

# TEORETYCZNE PODSTAWY INFORMATYKI

18/01/2015

WFAiS UJ, Informatyka Stosowana  
II stopień studiów

# Wykład 15b

2

Eksploracja  
danych

- Co rozumiemy pod pojęciem „eksploracja danych”
- Algorytmy klastrujące

# Graficzna reprezentacja danych

3

- Graficznie pokazujemy najistotniejsze charakterystyki danych
  - Tabela ze statystycznym podsumowaniem
  - Histogramy
  - Box-ploty
  - Scatter-ploty
- Mogą być wykonane bardzo szybko/łatwo, powinny pozwolić na pierwsze wrażenie „co jest w danych”
- Staramy się robić stosunkowo dużo rysunków, tabel sumarycznych w różny sposób grupując zmienne.

# Przykład

4

- Danie: zanieczyszczenie drobinkami materiałów powietrza na terenie USA.
  - <http://www.epa.gov/air/ecosystem.html>
- Dla poziomych zanieczyszczeń ustalona jest norma  $12\mu\text{g}/\text{m}^3$
- Zestawienie dzienne jest dostępne ze strony
  - <http://www.epa.gov/ttn/airs/airsaqs/detaildata/downloadaqdata.htm>
- Pytanie: czy są „stany” w USA w których ta norma jest przekraczana?

# Dane: (przykłady kodu w języku R)

5

Tu są dane za okres 2008-2010

```
pollution <- read.csv("data/avgpm25.csv", colClasses = c("numeric", "character",  
  "factor", "numeric", "numeric"))  
head(pollution)
```

```
##      pm25  fips region longitude latitude  
## 1  9.771 01003  east    -87.75    30.59  
## 2  9.994 01027  east    -85.84    33.27  
## 3 10.689 01033  east    -87.73    34.73  
## 4 11.337 01049  east    -85.80    34.46  
## 5 12.120 01055  east    -86.03    34.02  
## 6 10.828 01069  east    -85.35    31.19
```

Czy w jakiś stanach przekroczona jest ta norma?

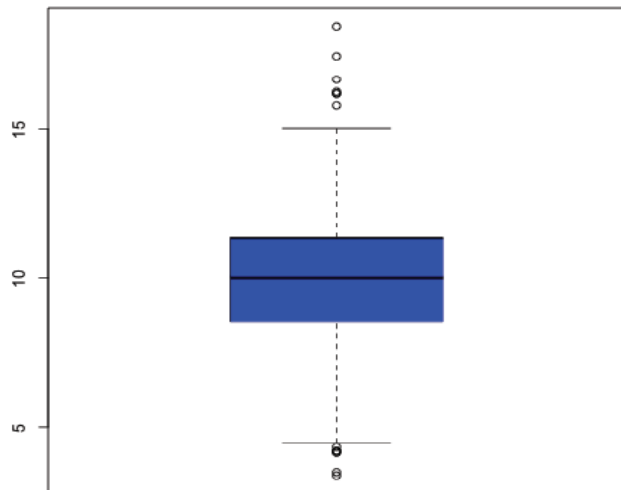
# Tabela summaryczna, box plot, histogram

6

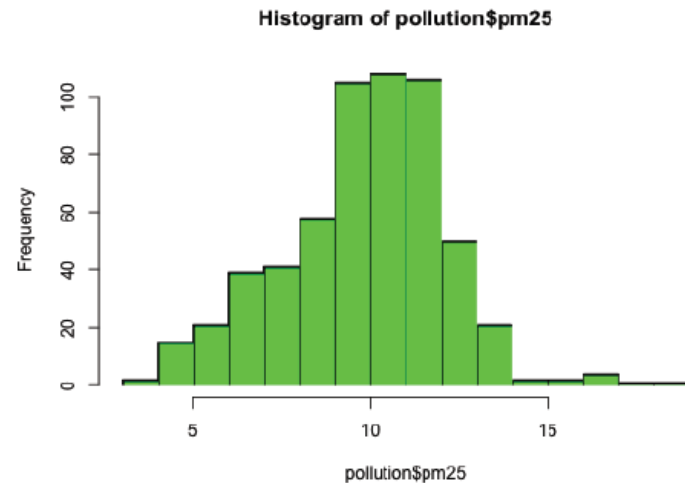
```
summary(pollution$pm25)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      3.38   8.55   10.00   9.84  11.40   18.40
```

```
boxplot(pollution$pm25, col = "blue")
```



```
hist(pollution$pm25, col = "green")
```

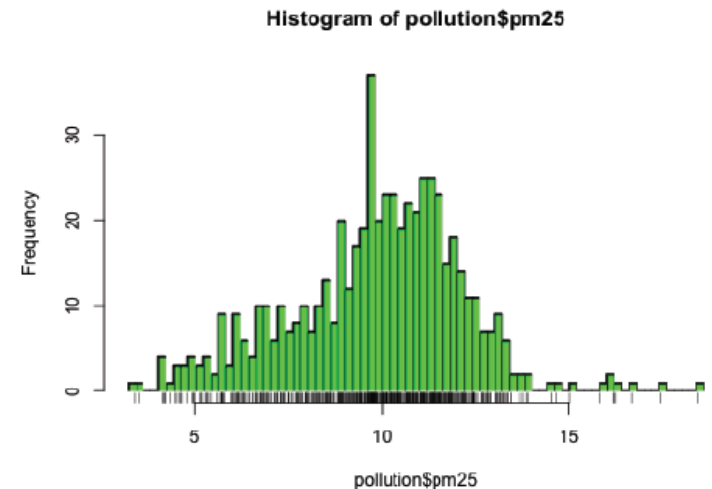
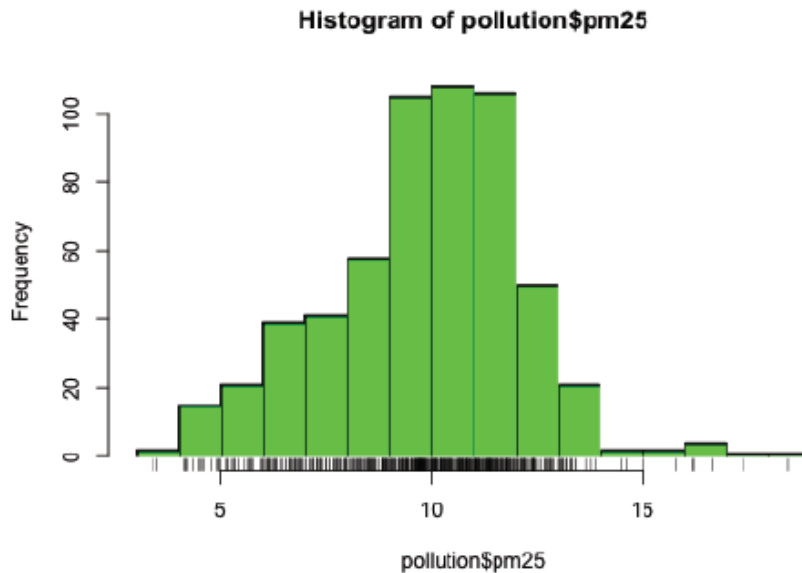


# Tabela summaryczna, box plot, histogram

7

```
hist(pollution$pm25, col = "green")  
rug(pollution$pm25)
```

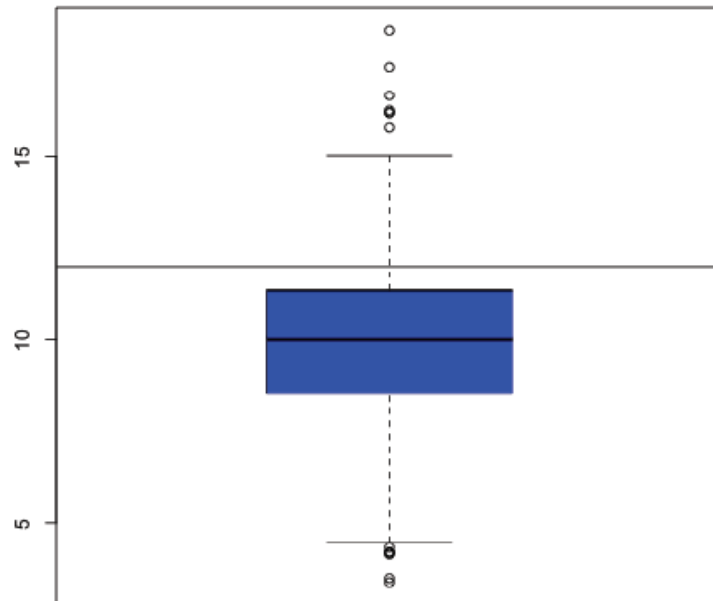
```
hist(pollution$pm25, col = "green", breaks = 100)  
rug(pollution$pm25)
```



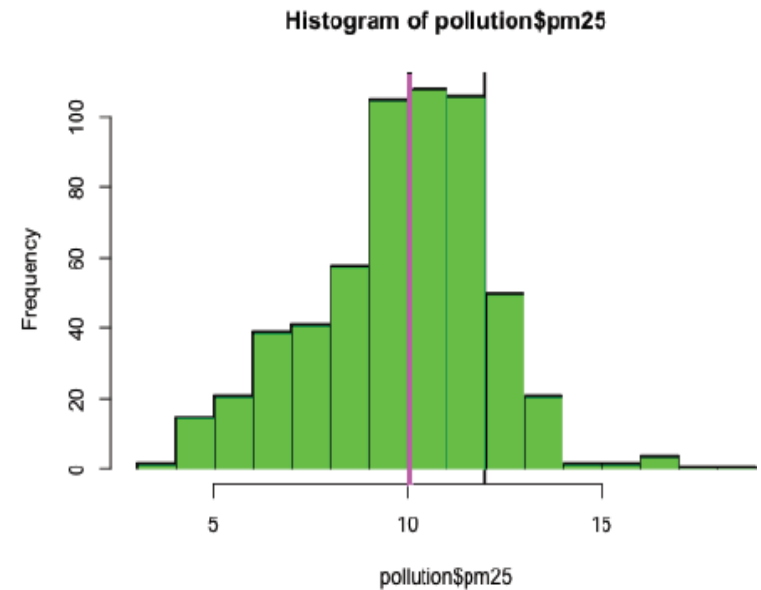
# Tabela summaryczna, box plot, histogram

8

```
boxplot(pollution$pm25, col = "blue")  
abline(h = 12)
```



```
hist(pollution$pm25, col = "green")  
abline(v = 12, lwd = 2)  
abline(v = median(pollution$pm25), col = "magenta", lwd = 4)
```

