K-Means Problem

Jack Reddan

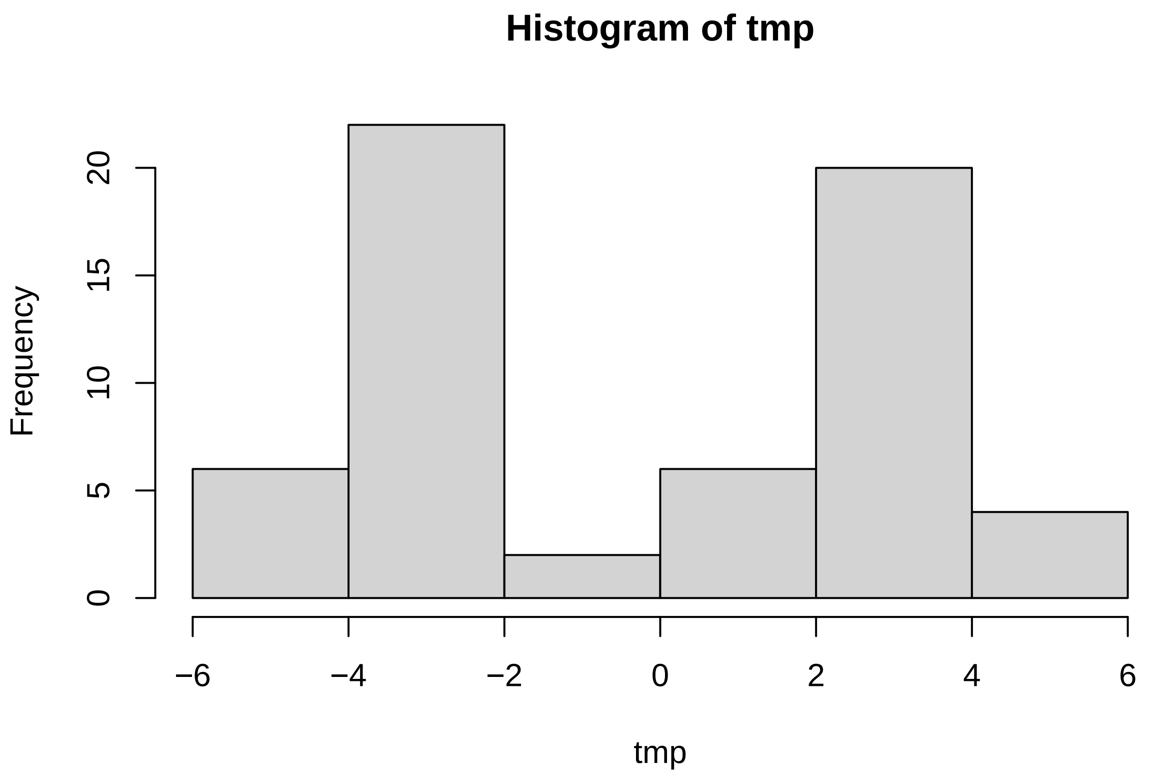
10/21/2021

Try K-Means Clustering

Generate fake data and explore how the method works.

Generate example data

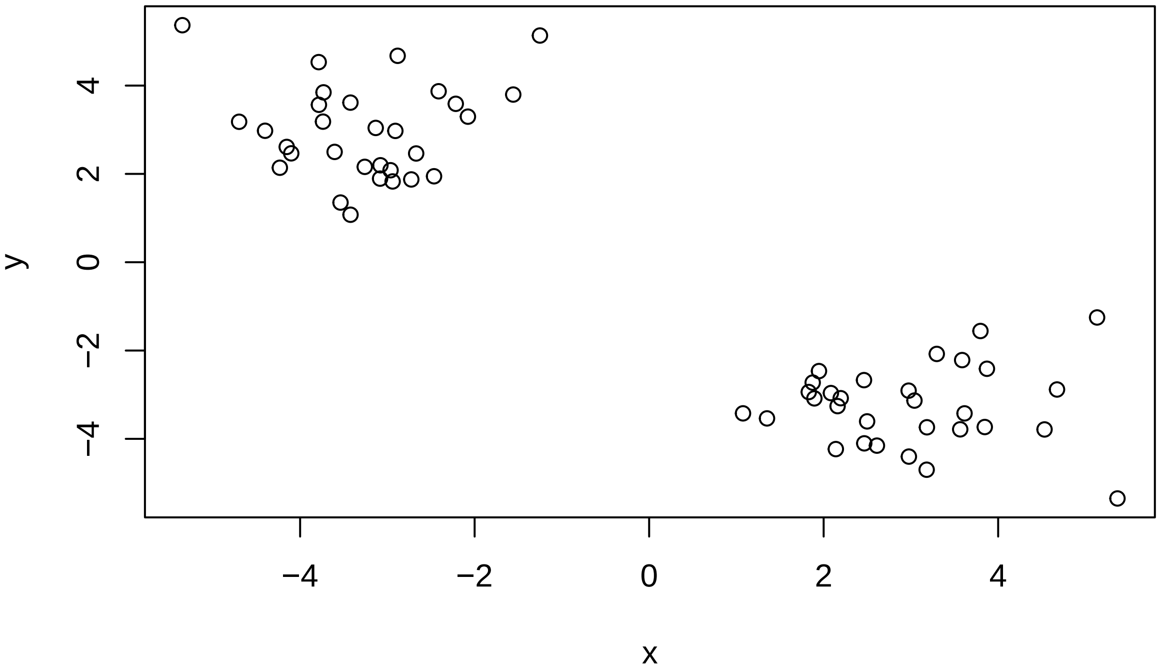
```
tmp <- c(rnorm(30,-3), rnorm(30,3))
hist(tmp)
```



