Non - Hierarchical Cluster

Enrique J. De La Hoz D.

UTB - Data Science

K - means algorithm

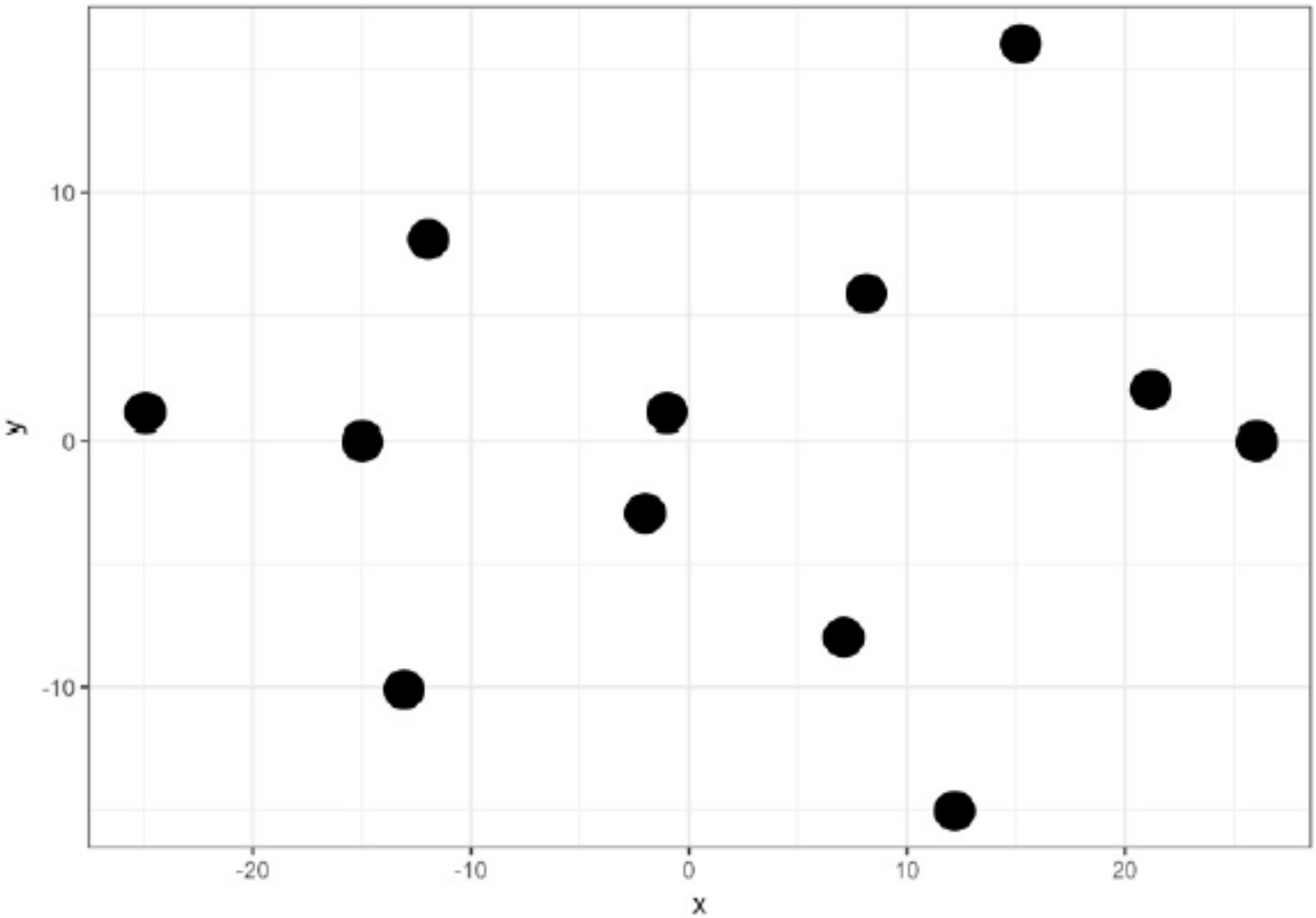


Figure 1:

K - means algorithm

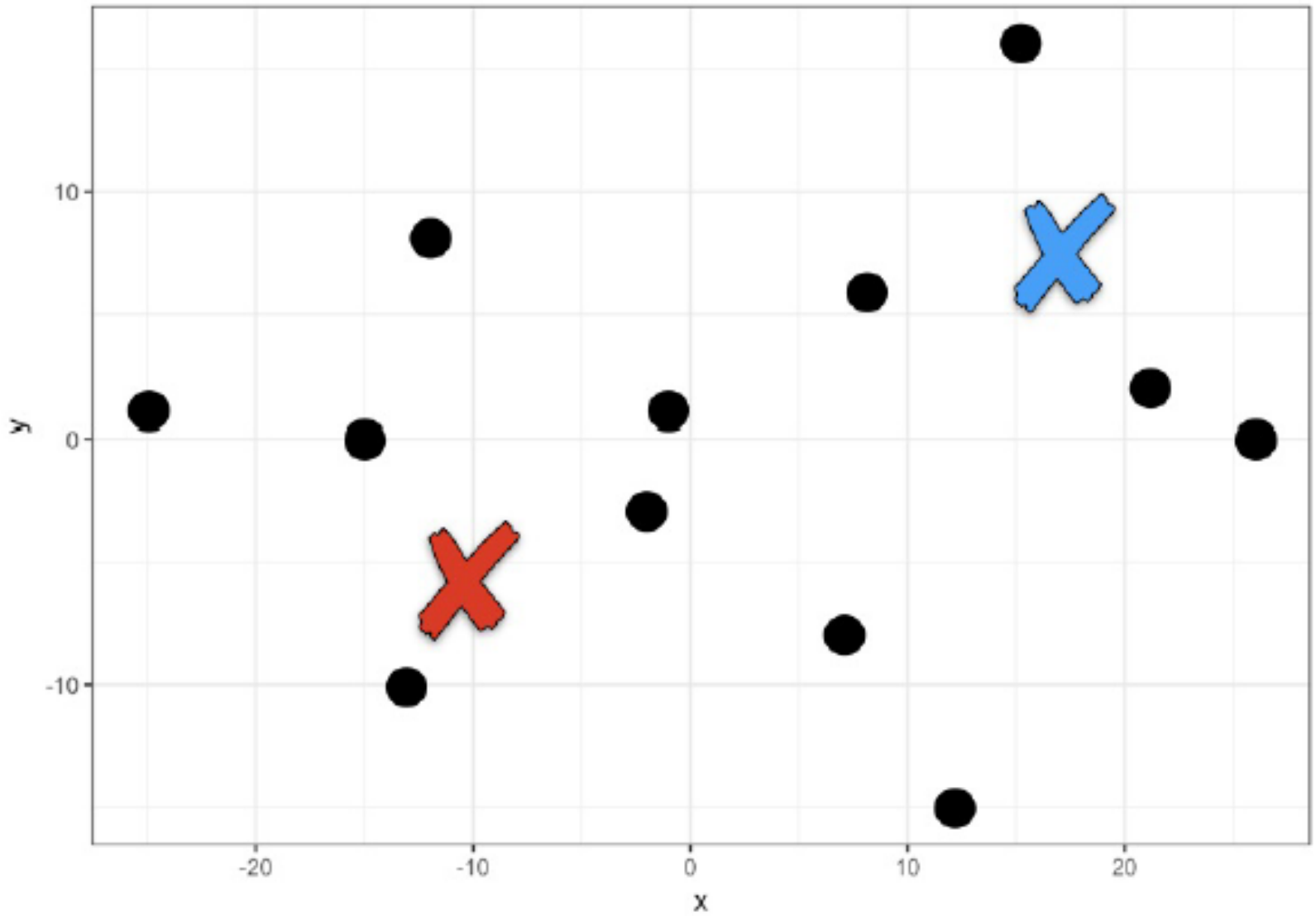


Figure 2:

K - means algorithm

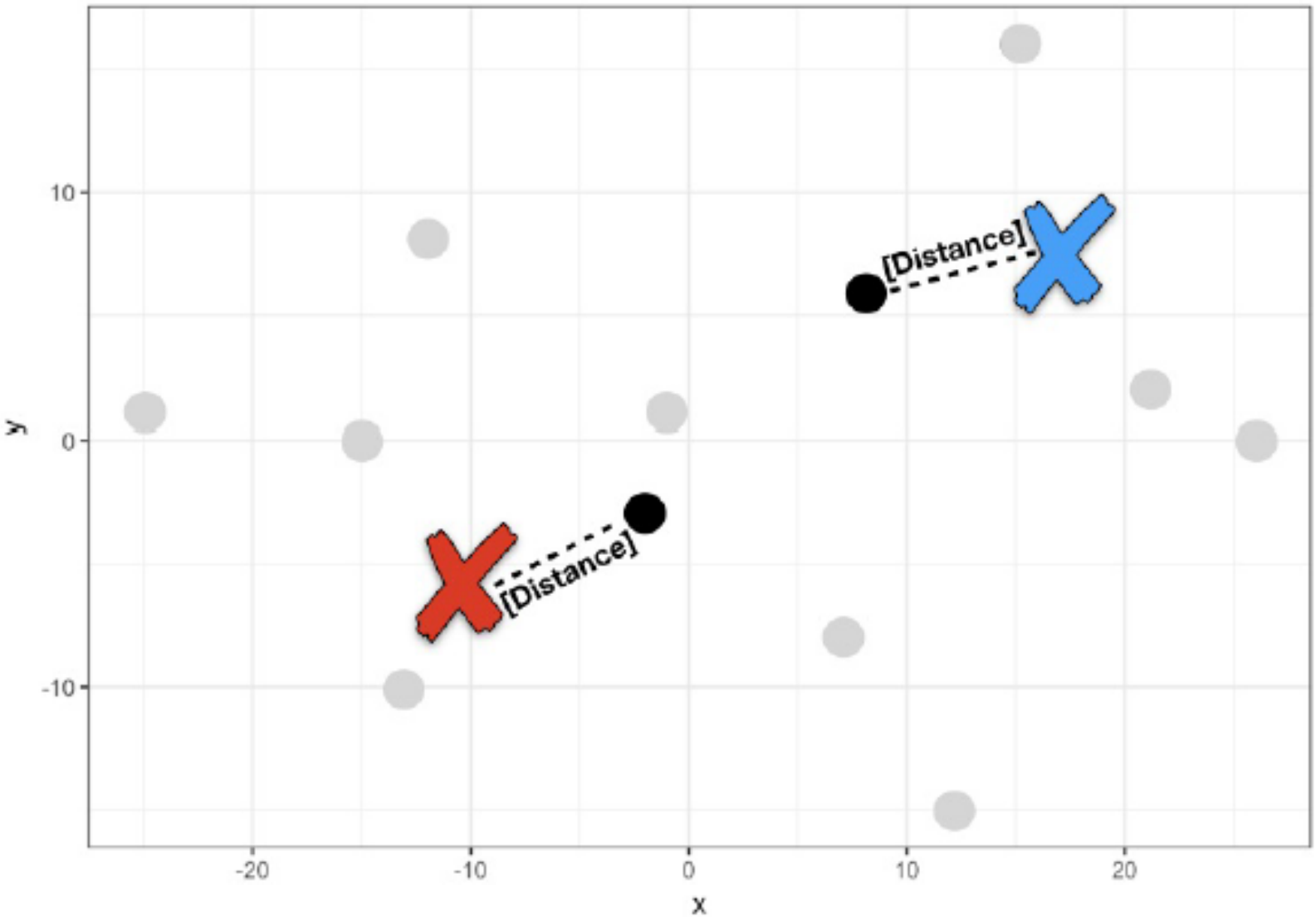


Figure 3:

K - means algorithm

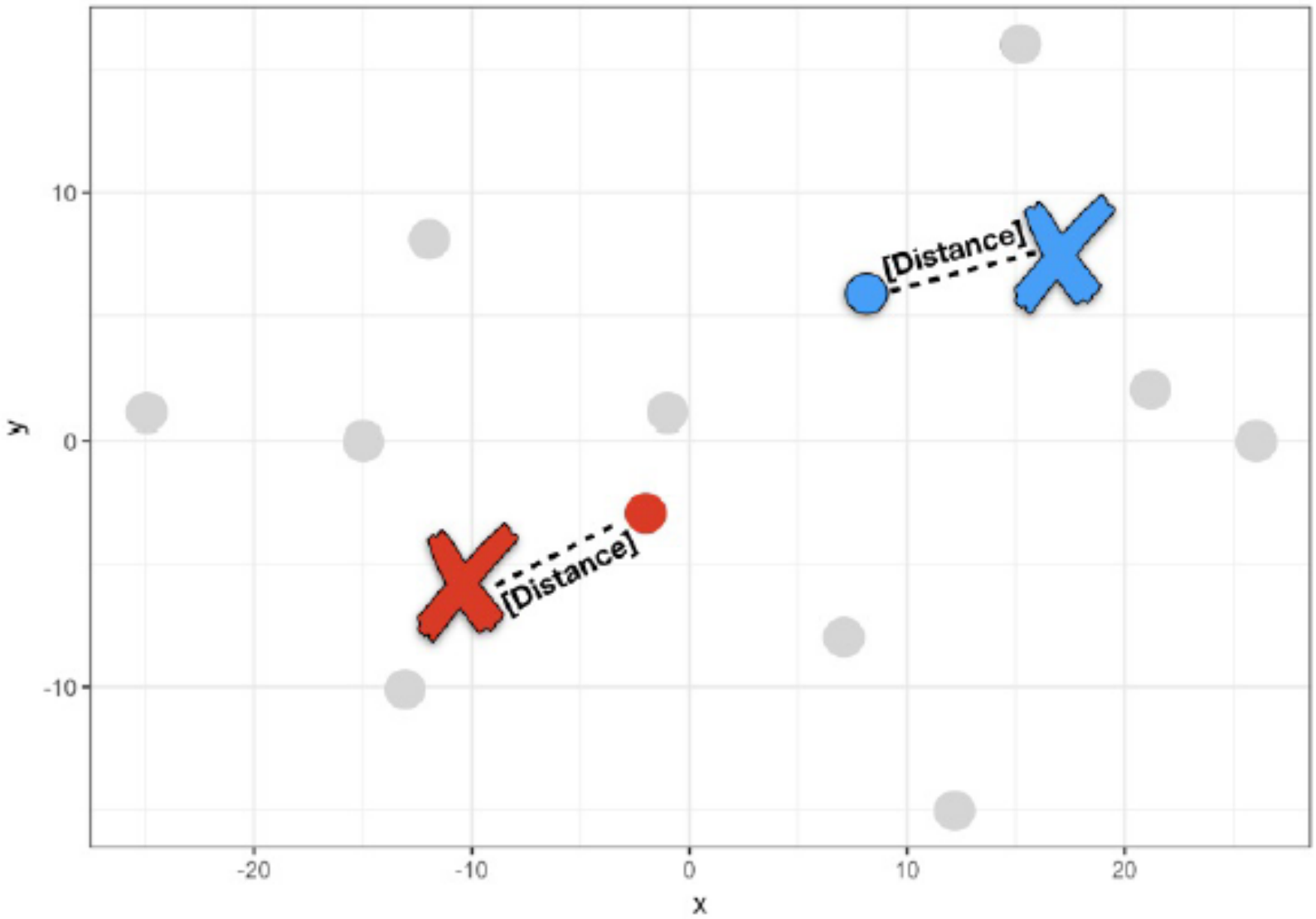


Figure 4:

K - means algorithm

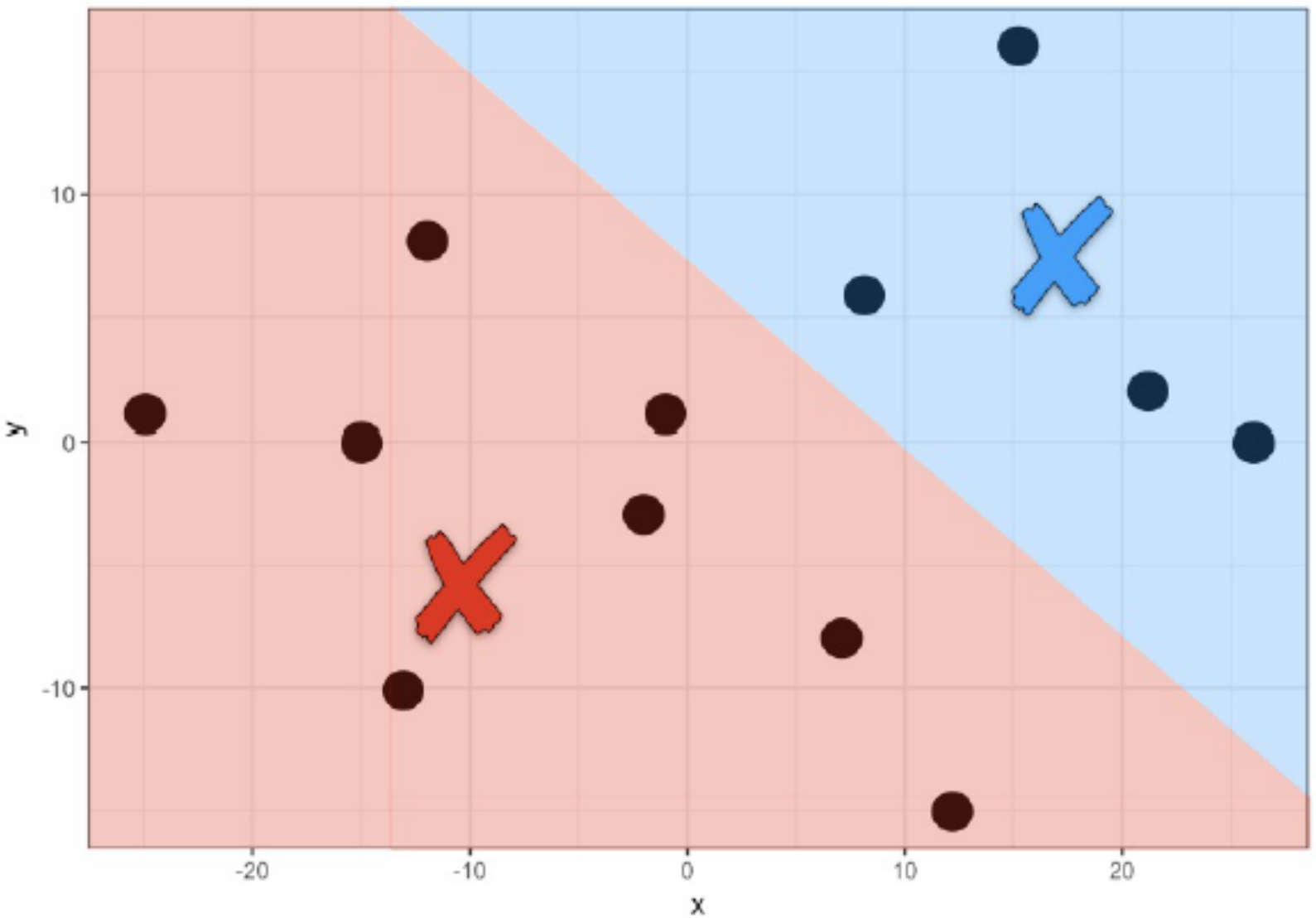


Figure 5:

K - means algorithm

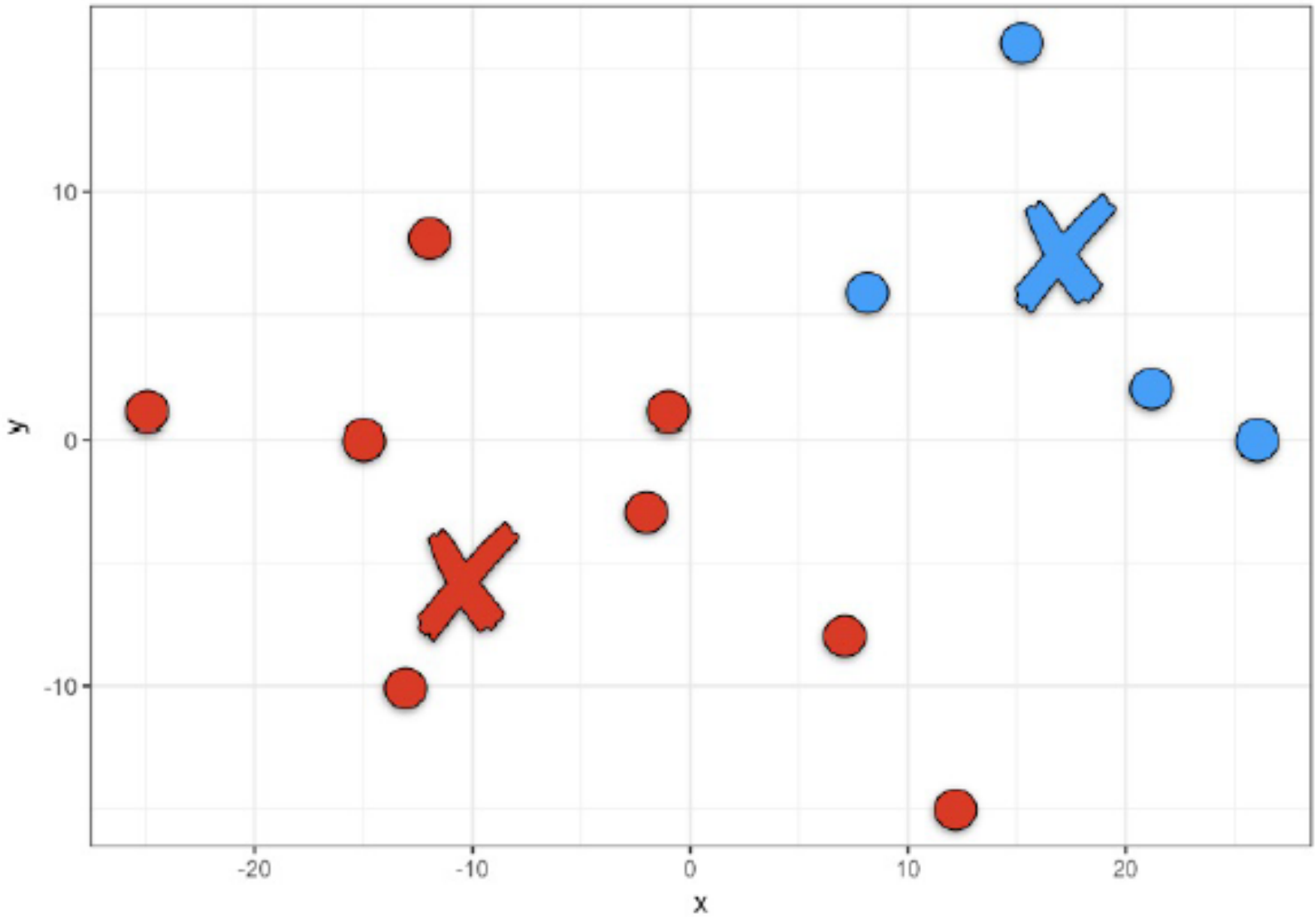


Figure 6:

K - means algorithm

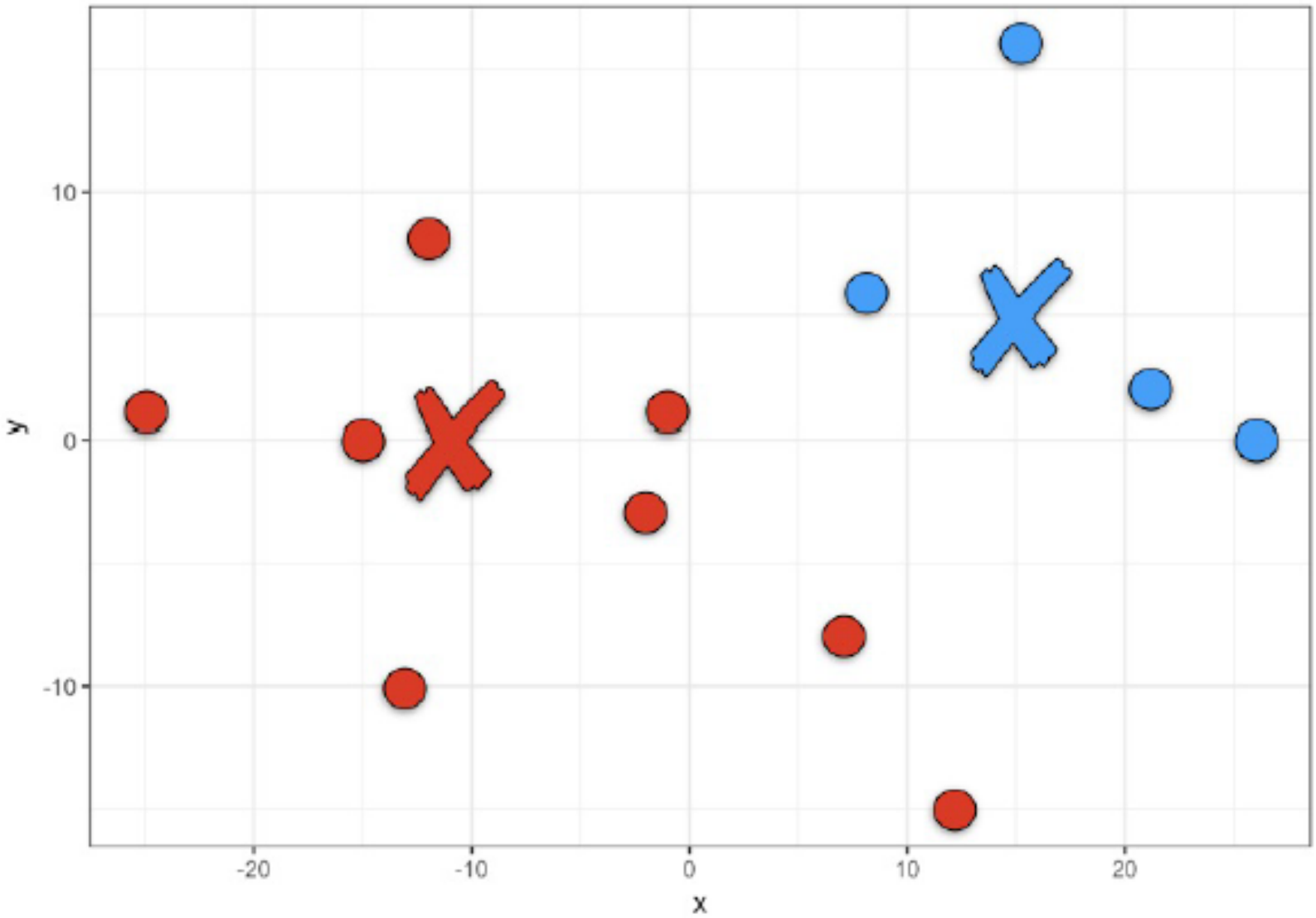


Figure 7:

K - means algorithm

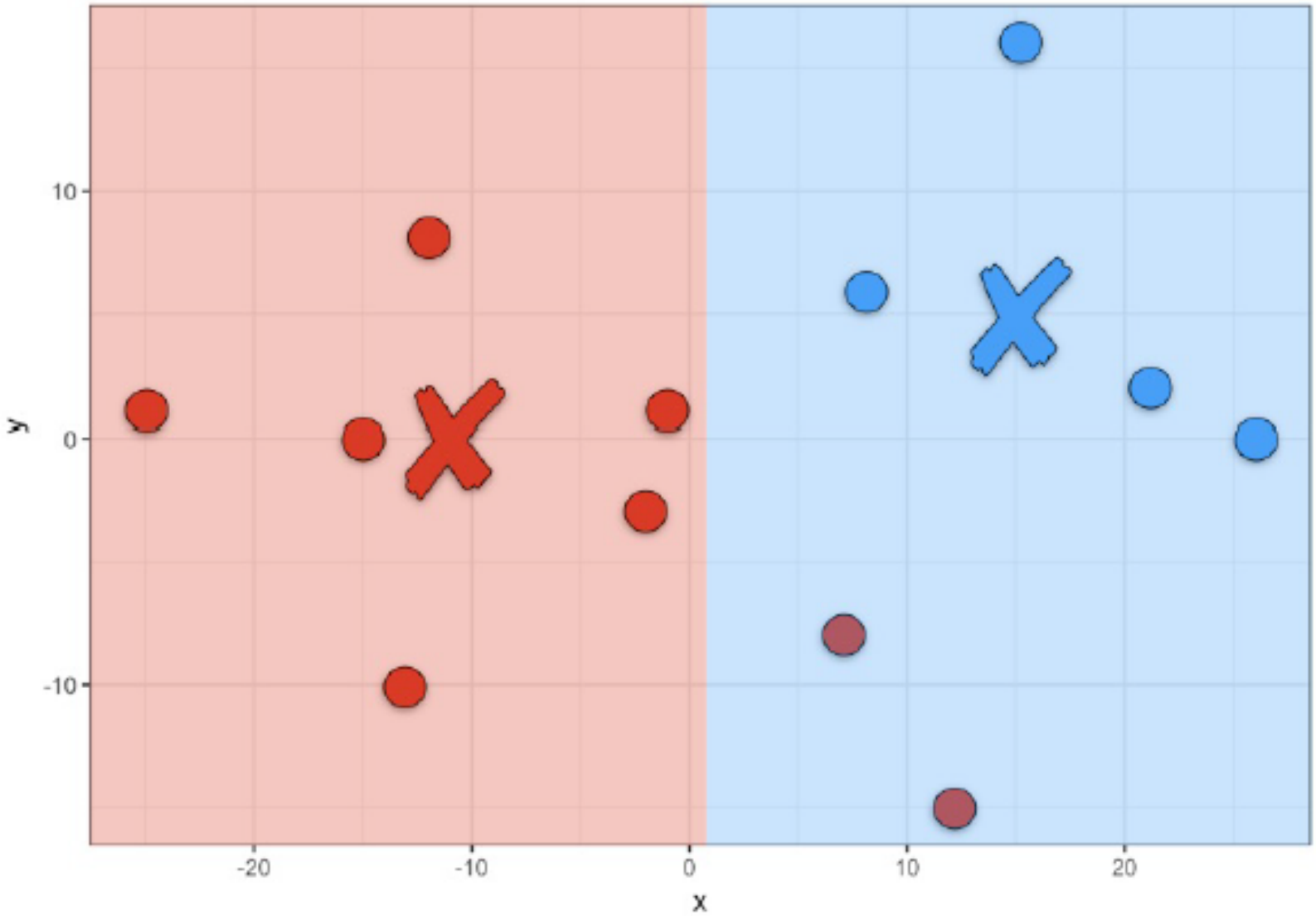


Figure 8:

K - means algorithm

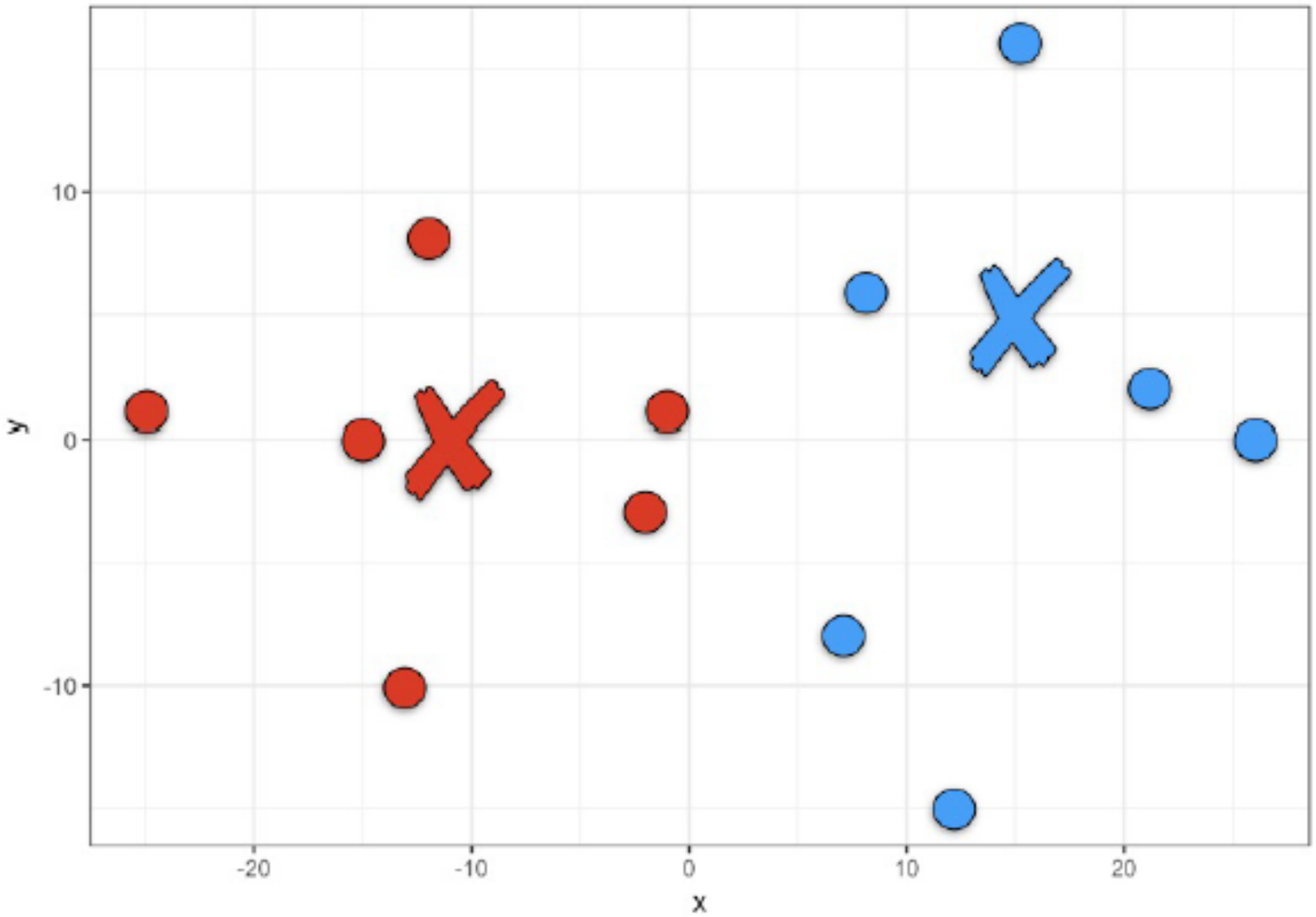


Figure 9:

K - means algorithm

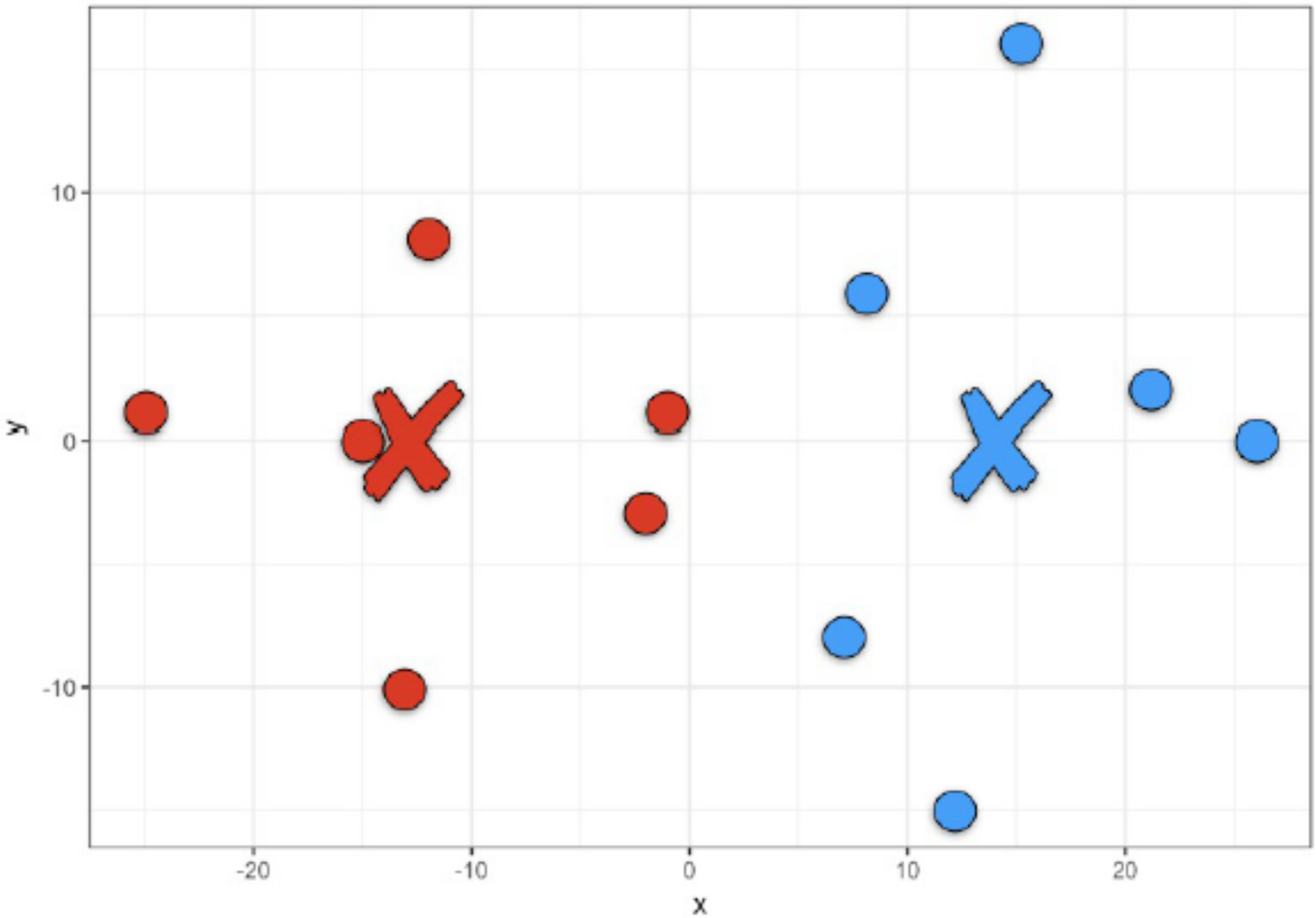


Figure 10:

kmeans in r with function kmeans()

```
lineup
```

```
##      X  Y
## 1   -1  1
## 2    2 -3
## 3    8  6
## 4    7 -8
## 5  -12  8
## 6  -15  0
```

```
model <- kmeans(lineup, centers = 2)
```

Assigning Clusters

model$cluster

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```
##
## Attaching package: 'dplyr'
```

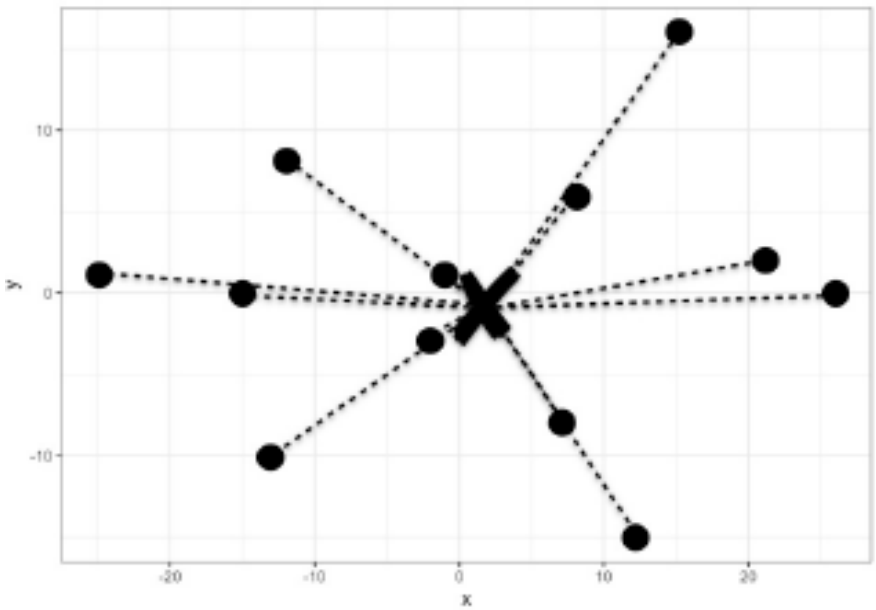
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
print(model$cluster)
```

