- Shlukování (clustering) – nalezení shluků dat s podobnými vlastnostmi (třídění dat do skupin)

Vzdálenost mezi datovými body

Euklidovská	$m_E(x,y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$
Manhattanská	$m_{Ma}(x,y) = \lvert(x_1 - y_1)\rvert + \ldots + \lvert(x_n - y_n)\rvert$
Minkowského	$m_{Mi}(x,y) = (\lvert(x_1 - y_1)\rvert^p + \ldots + \lvert(x_n - y_n)\rvert^p)^{\frac{1}{p}}$

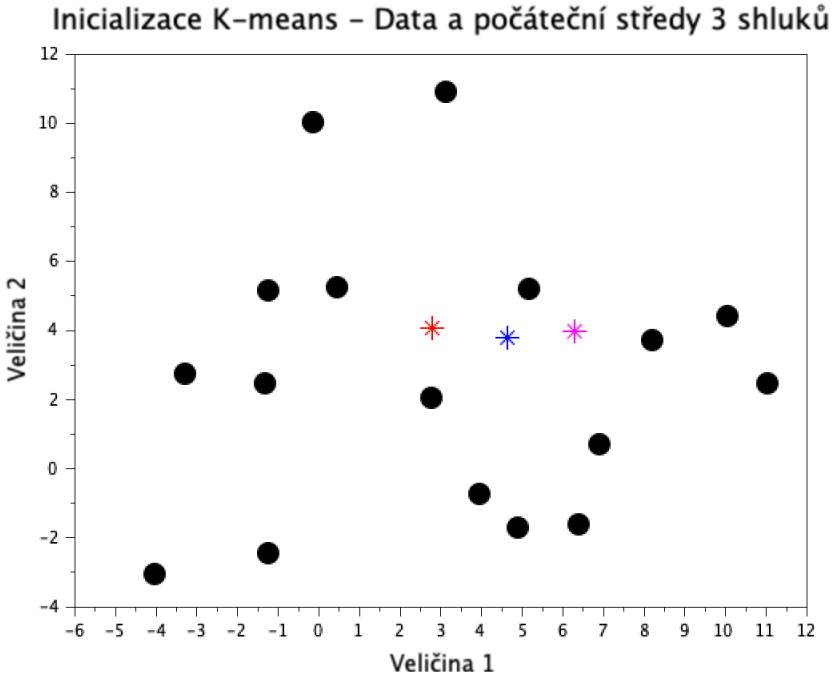
Algoritmus K-means

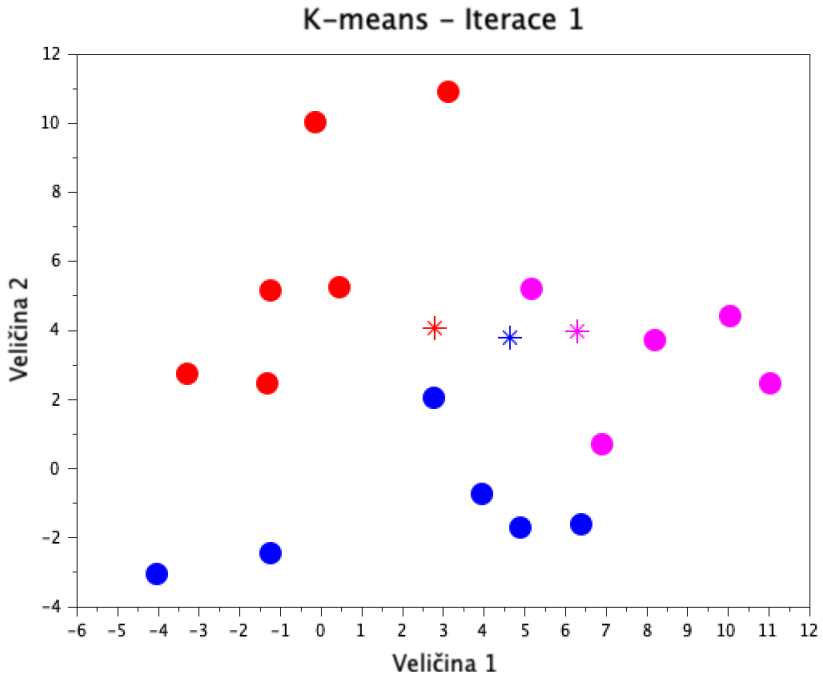
offline bez učitele

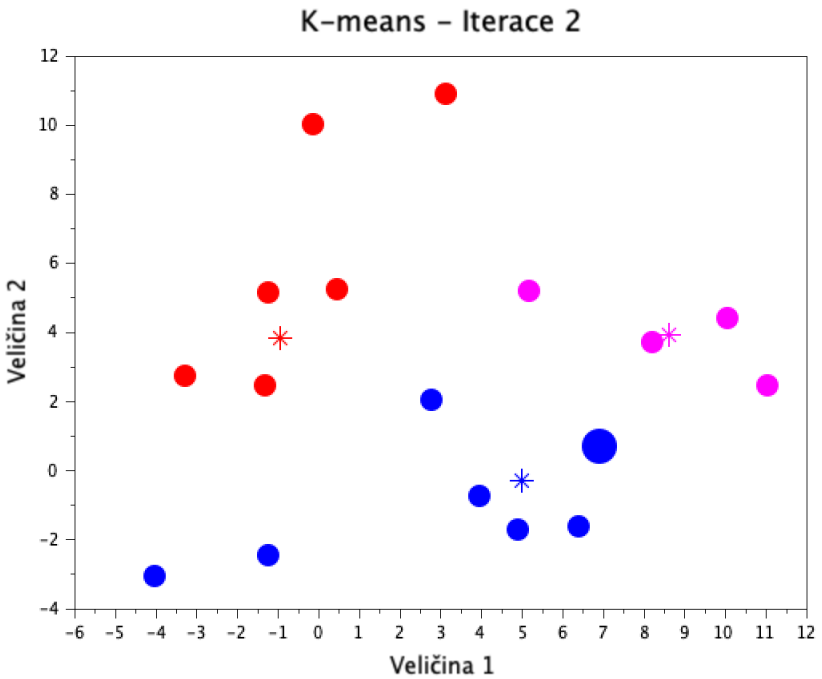
Inicializace: počet shluků, počáteční středy

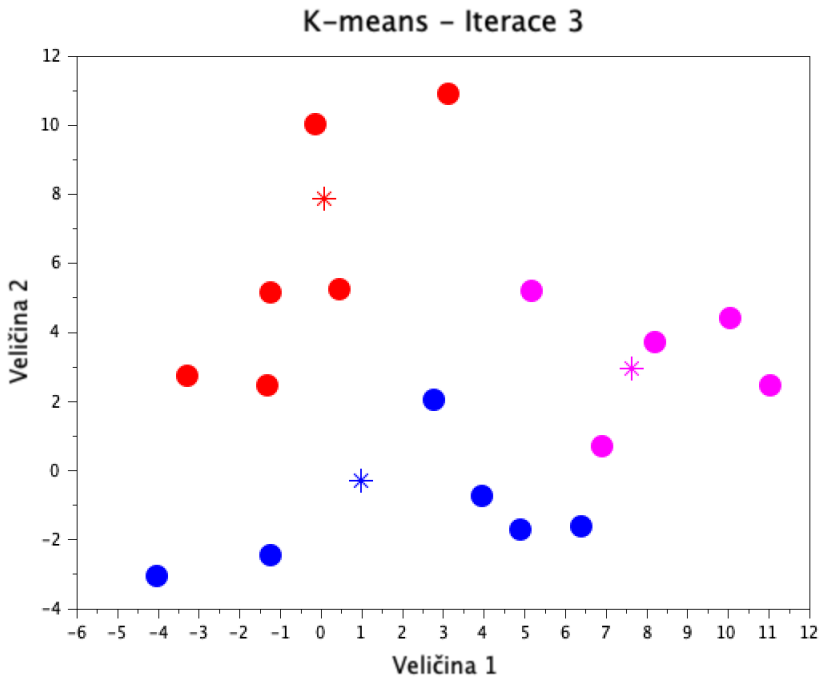
Iterace:

1. vzdálenost mezi každým datovým bodem a každým středem
2. přiřadíme každý bod k nejbližšímu středu – vytvoříme shluk
3. průměr bodů v každém shluku = nový střed
4. Jdeme na krok 1, dokud se středy mění

