm relokacyjny.

1.2.1. Grupowanie hierarchiczne

Metoda grupowania hierarchicznego polega na sekwencyjnym grupowaniu obiektów, łączeniu klastrów w coraz to większe. Proces łączenia realizowany jest na zasadzie poszukiwania klastrów leżących najbliżej siebie w sensie zdefiniowanej poniżej odległości i zastępowaniu ich nowym większym klastrem, będącym scaleniem dwóch poprzednich. Proces ten stopniowo postępuje aż do chwili, w której zostanie osiągnięta właściwa liczba klastrów (określona przez użytkownika) lub do momentu gdy wszystkie obiekty znajdują się w jednym klastrze.

Definicja 1.2.1 *(Odległość między klastrami)*
Odległością między klastrami C_1 *i* C_2 *nazywamy liczbę* d *daną przez wzór:*

$$d(C_1, C_2) = h(C_1) + h(C_2) - h(C_1 \cup C_2),$$

gdzie $h(C)$ *jest jednorodnośćią klastra (def. 1.1.3).*

- **Dane wejściowe**: baza danych p obiektów.

- **Dane wyjściowe**: drzewo klastrów (tzw. dendrogram) reprezentujący grupowanie obiektów.

1. Krok i = 0: inicjalizacja. Startujemy z partycją złożoną z p klastrów (Każdy obiekt jest w osobnym klastrze).

2. Krok i = 1,...,$p-2$: łączymy dwa klastry z partycji złożonej z $p-i+1$ klastrów, aby uzyskać partycję o $p-i$ klastrach. W tym celu wybieramy klastry A i B z najmniejszą odległością d (def. 1.2.1).

3. Krok i = $p-1$: stop. Otrzymaliśmy partycję złożoną z jednego klastra.

Odległość d mierzy utratę jednorodności obserwowaną, gdy dwa klastry C_1 i C_2 są ze sobą scalane. Ten sposób grupowania zmiennych tworzy nową partycję złożoną z $p-i$ klastrów, która maksymalizuje H (def.1.1.4) spośród wszystkich partycji złożonych z $p-i$ klastrów uzyskanych przez połączenie dwóch klastrów partycji zawierającej $p-i+1$ klastrów. Rzeczywiście, biorąc np. $p-i=2$ i przykładową partycję P_3 złożoną z klastrów C_{k_1}, C_{k_2} i C_{k_3} oraz P_2 złożoną z klastrów $C_{j_1}=C_{k_1}$ i $C_{j_2}=C_{k_2}\cup C_{k_3}$ otrzymujemy:

$$H(P_3)-H(P_2)=\sum_{n=1}^{3}h(C_{k_n})-\sum_{n=1}^{2}h(C_{j_n})=h(C_{k_2})+h(C_{k_3})-h(C_{k_2}\cup C_{k_3})=d(C_{k_2},C_{k_3}).$$

Zatem jeśli C_{k_2} i C_{k_3} są parą klastrów o najmniejszej odległości d spośród wszystkich par z P_3, to $H(P_3)-H(P_2)$ jest najmniejsze, a $H(P_2)$ jest maksymalne. Ten algorytm realizowany jest przez funkcję `hclustvar`, która tworzy hierarchię p zmiennych. Funkcja `plot.hclustvar` zwraca dendrogram dla tej hierarchii. Na końcu funkcja `cutreevar` przycina nasz dendrogram do pożądanej przez użytkownika ilości klastrów i zwraca odpowiednią partycję. Szczegółowy opis funkcji `hclustvar` i `cutreevar` znajduje się w rozdziale 2.

1.2.2. Algorytm relokacyjny

Przed opisem algorytmu relokacyjnego należy zdefiniować miarę podobieństwa dla dwóch zmiennych jakościowych. Dla zmiennej jakościowej y_i niech $Y_i=JG_iD_i^{-\frac{1}{2}}$ (oznaczenia macierzy po prawej stronie tego równania zostały wprowadzone w podrozdziale 1.1.1). Y_i jest wymiaru $n\times m_j$, gdzie m_j jest liczbą kategorii. Niech $\varrho(Y_i,Y_j)$ będzie pierwszą wartością własną macierzy $R(Y_i,Y_j)$, gdzie

$$R(Y_i,Y_j)=\begin{cases} Y_iY_j^TY_jY_i^T & \text{gdy } min(n,m_i,m_j)=n \\ Y_i^TY_jY_j^TY_i & \text{gdy } min(n,m_i,m_j)=m_i \\ Y_j^TY_iY_i^TY_j & \text{gdy } min(n,m_i,m_j)=m_j. \end{cases}$$

$\varrho(Y_i,Y_j)$ możemy interpretować jako miarę podobieństwa zmiennych y_i i y_j. W szczególności, jak podaje [1], jeśli dodatkowo dla zmiennej ilościowej x_k oznaczymy przez X_k jej znormalizowaną wersję (jak w podrozdziale 1.1.1) i przyjmiemy $m_k=1$, otrzymamy:
$\varrho(X_i,X_j)=r^2(x_i,x_j)$ oraz $\varrho(Y_i,X_j)=\eta^2(x_j;y_j)$, gdzie r i η^2 są zdefiniowane przez wzory (1.1) i (1.2). Aby się o tym przekonać wystarczy uważnie prześledzić Krok 2. i Krok 3. z dowodu Stwierdzenia 1.1.1 umieszczonego w Dodatku A.

Algorytm relokacyjny jest wdrożony przy pomocy funkcji `kmeansvar` i buduje partycję K klastrów w następujący sposób:

1. Inicjalizacja: dostępne są dwie możliwości.

   (a) Początkowy podział na K klastrów jest podany jako argument funkcji `kmeansvar`.

   (b) Inicjalizacja losowa:

      i. Z grupy wszystkich zmiennych wybierane w losowy sposób jest K zmiennych jako początkowe zmienne centralne (def. 1.1.2).

      ii. Początkowy podział na K klastrów jest zbudowany tak, że pozostałe zmienne są dołączone do klastrów z najbliższą (w sensie zdefiniowanej powyżej miary) zmienną centralną.

2. Iteracja:

   (a) Dla każdego klastra C_k wyznaczana jest jego zmienna centralna c_k (def. 1.1.2).

   (b) Każda zmienna jest relokowana do klastra o najbliżej (w sensie zdefiniowanej powyżej miary) zmiennej centralnej.

3. Zakończenie:

   Krok 2 jest przerywany, jeśli po wykonaniu cyklu nie nastąpiła żadna zmiana, lub gdy przekroczono maksymalną ilość iteracji (ustaloną na wstępie przez użytkownika).

Powyższe iteracje funkcji `kmeansvar` próbują dostarczyć partycję P_K złożoną z K klastrów, która maksymalizuje H (def.1.1.4), ale rozwiązanie może być optymalne tylko lokalnie i może zależeć od wyboru początkowej partycji. Aby przezwyciężyć ten problem i uniknąć wpływu wyboru początkowej partycji, rozważa się wiele losowych uruchomień. W tym przypadku kroki 1(b), 2 i 3 są powtarzane i w efekcie końcowym wybieramy tę partycję, która daje największą wartość H.

1.3. Stabilność

W wyborze ilości klastrów, z których ma składać się docelowy podział, pomocna jest stabilność partycji. Załóżmy, że mamy p zmiennych $z_1, ..., z_p$, każda o wymiarze $n \times 1$. Zbiór tych zmiennych możemy rozpatrywać jako macierz $Z = (z_1|...|z_p)$ wymiaru $n \times p$. Głównym celem jest uzyskanie partycji tych zmiennych, ale należy również podjąć decyzję o ilości klastrów. Ponadto chcemy, żeby partycja nie zależała od małych zmian w danych, tzn. aby małe zaburzenia danych nie powodowały, że zmienne zostaną podzielone w inny sposób. Dzięki temu zmienne w każdym klastrze faktycznie będą ze sobą silnie powiązane. Przy liczeniu stabilności używa się tzw. skorygowanego indeksu Rand. Jest to funkcja dwóch partycji, która przyjmuje wartości liczbowe mniejsze od 1. Im skorygowany indeks Rand dwóch partycji jest większy, tym bardziej są one do siebie podobne. Przypuśćmy, że mamy dwie różne partycje Z na k klastrów: $P_k = A_1 \cup ... \cup A_k$ i $P_k^{'} = B_1 \cup ... \cup B_k$. Niech n_{ij} będzie liczbą zmiennych, które należą zarówno do A_i i do B_j. Niech $n_{i.} = \sum_{j=1}^{k} n_{ij}$ (ilość zmiennych w A_i), $n_{.j} = \sum_{i=1}^{k} n_{ij}$ (ilość zmiennych w B_j). Skorygowany indeks Rand partycji P_k i $P_k^{'}$ jest obliczany ze wzoru:

$$\frac{\sum_{i,j} \binom{n_{ij}}{2} - N}{\frac{1}{2}\left(\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right) - N},$$

gdzie $N = \frac{\sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2}}{\binom{n}{2}}$.

