Bios 611 HW

Li Jiang

10/20/2024

Q1

```
my_kmeans <- function(data, k) {

  cluster_labels <- sample(1:k, nrow(data), replace = T)

  data_mat <- data %>% as.matrix()
  centroids <- c()
    for(i in 1:k) {
    submat <- data_mat[i == cluster_labels, ]
    centroid <- colSums(submat) / nrow(submat)
    centroids <- rbind(centroids, centroid)
    }

  while (T) {
  old_centroids<-centroids
  for(i in 1:nrow(data_mat)) {
    row <- data_mat[i, ]
    min_d <- Inf
    min_j <- 0
    for(j in 1:k) {
      d <- sqrt(sum((row - centroids[j, ]) * (row - centroids[j, ])))
      if(d < min_d) {
        min_d <- d
        min_j <- j
      }
    }
    cluster_labels[i]<- min_j
  }
  centroids <- c()
  for(i in 1:k) {
    submat <- data_mat[i == cluster_labels, ]
    centroid <- colSums(submat) / nrow(submat)
    centroids <- rbind(centroids, centroid)
  }
  centroid_diff <- old_centroids-centroids
  centroid_distance <- mean(rowSums(centroid_diff*centroid_diff))
  if(centroid_distance<1e-6){
    break
  }
```

```
  }

  return(cluster_labels)
}
```

Q2

```
library(tibble)
set.seed(123)
data_1<-tibble(
  X1 = rnorm(100,5,1),
  X2 = rnorm(100,0,1),
  X3 = rnorm(100,0,1),
  X4 = rnorm(100,0,1),
  X5 = rnorm(100,0,1),
)
data_2 <- tibble(
  X1 = rnorm(100, -5, 1),
  X2 = rnorm(100, 0, 1),
  X3 = rnorm(100, 0, 1),
  X4 = rnorm(100, 0, 1),
  X5 = rnorm(100, 0, 1)
)

data_3 <- tibble(
  X1 = rnorm(100, 0, 1),
  X2 = rnorm(100, 0, 1),
  X3 = rnorm(100, 0, 1),
  X4 = rnorm(100, 3, 1),
  X5 = rnorm(100, 0, 1)
)

data_4 <- tibble(
  X1 = rnorm(100, 0, 1),
  X2 = rnorm(100, 0, 1),
  X3 = rnorm(100, 0, 1),
  X4 = rnorm(100, -2, 1),
  X5 = rnorm(100, 0, 1)
)

data_5 <- tibble(
  X1 = rnorm(100, 4, 1),
  X2 = rnorm(100, 0, 1),
  X3 = rnorm(100, 0, 1),
  X4 = rnorm(100, -3, 1),
  X5 = rnorm(100, 0, 1)
)
data<-rbind(data_1, data_2, data_3, data_4, data_5)
```

