

K-means clustering

Tomáš Vinař

21.11.2024

Formulácia problému

Vstup: n -rozmerné vektory x_1, x_2, \dots, x_t a počet zhlukov k

Výstup: Rozdelenie vektorov do k zhlukov:

- priradenie vstupných vektorov do zhlukov zapísané ako čísla c_1, c_2, \dots, c_t , kde $c_i \in \{1, 2, \dots, k\}$ je číslo zhuku pre x_i
- centrum každého zhuku, t.j. n -rozmerné vektory $\mu_1, \mu_2, \dots, \mu_k$

Hodnoty c_1, \dots, c_t a μ_1, \dots, μ_k volíme tak, aby sme minimalizovali súčet štvorcov vzdialeností od každého vektoru k centru jeho zhuku:

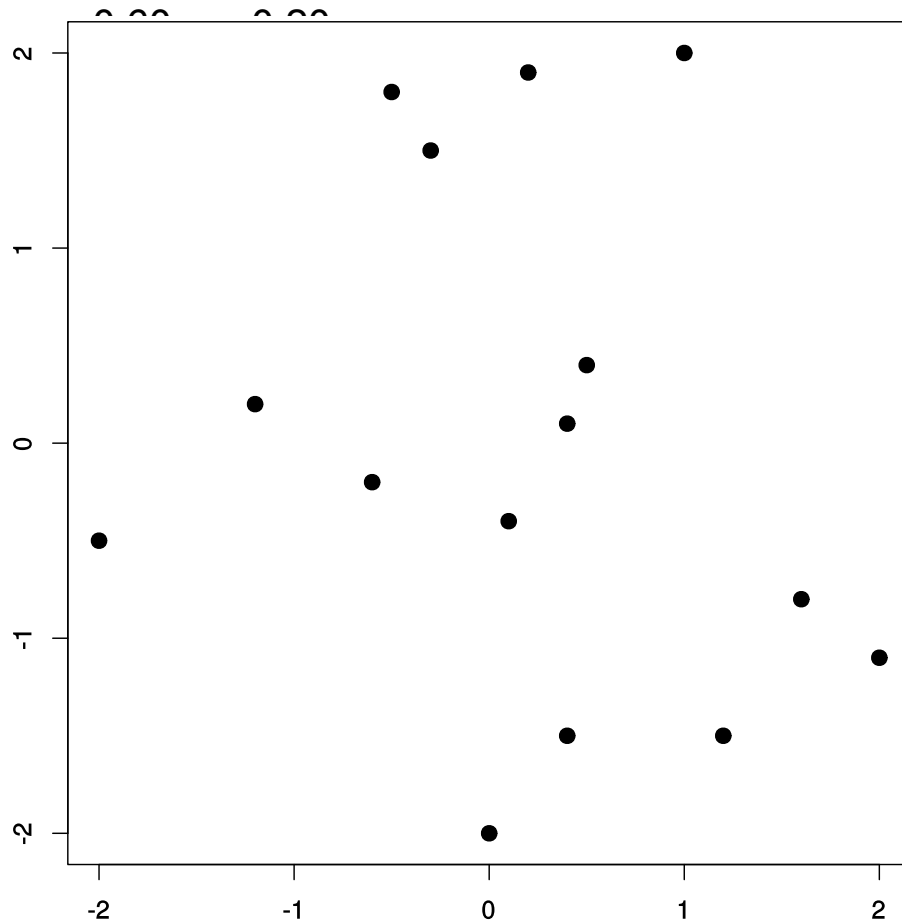
$$\sum_{i=1}^t \|x_i - \mu_{c_i}\|_2^2$$

Pre vektory $a = (a_1, \dots, a_n)$ a $b = (b_1, \dots, b_n)$ je druhá mocnina vzdialenosti $\|a - b\|_2^2 = \sum_{i=1}^n (a_i - b_i)^2$

Príklad vstupu

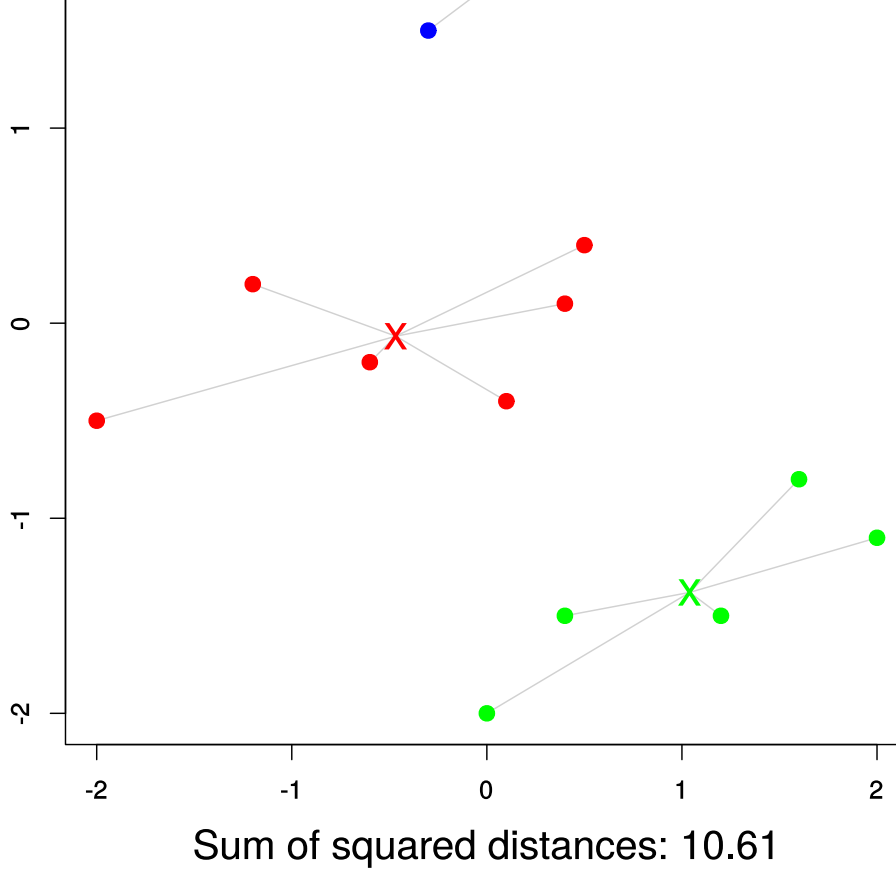
x_1 -2.00 -0.50

x_2 -1.20 0.20



Príklad výstupu

x_1	-2.00	-0.50	1
x_2	-1.20	0.20	1
x_3	-0.60	-0.20	1
x_4	-0.50	1.80	3
x_5	-0.30	1.50	3
x_6	0.00	-2.00	2
x_7	0.10	-0.40	1
x_8	0.20	1.90	3
x_9	0.40	0.10	1
x_{10}	0.40	-1.50	2
x_{11}	0.50	0.40	1
x_{12}	1.00	2.00	3
x_{13}	1.20	-1.50	2
x_{14}	1.60	-0.80	2
x_{15}	2.00	-1.10	2
μ_1	-0.47	-0.07	
μ_2	1.04	-1.38	
μ_3	0.10	1.80	



Algoritmus

Heuristika, ktorá nenájde vždy najlepšie zhlukovanie.