Bardziej szczegółowe informacje o skorygowanym indeksie Rand można znaleźć w [2].

Do obliczenia stabilności proponowany jest następujący algorytm:

1. Dendrogram zwrócony przez funkcję `hclustvar` stanowi zbiór początkowych partycji P_i, $i = 1, ..., p$.

2. Tworzone jest N (ustalona wcześniej liczba) nowych zestawów danych przez bootstrapowy wybór n wierszy z macierzy Z. Następnie dla każdego z nowo uzyskanych zestawów oblicza się za pomocą funkcji `hclustvar` dendogram, otrzymując w ten sposób partycje $P_i^{(j)}$ (podział j-tego zestawu na i klastrów), gdzie $i = 1, ..., p$, $j = 1, ..., N$.

3. Zbiór początkowych partycji jest porównywany z partycjami uzyskanymi w kroku 2. w następujący sposób: dla wszystkich par i, j liczone są skorygowane indeksy Rand partycji P_i i $P_i^{(j)}$.

4. Dla $i = 1, ..., p$ stabilność partycji P_i jest obliczana jako średnia skorygowanych indeksów Rand P_i i $P_i^{(j)}$ $(j = 1, ..., N)$.

Ta procedura pomaga dokonać wyboru ilości klastrów, która czyni partycję mniej podatną na zmiany danych. Powyższy algorytm jest zaimplementowany w funkcji `stability` w pakiecie ClustOfVar.

Rozdział 2

Funkcje pakietu ClustOfVar

2.1. Funkcja `hclustvar`

Funkcja ta dokonuje grupowania hierarchicznego zmiennych. Grupowanie odbywa się na podstawie kryterium agregacji, tzn. łączy ze sobą dwa klastry, które po połączeniu mają najmniejszą wartość `height`. Wartość `height` klastra $C = A \cup B$ wyraża się wzorem g(C)=d(A,B) (def. 1.2.1) i mówi nam jak bardzo nasz podział odbiega od poprzedniego stanu, tzn. jak bardzo informacja niesiona przez partycję P_{K-1} różni się od przekazu partycji P_K. Funkcja `hclustvar` brakujące wartości zastępuje przez średnie w przypadku zmiennych ilościowych i zera w macierzy wskaźników dla zmiennych jakościowych.

Wywołanie

`hclustvar`(X.quanti = $NULL$, X.quali =$NULL$)

Argumenty

`X.quanti`	macierz danych ilościowych.
`X.quali`	macierz danych jakościowych.

Wartości

`height`	zestaw $p-1$ niemalejących wartości rzeczywistych. Są to odległości (def. 1.2.1) pomiędzy łączonymi w danym kroku klastrami.
`clusmat`	macierz rozmiaru p×p z członkostwem grupy, gdzie każda kolumna k opowiada partycji P_K, tzn. pokazuje do którego klastra należy każda zmienna przy podziale na K klastrów.
`merge`	macierz rozmiaru $(p-1)$ ×2, gdzie i-ty wiersz pokazuje które dwa klastry zostały połączone w i-tym kroku. Jeśli element j macierzy jest ujemny to znaczy, że nie został on wcześniej połączony z innym klastrem i zmienna j została połączona dopiero na tym etapie. Jeśli element j jest dodatni to zmienna j została połączona na wcześniejszym etapie. Tak więc ujemne wpisy oznaczają łączenie singletonów (tzn. pojedynczych zmiennych), a dodatnie oznaczają łączenie klastra w którym znajdują się co najmniej dwie zmienne.

2.2. Funkcja `cutreevar`

Funkcja ta przycina otrzymane przy pomocy funkcji `hlustvar` drzewo klastrów (dendrogram) do określonego przez użytkownika rozmiaru.

Wywołanie

`cutreevar`(obj, k = $NULL$, matsim = $FALSE$)

Argumenty

`obj`	obiekt klasy `"hclustvar"`.
`k`	liczba całkowita oznaczająca z ilu klastrów będzie składał się podział.
`matsim`	obiekt typu logicznego, jeśli "$TRUE$" to funkcja oblicza macierz podobieństwa pomiędzy zmiennymi w tym samym klastrze. Podobieństwo jest obliczane na podstawie wzorów (1.1), (1.2).

Wartości

`var`	lista macierzy obciążenia kwadratowego. Macierzy tych jest tyle ile klastrów i obciążenie to jest podobieństwem pomiędzy zmienną i centralną zmienną klastra w którym ta zmienna się znajduje. Dla zmiennych ilościowych jest to po prostu korelacja (wzór (1.1)), a dla zmiennych jakościowych opisuje to podobieństwo wzór (1.2).
`sim`	lista macierzy podobieństwa pomiędzy zmiennymi w tym samym klastrze. Dla zmiennych ilościowych jest to kwadrat korelacji Pearsona (wzór (1.1)), a w przypadku jednej zmiennej ilościowej i drugiej jakościowej podobieństwo definiuje wzór (1.2). W przypadku dwóch zmiennych jakościowych podobieństwo jest wyrażone przez kwadrat korelacji kanonicznej, jest to swojego rodzaju uogólniony współczynnik korelacji Pearsona. Jeśli "matsim" jest "$FALSE$" wtedy `sim` jest "$NULL$".
`cluster`	wektor liczb całkowitych wskazujący do którego klastra należy poszczególna zmienna.
`wss`	lista jednorodności klastrów (def 1.1.3).
`E`	przyrost spójności - wyrażony jest przez wartość procentową jednorodności partycji uzyskiwanej przez partycję P_K w stosunku do partycji składającej się z jednego klastra. Zdefiniowana jest wzorem $E(P_K) = \frac{H(P_K)-H(P_1)}{H(P_p)-H(P_1)}$, gdzie funkcja jednorodności H została określona w definicji (1.1.4).
`size`	liczba zmiennych w każdym klastrze.
`scores`	macierz rozmiaru n×k, gdzie n jest ilością obserwacji, a k to ilość klastrów. W kolumnach tej macierzy znajdują się centralne zmienne otrzymanych klastrów. Centralna zmienna klastra została określona w definicji (1.1.2).

2.3. Funkcja `kmeansvar`

Funkcja ta poszukuje lokalnie optymalnej partycji zmiennych ilościowych lub jakościowych, o zadanej liczbie klastrów, wykorzystując algorytm relokacyjny.

Wywołanie

`kmeansvar`(X.quanti = $NULL$, X.quali = $NULL$, init, iter.max = 150, nstart = 1, matsim = $FALSE$)

Argumenty

`X.quanti`	macierz danych ilościowych.
`X.quali`	macierz danych jakościowych.
`init`	może być liczbą naturalną (wtedy algorytm losowo wybiera podaną jako `init` liczbę zmiennych i traktuje je jako początkowe zmienne centralne klastrów) lub wektorem liczb naturalnych, inicjujących przynależność zmiennych do klastórw.
`iter.max`	maksymalna ilość iteracji algorytmu.
`nstart`	liczba uruchomień z losowym podziałem startowym jeśli `init` był liczbą.
`matsim`	jeśli jest ustawione jako "$TRUE$", w każdym klastrze liczone są macierze podobieństwa między zmiennymi.