Generate multidimensional example data

```
x <- cbind(x = tmp, y = rev(tmp))

plot(x)
```



Use the kmeans() function to explore the fake data

Use it while specifying 2 expected clusters and iterating 20 times.

```
clusters <- kmeans(x, centers = 2, nstart = 20)

clusters
```

```
## K-means clustering with 2 clusters of sizes 30, 30
##
## Cluster means:
##           x         y
## 1 -3.253022  2.974293
## 2  2.974293 -3.253022
##
## Clustering vector:
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
## [39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 57.29033 57.29033
##  (between_SS / total_SS =  91.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

[Q] How many points are in each cluster?

There are 30 points in each cluster.

```
clusters$size
```

```
## [1] 30 30
```

[Q] What component of your results object dteails:

Cluster size

```
clusters$size
```

```
## [1] 30 30
```

Cluster assignment

```
clusters$cluster
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
## [39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Cluster center

```
clusters$centers
```

```
##           x         y
## 1 -3.253022  2.974293
## 2  2.974293 -3.253022
```

Plot x colored by the kmeans cluster centers as blue points

Load ggplot2

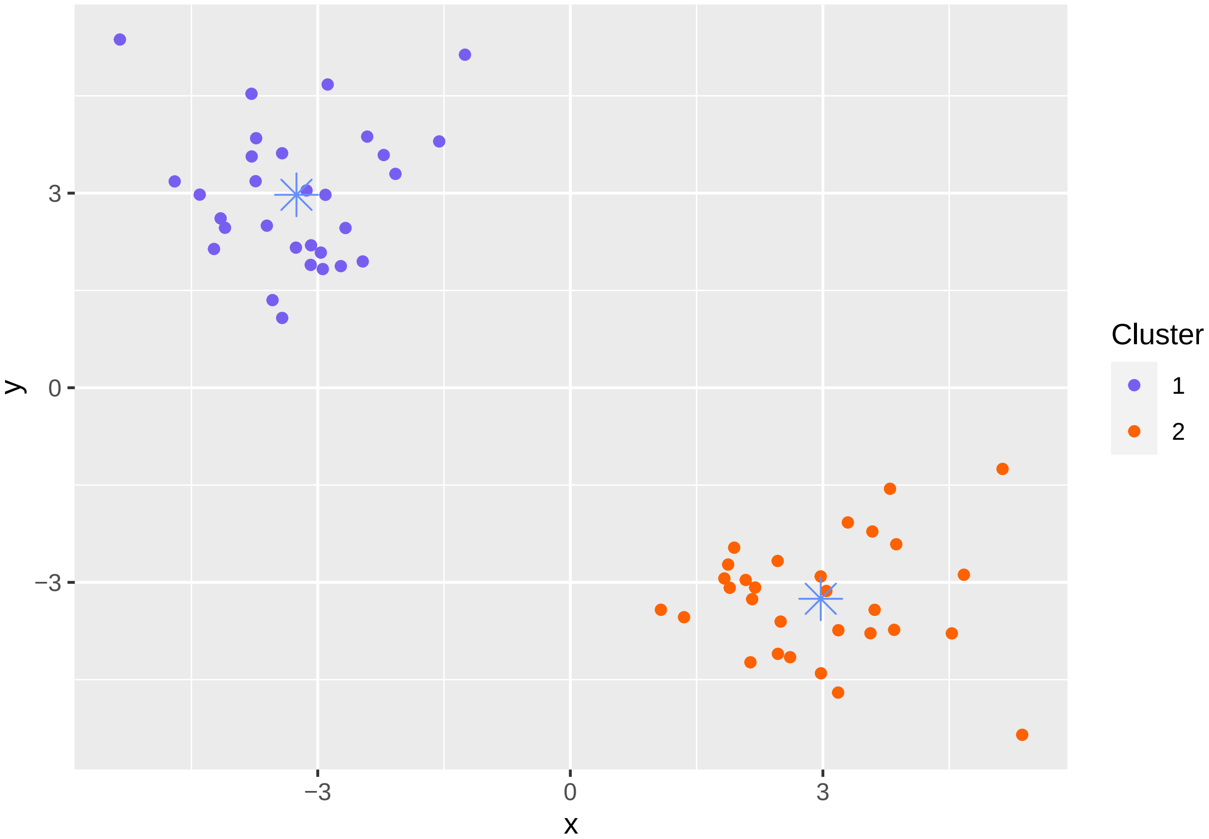
```
library(ggplot2)
```

Convert matrices to be used in ggplot to data frames.

```
df <- data.frame(x)
centroids <- data.frame(clusters$centers)
```

Plot the original data colored by kmenas clusters and add blue centroids. IBM’s colorblind palette is used.

```
ggplot(data = df) +
  aes(x = x, y = y, color = factor(clusters$cluster)) +
  geom_point() +
  scale_color_manual(values = c("#785EF0", "#FE6100"), name = "Cluster") +
  geom_point(data = centroids, aes(x = x, y = y), color = "#648FFF", shape = 8, size = 5)
```



#

Try Hierarchical Clustering

Using the same example data *'x'*.

Generate the distance matrix

```
dm <- dist(x)

str(dm)
```

```
##  'dist' num [1:1770] 0.498 1.724 3.358 0.935 0.723 ...
##  - attr(*, "Size")= int 60
##  - attr(*, "Diag")= logi FALSE
##  - attr(*, "Upper")= logi FALSE
##  - attr(*, "method")= chr "euclidean"
##  - attr(*, "call")= language dist(x = x)
```

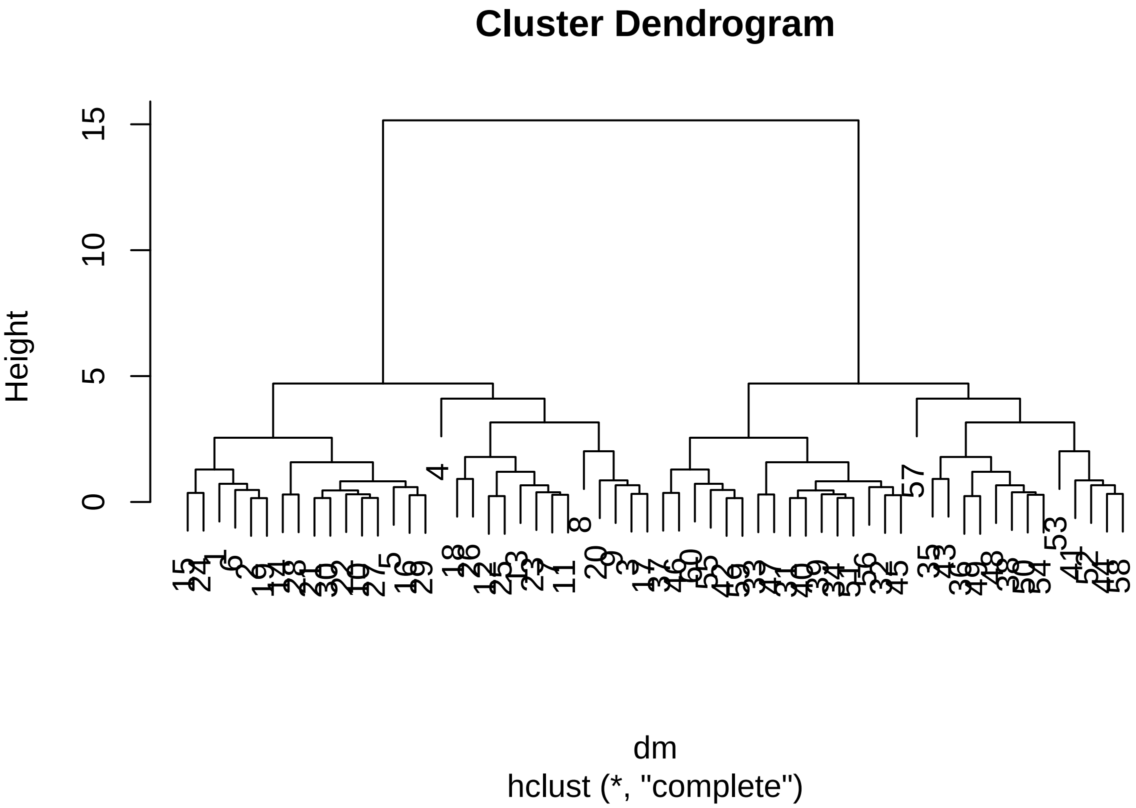
Call hclust() to determine clusters

```
hc <- hclust(dm)
hc
```

```
##
## Call:
## hclust(d = dm)
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 60
```

