AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

DBSCAN – segmentacja danych punktowych oraz rastrowych w środowisku wolnodostępnego oprogramowania R

Mgr inż. Agnieszka Ochałek
Narzędzia informatyczne w badaniach naukowych
Wydział Geodezji Górniczej i Inżynierii Środowiska
Katedra Ochrony Terenów Górniczych, Geoinformacji i Geodezji Górniczej

Kraków 12.01.2018 r.

KONCEPCJA ALGORYTMÓW GĘSTOŚCIOWYCH GRUPOWANIA DANYCH

➢ Analiza skupień (grupowanie obiektowe, ang. *cluster analysis*) jest zadaniem eksploracji danych, które polega na dzieleniu zbioru danych na grupy w taki sposób, by elementy w tej samej grupie były do siebie podobne, a jednocześnie jak najbardziej odmienne od elementów z pozostałych grup.

➢ Przykładowe algorytmy grupowania danych:

- Algorytmy oparte na podziale (K-means, PAM clustering)
- Algorytmy hierarchiczne
- Algorytmy oparte na gęstościach (ang. density-based algorithms) – (DBSCAN, OPTICS, DENCLUE)

Algorytmy gęstościowe – jest to grupa algorytmów, która obok pojęcia odległości wykorzystuje pojęcie gęstości. Główną ideą takich algorytmów jest obserwacja, że wewnątrz grup gęstość punktów jest znacznie większa niż poza nimi. Oznacza to, że dopóki liczba obiektów wokół klastra jest duża (z zadanym parametrem) klaster ten będzie się powiększał.

Zalety algorytmów gęstościowych:

✓ Nie wymagają od użytkownika określenia liczby generowanych klastrów.
✓ Znajdują dowolny kształt klastrów - klaster nie musi być sferyczny.
✓ Identyfikują wartości odstające („szumy”).

BIBLIOTEKA „DBSCAN” - DENSITY BASED CLUSTERING OF APPLICATIONS WITH NOISE AND RELATED ALGORITHMS

➢Wersja: 1.1-1

➢Data: 19.03.2017 r.

➢Autorzy: Michael Hahsler, Matthew Piekenbrock, Sunil Arya, David Moun

➢Opis:

Biblioteka zawierająca kilka algorytmów grupowania danych przestrzennych opartych na gęstości.

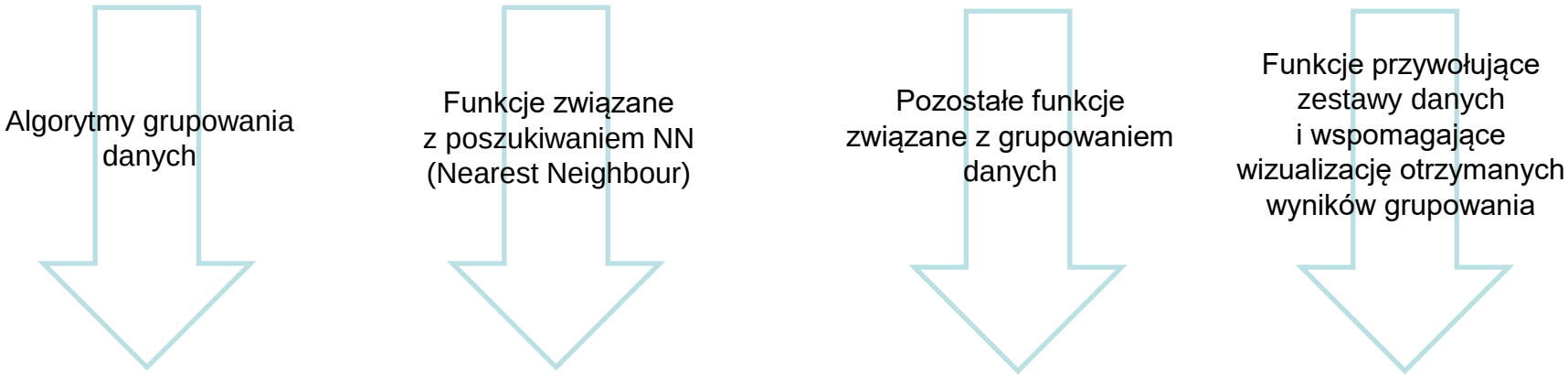
Obejmuje algorytmy:

- DBSCAN (density-based spatial clustering of applications with noise),
- OPTICS (ordering points to identify the clustering structure),
- HDBSCAN (hierarchical DBSCAN)
- LOF (local outlier factor)
- GLOSH (Global-Local Outlier Score).