Začne z nejakého zhlukovania a postupne ho zlepšuje.

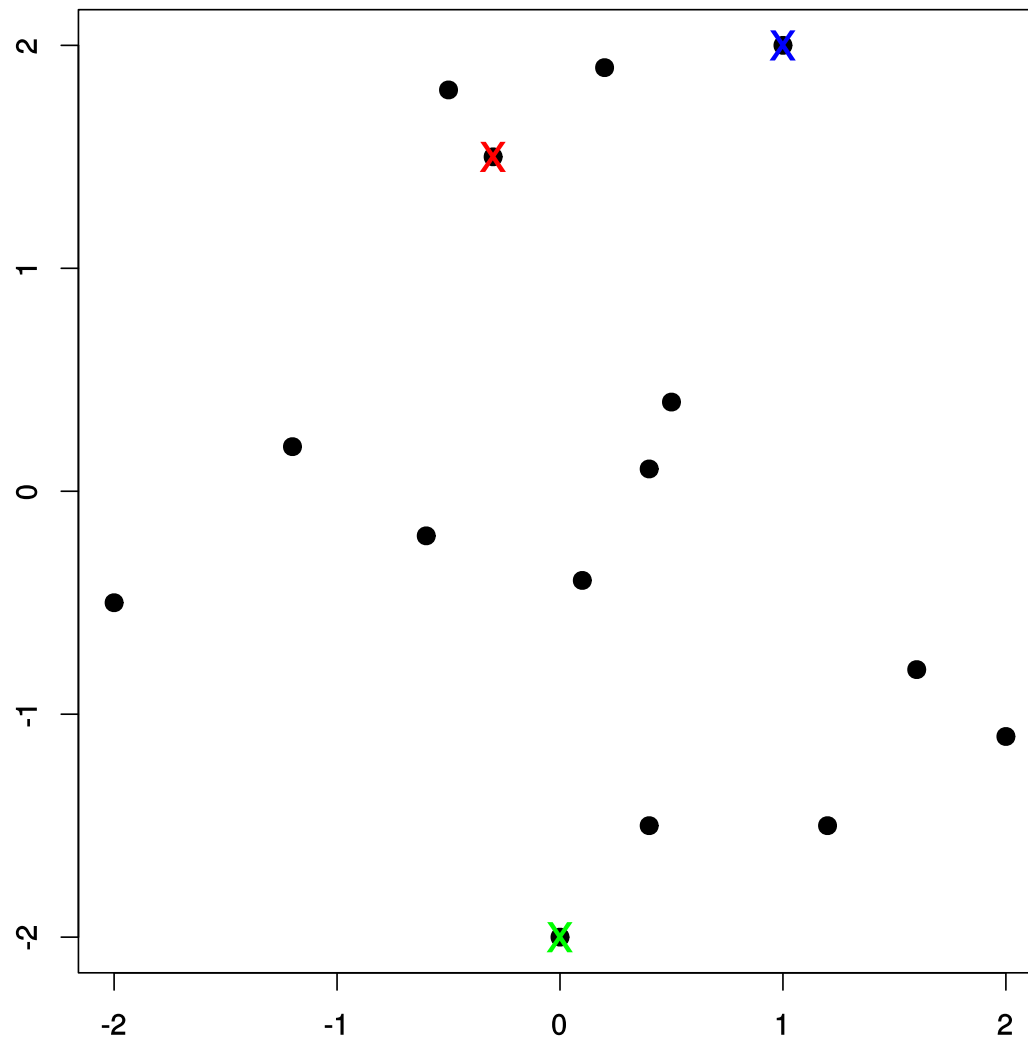
Inicializácia:

náhodne vyber k centier $\mu_1, \mu_2, \dots, \mu_k$ spomedzi vstupných vektorov

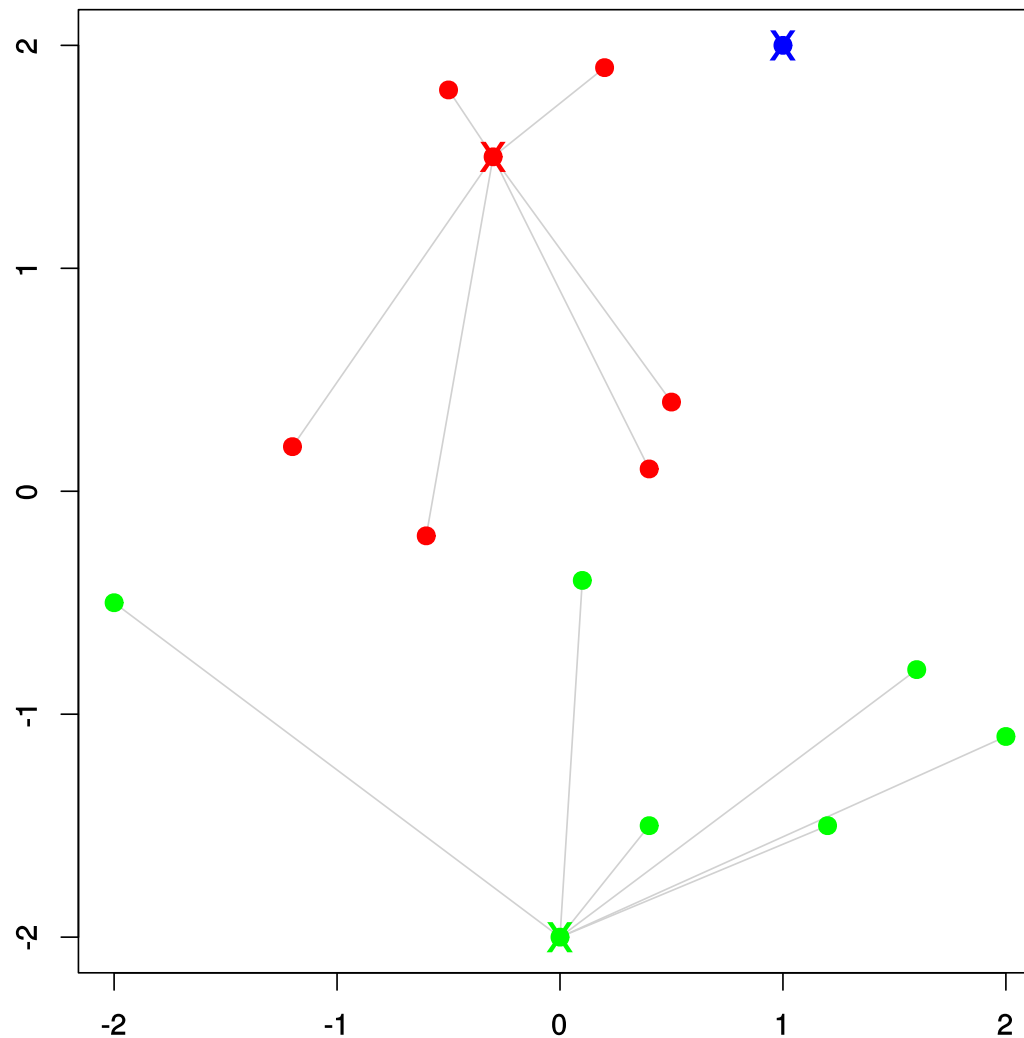
Opakuj, kým sa niečo mení:

