k-means Clustering: Problem Formulation

Input: n-dimensional data points $x_1, x_2, \ldots, x_t$, number of clusters k

Output: Division of all data points into k clusters:

- $c_1, c_2, \ldots, c_t$, $c_i \in \{1, 2, \ldots, k\}$ is the number of a cluster to which x_i is assigned to
- n-dimensional vectors $\mu_1, \mu_2, ..., \mu_k$, where μ_j is the center of j-th cluster

Values $c_1, \ldots, c_t$ and $\mu_1, ..., \mu_k$ are chosen to minimise:

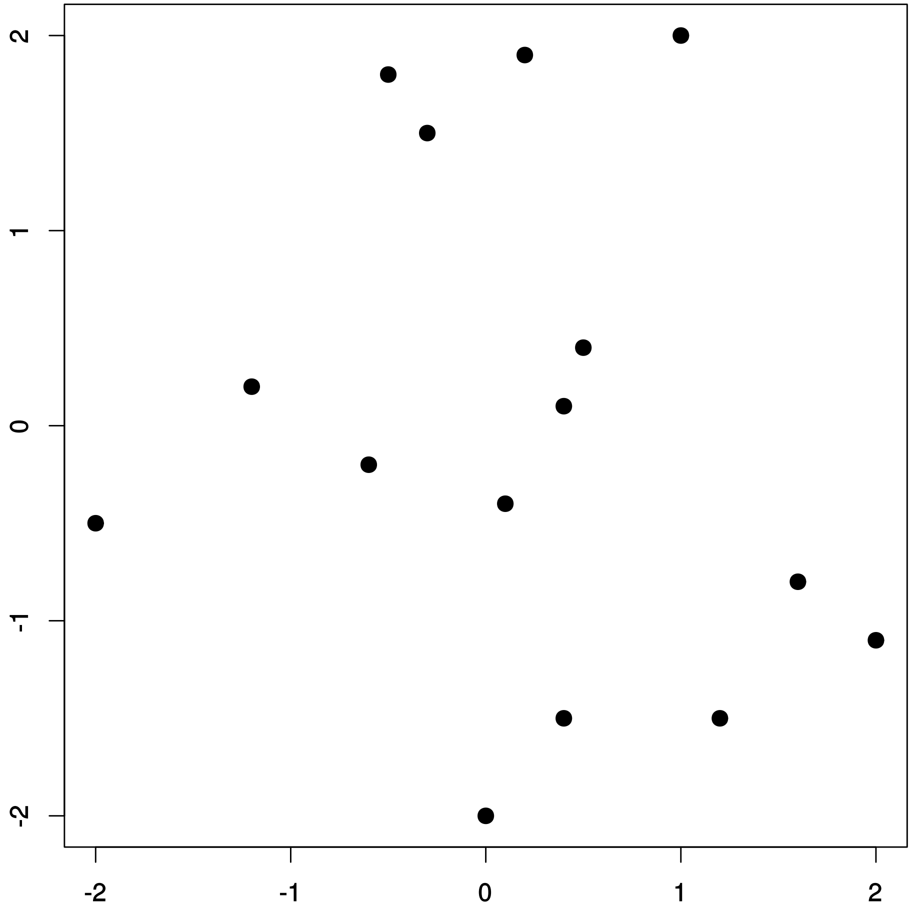
$$J(c, \mu) = \sum_{i=1}^{t} \left\| x_i - \mu_{c_i} \right\|_2^2,$$

For vectors $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots b_n)$, the square of their distance is $\left\| a - b \right\|_2^2 = \sum_{i=1}^{n} (a_i - b_i)^2$

Input example

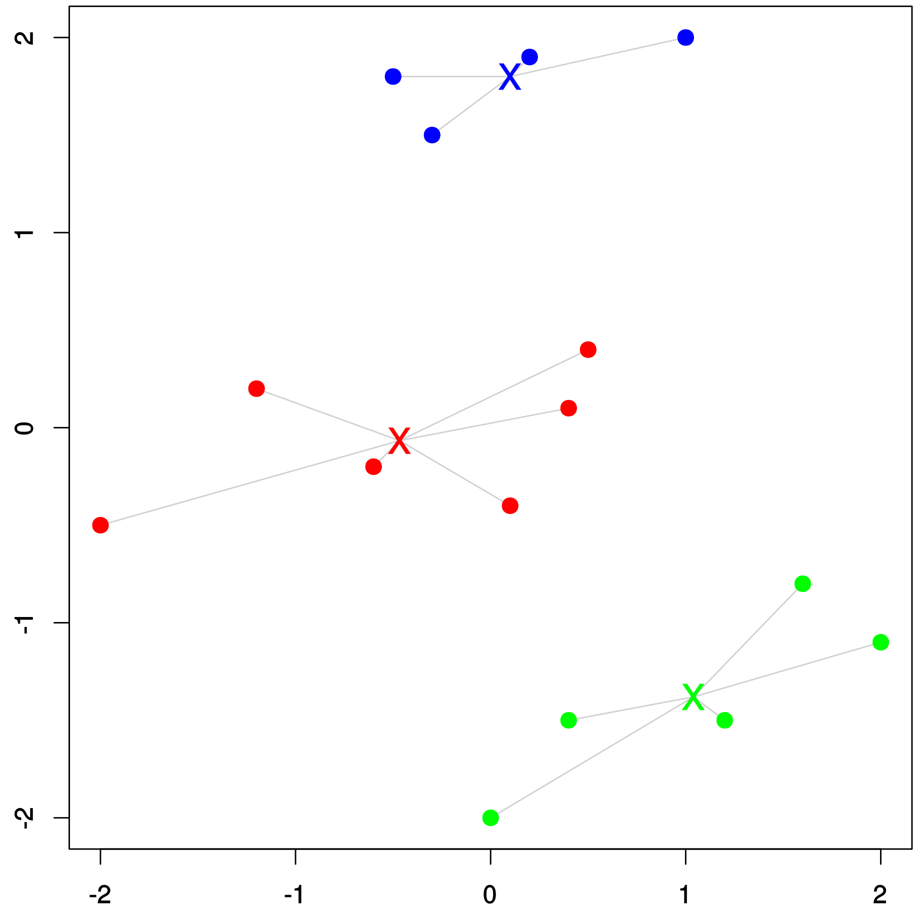
x_1	-2.00	-0.50
x_2	-1.20	0.20
x_3	-0.60	-0.20
x_4	-0.50	1.80
x_5	-0.30	1.50
x_6	0.00	-2.00
x_7	0.10	-0.40
x_8	0.20	1.90
x_9	0.40	0.10
x_{10}	0.40	-1.50
x_{11}	0.50	0.40
x_{12}	1.00	2.00
x_{13}	1.20	-1.50
x_{14}	1.60	-0.80
x_{15}	2.00	-1.10

$k = 3$



Output example

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**	



Sum of squared distances: 10.61

k-means Algorithm

Heuristics that does not always find the best clustering.

We start from an initial clustering and iteratively improve it.