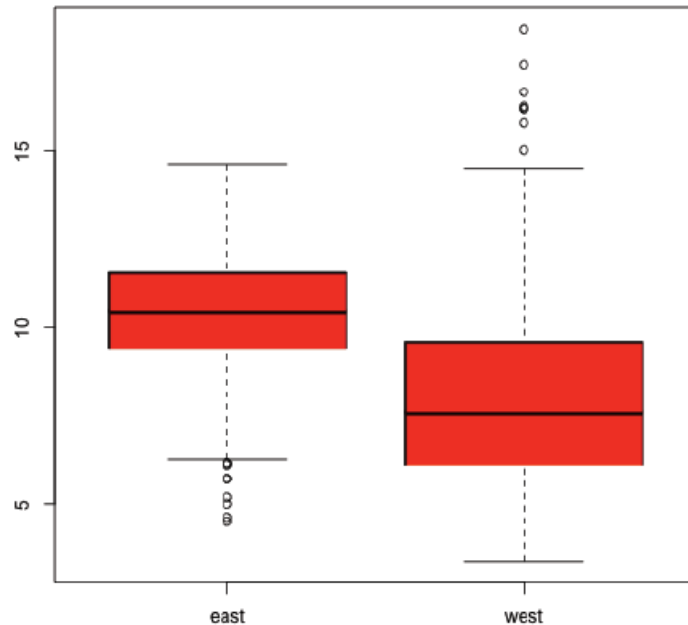




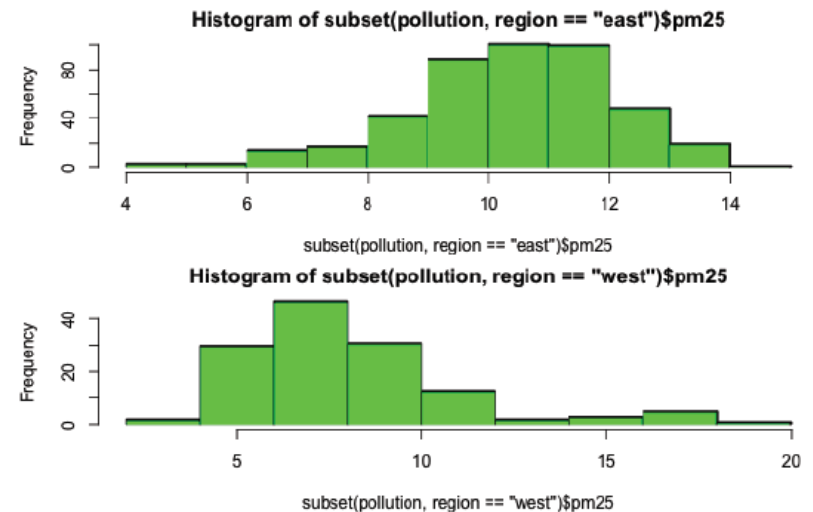
# Multi-box plot, multi-histograms

9

```
boxplot(pm25 ~ region, data = pollution, col = "red")
```



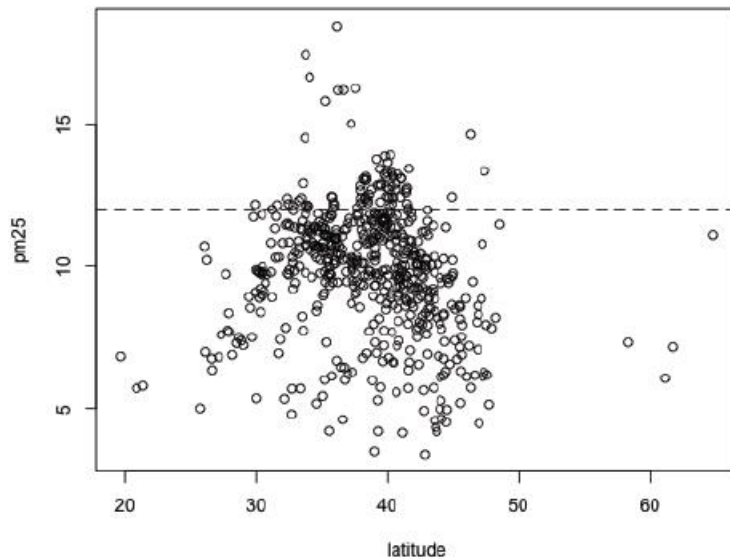
```
par(mfrow = c(2, 1), mar = c(4, 4, 2, 1))  
hist(subset(pollution, region == "east")$pm25, col = "green")  
hist(subset(pollution, region == "west")$pm25, col = "green")
```



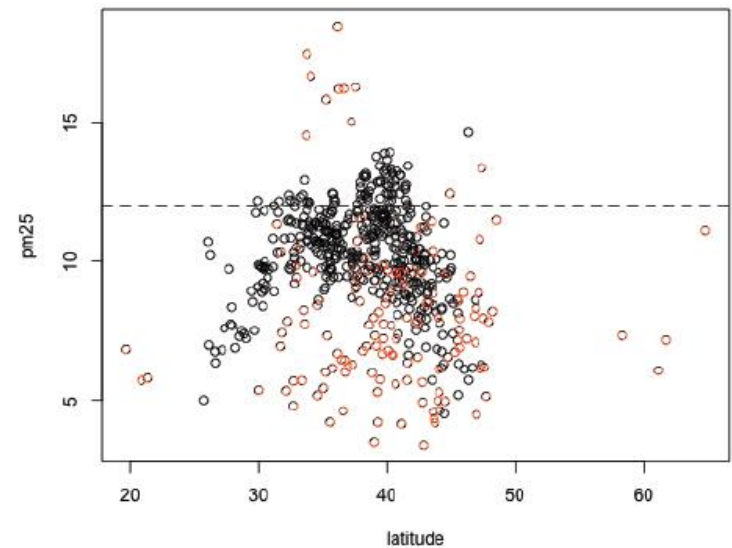
# Scatter-plot

10

```
with(pollution, plot(latitude, pm25))  
abline(h = 12, lwd = 2, lty = 2)
```



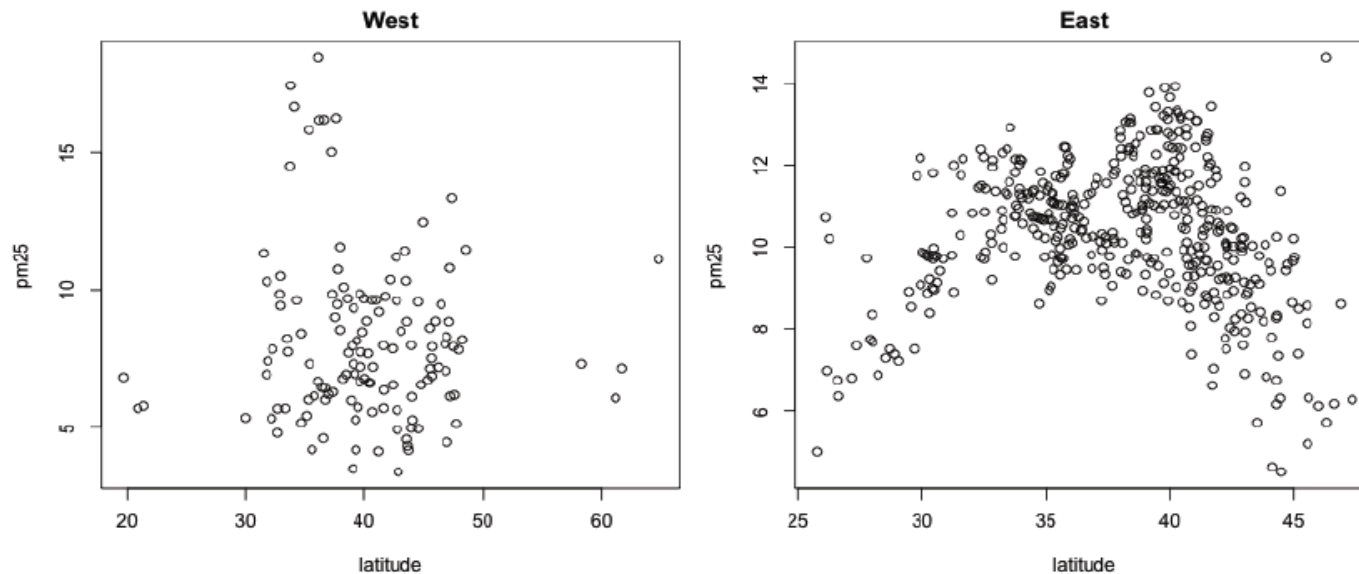
```
with(pollution, plot(latitude, pm25, col = region))  
abline(h = 12, lwd = 2, lty = 2)
```



# Multiple scatter plots

11

```
par(mfrow = c(1, 2), mar = c(5, 4, 2, 1))  
with(subset(pollution, region == "west"), plot(latitude, pm25, main = "West"))  
with(subset(pollution, region == "east"), plot(latitude, pm25, main = "East"))
```



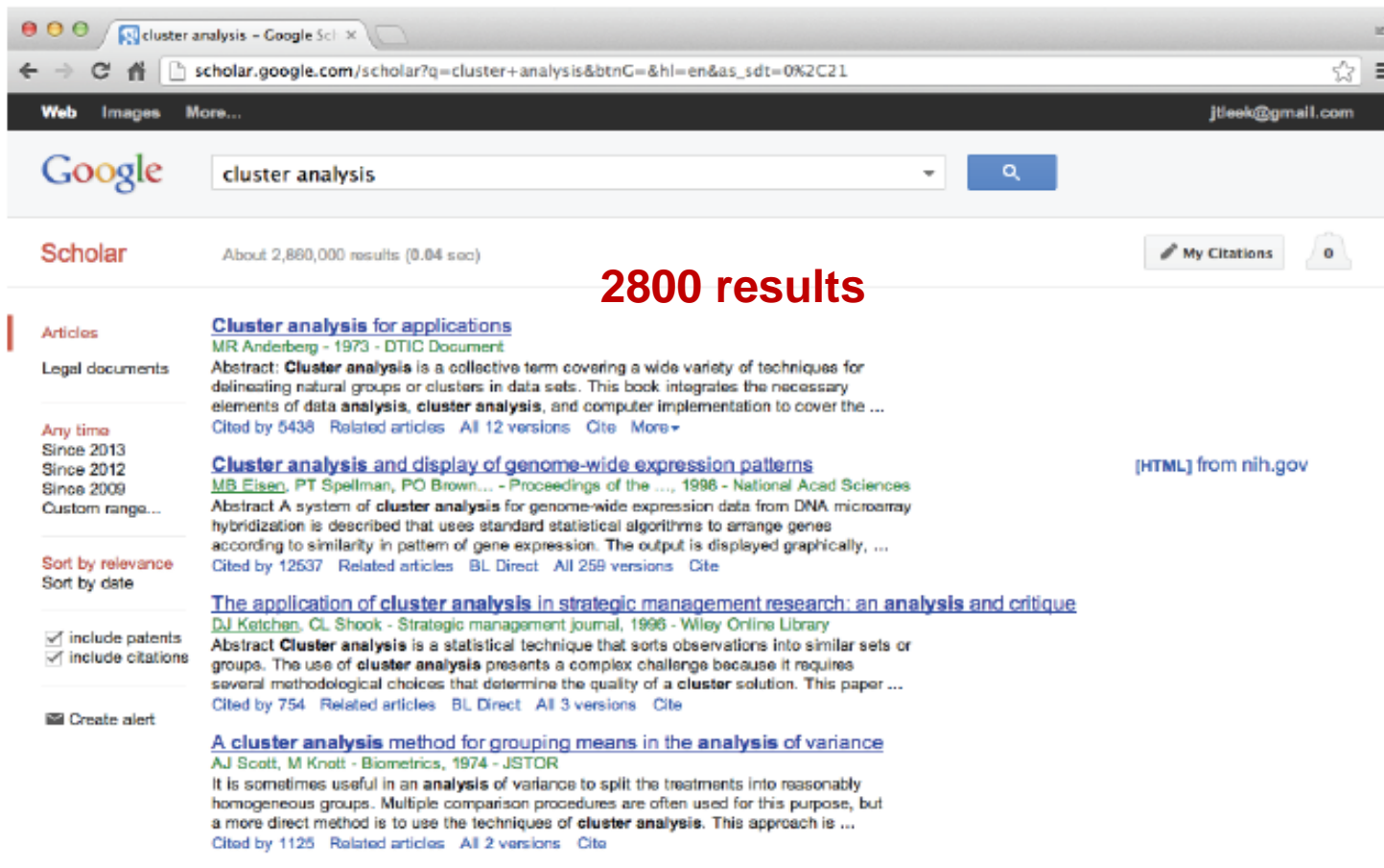
# Jak badać co się dzieje w danych które są w wielu wymiarach

12

- Klastrowanie organizuje dane które są „blisko” w pewne grupy czyli klastry.
  - Co to znaczy że dane są blisko
  - Co to znaczy że grupujemy?
  - Jak pokazać graficznie grupowanie?
  - Jak interpretować grupowanie?

