Assignment_4 FML

MICHAEL BALLAMUDI

2023-11-13

#Applying the knit functions

#Reading the data set and loading the data and transforming data.

```
Pharmaceuticals <- read.csv("C:/Users/micha/OneDrive/Desktop/SEM_1/FML/Pharmaceuticals.csv")
View(Pharmaceuticals)
```

#loading the required package

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## Warning: package 'readr' was built under R version 4.3.2
```

```
## Warning: package 'forcats' was built under R version 4.3.2
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

#loading the required package

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.3.2
```

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

#loading the required package

```
library(ISLR)

library(flexclust)
```

```
## Warning: package 'flexclust' was built under R version 4.3.2
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

#loading the required package

```
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

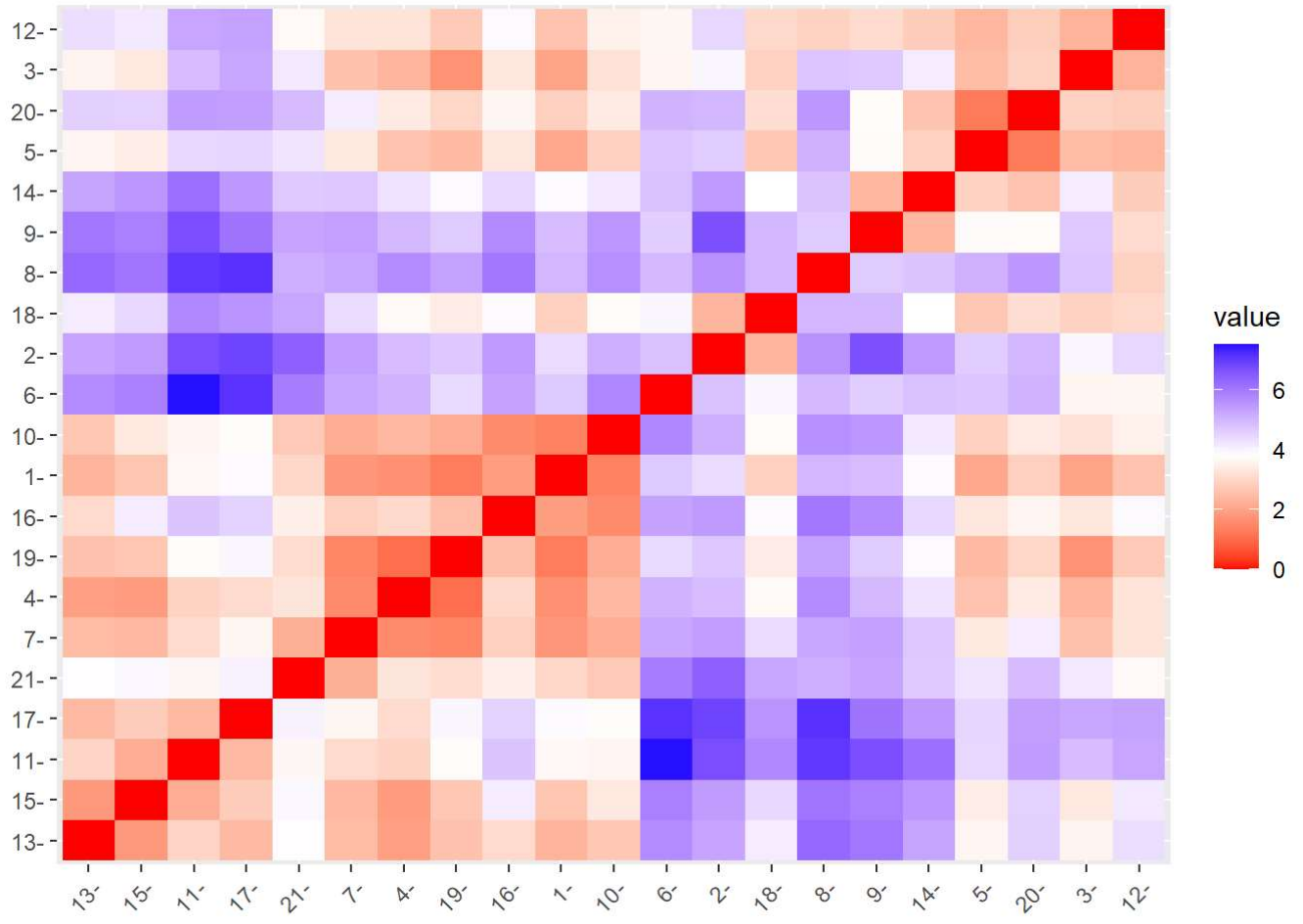
```
Scaled_pharma_data <- scale(Pharmaceuticals[,3:11])
summary(Scaled_pharma_data)
```