- priradiť každý bod najbližšiemu centru: $c_i = \arg \min_j \|x_i - \mu_j\|_2$
- vypočítaj nové centrá: μ_j bude priemerom (po zložkách) z vektorov x_i , pre ktoré $c_i = j$

Zvolíme náhodné centrá μ_i

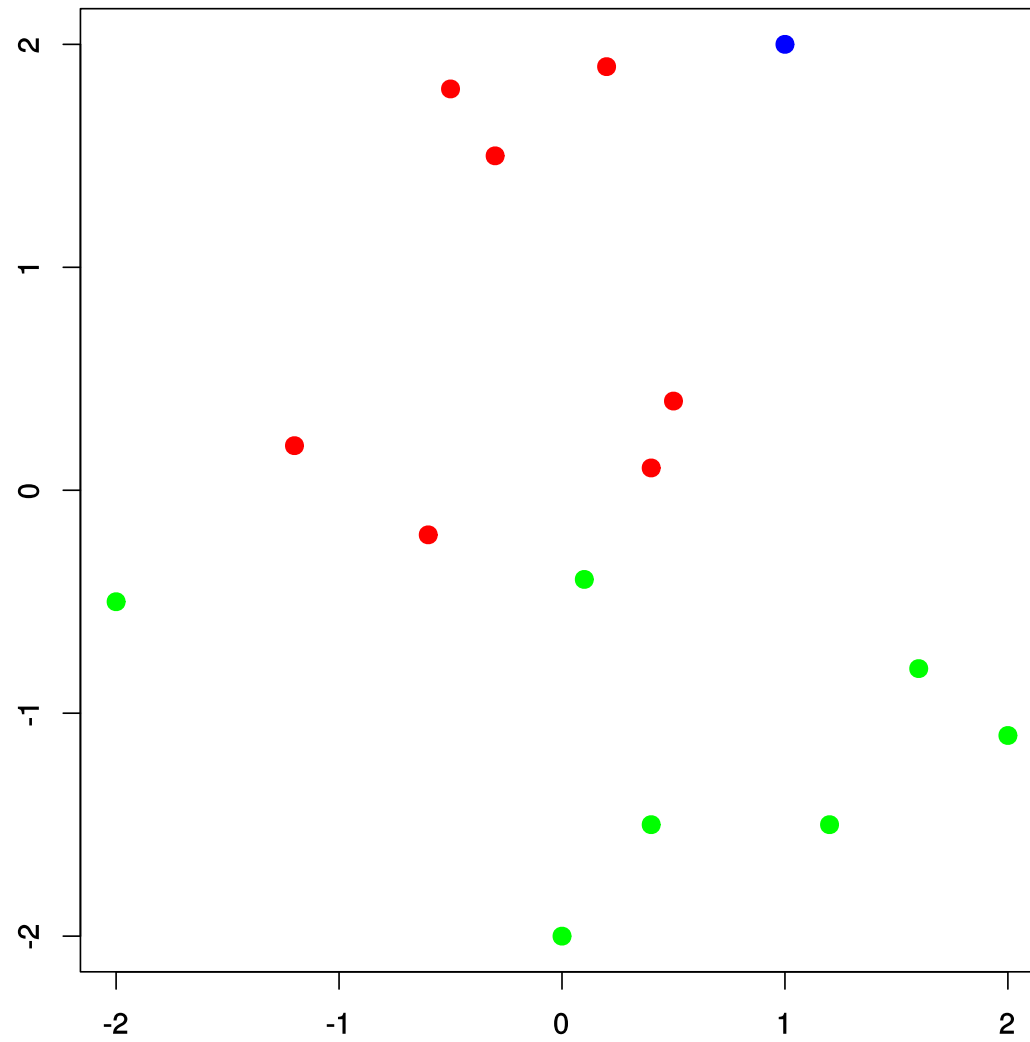


Vektory priradíme do zhlukov (hodnoty c_i)