# Klastrowanie jest bardzo ważną techniką

13



The screenshot shows a Google Scholar search interface. The search bar contains the text "cluster analysis". Below the search bar, the text "Scholar" is displayed, followed by "About 2,880,000 results (0.04 sec)". A red text overlay "2800 results" is positioned over the search results area. The search results are listed under the heading "Articles". The first result is "Cluster analysis for applications" by MR Anderberg, 1973, DTIC Document. The second result is "Cluster analysis and display of genome-wide expression patterns" by MR Eisen, PT Spellman, PO Brown, 1998, National Acad Sciences. The third result is "The application of cluster analysis in strategic management research: an analysis and critique" by DJ Ketchen, CL Shook, 1996, Wiley Online Library. The fourth result is "A cluster analysis method for grouping means in the analysis of variance" by AJ Scott, M Knott, 1974, JSTOR. The interface also includes a sidebar with filters for "Any time" (Since 2013, Since 2012, Since 2009, Custom range...), "Sort by relevance" and "Sort by date", and checkboxes for "include patents" and "include citations". A "Create alert" button is at the bottom of the sidebar. The top of the browser window shows the URL "scholar.google.com/scholar?q=cluster+analysis&btnG=&hl=en&as\_sdt=0%2C21".

# Hierarchiczne klastrowanie

14

- Aglomeracyjne podejście (bottom-up)
  - Znajdź dwa najbliższe położone punkty
  - Połącz je ze sobą w „super-punkt”
  - Znajdź kolejne najbliższe punkty (traktując już połączone jako jeden super-punkt)
- Parametry algorytmu
  - definicję odległości punktów
  - definicję „łączenia” punktów
- Wynik
  - Prezentujemy jako pewne „drzewo” (dendogram)

# Jak definiujemy odległość

15

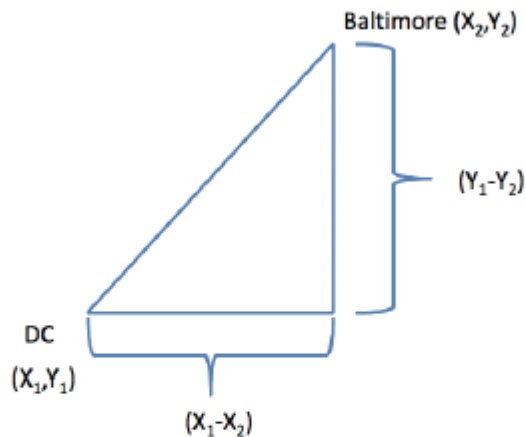
- Odległość:
  - Ciągła: euklidesowa metryka
  - Ciągła: stopień podobieństwa lub korelacji
  - Dyskretna: „Manhattan”
- Wybieramy taką definicję która stosuje się do naszych danych

# Jak definiujemy odległość

16

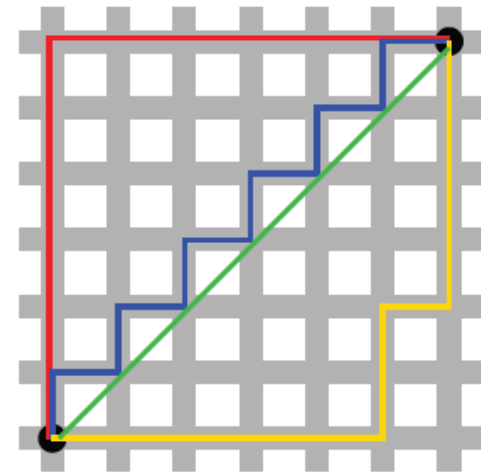
- Euklidesowa metryka vs Manhattan metryka

$$\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$



Naturalnie rozszerzalne do wielu wymiarów

$$\sqrt{(A_1 - A_2)^2 + (B_1 - B_2)^2 + \dots + (Z_1 - Z_2)^2}$$



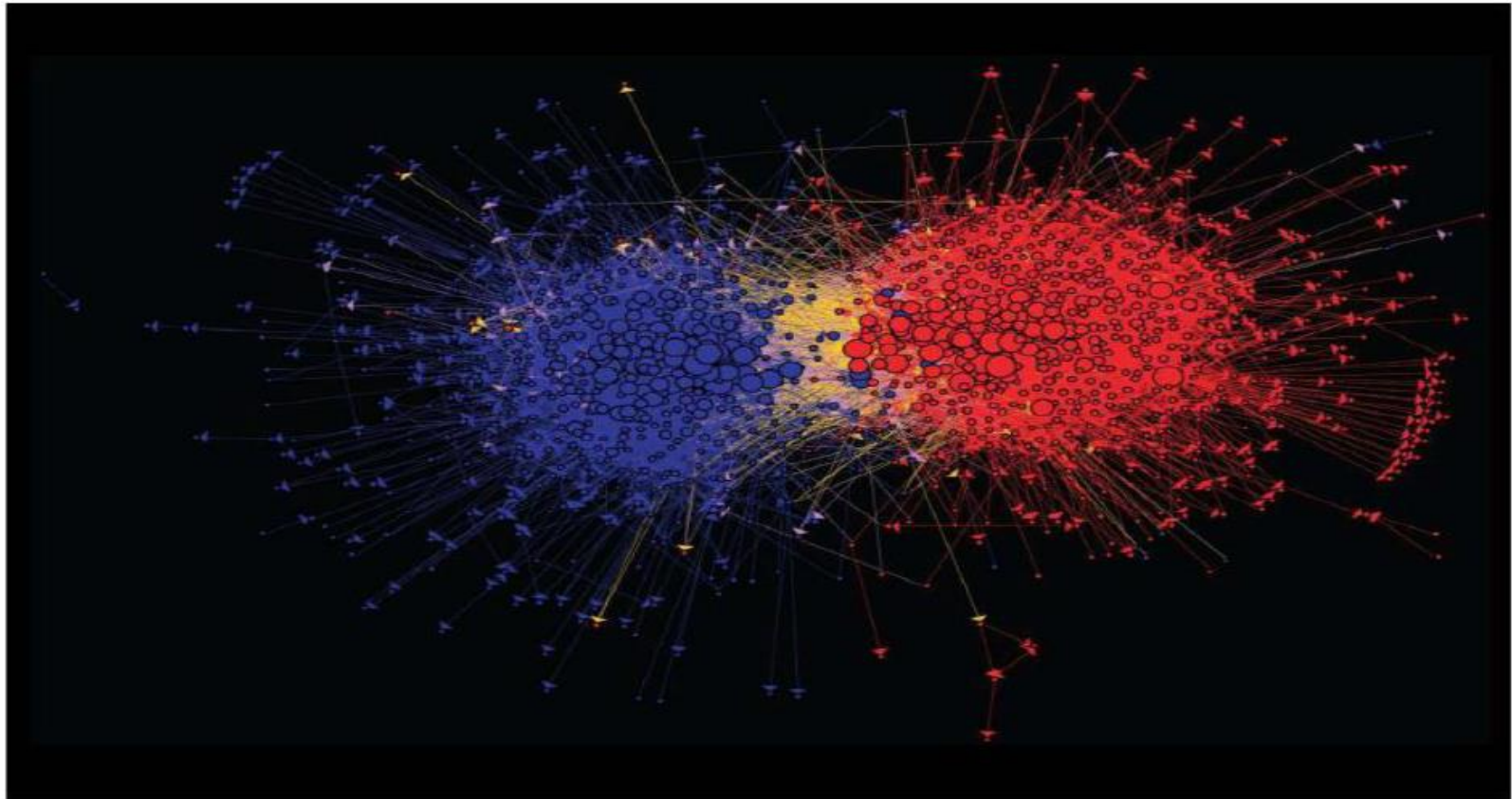
Musimy chodzić po ulicach

$$|A_1 - A_2| + |B_1 - B_2| + \dots + |Z_1 - Z_2|$$



# Przykład: media network

17



**Connections between political blogs**  
Polarization of the network [Adamic-Glance, 2005]



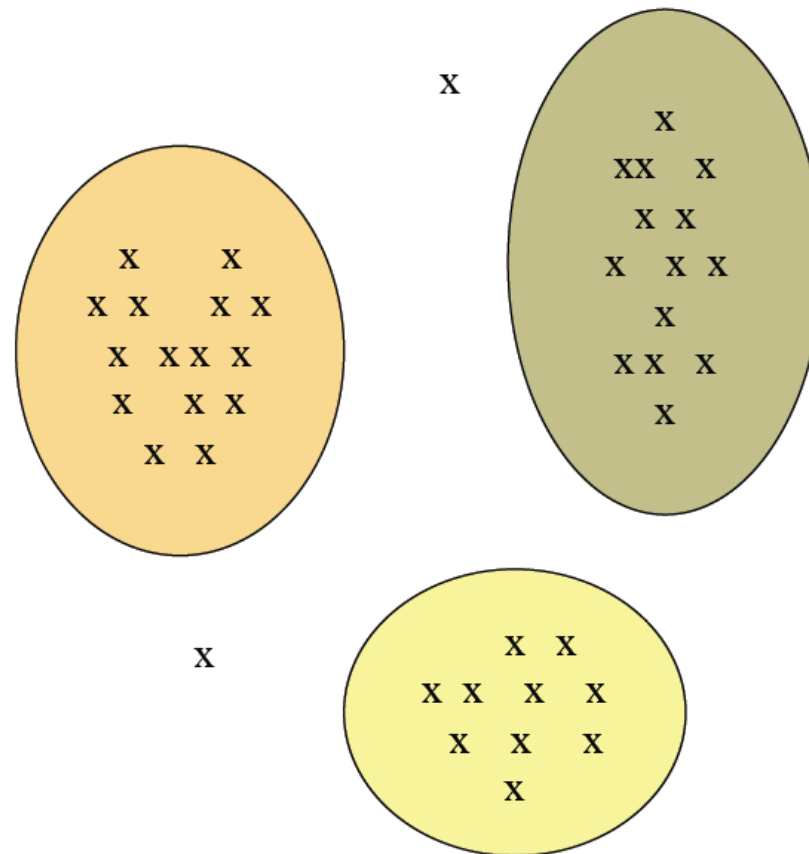
# Bardziej formalnie

19

- Mając dany zbiór punktów pomiarowych oraz podaną miarę odległości pomiędzy punktami pogrupuj dane w klastry
  - Punkty należące do tego samego klastra powinny być „podobne” do siebie
  - Punkty należące do dwóch różnych klastrów powinny się istotnie od siebie różnić
- Zazwyczaj:
  - Punkty są w przestrzeni wielowymiarowej
  - Podobieństwo jest zdefiniowane jako miara odległości pomiędzy punktami
    - Euklidesowa, Cosinus kąta, etc...