```
##    Market_Cap           Beta            PE_Ratio            ROE
##  Min.   :-0.9768   Min.   :-1.3466   Min.   :-1.3404   Min.   :-1.4515
##  1st Qu.:-0.8763   1st Qu.:-0.6844   1st Qu.:-0.4023   1st Qu.:-0.7223
##  Median :-0.1614   Median :-0.2560   Median :-0.2429   Median :-0.2118
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2762   3rd Qu.: 0.4841   3rd Qu.: 0.1495   3rd Qu.: 0.3450
##  Max.   : 2.4200   Max.   : 2.2758   Max.   : 3.4971   Max.   : 2.4597
##       ROA          Asset_Turnover       Leverage          Rev_Growth
##  Min.   :-1.7128   Min.   :-1.8451   Min.   :-0.74966   Min.   :-1.4971
##  1st Qu.:-0.9047   1st Qu.:-0.4613   1st Qu.:-0.54487   1st Qu.:-0.6328
##  Median : 0.1289   Median :-0.4613   Median :-0.31449   Median :-0.3621
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.8430   3rd Qu.: 0.9225   3rd Qu.: 0.01828   3rd Qu.: 0.7693
##  Max.   : 1.8389   Max.   : 1.8451   Max.   : 3.74280   Max.   : 1.8862
##  Net_Profit_Margin
##  Min.   :-1.99560
##  1st Qu.:-0.68504
##  Median : 0.06168
##  Mean   : 0.00000
##  3rd Qu.: 0.82364
##  Max.   : 1.49416
```

```
#Performing range scaling for the the dataframe
range_pharma_data <- scale(Pharmaceuticals[,3:11])
#summarizing the scaled data frame
summary(range_pharma_data)
```

```
##    Market_Cap           Beta            PE_Ratio            ROE
##  Min.   :-0.9768   Min.   :-1.3466   Min.   :-1.3404   Min.   :-1.4515
##  1st Qu.:-0.8763   1st Qu.:-0.6844   1st Qu.:-0.4023   1st Qu.:-0.7223
##  Median :-0.1614   Median :-0.2560   Median :-0.2429   Median :-0.2118
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2762   3rd Qu.: 0.4841   3rd Qu.: 0.1495   3rd Qu.: 0.3450
##  Max.   : 2.4200   Max.   : 2.2758   Max.   : 3.4971   Max.   : 2.4597
##       ROA          Asset_Turnover       Leverage          Rev_Growth
##  Min.   :-1.7128   Min.   :-1.8451   Min.   :-0.74966   Min.   :-1.4971
##  1st Qu.:-0.9047   1st Qu.:-0.4613   1st Qu.:-0.54487   1st Qu.:-0.6328
##  Median : 0.1289   Median :-0.4613   Median :-0.31449   Median :-0.3621
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.8430   3rd Qu.: 0.9225   3rd Qu.: 0.01828   3rd Qu.: 0.7693
##  Max.   : 1.8389   Max.   : 1.8451   Max.   : 3.74280   Max.   : 1.8862
##  Net_Profit_Margin
##  Min.   :-1.99560
##  1st Qu.:-0.68504
##  Median : 0.06168
##  Mean   : 0.00000
##  3rd Qu.: 0.82364
##  Max.   : 1.49416
```

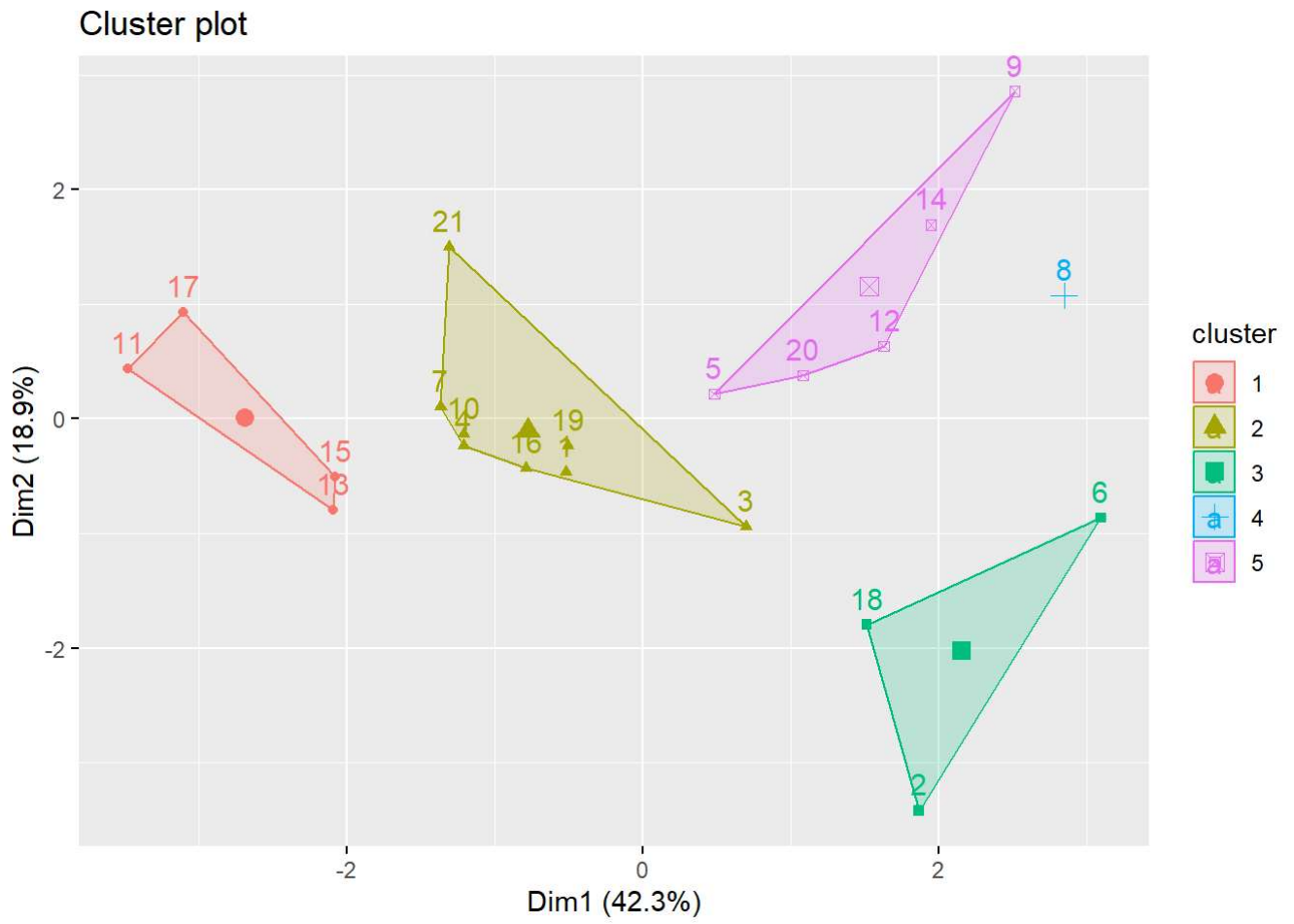
```
#calculating the distance of the scaled pharmaceuticals data
distance <- get_dist(Scaled_pharma_data)
fviz_dist(distance) #visualizing the distance between rows of the distance
```