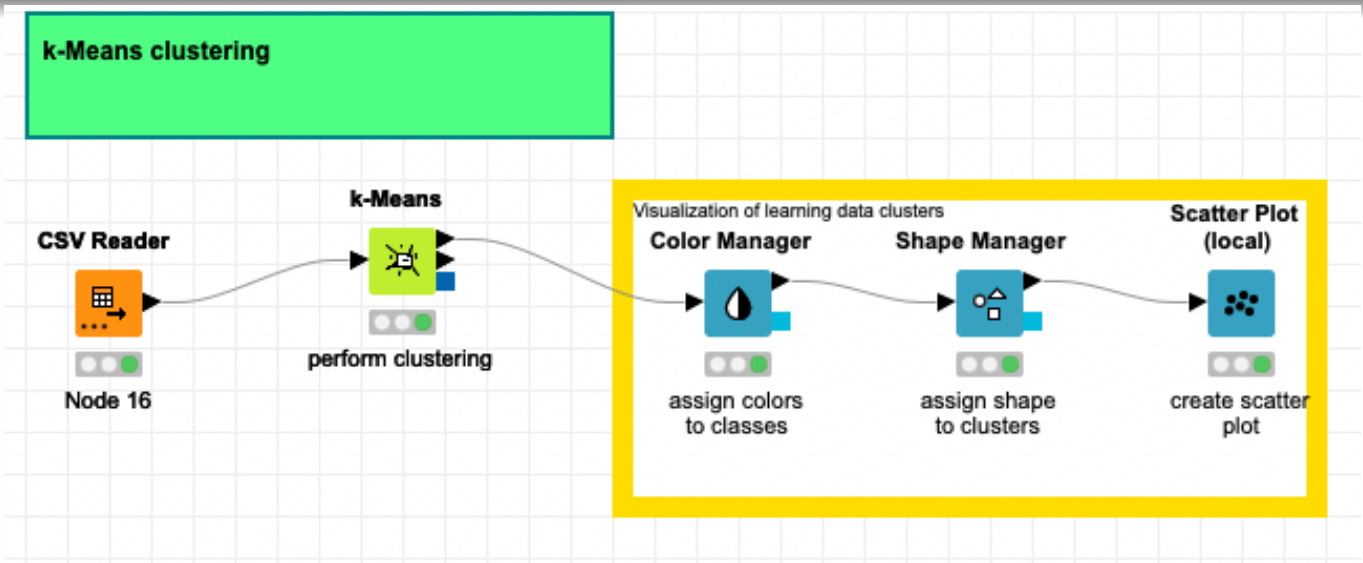

Inicializace K-means – Data a počáteční středy 3 shluků
Veličina 2
Veličina 1


K-means – Iterace 1
Veličina 2
Veličina 1


K-means – Iterace 2
Veličina 2
Veličina 1


K-means – Iterace 3
Veličina 2
Veličina 1

Program v KNIME



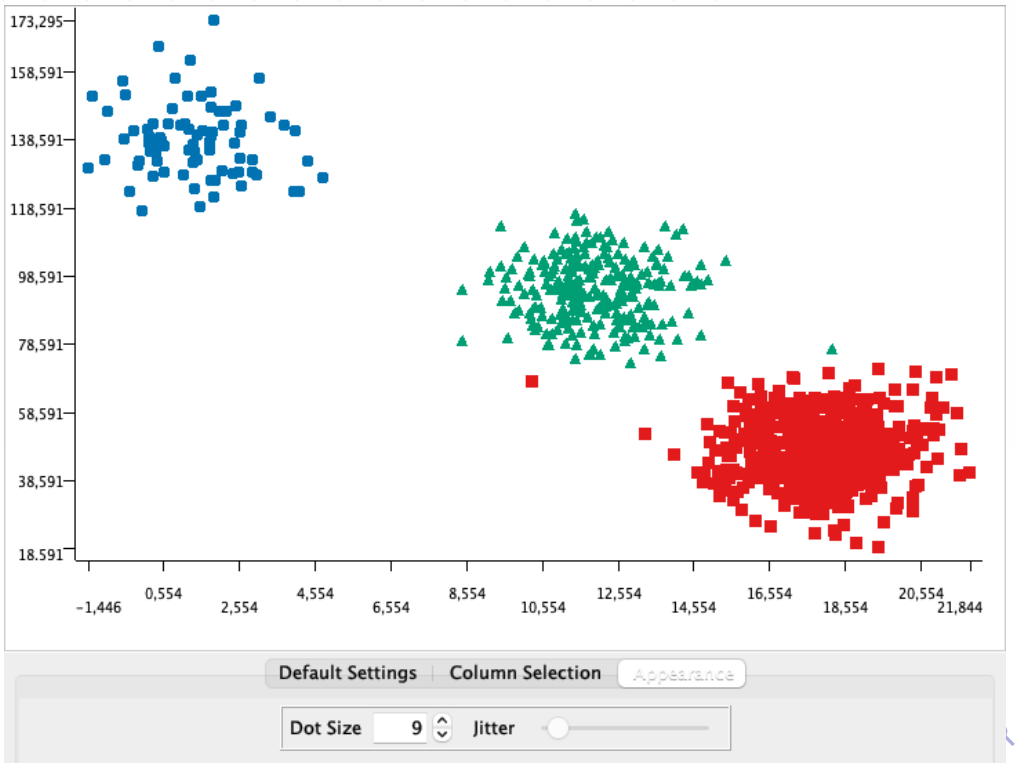
Příklad:

$y_{1;t}$ – tlak brzdové soustavy (bar)

$y_{2;t}$ – rychlost (km/h)

t – sekundy

shluky – styl jízdy (offline shlukování)



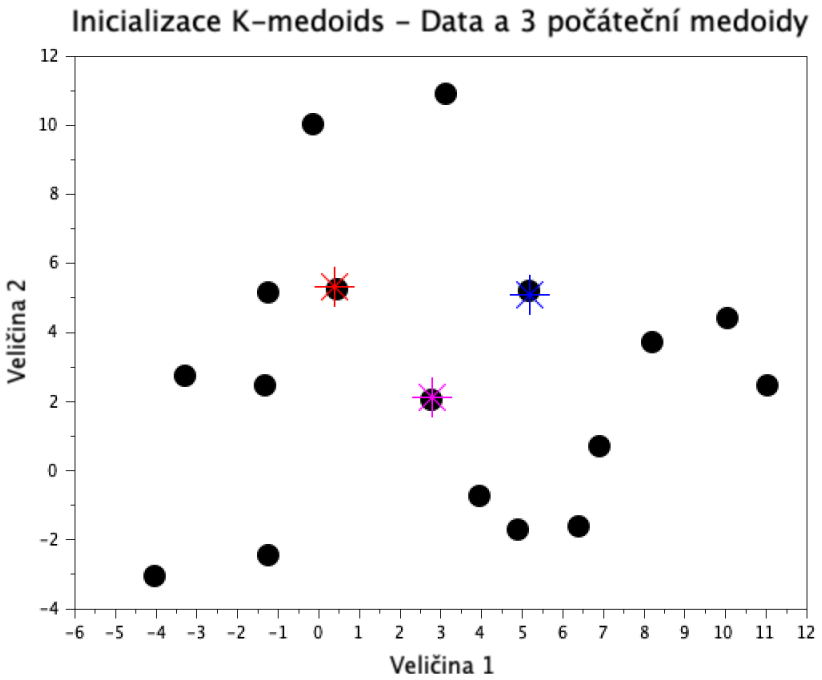
Algoritmus K-medoids

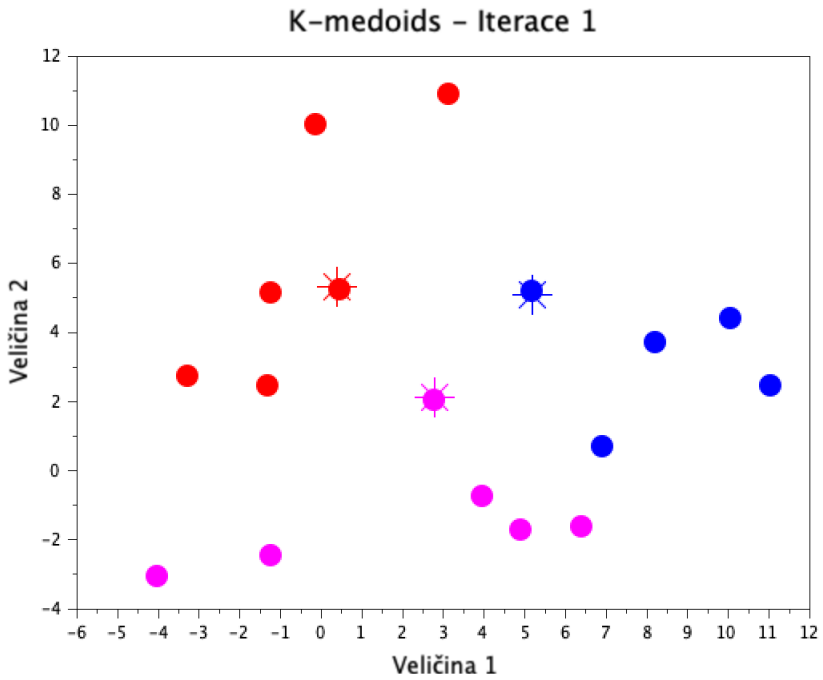
Inicializace:

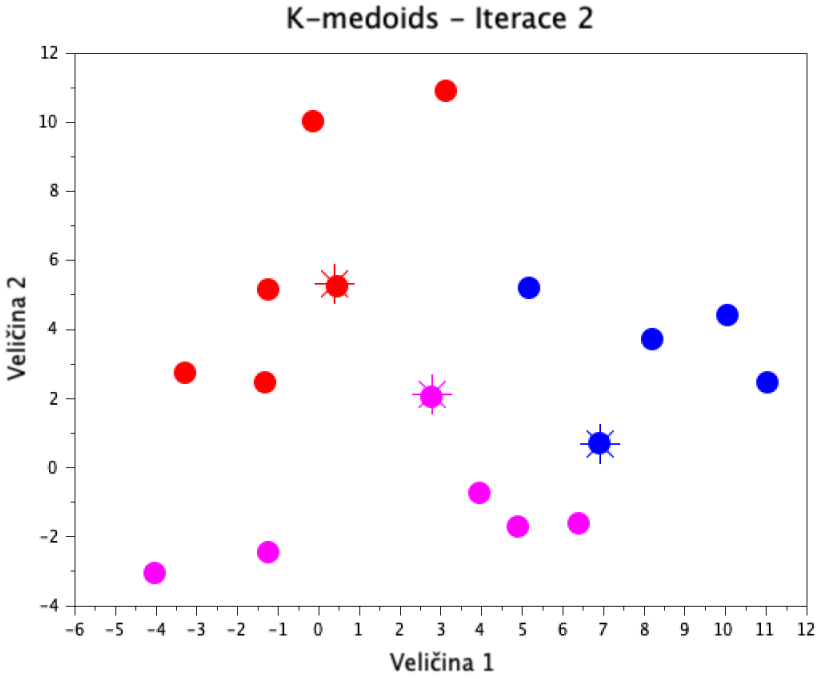
1. k náhodně vybraných datových bodů = medoidy, zbytek dat = ne-medoidy
2. vzdálenost mezi každým datovým bodem a každým medoidem
3. přiřadíme každý bod k nejbližšímu medoidu – vytvoříme shluk
4. celková vzdálenost bodů od svých medoidů uvnitř shluků

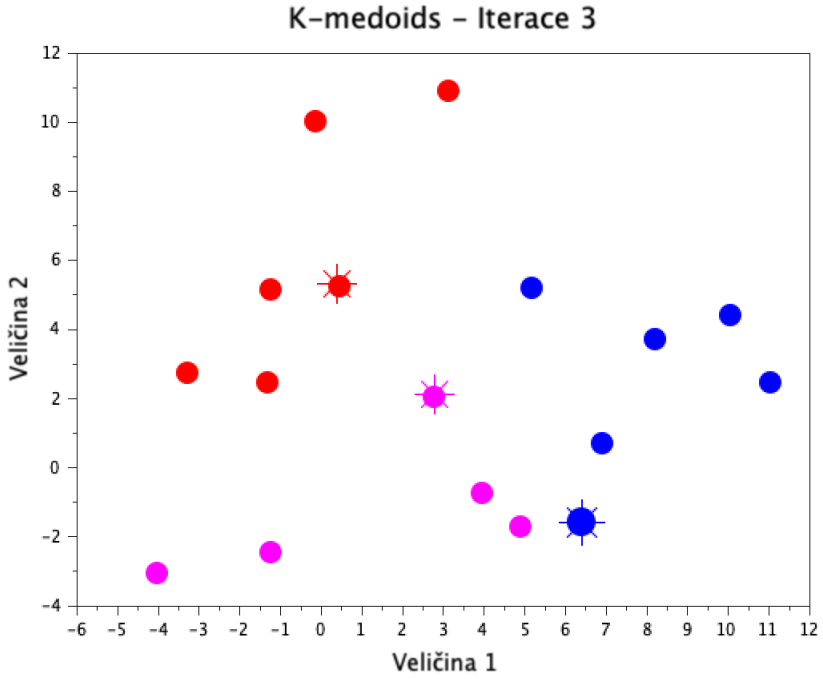
Iterace:

1. Náhodně vyměníme jeden medoid za ne-medoid
2. vzdálenost mezi každým datovým bodem a každým medoidem
3. přiřadíme každý bod k nejbližšímu medoidu
4. Spočteme novou celkovou vzdálenost bodů uvnitř shluků
5. Pokud je nová vzdálenost větší než minulá, výměnu vrátíme
6. Jdeme na krok 1

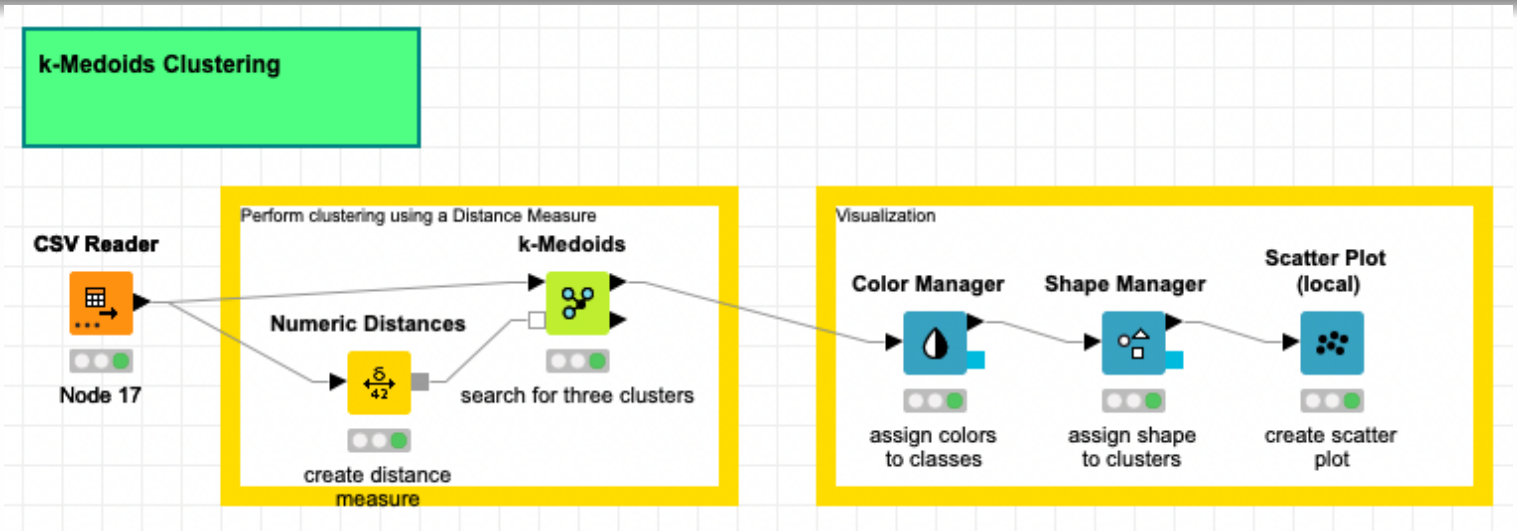

Inicializace K-medoids - Data a 3 počáteční medoidy
Veličina 2
Veličina 1

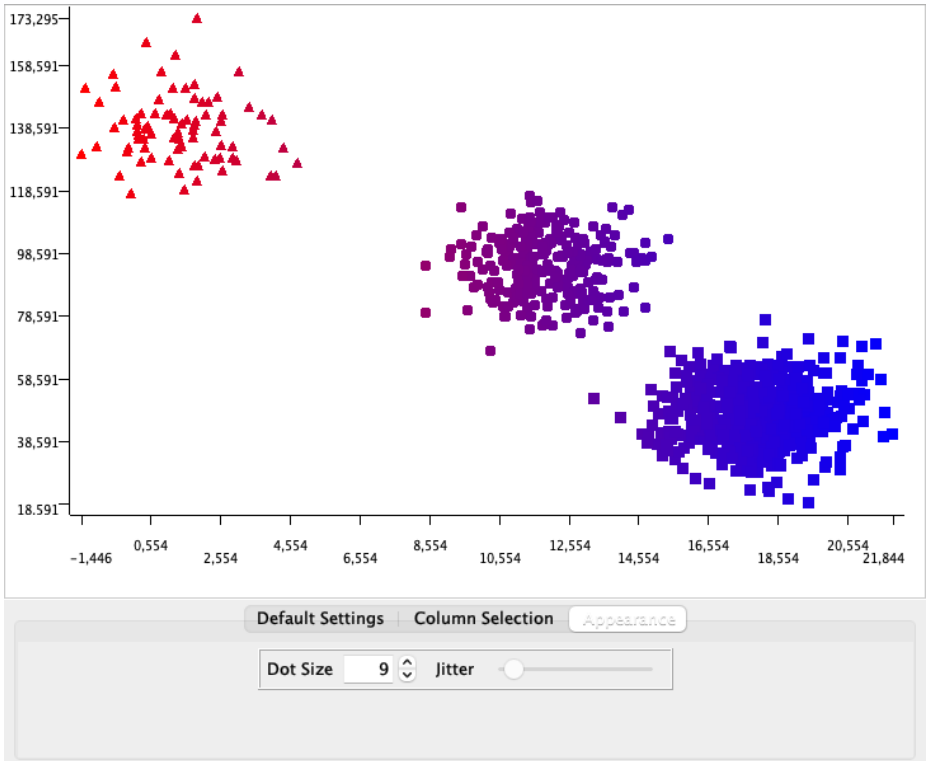

K-medoids - Iterace 1
Veličina 2
Veličina 1


K-medoids - Iterace 2
Veličina 2
Veličina 1


K-medoids - Iterace 3
Veličina 2
Veličina 1

Program v KNIME





Poznámky:

- rozdíl oproti K-means – středy (medoidy) jsou vždy datové body, proto – delší doba výpočtů
- odolnější vůči šumu a odlehlým hodnotám
- odkaz na web

Fuzzy shlukování (c-means)

Princip:

- Minimalizujeme kritérium

$$J = \sum_{i=1}^{n_d} \sum_{j=1}^{n_c} u_{ij}^m ||y_i - c_j||^2, \quad m \geq 1$$

y_i – datové body, c_j – středy shluků,
u_{ij} – funkce příslušnosti
m – parametr (vyšší m = rozmazané shluky)

- Přepočet

$$c_j = \frac{\sum_{i=1}^{n_d} u_{ij}^m y_i}{\sum_{i=1}^{n_d} u_{ij}^m}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{n_c} \left(\frac{||y_i - c_j||}{||y_i - c_k||} \right)^{\frac{2}{m-1}}}$$

Algoritmus:

Inicializace: počet shluků, počáteční matice příslušnosti (počet dat× počet shluků)
Iterace:

1. Vypočteme středy shluků
2. Přepočteme matici příslušnosti
3. Jdeme na krok 1, dokud rozdíl matic příslušnosti není zanedbatelný

Matice příslušnosti – vliv parametru m

m=2

Row...	Column0 Number (double)	Column1 Number (double)	cluster_0 Number (double)	cluster_1 Number (double)	cluster_2 Number (double)	Winner Cluster String
Row0	4.928	-1.734	0.904	0.061	0.035	cluster_0
Row1	-1.232	-2.486	0.584	0.115	0.3	cluster_0
Row2	-3.278	2.736	0.133	0.056	0.811	cluster_2
Row3	-1.321	2.436	0.124	0.043	0.833	cluster_2
Row4	3.948	-0.752	0.994	0.004	0.003	cluster_0
Row5	-4.01	-3.078	0.451	0.141	0.408	cluster_0
Row6	11.043	2.457	0.105	0.848	0.048	cluster_1
Row7	6.391	-1.618	0.732	0.197	0.07	cluster_0
Row8	8.194	3.681	0.002	0.997	0.001	cluster_1

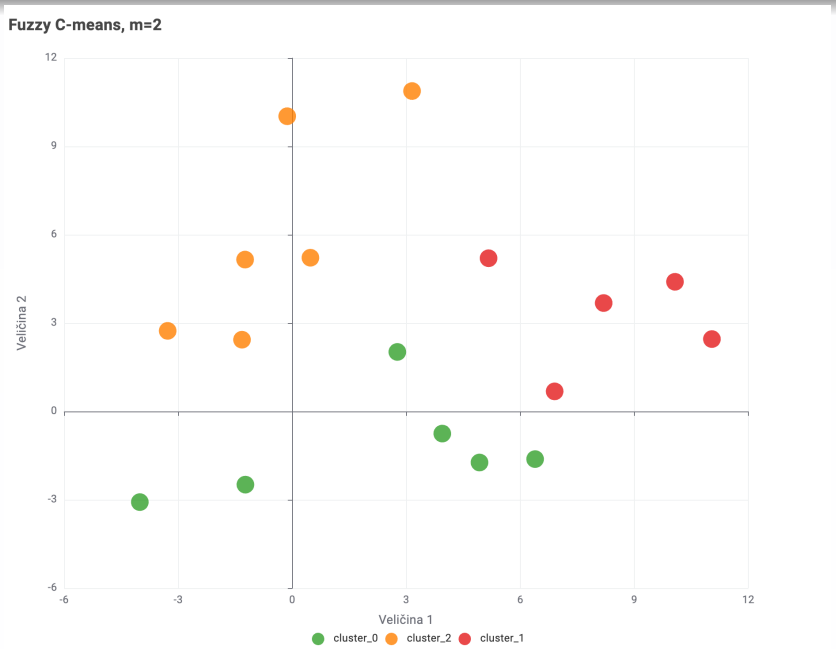
m=5

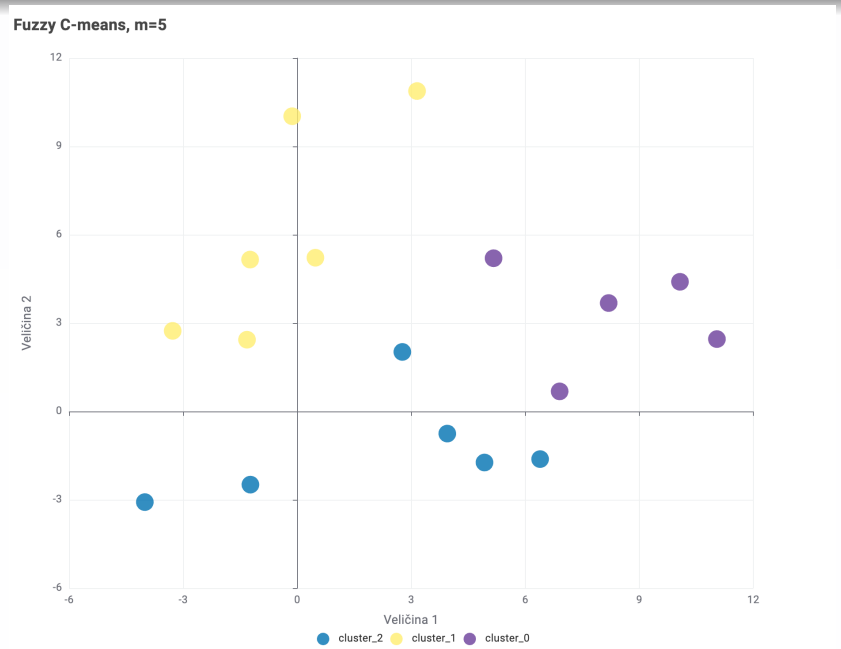
Row...	Column0 Number (double)	Column1 Number (double)	cluster_0 Number (double)	cluster_1 Number (double)	cluster_2 Number (double)	Winner Cluster String
Row0	4.928	-1.734	0.256	0.22	0.524	cluster_2
Row1	-1.232	-2.486	0.271	0.341	0.388	cluster_2
Row2	-3.278	2.736	0.243	0.464	0.292	cluster_1
Row3	-1.321	2.436	0.231	0.48	0.289	cluster_1
Row4	3.948	-0.752	0.113	0.105	0.783	cluster_2
Row5	-4.01	-3.078	0.279	0.36	0.361	cluster_2
Row6	11.043	2.457	0.465	0.239	0.296	cluster_0
Row7	6.391	-1.618	0.311	0.239	0.45	cluster_2
Row8	8.194	3.681	0.847	0.069	0.084	cluster_0

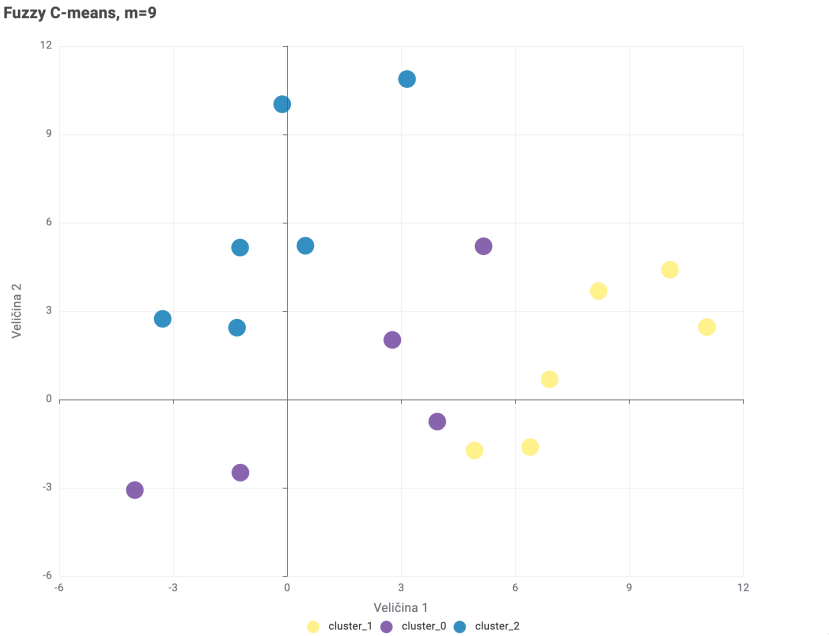
m=9

Row...	Column0 Number (double)	Column1 Number (double)	cluster_0 Number (double)	cluster_1 Number (double)	cluster_2 Number (double)	Winner Cluster String
Row0	4.928	-1.734	0.342	0.366	0.292	cluster_1
Row1	-1.232	-2.486	0.35	0.321	0.33	cluster_0
Row2	-3.278	2.736	0.336	0.297	0.367	cluster_2
Row3	-1.321	2.436	0.342	0.29	0.368	cluster_2
Row4	3.948	-0.752	0.357	0.352	0.291	cluster_0
Row5	-4.01	-3.078	0.344	0.32	0.337	cluster_0
Row6	11.043	2.457	0.324	0.373	0.302	cluster_1
Row7	6.391	-1.618	0.33	0.383	0.287	cluster_1
Row8	8.194	3.681	0.324	0.379	0.297	cluster_1