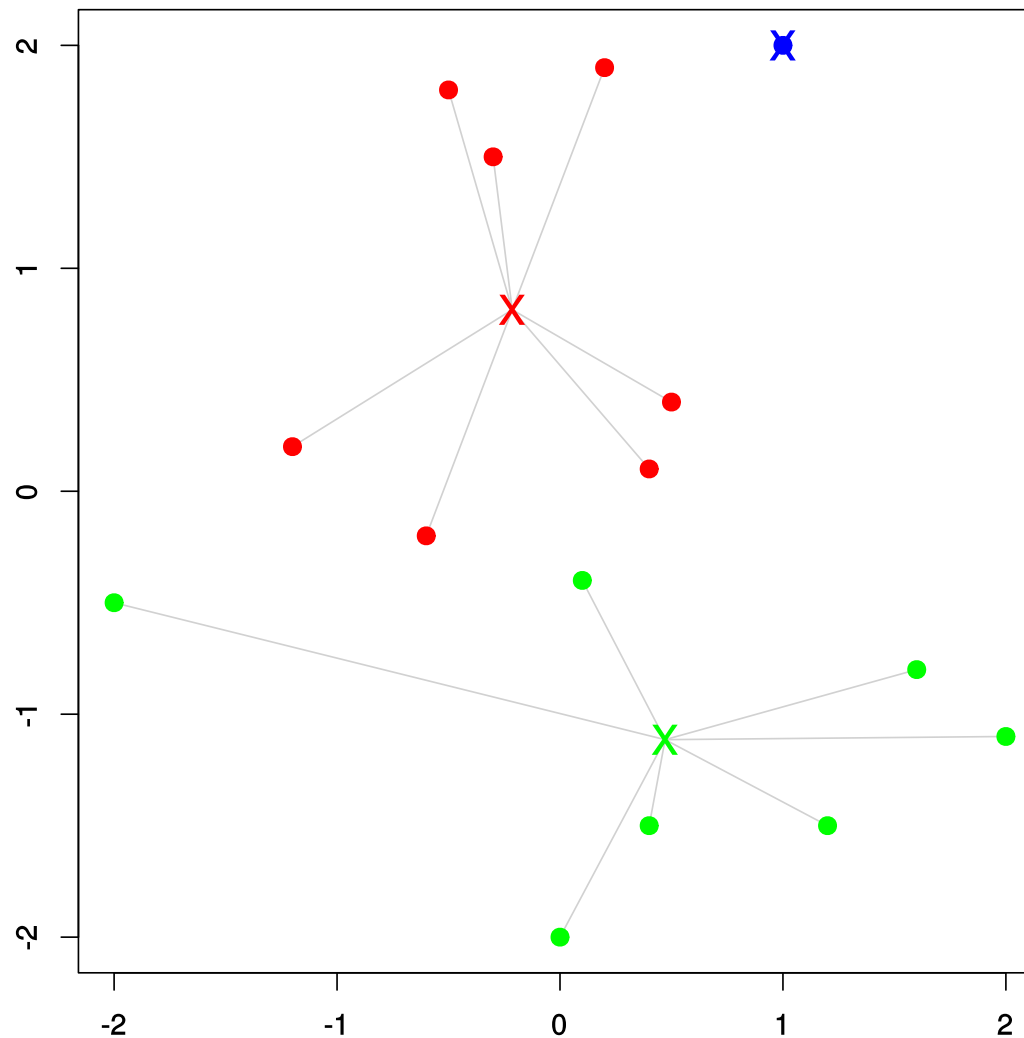


Sum of squared distances: 30.05

Zabudneme μ_i

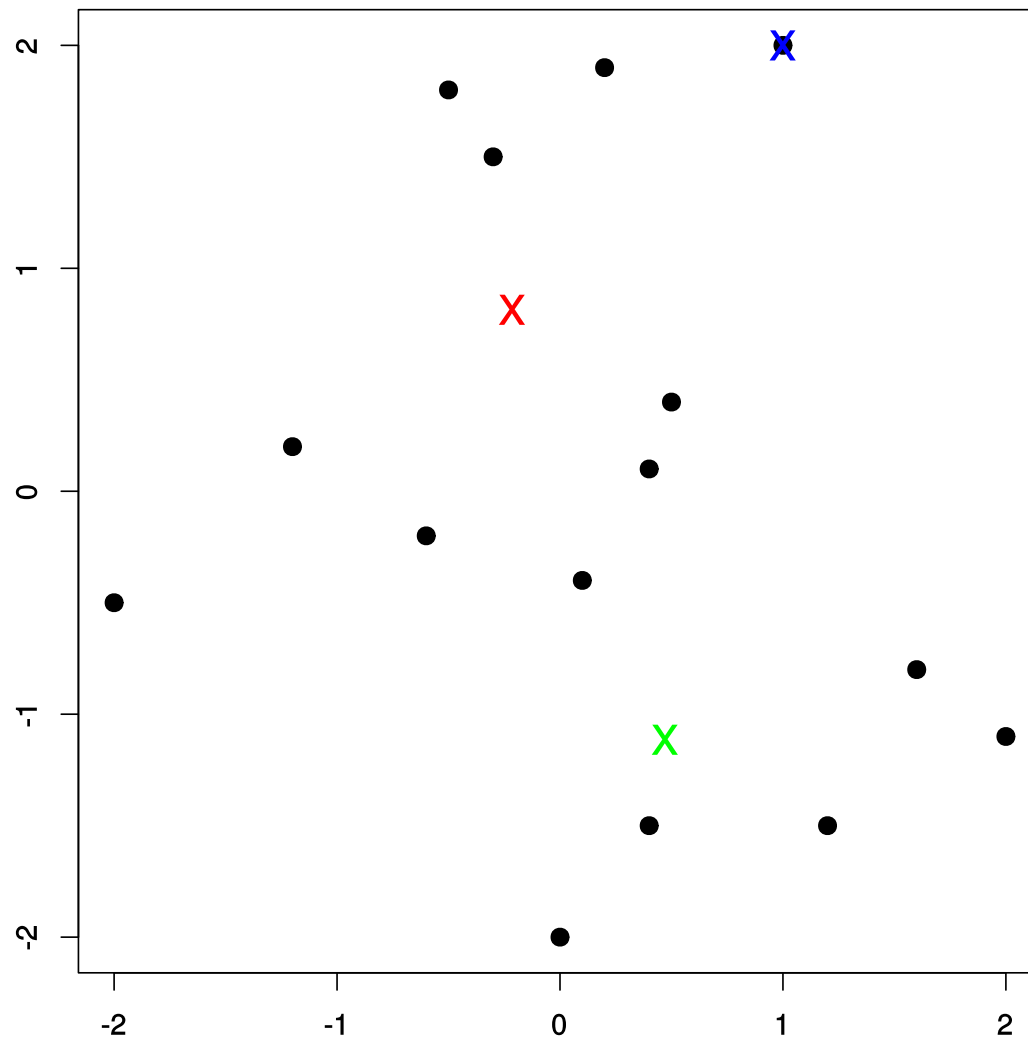


Dopočíame nové μ_i (suma klesla z 30.05 na 19.66)