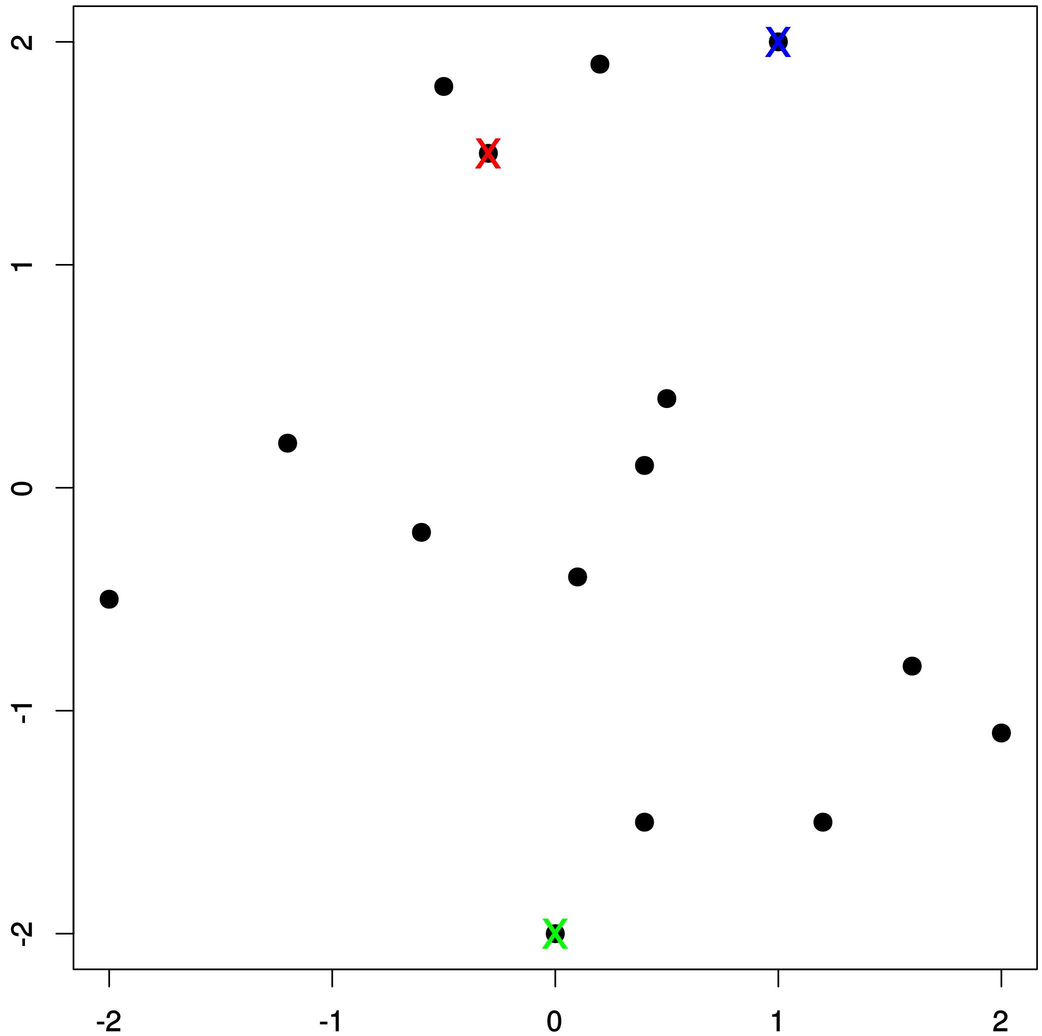
Initialization:

choose k centers $\mu_1, \mu_2, ..., \mu_k$ randomly out of the input data points

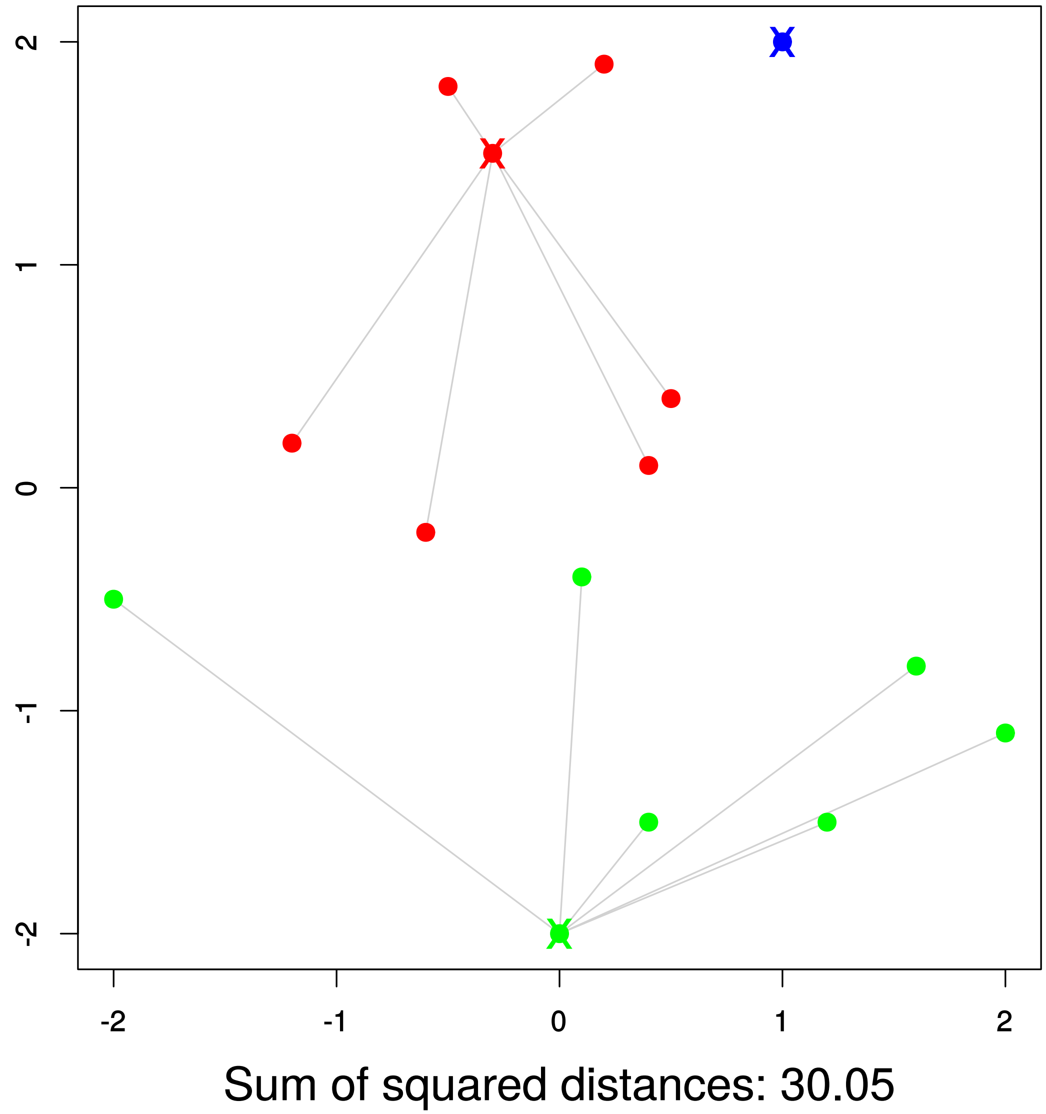
Repeat until convergence:

- assign each data point to the nearest center:
  $c_i = \arg\min_j \left\|x_i - \mu_j\right\|_2$
- computer new centers: μ_j will be average of x_i, for which $c_i = j$

