ALGORYTMICZNA I STATYSTYCZNA ANALIZA DANYCH

27/11/2014

WFAiS UJ, Informatyka Stosowana
II stopień studiów

Eksploracja danych

Co to znaczy „eksploracja danych”

Klastrowanie (grupowanie) hierarchiczne

Klastrowanie (grupowanie) k-means

Graficzna reprezentacja danych

- 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 plotów, tabel sumarycznych w różny sposób grupując zmienne.

Przykład

- Danie: zanieczyszczenie drobinkami materiałów powietrza na terenie USA.
  - http://www.epa.gov/air/ecosystem.html
- Dla poziomu zanieczyszczeń ustalona jest norma 12μg/m3
- Zestawienie dzienne jest dostępne ze strony
  - http://www.epa.gov/ttn/airs/airsaqs/detaildata/downloadaqsdata.htm
- Pytanie: czy są stany w których ta norma jest przekraczana?

Dane:

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
```

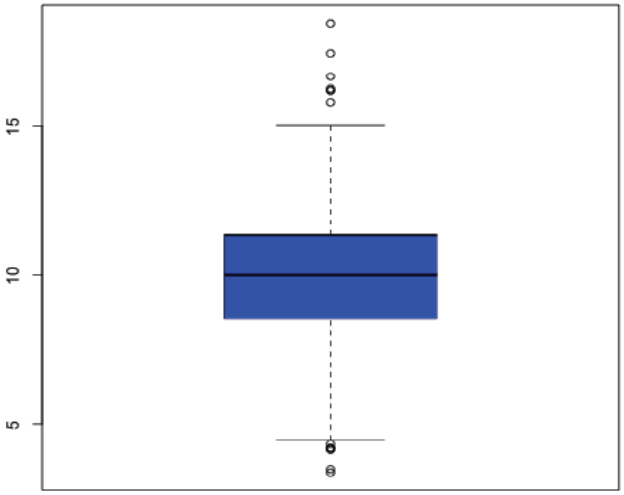
Czy w jakiś stanach przekroczona jest ta norma?

Tabela sumaryczna, box plot, histogram


```
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")
```

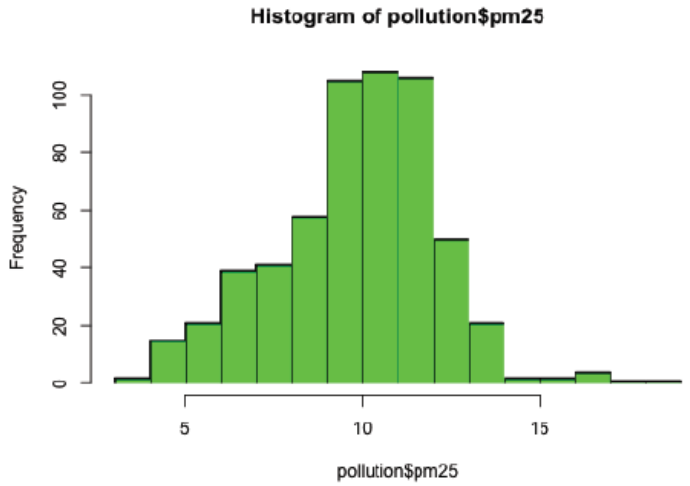
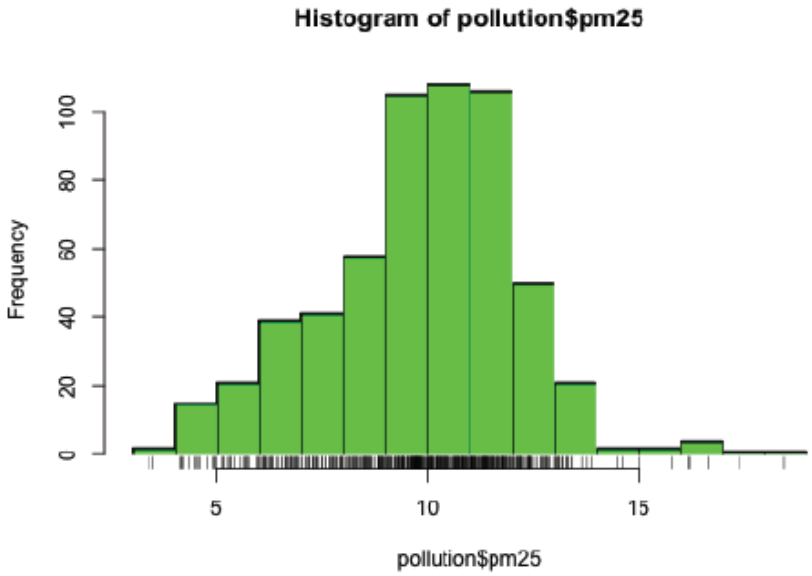


Tabela sumaryczna, box plot, histogram


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




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

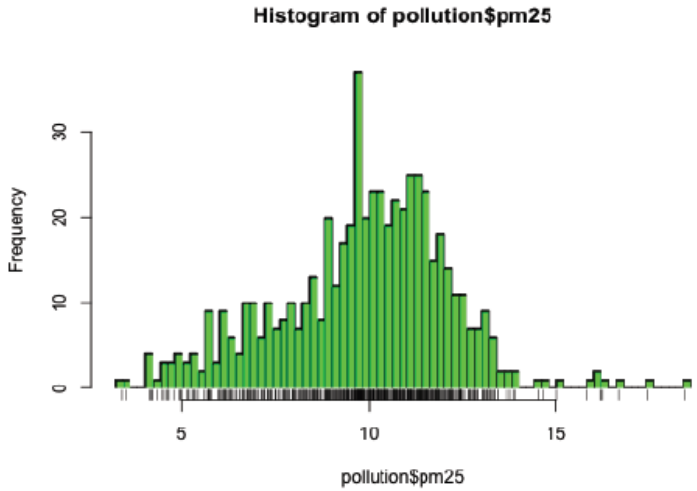
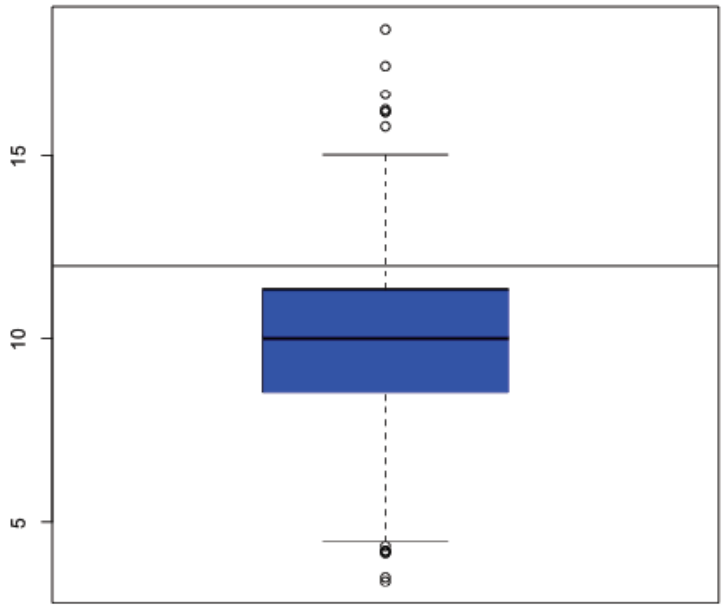
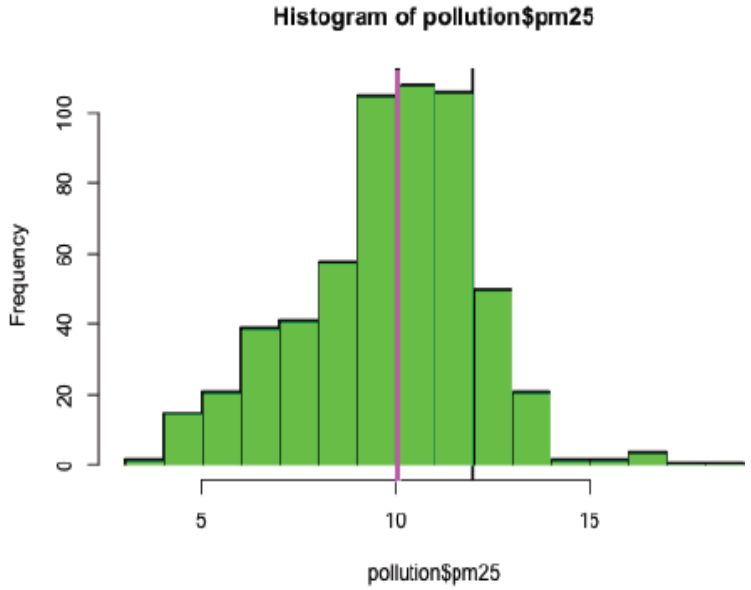


Tabela sumaryczna, box plot, histogram


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

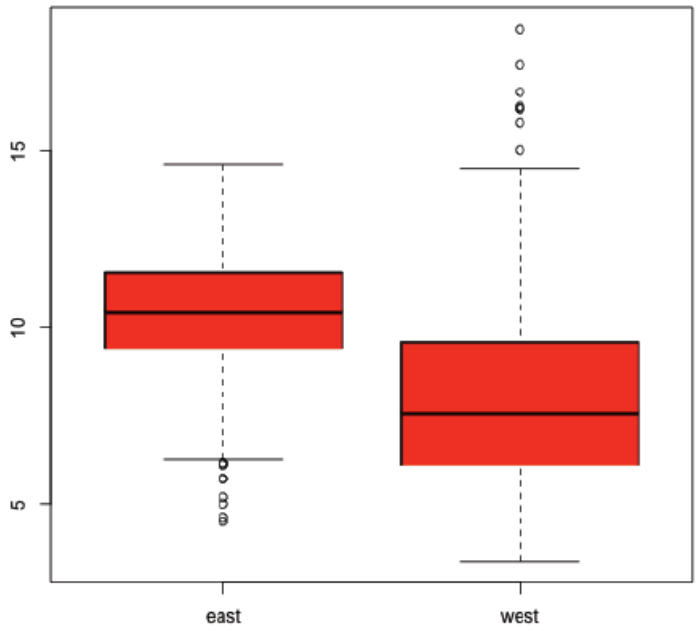