Biblioteka używa struktury kd-tree (z biblioteki ANN) do szybszego wyszukiwania najbliższego sąsiada (k-nearest neighbor - kNN). Wyszukiwanie najbliższego sąsiada kNN oraz sąsiada po określonym promieniu (fixed-radius NN) jest zapewnione przez **Jarvis-Patrick clustering** oraz **Shared Nearest Neighbor Clustering**. Dodatkowo, możliwe jest wykorzystanie algorytmu **Framework for Optimal Selection of Clusters (FOSC),** który obsługuje nienadzorowane i półnadzorowane grupowanie hierarchiczne przy użyciu drzewa klastrów (obiekt "hclust"). Obsługuje dowolne kryterium powiązania.

BIBLIOTEKA „DBSCAN” - DENSITY BASED CLUSTERING OF APPLICATIONS WITH NOISE AND RELATED ALGORITHMS

18 FUNKCJI



Algorytmy grupowania danych

- dbscan
- extractFOSC
- OPTICS
- GLOSH
- hdbscan
- jpclust

Funkcje związane z poszukiwaniem NN (Nearest Neighbour)

- NN
- frNN
- kNN
- kNNdist
- sNN
- sNNclust

Pozostałe funkcje związane z grupowaniem danych

- lof
- pointdensity

Funkcje przywołujące zestawy danych i wspomagające wizualizację otrzymanych wyników grupowania

- moons
- DS3
- hullplot
- reachability

ALGORYTMY GRUPOWANIA DANYCH

- dbscan - *Density-based spatial clustering of applications with noise*
- extractFOSC - *Framework for Optimal Selection of Clusters*
- OPTICS - *Ordering points to identify the clustering structure*
- *GLOSH* - *Global-Local Outlier Score from Hierarchies*
- hdbscan – *hierarchiczny DBSCAN*
- jpclust - *Jarvis-Patrick Clustering*

Funkcje związane z poszukiwaniem NN (Nearest Neighbour)

➢frNN - *Find the Fixed Radius Nearest Neighbors* – funkcja używa struktury kd-tree do odnajdywania najbliższych sąsiadów (wraz z odległościami) przy podaniu określonego promienia wyszukiwania.

```
frNN(x, eps, sort = TRUE, search = "kdtree", bucketSize = 10, splitRule = "suggest", approx = 0)
```

➢kNN - *Find the k Nearest Neighbors* - używa struktury kd-tree do znalezienia wszystkich k-najbliższych sąsiadów w macierzy danych (łącznie z odległościami). Funkcja szybka dla dużej ilości danych.

```
kNN(x, k, sort = TRUE, search = "kdtree", bucketSize = 10, splitRule = "suggest", approx = 0)
```

➢NN - *Nearest Neighbors Auxiliary Functions* – funkcja pomocnicza dla funkcji kNN oraz fNN. Zwraca listę przyległości (*adjacency list*) dla każdego oryginalnego punktu danych. Lista zawiera numery identyfikacyjne najbliższych sąsiadów wszystkich punktów. Lista przyległości przykładowo może zostać użyta do stworzenia wykresu obiektu.