Q3

```
my_kmeans(data,5)
```

```
##   [1] 5 5 5 5 5 5 1 5 5 5 5 5 5 5 5 5 1 5 5 5 5 1 5 5 5 5 5 5 5 5 5 5 5 1 5
##  [38] 5 5 5 5 5 5 5 5 1 5 5 5 5 5 1 5 5 5 5 5 5 1 5 5 5 5 5 5 5 5 5 5 5 1 5 5
##  [75] 5 5 5 5 5 1 5 5 5 5 5 5 5 5 5 5 5 1 5 5 5 5 5 5 5 5 2 2 2 2 2 2 2 2 2 2
## [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [149] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [186] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [223] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [260] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [297] 3 3 3 3 4 4 4 4 4 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [334] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [371] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 4
## [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [445] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 1 1 1 5 1 4 1 1 1 1 1 1 1 1 4 1 1 1
## [482] 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

This makes sense because most of the rows were clustered as we wanted.

Q4

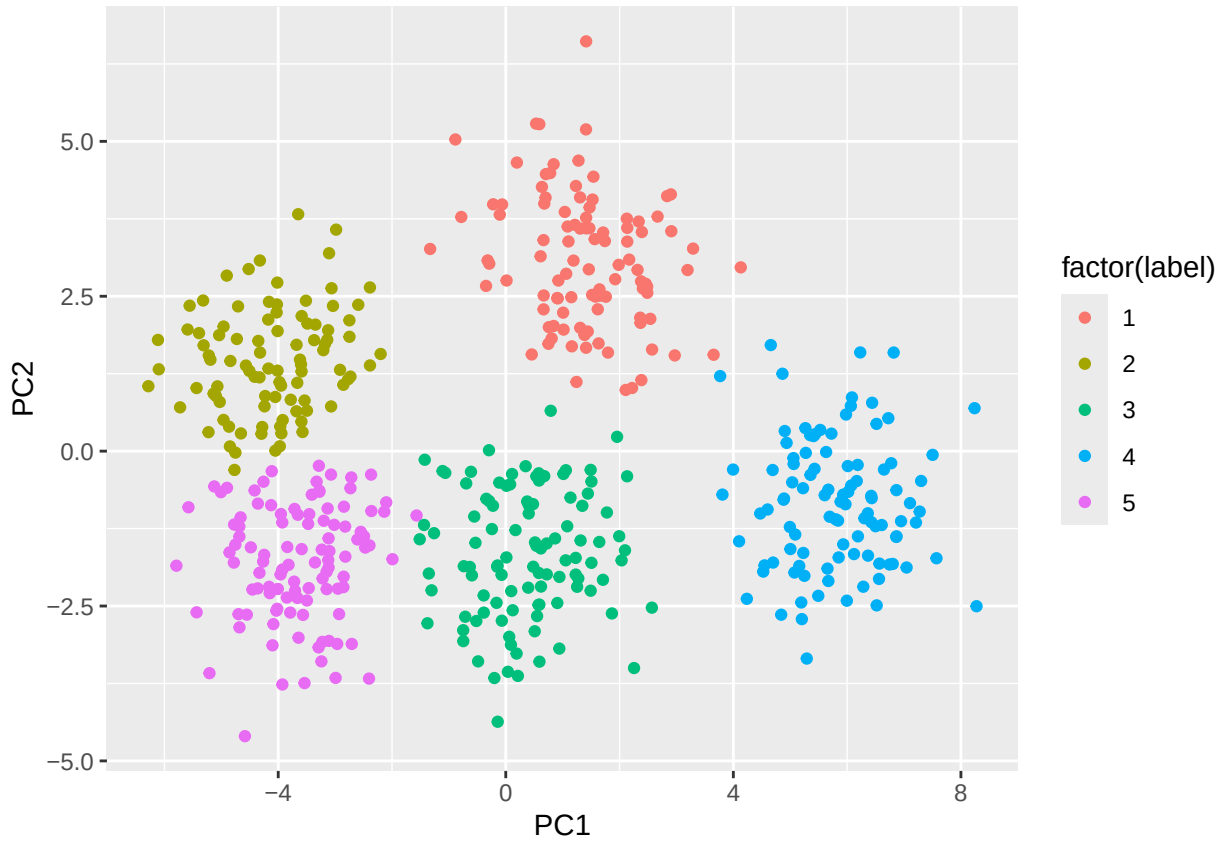
```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.0.2
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.5.1     v stringr   1.5.1
## v lubridate 1.9.3     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
result <- prcomp(data %>% as.matrix())
pcd <- result$x %>% as.tibble() %>% mutate(label=my_kmeans(data,5)) %>% select(PC1,PC2,label)
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` instead.
## i The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
ggplot(pcd, aes(PC1,PC2))+
  geom_point(aes(color=factor(label)))
```



Q5

```
library(cluster)
generate_sys<-function(r){
  data_1<-tibble(
  X1 = rnorm(100,5*r,1),
  X2 = rnorm(100,0,1),
  X3 = rnorm(100,0,1),
  X4 = rnorm(100,0,1),
  X5 = rnorm(100,0,1),
)
data_2 <- tibble(
  X1 = rnorm(100, -5*r, 1),
  X2 = rnorm(100, 0, 1),
  X3 = rnorm(100, 0, 1),
  X4 = rnorm(100, 0, 1),
  X5 = rnorm(100, 0, 1)
)

data_3 <- tibble(
  X1 = rnorm(100, 0, 1),
  X2 = rnorm(100, 0, 1),
  X3 = rnorm(100, 0, 1),
  X4 = rnorm(100, 3*r, 1),
  X5 = rnorm(100, 0, 1)
)
```

```
data_4 <- tibble(
  X1 = rnorm(100, 0, 1),
  X2 = rnorm(100, 0, 1),
  X3 = rnorm(100, 0, 1),
  X4 = rnorm(100, -2*r, 1),
  X5 = rnorm(100, 0, 1)
)

data_5 <- tibble(
  X1 = rnorm(100, 4*r, 1),
  X2 = rnorm(100, 0, 1),
  X3 = rnorm(100, 0, 1),
  X4 = rnorm(100, -3*r, 1),
  X5 = rnorm(100, 0, 1)
)
data<-rbind(data_1, data_2, data_3, data_4, data_5)
return(data)
}

rs<-c()
ncs<-c()
for (r in seq(0, 2, length.out = 4)){
  dataset<-generate_sys(r)
  result<-clusGap(dataset, kmeans, 10)
  nc <- maxSE(f = result$Tab[,"gap"],
              SE.f = result$Tab[,"SE.sim"],
              method='Tibs2001SEmax',
              SE.factor = 1)
  rs <- c(rs,r)
  ncs <- c(ncs,nc)
}
ncs_df <- tibble(r=rs,nc=ncs)
ncs_df
```

```
## # A tibble: 4 x 2
##       r    nc
##   <dbl> <int>
## 1 0         1
## 2 0.667     2
## 3 1.33      5
## 4 2         4
```

This is what we expected, as r heads towards zero we get an estimate of one cluster and as r gets larger we get an estimate of five clusters as we desired.