Big Data in Social Sciences

Week 6. Clustering

Nicola Barban

Data reduction techniques

- A common goal of data analysis is to **reduce complexity** to explain social phenomena
- Principal component analysis is used to **reduce the number of variables,** often to describe complex concepts that are measured: e.g. political ideology
- Another common problem is to **classify individual observations into groups.**
- The most common technique is called **cluster analysis**

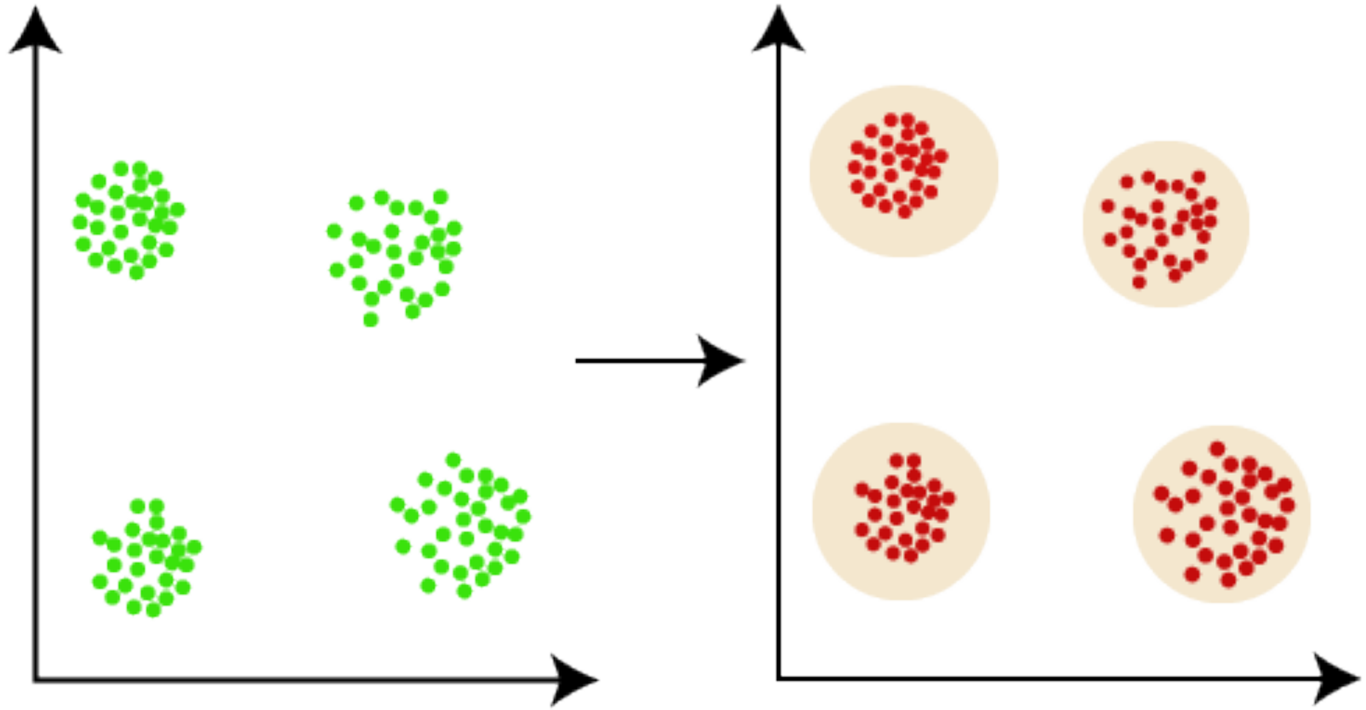
Unsupervised Machine Learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention.

Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, customer segmentation, and image recognition.

What is Cluster Analysis?

- Statistical methods to identify groups of observations based on their characteristics (“variables”)
- Reduce number of observations into “clusters”



Showing four clusters formed from the set of unlabeled data

Why?

- Identify patterns in the data
- Market segmentation
- Identify anomalous observations
- To automatically divide the sample into groups
- When the researcher does not have any pre-conceived hypotheses
- Often used in exploratory data analysis

Examples

- **Business and Marketing:** In customer segmentation, businesses use clustering to group customers based on similar behaviors or preferences. This enables targeted marketing, improves customer service, and aids in product development.
- **Healthcare and Medicine:** Clustering is used for patient classification based on symptoms, genetics, or response to treatments. This can guide diagnoses and therapeutic strategies. It's also used in genomic research, such as clustering genes with similar expression patterns.
- **Finance:** Financial institutions use cluster analysis for portfolio management, risk analysis, and customer segmentation. For instance, customers can be grouped based on their credit scores, income levels, and investment behaviors, allowing for customized financial advice.
- **Environment:** Clustering can help in identifying geographical areas with similar climate patterns or biodiversity, which is useful for environmental management and conservation planning.
- **Information Technology:** In cybersecurity, it's used for anomaly detection to identify unusual patterns or activities.
- **Social Science:** For instance, it can be used to segment populations based on socio-economic variables.
- **Education:** Clustering is used to group students based on their learning patterns and performance. This can inform differentiated instruction strategies and early intervention efforts.

Aim of clustering

Find clusters that:

1. **Minimize intra-cluster variation (known as total within-cluster variation).**

Observations in the same cluster should be similar

1. **Maximise cluster variation (known as total between-cluster variation).**

Different clusters should be very different

K-means clustering

- Algorithm-based technique
- Only need to specify number K of clusters (but we can try different solutions)
- K=1 is equivalent of having all observation belonging to 1 cluster
- K=N is equivalent of having each observation in a different cluster
- Algorithm minimises within-cluster variation

K-means algorithm

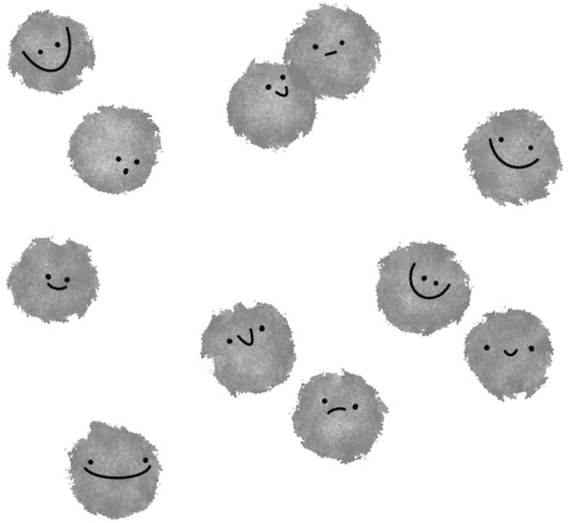
K-means algorithm can be summarized as follows:

1. Specify the number of clusters (K) to be created (by the analyst)
2. Select randomly k objects from the data set as the initial cluster centers or means
3. Assigns each observation to their closest centroid, based on the Euclidean distance between the object and the centroid
4. For each of the k clusters update the cluster centroid by calculating the new mean values of all the data points in the cluster. The centroid of a K-th cluster is a vector of length *p* containing the means of all variables for the observations in the kth cluster; *p* is the number of variables.
5. Iteratively minimize the total within sum of square. That is, iterate steps 3 and 4 until the cluster assignments stop changing or the maximum number of iterations is reached.

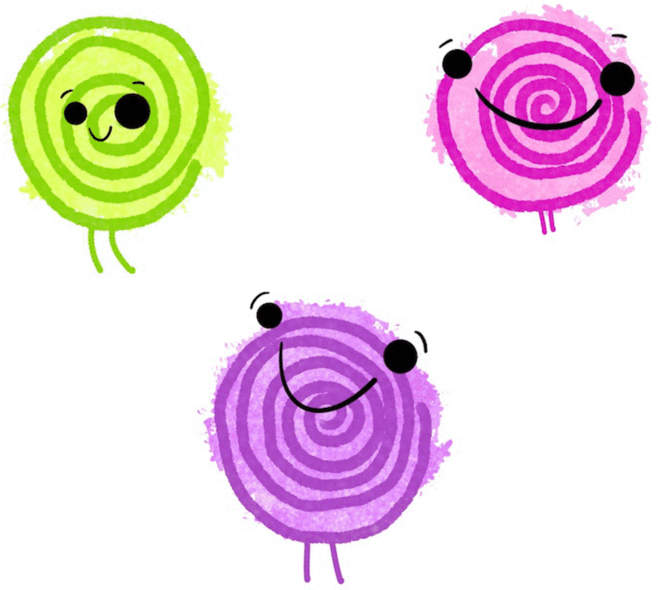
k-means clustering

: assign each observation to one of k clusters based on the nearest cluster centroid.

