WWD\_3

zhou ziliang

2019/11/12

## load the packeage  
 library(cclust)  
 library(tidyverse)

## -- Attaching packages ----------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## √ ggplot2 3.2.1 √ purrr 0.3.2  
## √ tibble 2.1.3 √ dplyr 0.8.3  
## √ tidyr 1.0.0 √ stringr 1.4.0  
## √ readr 1.3.1 √ forcats 0.4.0

## -- Conflicts -------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

## read file  
 animals <- read.csv("Animals.csv")  
## rename the animals  
 animals <-rename(animals,Weight = 'x',Height = 'y',Species = 'z')   
 animals$Species <- factor(animals$Species, levels = c("a","b","c","d"),  
 labels = c("Ostrich","Deer","Bear","Gaint tortise"))

# Question 1

d <- cbind(animals["Weight"],animals["Height"]) %>%   
 dist()  
 hd <- hclust(d)  
 group.4 <- cutree(hd,4)  
 table(animals$Species,group.4)

## group.4  
## 1 2 3 4  
## Ostrich 541 383 7 0  
## Deer 210 434 72 0  
## Bear 32 478 263 38  
## Gaint tortise 0 81 262 225

the method cannot separate the animals very well. For example, the most of Deer and Bear are separated in group 2

# function

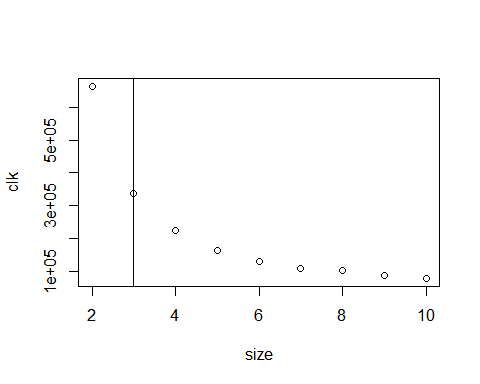
curv <- function(x,y){  
 ta <- sqrt((x[2]-x[1])^2+(y[2]-y[1])^2)  
 tb <- sqrt((x[3]-x[2])^2+(y[3]-y[2])^2)  
 M\_matrix <- matrix(c(1,ta,ta^2,1,0,0,1,tb,tb^2),3,3,byrow=TRUE)  
 M\_inverse <- solve(M\_matrix)  
 a <- M\_inverse %\*% x  
 b <- M\_inverse %\*% y  
 curvature <- 2\*(a[3]\*b[2]-a[2]\*b[3])/sqrt(a[2]^2+b[2]^2)  
 return(curvature)  
 }  
 max.curv <- function(x,y) {  
 I <- c(2:(length(x)-1))  
 cur <- I  
 for (i in I){  
 s.x <- c(x[i-1],x[i],x[i+1])  
 s.y <- c(y[i-1],y[i],y[i+1])  
 cur[i] <- curv(s.x,s.y)  
 }  
 cur <- abs(cur)  
 max <- max(cur)  
 n <- which.max(cur)+1  
 result <- list(max,n,cur)   
 names(result) <-c("max value","number","curature")  
 return(result)  
 }

# Question 2

animals\_df <- cbind(animals["Weight"],animals["Height"])  
  
 tot.withinss <- function(number,df){  
 c1 <- kmeans(df,number)  
 return(c1$tot.withinss)  
 }  
 size = c(2:10)  
 clk <- sapply(size,tot.withinss,df = animals\_df)  
   
 # find the max curvature  
 max.curv(size,clk)

## $`max value`  
## [1] 2  
##   
## $number  
## [1] 2  
##   
## $curature  
## [1] 2.000000e+00 2.661975e-11 8.840354e-11 2.866554e-10 6.900367e-10  
## [6] 5.066767e-09 4.781492e-09 5.907937e-09

# plot the clk against the size, and plot the vertical line  
 plot(size,clk)  
 abline(v = 3)



kc <- kmeans(animals\_df,3)  
 table(animals$Species,kc$cluster)

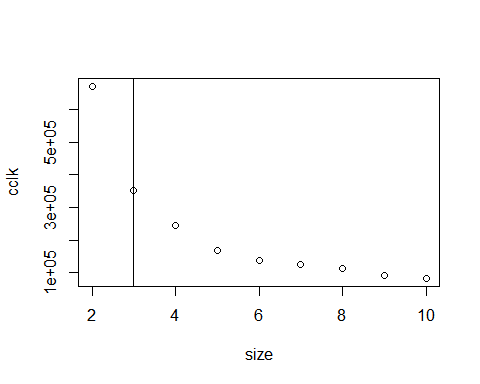
##   
## 1 2 3  
## Ostrich 5 652 274  
## Deer 36 302 378  
## Bear 292 54 465  
## Gaint tortise 488 0 80

# Question 3

ctot.withinss <- function(number,df,dist="manhattan"){  
 df <- as.matrix(df)  
 c2 <- cclust(df,number,dist = dist)  
 return(sum(c2$withinss))  
 }  
 cclk <- sapply(size,ctot.withinss,df = animals\_df)  
 max.curv(size,cclk)

## $`max value`  
## [1] 2  
##   
## $number  
## [1] 2  
##   
## $curature  
## [1] 2.000000e+00 2.886878e-11 4.134058e-11 3.276273e-10 2.609353e-09  
## [6] 5.097344e-10 2.104991e-09 4.829793e-09

plot(size,cclk)  
 abline(v=3)



ckc <- animals\_df %>% as.matrix() %>%   
 cclust(3,dist = "manhattan")  
 table(animals$Species,ckc$cluster)

##   
## 1 2 3  
## Ostrich 7 283 641  
## Deer 61 368 287  
## Bear 339 422 50  
## Gaint tortise 516 52 0