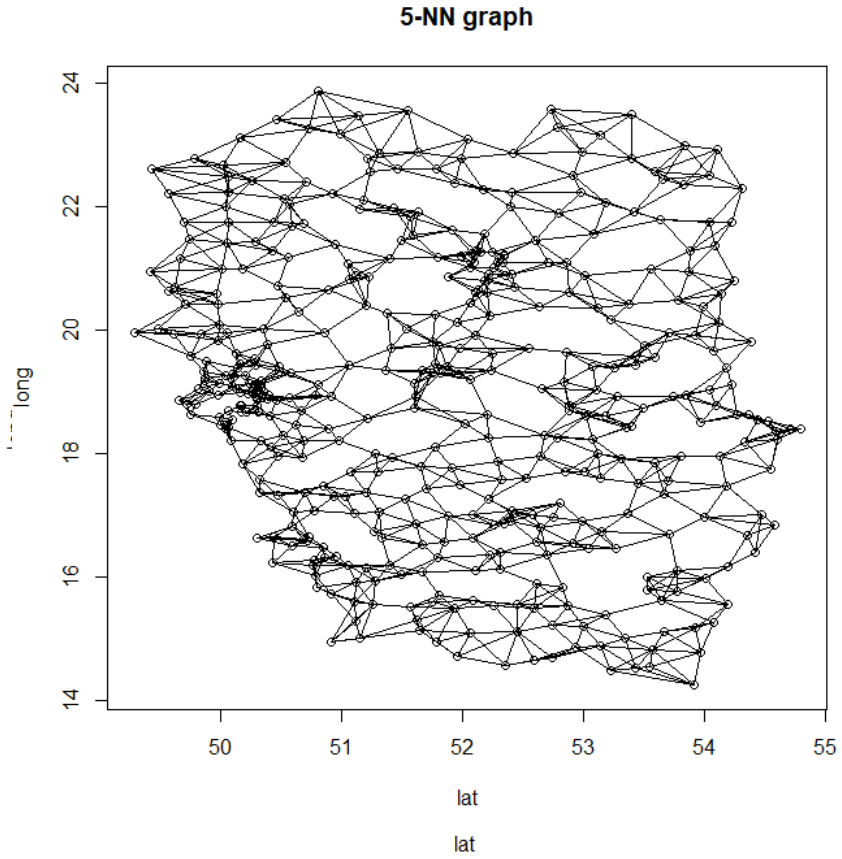
➢kNNdist - *Calculate and plot the k-Nearest Neighbor Distance* – pozwala na szybkie wyznaczenie odległości k-najbliższych sąsiadów w macierzy punktów. Wykres stworzony na bazie funkcji może być używany do pomóc w znalezieniu odpowiedniej wartości parametru eps dla funkcji DBSCAN (szukanie „kolana” w wykresie).

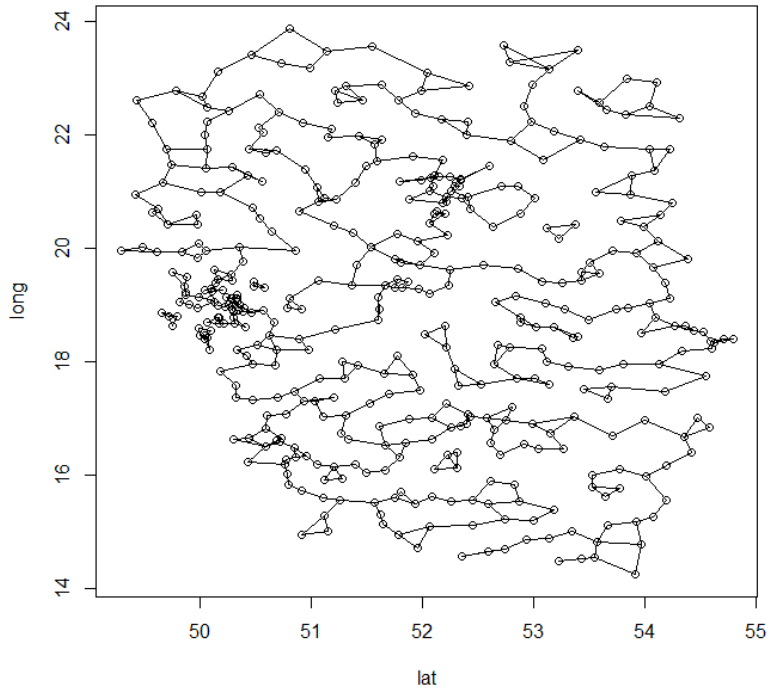
➢sNN - *Shared Nearest Neighbors* – pozwala na wyznaczenie liczby wspólnych najbliższych sąsiadów. Liczba współdzielonych najbliższych sąsiadów jest przecięciem sąsiedztwa kNN dwóch punktów. Każdy punkt jest uważany za część jego własnej dzielnicy kNN. Zakres współdzielenia sąsiadów to [0, k].

➢sNNclust - *Shared Nearest Neighbor Clustering* – wyszukiwanie współdzielonych najbliższych sąsiadów na podstawie algorytmu Ertoza, Steinbacha i Kumara.