OBSERVATIONS



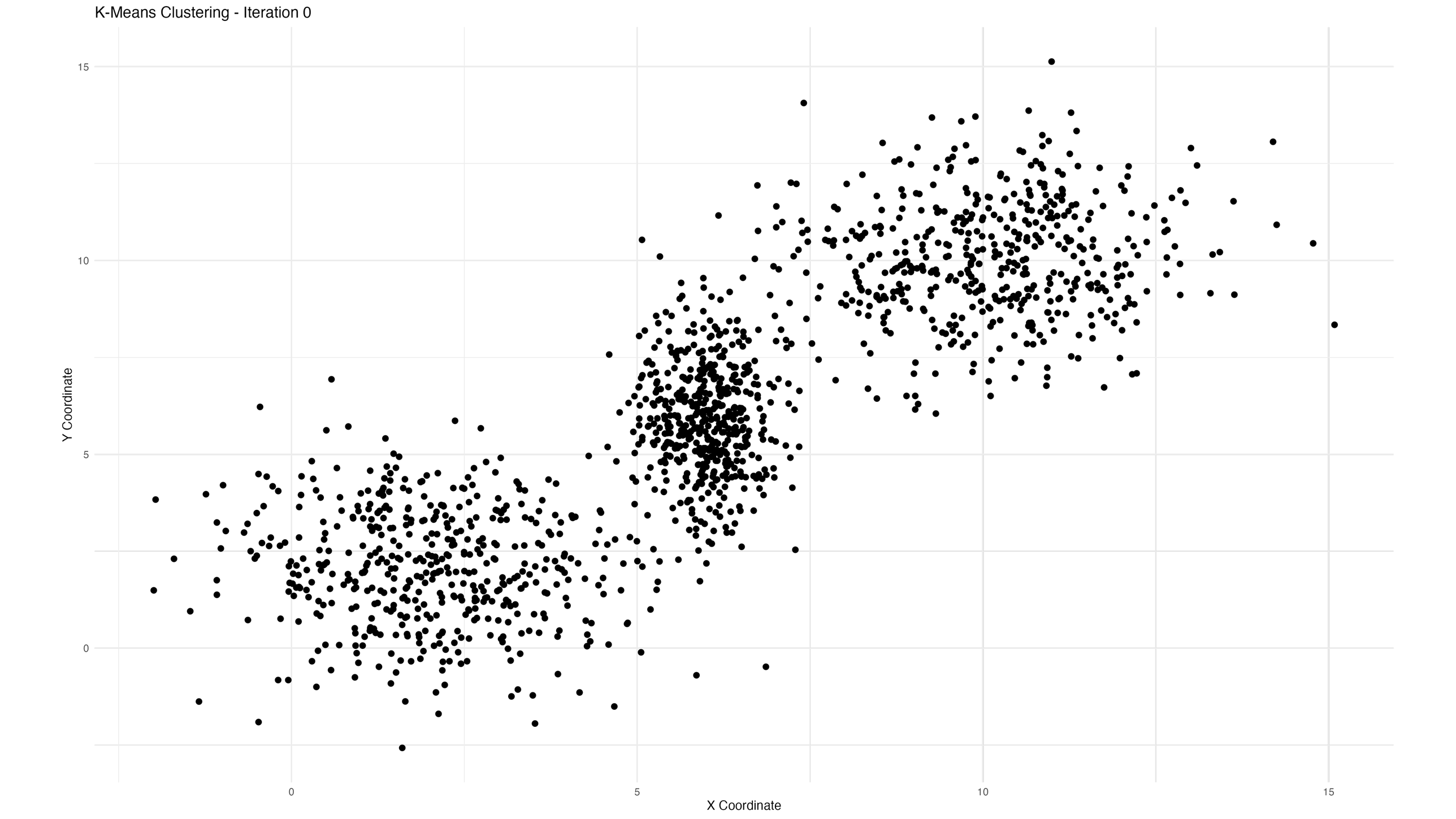
cluster CENTROIDS

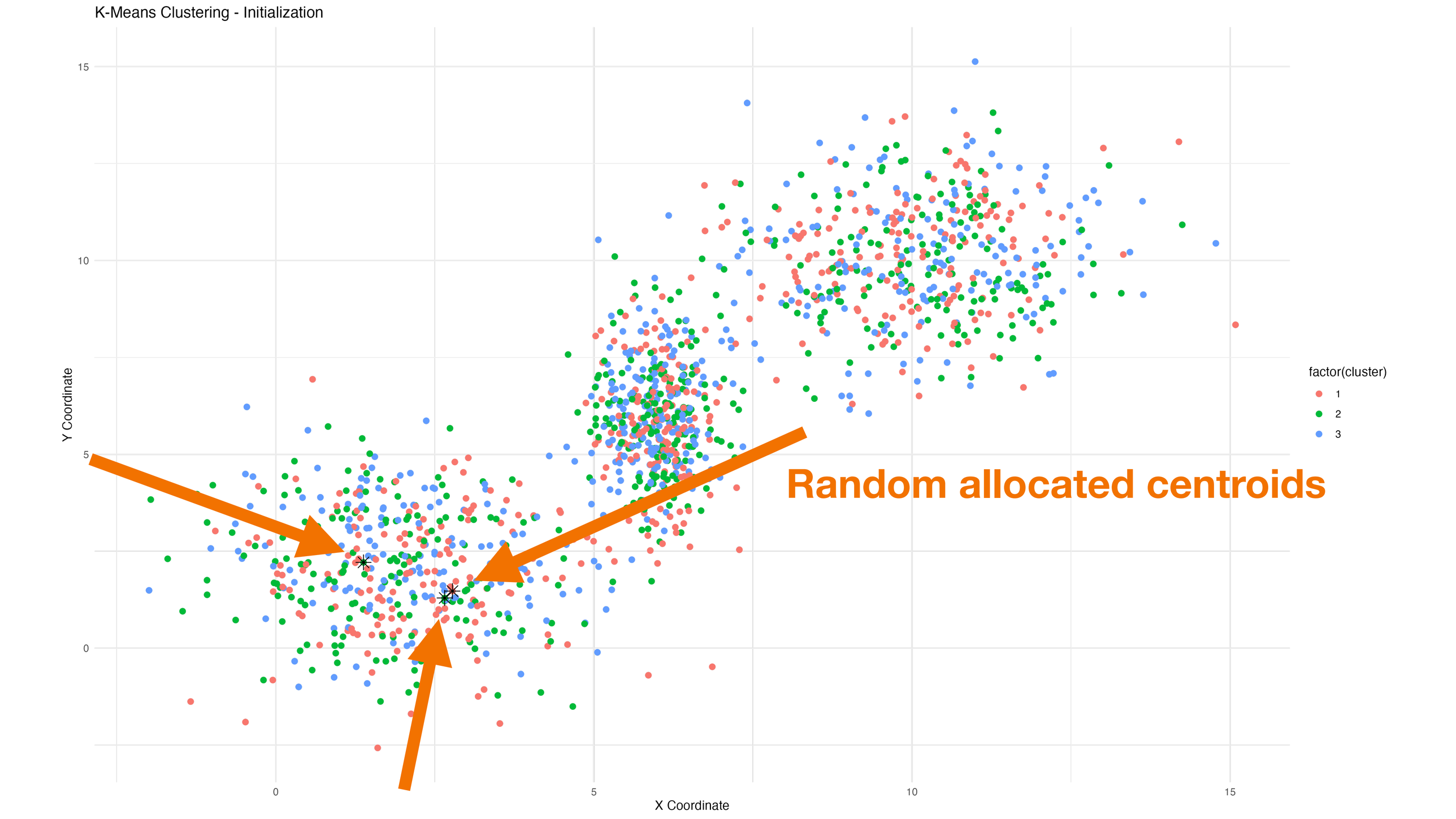


K-Means Clustering algorithm

1. Initialization: Selecting the Number of Clusters (k)

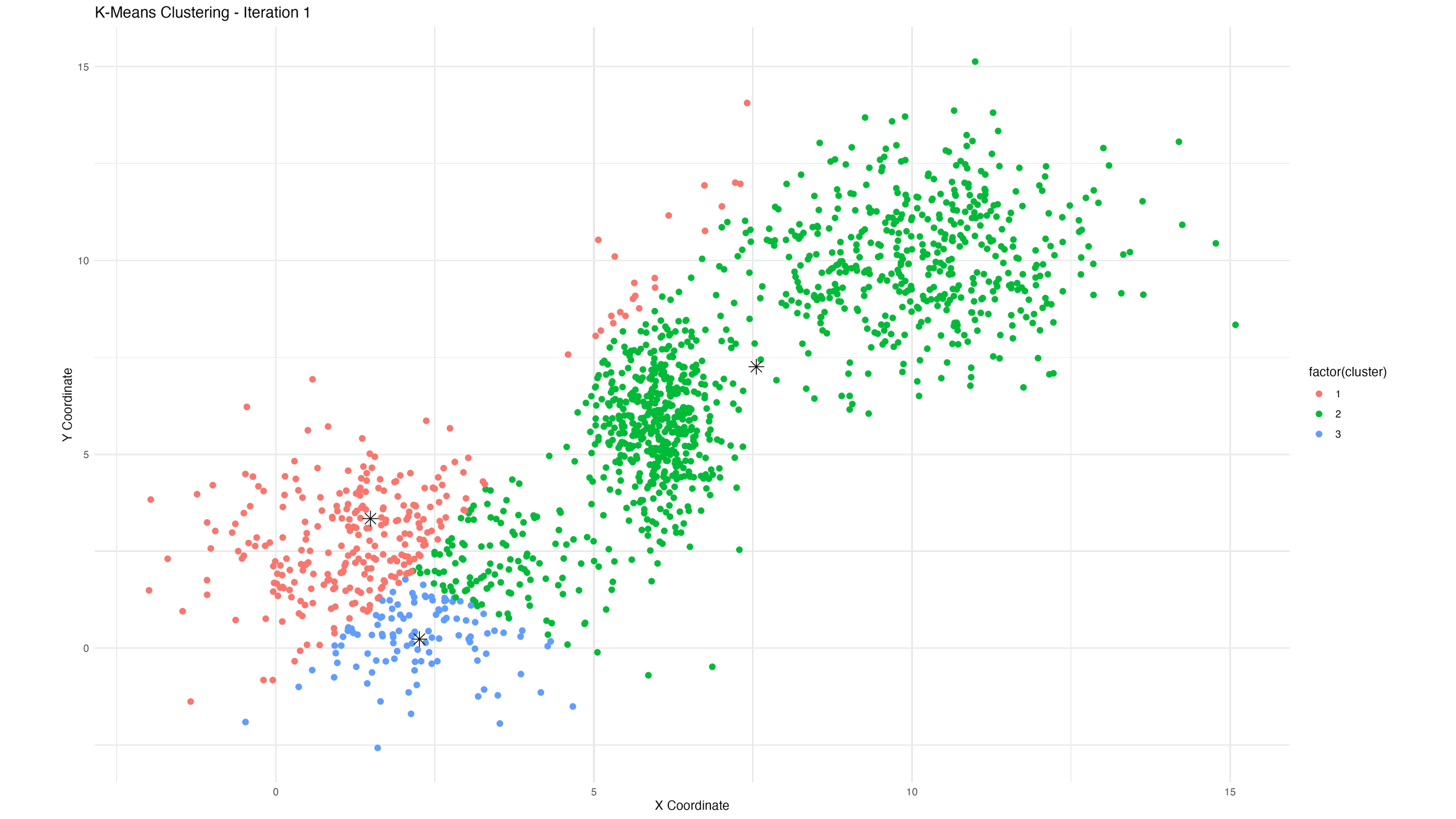
- decide how many clusters (k) you want to find. For instance, if $k=3$, you're aiming to group the data into three distinct clusters.
- You place k initial **centroids** (central points of clusters) randomly within the data space.


K-Means Clustering - Iteration 0
15
10
5
0
Y Coordinate
0
5
10
15
X Coordinate


K-Means Clustering - Initialization
15
10
5
0
Y Coordinate
0
5
10
15
X Coordinate
factor(cluster)
1
2
3
Random allocated centroids

Step 1: Assigning Points to the Nearest Centroid

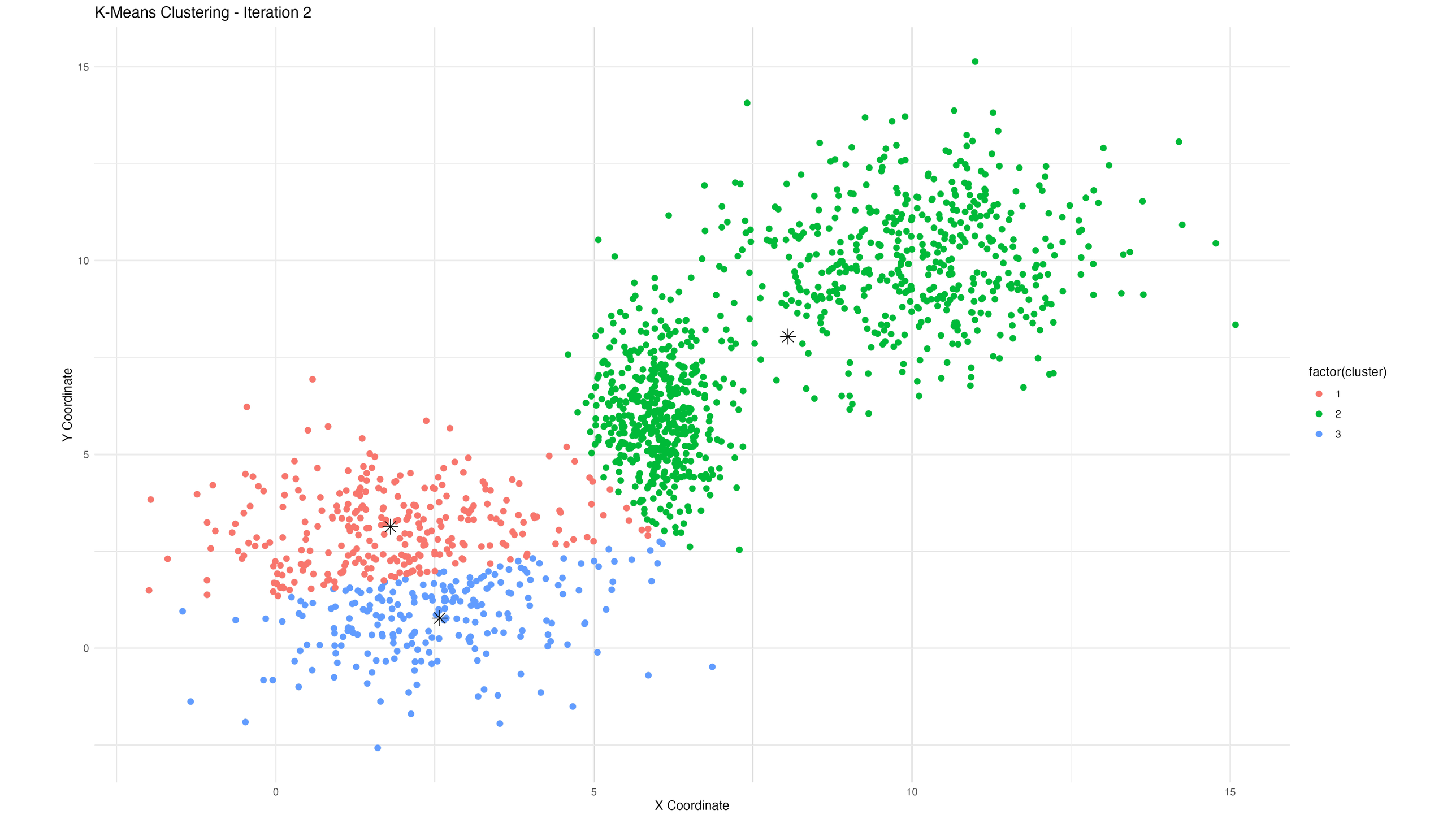
- Picture each data point checking the **distance** to each centroid
- Each point "chooses" the centroid it’s closest to, effectively forming a cluster around that centroid.

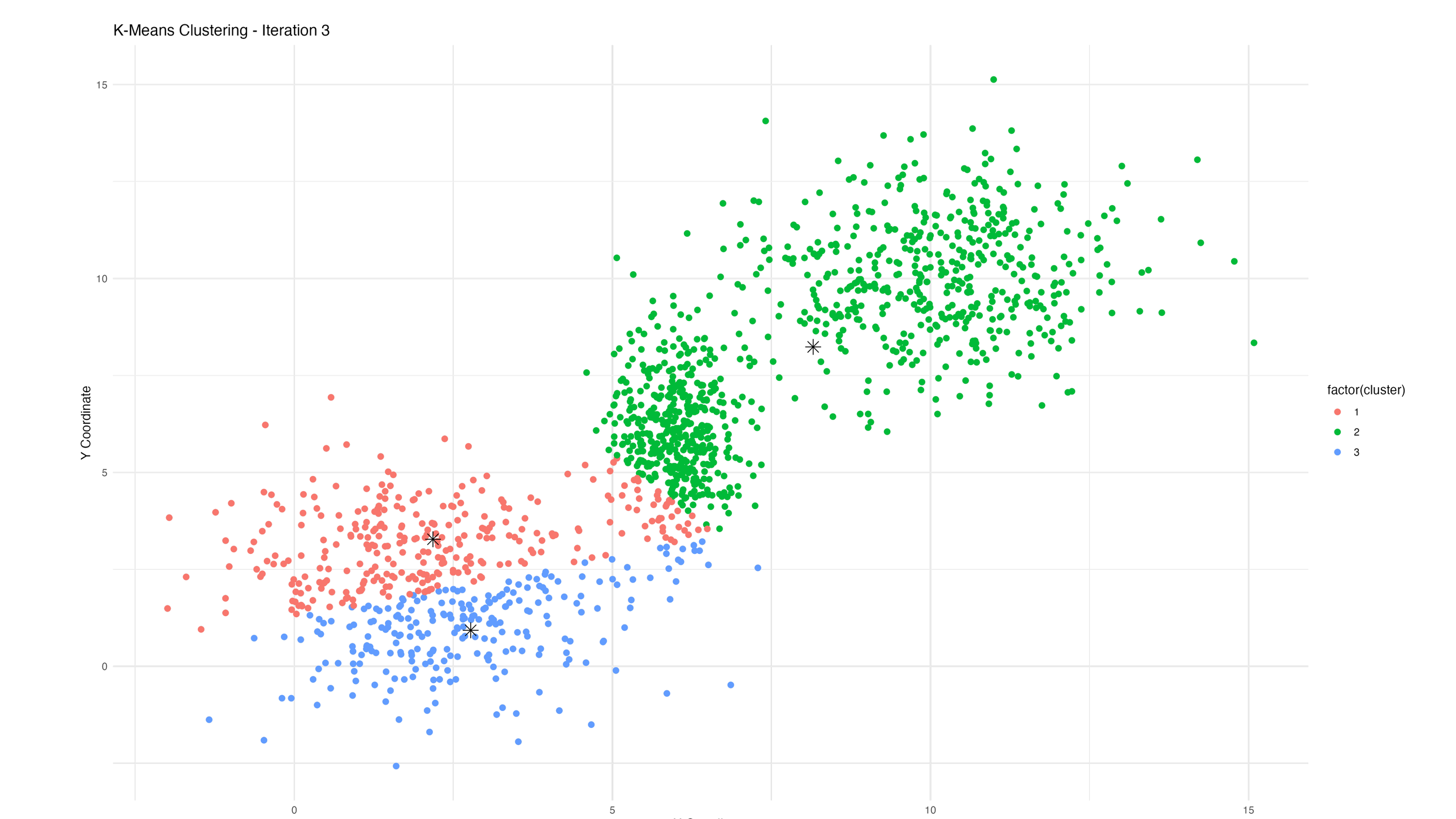

K-Means Clustering - Iteration 1
Y Coordinate
X Coordinate
0
5
10
15
factor(cluster)
1
2
3

