Tut_6

Machine Learning — Clustering

Datensatz erstellen

```
x <- c(2, 2, 8, 5, 7, 6, 1, 4)
y <- c(10, 5, 4, 8, 5, 4, 2, 9)

data <- matrix(
  data = c(x, y),
  nrow = 8,
  ncol = 2,
  dimnames = list(paste0("A", 1:8), c("x", "y")) # paste0 -> paste mit sep = ""
)

data
```

```
   x  y
A1 2 10
A2 2  5
A3 8  4
A4 5  8
A5 7  5
A6 6  4
A7 1  2
A8 4  9
```

Aufgabe 1

Benutzen Sie die Funktion dist() mit dem Argument method = “euclidean”, um die euklidische Distanz zwischen den Punkten zu berechnen. Speichern Sie das Ergebnis in einer Variablen

dist_matrix. Dies ist nicht Teil des eigentlichen K-Means-Algorithmus, sondern dient nur der Visualisierung:

```
dist_matrix <- as.matrix(dist(data, method = "euclidean"))

dist_matrix
```

```
         A1       A2       A3       A4       A5       A6       A7       A8
A1 0.000000 5.000000 8.485281 3.605551 7.071068 7.211103 8.062258 2.236068
A2 5.000000 0.000000 6.082763 4.242641 5.000000 4.123106 3.162278 4.472136
A3 8.485281 6.082763 0.000000 5.000000 1.414214 2.000000 7.280110 6.403124
A4 3.605551 4.242641 5.000000 0.000000 3.605551 4.123106 7.211103 1.414214
A5 7.071068 5.000000 1.414214 3.605551 0.000000 1.414214 6.708204 5.000000
A6 7.211103 4.123106 2.000000 4.123106 1.414214 0.000000 5.385165 5.385165
A7 8.062258 3.162278 7.280110 7.211103 6.708204 5.385165 0.000000 7.615773
A8 2.236068 4.472136 6.403124 1.414214 5.000000 5.385165 7.615773 0.000000
```

Aufgabe 2

Wählen sie zwei beliebige Punkte als Startpunkte für die Clusterzentren. Speichern Sie diese in einer Variablen seeds. Zum Beispiel so:

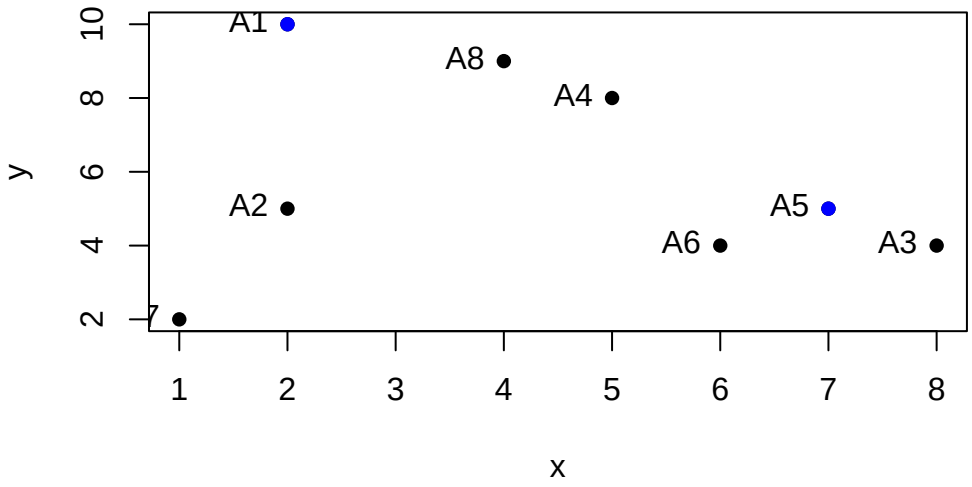
```
seeds <- data[c("A1", "A5"), ]

seeds
```

```
   x  y
A1 2 10
A5 7  5
```

Aufgabe 3

Plotten Sie die Punkte und die Startpunkte. Benutzen Sie ggplot2 oder die Base R Funktionen plot() und text(), um die Punkte zu beschriften und points() um die Startpunkte zu plotten.

```
plot(data, pch = 16)

points(seeds, pch = 16, col = "blue")

text(x = data[,1], y = data[,2], labels = rownames(data), pos = 2)
```



Aufgabe 4

Schreiben Sie eine Funktion euclidean_distance(). Die Funktion soll die euklidische Distanz zwischen zwei Punkten berechnen. Die Funktion soll zwei Argumente point und seed haben und die Wurzel der Summe der quadrierten Differenzen der Koordinaten der Punkte zurückgeben.