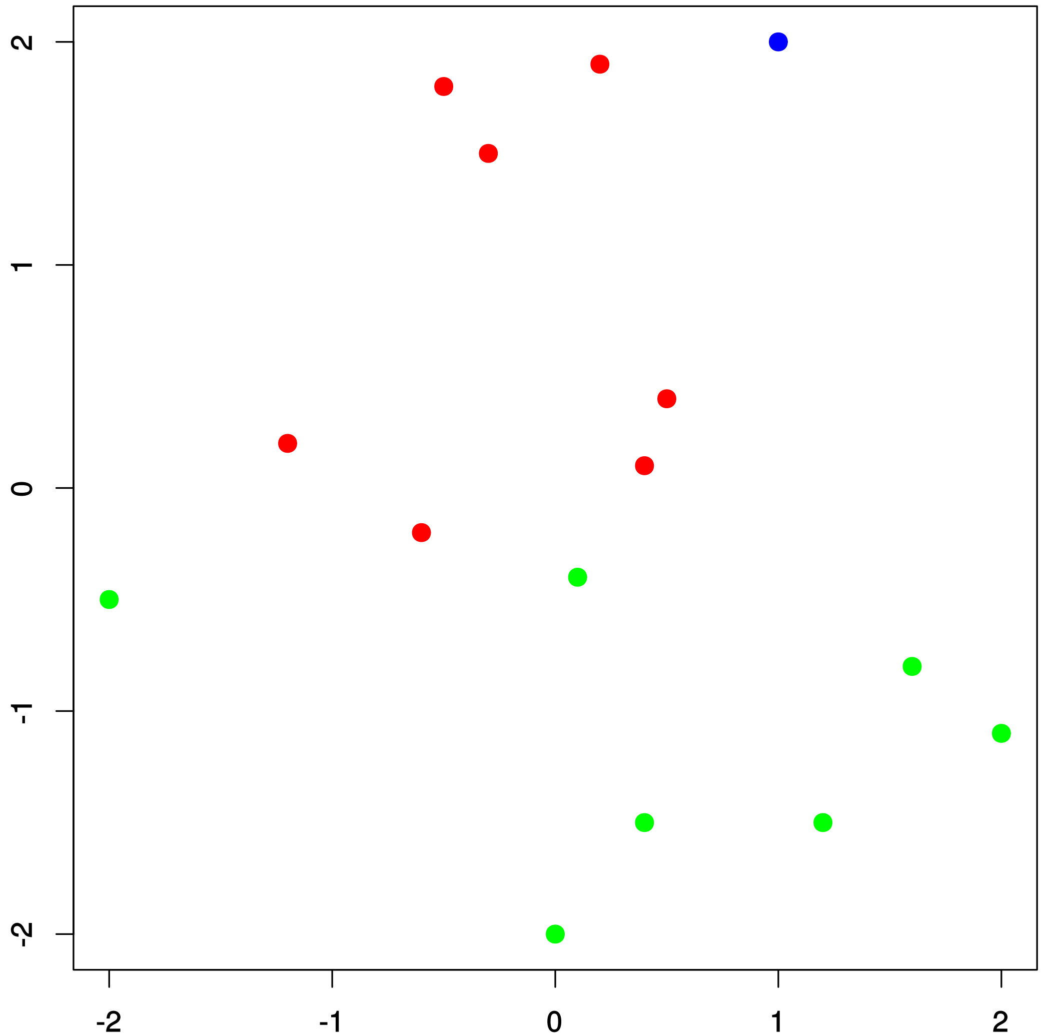
Choose random centers μ_i



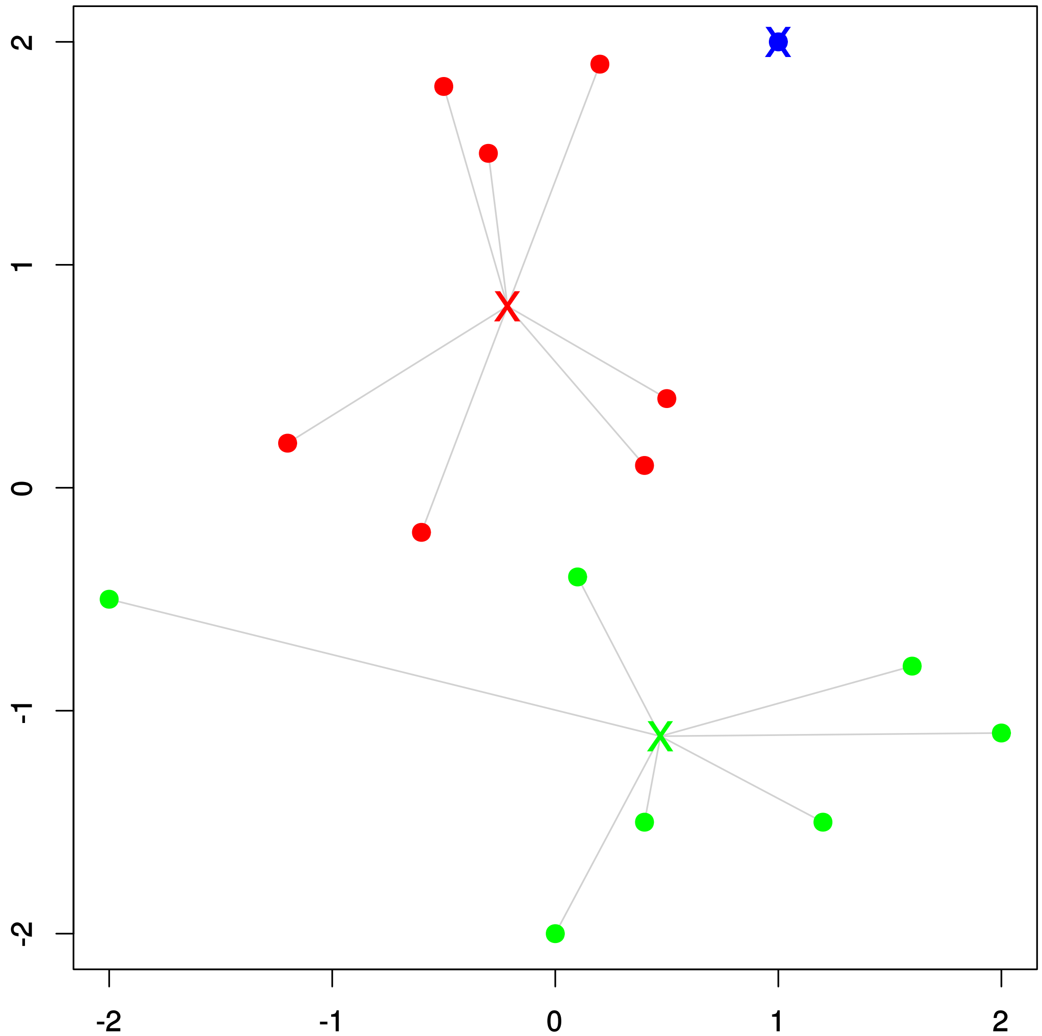
Assign data points to clusters (values c_i)



Forget μ_i

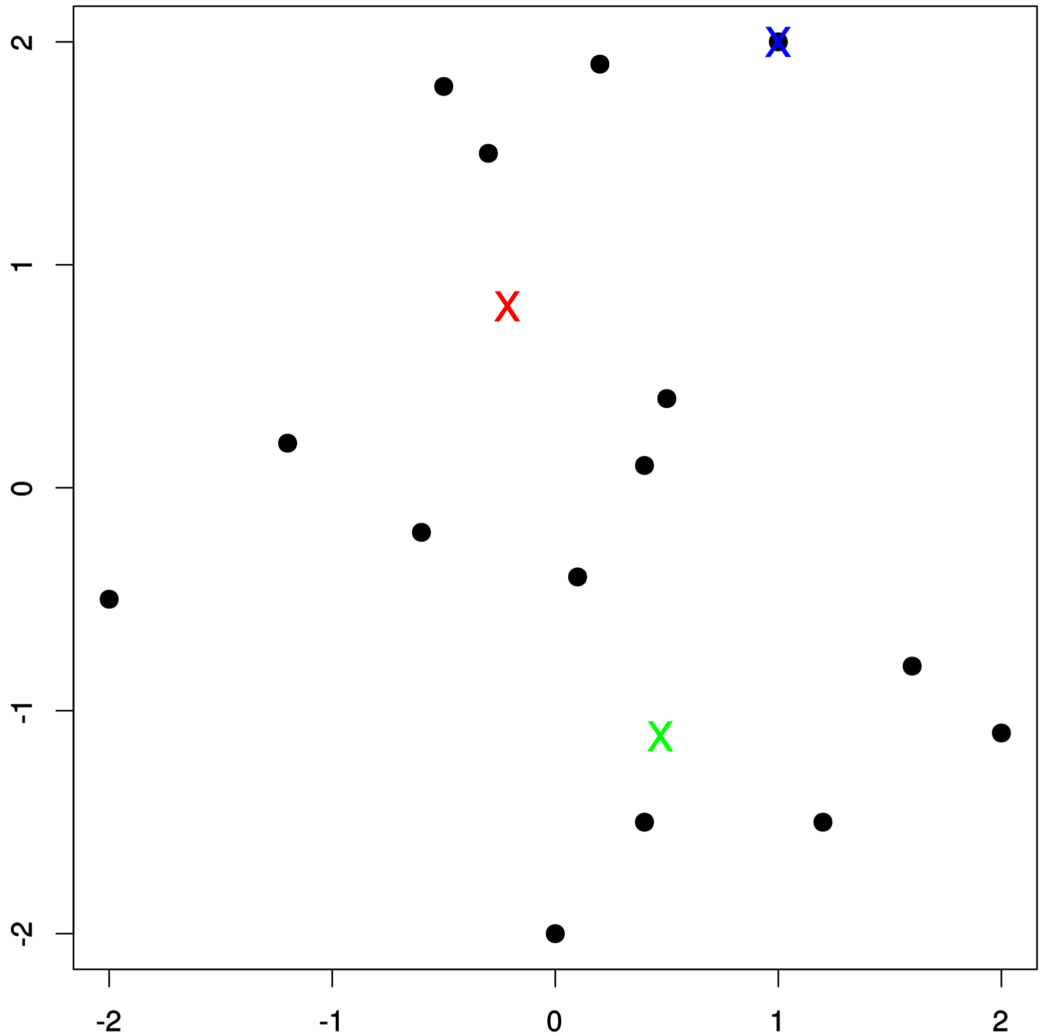


Compute new μ_i (the error decreases from 30.05 to 19.66)