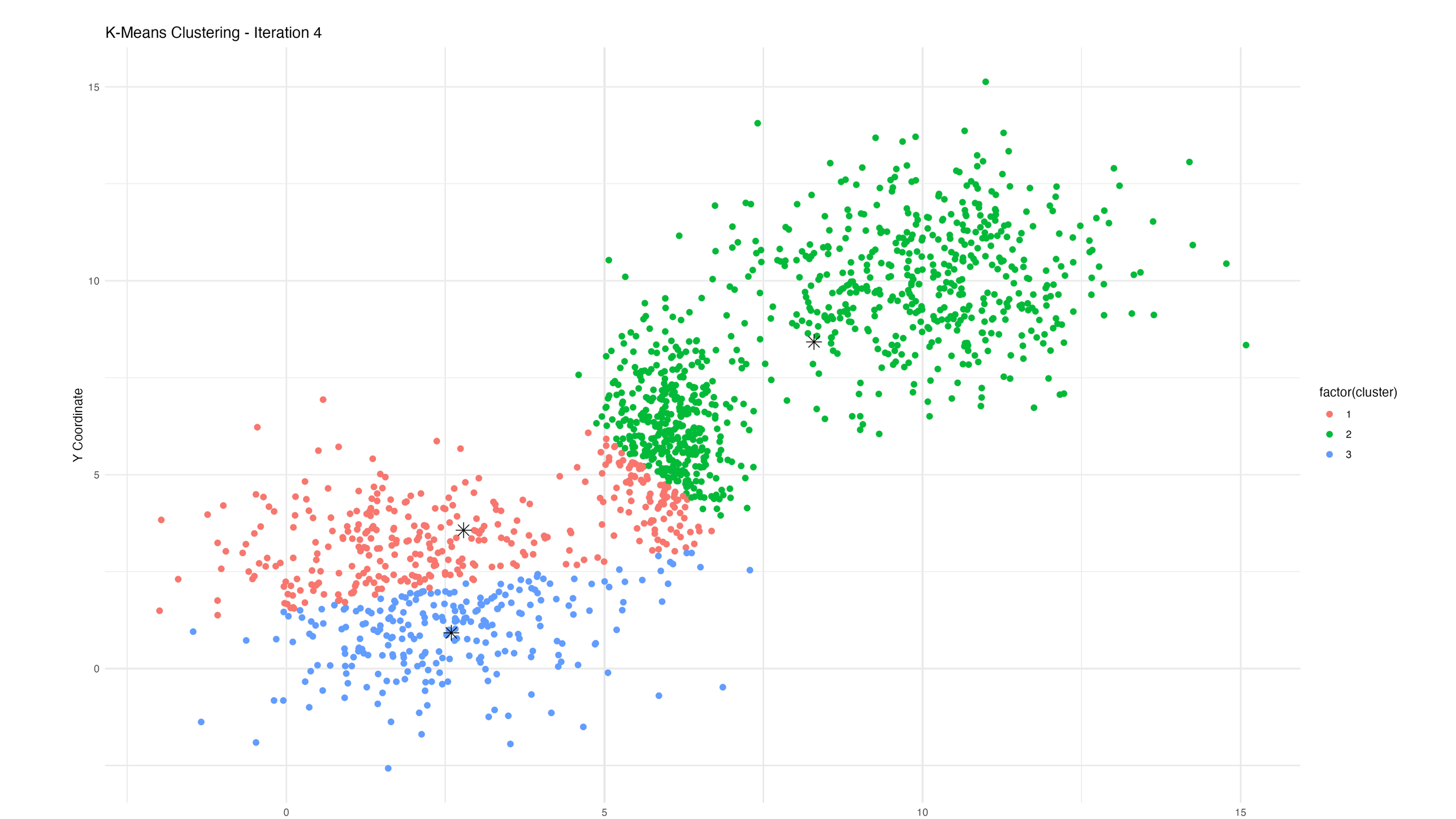
Step 2: Updating Centroids

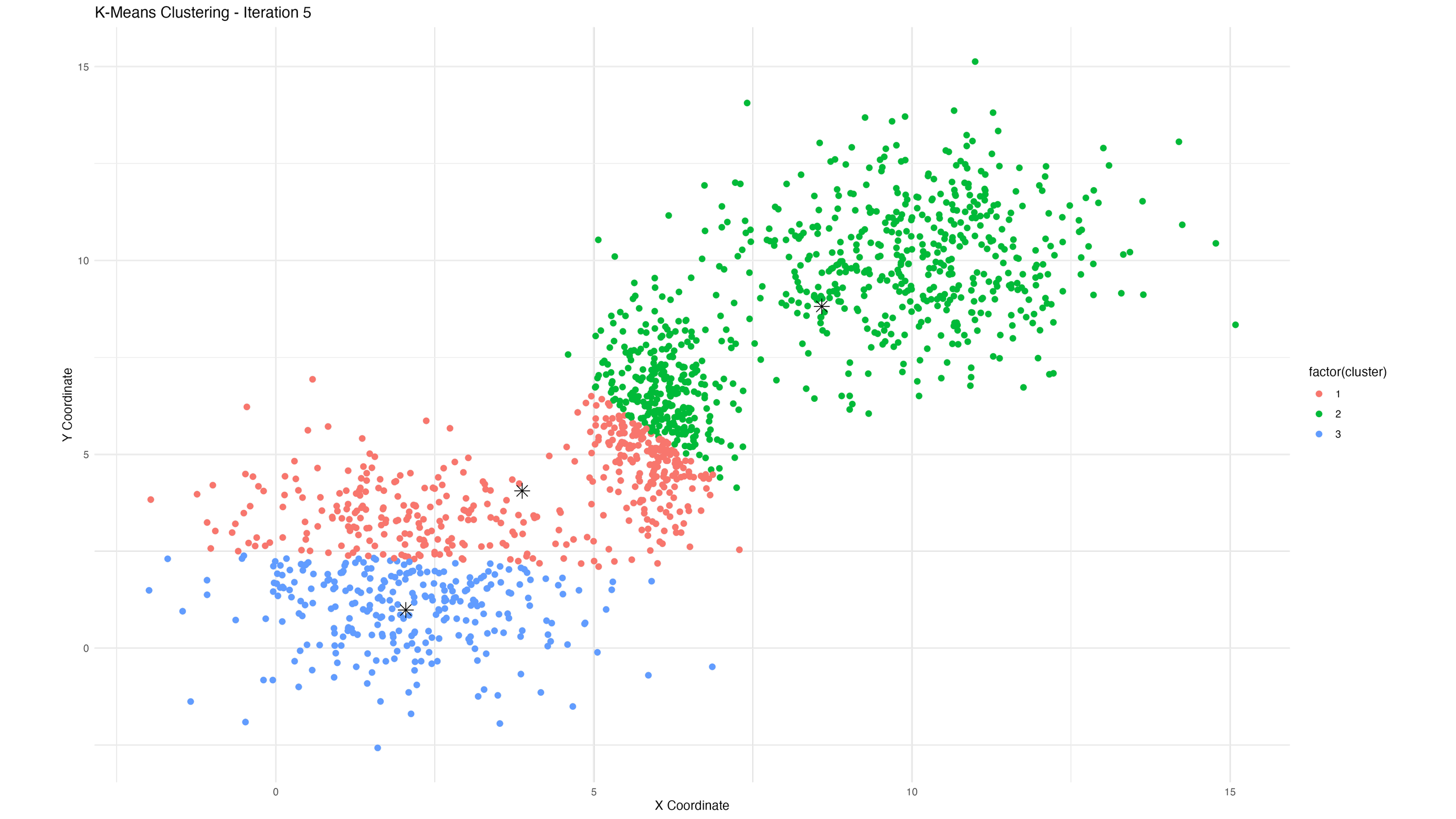
- Take the average position of all the points assigned to each centroid, and move the centroid to this new mean position.
- The centroids shift to the center of their assigned points, pulling them closer to the middle of their respective clusters. You'll see each centroid "jump" to a new location based on where the points around it are.