```
matrix
```

```
## function (data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)
## {
##     if (is.object(data) || !is.atomic(data))
##         data <- as.vector(data)
##     .Internal(matrix(data, nrow, ncol, byrow, dimnames, missing(nrow),
##         missing(ncol)))
## }
## <bytecode: 0x000001c992360208>
## <environment: namespace:base>
```

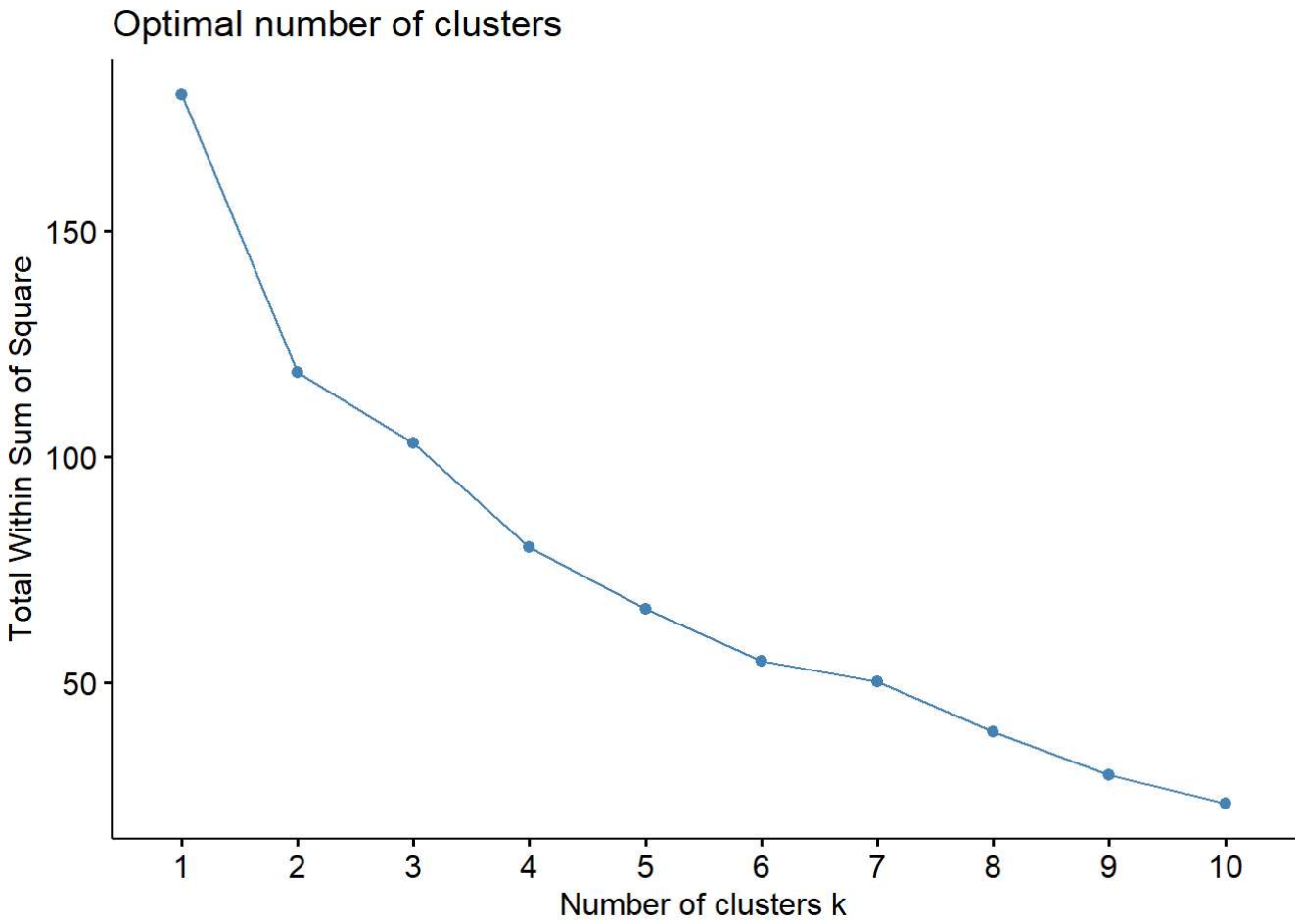
```
#applying K-means clustering for the scaled data
kmeans_1 <- kmeans(Scaled_pharma_data, centers = 5, nstart = 25)
#visualizing the clusters on agraph
fviz_cluster(kmeans_1, data = Scaled_pharma_data)
```