Defining values of k by Eye

- Total Within-Cluster Sum of Squares: k = 1



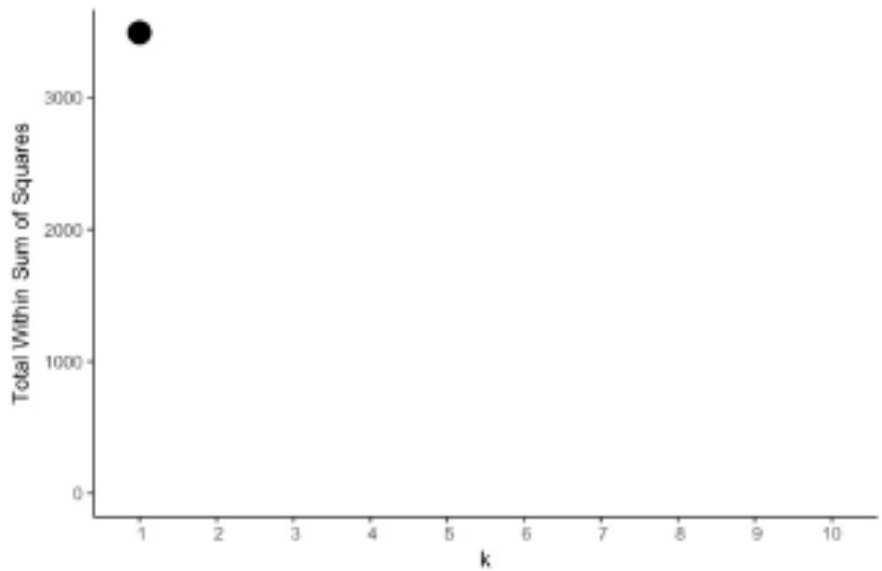
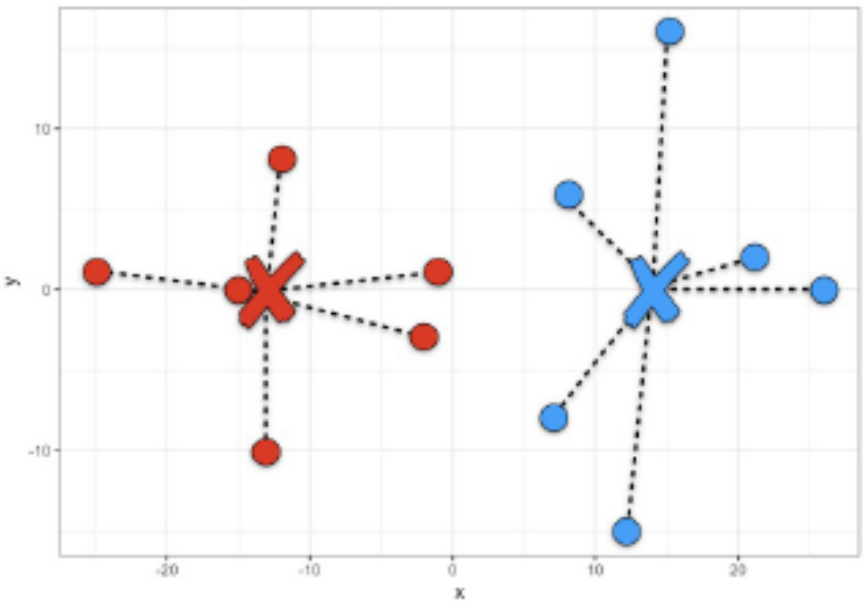


Figure 11:

Defining values of k by Eye

- Total Within-Cluster Sum of Squares: k = 2



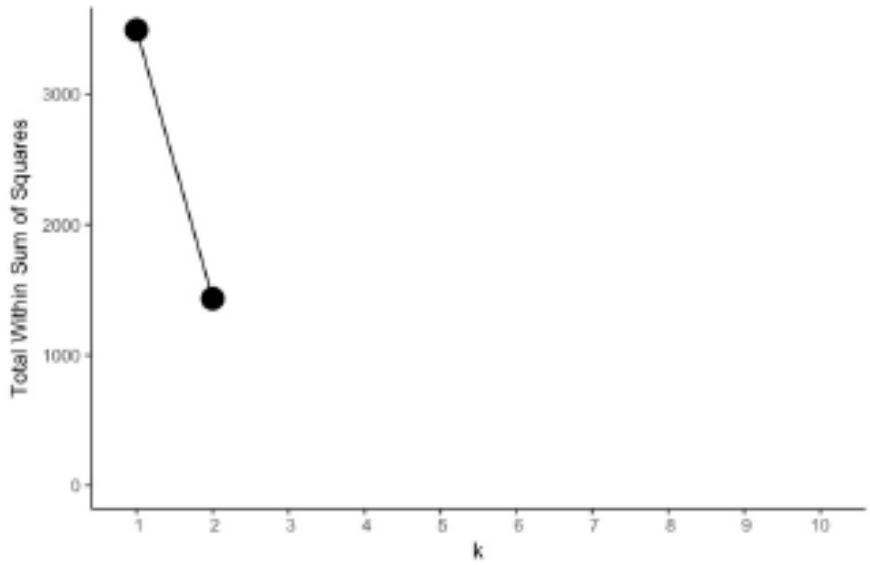


Figure 12:

Defining values of k by Eye

- Total Within-Cluster Sum of Squares: k = 3

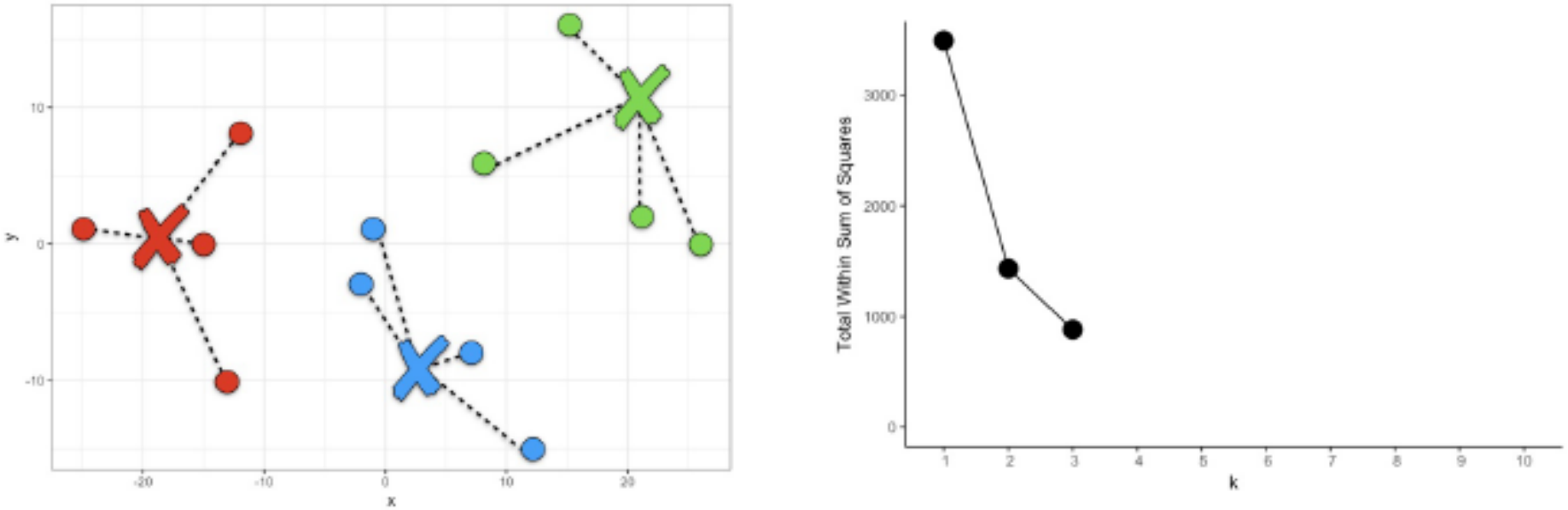
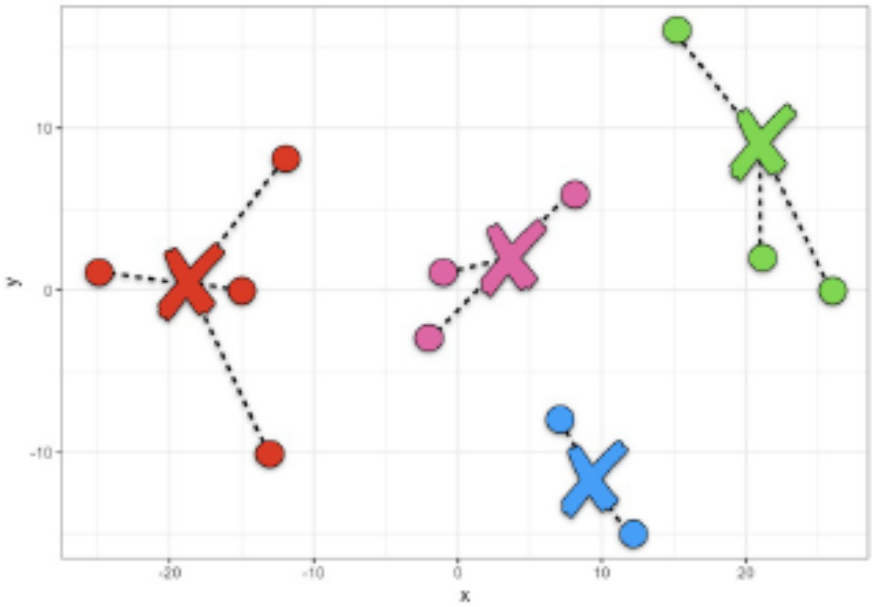


Figure 13:

Defining values of k by Eye

- Total Within-Cluster Sum of Squares: k = 4



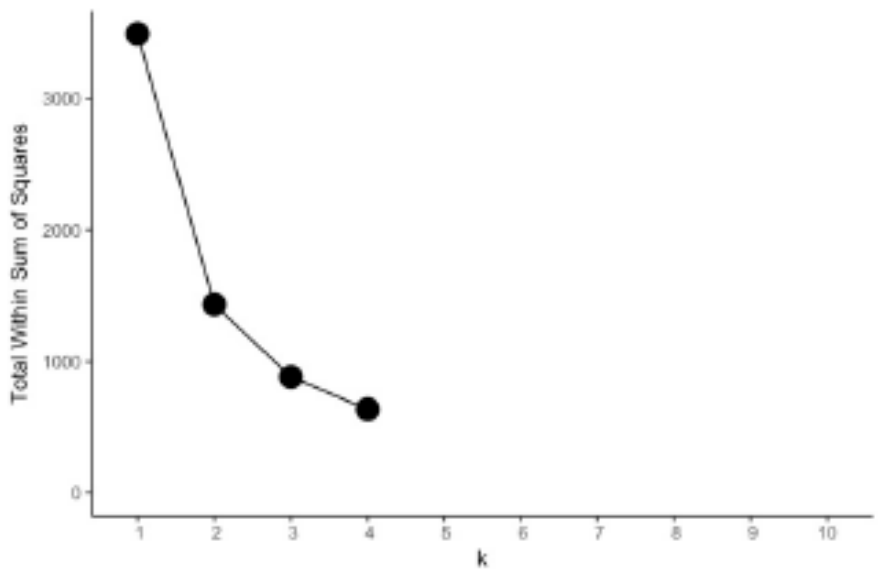


Figure 14:

Elbow Plot

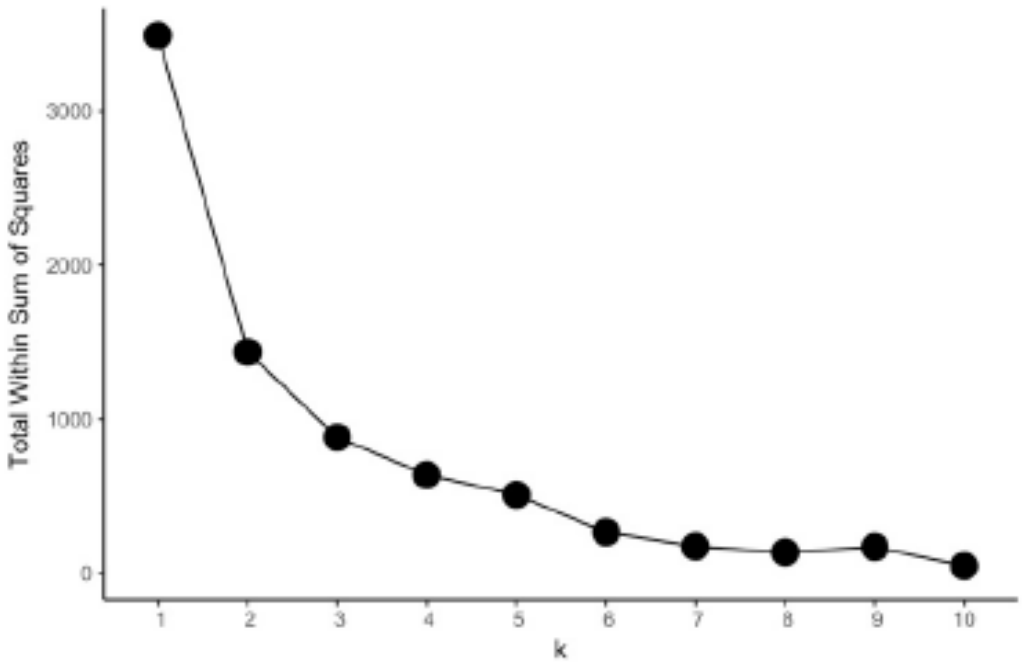


Figure 15:

Elbow Plot

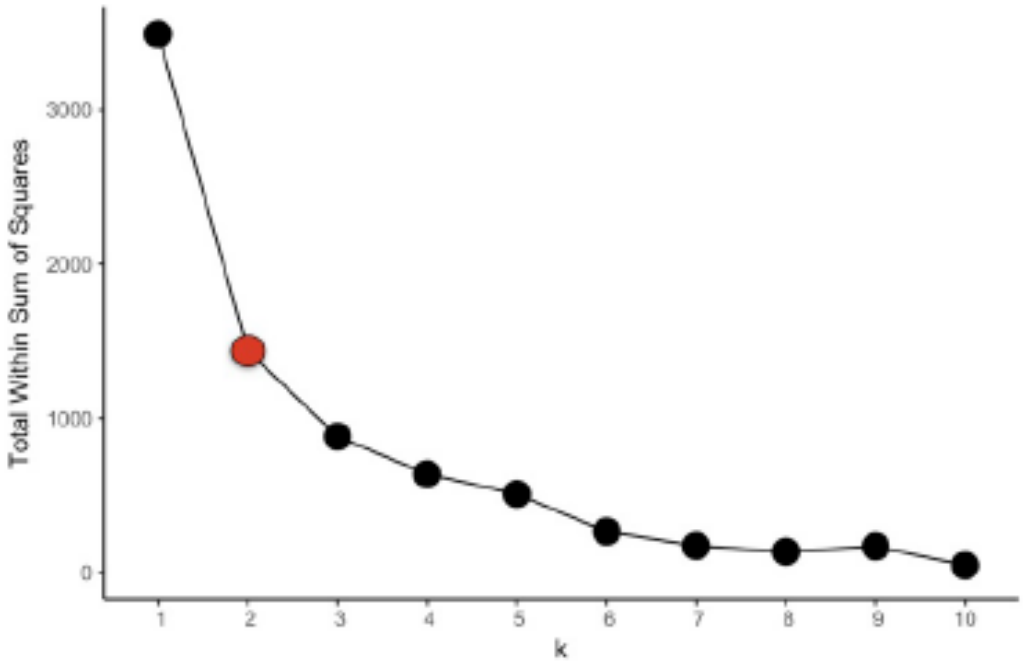


Figure 16:

Generating the Elbow Plot

```
model <- kmeans(x = lineup, centers = 2)
model$tot.withinss
```

```
## [1] 196.5
```

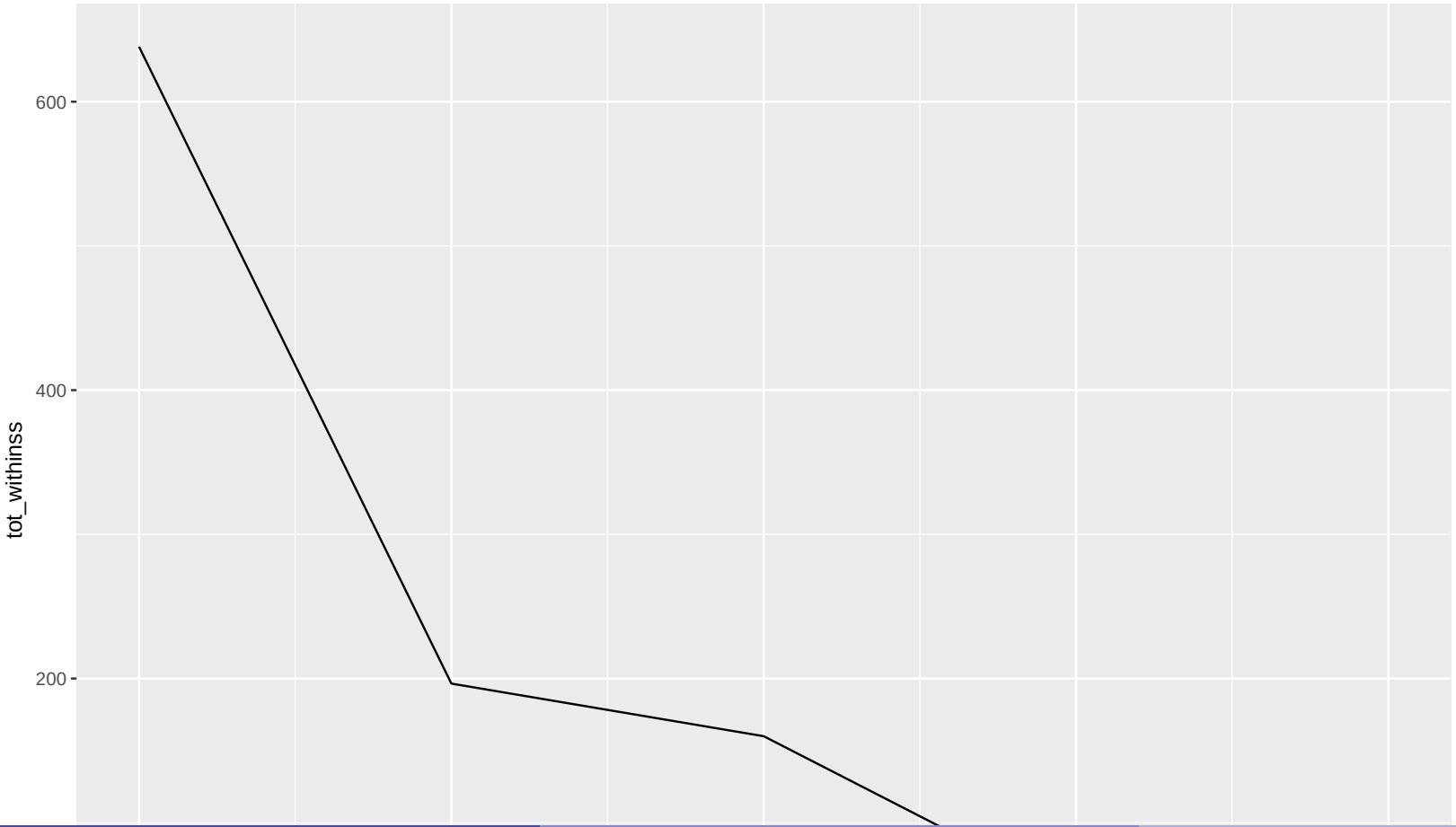
Generating the Elbow Plot

```
library(purrr)
tot_withinss <- map_dbl(1:5, function(k){
model <- kmeans(x = lineup, centers = k)
model$tot.withinss
})
elbow_df <- data.frame(
k = 1:5,
tot_withinss = tot_withinss
)
elbow_df
```

```
##   k tot_withinss
## 1 1     638.1667
## 2 2     196.5000
## 3 3     160.0000
## 4 4      49.0000
## 5 5      36.5000
```

Plotting the Elbow

```
library(ggplot2)
ggplot(elbow_df, aes(x = k, y = tot_withinss)) +
geom_line() +
scale_x_continuous(breaks = 1:10)
```



Silhouette Analysis

- Soccer Lineup with K = 3

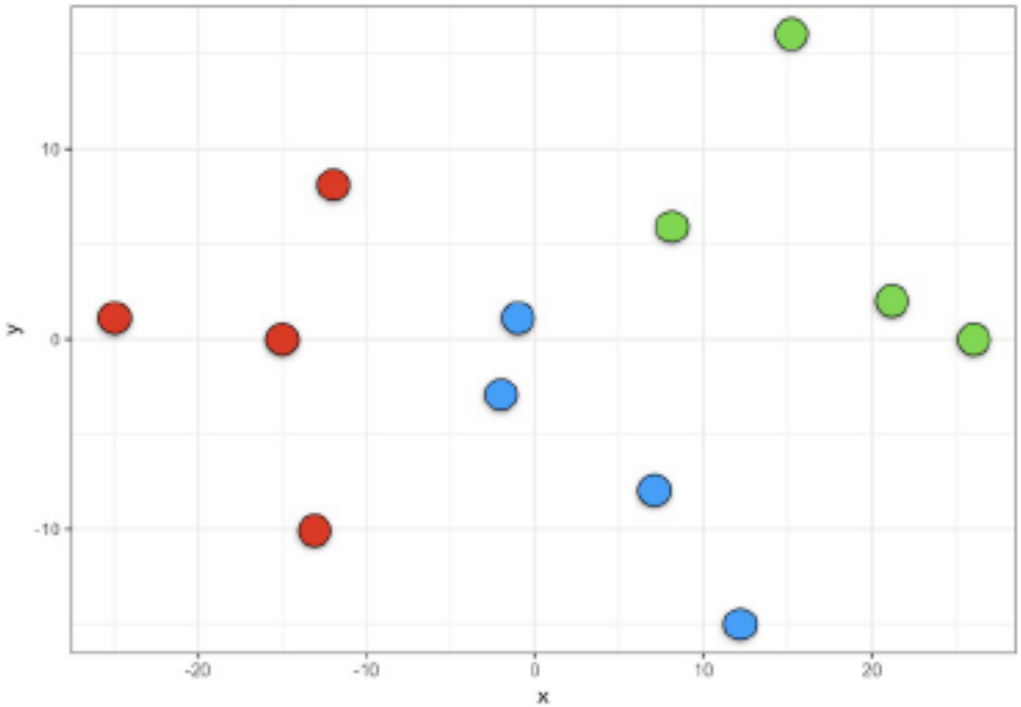


Figure 17:

Silhouette Width

Within Cluster Distance: C(i) **Closest Neighbor Distance: N(i)**

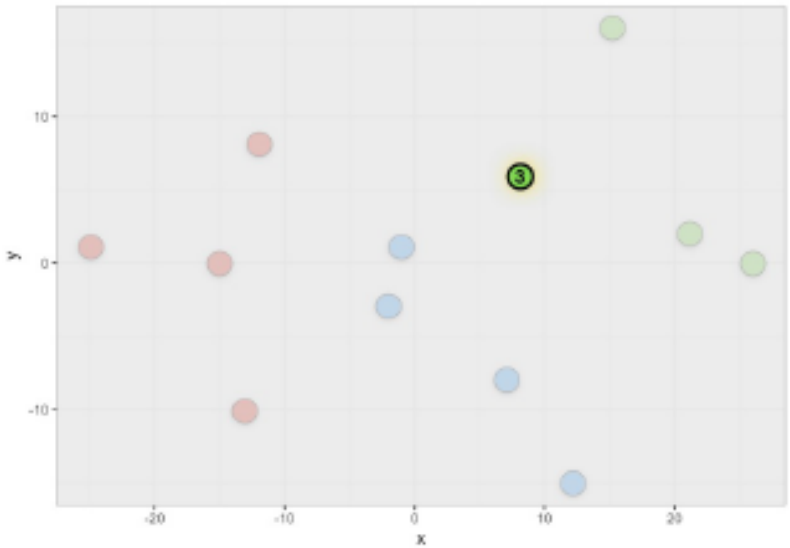


Figure 18:

Silhouette Width

Within Cluster Distance: C(i) **Closest Neighbor Distance: N(i)**

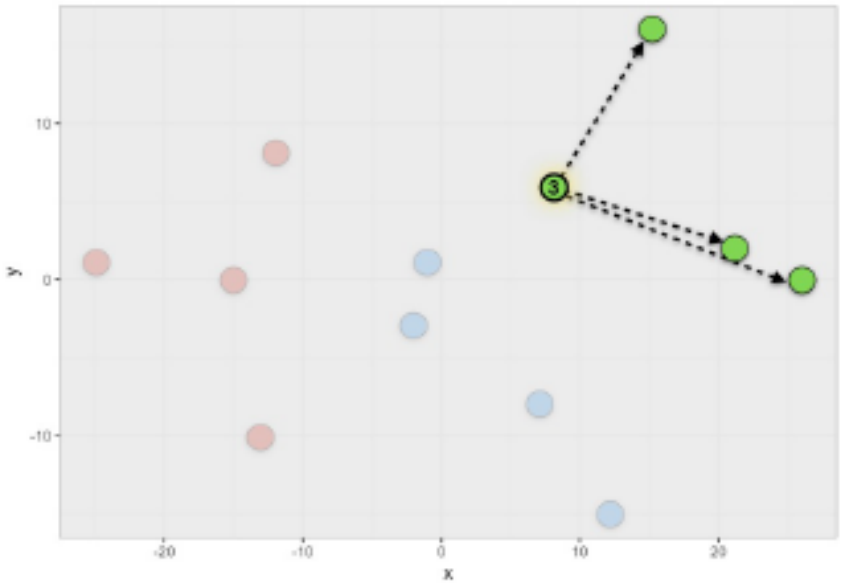


Figure 19:

Silhouette Width

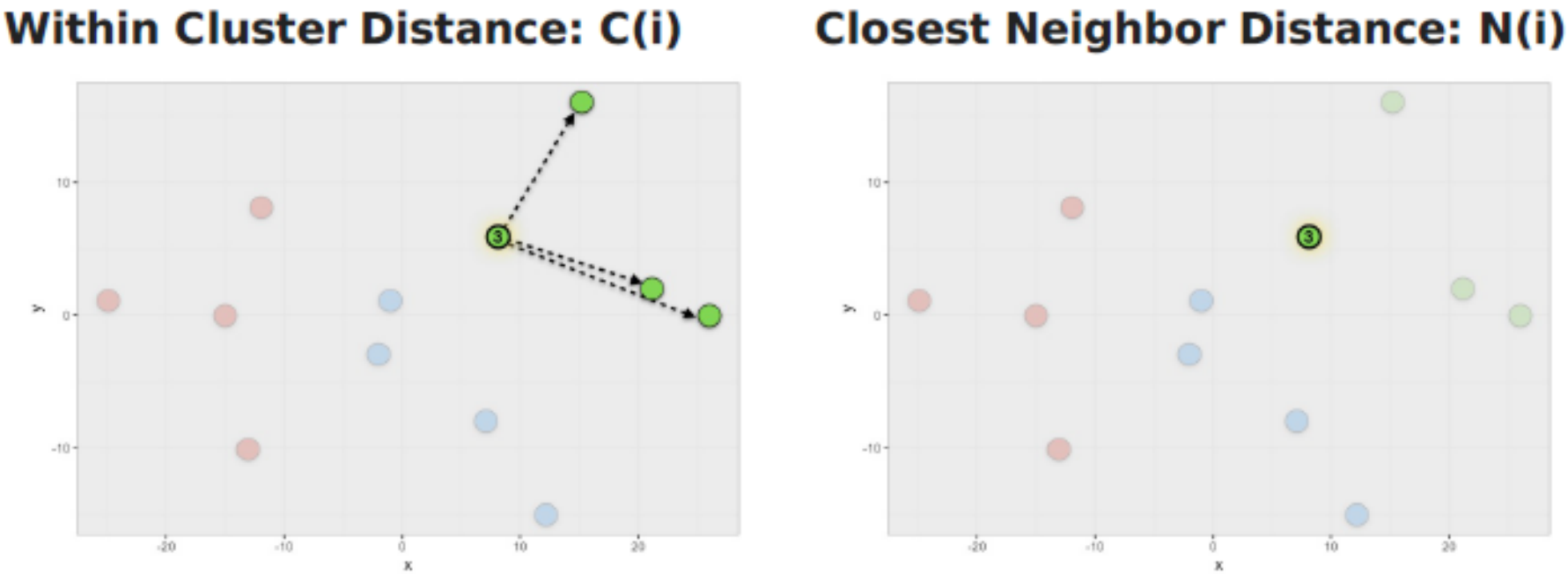


Figure 20:

Silhouette Width

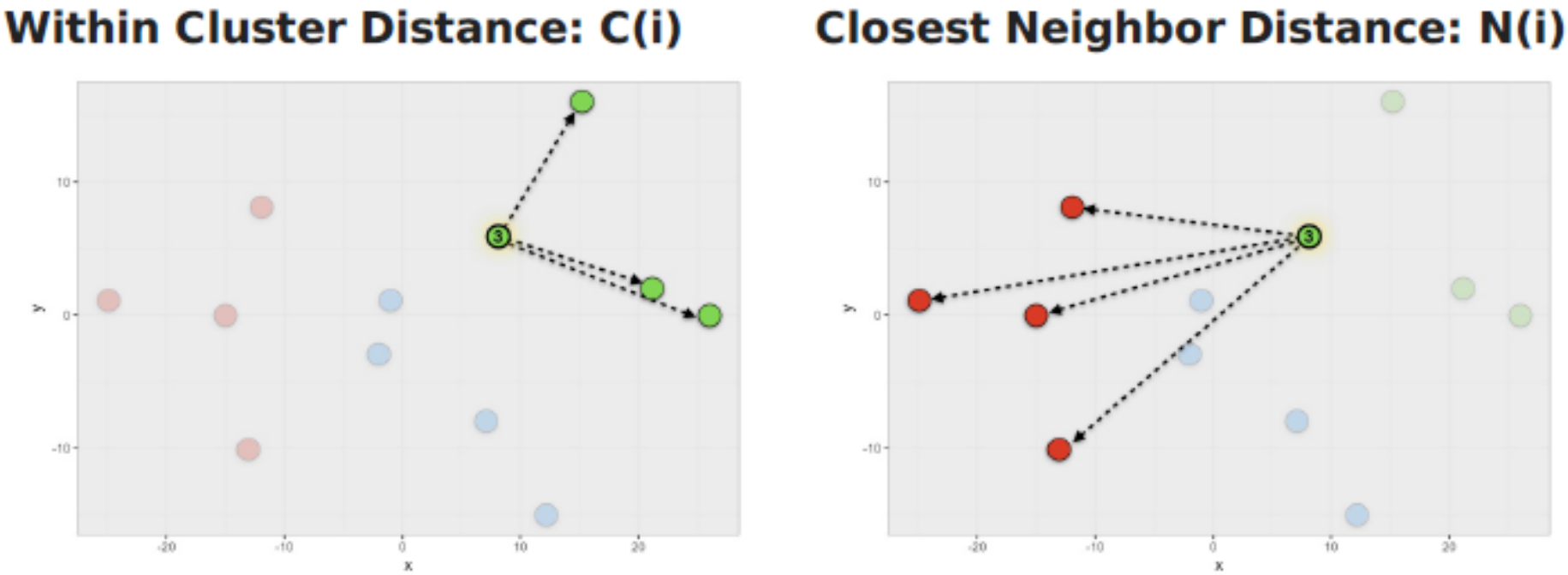


Figure 21:

Silhouette Width

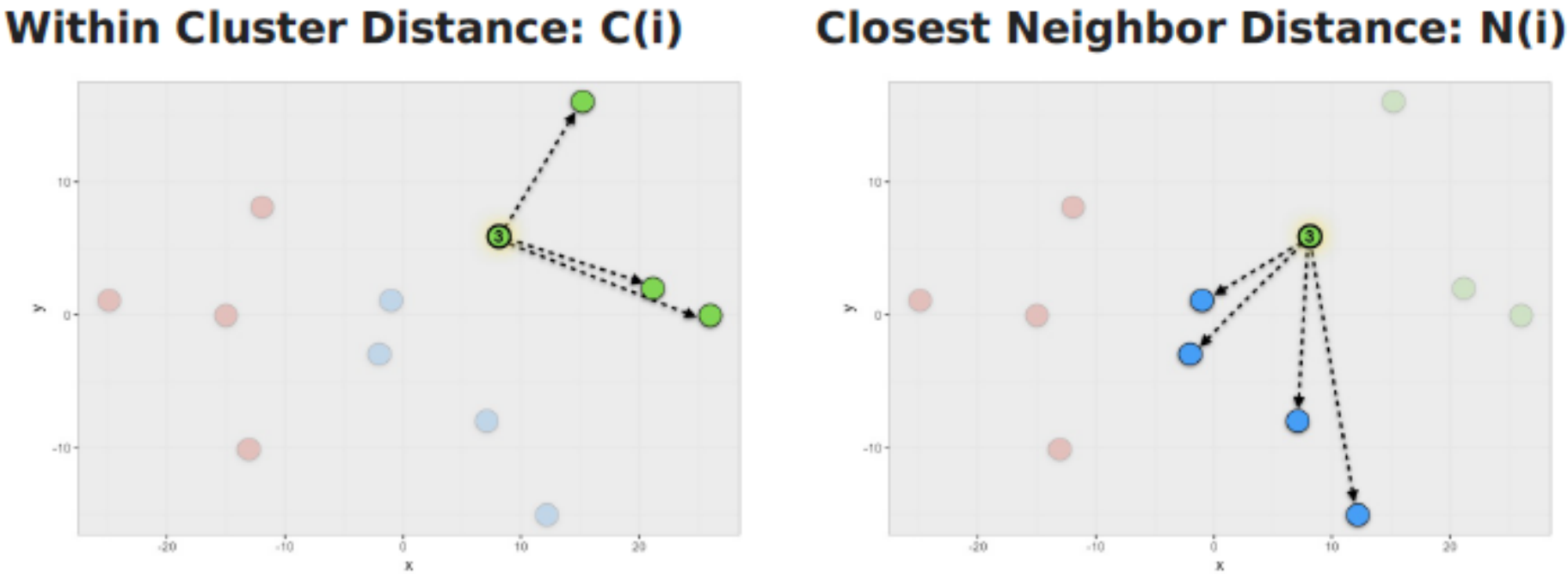
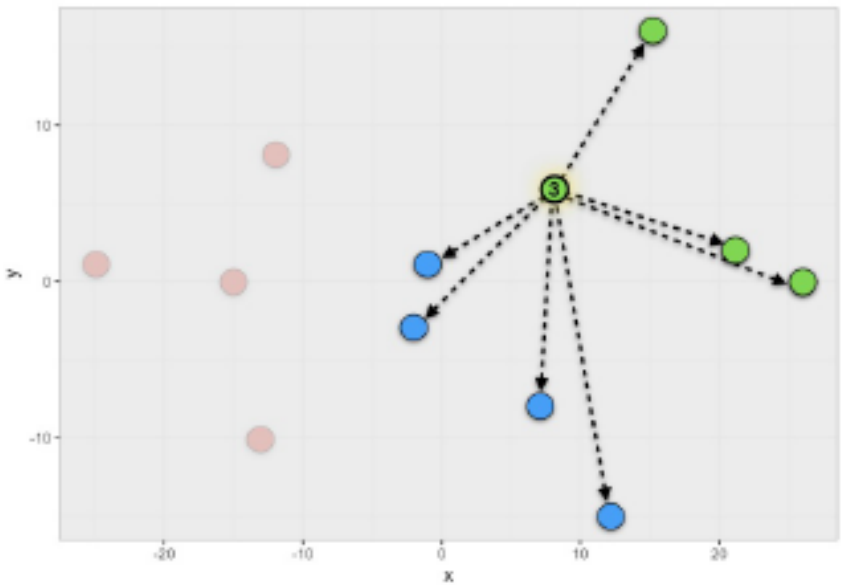


Figure 22:

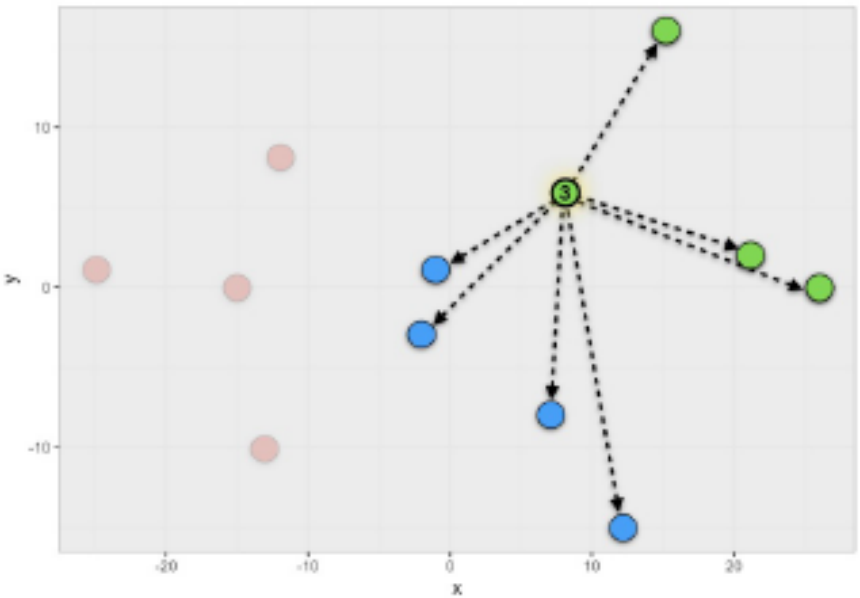
Silhouette Width S(i)



$$s(i) = \begin{cases} 1 - C(i)/N(i), & \text{if } C(i) < N(i) \\ 0, & \text{if } C(i) = N(i) \\ N(i)/C(i) - 1, & \text{if } C(i) > N(i) \end{cases}$$

Figure 23:

Silhouette Width S(i)



- **1:** Well matched to cluster
- **0:** On border between two clusters
- **-1:** Better fit in neighboring cluster

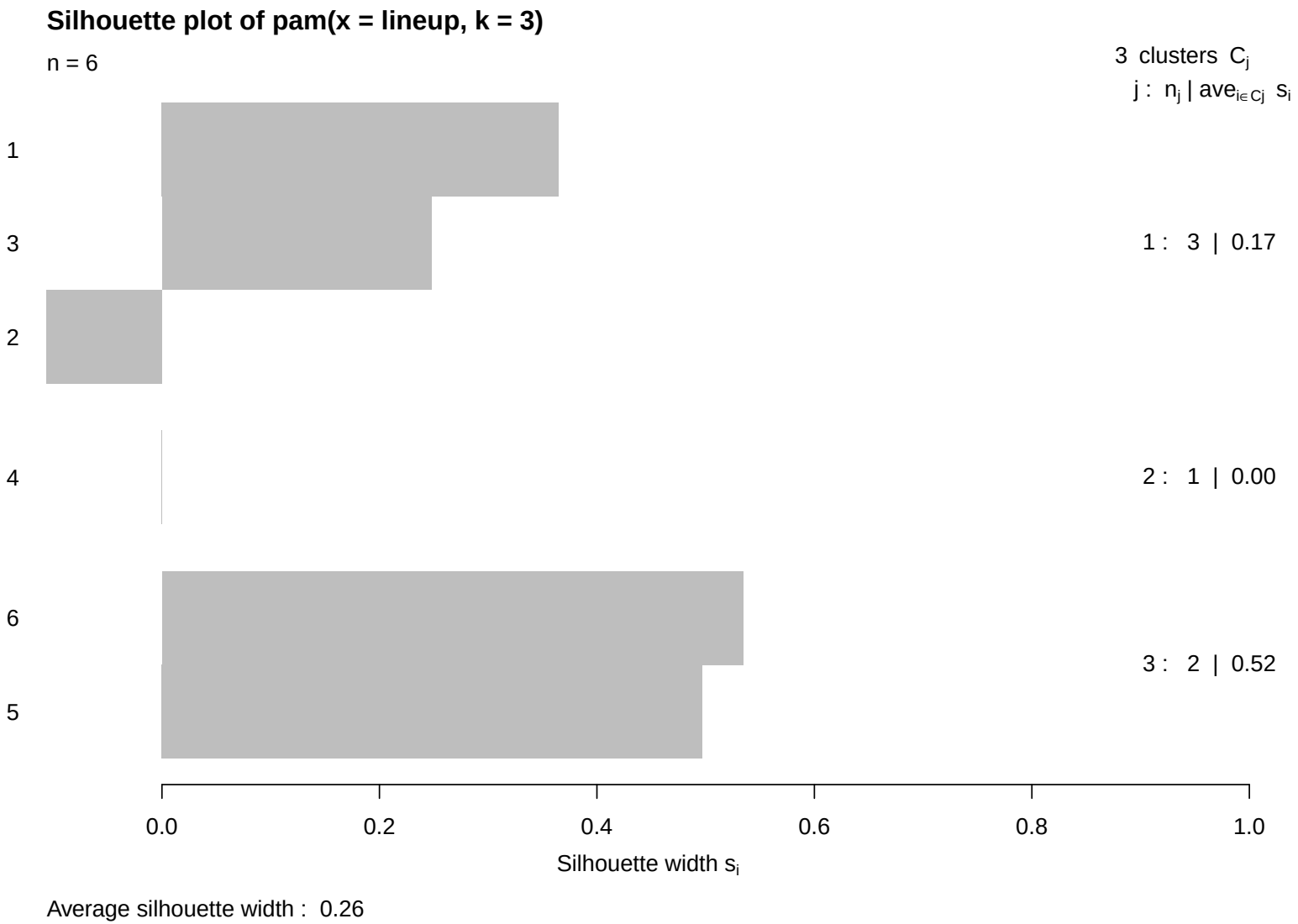
Figure 24:

Calculating S(i) in R

```
library(cluster)
pam_k3 <- pam(lineup, k = 3)
pam_k3$silinfo$widths
```

```
##   cluster neighbor  sil_width
## 1       1        2  0.3648835
## 3       1        2  0.2479060
## 2       1        2 -0.1058706
## 4       2        1  0.0000000
## 6       3        1  0.5345395
## 5       3        1  0.4968457
```

Silhouette Plot



Average Silhouette Width

```
## [1] 0.256384
```

- 1: Well matched to each cluster
- 0: On border between clusters
- -1: Poorly matched to each cluster

Highest Average Silhouette Width

```
##   k sil_width
## 1 2 0.5021865
## 2 3 0.2563840
## 3 4 0.2845408
## 4 5 0.1345417
```

Choosing K using Average Silhouette Width