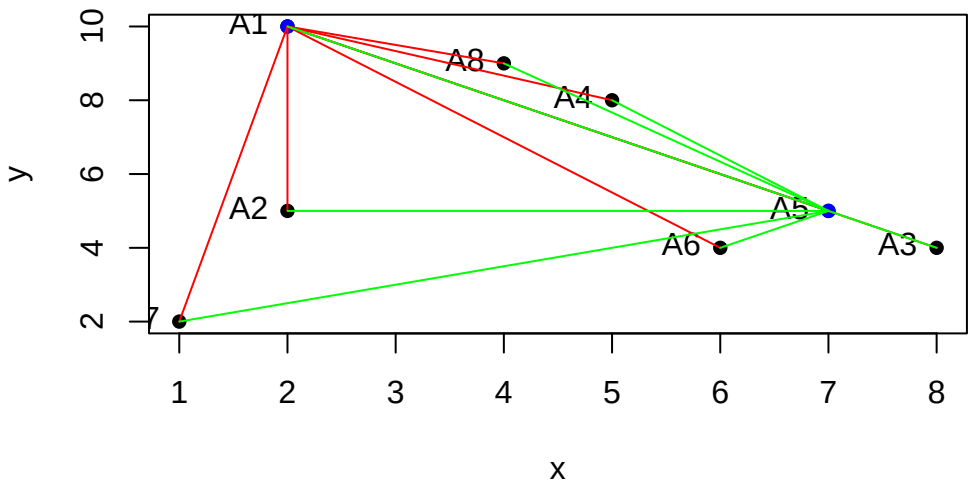
$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

```
e_dist <- function(p, q) {
  sqrt(sum((p - q)^2))
}
```

Aufgabe 5

Zeichnen Sie Linien von jedem Punkt zu den Startpunkten: Benutzen Sie dazu die Base R Funktion lines() oder in ggplot2 die Funktion geom_segment().

```r
plot(data, pch = 16)

points(seeds, pch = 16, col = "blue")

text(x = data[,1], y = data[,2], labels = rownames(data), pos = 2)


line_colors <- c("red", "green")

for (i in 1:nrow(seeds)) {
  for (j in 1:nrow(data)) {
    lines(x = c(seeds[i, 1], data[j, 1]),
          y = c(seeds[i, 2], data[j, 2]),
          col = line_colors[i])
  }
}
```



Aufgabe 6

Berechnen Sie die Distanzen von jedem Punkt zu den Startpunkten. Speichern Sie die Distanzen in einer Matrix dist_to_seeds unter Verwendung von zwei verschachtelten for-Schleifen.

```r
dist_to_seeds <- matrix(NA, nrow = nrow(data), ncol = nrow(seeds),
                        dimnames = list(paste0("A", 1:8), c("A1", "A5")))
for (i in 1:nrow(data)) {
  for (j in 1:nrow(seeds)) {
    dist_to_seeds[i, j] <- e_dist(data[i, 1:2], seeds[j, 1:2])
  }
}

dist_to_seeds # Prüfen der Ergebnisse mit dist_matrix
```

```
         A1       A5
A1 0.000000 7.071068
A2 5.000000 5.000000
A3 8.485281 1.414214
A4 3.605551 3.605551
A5 7.071068 0.000000
A6 7.211103 1.414214
A7 8.062258 6.708204
A8 2.236068 5.000000
```

Aufgabe 7

Weisen Sie jedem Punkt das Clusterzentrum zu, das am nächsten ist. Speichern Sie die Zuweisungen in einem Vektor cluster_assignments. Benutzen Sie dazu die Funktion apply().

```r
cluster_assignments <- apply(dist_to_seeds, 1, which.min)

cluster_assignments
```

```
A1 A2 A3 A4 A5 A6 A7 A8
 1  1  2  1  2  2  2  1
```

Aufgabe 8

Berechnen Sie die neuen Clusterzentren. Speichern Sie die neuen Clusterzentren in einer Matrix new_centroids. Benutzen Sie dazu eine for-Schleife.