FUNKCJA KNN – PRZYKŁAD

```
# wczytanie danych punktowych, np. miasta Polski przy użyciu biblioteki world cities
data("world.cities")
PL <- world.cities %>% filter(country.etc == "Poland")

# wybranie odpowiednich danych do macierzy
x <- select(PL, lat, long)

# bezpośrednie znajdowanie kNN w danych (przy użyciu kd-tree)
nn <- kNN(x, k=5)

# wizualizacja 5 najbliższych sąsiadów
plot(nn, x)

# wizualizacja 2-NN sąsiadów
plot(kNN(nn, k = 2), x)
```





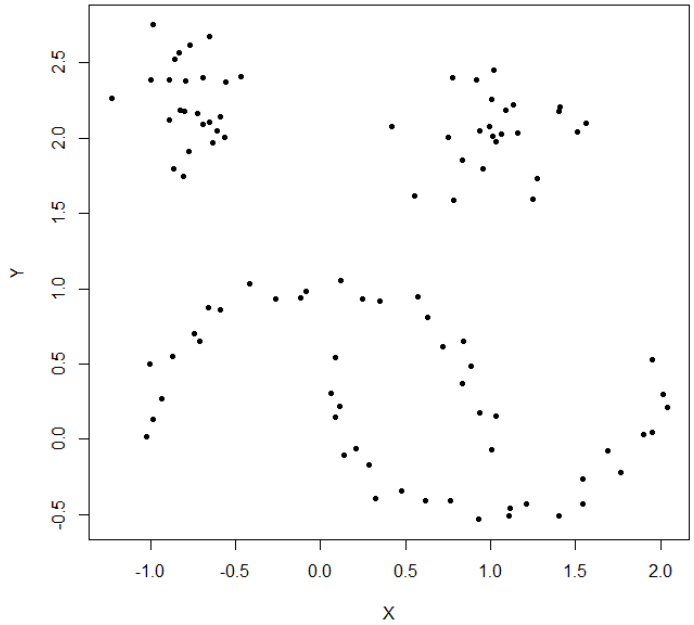
FUNKCJE PRZYWOŁUJĄCE ZESTAWY DANYCH I WSPOMAGAJĄCE WIZUALIZACJĘ OTRZYMANYCH WYNIKÓW GRUPOWANIA

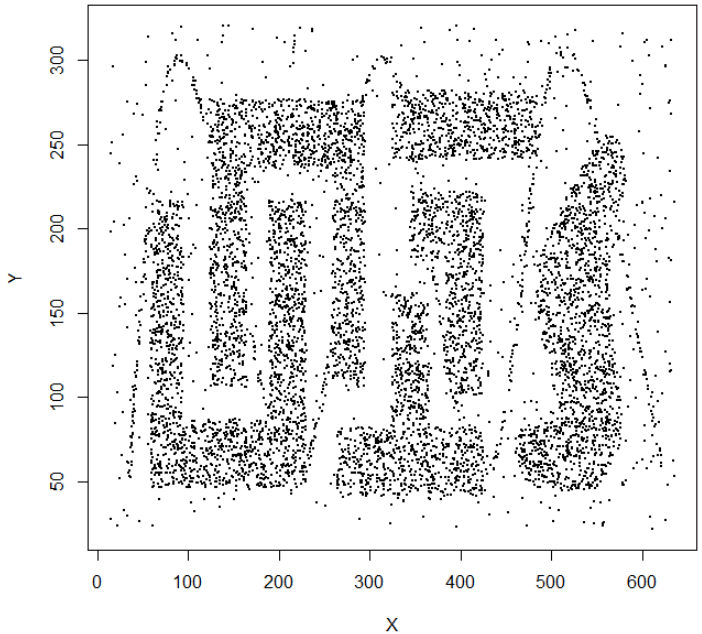
- moons - *Moons Data - zbiór 100 punktów 2d, z których połowa jest zawarta w dwóch księżycach (po 25 punktów każdy), a drugą połowa w asymetrycznych kształtach półksiężyca. Wszystkie trzy kształty są liniowo rozdzielne.*

- DS3: *Spatial data with arbitrary shapes – zbiór 8000 punktów 2d z „naturalnie" wyglądającymi kształtami. Pierwotnie używany jako zestaw danych porównawczych dla algorytmu grupowania Chameleon do zilustrowania zestawu danych, który zawieraja dowolnie ukształtowane dane przestrzenne otoczone zarówno szumem, jak i artefaktami.*