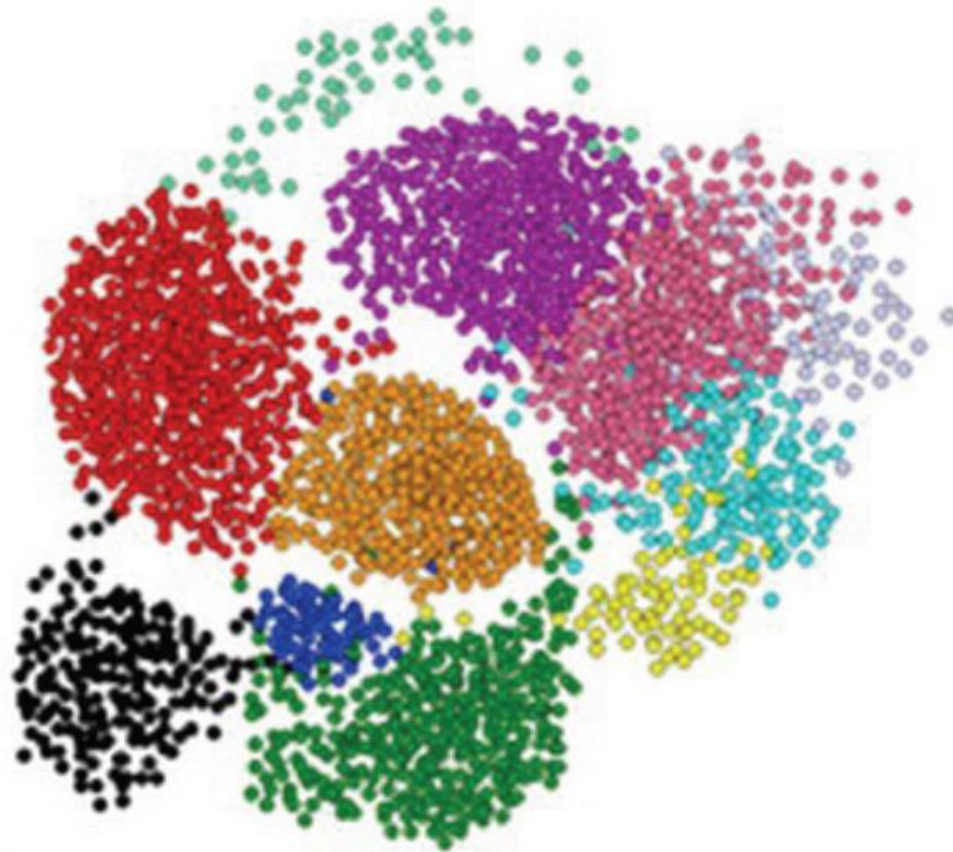
# Przykład klastrów

20



# Problem jest nietrywialny

21



# Gdzie jest trudność

22

- Klastrowanie w dwóch wymiarach jest na ogół łatwe
- Klastrowanie małej ilości danych jest na ogół łatwe
- Wiele praktycznych zastosowań dotyczy problemów nie 2 ale 10 lub 10000 wymiarowych.
- W dużej ilości wymiarów trudność polega na fakcie że większość danych jest w „tej samej odległości”.

# Przykład: klastowanie obiektów widocznych na niebie: SkyCat

23

- Katalog 2 bilionów „obektów” klastruje obiekty ze względu na częstości emisji promieniowania w 7 zakresach
- Klastrowanie powinno grupować obiekty tego samego typu, np. galaktyki, gwiazdy, kwazary, etc.

# Przykład: klastrowanie CD's

24

- Intuicyjnie: muzyka może być klasyfikowana w kilka kategorii i kupujący na ogół preferują pewne kategorie.
  - Na czym polegają te kategorie?
- A może klastrować CD's ze względu na kategorię osób które je kupują?
  - Podobne CD's mają podobną grupę kupujących i odwrotnie



# Przykład: klastrowanie dokumentów

25

- Problem: pogrupuj razem dokumenty na ten sam temat.
- Dokumenty z podobnym zestawem słów prawdopodobnie są na ten sam temat.
- A może inaczej sformułować: „temat” to podobny zestaw słów który występuje w wielu dokumentach. Więc może należy grupować słowa w klastry i te klastry będą definiować tematy?

# Miara odległości

26

- Dokument: zbiór słów
  - „Jaccard” odległość: podzbiór tych samych elementów
- Dokument: punkt w przestrzeni słów,  $(x_1, x_2, \dots, x_n)$ , gdzie  $x_i = 1$  jeżeli dane słowo występuje.
  - „Euklidesowa” odległość
- Dokument: vector w przestrzeni słów  $(x_1, x_2, \dots, x_n)$ .
  - „Cosinus” odległość: iloczyn skalarny unormowanych wektorów

# Metody klastrowania

27

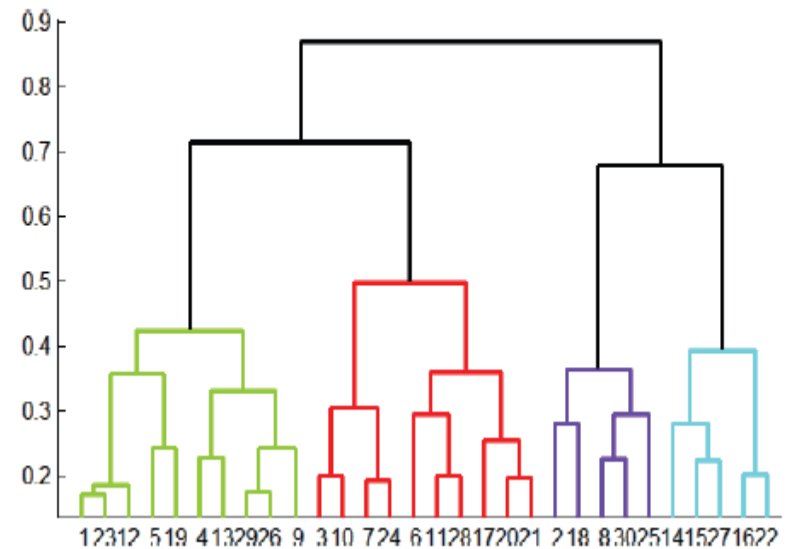
## □ Hierarchiczne:

### □ Bottom-up

- Początkowo każdy punkt jest klastrem
- Łączymy dwa klastry o najmniejszej odległości w jeden klaster
- Powtarzamy iteracyjnie

### □ Top-down

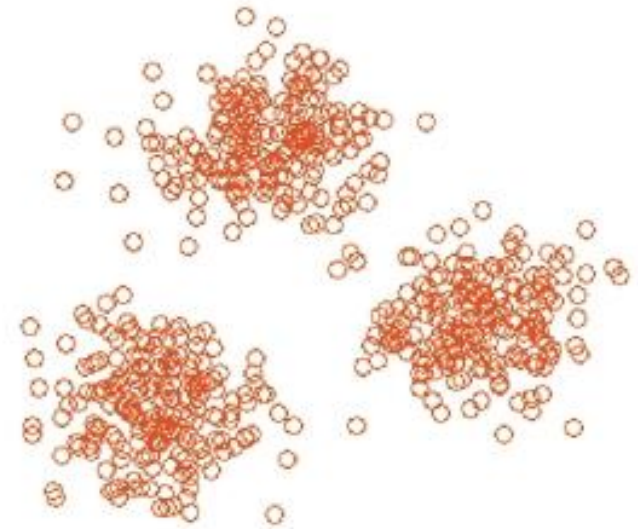
- Startujemy z jednego wielkiego klastra
- Dzielimy na dwa klastry jeżeli spełnione są pewne warunki
- Powtarzamy iteracyjnie aż osiągnięty inny warunek



# Metody klastrowania

28

- **K-means:**
  - Zakładamy na ile klastrów chcemy pogrupować dane.
  - Wybieramy początkowe centra klastrów.
  - Przeglądamy listę punktów, przypisujemy do najbliższego klastra.
  - Powtarzamy iteracyjnie po każdej iteracji poprawiając położenie centrów klastrów.



29

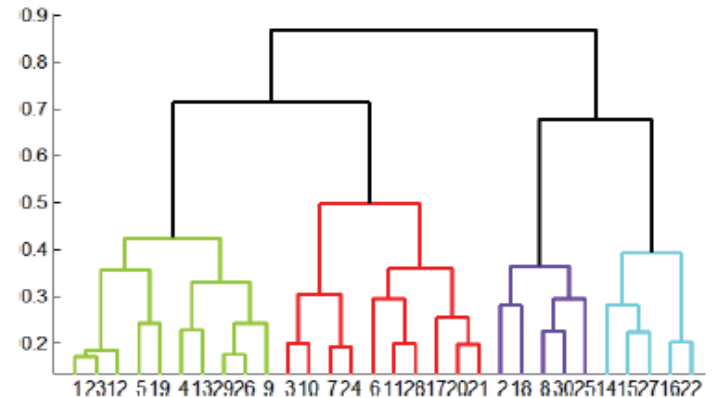
# Klastrowanie hierarchiczne

Bottom-up klastrowanie

# Klastrowanie hierarchiczne

30

- Podstawowa operacja: iteracyjnie powtarzają sklejanie dwóch klastrów w jeden.



- Podstawowe pytania:
  - ▣ Jak reprezentować klaster który zawiera więcej niż jeden punkt?
  - ▣ W jaki sposób zdefiniować „dwa najbliższe” klastry?
  - ▣ Kiedy zatrzymać procedurę sklejącą?

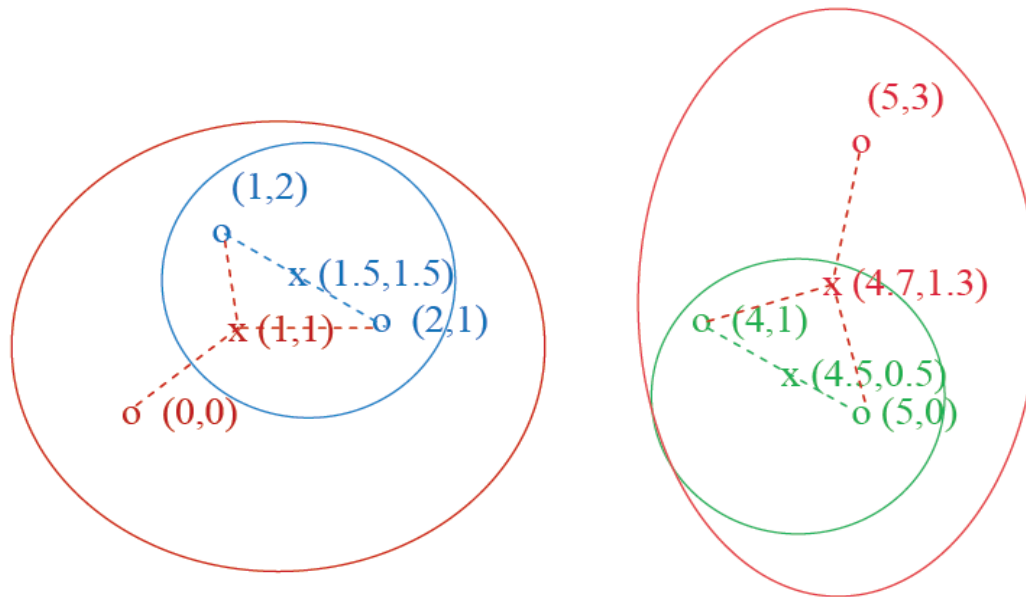
# Metryka Euklidesowa

31

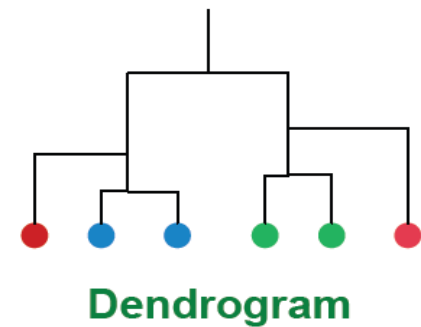
- Jak reprezentować klaster który zawiera więcej niż jeden punkt?
  - ▣ Reprezentujemy każdy klaster przez położenie jego środka ciężkości wg. wybranej metryki. Nazywamy go „centroid”, czyli centrum klastra nie jest jednym z punktów danych.
- W jaki sposób zdefiniować „dwa najbliższe” klastry?
  - ▣ Mierzymy odległość pomiędzy centrami klastrów, wybieramy najbliższe.

# Przykład:

32



**Data:**  
o ... data point  
x ... centroid





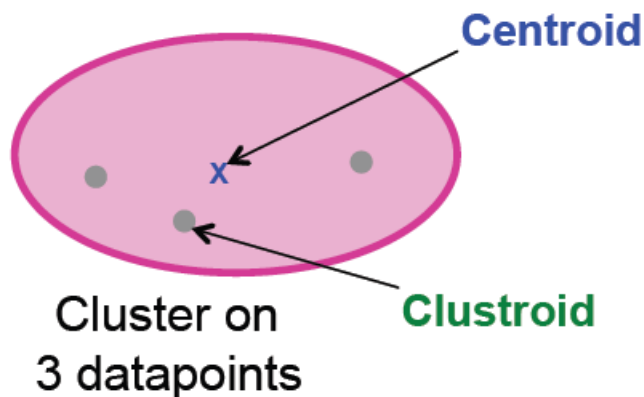
# Metryka nie-euklidesowa

33

- Jedyne położenia o których możemy mówić to punkty danych. Nie istnieje pojęcie „średniej”.
  - ▣ Klaster reprezentowany przez jego punkt będący najbliższym wszystkim innym punktom. Najbliższym wg. zadanej metryki. Nazywamy go „klastroid”.