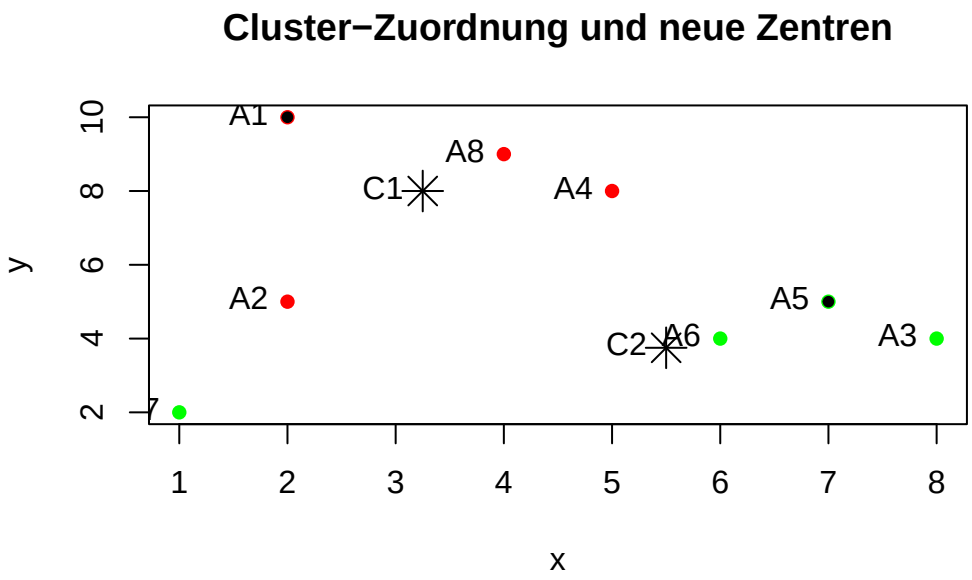
```r
k <- nrow(seeds)
new_centroids <- matrix(NA, nrow = k, ncol = 2,
                        dimnames = list(c("1", "2"), c("x", "y")))

for (i in 1:k) {
  cluster_points <- data[cluster_assignments == i, 1:2]

  new_centroids[i, ] <- colMeans(cluster_points)
}

new_centroids
```

```
     x    y
1 3.25 8.00
2 5.50 3.75
```

Aufgabe 9

Nutzen Sie ggplot2 oder Base R zum Plotten der Ergebnisse.

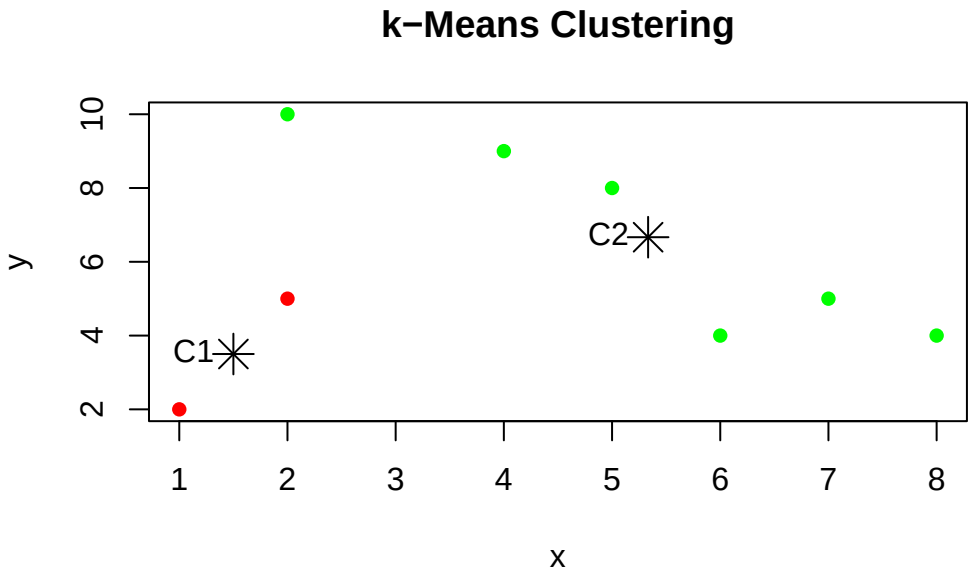
```r
cluster_colors <- c("red", "green")

plot(data, pch = 16, col = cluster_colors[cluster_assignments],
     xlab = "x", ylab = "y", main = "Cluster-Zuordnung und neue Zentren")

points(seeds, pch = 20)

text(x = data[,1], y = data[,2], labels = rownames(data), pos = 2)

points(new_centroids, pch = 8, cex = 2)

text(x = new_centroids[, "x"], y = new_centroids[, "y"],
     labels = paste0("C", 1:nrow(new_centroids)), pos = 2)
```