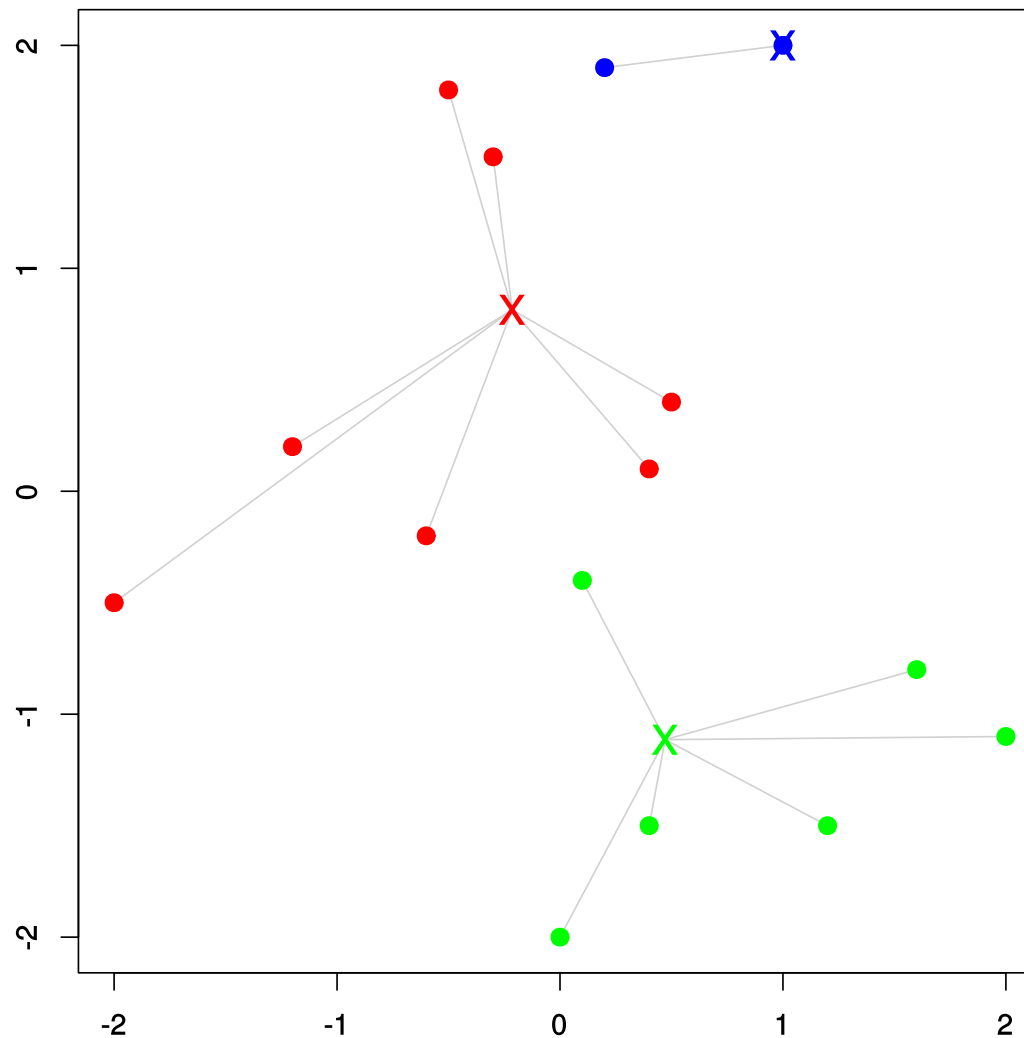


Sum of squared distances: 19.66

Zabudneme c_i

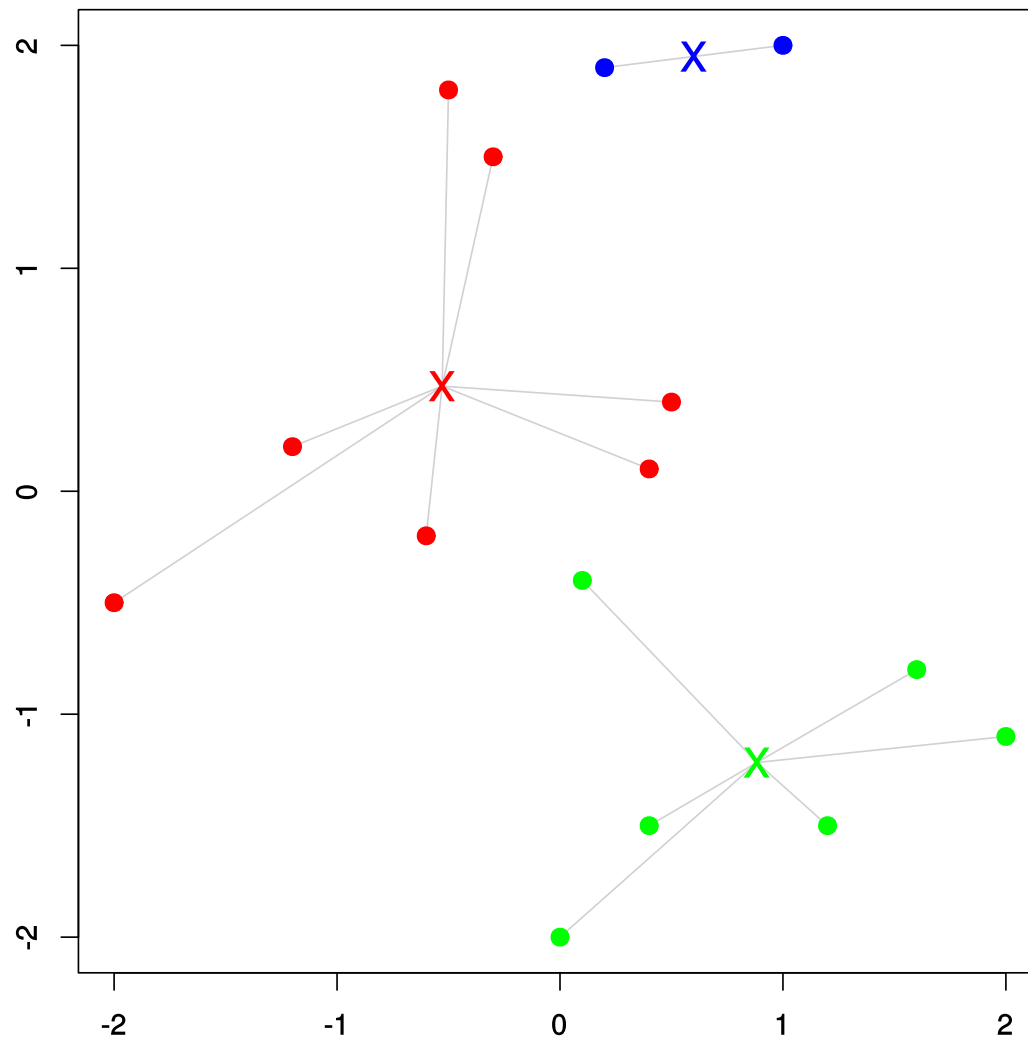


Dopočíame nové c_i (suma klesla z 19.66 na 17.39)