```
hist(pollution$pm25, col = "green")
abline(v = 12, lwd = 2)
abline(v = median(pollution$pm25), col = "magenta", lwd = 4)
```

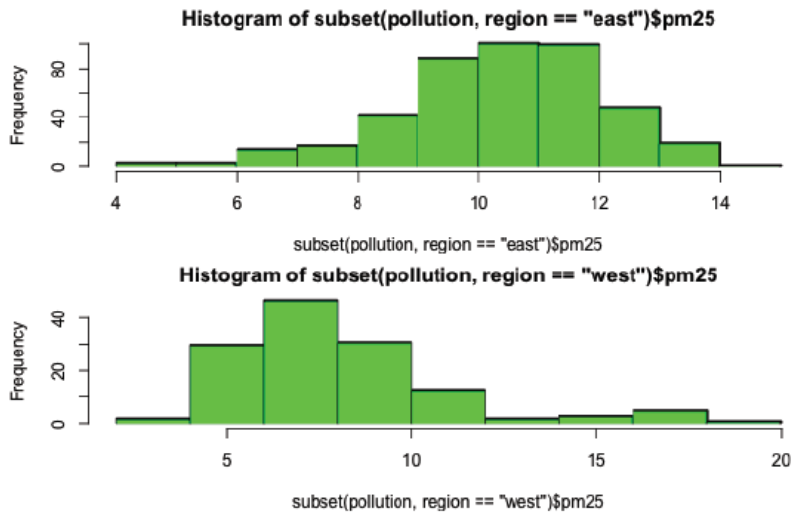


Multi-box plot, multi-histograms


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

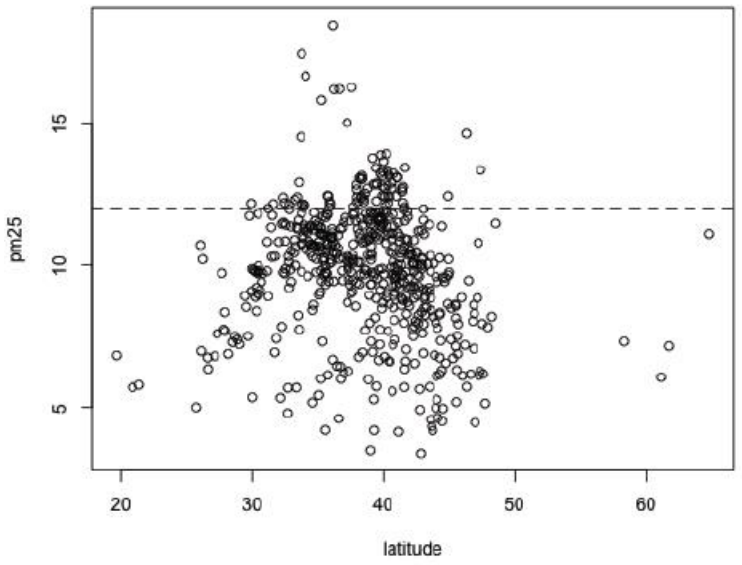