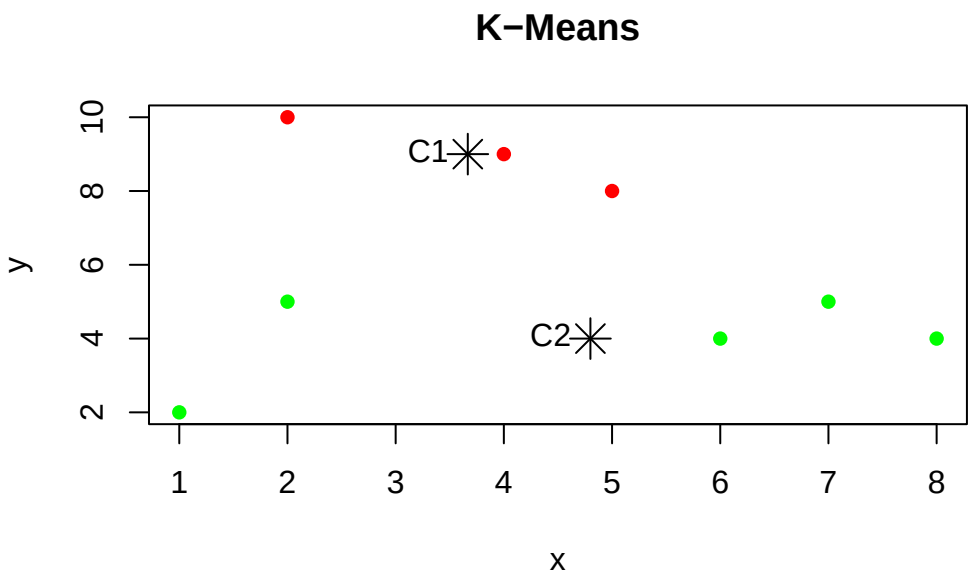
Die letzten beiden Schritte werden Wiederholt, bis keine Veränderungen mehr auftreten.

K-Means in R

```r
# K = 2
set.seed(95)

kmeans <- kmeans(data[, 1:2], centers = 2)

cluster_colors <- c("red", "green")

plot(data[, 1:2], col = cluster_colors[kmeans$cluster],
     pch = 16, main = "k-Means Clustering")

points(kmeans$centers, pch = 8, cex = 2)

text(kmeans$centers, labels = paste0("C", 1:2), pos = 2)
```

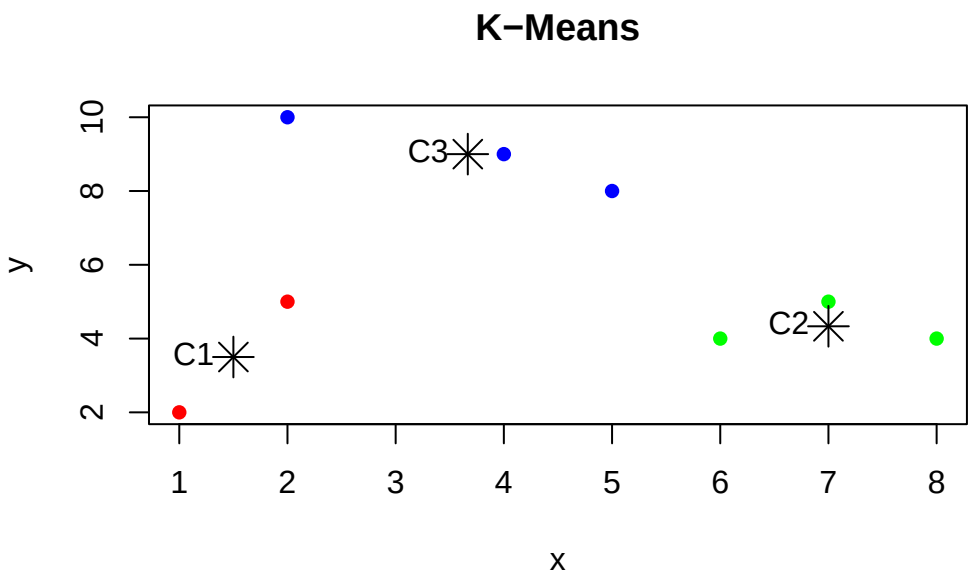



```r
set.seed(1)

kmeans <- kmeans(data[, 1:2], centers = 2)

cluster_colors <- c("red", "green")

plot(data[, 1:2], col = cluster_colors[kmeans$cluster],
     pch = 16, main = "K-Means")

points(kmeans$centers, pch = 8, cex = 2)

text(kmeans$centers, labels = paste0("C", 1:2), pos = 2)
```



Der K-Means Algorithmus verwendet zufällige Startpunkte. Das beeifnlusst die Clusterentwicklung. Daher erhält man nicht immer das Gleiche Ergebnis.

```
# K = 3
set.seed(95)

kmeans <- kmeans(data[, 1:2], centers = 3)

cluster_colors <- c("red", "green", "blue")

plot(data[, 1:2], col = cluster_colors[kmeans$cluster],
     pch = 16, main = "K-Means")

points(kmeans$centers, pch = 8, cex = 2)

text(kmeans$centers, labels = paste0("C", 1:3), pos = 2)
```


K−Means
y
x
C1
C2
C3
2
4
6
8
10
1
2
3
4
5
6
7
8