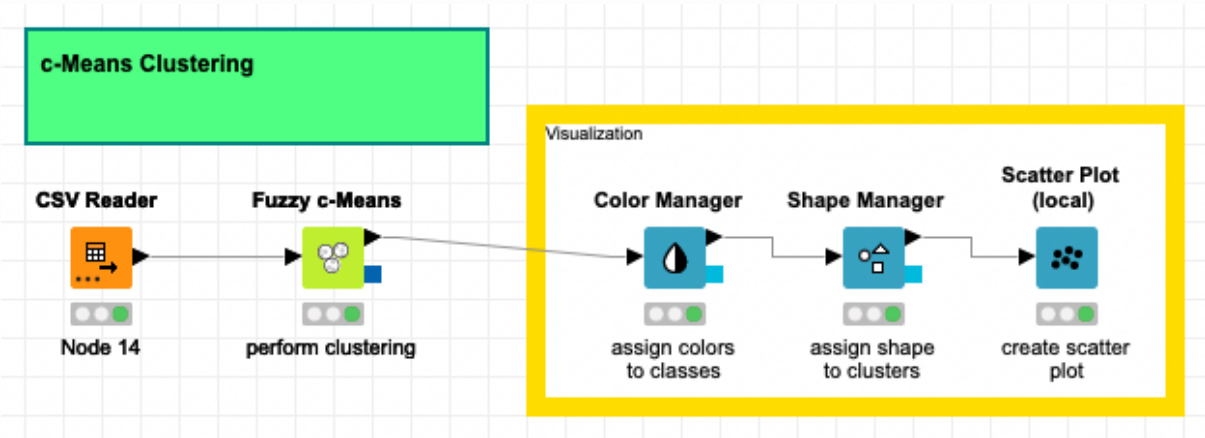
Shluky – vliv parametru m

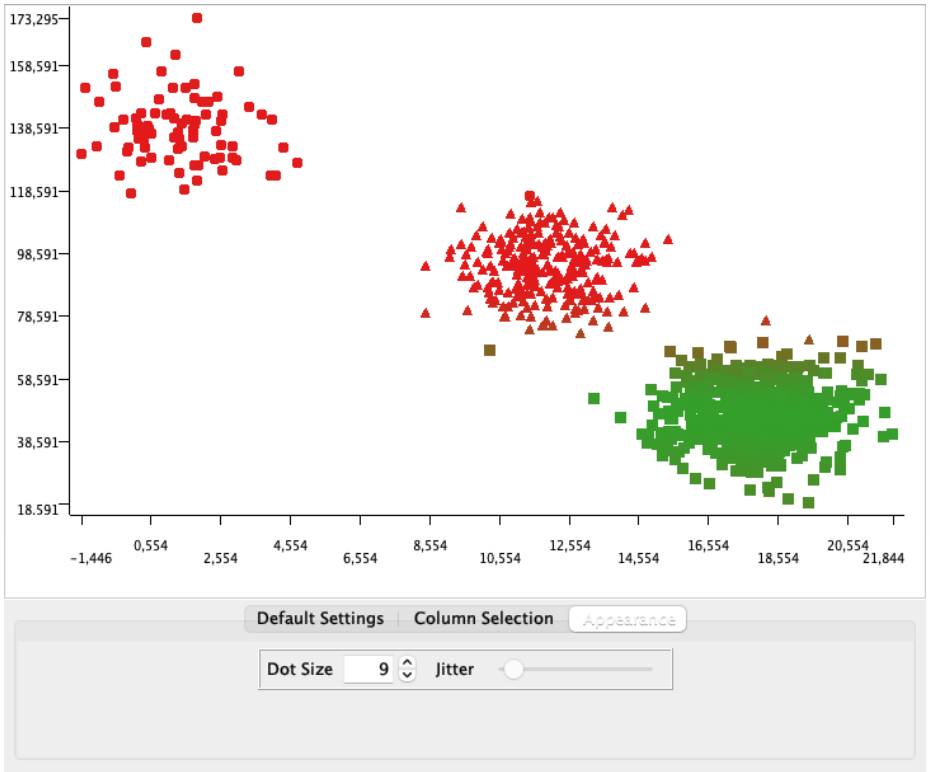






Program v KNIME





Poznámky:

- rozdíl oproti K-means – fuzzy přístup

Shlukování na základě hustoty (DBSCAN)

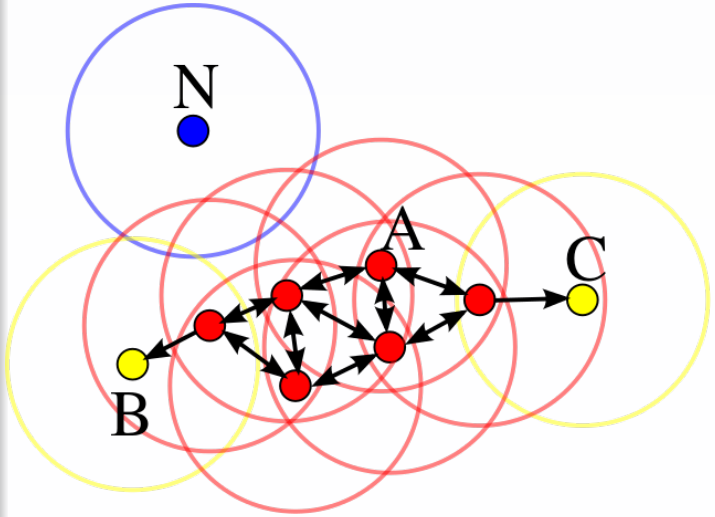
offline bez učitele

Density-Based Spatial Clustering of Applications with Noise

Definice:

- ϵ-okolí bodů A,B: vzdálenost(A,B) $< \epsilon$
- bod A – vnitřní bod, pokud má ve svém ϵ-okolí aspoň minimální počet bodů
- bod B – dosažitelný z bodu A, jestliže z A do B v ϵ-okolí vede cesta (přímá nebo posloupnost vnitřních bodů)
- bod N – šumový, pokud není dosažitelný z žádného jiného bodu
- mezi A a B existuje spojení, pokud jsou oba dosažitelné z nějakého vnitřního bodu (density-connectedness)
- Shluk: vnitřní bod + všechny z něj dosažitelné