# Metryka nie-euklidesowa

34

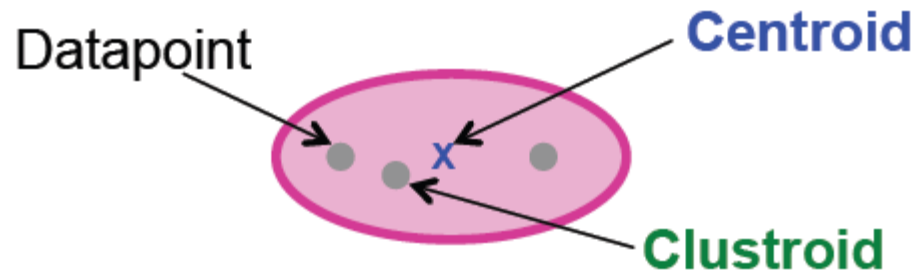


**Centroid** is the avg. of all (data)points in the cluster. This means centroid is an “artificial” point.

**Clustroid** is an **existing** (data)point that is “closest” to all other points in the cluster.

# Metryka nie-euklidesowa

35



- Klastroid: punkt najbliższy do wszystkich innych.
- Co to znaczy „najbliższy”?
  - ▣ Najmniejsza maksymalna odległość od wszystkich innych
  - ▣ Najmniejsza suma odległości od wszystkich innych
  - ▣ Najmniejsza suma kwadratów odległości od wszystkich innych.

# Kiedy zakończyć klastrowanie

36

- Jeżeli oczekujesz że dane powinny się grupować w  $k$  –klastrów zakończ jeżeli pogrupujesz w  $k$ -klastry
- Zatrzymaj jeżeli kolejne klastrowanie doprowadza do klastrów o gorszej jakości:
  - np. średnica (maksymalna odległość) większa niż wymagana granica
  - Np. promień klastra większy niż wymagana granica
  - Np. potęga promienia klastra większa niż wymagana granica.

# Implementacja

37

- Naiwna implementacja: w każdym kroku obliczaj odległość każdej pary klastrów a potem sklejjaj: złożoność  $O(N^3)$  dla każdej iteracji
- Optymalna implementacja z wykorzystaniem kolejki priorytetowej:  $O(N^2 \log N)$ 
  - Złożoność zbyt duża dla dużych zbiorów danych lub danych nie mieszczących się w pamięci.

38

# Klastrowanie k-means

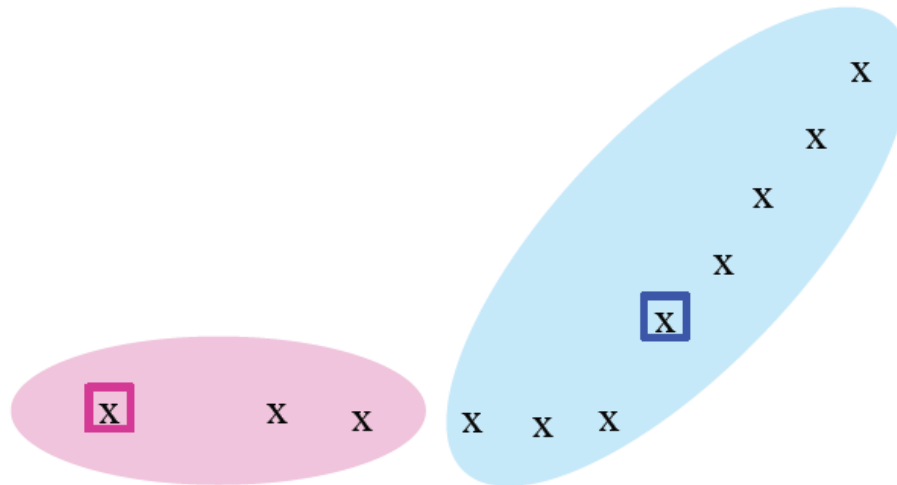
# K-means algorytm

39

- Zakłada metrykę euklidesową
- W pierwszym kroku wybieramy liczbę k-klastrów
  - Wybieramy k- punktów danych będących reprezentantami tych klastrów.
  - Na razie założmy że wybieramy je losowo.
- Przeglądamy listę punktów danych i przyporządkowujemy do klastrów
- Korygujemy położenie centrów klastrów wyliczając ich pozycję na podstawie punktów przypisanych do klastrów
- Powtarzamy operację przeglądania listy punktów, przypisujemy do najbliższych klastrów, punkty mogą migrować pomiędzy klastrami.
- Powtarzamy dopóki punkty nie przestają migrować

# Przykład: $k = 2$

40



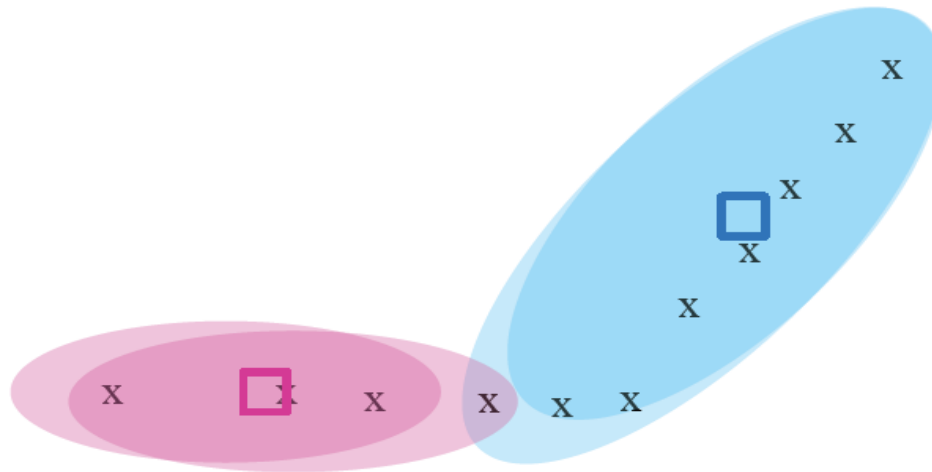
x ... data point  
□ ... centroid

**Round 1**



# Przykład: przyporządkuj punkty

41

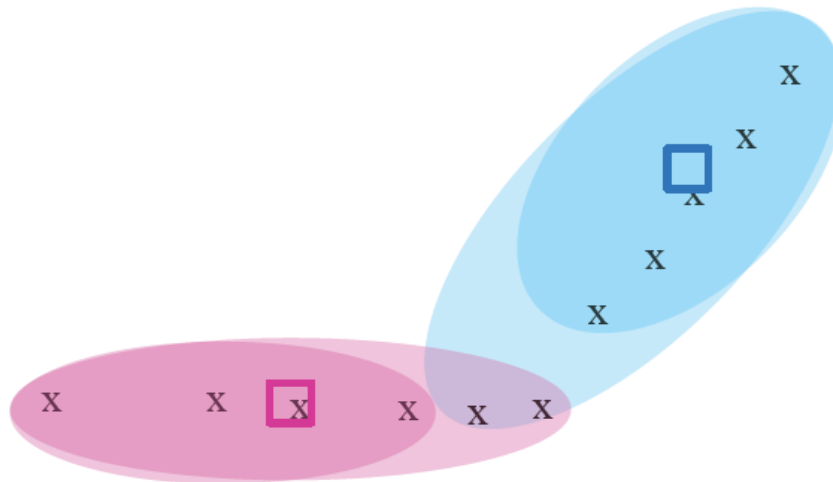


x ... data point  
□ ... centroid

**Round 2**

# Przykład: popraw centra

42



x ... data point

□ ... centroid

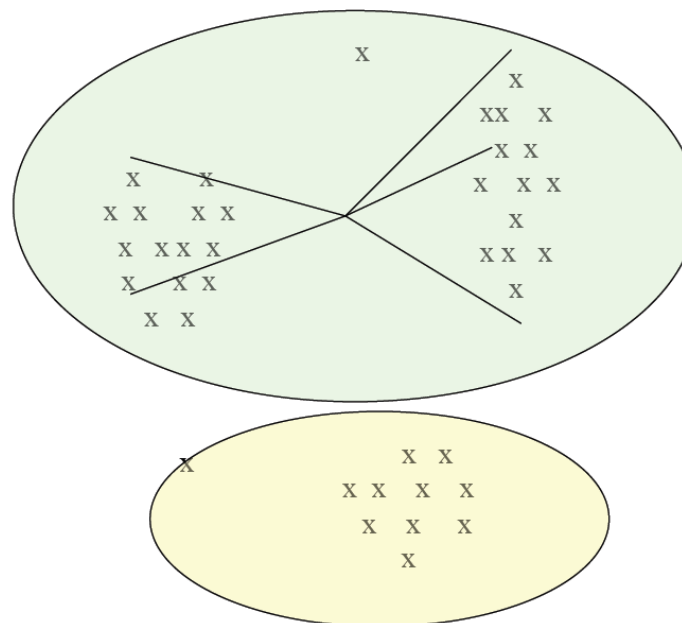
**Round 3**

# W jaki sposób wybrać ilość $k$ ?

43

- Spróbuj kilka różnych wartości, badaj jaka jest zmiana średniej odległości jak zwiększasz  $k$

Too few;  
many long  
distances  
to centroid.

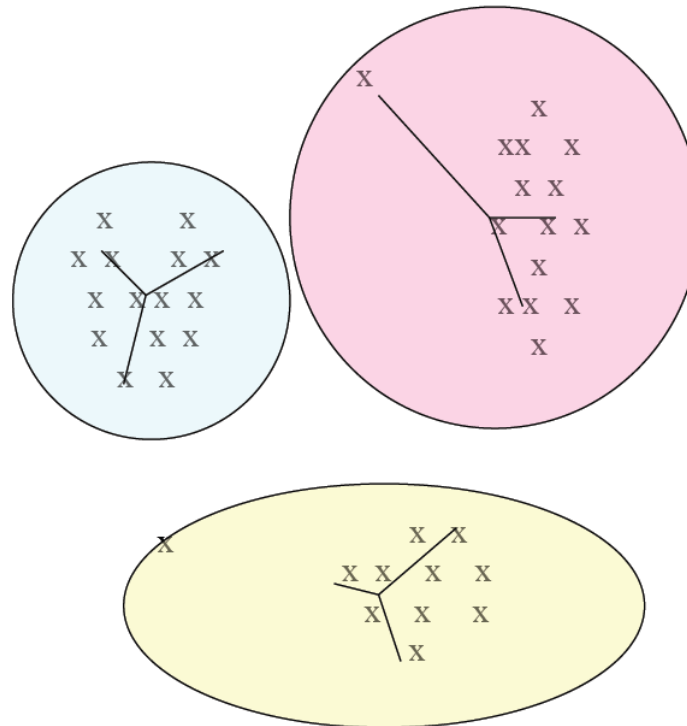


# W jaki sposób wybrać ilość $k$ ?

44

- Spróbuj kilka różnych wartości, badaj jaka jest zmiana średniej odległości jak zwiększasz  $k$

Just right;  
distances  
rather short.