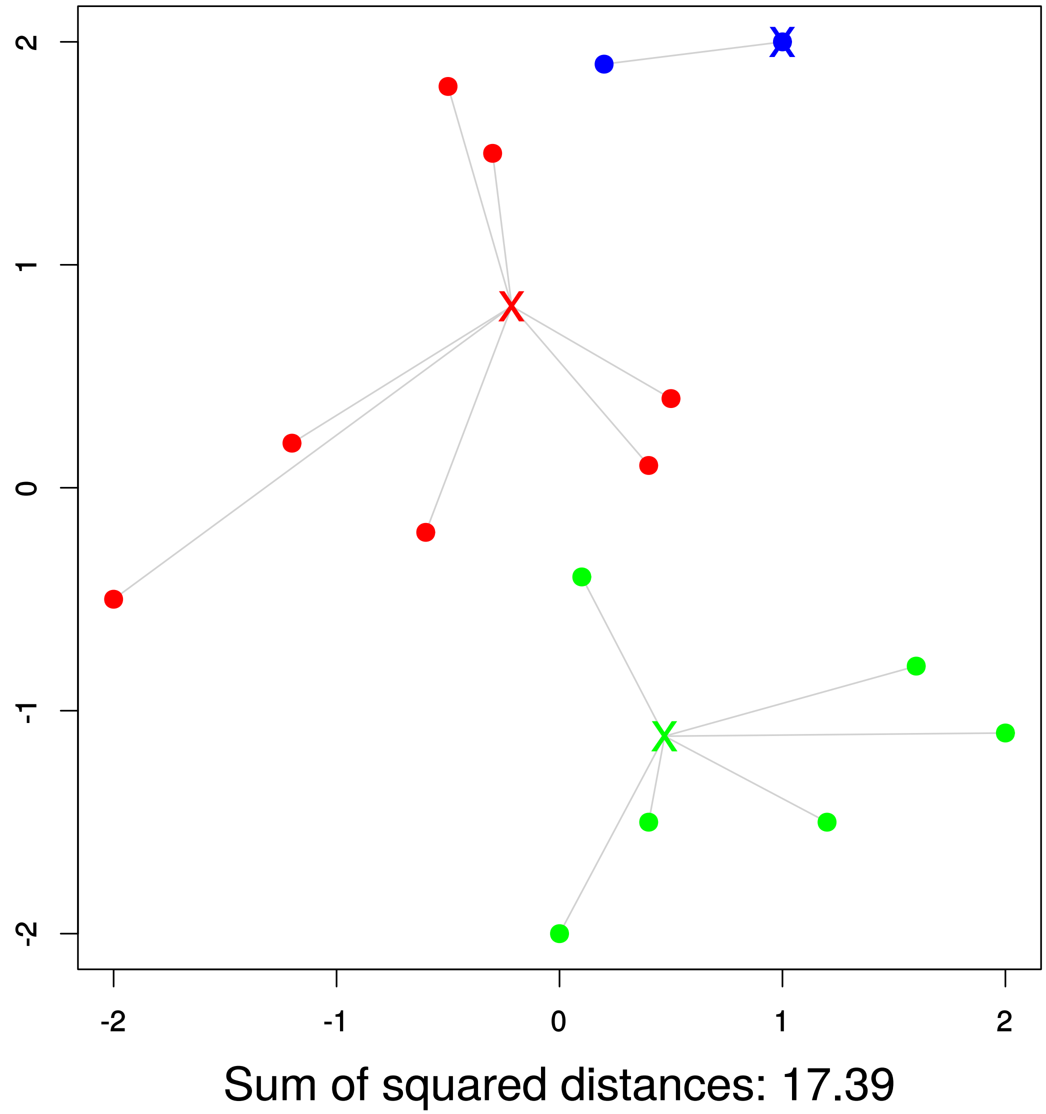


Sum of squared distances: 19.66

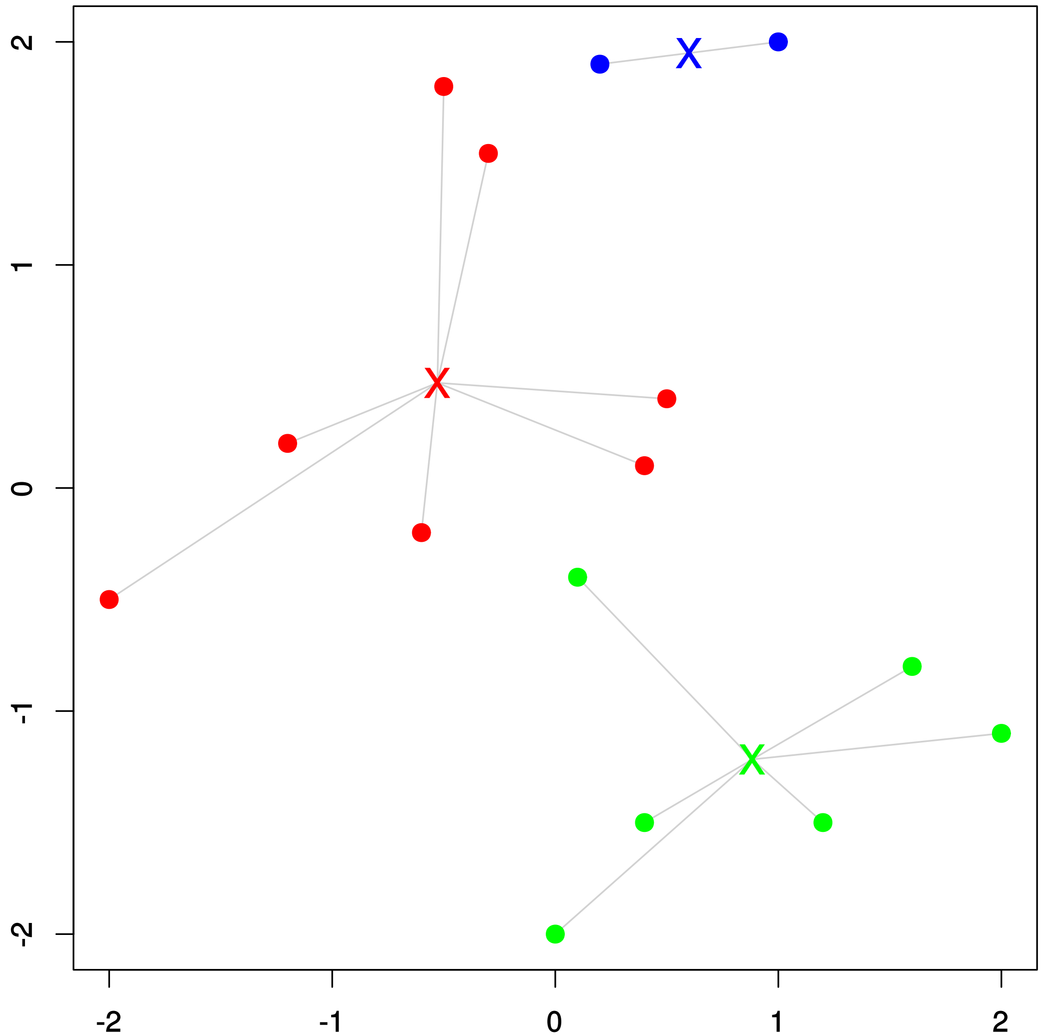
Forget c_i



Compute new c_i (the error decreases from 19.66 to 17.39)