Wartości

`var`	lista macierzy obciążenia kwadratowego. Macierzy tych jest tyle ile klastrów i obciążenie to jest podobieństwem pomiędzy zmienną i centralną zmienną klastra w którym ta zmienna się znajduje. Dla zmiennych ilościowych jest to po prostu korelacja (wzór (1.1)), a dla zmiennych jakościowych opisuje to podobieństwo wzór (1.2).
`sim`	lista macierzy podobieństwa pomiędzy zmiennymi w tym samym klastrze. Dla zmiennych ilościowych jest to kwadrat korelacji Pearsona (wzór (1.1)), a w przypadku jednej zmiennej ilościowej i drugiej jakościowej podobieństwo definiuje wzór (1.2). W przypadku dwóch zmiennych jakościowych podobieństwo jest wyrażone przez kwadrat korelacji kanonicznej, jest to swojego rodzaju uogólniony współczynnik korelacji Pearsona. Jeśli "matsim" jest "$FALSE$" wtedy `sim` jest "$NULL$".
`cluster`	wektor liczb całkowitych wskazujący do którego klastra należy poszczególna zmienna.
`wss`	lista jednorodności klastrów (def 1.1.3).
`E`	przyrost spójności - wyrażony jest przez wartość procentową jednorodności partycji uzyskiwanej przez partycję P_K w stosunku do partycji składającej się z jednego klastra. Zdefiniowana jest wzorem $E(P_K) = \frac{H(P_K)-H(P_1)}{H(P_p)-H(P_1)}$, gdzie funkcja jednorodności H została określona w definicji (1.1.4).
`size`	liczba zmiennych w każdym klastrze.
`scores`	macierz rozmiaru n×k, gdzie n jest ilością obserwacji, a k to ilość klastrów. W kolumnach tej macierzy znajdują się centralne zmienne otrzymanych klastrów. Centralna zmienna klastra została określona w definicji (1.1.2).

2.4. Funkcja `stability`

Funkcja ta oblicza stabilność partycji otrzymanych w funkcji `hclustvar`. Dla każdej partycji z hierarchii liczy B bootstrapowych partycji i średnią ich skorygowanych indeksów Rand.

Wywołanie

`stability`(tree, B = 100, graph = $TRUE$)

Argumenty

`tree`	obiekt klasy "`hclastvar`".
`B`	ilość bootstrapowych partycji.
`graph`	jeśli jest ustawione jako "$TRUE$", zostaje wyświetlony wykres.

Wartości

`matCR`	macierz skorygowanych indeksów Rand.
`meanCR`	wektor średnich skorygowanych indeksów Rand.

Więcej informacji na temat pakietu ClustOfVar i jego funkcji można znaleźć w [5].

Rozdział 3

Analiza danych

3.1. Wprowadzenie

Zastosowanie pakietu ClustOfVar przedstawimy na rzeczywistych danych medycznych dotyczących przeszczepu nerki. Badanie parametrów nerki i osób którym zostanie ona przeszczepiona jest czasochłonne i czasami trudne do zrealizowania ze względu na ich ilość. W tym rozdziale zaproponujemy, jak możemy te właściwości pogrupować w klastry (grupy) w taki sposób, aby zmniejszyć ilość zmiennych przy zachowaniu niesionych przez nie informacji.

3.2. Opis danych

Dane pochodzą z Centralnego Szpitala Klinicznego Ministerstwa Obrony Narodowej i zostały przetworzone w Instytucie Biocybernetyki i Inżynierii Biomedycznej. Zawierają 90 obserwacji i 21 zmiennych. Każda obserwacja przedstawia charakterystykę pacjenta, któremu została przeszczepiona nerka i gromadzi dane czynników, które mogły oddziaływać na prawidłowe funkcjonowanie nerki po przeszczepie. Mamy do czynienia z danymi jakościowymi (`Sex` i `Diabetes`) i ilościowymi (pozostała część). Tabela 3.1 przedstawia krótki opis wszystkich zmiennych, a tabela 3.2 obrazuje fragment analizowanych danych.

3.3. Analiza danych

Celem naszej analizy jest znalezienie macierzy liczb rzeczywistych o możliwie najmniejszym rozmiarze, którą będziemy mogli zastąpić naszą macierz 90×21 danych ilościowych i jakościowych, tak by niesione informacje przez obydwie macierze były zbliżone. W pierwszej kolejności pokażemy zastosowanie funkcji `hclustvar`, która tworzy hierarchię naszych zmiennych i dzięki niej będziemy już w stanie co nieco powiedzieć o naszym podziale.

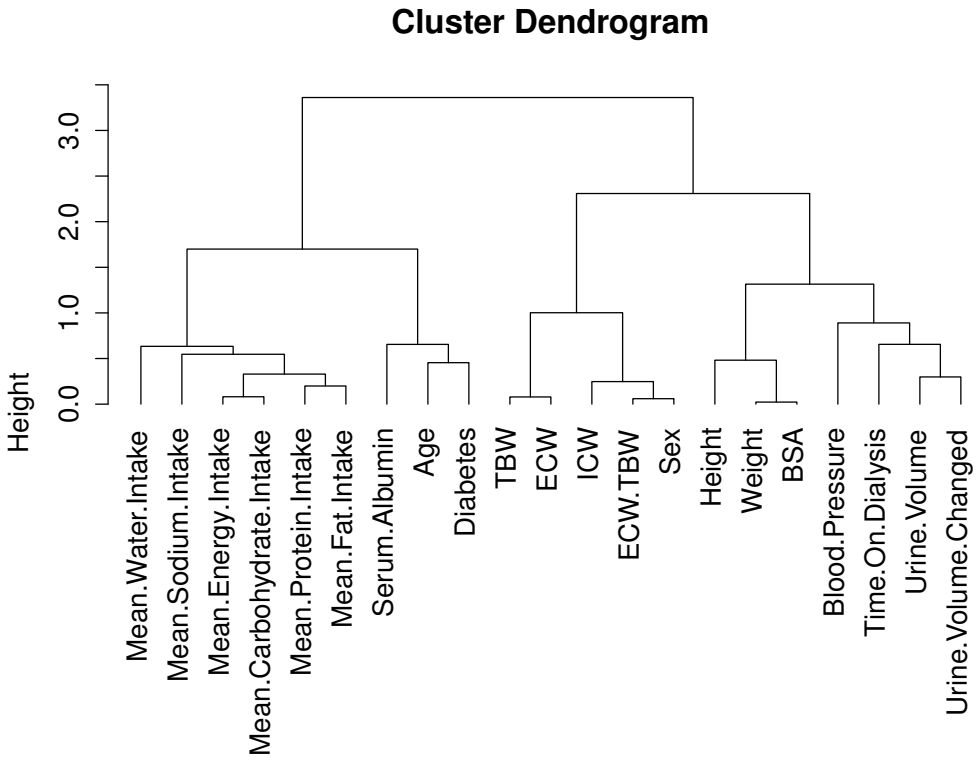
Rysunek 3.1 obrazuje zależności podobieństw między klastrami, lecz musimy pamiętać, że jest to relacja liczona w terminie r^2 i η^2 , jak to przedstawia wzór przedstawiony w definicji (1.1.3). W efekcie czego dendrogram nie pokazuje znaku tej relacji, ale możemy z niego odczytać kilka ważnych informacji m.in. zmienna jakościowa `Diabetes` jest związana ze zmienną ilościową `Age` (co do znaku korelacji), a średnio spożyte poszczególne składniki odżywcze, energetyczne tworzą razem silnie powiązaną grupę, co świadczy o istotnym związku tych zmiennych.