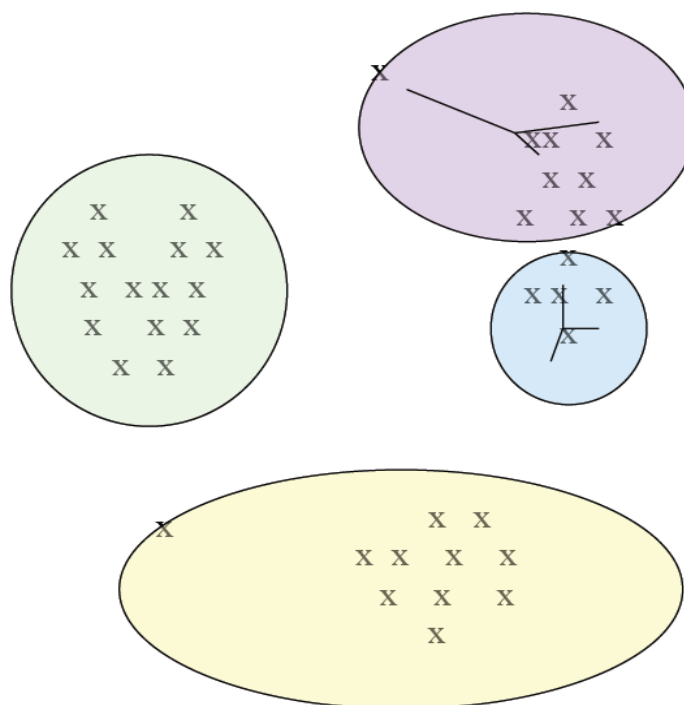


# W jaki sposób wybrać ilość $k$ ?

45

- Spróbuj kilka różnych wartości, badaj jaka jest zmiana średniej odległości jak zwiększasz  $k$

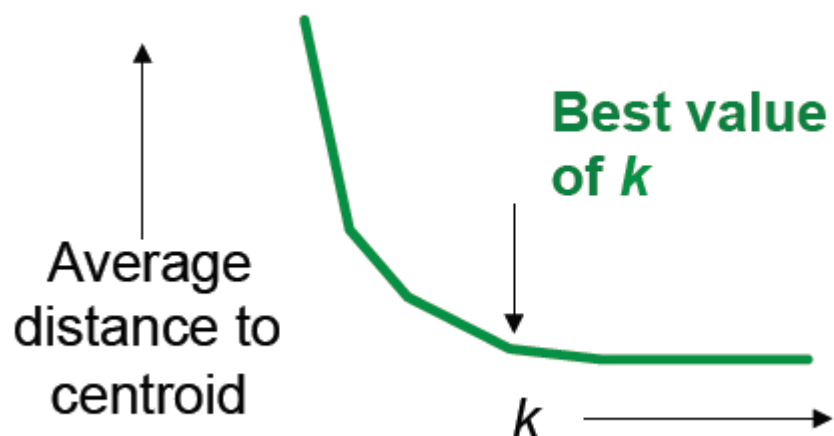
Too many;  
little improvement  
in average  
distance.



# W jaki sposób wybrać ilość $k$ ?

46

- Spróbuj kilka różnych wartości, badaj jaka jest zmiana średniej odległości jak zwiększasz  $k$



# W jaki sposób wybieramy startowe punkty?

47

- Na podstawie podzbioru danych:
  - Wybierz podzbiór danych, przeprowadź klastrowanie hierarchiczne aby otrzymać  $k$ -klastrów.
  - Wybierz dla każdego punkt najbliższe do środka klastra
  - Użyj tych punktów jako punktów startowych
- Najbardziej odległe punkty:
  - Wybierz pierwszy punkt losowo
  - Jako następny wybierz najbardziej od niego odległy
  - Powtarzaj zawsze wybierając najbardziej odległy od już wybranych punktów, aż wybierzesz  $k$ -punktów

# Złożoność obliczeniowa

48

- Za każdym razem przeglądamy pełną listę punktów aby przypisać punkt do jednego z  $k$ -centrów
- Każda iteracja dla  $N$  punktów i  $k$ -centrów to  $O(N k)$
- Liczba wymaganych iteracji może być bardzo duża.

**Czy moglibyśmy ten algorytm zrealizować tylko raz przeglądając dane?**



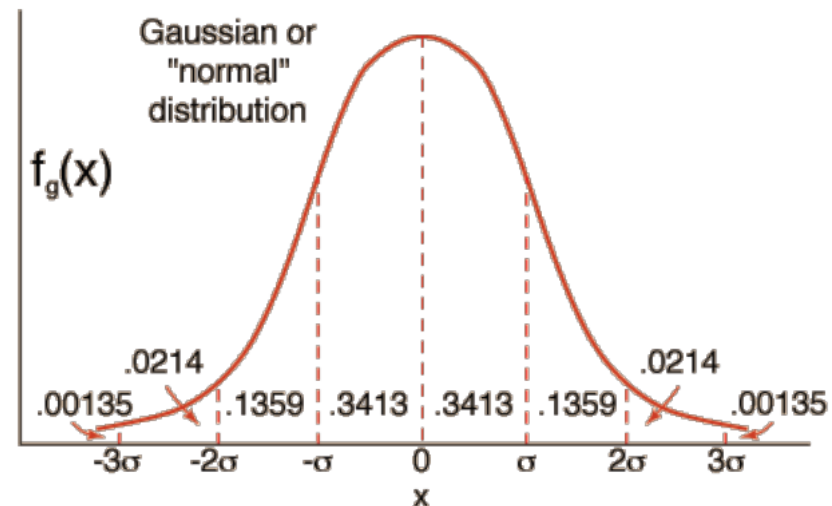
# Klastrowanie

Algorytm BFR (Bradley- Fayyard-Reina)

# Algorytm BFR

50

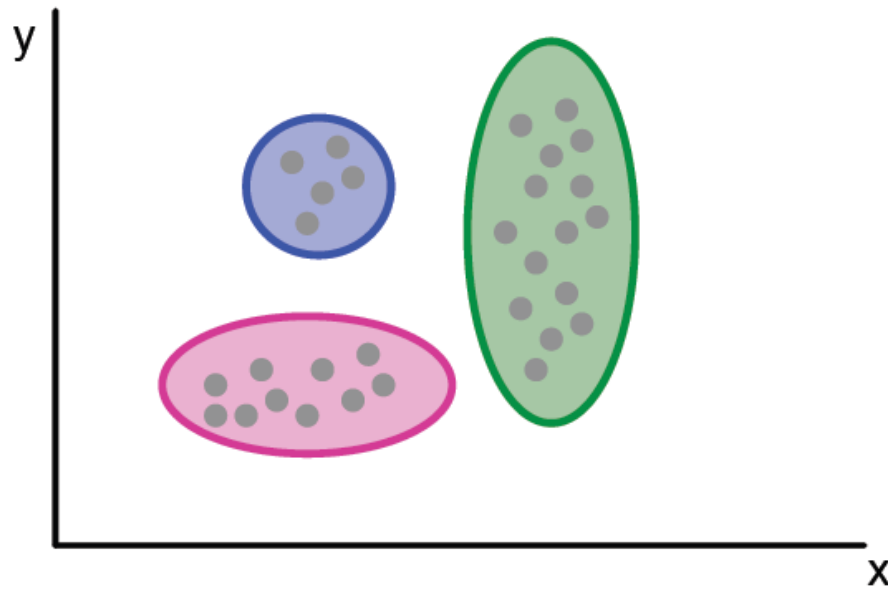
- To jest wersja algorytmu k-means dla bardzo dużych zbiorów danych.
- Zakłada euklidesowa metrykę.
- Zakłada że dane w klastrze mają rozkład normalny względem centrum klastra i każdego wymiaru



# BFR klastry

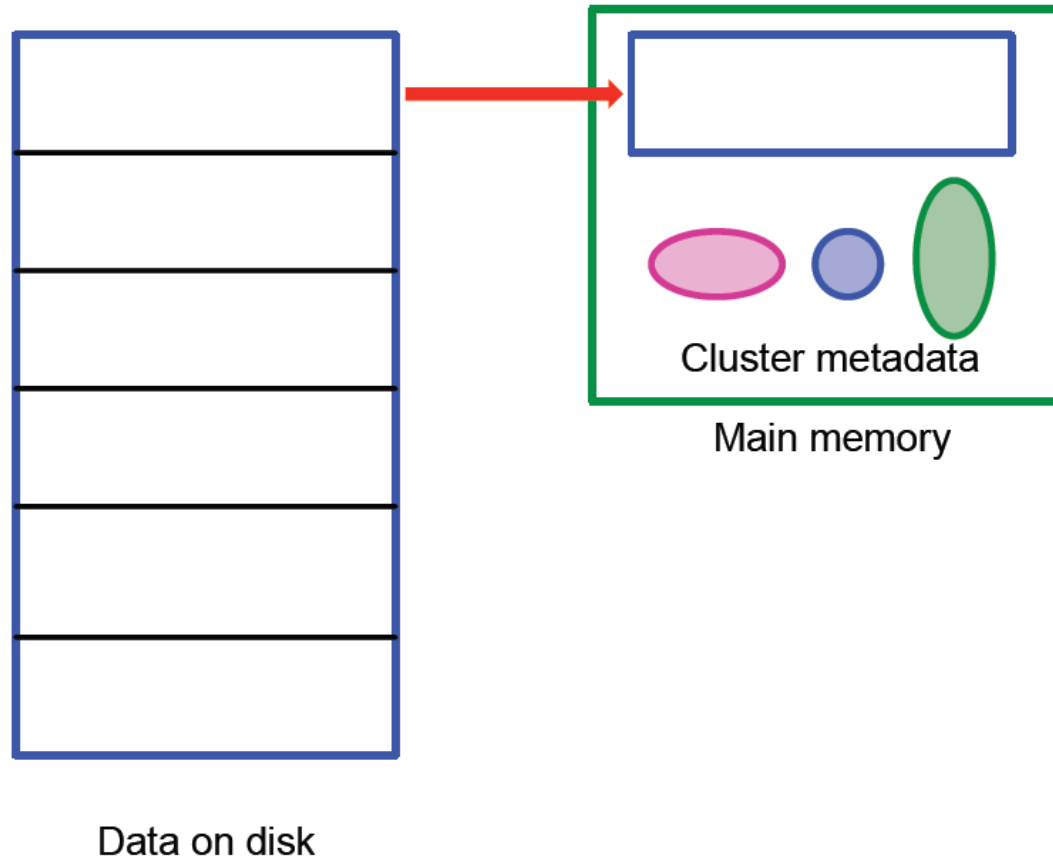
51

- Z założenia o normalności rozkładu wynika że klastry wyglądają jak elipsy o osiach równoległych do kierunków wymiarów.



# BFR algorytm

52



# BFR algorytm

53

- Punkty danych przeglądane są tylko raz, na raz w pamięci tylko podzbiór danych
- Informacja o większości z punktów z każdej partii przechowywana w postaci kilku sumarycznych wielkości statystycznych.
- Na początku, z pierwszej partii punktów wybieramy  $k$ -centrów wg. jednej z poznanych metod.