```
print(kmeans_1)
```

```
## K-means clustering with 5 clusters of sizes 4, 8, 3, 1, 5
##
## Cluster means:
##    Market_Cap       Beta    PE_Ratio        ROE        ROA Asset_Turnover
## 1  1.69558112 -0.1780563 -0.19845823  1.2349879  1.3503431   1.153164e+00
## 2 -0.03142211 -0.4360989 -0.31724852  0.1950459  0.4083915   1.729746e-01
## 3 -0.52462814  0.4451409  1.84984387 -1.0404550 -1.1865838   1.480297e-16
## 4 -0.97676686  1.2630872  0.03299122 -0.1123792 -1.1677918  -4.612656e-01
## 5 -0.79605926  0.3205014 -0.45014035 -0.6533148 -0.7881923  -1.107037e+00
##     Leverage Rev_Growth Net_Profit_Margin
## 1 -0.4680782  0.4671788         0.5912425
## 2 -0.2744931 -0.7041516         0.5569544
## 3 -0.3443544 -0.5769454        -1.6095439
## 4  3.7427970 -0.6327607        -1.2488842
## 5  0.2717048  1.2256188        -0.1486179
##
## Clustering vector:
##  [1] 2 3 2 2 5 3 2 4 5 2 1 5 1 5 1 2 1 3 2 5 2
##
## Within cluster sum of squares by cluster:
## [1]  9.284424 21.879320 14.938904  0.000000 16.542597
##  (between_SS / total_SS =  65.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