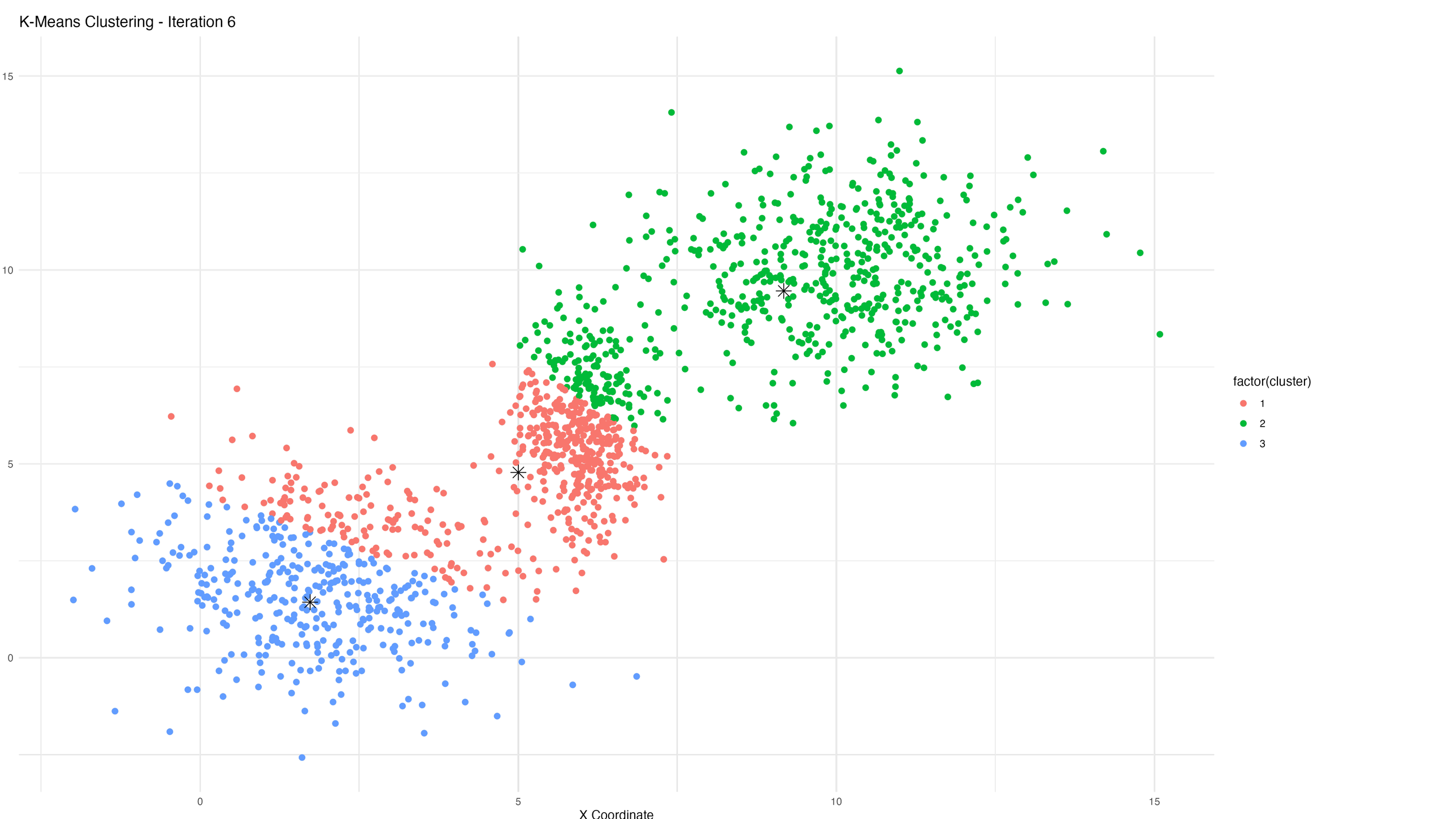
Repeat Steps 1 and 2

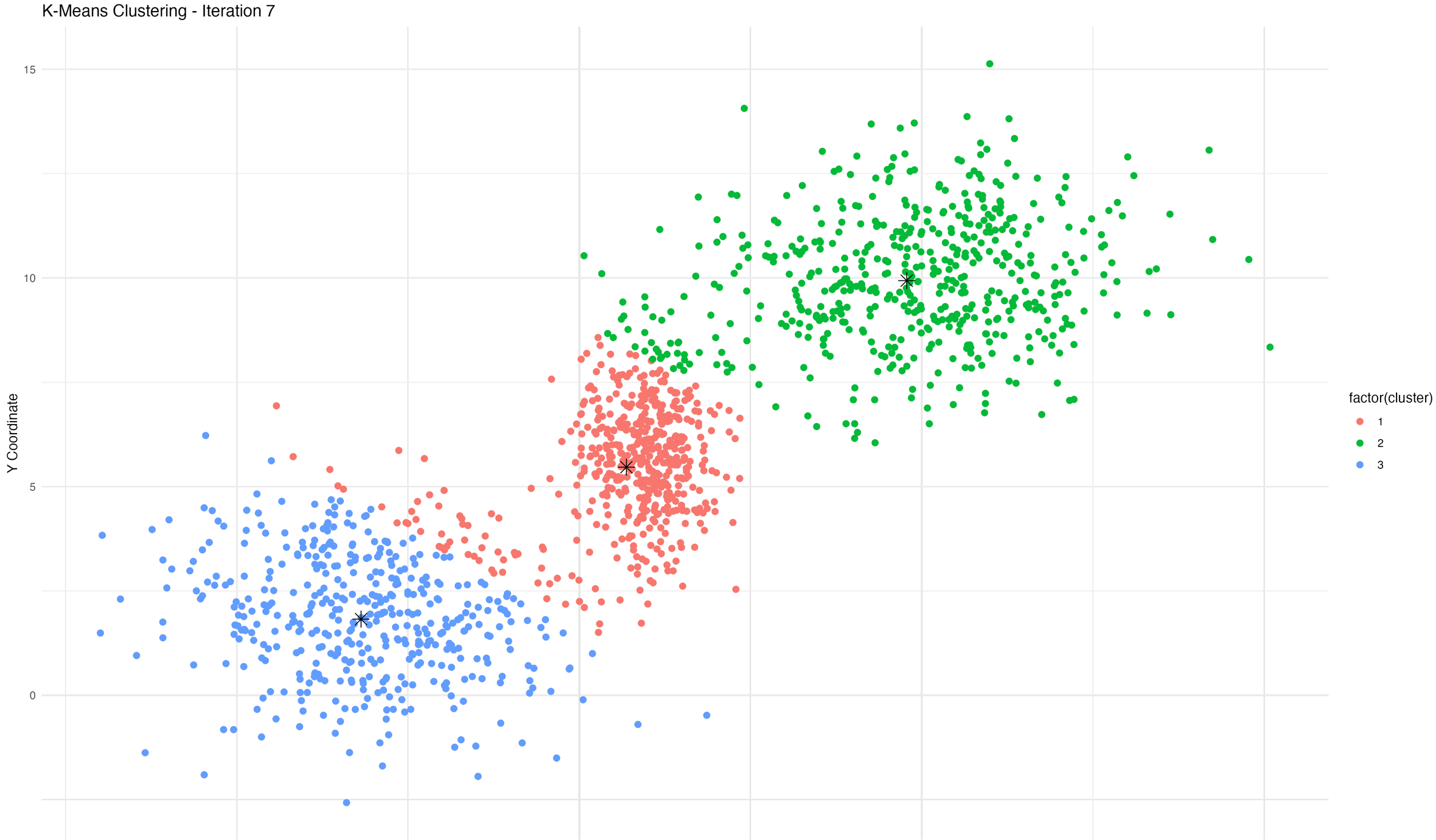

K-Means Clustering - Iteration 2
Y Coordinate
X Coordinate
15
10
5
0
0
5
10
15
factor(cluster)
1
2
3

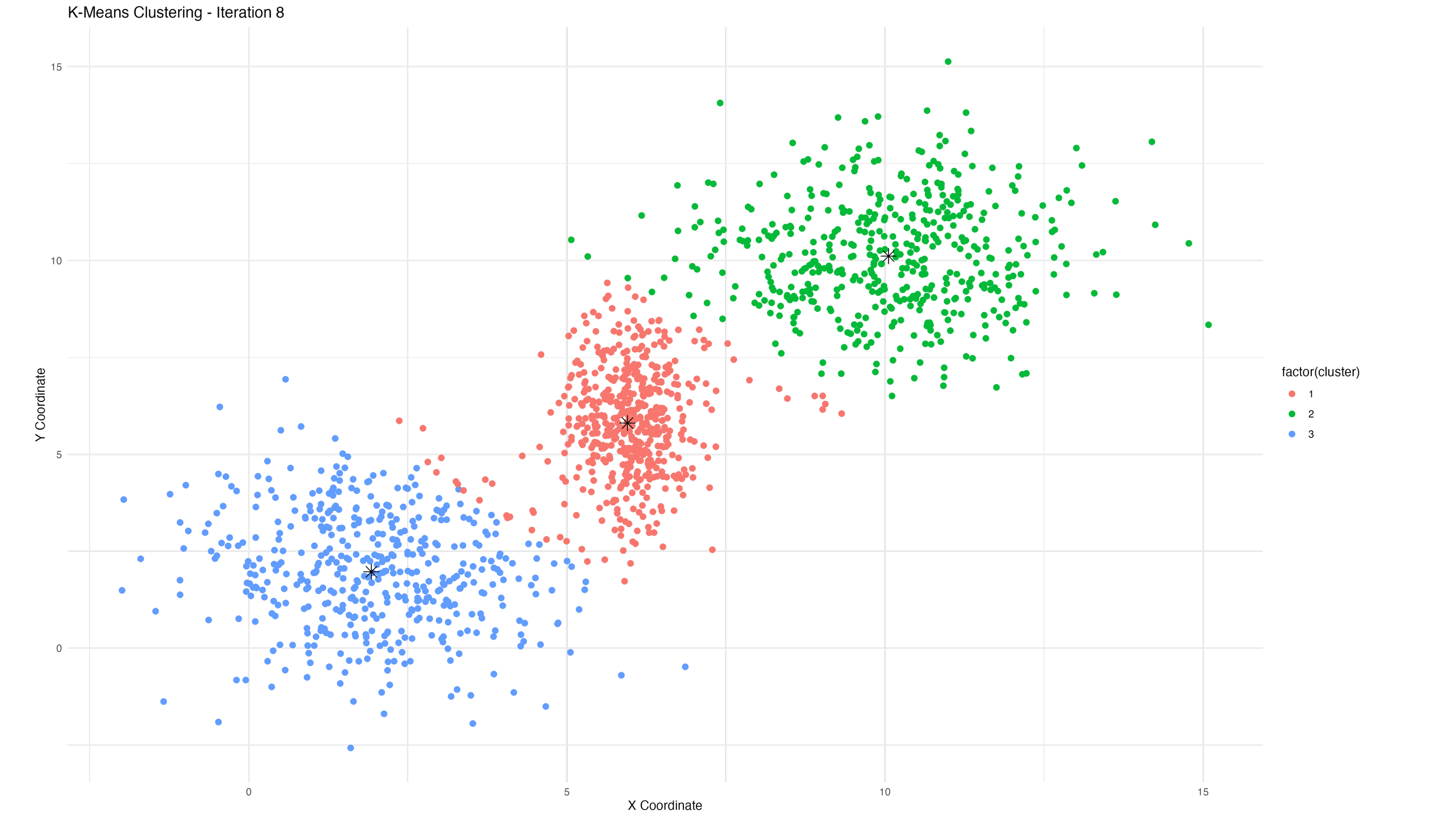

K-Means Clustering - Iteration 3
Y Coordinate
15
10
5
0
0
5
10
15
factor(cluster)
1
2
3

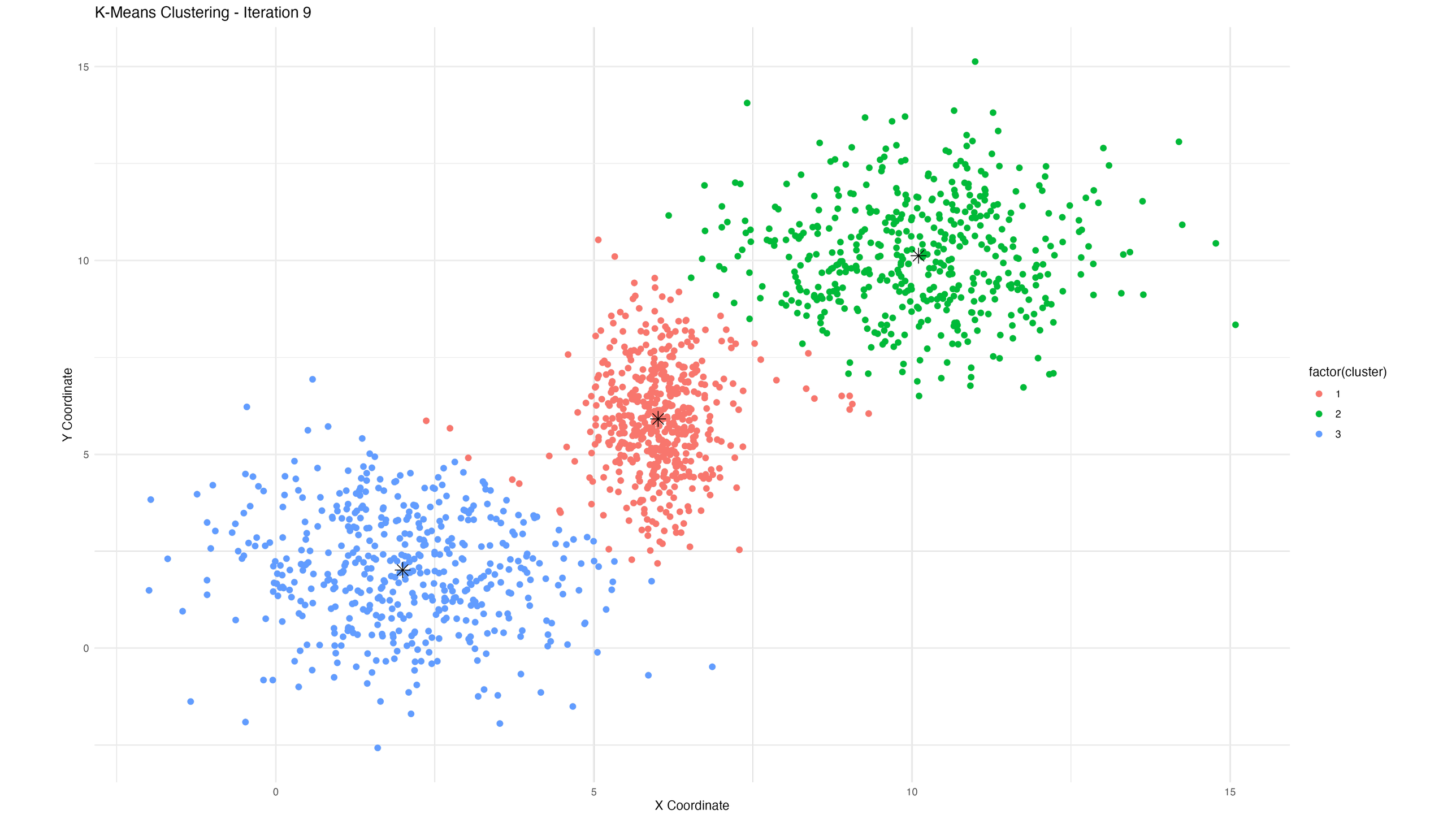

K-Means Clustering - Iteration 4
Y Coordinate
15
10
5
0
0
5
10
15
factor(cluster)
1
2
3