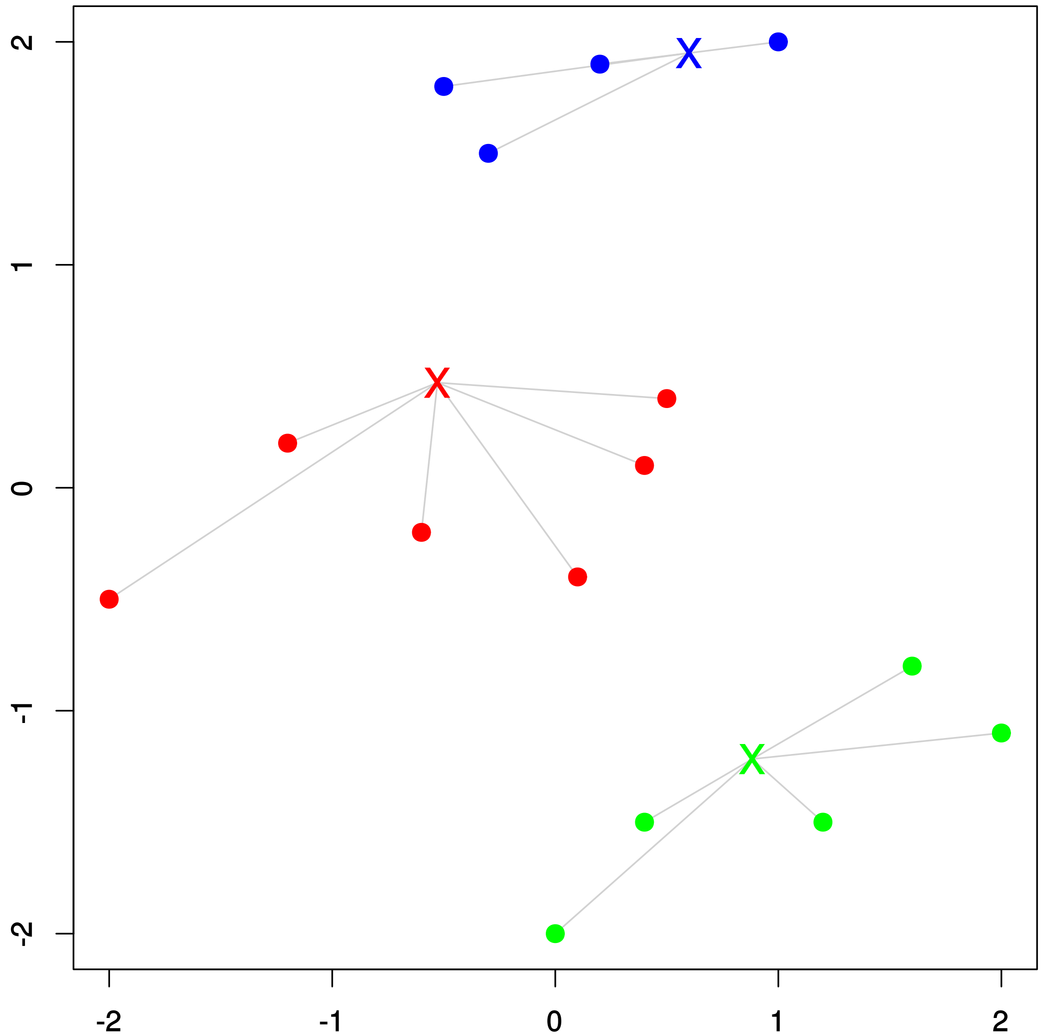


Recompute μ_i



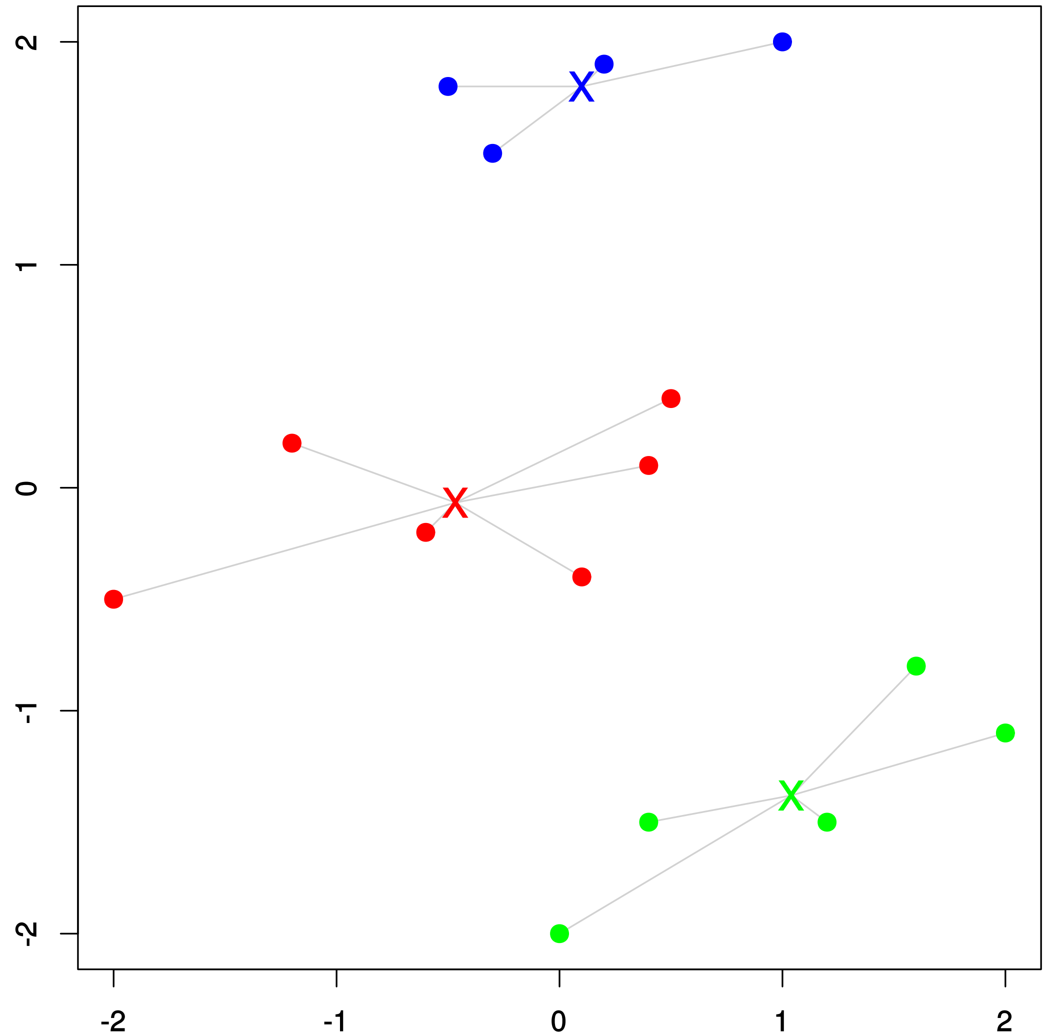
Sum of squared distances: 14.47

Recompute c_i



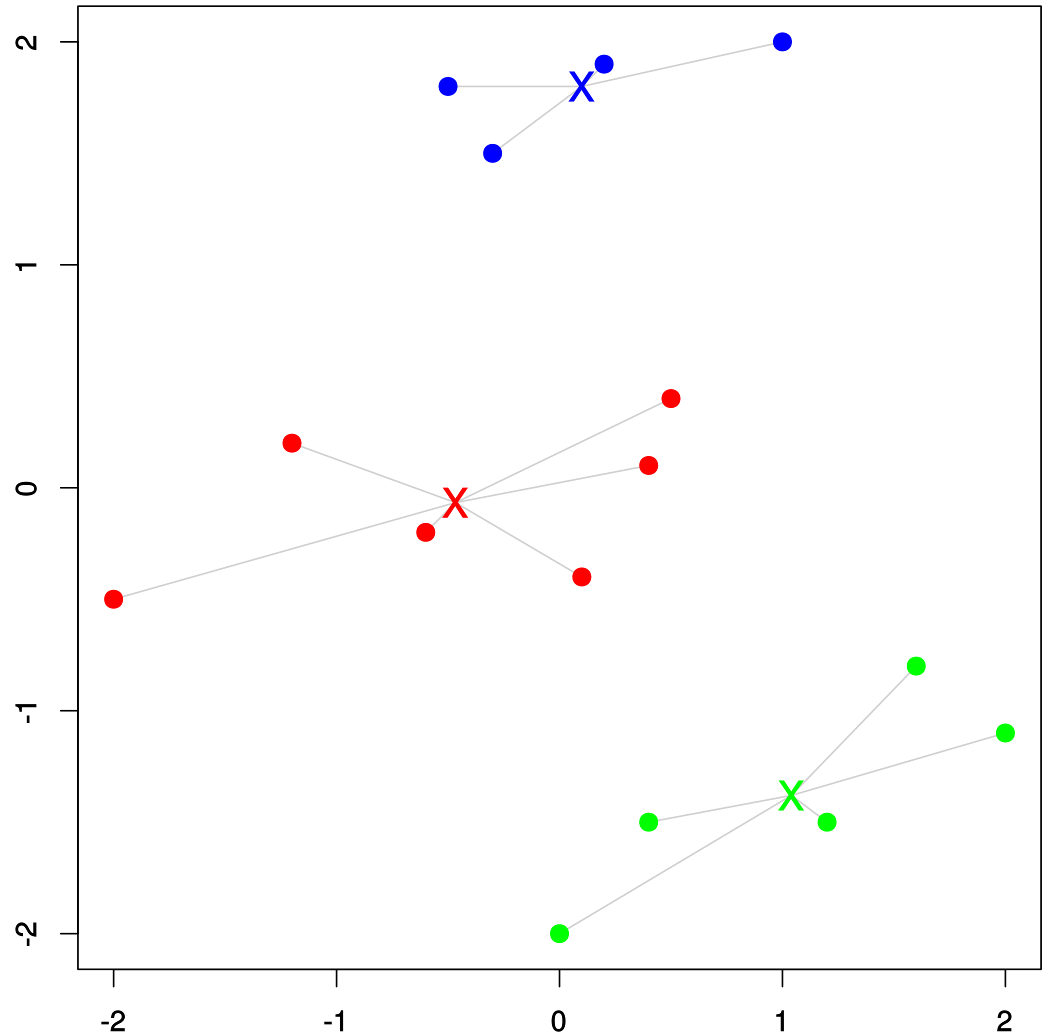