Lp.	Zmienna	Opis
1	Sex	Płeć
2	Time.On.Dialysis	Czas dializy
3	Age	Wiek
4	Diabetes	Informacja czy dany pacjent jest cukrzykiem
5	Weight	Waga
6	Height	Wzrost
7	BSA	Powierzchnia ciała
8	Serum.Albumin	Zawartość białka w klarownej części krwi
9	Blood.Pressure	Ciśnienie krwi
10	TBW	Suma wody wewnątrzkomórkowej i zewnątrzkomórkowej (objętość)
11	ECW	Objętość wody zewnątrzkomórkowej
12	ICW	Objętość wody wewnątrzkomórkowej
13	ECW.TBW	Stosunek objętości wody zewnątrzkomórkowej do całej objętości wody w ciele
14	Mean.Energy.Intake	Średnia dzienna dawka energi
15	Mean.Protein.Intake	Średnia dzienna dawka białka
16	Mean.Carbohydrate.Intake	Średnia dzienna dawka węglowodanów
17	Mean.Fat.Intake	Średnia dzienna dawka tłuszczu
18	Mean.Sodium.Intake	Średnia dzienna dawka sodu
19	Mean.Water.Intake	Średnia dzienna dawka wody
20	Urine.Volume	Objętość moczu
21	Urine.Volume.Changed	Ilu krotnie zwiększyła się objętość moczu po przeszczepie

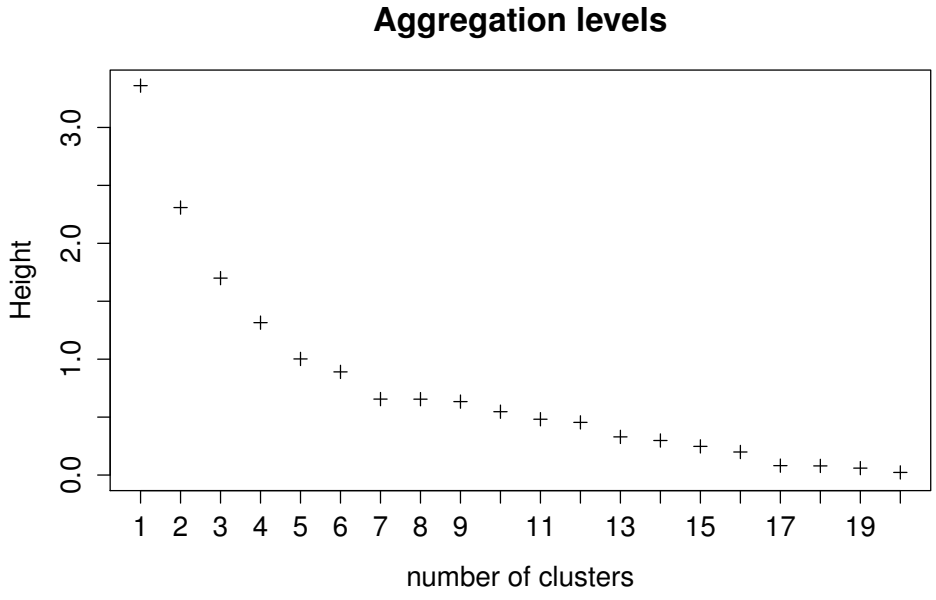
Tabela 3.1: Opis zmiennych (w kolumnie `Lp.` znajdują się numery zmiennych, w kolumnie `Zmienna` znajdują się nazwy zmiennych, a w kolumnie `Opis` przedstawione jest znaczenie zmiennych, tzn. informacja co poszczególna zmienna oznacza).

	Sex	Time.On.Dialysis	Age	Diabetes	Weight	Height	BSA	...
1	0	5	64	1	57.5	154	1.568	...
2	0	18	57	1	75.3	149	1.765	...
3	0	9	65	0	66.7	151	1.673	...
4	0	108	26	0	45.1	146	1.352	...
5	0	71	24	0	37.8	153	1.267	...
6	1	18	35	0	81.5	168	1.950	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	

Tabela 3.2: Fragment analizowanych danych (pierwsze 6 wierszy, pierwszych 7 zmiennych).



Rysunek 3.1: Dendrogram ukazujący budowę poszczególnych partycji.



Rysunek 3.2: Wykres przedstawiający poziom agregacji, tzn. jak bardzo informacja niesiona przez dany podział różni się od informacji niesionej przez poprzedni podział.

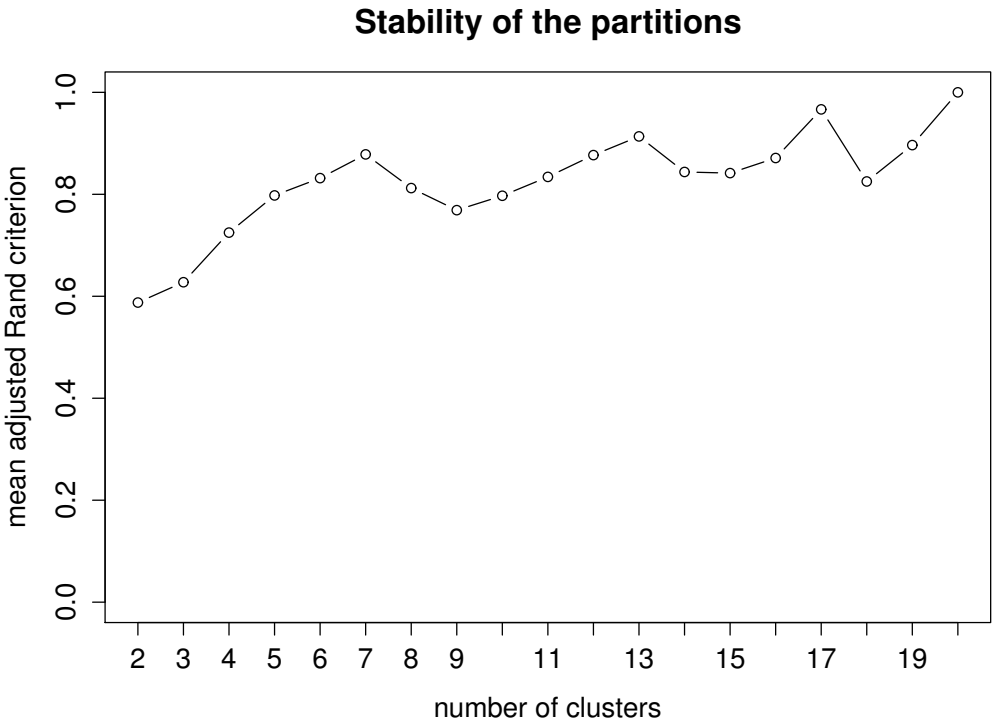
Lp.	**Maksymalna wartość funkcji height**	**Różnica maksymalnych wartości funkcji height**
20	0.023	
19	0.060	0.037
18	0.079	0.019
17	0.081	0.002
16	0.199	0.118
15	0.247	0.048
14	0.298	0.051
13	0.329	0.031
12	0.455	0.126
11	0.488	0.033
10	0.547	0.059
9	0.634	0.087
8	0.655	0.021
7	0.656	0.001
6	0.890	0.234
5	1.002	0.112
4	1.315	0.313
3	1.699	0.384
2	2.309	0.610
1	3.361	1.052

Tabela 3.3: W kolumnie `Lp.` znajduje się ilość klastrów w partycji, w drugiej kolumnie podana jest minimalna wartość funkcji height spośród partycji złożonych z Lp. klastrów, a w trzeciej kolumnie znajdują się różnice między maksymalnymi wartościami funkcji height partycji złożonych z *Lp.* i *Lp.* + 1 klastrów.

Takie przedstawienie danych sprawia, że jesteśmy w stanie prześledzić proces grupowania zmiennych i możemy np. stwierdzić jakie zmienne są najbardziej podobne do siebie, gdyż są łączone w pierwszej kolejności i tak odczytując z wykresu wynika, że zmienne `Weight` i `BSA` są najbardziej "spokrewnioną" parą ze wszystkich zmiennych (ich podobieństwo liczymy na podstawie wzorów (1.1),(1.2)). Taka informacja pozwala zaobserwować formowanie się pierwszych podziałów, co jest bardzo ważne przy stworzeniu sobie pierwszej intuicji przy grupowaniu zmiennych.

Za to rysunek 3.2 przedstawia poziom agregacji, tzn. obrazuje poszczególne poziomy łączenia grup zmiennych w większą całość. Poziom height klastra $C = A \cup B$ na rysunku 3.1 i rysunku 3.2 przedstawia tabela 3.3, i jest on definiowany wzorem g(C)=d(A,B), gdzie d (def. 1.2.1). Wartości funkcji height pozwalają sprawdzić, jak bardzo nasz podział odbiega od poprzedniego stanu, tzn. jak bardzo informacja niesiona przez partycję P_{K-1} różni się od przekazu partycji P_K. Przykładowo partycja mająca 17 klastrów różni się od partycji złożonej z 18 klastrów o ok. 0.03, jest to wielkość stosunkowo niewielka patrząc po wszystkich wartościach w tabeli 3.3, więc możemy wnioskować, iż podział na 17 klastrów jest równie dobry co do niesionej informacji jak podział na 18 klastrów.