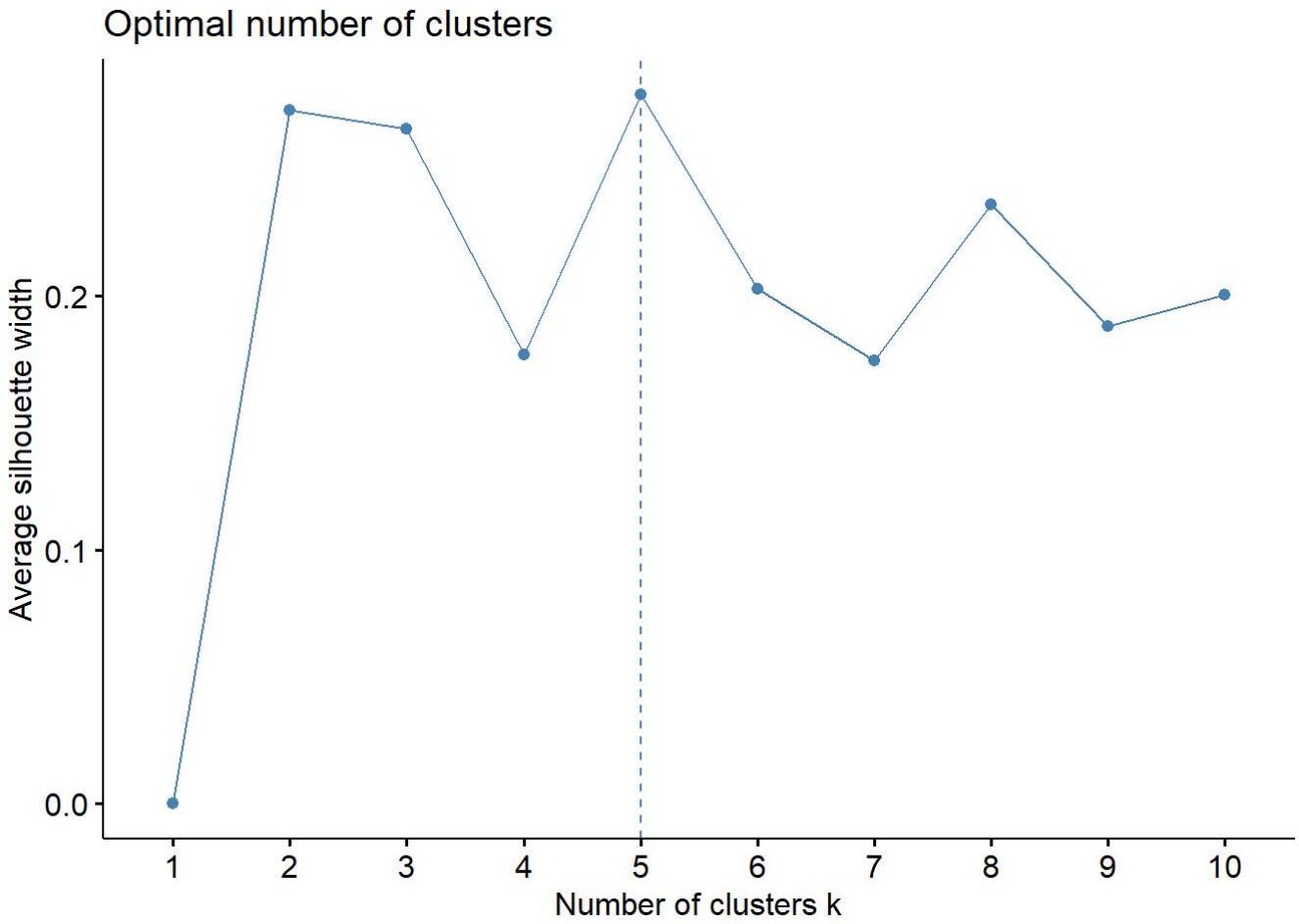
```
#plotting the number of clusters vs the total value
fviz_nbclust(Scaled_pharma_data, kmeans, method = "wss")
```




```
#figuring ou the number of optimal clusters by plotting the number of
clusters
```

```
## standardGeneric for "clusters" defined from package "modeltools"
##
## function (object, newdata, ...)
## standardGeneric("clusters")
## <bytecode: 0x000001c99581bcf0>
## <environment: 0x000001c99580f600>
## Methods may be defined for arguments: object, newdata
## Use  showMethods(clusters)  for currently available ones.
```

```
#against average silhoutte width
fviz_nbclust(Scaled_pharma_data, kmeans, method = "silhouette")
```




```
#Calculating the mean value from the actual data that is plotted in the
clusters
```

```
## standardGeneric for "clusters" defined from package "modeltools"
##
## function (object, newdata, ...)
## standardGeneric("clusters")
## <bytecode: 0x000001c99581bcf0>
## <environment: 0x000001c99580f600>
## Methods may be defined for arguments: object, newdata
## Use  showMethods(clusters)  for currently available ones.
```

```
#perfroming the aggregate function for the pharma data
aggregate(Pharmaceuticals[3:11], by=list(cluster=kmeans_1$cluster), mean)
```

```
##   cluster Market_Cap    Beta PE_Ratio     ROE     ROA Asset_Turnover  Leverage
## 1       1  157.01750 0.48000 22.22500 44.4250 17.7000         0.9500 0.2200000
## 2       2   55.81000 0.41375 20.28750 28.7375 12.6875         0.7375 0.3712500
## 3       3   26.90667 0.64000 55.63333 10.1000  4.2000         0.7000 0.3166667
## 4       4    0.41000 0.85000 26.00000 24.1000  4.3000         0.6000 3.5100000
## 5       5   11.00000 0.60800 18.12000 15.9400  6.3200         0.4600 0.7980000
##   Rev_Growth Net_Profit_Margin
## 1  18.532500         19.575000
## 2   5.591250         19.350000
## 3   6.996667          5.133333
## 4   6.380000          7.500000
## 5  26.912000         14.720000
```

```
#perfroming the merging of the data frames using cbind
temp_data <- cbind(Pharmaceuticals, cluster = kmeans_1$cluster)
tibble(temp_data)
```

```
## # A tibble: 21 × 15
##    Symbol Name     Market_Cap  Beta PE_Ratio   ROE   ROA Asset_Turnover Leverage
##    <chr>  <chr>         <dbl> <dbl>    <dbl> <dbl> <dbl>          <dbl>    <dbl>
##  1 ABT    Abbott …      68.4   0.32     24.7  26.4  11.8            0.7     0.42
##  2 AGN    Allerga…       7.58  0.41     82.5  12.9   5.5            0.9     0.6
##  3 AHM    Amersha…       6.3   0.46     20.7  14.9   7.8            0.9     0.27
##  4 AZN    AstraZe…      67.6   0.52     21.5  27.4  15.4            0.9     0
##  5 AVE    Aventis       47.2   0.32     20.1  21.8   7.5            0.6     0.34
##  6 BAY    Bayer AG      16.9   1.11     27.9   3.9   1.4            0.6     0
##  7 BMY    Bristol…      51.3   0.5      13.9  34.8  15.1            0.9     0.57
##  8 CHTT   Chattem…       0.41  0.85     26    24.1   4.3            0.6     3.51
##  9 ELN    Elan Co…       0.78  1.08      3.6  15.1   5.1            0.3     1.07
## 10 LLY    Eli Lil…      73.8   0.18     27.9  31    13.5            0.6     0.53
## # ℹ 11 more rows
## # ℹ 6 more variables: Rev_Growth <dbl>, Net_Profit_Margin <dbl>,
## #   Median_Recommendation <chr>, Location <chr>, Exchange <chr>, cluster <int>
```

```
#Summarizing the detailed beakdown by cluster
by(temp_data, factor(temp_data$cluster), summary)
```

```