```
par(mfrow = c(2, 1), mar = c(4, 4, 2, 1))
hist(subset(pollution, region == "east")$pm25, col = "green")
hist(subset(pollution, region == "west")$pm25, col = "green")
```

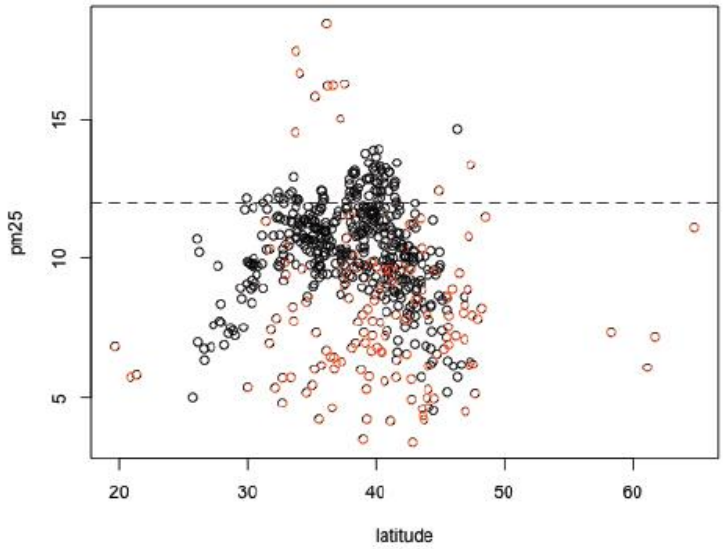


Scatter-plot


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

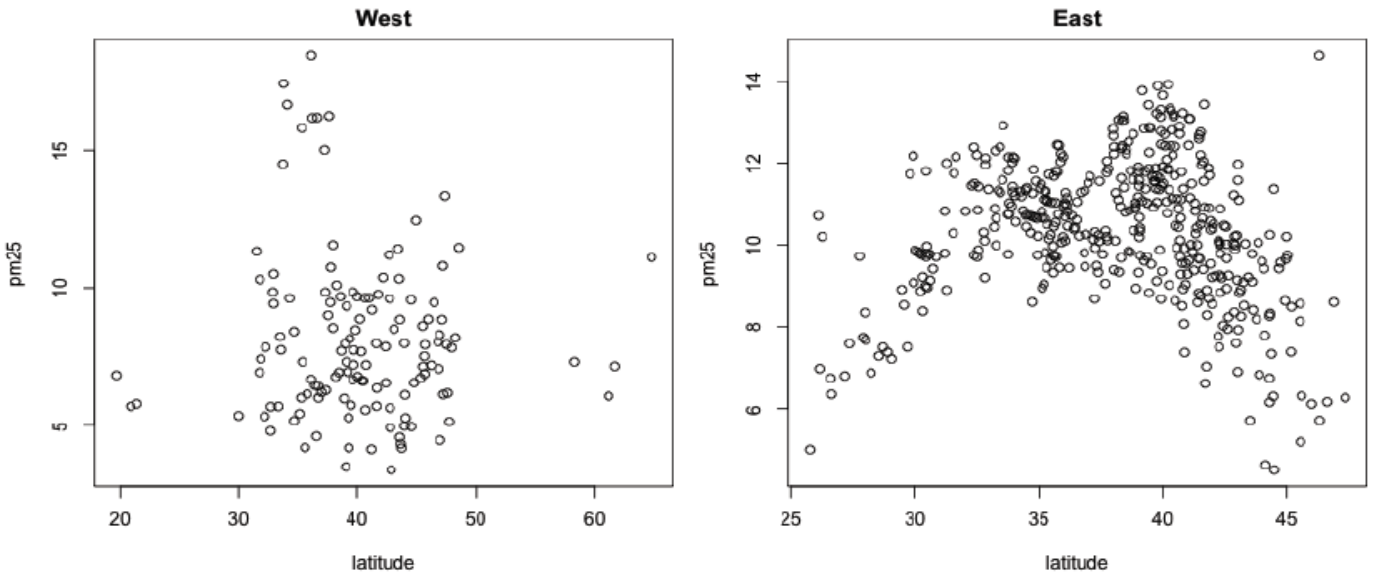