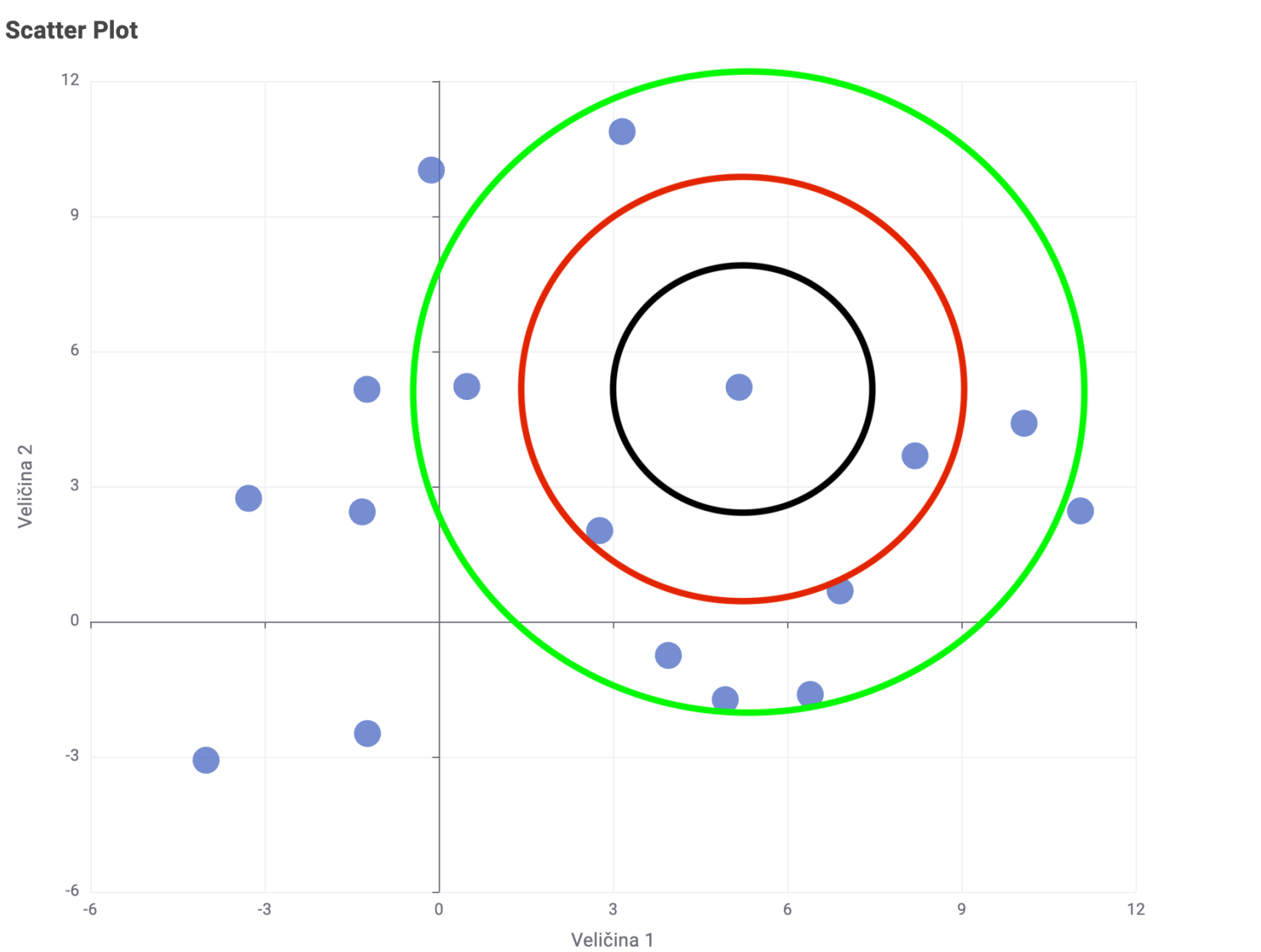
Příklad: minPts = 4



- – vnitřní body
- – dosažitelné
- – šumový bod

ϵ-okolí – 4 body všechny navzájem dosažitelné – tvoří shluk

Příklad: ϵ-okolí

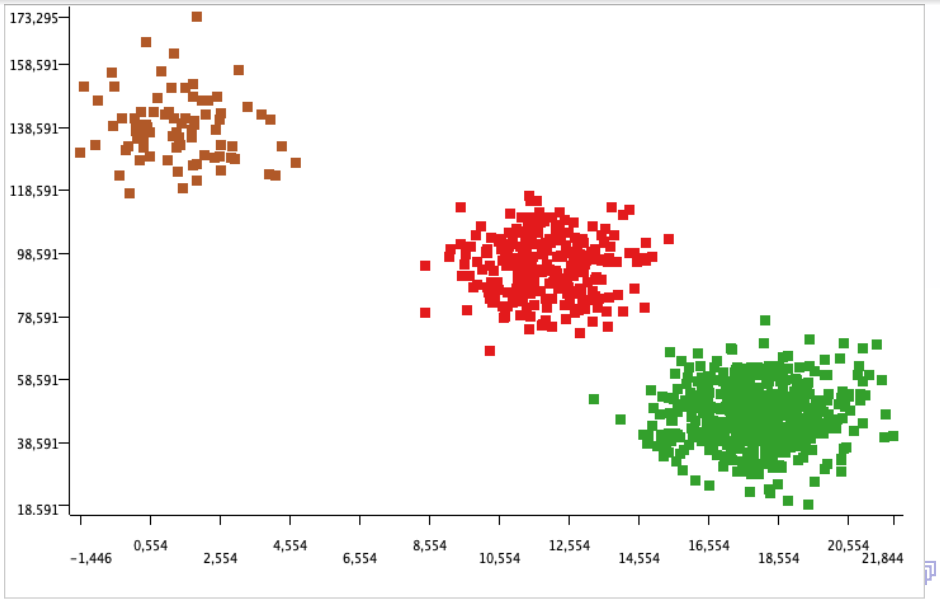


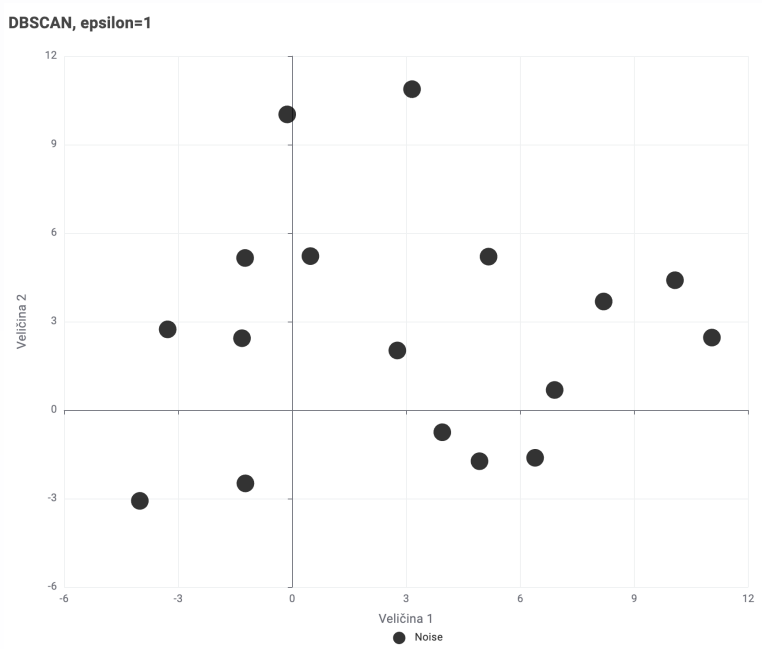
Princip algoritmu DBSCAN

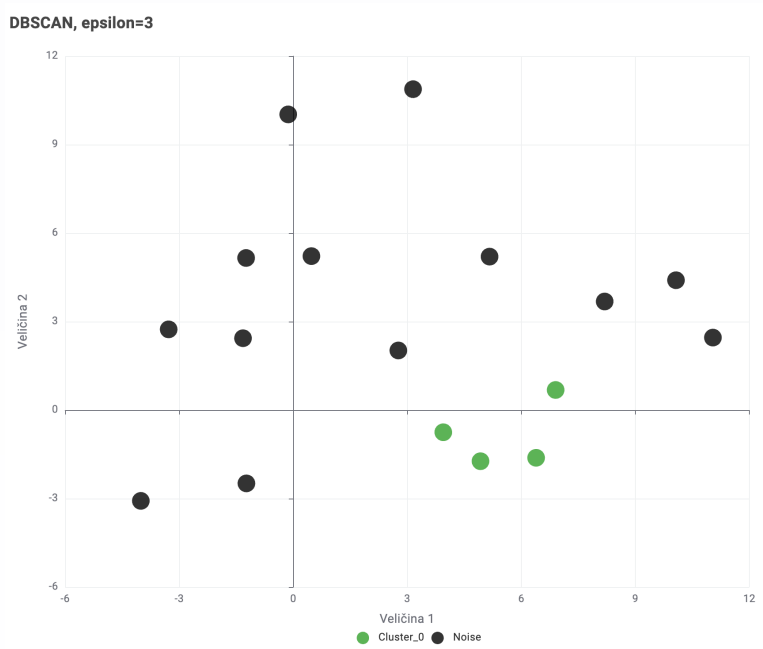
Inicializace: ϵ-okolí, minimální počet bodů minPts

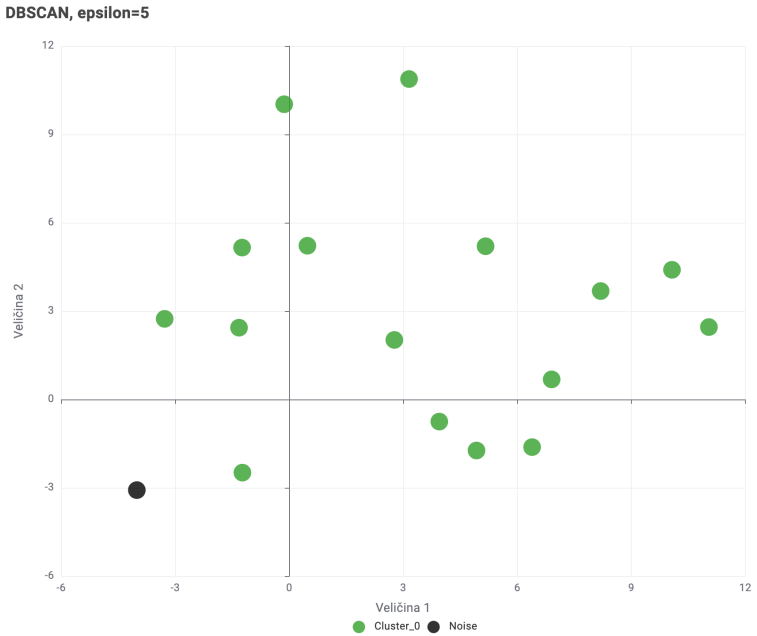
Iterace:

1. Určíme ϵ-okolí libovolného bodu, a pokud obsahuje dostatečný počet bodů, založíme shluk, jinak je to šum
2. Určíme dosažitelné body v ϵ-okolí a přidáme je do shluku
3. Pokračujeme, dokud není hustě propojený shluk zcela nalezen
4. Jdeme na krok 1 pro další libovolný bod, který zatím není v žádném shluku