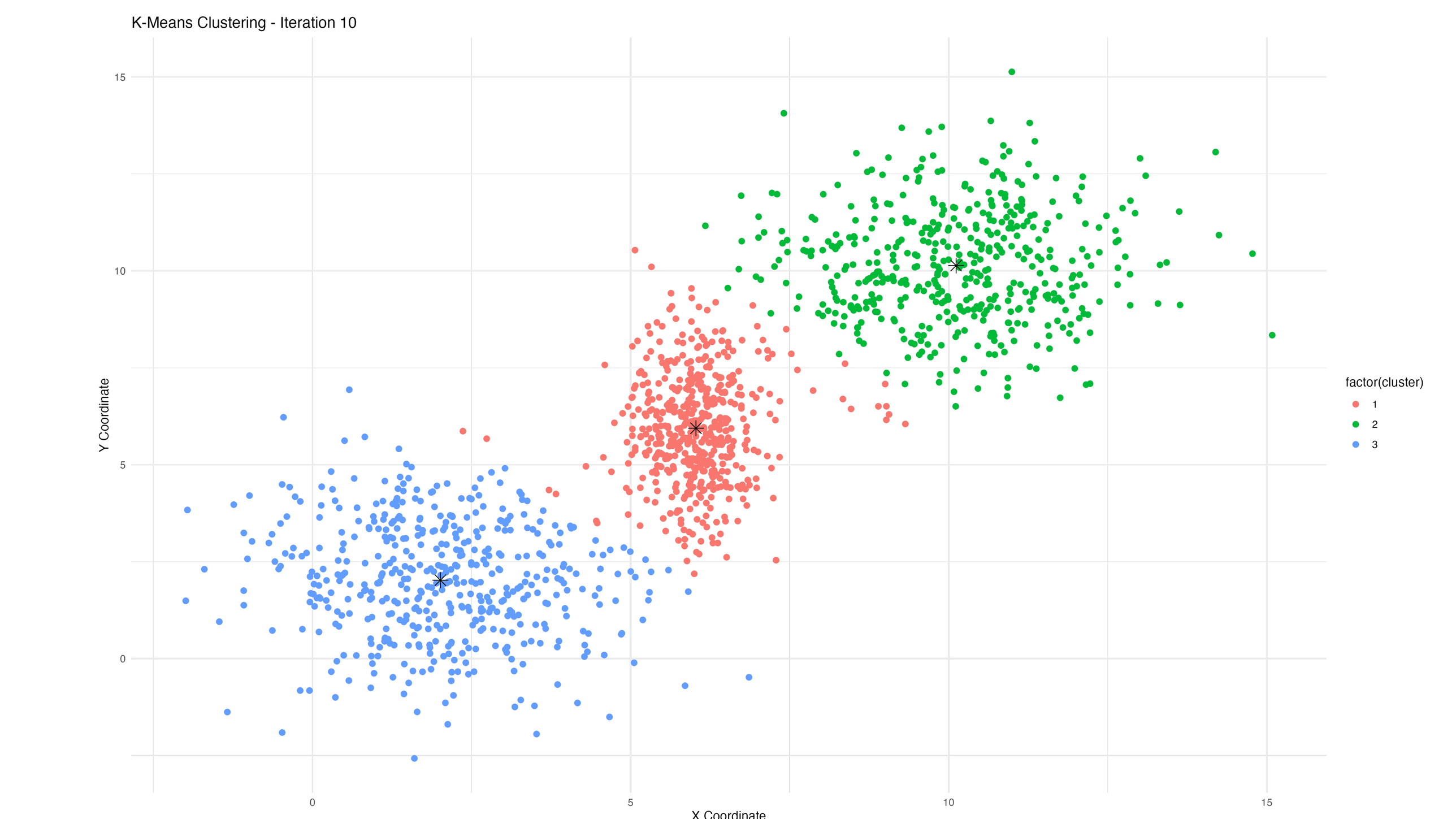

K-Means Clustering - Iteration 5
Y Coordinate
X Coordinate
0
5
10
15
factor(cluster)
1
2
3


K-Means Clustering - Iteration 6
15
10
5
0
0
5
10
15
X Coordinate
factor(cluster)
1
2
3


K-Means Clustering - Iteration 7
15
10
5
0
Y Coordinate
factor(cluster)
1
2
3


K-Means Clustering - Iteration 8
15
10
5
0
Y Coordinate
0
5
10
15
X Coordinate
factor(cluster)
1
2
3

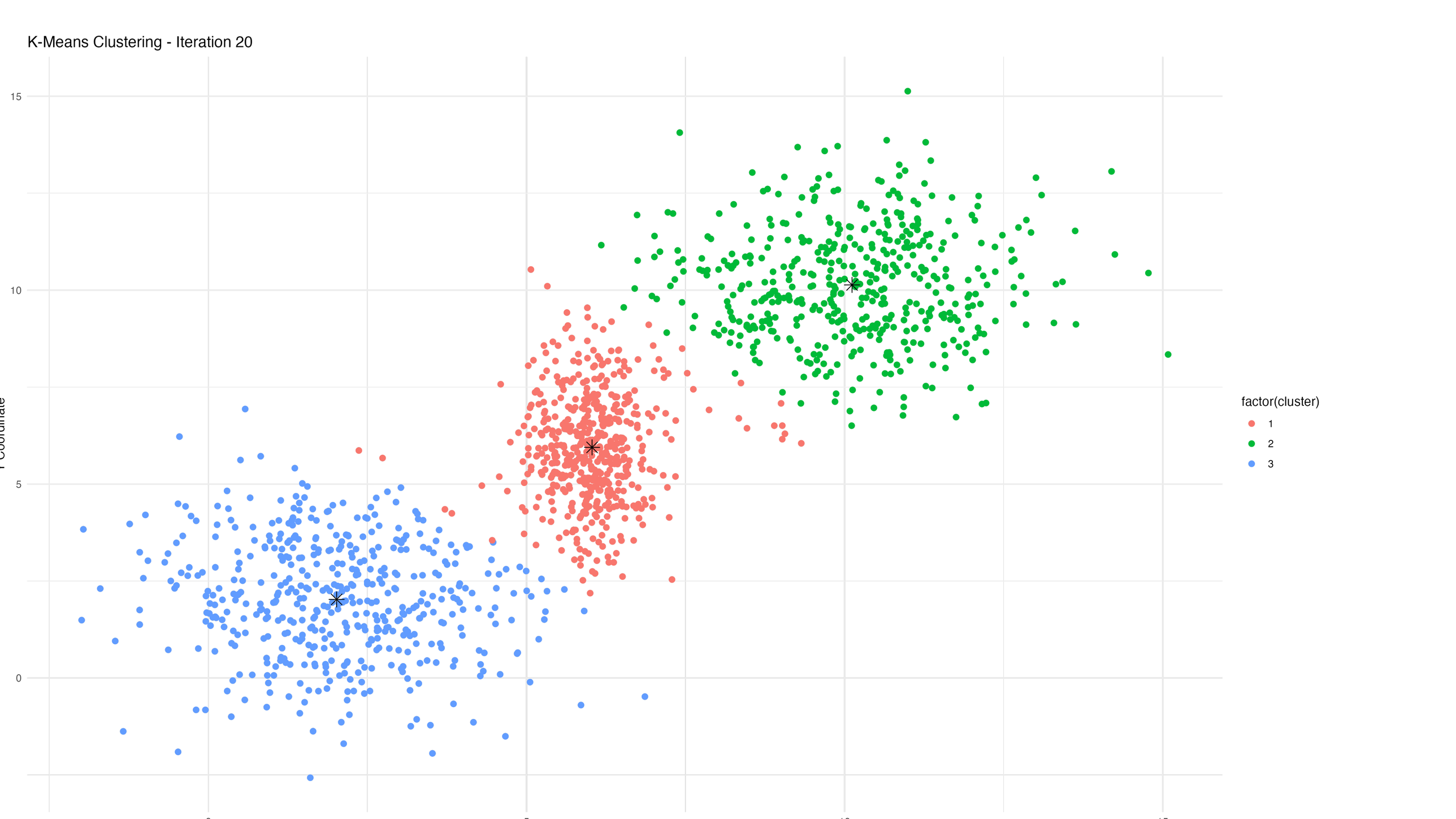

K-Means Clustering - Iteration 9
15
10
5
0
Y Coordinate
0
5
10
15
X Coordinate
factor(cluster)
1
2
3


K-Means Clustering - Iteration 10
Y Coordinate
X Coordinate
15
10
5
0
0
5
10
15
factor(cluster)
1
2
3

Convergence: Clusters Stabilize

Eventually, the centroids stop moving (or move very little). When this happens, the algorithm has "converged."

At this stage, each point is firmly in one cluster, and each centroid represents the center of a cluster.


K-Means Clustering - Iteration 20
15
10
5
0
factor(cluster)
1
2
3

Clustering Distance Measures

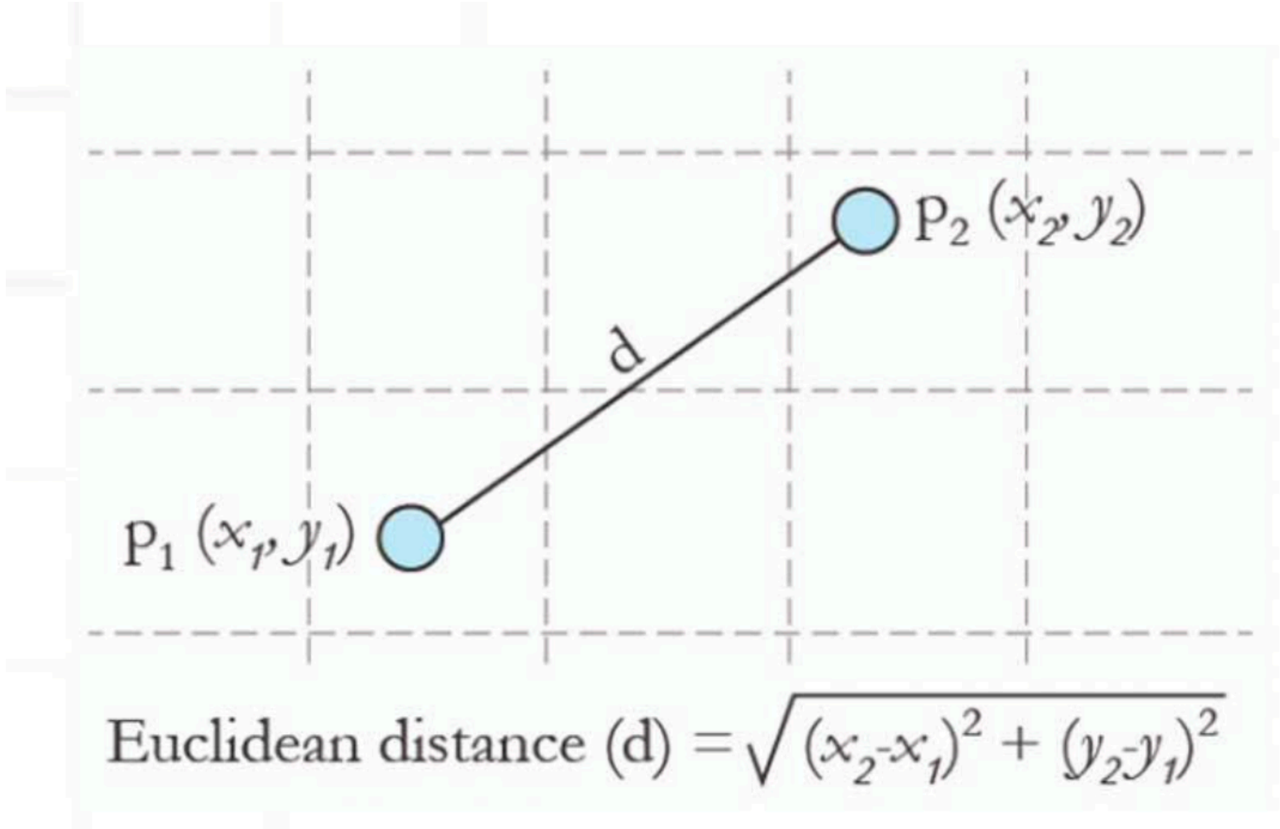
The classification of observations into groups requires some methods for computing the distance or the (dis)similarity between each pair of observations.

The result of this computation is known as a dissimilarity or distance matrix. There are many methods to calculate this distance information;

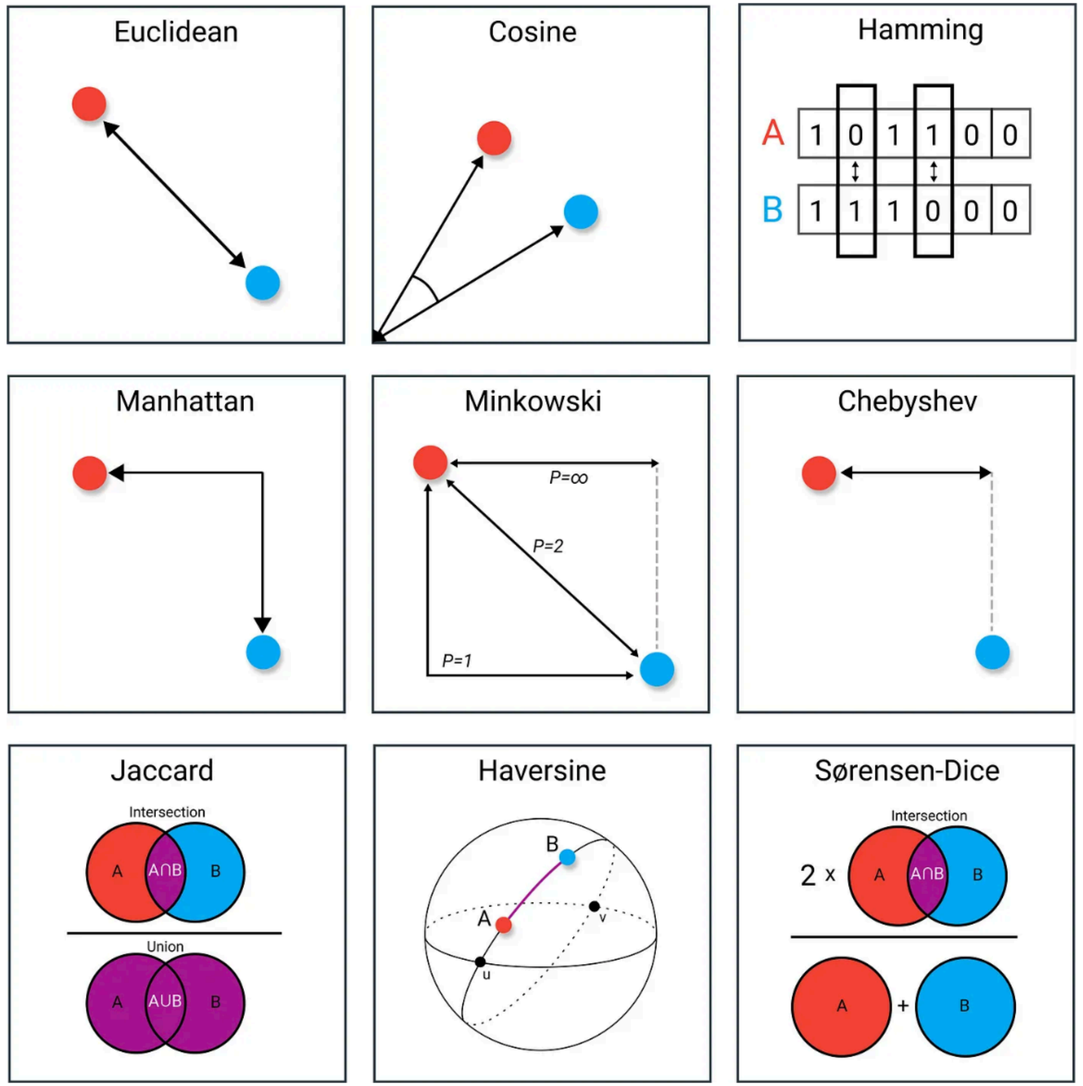
The classical methods for distance measures is *Euclidean distance*, which is defined as follow (in a 2-dimensional space):

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Euclidean distance



Other distances



Dataset USArrests

Built-in R data set USArrest

statistics in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973.