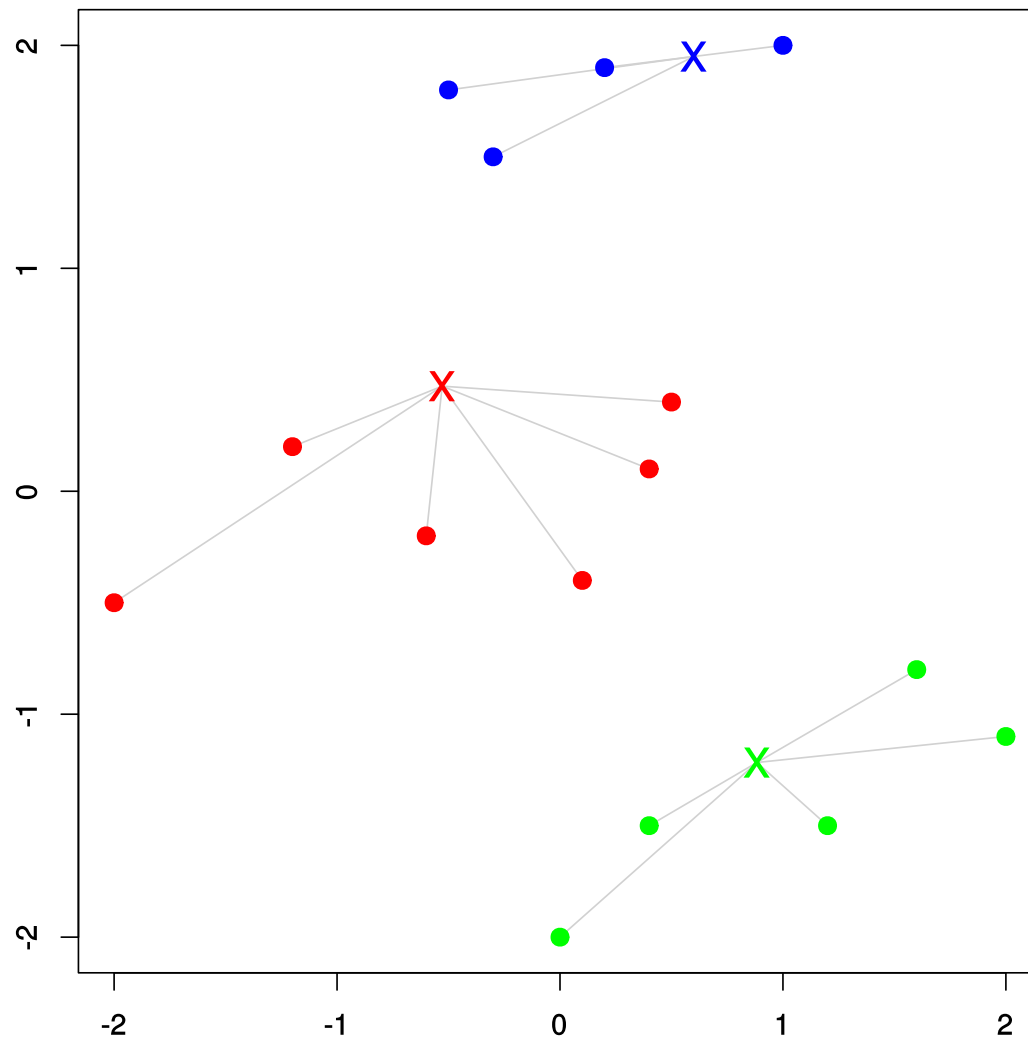
Sum of squared distances: 17.39

Prepočítame μ_i



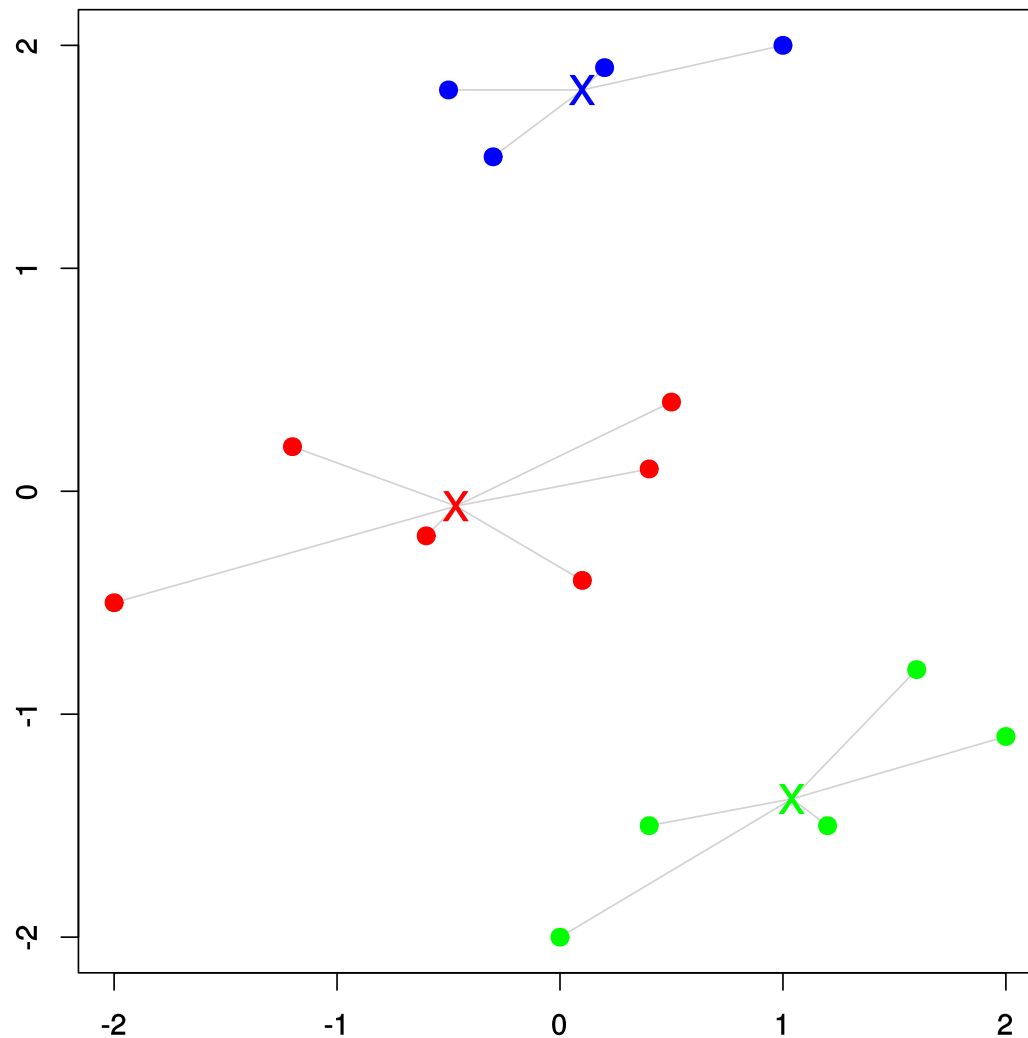
Sum of squared distances: 14.47

Prepočítame C_i



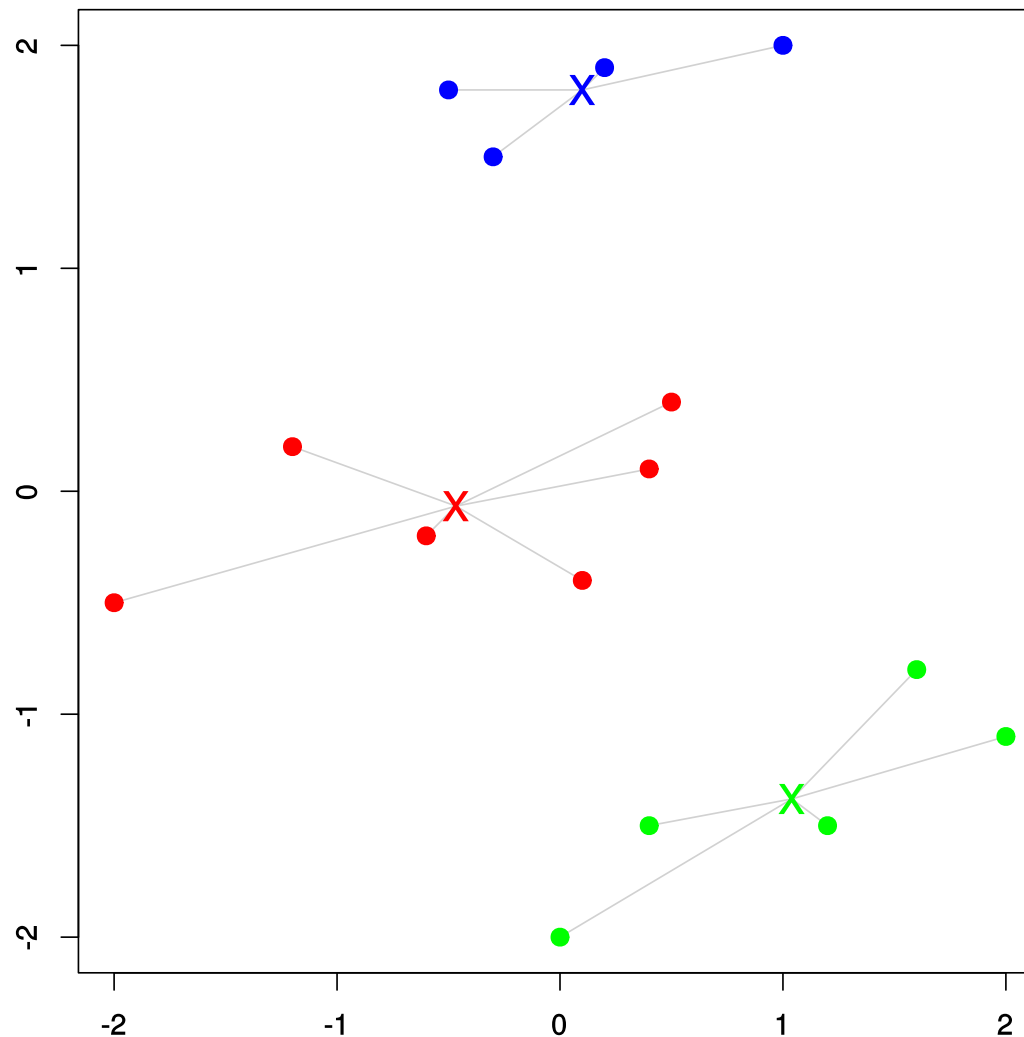