Plot the hierachical cluster

```
plot(hc)
```



## ‘*Cut*’ the dendrogram to assign membership of ‘*leaves*’ to clusters Can either specify the height (h) or desired number of clusters (k).

```
trimmed_hc <- cutree(hc, h = 6)
trimmed_hc
```

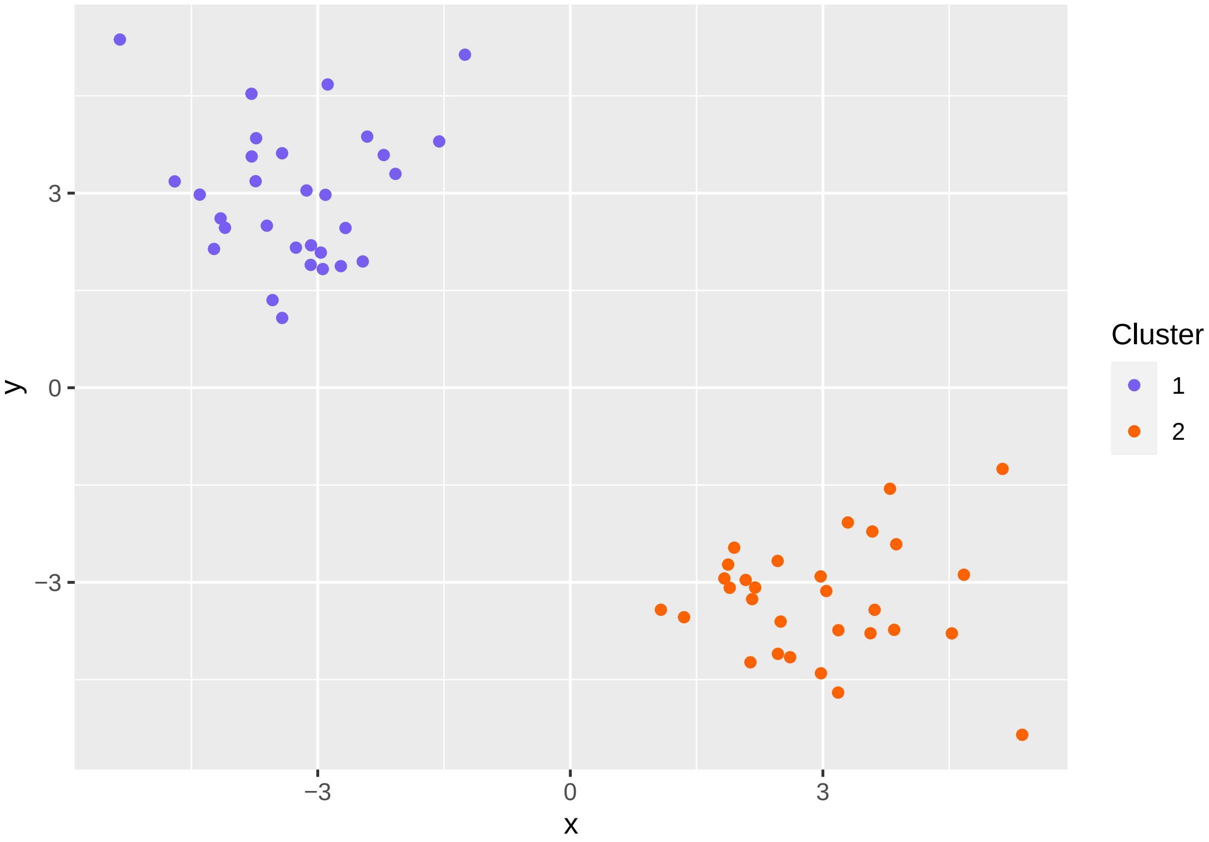
```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
## [39] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
trimmed_hc_v2 <- cutree(hc, k = 2)
trimmed_hc == trimmed_hc_v2
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Plot x data colored by the hierarchical cluster membership

```
ggplot(data = df) +
  aes(x = x, y = y, color = factor(trimmed_hc)) +
  geom_point() +
  scale_color_manual(values = c("#785EF0", "#FE6100"), name = "Cluster")
```


3
0
−3
y
−3
0
3
x
Cluster
1
2