- hullplot - *Plot Convex Hulls of Clusters - tworzy dwuwymiarowy wykres punktowy z dodanymi wypukłymi pokrywami dla klastrów.*

- reachability - *Density Reachability Structures - zapewnia ogólne funkcje reprezentowania hierarchicznego grupowania jako "wykresy osiągalności" (dendrogramy).*





FUNKCJE ZWIĄZANE Z ALGORYTMAMI GRUPOWANIA DANYCH

➢LOF - *Local Outlier Factor Score* – funkcja wykorzystywana do przyspieszenia działania kNN. Oblicza współczynnik odchylenia lokalnego. LOF porównuje lokalną gęstość punktu do lokalnych gęstości jego sąsiadów. Wskazuje to mają znacznie niższą gęstość niż ich sąsiedzi są uważane za wartości odstające. Wynik LOF około 1 oznacza, że gęstość wokół punktu jest porównywalna z sąsiadami. Wyniki znacznie większy niż 1 wskazuje wartości odstające.

➢pointdensity - *Calculate Local Density at Each Data Point - oblicza gęstość lokalną w każdym punkcie danych jako:*

- *liczba punktów w sąsiedztwie eps (jak w funkcji DBSCAN)*
- *kernel density estimate (kde) of uniform kernel.*

FUNKCJA DBSCAN

Funkcja wykorzystuje algorytm gęstościowy grupowania danych DBSCAN przy użyciu struktury kd-tree (struktura danych używana do dzielenia przestrzeni). Funkcja jest szybsza i działa dla większej ilości danych niż dbscan w bibliotece fpc.

Funkcja oparta na algorytmie opisanym w 1996 roku przez Martina Estera, Hansa-Petera Kriegela, Joerga Sandera, Xiaowei Xu w artykule „A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”.

SKŁADNIA FUNKCJI DBSCAN

dbscan(x, eps, minPts = 5, weights = NULL, borderPoints = TRUE, ...)

X – macierz, ramka danych, obiekt „dist” (dissimilarity matrix). Alternatywnie, można użyć obiektu fNN (z określonym promieniem najbliższego sąsiada). W tym wypadku eps może zostać pominięte, ponieważ zostanie pobrane z obiektu fNN.

eps – maksymalny promień sądziedztwa.

minPts – minimalna liczba obiektów w regionie określonego eps. Domyślnie 5 punktów.

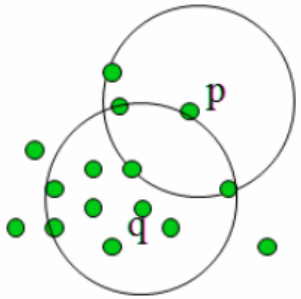
weights – parametr numeryczny; wagi dla danych punktów. Wymagane tylko do wykonywania grupowania ważonego.

borderPoints – parametr logiczny; ustalenie punktów granicznych. Domyślnie ustawiona wartość – TRUE. Jeśli FALSE, to punkty graniczne uznawane są za szum.

… - dodatkowe argumenty przekazywane do algorytmu frNN.

ALGORYTM DBSCAN W R

DBSCAN jako wejście przyjmuje dwa parametry. Pierwszy z nich to minimalna ilość punktów wymagana do utworzenia grupy oznaczany (minPts), natomiast drugi to maksymalny promień sąsiedztwa (Eps). Samo sąsiedztwo (lub otoczenie) jest to zbiór punktów D leżących w odległości mniejszej bądź równiej Eps od danego punktu p i jest definiowane jako:



MinPts = 5

Eps = 1 cm

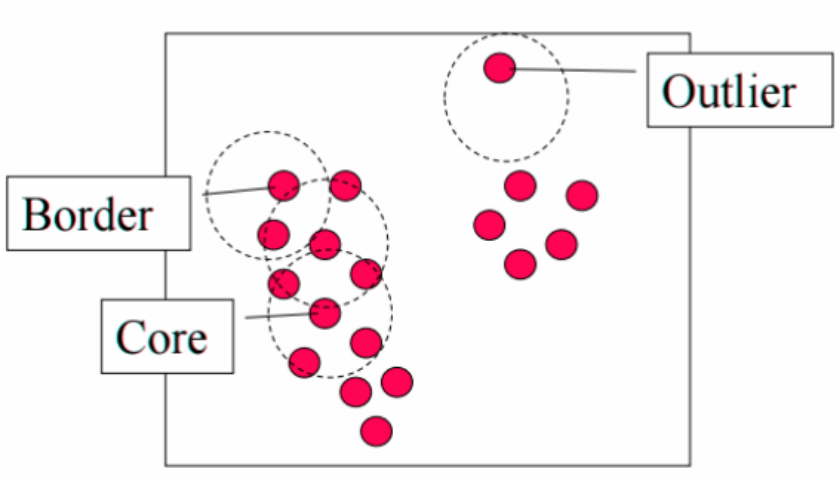
$$N_{Eps}(p) = \{q \in D : d(p,q) \leq Eps\},$$