Sum of squared distances: 13.71

Prepočítame μ_i



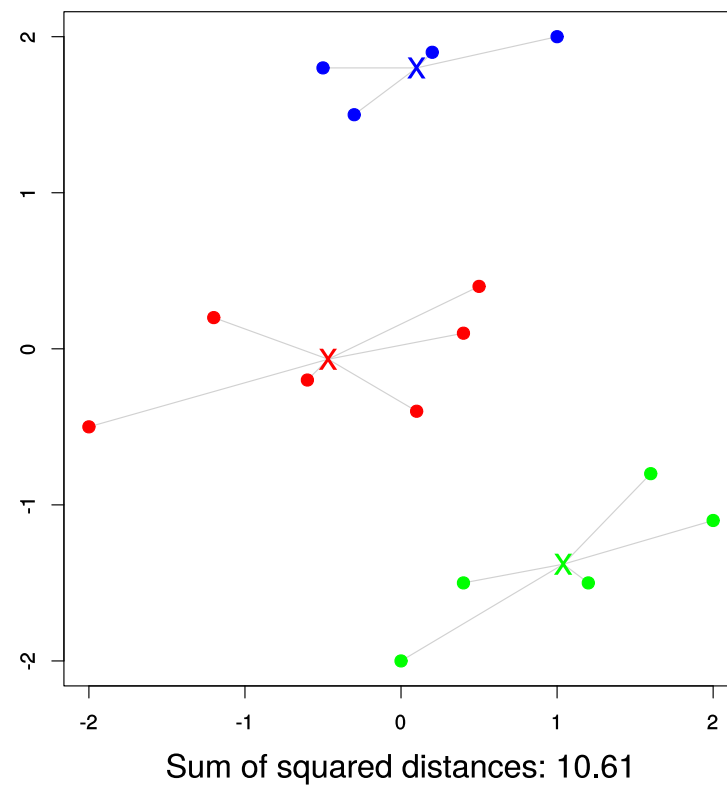
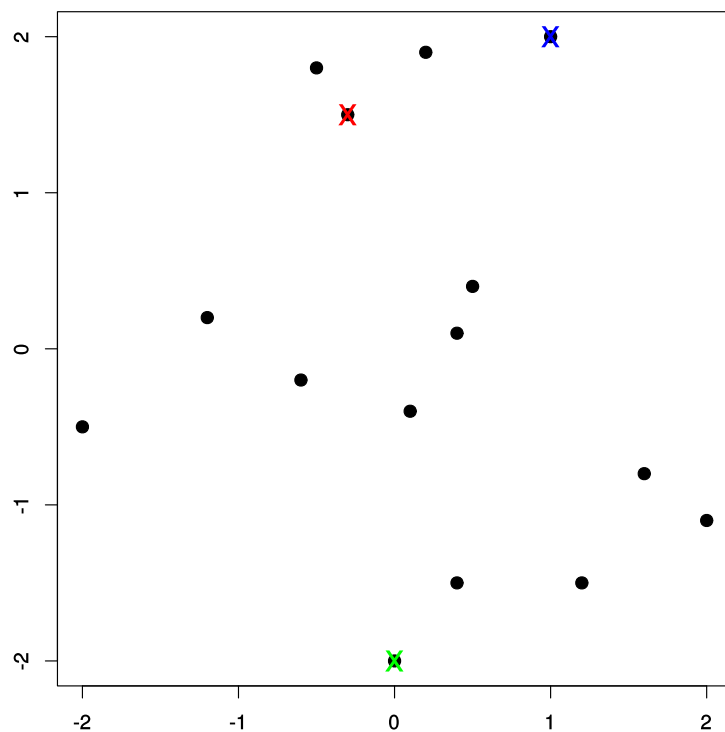
Sum of squared distances: 10.61

Prepočítame c_i (žiadna zmena, končíme)