Sum of squared distances: 13.71

Recompute μ_i



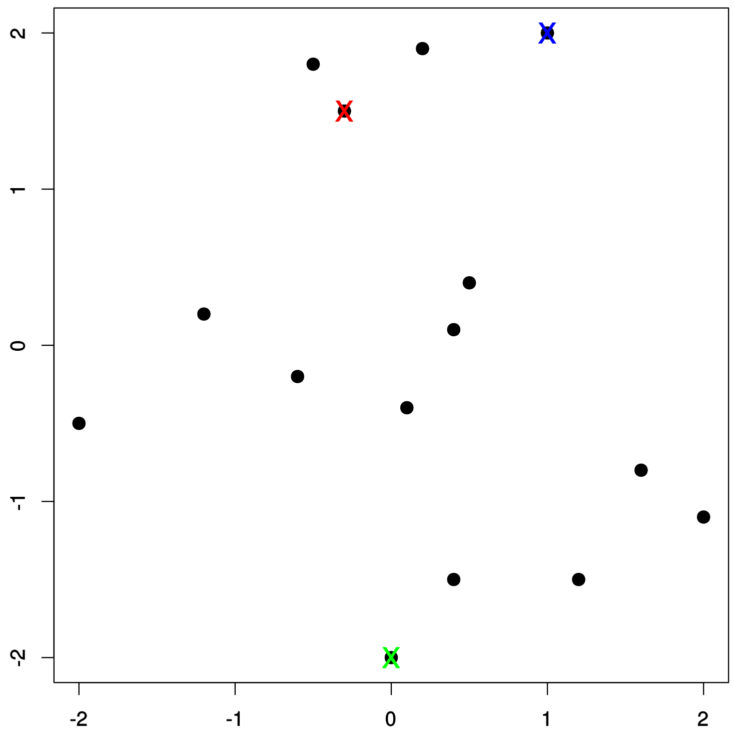
Sum of squared distances: 10.61

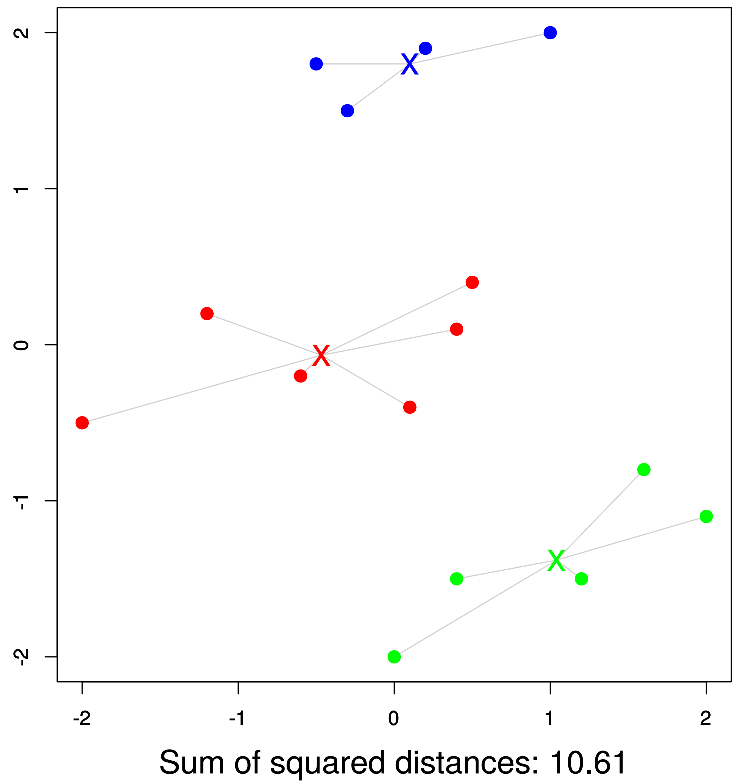
Recompute c_i (no change, finished)