```
with(pollution, plot(latitude, pm25, col = region))
abline(h = 12, lwd = 2, lty = 2)
```



Multiple scatter plots

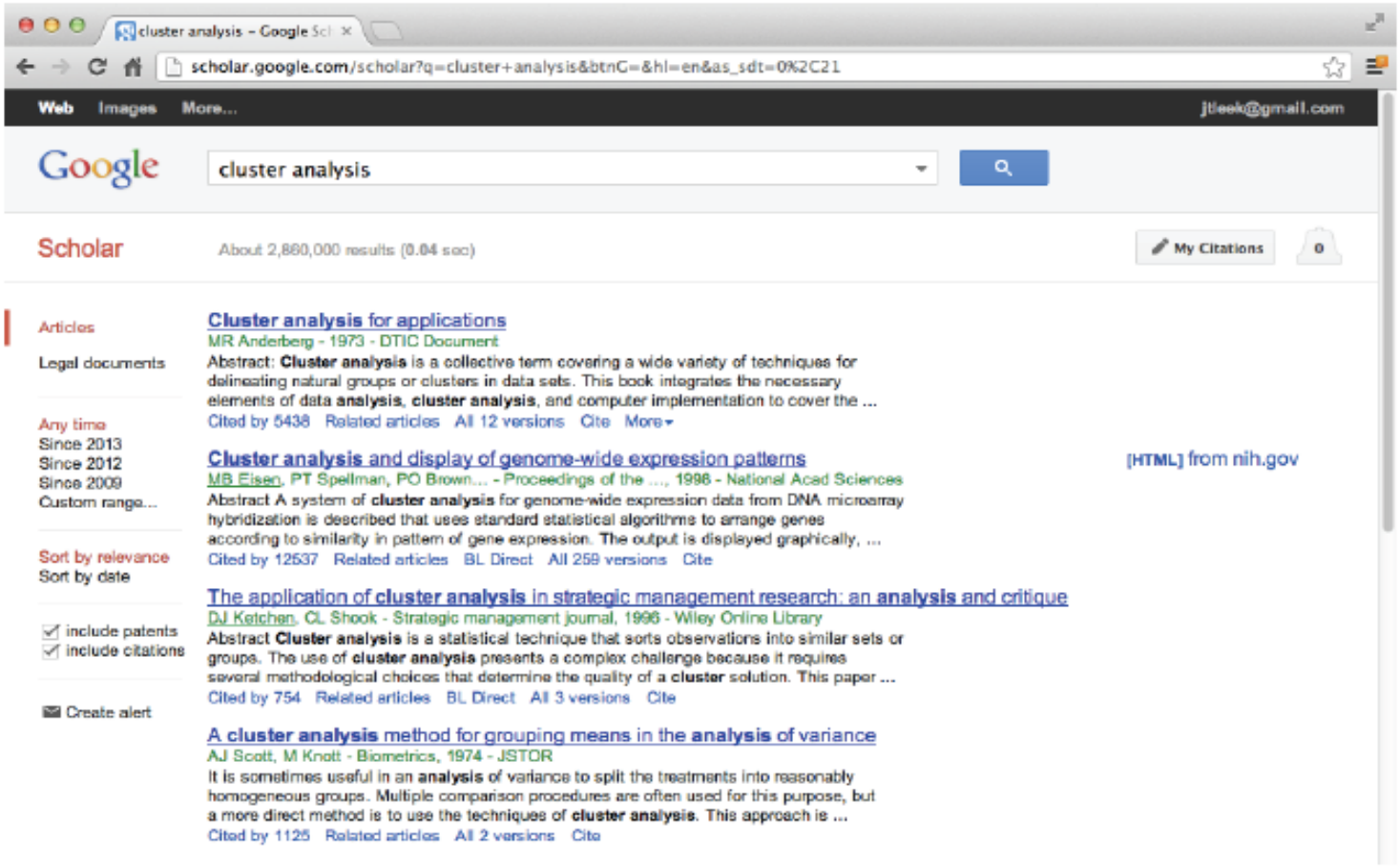

```
par(mfrow = c(1, 2), mar = c(5, 4, 2, 1))
with(subset(pollution, region == "west"), plot(latitude, pm25, main = "West"))
with(subset(pollution, region == "east"), plot(latitude, pm25, main = "East"))
```



Co się dzieje w danych które są w wielu wymiarach

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

Klastrowanie jest bardzo ważną techniką



Hierarchiczne klastrowanie

- Aglomeracyjne podejście (bottom-up)
  - Znajdź dwa najbliżej 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

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

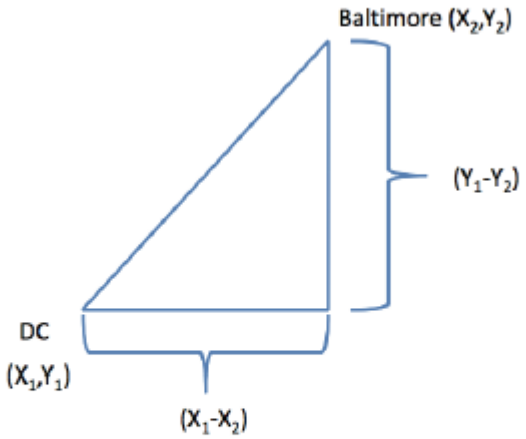
Jak definiujemy odległość

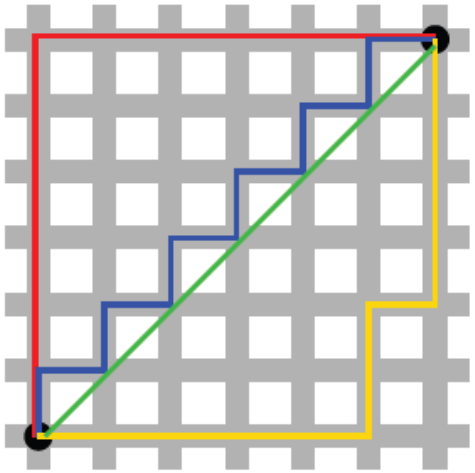
- Odległość:
  - Ciągła: euklidesowa metryka
  - Ciągła: stopień podobieństwa lub korelacji
  - Dyskretna: „Manhattan”
- Wybieramy taka definicję która stosuje się do naszych danych

Jak definiujemy odległość

- Euklidesowa metryka vs Manhattan metryka

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





Naturalnie rozszerzalne do wielu wymiarów

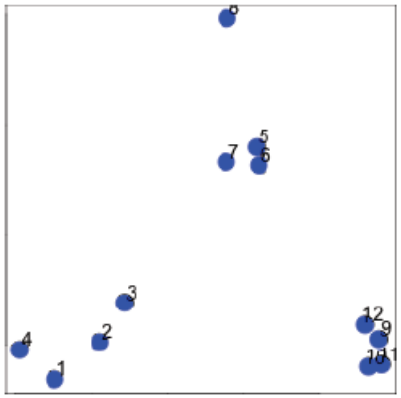
Musimy chodzić po ulicach

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

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

Hierarchiczne klastrowanie


```
set.seed(1234)
par(mar = c(0, 0, 0, 0))
x <- rnorm(12, mean = rep(1:3, each = 4), sd = 0.2)
y <- rnorm(12, mean = rep(c(1, 2, 1), each = 4), sd = 0.2)
plot(x, y, col = "blue", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
```



Symulujemy dane: 12 punków, 3 klastry

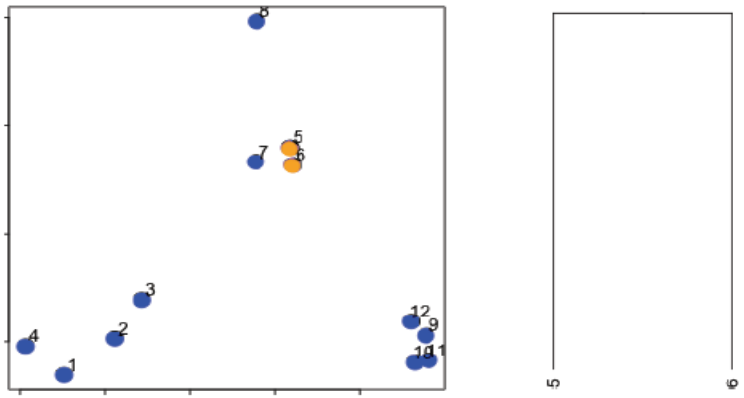
Hierarchine klastrowanie: **dist**