gdzie d(p,q) odległość między punktami p i q.

Rdzeń (ang. core point) – punkt p spełniający warunek taki, że w jego epsilonowym sąsiedztwie znajduje się co najmniej minPts punktów.

$$|N_{Eps}(p)| \geq minPts$$

Punkt graniczny (ang. border point) – punkt,
który nie jest rdzeniem ale jest osiągalny z innego rdzenia.

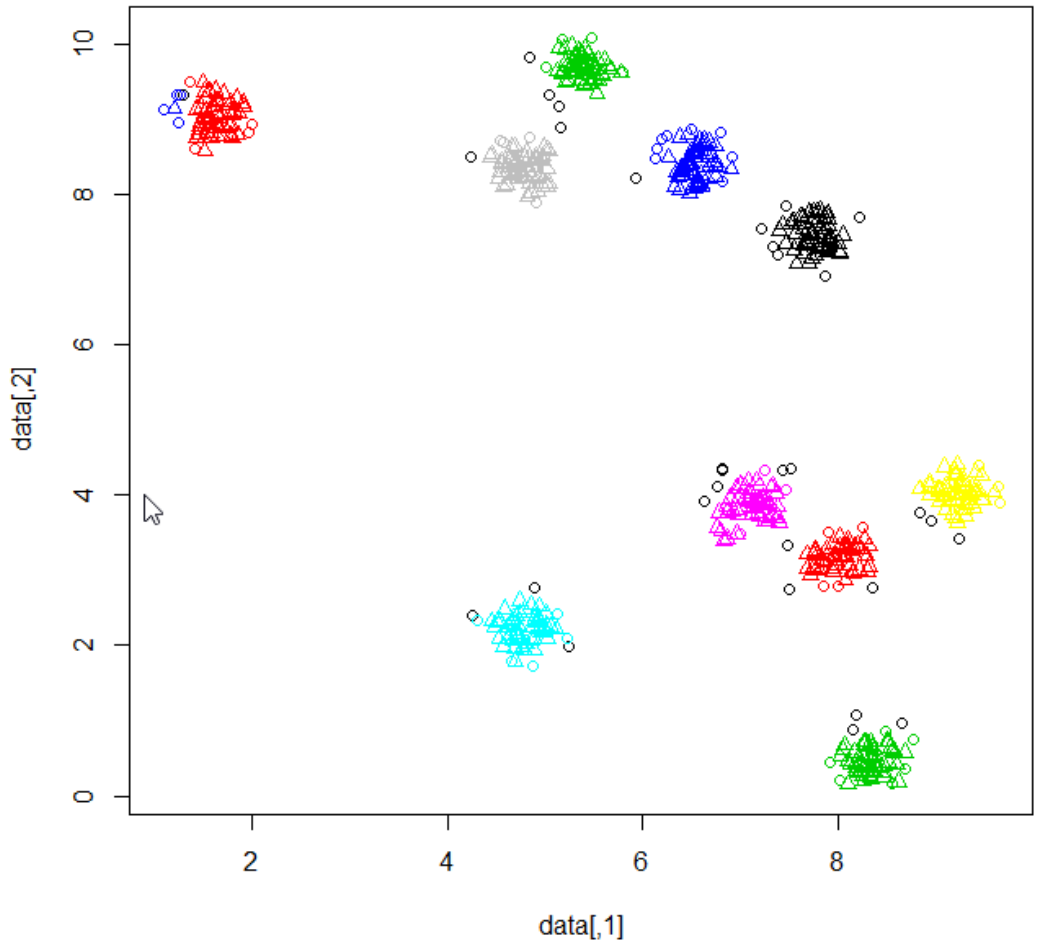


Szum (ang. noise) – jest to podzbiór wszystkich punktów z bazy danych nie należący do żadnej znalezionej grupy.