Rysunek 3.1, rysunek 3.2 i tabela 3.3 sugerują wybrać 7 klastrów, lecz ten wybór nigdy nie jest jednoznaczny. Jedne osoby mogą przykładać większą wagę do liczby klastrów,



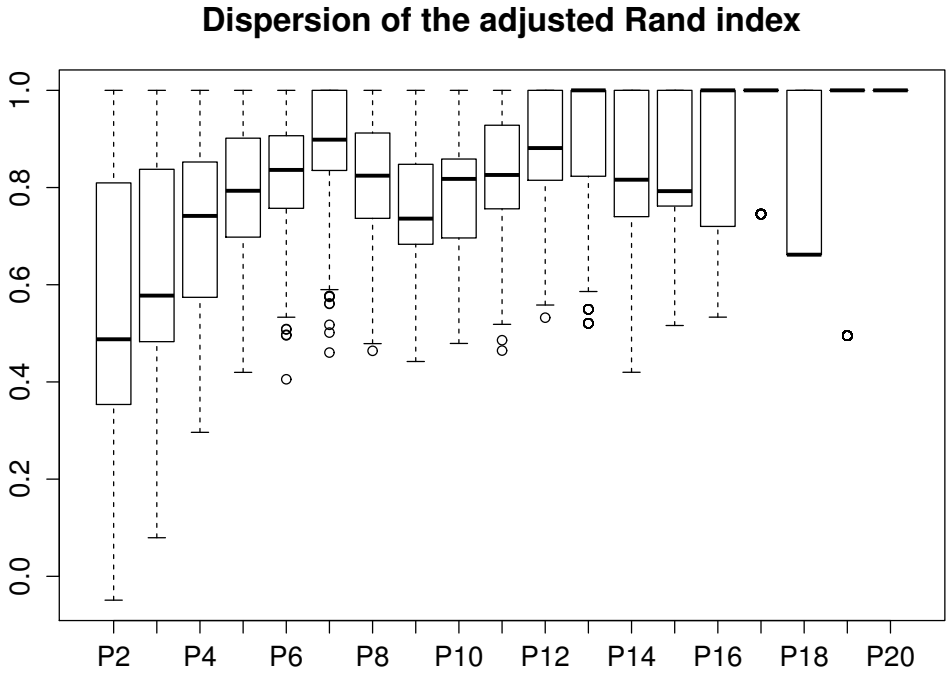
Rysunek 3.3: Wykres stabilności partycji.

a inne do wartości funkcji height. Dokonamy wyboru takiej ilości klastrów, przy której wartość height partycji o klaster większej jest stosunkowo większa od poprzednich wyników. W naszym przypadku taka różnica jest widoczna przy wyborze między 7 i 6 klastrami. Dla 7 klastrów otrzymujemy wartość 0.656, a dla 6 grup 0.890, więc różnica wynosi 0.234 i jest to wyróżniająca się zmiana.

Teraz posłużymy się funkcją `stability`, aby przy jej pomocy sprawdzić stabilność partycji.

Rysunek 3.3 przedstawia średni skorygowany indeks Rand otrzymany dla 1000 próbek bootstrap (ilość replikacji zależy od użytkownika, lecz branie ich zbyt dużo powoduje znaczne wydłużenie czasu oczekiwania na wynik, zaś zbyt mała ilość może prowadzić do niedokładnego rezultatu). Sugeruje on aby wybrać partycję z podziałem na 20, 17, 13 lub 7 klastrów, gdyż właśnie dla takich podziałów stabilność jest największa i są to maksima lokalne. Za to rysunek 3.4 obrazuje rozproszenie tych indeksów dla 1000 próbek bootstrap i podaje on więcej informacji niż rysunek 3.3 tzn. można odczytać z niego, że przy podziale na 20 klastrów wszystkie 1000 próbek ma indeks równy 1, przy podziale na 17 klastrów znajdują się pojedyncze indeksy poniżej wartości 0.8, a cała reszta jest równa 1. Podział na 13 klastrów zapewnia, że prawie wszystkie indeksy znajdą się w przedziale $[0.8; 1]$, tylko pojedyncze będą miały wartość ok. 0,5, a przy podziale na 7 klastrów rozproszenie indeksów jest zbliżone do podziału na 13 klastrów.

Podsumowując zebrane informacje wynika z nich, że najlepszym podziałem dla nas będzie partycja z 7 klastrami i taki też podział zażądamy od funkcji `cutreevar`, która przytnie nasz



Rysunek 3.4: Rozproszenie indeksu Rand.

dendrogram do podanego rozmiaru, zwracając m.in. informacje o rozmieszczeniu poszczególnych zmiennych (druga kolumna tabeli 3.4).

Mając już gotowy podział na klastry i reprezentantów tych klastrów tzn. zmienne centralne zdefiniowane w 1.1.2, które otrzymujemy z funkcji `cutreevar`, możemy sprawdzić jak dobrze te centralne zmienne klastra reprezentują dane klastry. W tabeli 3.5 przedstawiliśmy otrzymany klaster nr 6 i obciążenie kwadratowe poszczególnych zmiennych w nim zawartych. Obciążenie to jest podobieństwem pomiędzy zmienną i centralną zmienną klastra, w którym ta zmienna się znajduje. Dla zmiennych ilościowych jest to po prostu korelacja opisana wzorem (1.1), a dla zmiennych jakościowych opisuje to podobieństwo wzór (1.2). Wynika z niego, że kwadrat korelacji pomiędzy zmienną jakościowa `sex` i syntetyczną zmienną klastra szóstego wynosi w przybliżeniu 0.91, a kwadrat korelacji pomiędzy zmienną ilościową `ICW` i centralną zmienną klastra w przybliżeniu wynosi 0.83. Im ten wynik jest bliższy wartości 1 tym zmienne są podobniejsze.

Funkcja `cutreevar` zwraca nam macierz centralnych syntetycznych zmiennych 7 klastrów. Tą 90×7 macierzą danych numerycznych możemy zastąpić oryginalną 90×21 macierz przemieszanych danych zmiennych ilościowych i jakościowych.

Mając już pewien obraz naszej sytuacji, sprawdzimy funkcję `kmeansvar` dla 7 klastrów. Musimy jeszcze podać ilość losowych zestawów stosowanych w procesie i zrobimy to dla 10 takich zestawów (init = 10). W efekcie otrzymaliśmy partycję którą przedstawia trzecia kolumna tabeli 3.4.

Zmienna	**Przynależność zmiennych (hlustvar)**	**Przynależność zmiennych (kmeansvar)**
Time.On.Dialysis	1	1
Urine.Volume	1	2
Urine.Volume.Changed	1	2
Age	2	3
Diabetes	2	3
Serum.Albumin	2	4
Weight	3	4
Height	3	4
BSA	3	4
Blood.Pressure	4	3
TBW	5	5
ECW	5	5
Sex	6	6
ICW	6	6
ECW.TBW	6	6
Mean.Energy.Intake	7	7
Mean.Protein.Intake	7	7
Mean.Carbohydrate.Intake	7	7
Mean.Fat.Intake	7	7
Mean.Sodium.Intake	7	7
Mean.Water.Intake	7	7

Tabela 3.4: Przynależność zmiennych w partycji z 7 klastrami (w kolumnie `Zmienna` znajdują się nazwy zmiennych, w drugiej kolumnie znajduje się informacja o rozmieszczeniu zmiennych w klastrach dostarczona przez funkcję `hclustvar`, a w trzeciej kolumnie znajduje się informacja o rozmieszczeniu zmiennych w klastrach dostarczona przez funkcję `kmeansvar`, kolejność klastrów jest dowolna).

	squared loading
ICW	0.8289526
ECW.TBW	0.9501577
Sex	0.9138464

Tabela 3.5: Kwadrat obciążenia zmiennych w klastrze 6.