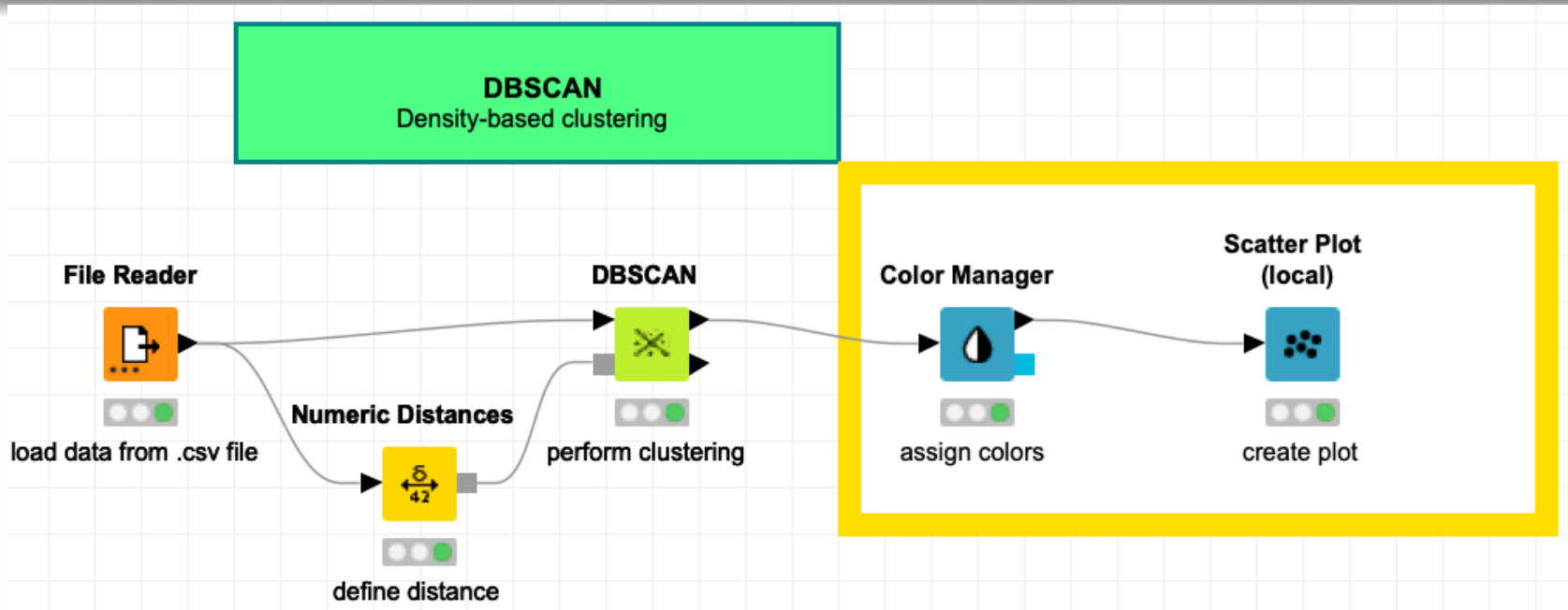


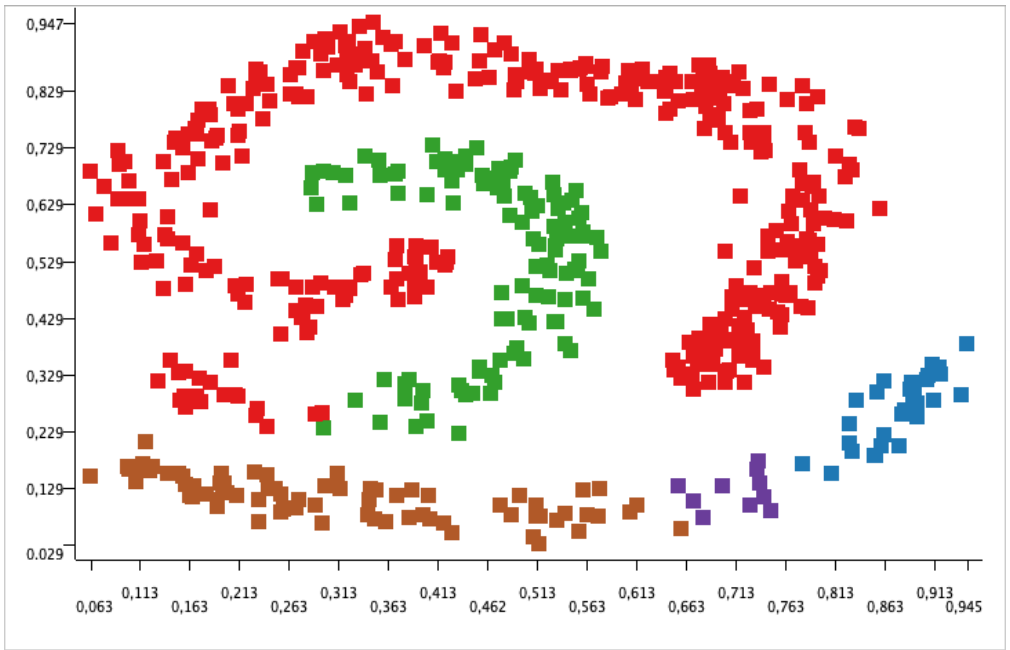

DBSCAN, epsilon=1
Veličina 2
Veličina 1
Noise


DBSCAN, epsilon=3
Veličina 2
Veličina 1
Cluster_0
Noise


DBSCAN, epsilon=5
Veličina 2
Veličina 1
Cluster_0
Noise

Program v KNIME

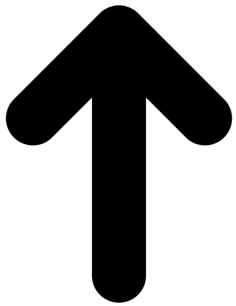


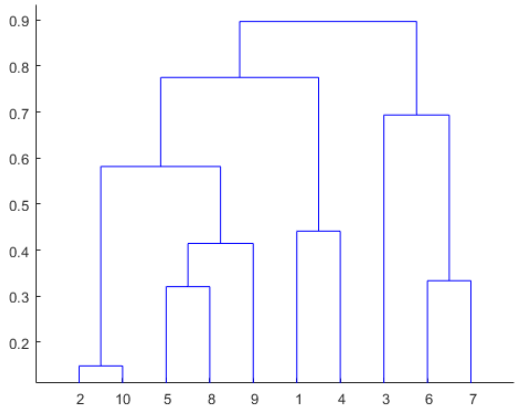


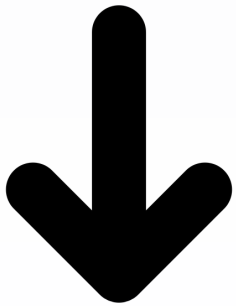
Poznámky:

- není třeba počet shluků
- libovolně tvarované shluky
- odolný vůči šumu
- obtížná volba ϵ a minPts

Hierarchické shlukování







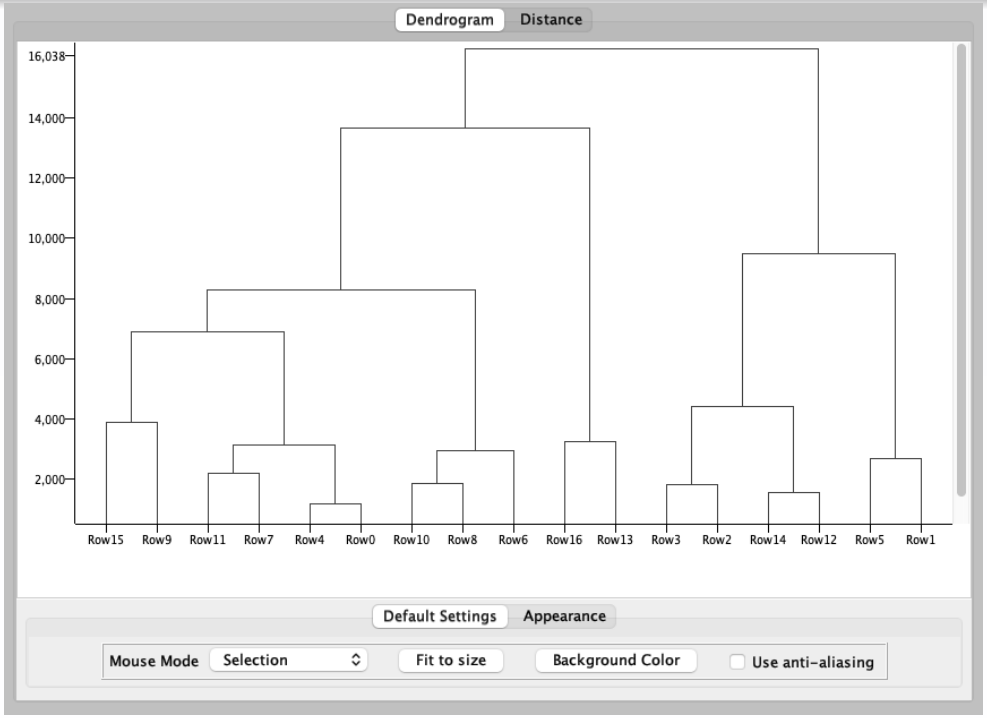
Aglomerativní – "zdola nahoru"