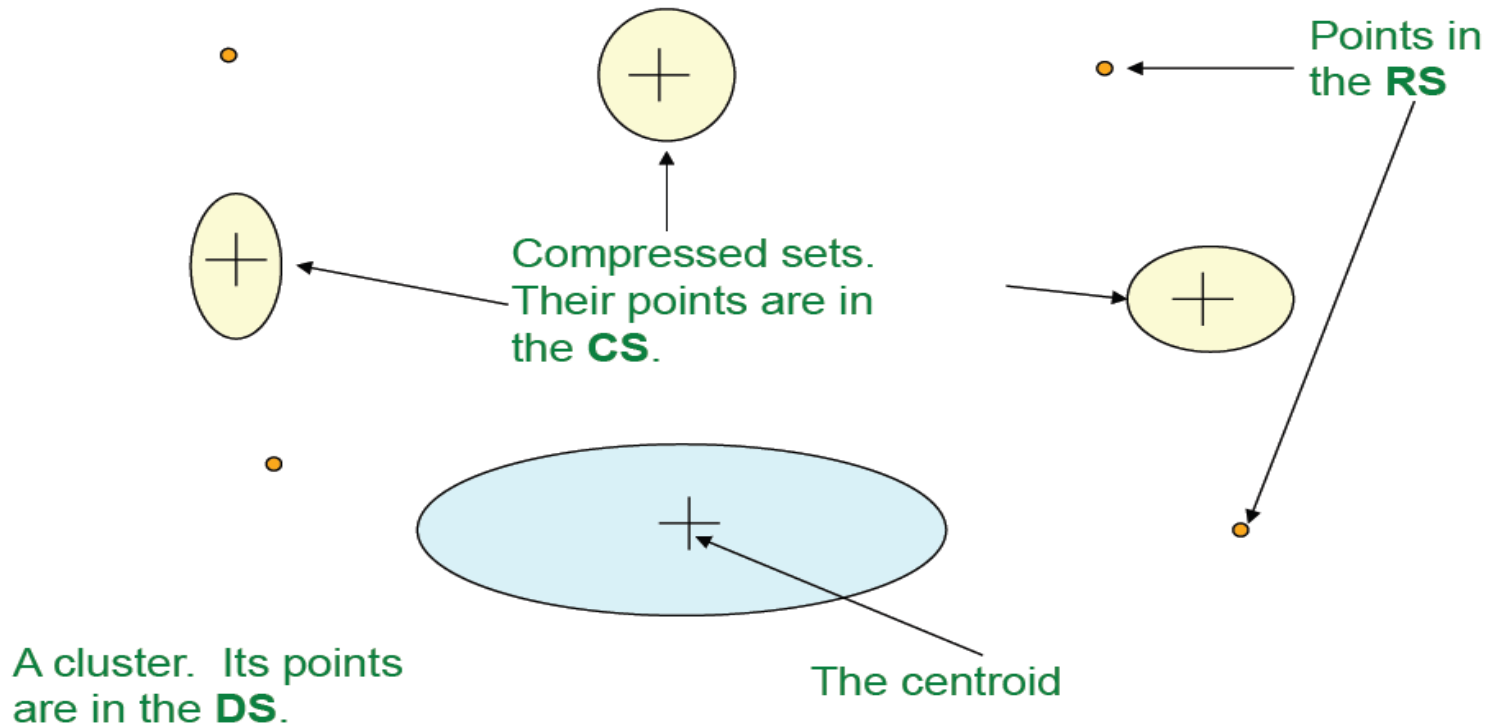
# Trzy klasy punktów

54

- Odrzucone punkty (discarded set DS): punkty będące blisko centrów klastrów tak że wystarczy zapisanie informacji sumarycznej
- Punkty w mini-klastrach (compression set CS): punkty które są blisko siebie ale nie wystarczająco blisko żadnego z centrów, zapisujemy tylko informację sumaryczną
- Punkty izolowane (retained set RS): przechowujemy izolowane punkty do następnego etapu.

# Trzy klasy punktów

55



**Discard set (DS):** Close enough to a centroid to be summarized  
**Compression set (CS):** Summarized, but not assigned to a cluster  
**Retained set (RS):** Isolated points

# Sumaryczna informacja

56

- Dla każdego klastra (DS) i każdego mini-klastra (CS) przechowywana jest informacja sumaryczna:
  - Liczba punktów  $N$
  - Wektor  $d$ -wymiarowy SUM: każda współrzędna to suma odległości punktów klastra od centrum w danym wymiarze
  - Wektor  $d$ -wymiarowy SUMSQ: każda współrzędna to suma kwadratów odległości punktów klastra od centrum w danym wymiarze



# Sumaryczna informacja

57

- $2d + 1$  wartości reprezentuje każdy klaster i mini-klaster
- Średnia w każdym wymiarze (centroid) może być przeliczona jako  $SUM_i/N$
- Wariancja w każdym wymiarze może być przeliczona jako  $(SUMSQ_i/N) - (SUM_i/N)^2$

# Procesowanie podzbioru danych

58

- Sprawdź czy punkt jest „wystarczająco blisko” do DS lub CS klastra, wybierz najbliższy, dodaj do sumarycznej informacji a następnie usuń punkt z pamięci.
- Jeżeli punkt nie był wystarczająco blisko sprawdź czy możesz utworzyć nowy CS klaster przeglądając RS punkty. Jeżeli nie, zapamiętaj ten punkt jako nowy RS punkt.
- Po analizie ostatniego podzbioru posklejaj wszystkie CS i RS do najbliższych DS. Ostatecznie utworzone zostanie tylko k klastrów.

# Mahalanobis odległość

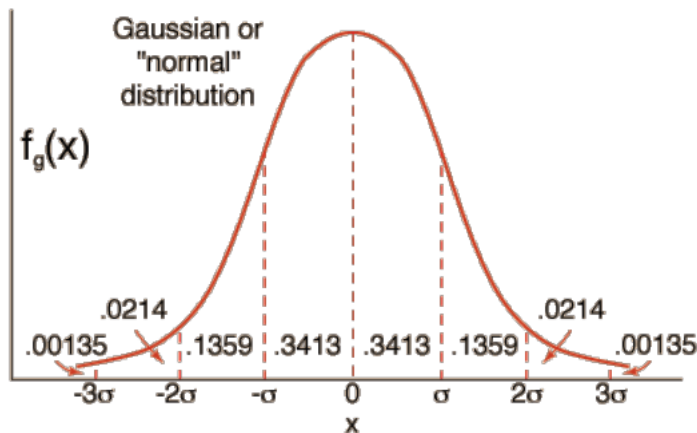
59

- Co to znaczy że punkt jest „wystarczająco blisko”?
  - Klaster C ma centroid w  $(c_1, c_2, \dots, c_d)$  i odchylenie standardowe  $(\sigma_1, \sigma_2, \dots, \sigma_d)$
  - Rozważany punkt  $P = (x_1, x_2, \dots, x_d)$
  - Znormalizowana odległość:  $y_i = (x_i - c_i) / \sigma_i$
- MD punktu P od klastra C: 
$$\sqrt{\sum_{i=1}^d y_i^2}$$

# Mahalanobis warunek

60

- Przypuśćmy że punkt P jest w odległości jednego odchylenia standardowego od centrum w każdym wymiarze.
- Każdy  $y_i = 1$  i wówczas  $MD = \sqrt{d}$



68% of points have  $MD \leq \sqrt{d}$

95% of points have  $MD \leq 2\sqrt{d}$

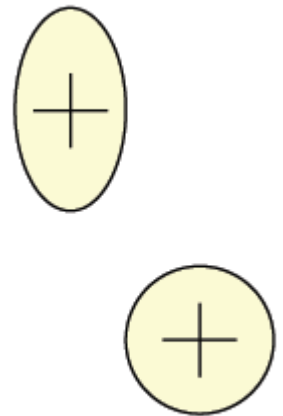
99% of points have  $MD \leq 3\sqrt{d}$

Akceptuj punkt P do klastra C jeżeli np. wartość  $MD < 3 \sqrt{d}$

# Kiedy sklejać dwa CS

61

- Policz wariancję dla sklejonych klastrów, sklej jeżeli wariancja poniżej wartości granicznej.
- Możliwe inne warunki:
  - ▣ Gęstość klastra
  - ▣ Odległości mogą mieć inną wagę dla każdej współrzędnej
  - ▣ Itd..



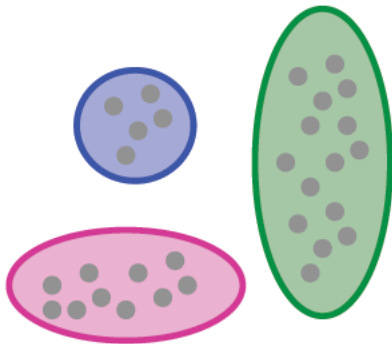
62

# Algorytm CURE

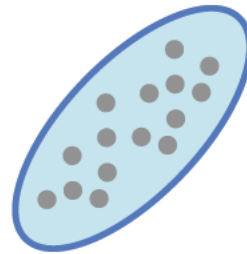
# Ograniczenia algorytmu BFR

63

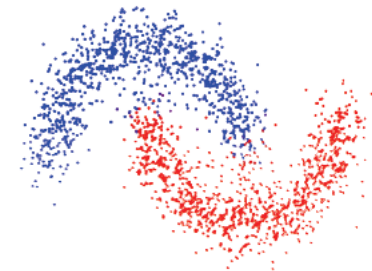
- Silne założenia:
  - ▣ Normalny rozkład punktów w każdym wymiarze
  - ▣ Osie wzdłuż osi współrzędnych



OK



Not OK

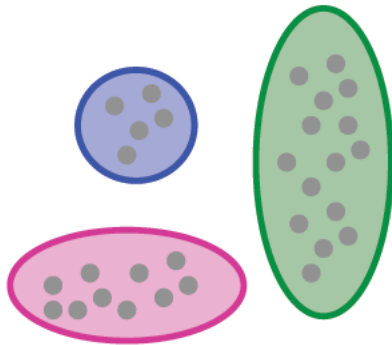


Not OK

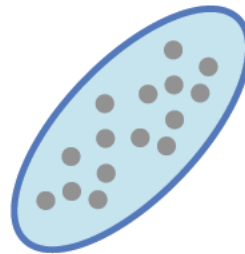
# Algorytm CURE

64

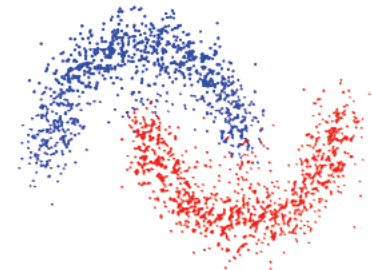
- CURE ( Clustering Using REpresentatives):
  - ▣ Zakłada metrykę Euklidesową
  - ▣ Dopuszcza klastry różnych kształtów
  - ▣ Używa podzbiór punktów do reprezentowania klastra