Porównując kolumnę 2 i 3 tabeli 3.4 widzimy, że nie przedstawiają tego samego (kolejność klastrów jest dowolna). Zmienna `Time.On.Dialysis` znajduje się sama w klastrze w wyniku funkcji `kmeansvar`, a funkcja `hclustvar` połączyła ją ze zmiennymi `Urine.Volume`, `Urine.Volume.Changed`. Więc dostajemy dwa podziały na 7 klastrów i aby wybrać lepszy z nich porównamy przyrost spójności obu partycji. Przyrost spójności jest wyrażoną wartością procentową jednorodności, która jest wykazywana przez partycję P_K. Zdefiniowana jest wzorem:

$$E(P_K) = \frac{H(P_K)-H(P_1)}{H(P_p)-H(P_1)},$$

gdzie funkcja jednorodności podziału $H(P_K)$ była zdefiniowana w (1.1.4). W naszym przypadku p=21 (dysponujemy 21 zmiennymi, więc $H(P_{21}) = 21$) i K=7 (badamy przyrost spójności dla partycji złożonej z 7 klastrów). W ten sposób otrzymujemy:

	Przyrost spójności(w %)
Partycja 7 klastrowa otrzymana metodą `hclustvar`	69.03894
Partycja 7 klastrowa otrzymana metodą `kmeansvar`	67.39863

Biorąc pod uwagę kryterium spójności, partycja otrzymana przez klastrowanie hierarhiczne za pomocą funkcji hclustvar jest lepszym wyborem. Dlatego ostatecznym rozwiązaniem naszego problemu podziału 21 zmiennych na grupy jest 7 klastrowa partycja, której podział przedstawia druga kolumna tabeli 3.4.

Podsumowanie

Celem naszej pracy było zrozumienie i opisanie pakietu ClustOfVar oraz zastosowanie go na rzeczywistych danych medycznych. Jest to swojego rodzaju wyjątkowy pakiet grupowania zmiennych, ponieważ wyróżnia go od innych to, iż potrafi grupować zmienne ilościowe, jakościowe oraz mieszane. Z tym że do grupowania zmiennych mieszanych trzeba zdefiniować specjalną miarę podobieństwa między zmienną ilościową i jakościową (wzór 1.2).

W pierwszym rozdziale przedstawiliśmy problem klastrowania zmiennych oraz opisaliśmy jego rozwiązanie z punktu widzenia omawianego pakietu. W tym celu opisaliśmy *centralną zmienną klastra, jednorodność klastra* i *jednorodność partycji* opisujące i wyjaśniające kryteria, na podstawie których dokonuje się podziału zmiennych na klastry. Udowodniliśmy także poprawność sposobu odnajdowania zmiennej centralnej oraz jednorodności klastra. Następnie przedstawiliśmy dwa algorytmy klastrowania zaimplementowane w pakiecie ClustOfVar. Algorytm hierarchiczny polega na stopniowym scalaniem mniejszych klastrów w coraz to większe przy jak najmniejszej utracie jednorodności partycji. Drugi z algorytmów polega na wyborze początkowego podziału i zmienianiu go przez relokowanie zmiennych tak, aby uzyskać jak największą jednorodność partycji.

Analizując dane użyliśmy najpierw algorytmu hierarchicznego. Uzyskaliśmy dzięki niemu podziały na różne ilości klastrów. Dendrogram partycji pozwolił nam zauważyć grupy powiązań wśród zmiennych oraz ich hierarchię, tzn. kolejność w jakiej dokonywały się kolejne scalenia. Na podstawie wartości funkcji height stwierdziliśmy wstępnie, że dobrym wyborem ilości klastrów będzie 7, ponieważ przy przejściu od podziału P_7 do P_6 zaczynamy obserwować zwiększoną utratę jednorodności partycji. Badanie stabilności potwierdziło, że jest to dobry wybór. Podział na 7 klastrów ma wysoką stabilność, co oznacza, że jest mało podatny na zaburzenia danych i może dobrze odzwierciedlać zależności wśród badanych zmiennych.

Następnie za pomocą funkcji `kmeansvar` dokonaliśmy innego podziału na 7 klastrów. Badając przyrost spójności obu partycji stwierdziliśmy, że lepszym wyborem będzie podział otrzymany za pomocą funkcji `hclustvar`.

Podsumowując analizę danych otrzymujemy, że najkorzystniejszym podziałem zmiennych jest partycja złożona z 7 klastrów. W ten sposób zmniejszyliśmy ilość zmiennych o ponad połowę (z 21 do 7), przy możliwie dobrym zachowaniu niesionej informacji, a początkową macierz 90×21 danych ilościowych i jakościowych zastąpimy macierzą 90×7 tylko i wyłącznie danych ilościowych, złożonej z centralnych zmiennych 7 klastrów.

A. Dowody stwierdzeń.

Dowód Stwierdzenia 1.1.1

Krok 1.

W tym kroku dla macierzy $X \in R^{n\times p}$ znajdziemy wektor $u \in R^n$ spełniający

$$u = \arg \max_{w\in R^n,\ ||w||=1} \left((w^T X)(w^T X)^T\right). \tag{3.1}$$

Powołując się na Twierdzenie 3.2. z [4] możemy dokonać rozkładu SVD macierzy X i zapisać:

$$X = ADB^T, \tag{3.2}$$

gdzie A i B są ortonormalne oraz $D = \begin{bmatrix} D^{'} & 0 \\ 0 & 0 \end{bmatrix}$, $D^{'}$ jest diagonalna, kwadratowa o rozmiarze równym rzędowi macierzy X, a wartości na jej diagonali są uporządkowane nierosnąco.

Niech $w \in R^n$, $||w|| = 1$. Kolumny A tworzą bazę ortonormalną przestrzeni R^n, więc istnieją stałe c_i t.że:

$$w = \sum_{i=1}^{n} c_i A^{(i)} \tag{3.3}$$

oraz

$$1 = ||w||^2 = ||\sum_{i=1}^{n} c_i A^{(i)}||^2 = \sum_{i=1}^{n} ||c_i A^{(i)}||^2 = \sum_{i=1}^{n} c_i^2, \tag{3.4}$$

zatem $\sum\limits_{i=1}^{n} c_i^2 = 1$.

Korzystając ze wzoru (3.2) otrzymujemy:

$$w^T X(w^T X)^T = w^T XX^T w = w^T ADB^T (ADB^T)^T w = (A^T w)^T DD^T (A^T w), \tag{3.5}$$

ale dzięki wzorowi (3.3) i ortonormalności A mamy:

$$A^T w = A^T \sum_{i=1}^{n} c_i A^{(i)} = \sum_{i=1}^{n} c_i A^T A^{(i)} = \sum_{i=1}^{n} c_i e_i = (c_1, ..., c_n)^T, \tag{3.6}$$

gdzie e_i to i-ty wektor bazy standardowej.

Zauważmy, że $DD^T = \begin{bmatrix} \left(D^{'}\right)^2 & 0 \\ 0 & 0 \end{bmatrix}$. Niech d_i oznacza i-ty wyraz z diagonali macierzy $D^{'}$. Ponieważ wyrazy z diagonali $D^{'}$ są uporządkowane niemalejąco zachodzą nierówności:

$d_1 \geqslant d_2 \geqslant ... \geqslant d_r > 0$ ($r = rank(X)$) i otrzymujemy:

$$w^T X(w^T X)^T = (c_1, ..., c_n) \begin{bmatrix} \left(D'\right)^2 & 0 \\ 0 & 0 \end{bmatrix} (c_1, ..., c_n)^T = \tag{3.7}$$

$$= \sum_{i=1}^{r} c_i^2 d_i^2 \leqslant \sum_{i=1}^{r} c_i^2 d_1^2 \leqslant \sum_{i=1}^{n} c_i^2 d_1^2 = d_1^2. \tag{3.8}$$

Zatem:

$$w^T X(w^T X)^T \leqslant d_1^2. \tag{3.9}$$

Ze wzorów (3.7) i (3.8) wynika, że $w^T X(w^T X)^T$ przyjmuje maksymalną wartość równą d_1^2 dla wektora $w = A^{(1)}$.