PRZYKŁADY - DBSCAN

```
# STWORZENIE PRZYKŁADOWEGO ZBIORU DANYCH
set.seed(665544)
n <- 600 x <- cbind(runif(10, 0, 10) + rnorm(n, sd=0.2),
runif(10, 0, 10) + rnorm(n, sd=0.2))
View(x)
plot (x)

#URUCHOMIENIE FUNKCJI DBSCAN
ds <- dbscan(x, eps = 0.2)

# WIZUALIZACJA
plot(ds, x)

#WYWOŁANIE INFORMACJI O KLASTRACH
ds
```

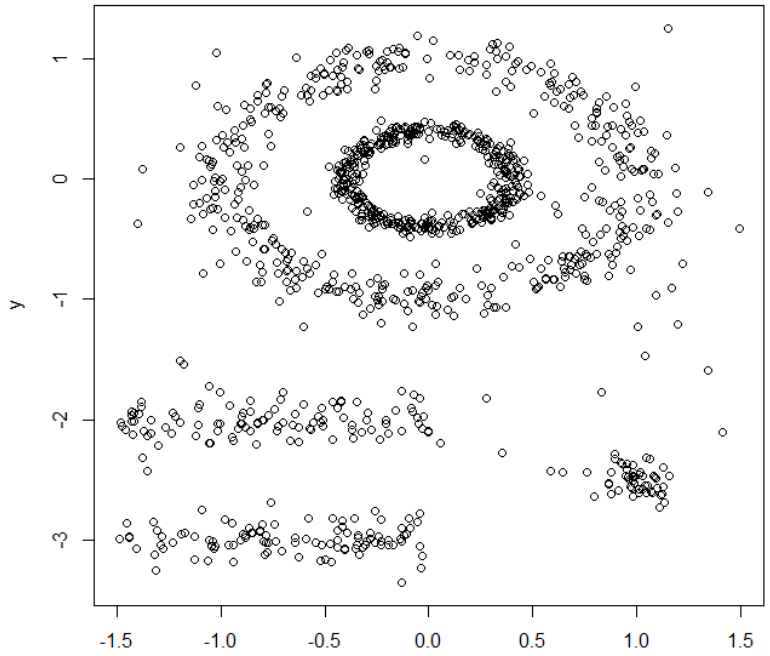
```
dbscan Pts=600 MinPts=5 eps=0.2
         0  1  2  3  4  5  6  7  8  9 10 11
border  28  4  4  8  5  3  3  4  3  4  6  4
seed     0 50 53 51 52 51 54 54 54 53 51  1
total   28 54 57 59 57 54 57 58 57 57 57  5
```