It includes also the percent of the population living in urban areas

```
df <- USArrests
```

To remove any missing value that might be present in the data, type this:

```
df <- na.omit(df)
head(df)
```

```
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
```

Rescaling variables

Let's have a look at the variable `murder'

```
df |>
  select(Murder) |>
  summary()
```

```
    Murder
Min.   : 0.800
1st Qu.: 4.075
Median : 7.250
Mean   : 7.788
3rd Qu.:11.250
Max.   :17.400
```

To center or *re-scale* the variable we can subtract it mean value

```
df |>
  select(Murder) |>
mutate(Murder = Murder - mean(Murder))  |>
  summary()
```

```
     Murder
 Min.   :-6.988
 1st Qu.:-3.713
 Median :-0.538
 Mean   : 0.000
 3rd Qu.: 3.462
 Max.   : 9.612
```

Using the function `scale( )'

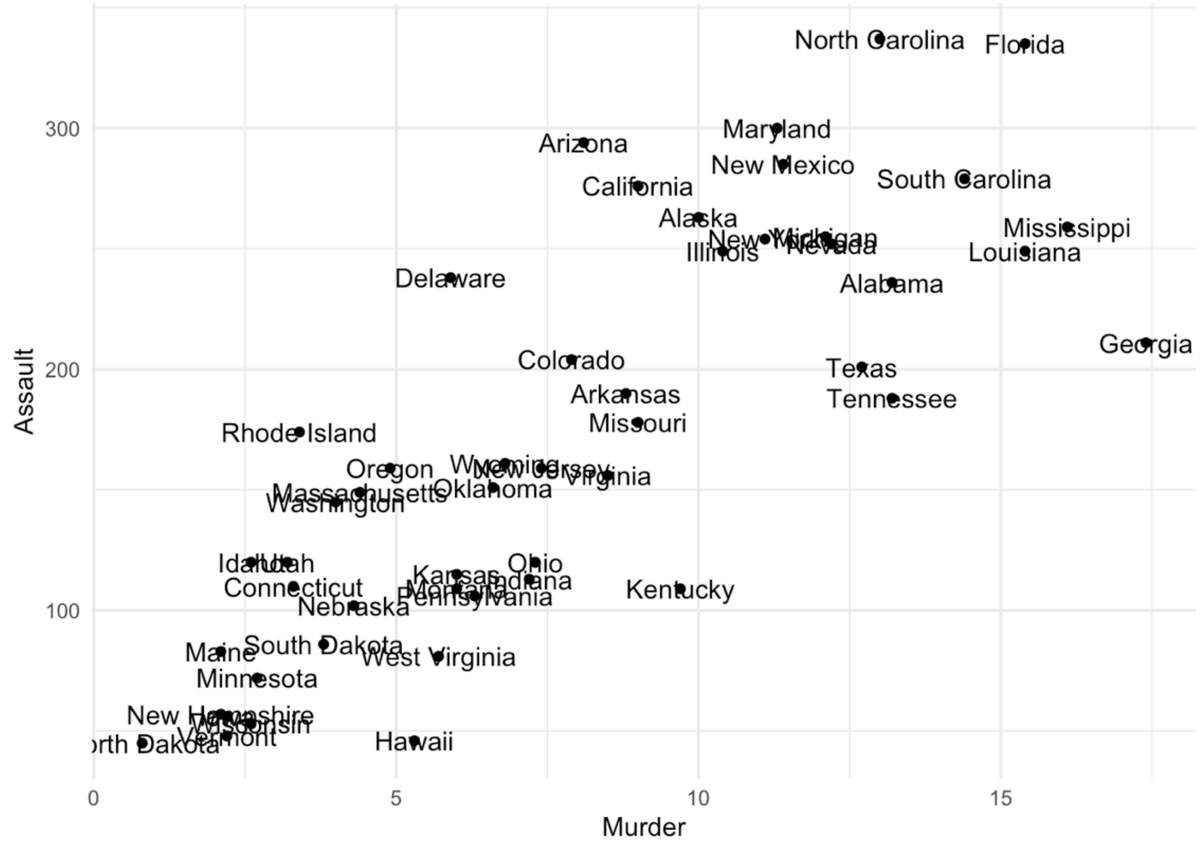
We can rescale all the variables usgin the following command

```
df <- scale(df)
summary(df)
```

```
     Murder          Assault         UrbanPop          Rape        
Min.   :-1.6044   Min.   :-1.5090   Min.   :-2.31714   Min.   :-1.4874  
1st Qu.:-0.8525   1st Qu.:-0.7411   1st Qu.:-0.76271   1st Qu.:-0.6574  
Median :-0.1235   Median :-0.1411   Median : 0.03178   Median :-0.1209  
Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000  
3rd Qu.: 0.7949   3rd Qu.: 0.9388   3rd Qu.: 0.84354   3rd Qu.: 0.5277  
Max.   : 2.2069   Max.   : 1.9948   Max.   : 1.75892   Max.   : 2.6444  
```

Plotting the data

```
USArrests |>
  ggplot(aes(x= Murder, y=Assault))+
   geom_point() +
  geom_text(aes(label = rownames(USArrests)))+
  theme_minimal()
```



Using PCA to reduce dimensions

Murder

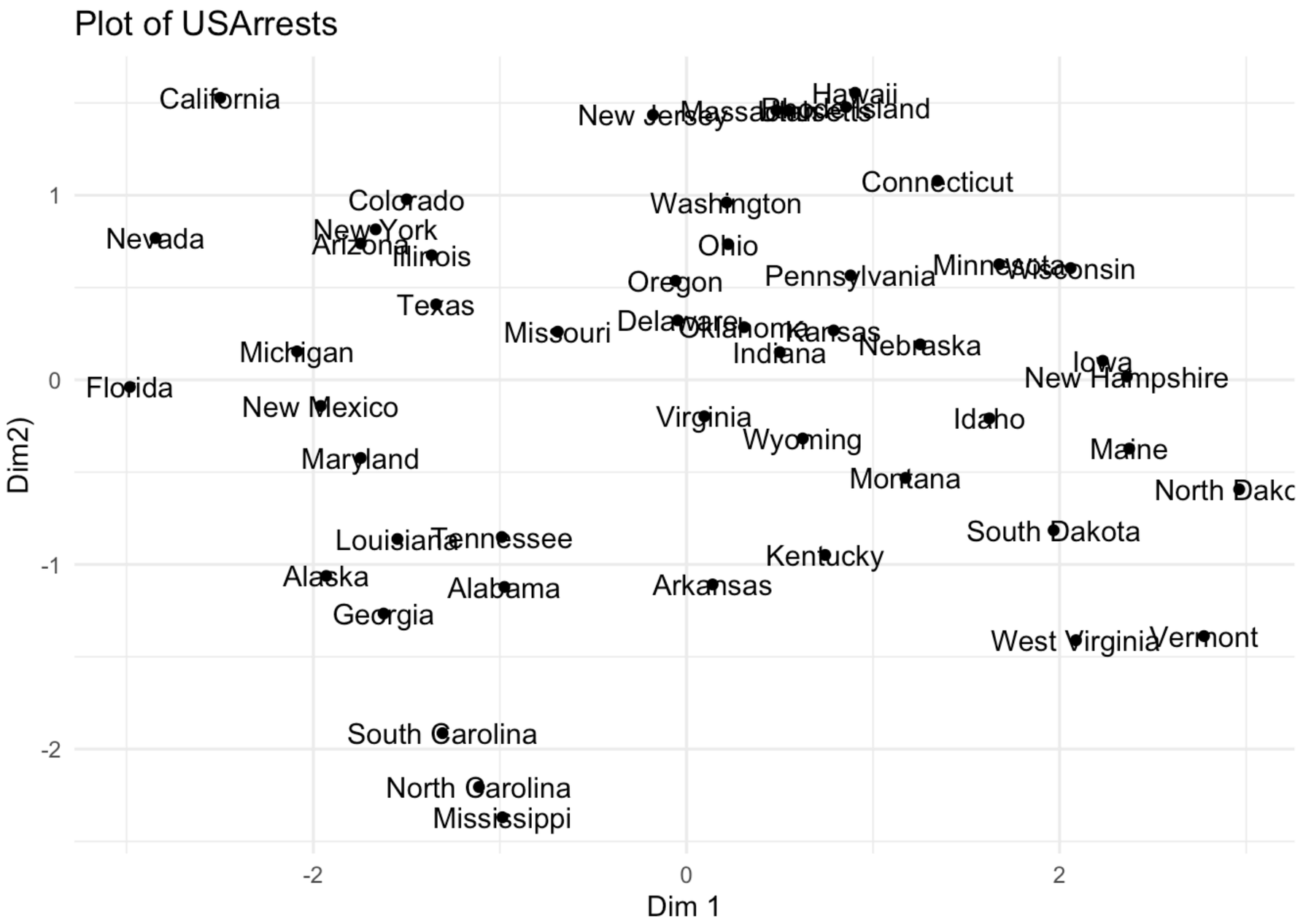
The dataset has multiple variables, so we would need to have a multidimensional plot.

We use then PCA to reduce the dimension and plot the data on a two dimensional space

```
PCA_arrest <- df |>
  as_tibble() |>
  prcomp()
```

These are the plots on the first 2 dimensions

```
PCA_arrest$x |>
  as_tibble() |>
  mutate(state = rownames(USArrests)) |>
  ggplot(aes(x= PC1, y=PC2))+
   geom_point()+
  geom_text(aes(label = state))+
   labs(
    title = 'Plot of USArrests',
    x='Dim 1',
    y='Dim2)'  )+
  theme_minimal()
```


Plot of USArrests
California
New Jersey
Hawaii
Connecticut
Colorado
Washington
Nevada
New York
Arizona
Illinois
Ohio
Minnesota
Wisconsin
Pennsylvania
Oregon
Texas
Delaware
Missouri
Oklahoma
Kansas
Michigan
Indiana
Nebraska
Iowa
New Hampshire
Florida
New Mexico
Virginia
Idaho
Wyoming
Maine
Maryland
Montana
North Dako
South Dakota
Louisiana
Tennessee
Kentucky
Alaska
Alabama
Arkansas
Georgia
West Virginia
Vermont
South Carolina
North Carolina
Mississippi
1
0
-1
-2
-2
0
2
Dim2)
Dim 1

Performing k-means with 2 clusters

We now load two packages that we need for the analysis

```
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

Random number generator. To assure replicability