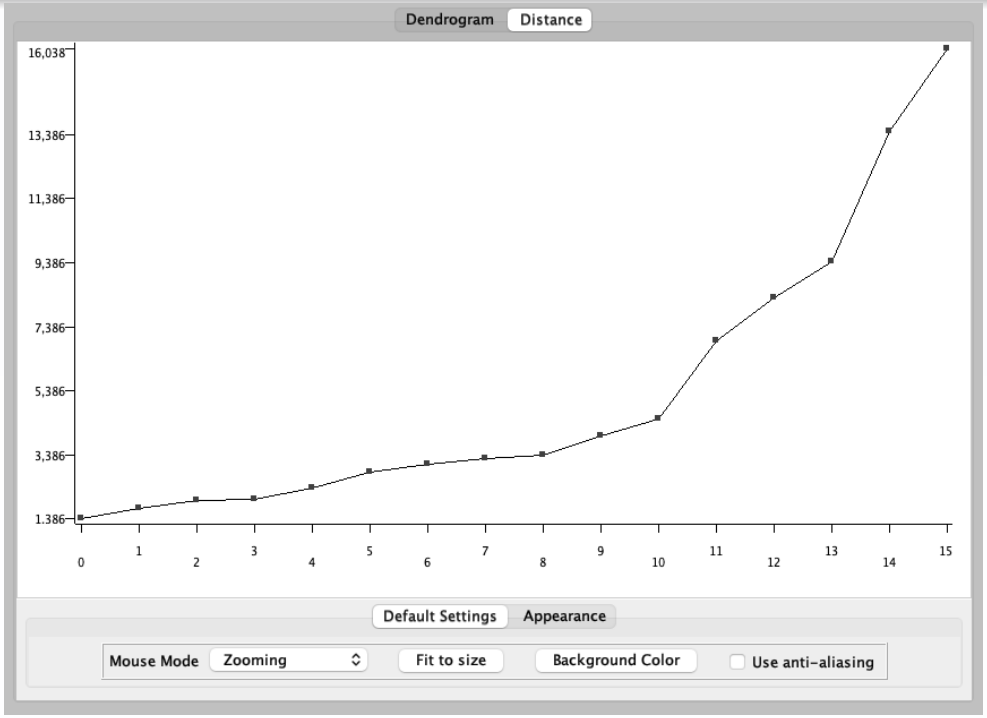
- každý bod – shluk
- podobné dvojice shluků se spojují, dokud neskončí v jednom shluku obsahujícím všechny podshluky

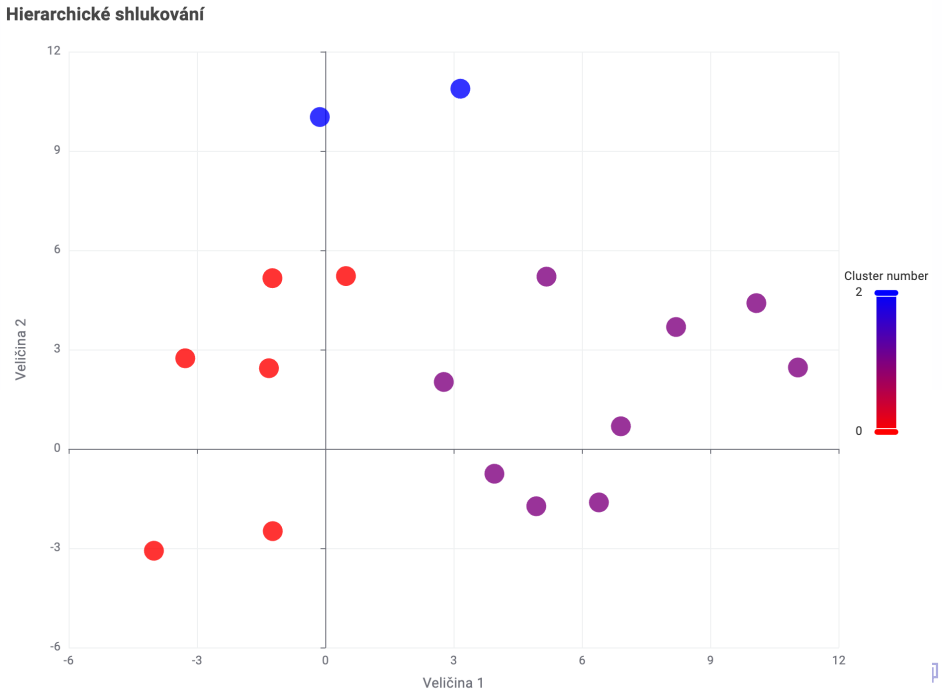
Divizivní – "shora dolů"

- všechny body – jeden shluk
- odlišné body jsou rozděleny do podshluků, dokud každý shluk neobsahuje přesně jeden datový bod

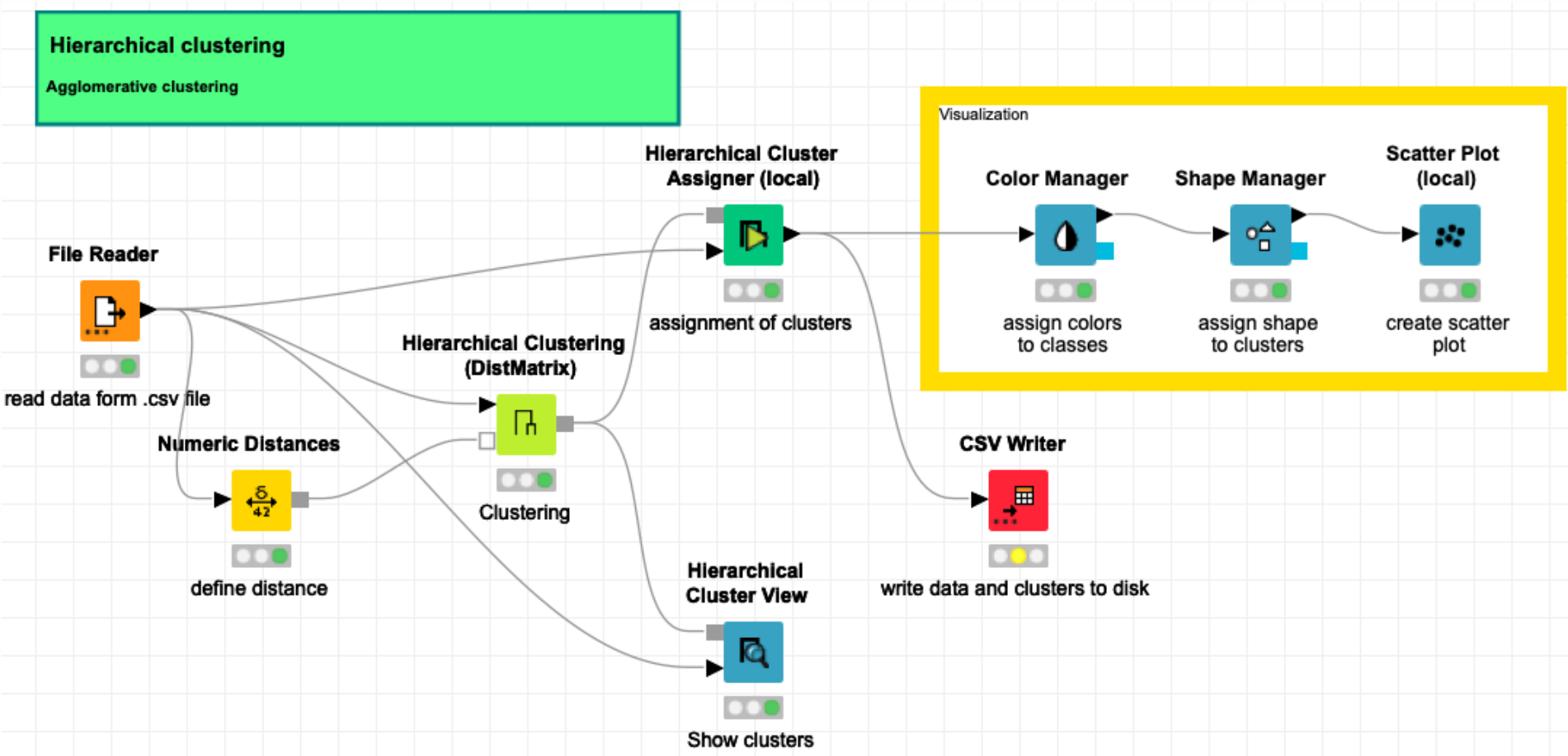
- vzdálenost mezi shluky (Euklidovská, Manhattanská, atd)
- propojení – min/max/průměrná vzdálenost mezi body ve shlucích
- hledaný počet shluků – pevný nebo práh vzdálenosti


Dendrogram
Distance
16,038
14,000
12,000
10,000
8,000
6,000
4,000
2,000
Row15 Row9 Row11 Row7 Row4 Row0 Row10 Row8 Row6 Row16 Row13 Row3 Row2 Row14 Row12 Row5 Row1
Default Settings
Appearance
Mouse Mode
Selection
Fit to size
Background Color
Use anti-aliasing


Dendrogram
Distance
16,038
13,386
11,386
9,386
7,386
5,386
3,386
1.386
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Default Settings
Appearance
Mouse Mode
Zooming
Fit to size
Background Color
Use anti-aliasing


Hierarchické shlukování
12
9
6
3
0
-3
-6
-6 -3 0 3 6 9 12
Veličina 2
Veličina 1
Cluster number
2
0

Program v KNIME



Shluky, dendrogram