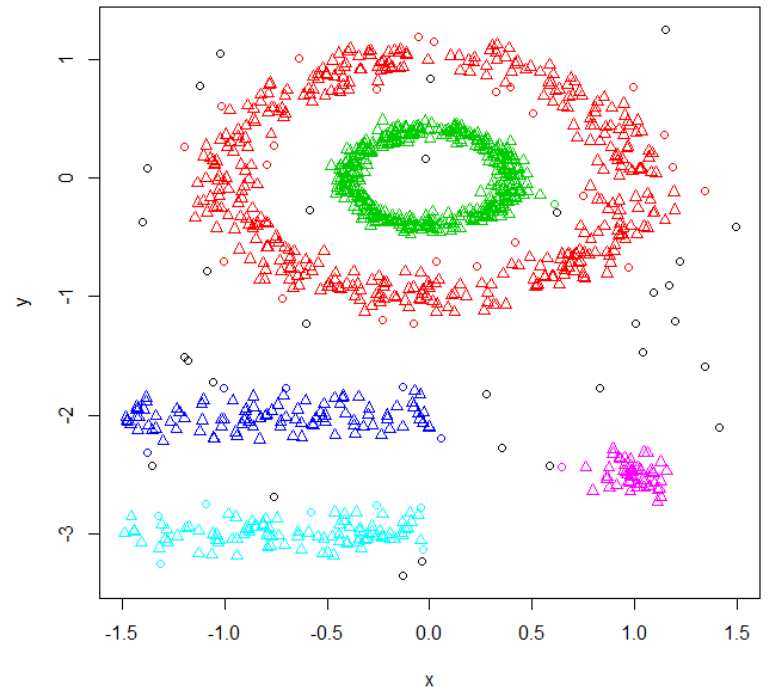


PRZYKŁADY - DBSCAN

```
# WCZYTANIE PRZYKŁADOWEGO ZBIORU DANYCH (BIBLIOTEKA FACTOREXTRA)
data("multishapes", package = "factoextra")
dane <- multishapes[, 1:2]
plot(dane)

#URUCHOMIENIE FUNKCJI DBSCAN
db <-  dbscan(dane, eps = 0.15, MinPts = 5)

# WIZUALIZACJA
plot(db, dane)

#WYWOŁANIE INFORMACJI O KLASTRACH
db
```

```
dbscan Pts=1100 MinPts=5 eps=0.15
        0   1   2   3  4  5
border 31  24   1   5  7  1
seed    0 386 404  99 92 50
total  31 410 405 104 99 51
```





A CO Z USTALENIEM OPTYMALNEGO EPS?

USTALENIE OPTYMALNEGO PARAMETRU EPS

Metoda polega na obliczeniu odległości kNN w macierzy punktów.

Wartość k jest określona przez użytkownika i odpowiada MinPts.

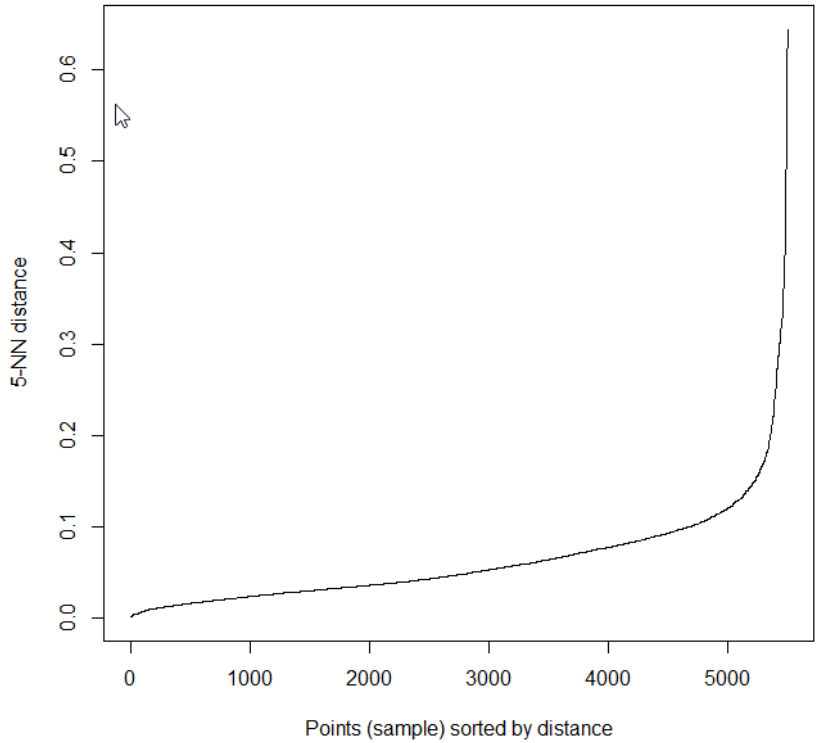
Następnie te odległości k są nanoszone w kolejności rosnącej na wykres. Celem jest określenie "kolana", które odpowiada optymalnemu parametrowi eps.

Kolano odpowiada progowi, w którym następuje ostra zmiana wzdłuż krzywej k-odległości.

Funkcji kNNdistplot () można użyć do narysowania wykresu odległości k:

➢kNNdist - *Calculate and plot the k-Nearest Neighbor Distance* – pozwala na szybkie wyznaczenie odległości k-najbliższych sąsiadów w macierzy punktów. Wykres stworzony na bazie funkcji może być używany do pomóc w znalezieniu odpowiedniej wartości parametru eps dla funkcji DBSCAN (szukanie „kolana” w wykresie).

```
kNNdistplot(dane, k = 5)
```



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

AGH

Dziękuję za uwagę!

Mgr inż. Agnieszka Ochałek
Narzędzia informatyczne w badaniach naukowych
Wydział Geodezji Górniczej i Inżynierii Środowiska
Katedra Ochrony Terenów Górniczych, Geoinformacji i Geodezji Górniczej

Kraków 12.01.2018 r.