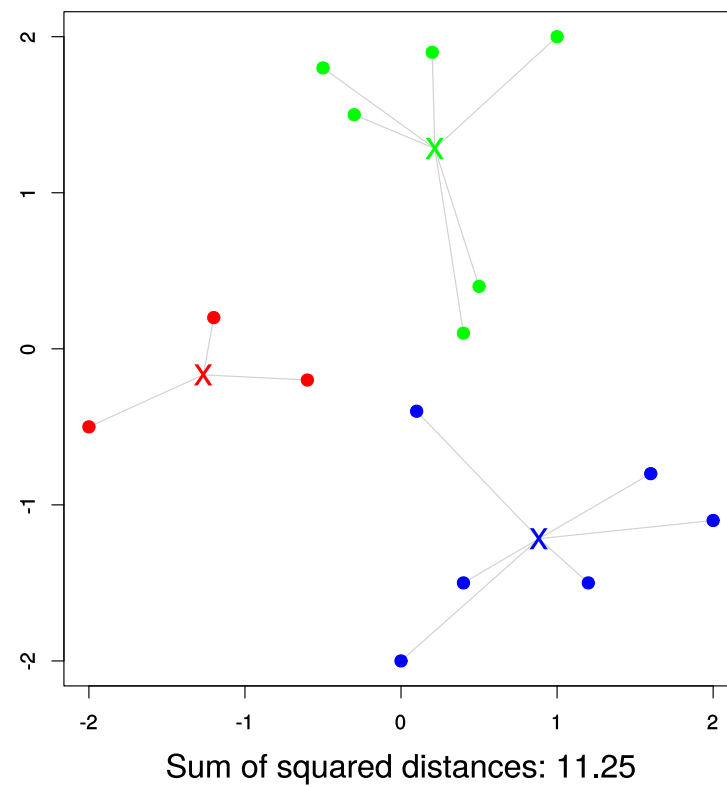
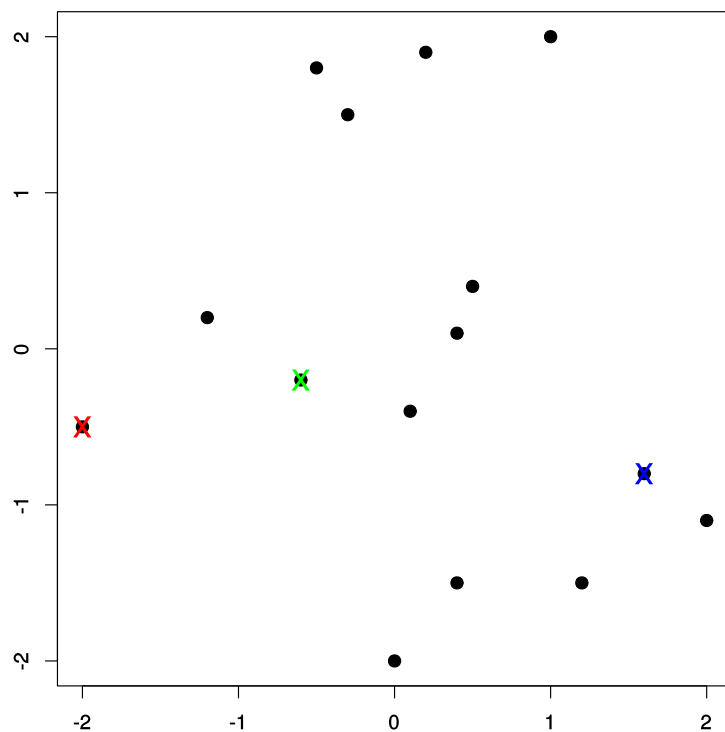


Sum of squared distances: 10.61

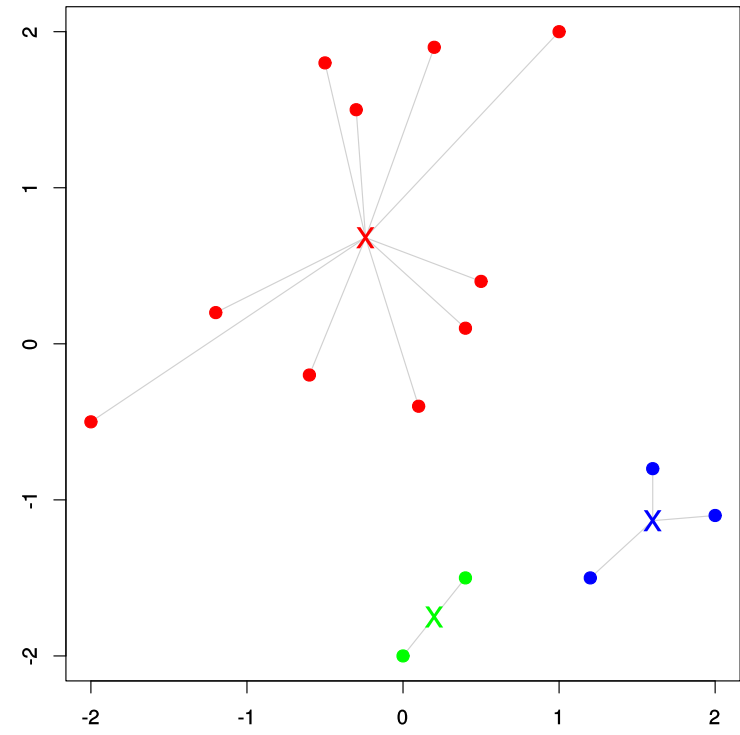
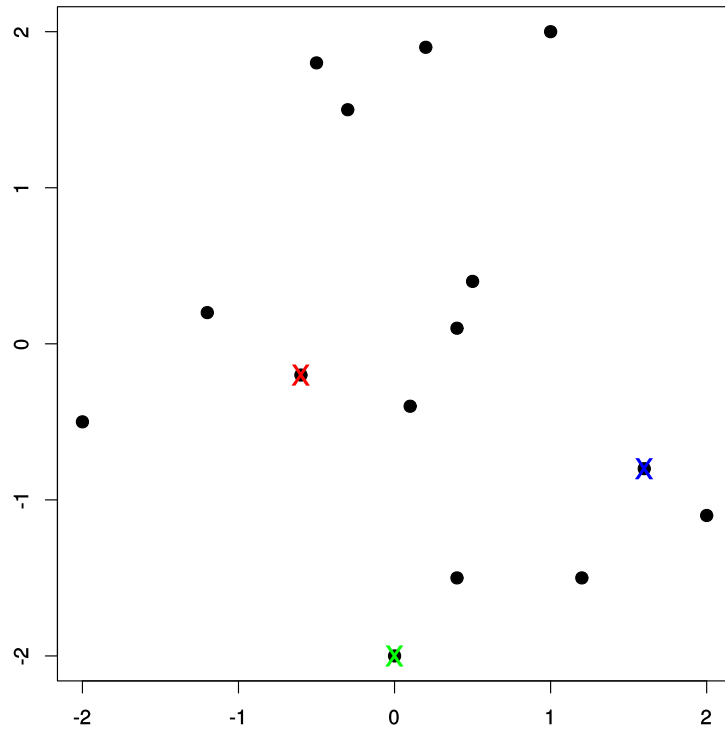
Príklady niekoľkých behov programu



Príklady niekoľkých behov programu



Príklady niekoľkých behov programu



Sum of squared distances: 16.93

Príklady niekoľkých behov programu