Zapisując $Z := \frac{1}{\sqrt{n}} X$ i przechodząc przez równania (3.5), (3.7) oraz (3.8) otrzymamy wynik $w^T Z(w^T Z)^T \leqslant \frac{1}{n} d_1^2$. Zatem $w^T Z(w^T Z)^T$ przyjmuje maksymalną wartość równą d_1^2 na wektorze $\sqrt{n} A^{(1)}$. W dalszych krokach ograniczymy się więc do dowodzenia tezy dla $Z = (X|Y)$ i nie będziemy przemnażali macierzy przez $\frac{1}{n}$.

Krok 2.

Pokażemy teraz związek wyniku uzyskanego w Kroku 1. z miarą r^2 (wzór (1.1)).

Niech $x_1, ..., x_p$ będą zmiennymi ilościowymi. Przypomnijmy, że X_i to przekodowany wektor x_i (tzn. $X_{i,j} = \frac{x_{i,j} - \overline{x_i}}{\rho_{x_i}}$) oraz $X = (X_1|...|X_p)$. Przez $< \cdot, \cdot >$ będziemy oznaczać standardowy iloczyn skalarny w przestrzeni R^n. Niech $w^T = (w_1, ..., w_n)$, $||w^T|| = 1$.

$$w^T X = (w_1, ..., w_n)(X_1|...|X_p) = (< w, X_1 >, ..., < w, X_p >), \tag{3.10}$$

zatem:

$$w^T X(w^T X)^T = \sum_{i=1}^{p} < w, X_i >^2 . \tag{3.11}$$

Zbadajmy pojedynczy składnik sumy z prawej strony równania (3.11):

$$< w, X_1 >= \sum_{i=1}^{n} w_i X_{1,i} = \sum_{i=1}^{n} (w_i - \overline{w} + \overline{w}) X_{1,i} = \tag{3.12}$$

$$= \sum_{i=1}^{n} (w_i - \overline{w}) X_{1,i} + \overline{w} \sum_{i=1}^{n} X_{1,i} = \sum_{i=1}^{n} (w_i - \overline{w}) X_{1,i} + \overline{w} n \overline{X_1} = \sum_{i=1}^{n} (w_i - \overline{w}) X_{1,i}, \tag{3.13}$$

bo X_1 jest scentrowany. Zatem pamiętając, że X_i jest przekodowaną wersją x_i, otrzymujemy:

$$< w, X_1 >= \sum_{i=1}^{n} (w_i - \overline{w}) X_{1,i} = \sum_{i=1}^{n} (w_i - \overline{w}) \left(\frac{x_{1,i} - \overline{x_1}}{\rho_{x_1}} \right) = \rho_w r(x_1, w). \tag{3.14}$$

Analogicznie $< w, X_i >= \rho_w r(x_i, w)$ dla pozostałych i. Mając (3.11) i (3.14) otrzymujemy:

$$w^T X(w^T X)^T = \rho_w^2 \sum_{i=1}^{p} r^2(x_i, w). \tag{3.15}$$

Teraz wiążąc rezultaty (3.9) i (3.15) otrzymujemy, że $A^{(1)}$ z rozkładu SVD macierzy X maksymalizuje wzór (3.15).

Krok 3.

Pokażemy teraz związek wyniku uzyskanego w Kroku 1. z miarą η^2 (wzór (1.2)).

Niech y_j, $j = 1, ..., p$ będą zmiennymi jakościowymi, każdy o m_j kategorii. Przypominamy oznaczenie $Y = JGD^{-\frac{1}{2}}$, gdzie $G = (G_1|...|G_p)$, G_j jest macierzą wymiaru $n \times m_j$, która w i-tej kolumnie ma jedynki na współrzędnych, na których y_j przyjmuje swoją i - tą kategorię i zera na pozostałych miejscach. $D = diag(D_1, ..., D_p)$, D_j na i-tym miejscu w diagonali ma ilość wystąpień i-tej kategorii w y_j. Możemy teraz zauważyć kilka faktów, które nieco ułatwią obliczenia:

(a) $GD^{-\frac{1}{2}} = (G_1 D_1^{-\frac{1}{2}}|...|G_p D_p^{-\frac{1}{2}})$,

(b) $J = Id_n - \frac{1}{n} \cdot \mathbb{1}_n = J^T$,

(c) dla wektora $w = (w_1, ..., w_n)^T$ mamy: $Jw = (w_1 - \overline{w}, ..., w_w - \overline{w})^T$.

Korzystając z (b) i (c) i oznaczając $\overline{W} := (\overline{w}, ..., \overline{w})^T$ możemy zapisać:

$$w^T Y Y^T w = w^T JGD^{-\frac{1}{2}} (JGD^{-\frac{1}{2}})^T w = w^T JGD^{-\frac{1}{2}} D^{-\frac{1}{2}} G^T J^T w = \tag{3.16}$$

$$= (Jw)^T GD^{-\frac{1}{2}} D^{-\frac{1}{2}} G^T (Jw) = (w - \overline{W})^T GD^{-\frac{1}{2}} D^{-\frac{1}{2}} G^T (w - \overline{W}). \tag{3.17}$$

Oznaczmy teraz:

$$C = GD^{-\frac{1}{2}} =: \left(C^{(1)}|...|C^{(m)}\right). \tag{3.18}$$

$$(w - \overline{W})^T C = (w - \overline{W})^T \left(C^{(1)}|...|C^{(m)}\right) = \left(< w - \overline{W}, C^{(1)} >, ..., < w - \overline{W}, C^{(m)} >\right). \tag{3.19}$$

Zatem

$$w^T Y Y^T w = ((w - \overline{W})^T C)((w - \overline{W})^T C)^T = \sum_{k=1}^{m} < w - \overline{W}, C^{(k)} >^2 . \tag{3.20}$$

Ustalmy teraz $j \leqslant p$, $i \leqslant m_j$. Na podstawie własności (a) możemy wybrać k takie, że $C^{(k)}$ jest i-tą kolumną $G_j D_j^{-\frac{1}{2}}$. $C^{(k)}$ ma na współrzędnych odpowiadających wystąpieniom i-tej kategorii w y_j wyrazy równe $\frac{1}{\sqrt{n_{j,i}}}$ (gdzie $n_{j,i}$ oznacza ilość wystąpień i-tej kategorii w y_j) oraz zera na pozostałych miejscach, zatem

$$< w - \overline{W}, C^{(k)} >^2 = (< w, C^{(k)} > - < \overline{W}, C^{(k)} >)^2 = (\sum_{l=1}^{n} w_l F(l) - < \overline{W}, C^{(k)} >)^2, \tag{3.21}$$

gdzie

$$F(l) := \begin{cases} \frac{1}{\sqrt{n_{j,i}}} = \frac{\sqrt{n_{j,i}}}{n_{j,i}} & \text{gdy } y_j \text{ na l-tej współrzędnej ma i-tą kategorię} \\ 0 & \text{w przeciwnym przypadku.} \end{cases}$$

Zatem

$$\sum_{l=1}^{n} w_l F(l) =: \sqrt{n_{j,i}} \cdot \overline{w_{j,i}}, \tag{3.22}$$

gdzie $\overline{w_{j,i}}$ jest średnią w policzoną na obserwacjach y_j należących do i-tej kategorii.

Ponieważ $C^{(k)}$ ma dokładnie $n_{j,i}$ wyrazów równych $\sqrt{n_{j,i}}$ i zera poza tym mamy:

$$< \overline{W}, C^{(k)} >= \overline{w}\frac{1}{\sqrt{n_{j,i}}}n_{j,i} = \overline{w}\sqrt{n_{j,i}}. \tag{3.23}$$

Zatem

$$< w - \overline{W}, C^{(k)} >^2 = n_{j,i}\left(\overline{w_{j,i}} - \overline{w}\right)^2, \tag{3.24}$$

$$\sum_{k:\ C^{(k)}\ jest\ kolumna\ G_j D_j^{-\frac{1}{2}}} < w - \overline{W}, C^{(k)} >^2 = \rho_w^2 \eta^2(w; y_j), \tag{3.25}$$

$$w^T Y Y^T w = \sum_{k=1}^{m} < w - \overline{W}, C^{(k)} >^2 = \rho_w^2 \sum_{j=1}^{p} \eta^2(w; y_j). \tag{3.26}$$

Krok 4.

Aby zakończyć dowód, dodamy kilka obserwacji do wyników uzyskanych w poprzednich krokach.