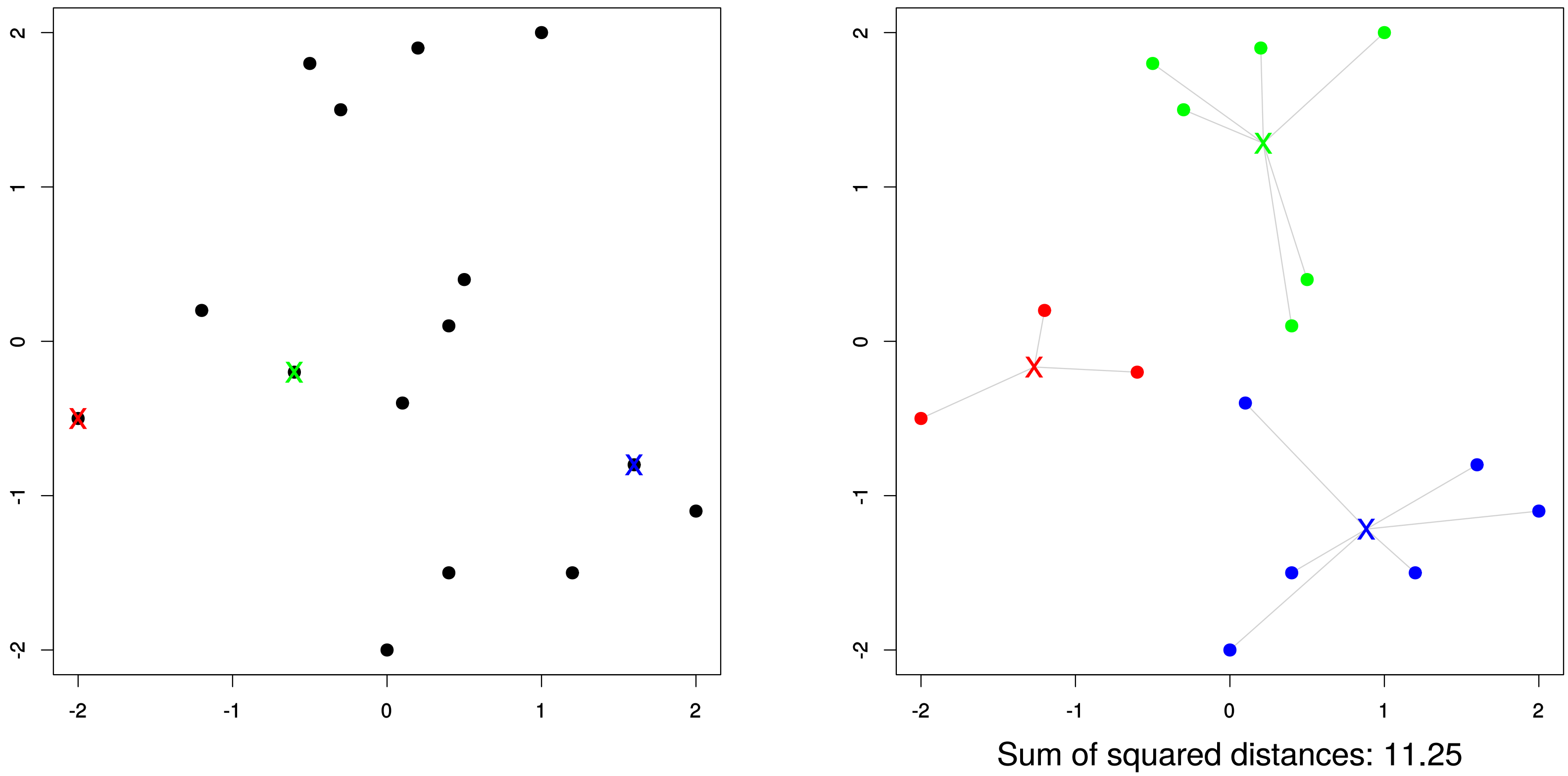
Sum of squared distances: 10.61

Different starting points can yield different results

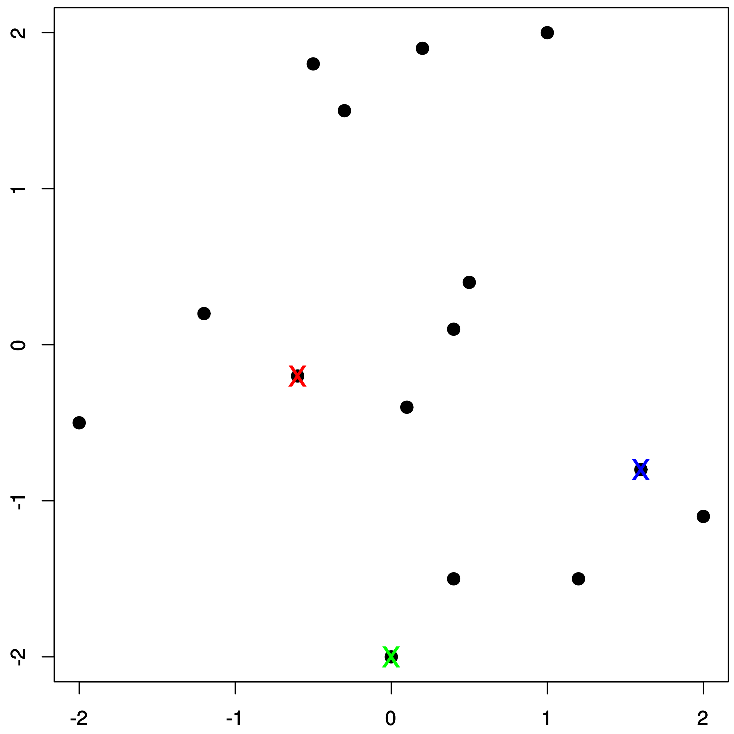


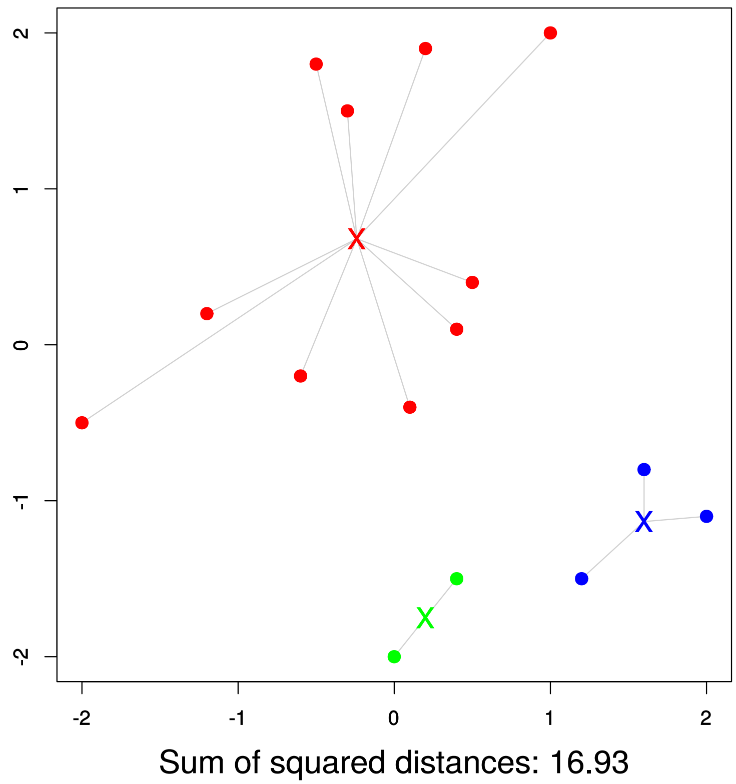


Different starting points can yield different results

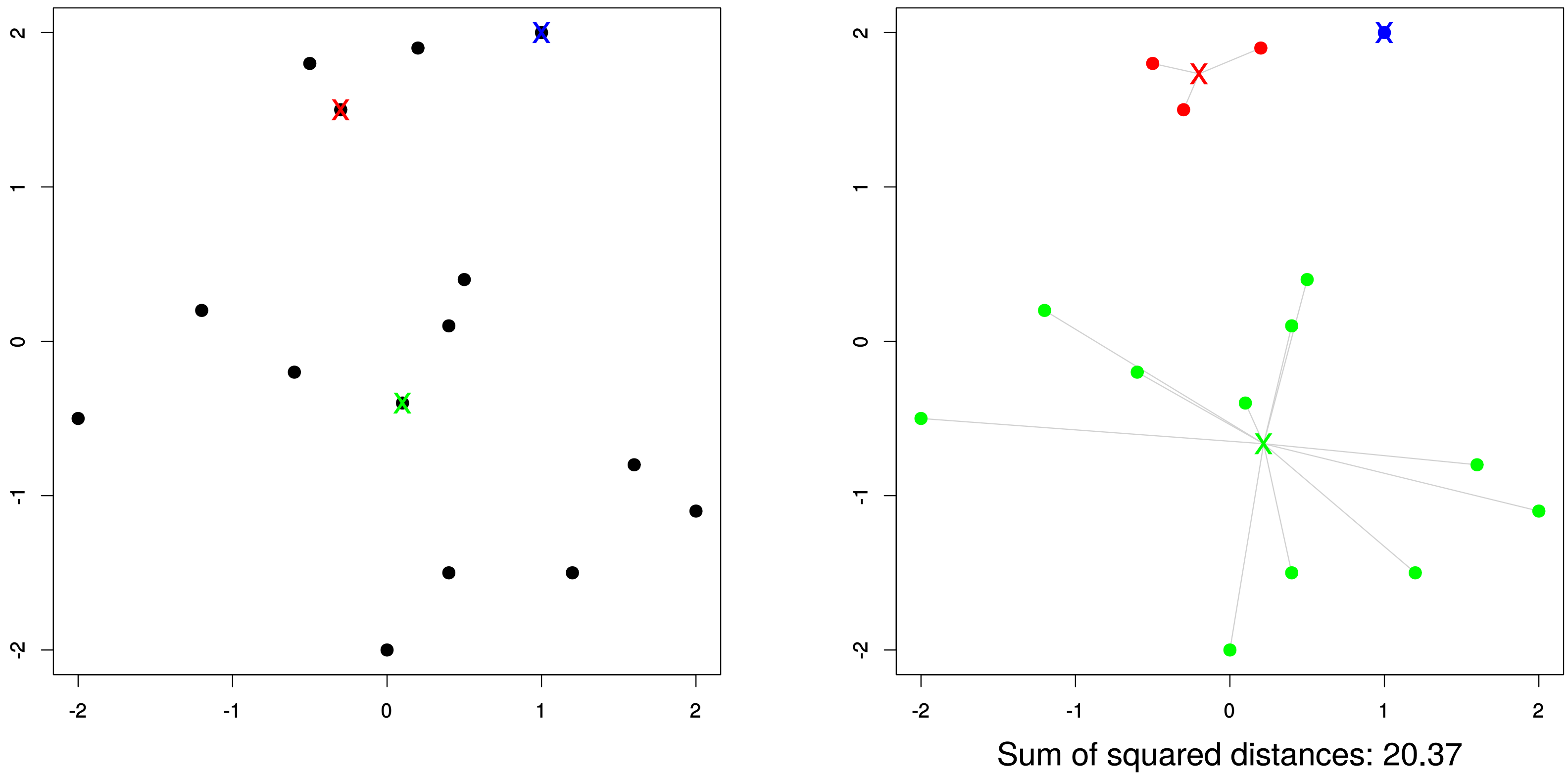


Different starting points can yield different results





Different starting points can yield different results



k-medoids algorithm

Arbitrary distance function $d(x, z)$:

$d(x, z) = 0$ if $x = z$

$d(x, z) = d(z, x)$

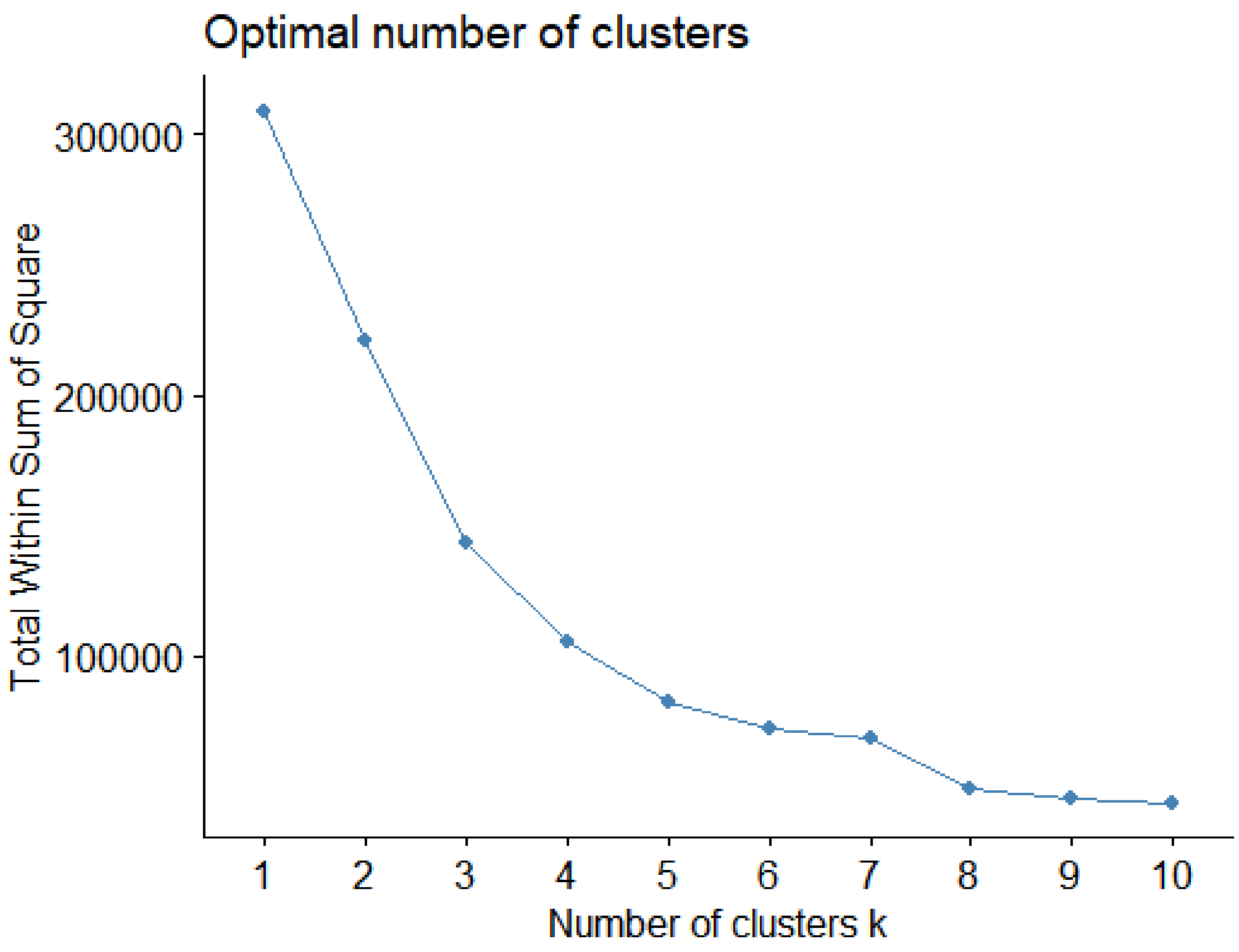
Initialization:

choose k centers $m_1, m_2, ..., m_k$ randomly out of the input data points

Repeat until convergence:

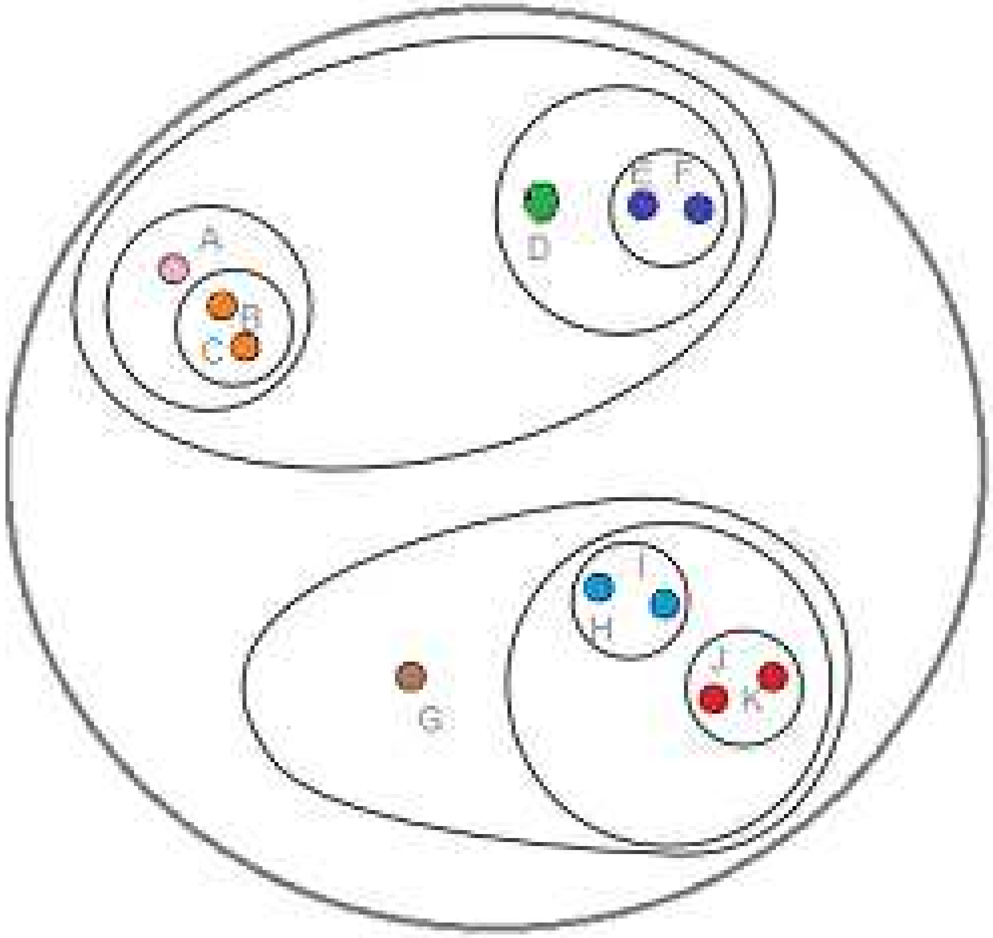
- assign each data point to the nearest center:
  $c_i = \arg\min_j d(x_i, m_j)$
- computer new centers: $m_j := \arg\min_{m_k : c_k = j} \sum_{i : c_i = j} d(x_i, m_k)$

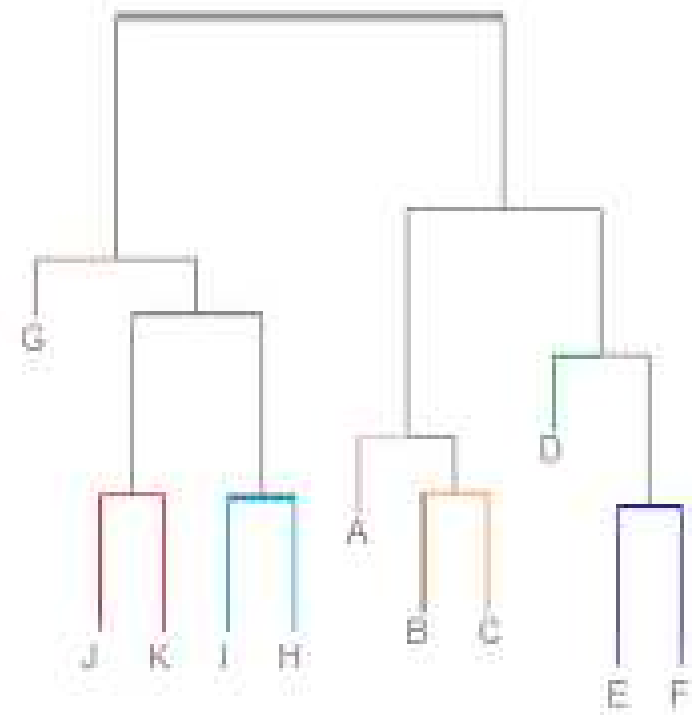
How many clusters?



kaggle / Rohan Shetty

Hierarické zhlukovanie





statisticshowto.com

Aglomeratívne zhlukovanie: zdola nahor

- Na začiatku každé dáto samostatný zhluk
- V každej iterácii zlúčime dva "najpodobnejšie" zhluky

  Kritériá podobnosti:

  - **single linkage:** vzdialenosť dvoch najbližší bodov
  - **group average:** vzdialenosť centier
  - **complete linkage:** priemer vzdialeností každý s každým

Divizívne zhlukovanie: zhora nadol

- Na začiatku všetky dáta v jedinom zhluku
- V každom kroku vyberieme jeden zhluk a rozdelíme ho na dva

  Napríklad zo zhluku G vyčleníme zhluk H:

  - vyber najvzdialenejší bod od centra a založ nový zhluk H
  - postupne presúvaj ďalšie body z, pre ktoré

    $\mathrm{avg}_{z \in H} d(x,z) - \mathrm{avg}_{z \in G} d(x,z)$

    je najmenšie a záporné