```
set.seed(12345)
```

This is the function to perfrom k-means clustering

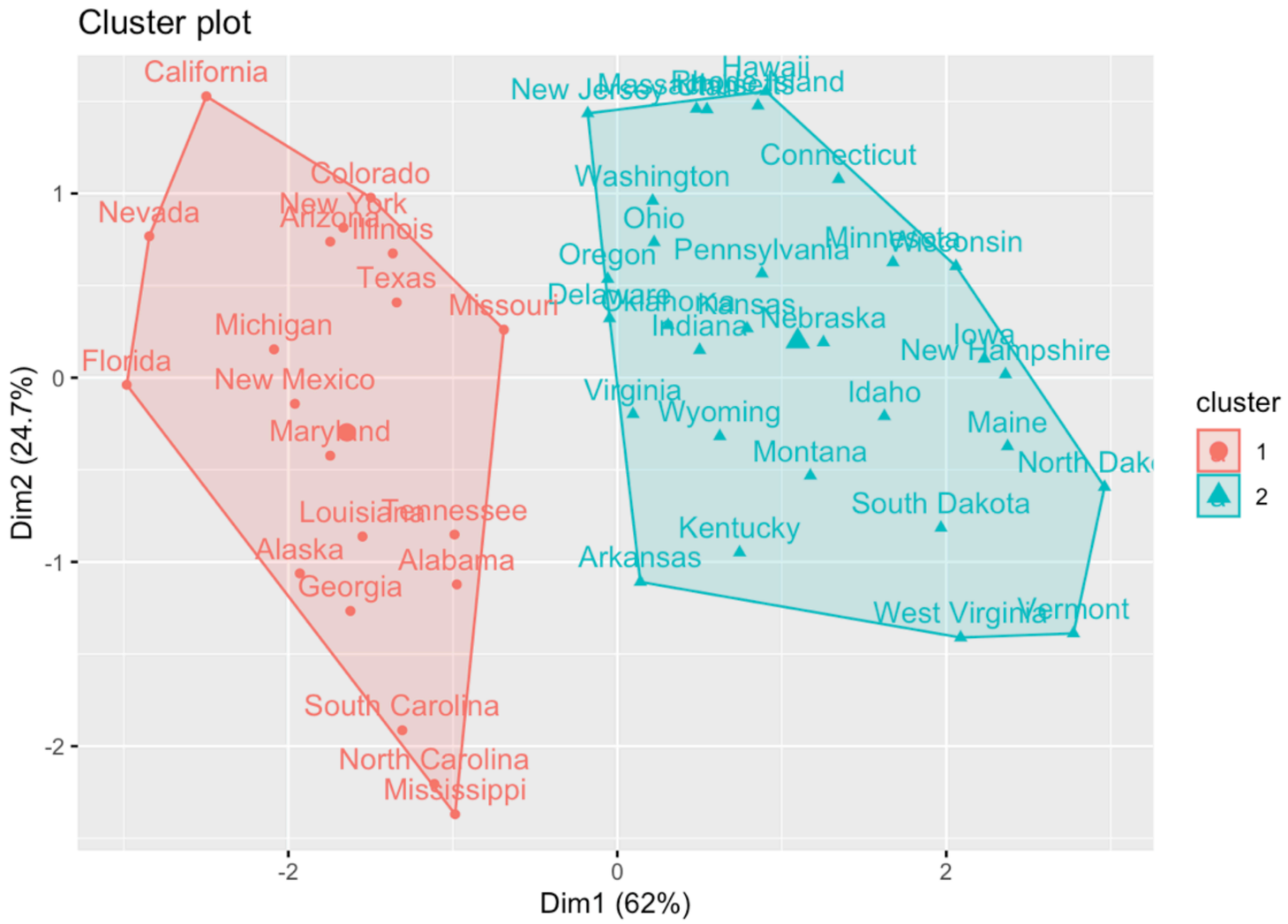
```
k2 <- kmeans(df, centers = 2)
str(k2)
```

```
List of 9
 $ cluster     : Named int [1:50] 1 1 1 2 1 1 2 2 1 1 ...
  ..- attr(*, "names")= chr [1:50] "Alabama" "Alaska" "Arizona" "Arkansas" ...
 $ centers     : num [1:2, 1:4] 1.005 -0.67 1.014 -0.676 0.198 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:2] "1" "2"
  .. ..$ : chr [1:4] "Murder" "Assault" "UrbanPop" "Rape"
 $ totss       : num 196
 $ withinss    : num [1:2] 46.7 56.1
 $ tot.withinss: num 103
 $ betweenss   : num 93.1
 $ size        : int [1:2] 20 30
 $ iter        : int 1
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
```

```
K-means clustering with 2 clusters of sizes 20, 30

Cluster means:
     Murder    Assault   UrbanPop       Rape
1  1.004934  1.0138274  0.1975853  0.8469650
2 -0.669956 -0.6758849 -0.1317235 -0.5646433

Clustering vector:
       Alabama         Alaska        Arizona       Arkansas     California
             1              1              1              2              1
      Colorado    Connecticut       Delaware        Florida        Georgia
             1              2              2              1              1
        Hawaii          Idaho       Illinois        Indiana           Iowa
             2              2              1              2              2
        Kansas       Kentucky      Louisiana          Maine       Maryland
             2              2              1              2              1
 Massachusetts       Michigan      Minnesota    Mississippi       Missouri
             2              1              2              1              1
       Montana       Nebraska         Nevada  New Hampshire     New Jersey
             2              2              1              2              2
    New Mexico       New York North Carolina   North Dakota           Ohio
             1              1              1              2              2
      Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
             2              2              2              2              1
  South Dakota      Tennessee          Texas           Utah        Vermont
             2              1              1              2              2
      Virginia     Washington  West Virginia      Wisconsin        Wyoming
             2              2              2              2              2

Within cluster sum of squares by cluster:
[1] 46.74796 56.11445
 (between_SS / total_SS =  47.5 %)
```

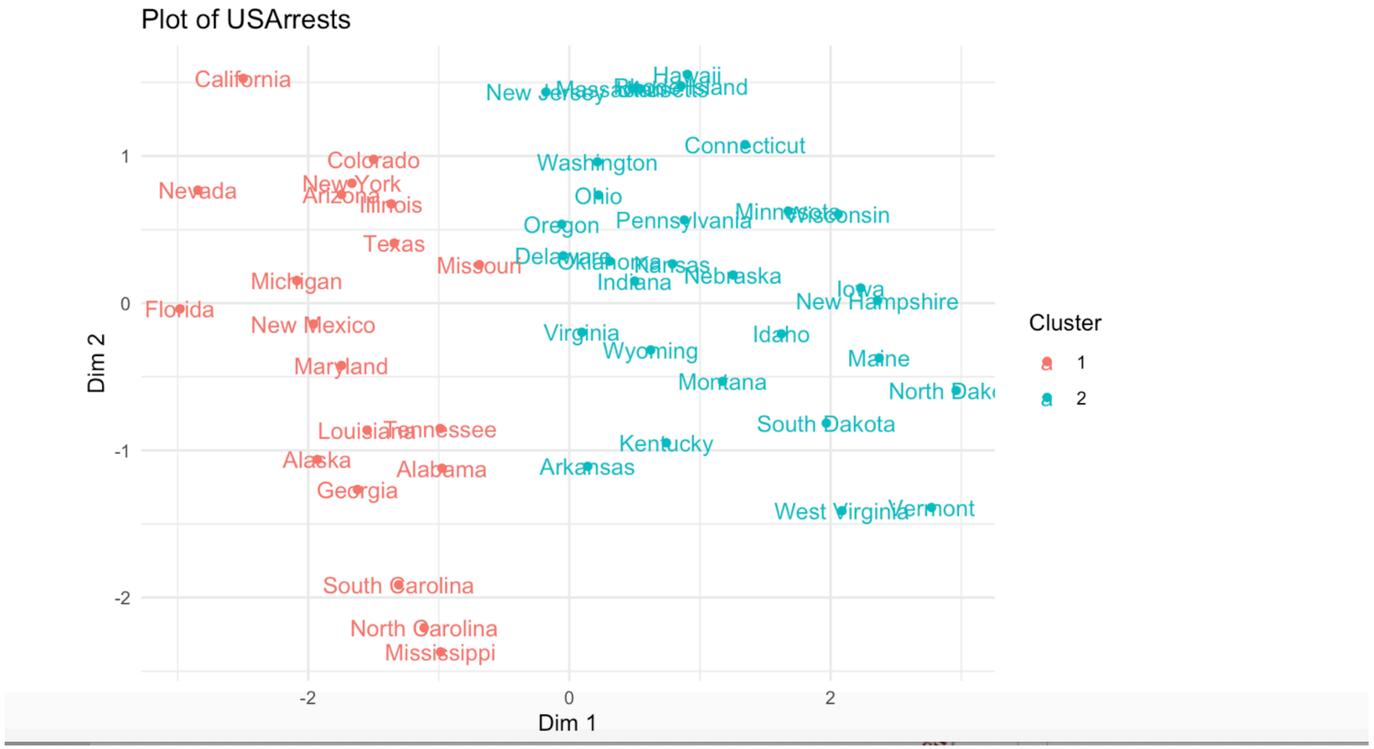
We use the package `factoextra' to plot the clustering result

```
fviz_cluster(k2, data = df)
```



Plotting with ggplot

```
PCA_arrest$x |>
  as_tibble() |>
  mutate(state = rownames(USArrests),
         cluster = k2$cluster) |>
  ggplot(aes(x= PC1, y=PC2, col=factor(cluster)))+
   geom_point()+
  geom_text(aes(label = state))+
   labs(
    title = 'Plot of USArrests',
    x = 'Dim 1',
    y = 'Dim 2',
    color = "Cluster")+
  theme_minimal()
```

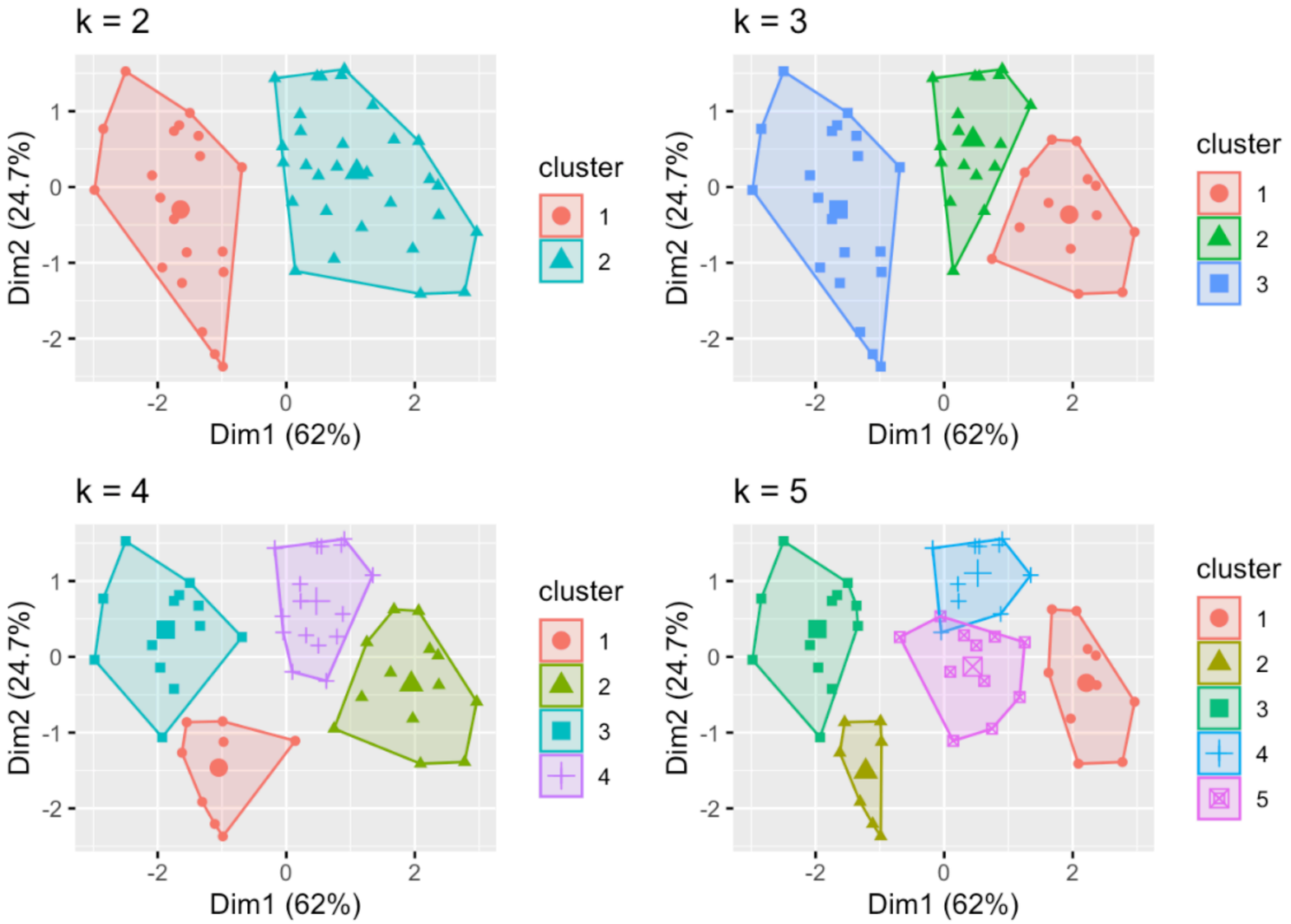


Varying the number of clusters

We can now repeat with multiple cluster solutions

```
k3 <- kmeans(df, centers = 3, nstart = 25)
k4 <- kmeans(df, centers = 4, nstart = 25)
k5 <- kmeans(df, centers = 5, nstart = 25)

# plots to compare
p1 <- fviz_cluster(k2, geom = "point", data = df) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = df) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = df) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = df) + ggtitle("k = 5")

library(gridExtra)
```

```
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



Finding the best number of cluster

the basic idea behind k-means clustering, is to define clusters such that the total intra-cluster variation (known as total within-cluster variation or total within-cluster sum of square) is minimized:

$$min \sum_{k=1}^{K} [W(C_k)]$$

Where C_k is the k-th cluster and $W(C_k)$ is the within-cluster variation.