```
dataFrame <- data.frame(x = x, y = y)
dist(dataFrame)
```

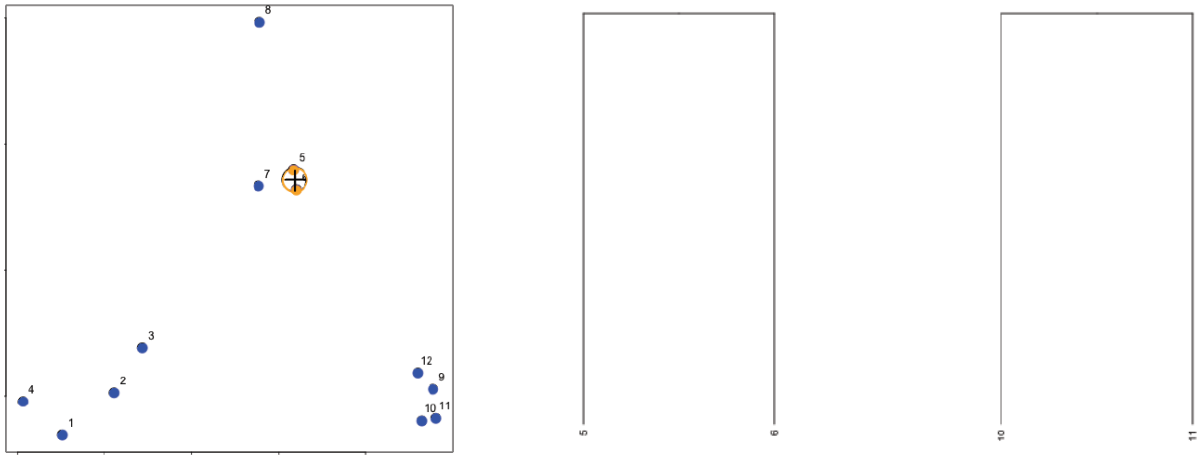
Obliczamy odległość pomiędzy wszystkimi parami punktów
Funkcja dist() ma jako parametr opcje wyboru metryki

```
##          1       2       3       4       5       6       7       8       9
## 2  0.34121
## 3  0.57494 0.24103
## 4  0.26382 0.52579 0.71862
## 5  1.69425 1.35818 1.11953 1.80667
## 6  1.65813 1.31960 1.08339 1.78081 0.08150
## 7  1.49823 1.16621 0.92569 1.60132 0.21110 0.21667
## 8  1.99149 1.69093 1.45649 2.02849 0.61704 0.69792 0.65063
## 9  2.13630 1.83168 1.67836 2.35676 1.18350 1.11500 1.28583 1.76461
## 10 2.06420 1.76999 1.63110 2.29239 1.23848 1.16550 1.32063 1.83518 0.14090
## 11 2.14702 1.85183 1.71074 2.37462 1.28154 1.21077 1.37370 1.86999 0.11624
## 12 2.05664 1.74663 1.58659 2.27232 1.07701 1.00777 1.17740 1.66224 0.10849
##         10      11
```

Hierarchiczne klastrowanie

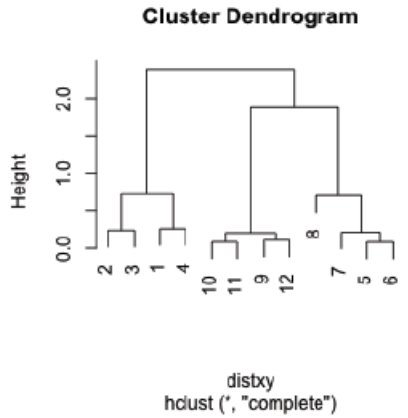


Stratujemy z punków które są od siebie najbliższe, formułujemy z nich pojedynczy super-point



Hierarchiczne klastrowanie


```
dataFrame <- data.frame(x = x, y = y)
distxy <- dist(dataFrame)
hClustering <- hclust(distxy)
plot(hClustering)
```



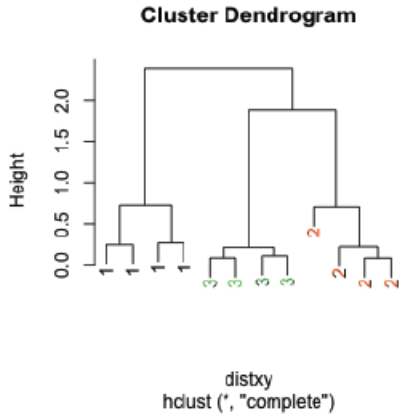
Ile mamy klastrów ?
To zależy gdzie utniemy to drzewo.
To jest już zadanie związane z interpretacją analizy

Ładniejsza graficzna reprezentacja

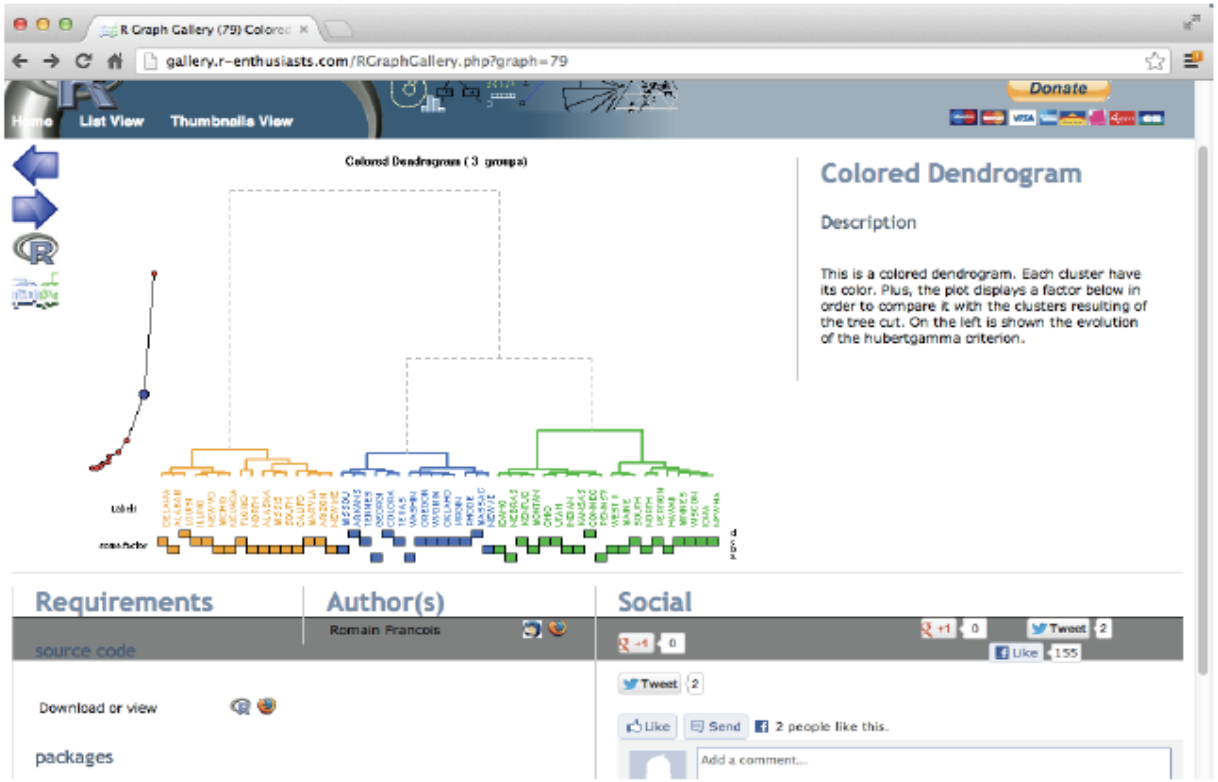