Jeśli macierz Z ma scentrowane kolumny, to korzystając z rozkładu SVD $Z = ADB^T$ mamy:

$$JADB^T = JZ = Z = ADB^T, \tag{3.27}$$

zatem A również ma scentrowane kolumny i

$$\rho_{A^{(1)}} = \sqrt{\sum_{i=1}^{n}\left(A_j^{(1)} - \overline{A^{(1)}}\right)^2} = \sqrt{\sum_{i=1}^{n}\left(A_j^{(1)} - 0\right)^2} = ||A^{(1)}|| = 1. \tag{3.28}$$

Teraz wiedząc, że $A^{(1)}$ maksymalizuje wzory (3.15) i (3.26), otrzymujemy, że $A^{(1)}$ maksymalizuje oddzielnie sumę r^2 dla klastra zmiennych ilościowych oraz sumę η^2 dla klastra zmiennych jakościowych. Aby pokazać, że $A^{(1)}$ również maskymalizuje łącznie sumy r^2 i η^2 dla klastra zmiennych obu typów zauważmy:

$$(X|Y)^T = \begin{pmatrix} X^T \\ Y^T \end{pmatrix}, \tag{3.29}$$

więc dla $Z = (X|Y)$ mamy:

$$w^T Z Z^T w = w^T (X|Y) \begin{pmatrix} X^T \\ Y^T \end{pmatrix} w = w^T (XX^T + YY^T) w = w^T XX^T w + w^T YY^T w, \tag{3.30}$$

czyli $w = A^{(1)}$ maksymalizuje sumę $r^2(x_i, w)$ i $\eta^2(w; y_j)$.

Ponadto łatwo zauważyć (np. przez zaprzeczenie), że jeśli w_0, o normie równej 1 maksymalizuje równanie $(w^T X)(w^T X)^T$, to wektorem o normie $\alpha \in R$, dla którego wartość $(w^T X)(w^T X)^T$ jest największa jest αw_0. Zatem wektor $A^{(1)} d_1$ również maksymalizuje sumę $r^2(x_i, u)$ i $\eta^2(w; y_j)$.

Dowód Stwierdzenia 1.1.2

To stwierdzenie wynika z dowodu Stwierdzenia 1.1.1. Patrząc na wzory (3.15) i (3.26) oraz uwagi z Kroku 4. otrzymujemy, że dla $c = \sqrt{n}U^{(1)}\lambda_1$

$$c^T Z Z^T c = \rho_c^2 \left(\sum_{x_j \in C_k} r^2(x_j, c_k) + \sum_{y_j \in C_k} \eta^2(c; y_j) \right). \tag{3.31}$$

Ponadto na podstawie uwagi do wzoru (3.27) i (3.28) otrzymamy:

$$\rho_c = ||c|| = \sqrt{n}\lambda_1, \tag{3.32}$$

przy czym $\sqrt{n}$ z wektora c skróci się z $\frac{1}{\sqrt{n}}$ z zapisu $Z = \frac{1}{\sqrt{n}}(X|Y)$, więc:

$$c^T Z Z^T c = \lambda_1^2 \left(\sum_{x_j \in C_k} r^2(x_j, c_k) + \sum_{y_j \in C_k} \eta^2(c; y_j) \right) = \lambda_1^2 h(C_k). \tag{3.33}$$

Z drugiej strony naśladując Krok 1. z dowodu Stwierdzenia 1.1.1 otrzymamy wynik:

$$c^T Z Z^T c = \lambda_1^2 \lambda_1^2. \tag{3.34}$$

Zatem

$$h(C_k) = \sum_{x_j \in C_k} r^2(x_j, c_k) + \sum_{y_j \in C_k} \eta^2(c_k; y_j) = \lambda_1^2. \tag{3.35}$$

B. Kody pakietu R użyte w pracy

instalujemy pakiet ClustOfVar wraz z wszystkimi pakietami wymaganymi do jego działania

```
> install.packages("ClustOfVar", dependencies = TRUE)
```

włączamy pakiet ClastOfVar

```
> library("ClustOfVar")
```

wczytanie danych z komputera, fragment danych przedstawia tabela 3.2

```
> dane = read.table("Study4nerki.csv", sep=";", header=TRUE, dec=",")
```

rozdzielamy nasze dane na dane ilościowe i dane jakościowe

```
> X.quanti <-dane[,c(2,3,5:21)]
> X.quali <- dane[,c(1,4)]
```

przy pomocy funckcji `factor` modyfikuję dane jakościowe, aby były typu categorical (ponieważ takiego typu danych jako jeden z argumentów używa funkcja `hclustvar`), przy okazji zmieniam etykiety tych danych, lecz jest to tylko zmiana kosmetyczna

```
> X.quali[,1]=factor(X.quali[,1], labels=c("men","women"))
> X.quali[,2]=factor(X.quali[,2], labels=c("no","yes"))
```

wywołuję funkcję `hclustvar`, a jej wartości zapisane będą pod nazwą `tree`

```
> tree <- hclustvar(X.quanti,X.quali)
```

dendrogram (rysunek 3.1) i wykres poziomu agregacji (rysunek 3.2)

```
> plot(tree)
```

wartości funkcji `height` (tabela 3.3)

```
> tree$height
```

wywołuję funkcję `stability`, a jej wartości zapisane będą pod nazwą `stab`

```
> stab <- stability(tree, B=1000)
```

wykres stabilności partycji (rysunek 3.3

```
> plot(stab, main = "Stability of the partitions")
```

wykres rozproszenia indeksu Rand (rysunek 3.4)

```
> boxplot(stab$matCR, main = "Dispersion of the adjusted Rand index" )
```

wywołuję funkcję `cutreevar`, a jej wartości zapisane będą pod nazwą `P7`

```
> P7 <- cutreevar(tree, 7)
```

przynależność poszczególnych zmiennych w partycji z 7 klastrami-funkcja `hclustvar` (druga kolumna tabeli 3.4)

```
> P7$cluster
```

lista macierzy obciążenia kwadratowego, fragment tej listy tzn. kwadrat obciążenia zmiennych w klastrze 6 przedstawia tabela 3.5

```
> P7$var
```

wywołuję funkcję `kmeansvar`, a jej wartości zapisane będą pod nazwą `part_km`

```
> part_km <- kmeansvar(X.quanti, X.quali, init = 7, nstart = 10)
```

przynależność poszczególnych zmiennych w partycji z 7 klastrami-funkcja `kmeansvar` (trzecia kolumna tabeli 3.4)

```
> part_km$cluster
```

przyrost spójności partycji złożonej z 7 klastrów, otrzymanej przy pomocy funckcji `hclustvar`

```
> P7$E
```

przyrost spójności partycji złożonej z 7 klastrów, otrzymanej przy pomocy funckcji `kmeansvar`

```
> part_km$E
```

macierz złożona z centralnych zmiennych 7 klastrów, to nią zastąpimy nasze początkowe dane

```
> P7$scores
```

Więcej informacji na temat pakietu R można znaleźć w [6].

Bibliografia

[1] M. Chavent, V. Kuentz-Simonet, B. Liquet, J. Saracco, *ClustOfVar: An R Package for the Clustering of Variables*, Journal of Statistical Software, 50 (2012).

[2] K. Y. Yeung, W. L. Ruzzo, *An empirical study on Principal Component Analysis for clustering gene expression data*, Bioinformatics, 17 (2001) 763–774.

[3] M. Chavent, V. Kuentz-Simonet, J. Saracco, *Orthogonal rotation in PCAMIX*, Advances in Classification and Data Analysis, 6 (2011) 131–146.

[4] P. Pokarowski, A. Prochenka, *Statystyka II wykłady*, `http://www.mimuw.edu.pl/ pokar/StatystykaII/wyklad.pdf`, dostęp dnia 29.04.2013r.

[5] *Package 'ClustOfVar'*, `http://cran.r-project.org/web/packages/ClustOfVar/ClustOfVar.pdf`, dostęp dnia 29.04.2013r.

[6] Przemysław Biecek, *Przewodnik po pakiecie R*, `http://www.biecek.pl/R/R.pdf`, dostęp dnia 29.04.2013r.