OK



OK

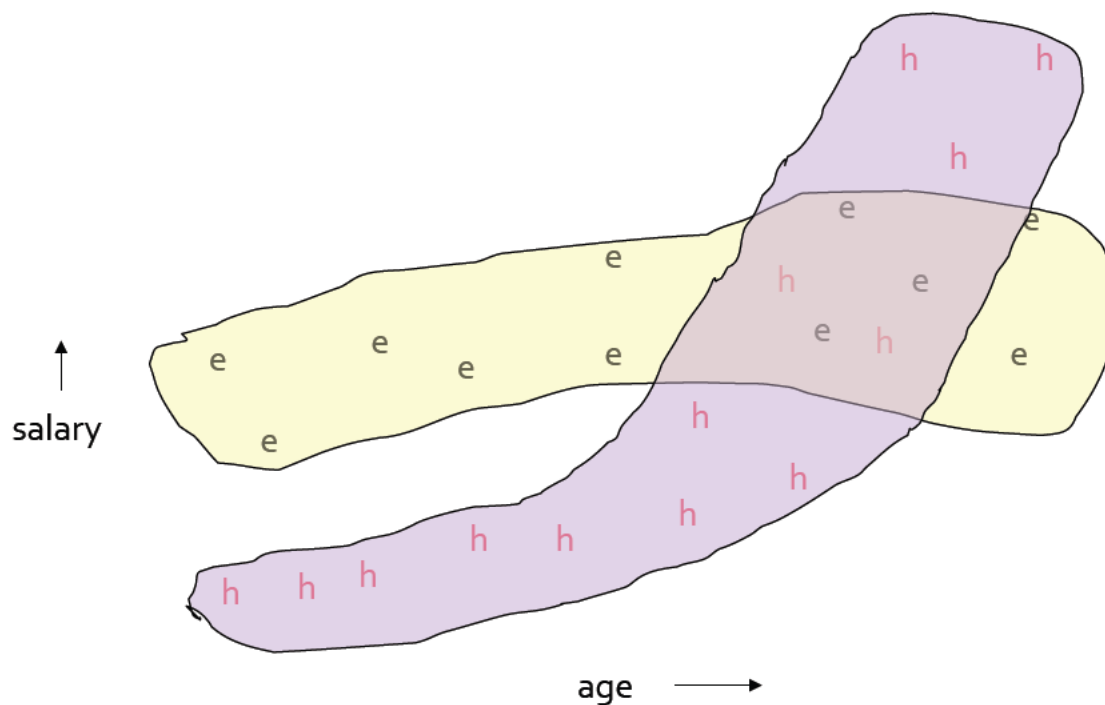


OK



# Przykład: płace w Stanford

65



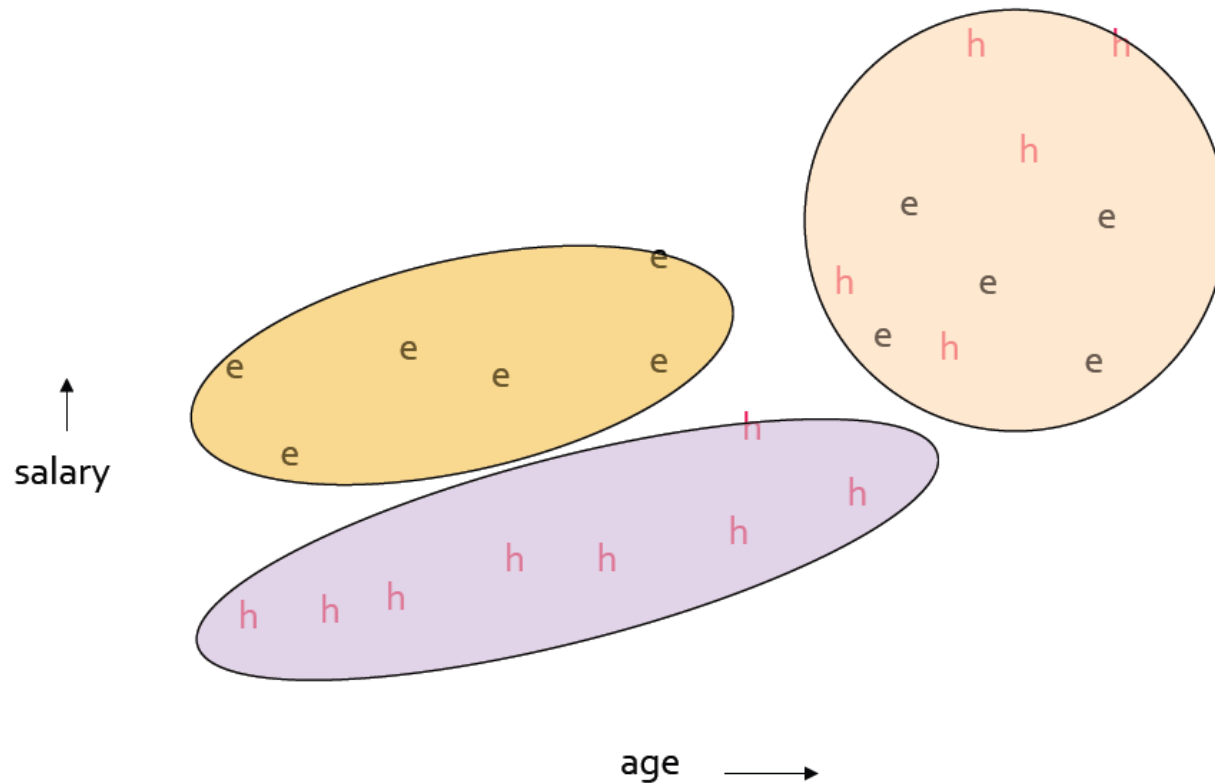
# Algorytm CURE

66

- Pierwsze przejście danych
  - Wybierz podzbiór danych który mieści się w pamięci
  - Przeprowadź klastrowanie hierarchiczne tego podzbioru danych.
  - Wybierz k-punkty reprezentujące klaster (np.  $k=4$ ), jak najbardziej od siebie odległe
  - Utwórz sztuczne punkty przez przesunięcie wybranych k punktów np. o 20% w stronę centrum klastra, to będą reprezentanci klastra

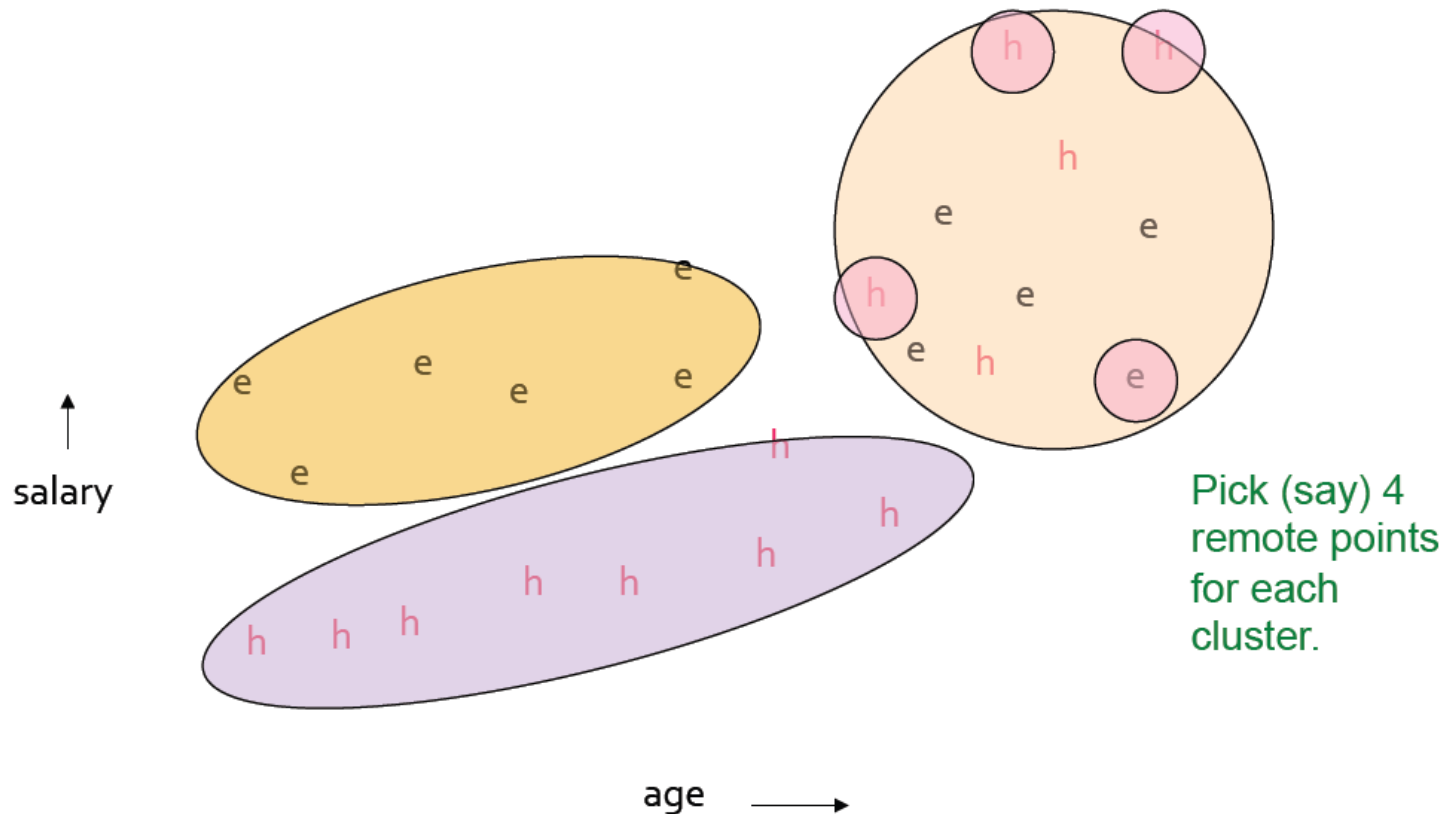
# Przykład: początkowe klastry

67



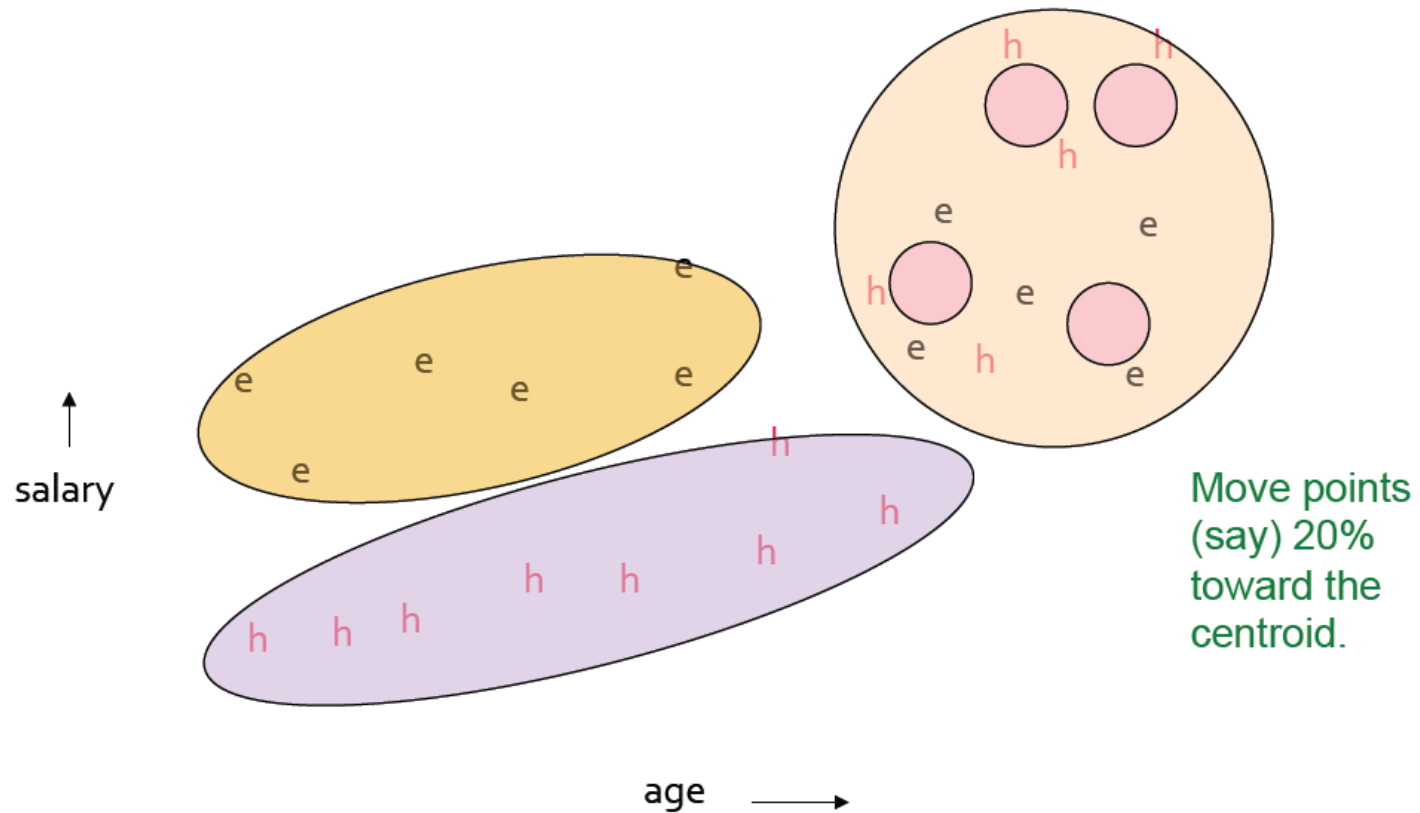
# Przykład: wybór reprezentatywnych punktów

68



# Przykład: wybór reprezentatywnych punktów

69



# Algorytm CURE

70

- Drugie przejrzanie danych
  - ▣ Teraz przejrzyj całość danych
  - ▣ Przypisz punkty w najbliższych klastrze: do określenia „najbliższy” użyj dla każdego klastra reprezentatywnych punktów

I to już jest koniec procesowania algorytmu!