```
myplclust <- function(hclust, lab = hclust$labels, lab.col = rep(1, length(hclust$labels)),
    hang = 0.1, ...) {
    ## modifiction of plclust for plotting hclust objects *in colour*!  Copyright
    ## Eva KF Chan 2009 Arguments: hclust: hclust object lab: a character vector
    ## of labels of the leaves of the tree lab.col: colour for the labels;
    ## NA=default device foreground colour hang: as in hclust & plclust Side
    ## effect: A display of hierarchical cluster with coloured leaf labels.
    y <- rep(hclust$height, 2)
    x <- as.numeric(hclust$merge)
    y <- y[which(x < 0)]
    x <- x[which(x < 0)]
    x <- abs(x)
    y <- y[order(x)]
    x <- x[order(x)]
    plot(hclust, labels = FALSE, hang = hang, ...)
    text(x = x, y = y[hclust$order] - (max(hclust$height) * hang), labels = lab[hclust$order],
        col = lab.col[hclust$order], srt = 90, adj = c(1, 0.5), xpd = NA, ...)
}
```

Hierarchiczne klastrowanie


```
dataFrame <- data.frame(x = x, y = y)
distxy <- dist(dataFrame)
hClustering <- hclust(distxy)
myplclust(hClustering, lab = rep(1:3, each = 4), lab.col = rep(1:3, each = 4))
```



Jeszcze ładniejsza graficzna reprezentacja

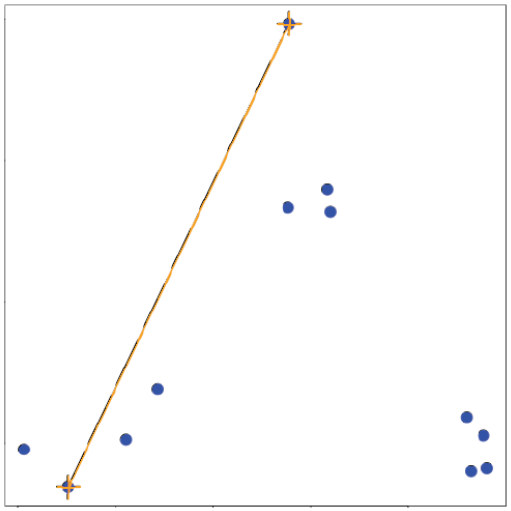


http://gallery.r-enthusiasts.com/RGraphGallery.php?graph=79

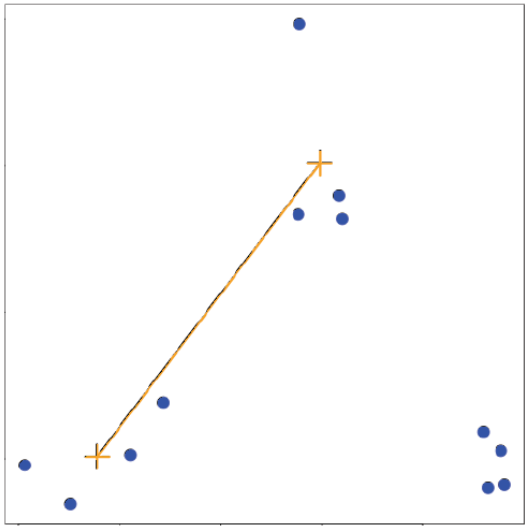
Łączenie punktów

Jaka jest odległość pomiędzy klastrami?

„Najbardziej odległe punkty”

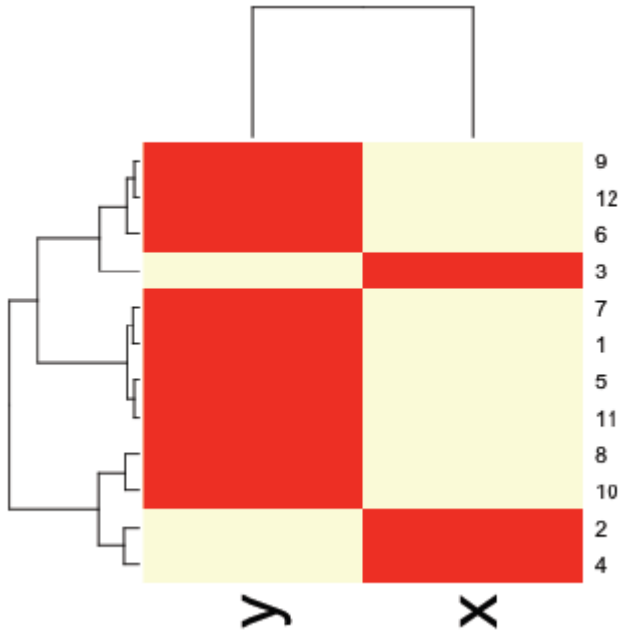


Środek ciężkości punktów



heatmap()