“Elbow method”

1. Compute clustering algorithm (e.g., k-means clustering) for different values of *k*. For instance, by varying *k* from 1 to 10 clusters

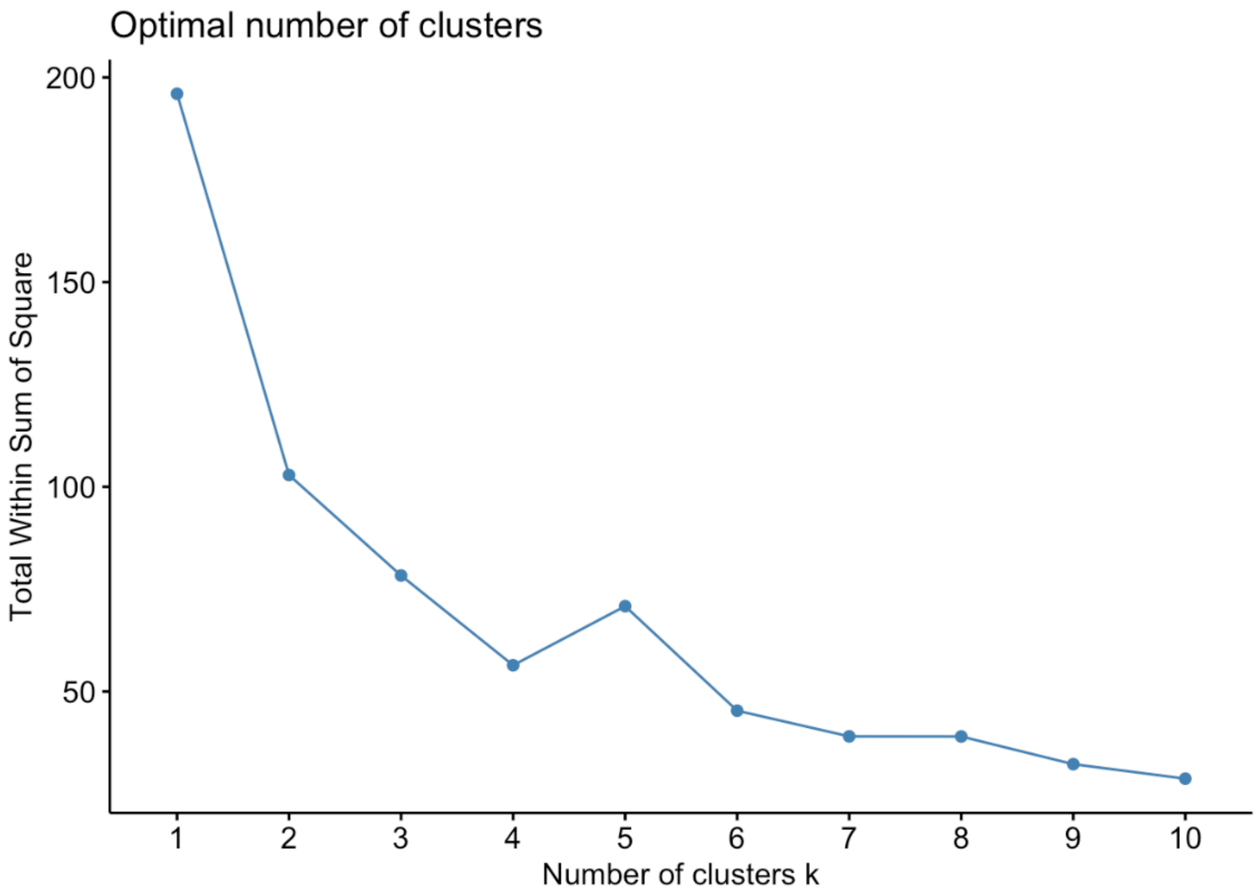
2. For each *k*, calculate the total within-cluster sum of square (wss)

3. Plot the curve of wss according to the number of clusters *k*.

4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

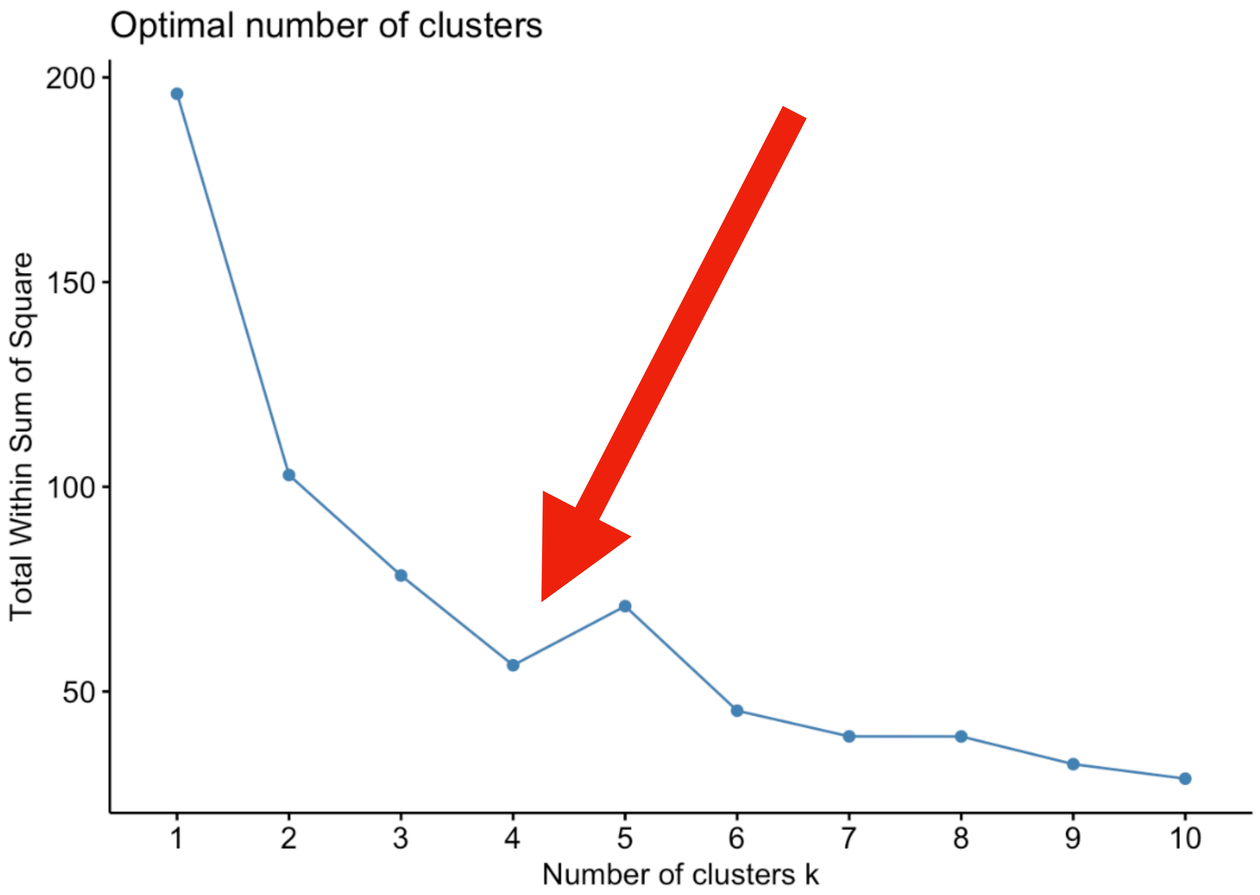
“Elbow method”

```
fviz_nbclust(df, kmeans, method = "wss")
```

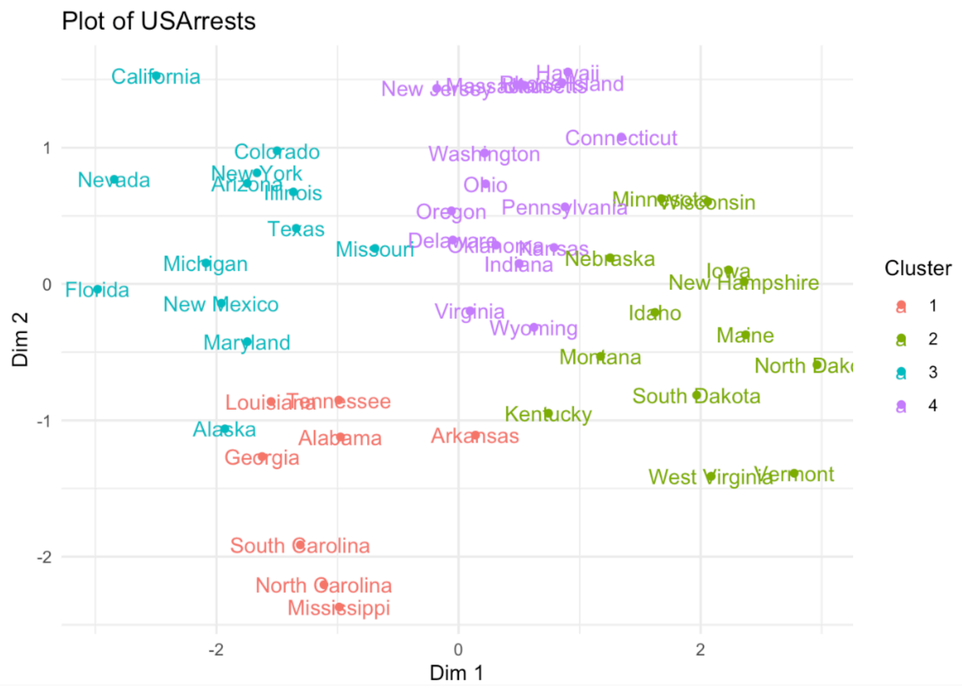


“Elbow method”

```
fviz_nbclust(df, kmeans, method = "wss")
```



```
PCA_arrest$x |>
  as_tibble() |>
  mutate(state = rownames(USArrests),
         cluster = k4$cluster) |>
  ggplot(aes(x= PC1, y=PC2, col=factor(cluster)))+
   geom_point()+
  geom_text(aes(label = state))+
   labs(
    title = 'Plot of USArrests',
    x='Dim 1',
    y='Dim 2',
    color="Cluster") +
  theme_minimal()
```

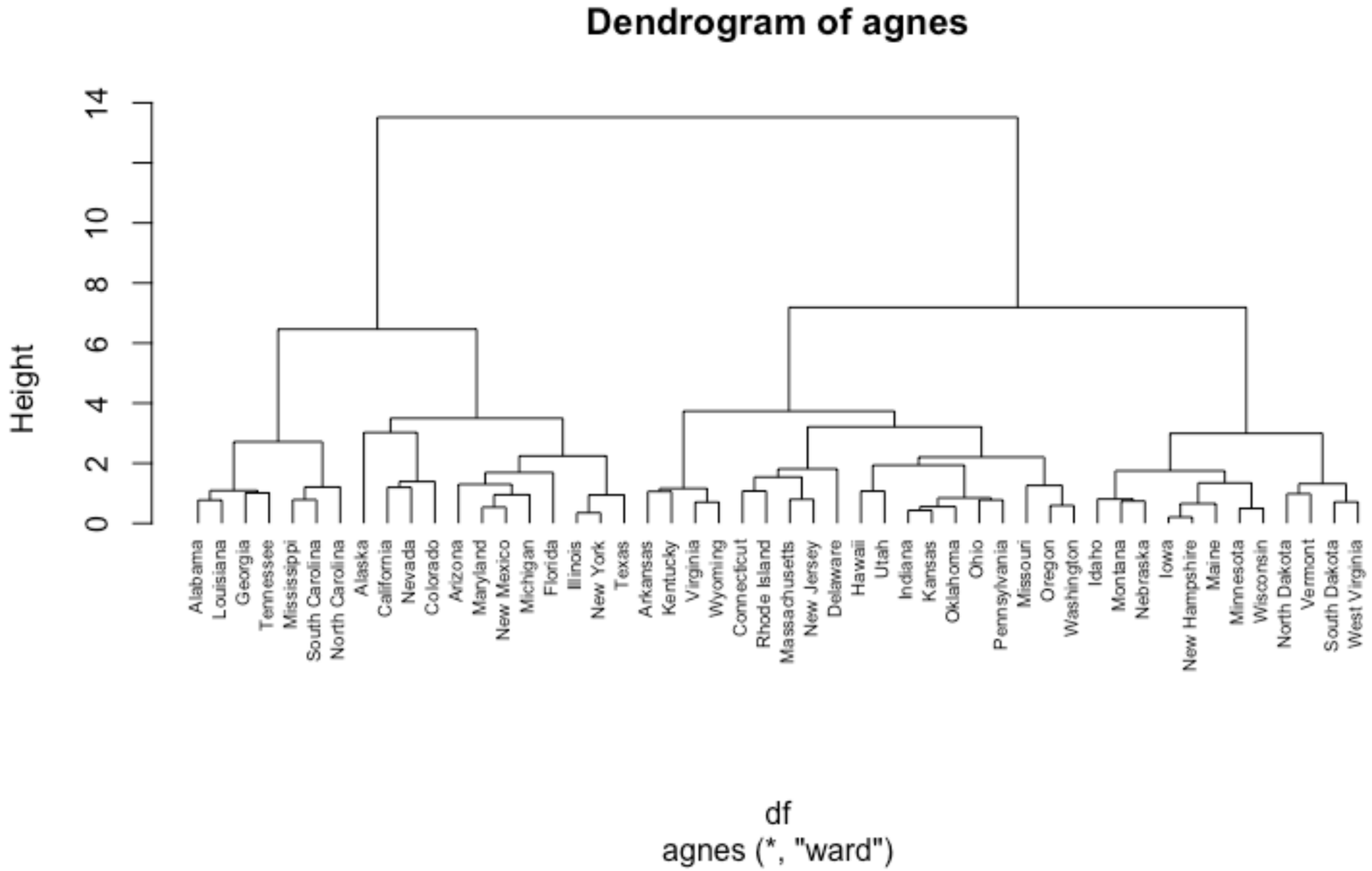


Other type of clustering

- Hierarchical clustering
- Partitioning clustering
- Density-based clustering
- Grid-based clustering
- Model-based clustering

Hierarchical clustering

- Alternative approach to k-means clustering for identifying groups in the dataset.
- It does not require us to pre-specify the number of clusters
- tree-based representation of the observations, called a dendrogram.



Limitations of cluster analysis

- It can be difficult to interpret the results of an ambiguous or ill-defined cluster
- The result of the analysis is affected by the choice of the clustering algorithm
- The success of cluster analysis depends on the data, the goal of the analysis, and the data scientist’s capability to interpret the result