```
dataFrame <- data.frame(x = x, y = y)
set.seed(143)
dataMatrix <- as.matrix(dataFrame)[sample(1:12), ]
heatmap(dataMatrix)
```



heatmap() to fukcja dla ilustrowania macierzy odległości.
Ta funkcja wykonuje hierarchiczne klastrowanie na wierszach
i kolumnach macierzy odległości

Hierarchiczne klastrowanie

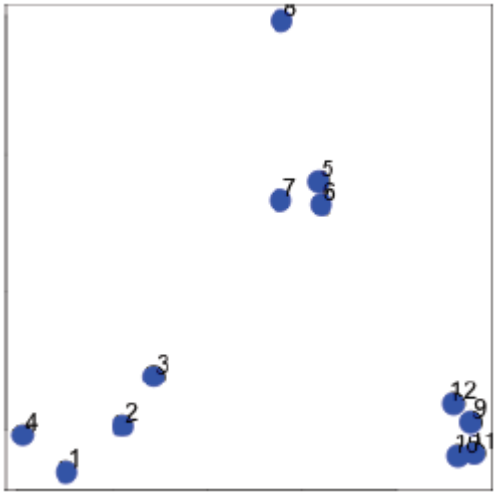
- Daje pewny obraz związku pomiędzy danymi, organizuje je w hierarchiczny sposób.
  - Jako opcje algorytmu wybieramy metrykę oraz metodę łączenia w „super-punkty”
- Wynik algorytmu klastrowania ulegnie zmianie jeżeli:
  - Usuniemy kilka punktów
  - Dane mają różne brakujące punkty
  - Zmienimy metrykę
  - Zmienimy strategie łączenia
  - Zmienimy skalę w jednym z wymiarów
- To klastrowanie jest deterministyczne (powtarzalne dla tego samego zestawu danych, nie ma losowości)
- Powinno być używane przede wszystkim dla „eksploracji” danych.

K-means klastrowanie

- Sposób na podzielenie danych na grupy
  - Zdecyduj ile chcesz mieć klastrów
  - Zdefiniuj pozycję „centrum” dla każdego klastra
  - Przypisz punkty pomiarowe do klastrów
  - Przelicz na podstawie przypisanych danych pomiarowych pozycje „centrum” dla każdego klastra
- Parametry wejściowe: metryka, ilość klastrów, punkt startowy dla centrum klastra
- Wynik: przeliczona pozycja centrum, asocjacja punktów pomiarowych do danych

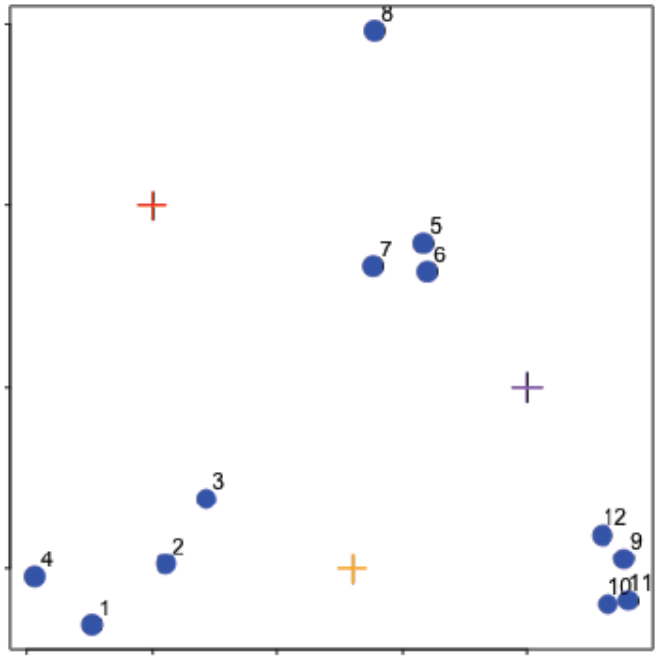
Przykład:


```
set.seed(1234)
par(mar = c(0, 0, 0, 0))
x <- rnorm(12, mean = rep(1:3, each = 4), sd = 0.2)
y <- rnorm(12, mean = rep(c(1, 2, 1), each = 4), sd = 0.2)
plot(x, y, col = "blue", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
```

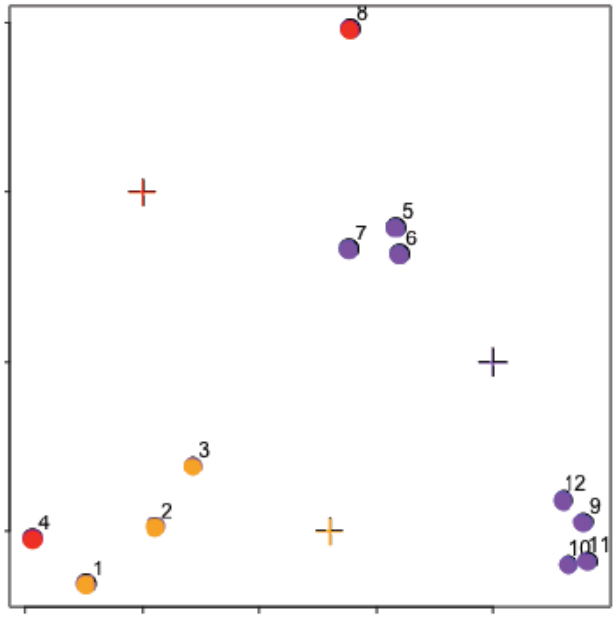


Przykład

Wybieramy startowe centra

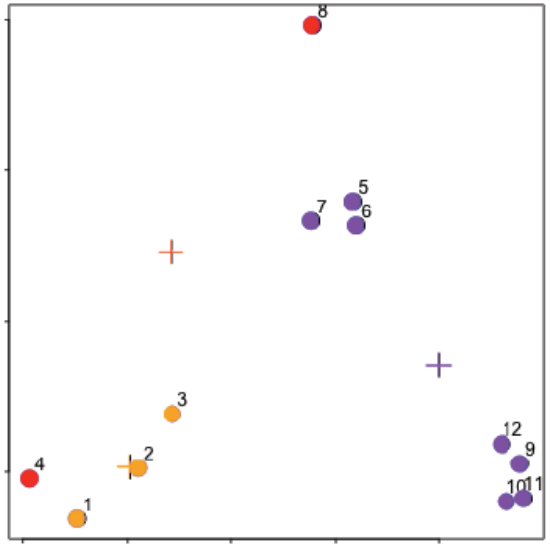


Przypisujemy punkty do klastrów początkowe grupowanie

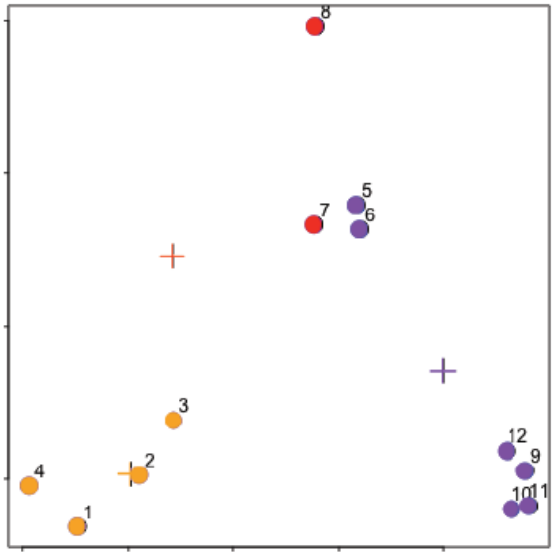


Przykład

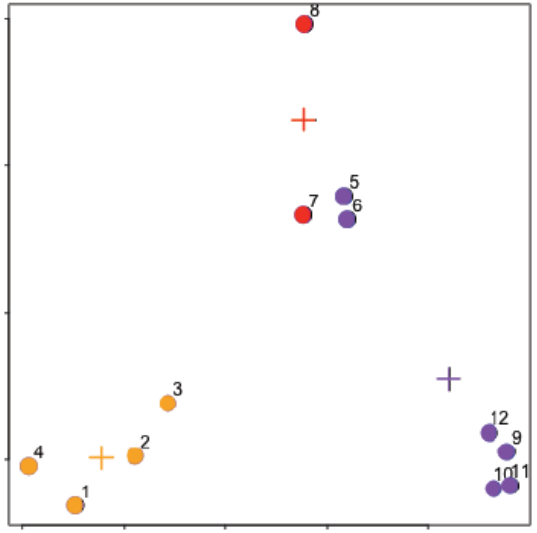
Przeliczamy pozycje centrów



Modyfikujemy przyporządkowanie



Przeliczamy raz jeszcze pozycje centrów



kmeans()


```
dataFrame <- data.frame(x, y)
kmeansObj <- kmeans(dataFrame, centers = 3)
names(kmeansObj)
```

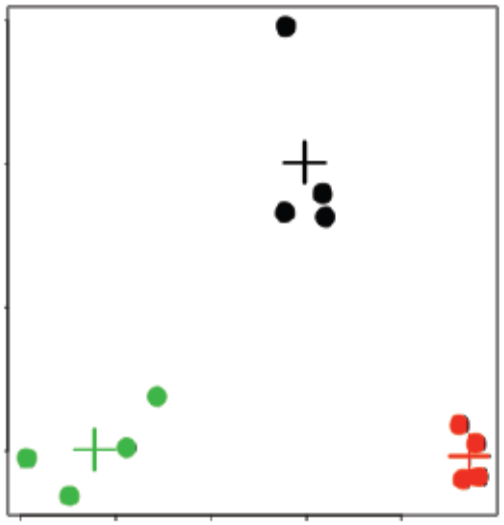
```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
kmeansObj$cluster
```

```
##  [1] 3 3 3 3 1 1 1 1 2 2 2 2
```

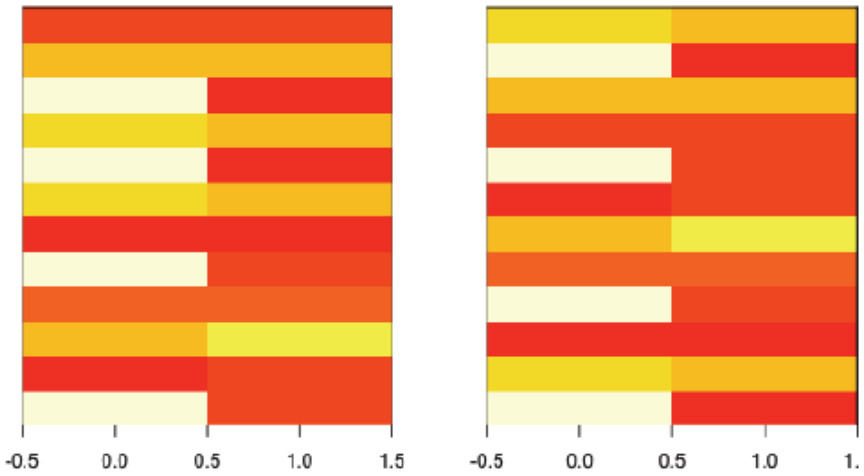
kmeans()


```
par(mar = rep(0.2, 4))
plot(x, y, col = kmeansObj$cluster, pch = 19, cex = 2)
points(kmeansObj$centers, col = 1:3, pch = 3, cex = 3, lwd = 3)
```



heatmaps()


```
set.seed(1234)
dataMatrix <- as.matrix(dataFrame)[sample(1:12), ]
kmeansObj2 <- kmeans(dataMatrix, centers = 3)
par(mfrow = c(1, 2), mar = c(2, 4, 0.1, 0.1))
image(t(dataMatrix)[, nrow(dataMatrix):1], yaxt = "n")
image(t(dataMatrix)[, order(kmeansObj$cluster)], yaxt = "n")
```



Reorganizujemy wiersze macierzy danych

K-means: podsumowanie

- K-means klastrowanie wymaga ustalenie liczby klastrów jako parameter wejściowy
  - Ocena wzrokowa scatter-plotów, intuicja
  - Techniki: cross-validation, teoria informacji, etc.
  - Inne techniki: patrz dodatkowe materiały
- Wynik algorytmu K-means:
  - Zależy od ustalonej liczby klastrów
  - Zależy od liczby iteracji