## factor(temp_data$cluster): 1
##     Symbol              Name             Market_Cap         Beta
##  Length:4           Length:4           Min.   :122.1   Min.   :0.3500
##  Class :character   Class :character   1st Qu.:129.9   1st Qu.:0.4325
##  Mode  :character   Mode  :character   Median :153.2   Median :0.4600
##                                        Mean   :157.0   Mean   :0.4800
##                                        3rd Qu.:180.3   3rd Qu.:0.5075
##                                        Max.   :199.5   Max.   :0.6500
##     PE_Ratio          ROE             ROA        Asset_Turnover
##  Min.   :18.00   Min.   :28.60   Min.   :15.00   Min.   :0.800
##  1st Qu.:18.68   1st Qu.:37.60   1st Qu.:15.97   1st Qu.:0.875
##  Median :21.25   Median :43.10   Median :17.75   Median :0.950
##  Mean   :22.23   Mean   :44.42   Mean   :17.70   Mean   :0.950
##  3rd Qu.:24.80   3rd Qu.:49.92   3rd Qu.:19.48   3rd Qu.:1.025
##  Max.   :28.40   Max.   :62.90   Max.   :20.30   Max.   :1.100
##     Leverage       Rev_Growth    Net_Profit_Margin Median_Recommendation
##  Min.   :0.100   Min.   : 9.37   Min.   :14.10     Length:4
##  1st Qu.:0.145   1st Qu.:15.36   1st Qu.:16.95     Class :character
##  Median :0.220   Median :19.61   Median :19.50     Mode  :character
##  Mean   :0.220   Mean   :18.53   Mean   :19.57
##  3rd Qu.:0.295   3rd Qu.:22.79   3rd Qu.:22.12
##  Max.   :0.340   Max.   :25.54   Max.   :25.20
##    Location           Exchange            cluster
##  Length:4           Length:4           Min.   :1
##  Class :character   Class :character   1st Qu.:1
##  Mode  :character   Mode  :character   Median :1
##                                        Mean   :1
##                                        3rd Qu.:1
##                                        Max.   :1
## ------------------------------------------------------------
## factor(temp_data$cluster): 2
##     Symbol              Name             Market_Cap         Beta
##  Length:8           Length:8           Min.   : 6.30   Min.   :0.1800
##  Class :character   Class :character   1st Qu.:44.67   1st Qu.:0.2875
##  Mode  :character   Mode  :character   Median :59.48   Median :0.4800
##                                        Mean   :55.81   Mean   :0.4138
##                                        3rd Qu.:69.79   3rd Qu.:0.5125
##                                        Max.   :96.65   Max.   :0.6300
##     PE_Ratio          ROE             ROA        Asset_Turnover
##  Min.   :13.10   Min.   :14.90   Min.   : 7.80   Min.   :0.5000
##  1st Qu.:17.65   1st Qu.:21.43   1st Qu.:11.65   1st Qu.:0.6000
##  Median :21.10   Median :26.90   Median :13.35   Median :0.7500
##  Mean   :20.29   Mean   :28.74   Mean   :12.69   Mean   :0.7375
##  3rd Qu.:22.38   3rd Qu.:31.95   3rd Qu.:13.90   3rd Qu.:0.9000
##  Max.   :27.90   Max.   :54.90   Max.   :15.40   Max.   :0.9000
##     Leverage        Rev_Growth     Net_Profit_Margin Median_Recommendation
##  Min.   :0.0000   Min.   :-2.690   Min.   :11.20     Length:8
##  1st Qu.:0.0450   1st Qu.: 2.115   1st Qu.:17.23     Class :character
##  Median :0.3450   Median : 6.630   Median :19.30     Mode  :character
##  Mean   :0.3713   Mean   : 5.591   Mean   :19.35
##  3rd Qu.:0.5400   3rd Qu.: 7.795   3rd Qu.:22.65
##  Max.   :1.1200   Max.   :15.000   Max.   :25.50
```

```
##    Location           Exchange            cluster
##  Length:8           Length:8           Min.   :2
##  Class :character   Class :character   1st Qu.:2
##  Mode  :character   Mode  :character   Median :2
##                                        Mean   :2
##                                        3rd Qu.:2
##                                        Max.   :2
## ------------------------------------------------------------
## factor(temp_data$cluster): 3
##     Symbol              Name             Market_Cap         Beta
##  Length:3           Length:3           Min.   : 7.58   Min.   :0.400
##  Class :character   Class :character   1st Qu.:12.24   1st Qu.:0.405
##  Mode  :character   Mode  :character   Median :16.90   Median :0.410
##                                        Mean   :26.91   Mean   :0.640
##                                        3rd Qu.:36.57   3rd Qu.:0.760
##                                        Max.   :56.24   Max.   :1.110
##     PE_Ratio          ROE             ROA       Asset_Turnover    Leverage
##  Min.   :27.90   Min.   : 3.9   Min.   :1.40   Min.   :0.60   Min.   :0.0000
##  1st Qu.:42.20   1st Qu.: 8.4   1st Qu.:3.45   1st Qu.:0.60   1st Qu.:0.1750
##  Median :56.50   Median :12.9   Median :5.50   Median :0.60   Median :0.3500
##  Mean   :55.63   Mean   :10.1   Mean   :4.20   Mean   :0.70   Mean   :0.3167
##  3rd Qu.:69.50   3rd Qu.:13.2   3rd Qu.:5.60   3rd Qu.:0.75   3rd Qu.:0.4750
##  Max.   :82.50   Max.   :13.5   Max.   :5.70   Max.   :0.90   Max.   :0.6000
##    Rev_Growth     Net_Profit_Margin Median_Recommendation   Location
##  Min.   :-3.170   Min.   :2.600     Length:3              Length:3
##  1st Qu.: 2.995   1st Qu.:4.050     Class :character      Class :character
##  Median : 9.160   Median :5.500     Mode  :character      Mode  :character
##  Mean   : 6.997   Mean   :5.133
##  3rd Qu.:12.080   3rd Qu.:6.400
##  Max.   :15.000   Max.   :7.300
##    Exchange            cluster
##  Length:3           Min.   :3
##  Class :character   1st Qu.:3
##  Mode  :character   Median :3
##                     Mean   :3
##                     3rd Qu.:3
##                     Max.   :3
## ------------------------------------------------------------
## factor(temp_data$cluster): 4
##     Symbol              Name             Market_Cap        Beta
##  Length:1           Length:1           Min.   :0.41   Min.   :0.85
##  Class :character   Class :character   1st Qu.:0.41   1st Qu.:0.85
##  Mode  :character   Mode  :character   Median :0.41   Median :0.85
##                                        Mean   :0.41   Mean   :0.85
##                                        3rd Qu.:0.41   3rd Qu.:0.85
##                                        Max.   :0.41   Max.   :0.85
##     PE_Ratio       ROE             ROA       Asset_Turnover    Leverage
##  Min.   :26   Min.   :24.1   Min.   :4.3   Min.   :0.6    Min.   :3.51
##  1st Qu.:26   1st Qu.:24.1   1st Qu.:4.3   1st Qu.:0.6    1st Qu.:3.51
##  Median :26   Median :24.1   Median :4.3   Median :0.6    Median :3.51
##  Mean   :26   Mean   :24.1   Mean   :4.3   Mean   :0.6    Mean   :3.51
##  3rd Qu.:26   3rd Qu.:24.1   3rd Qu.:4.3   3rd Qu.:0.6    3rd Qu.:3.51
```

```
##  Max.   :26   Max.   :24.1   Max.   :4.3   Max.   :0.6    Max.   :3.51
##    Rev_Growth   Net_Profit_Margin Median_Recommendation   Location
##  Min.   :6.38   Min.   :7.5       Length:1              Length:1
##  1st Qu.:6.38   1st Qu.:7.5       Class :character      Class :character
##  Median :6.38   Median :7.5       Mode  :character      Mode  :character
##  Mean   :6.38   Mean   :7.5
##  3rd Qu.:6.38   3rd Qu.:7.5
##  Max.   :6.38   Max.   :7.5
##    Exchange            cluster
##  Length:1           Min.   :4
##  Class :character   1st Qu.:4
##  Mode  :character   Median :4
##                     Mean   :4
##                     3rd Qu.:4
##                     Max.   :4
## ------------------------------------------------------------
## factor(temp_data$cluster): 5
##     Symbol              Name             Market_Cap         Beta
##  Length:5           Length:5           Min.   : 0.78   Min.   :0.240
##  Class :character   Class :character   1st Qu.: 1.20   1st Qu.:0.320
##  Mode  :character   Mode  :character   Median : 2.60   Median :0.650
##                                        Mean   :11.00   Mean   :0.608
##                                        3rd Qu.: 3.26   3rd Qu.:0.750
##                                        Max.   :47.16   Max.   :1.080
##     PE_Ratio          ROE             ROA       Asset_Turnover    Leverage
##  Min.   : 3.60   Min.   :10.20   Min.   :5.10   Min.   :0.30   Min.   :0.200
##  1st Qu.:18.40   1st Qu.:11.20   1st Qu.:5.40   1st Qu.:0.30   1st Qu.:0.340
##  Median :19.90   Median :15.10   Median :6.80   Median :0.50   Median :0.930
##  Mean   :18.12   Mean   :15.94   Mean   :6.32   Mean   :0.46   Mean   :0.798
##  3rd Qu.:20.10   3rd Qu.:21.40   3rd Qu.:6.80   3rd Qu.:0.60   3rd Qu.:1.070
##  Max.   :28.60   Max.   :21.80   Max.   :7.50   Max.   :0.60   Max.   :1.450
##    Rev_Growth    Net_Profit_Margin Median_Recommendation   Location
##  Min.   :13.99   Min.   :11.00     Length:5              Length:5
##  1st Qu.:26.81   1st Qu.:12.90     Class :character      Class :character
##  Median :29.18   Median :13.30     Mode  :character      Mode  :character
##  Mean   :26.91   Mean   :14.72
##  3rd Qu.:30.37   3rd Qu.:15.10
##  Max.   :34.21   Max.   :21.30
##    Exchange            cluster
##  Length:5           Min.   :5
##  Class :character   1st Qu.:5
##  Mode  :character   Median :5
##                     Mean   :5
##                     3rd Qu.:5
##                     Max.   :5
```

```
#median calculation
recommend_table <- table(temp_data$cluster, temp_data$Median_Recommendation)
names(dimnames(recommend_table)) <- c("Cluster", "Recommendation")
recommend_table <- addmargins(recommend_table)
recommend_table
```

```
##        Recommendation
## Cluster Hold Moderate Buy Moderate Sell Strong Buy Sum
##     1      2            2             0          0   4
##     2      4            1             2          1   8
##     3      2            1             0          0   3
##     4      0            1             0          0   1
##     5      1            2             2          0   5
##     Sum    9            7             4          1  21
```

```
#Location of firm headquarter's breakdown of clusters based on the merged
data
```

```
## function (..., list = character(), package = NULL, lib.loc = NULL,
##     verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
## {
##     fileExt <- function(x) {
##         db <- grepl("\\.[^.]+\\.(gz|bz2|xz)$", x)
##         ans <- sub(".*\\.", "", x)
##         ans[db] <- sub(".*\\.([^.]+\\.)(gz|bz2|xz)$", "\\1\\2",
##             x[db])
##         ans
##     }
##     my_read_table <- function(...) {
##         lcc <- Sys.getlocale("LC_COLLATE")
##         on.exit(Sys.setlocale("LC_COLLATE", lcc))
##         Sys.setlocale("LC_COLLATE", "C")
##         read.table(...)
##     }
##     stopifnot(is.character(list))
##     names <- c(as.character(substitute(list(...))[-1L]), list)
##     if (!is.null(package)) {
##         if (!is.character(package))
##             stop("'package' must be a character vector or NULL")
##     }
##     paths <- find.package(package, lib.loc, verbose = verbose)
##     if (is.null(lib.loc))
##         paths <- c(path.package(package, TRUE), if (!length(package)) getwd(),
##             paths)
##     paths <- unique(normalizePath(paths[file.exists(paths)]))
##     paths <- paths[dir.exists(file.path(paths, "data"))]
##     dataExts <- tools:::.make_file_exts("data")
##     if (length(names) == 0L) {
##         db <- matrix(character(), nrow = 0L, ncol = 4L)
##         for (path in paths) {
##             entries <- NULL
##             packageName <- if (file_test("-f", file.path(path,
##                 "DESCRIPTION")))
##                 basename(path)
##             else "."
##             if (file_test("-f", INDEX <- file.path(path, "Meta",
##                 "data.rds"))) {
##                 entries <- readRDS(INDEX)
##             }
##             else {
##                 dataDir <- file.path(path, "data")
##                 entries <- tools::list_files_with_type(dataDir,
##                   "data")
##                 if (length(entries)) {
##                   entries <- unique(tools::file_path_sans_ext(basename(entries)))
##                   entries <- cbind(entries, "")
##                 }
##             }
##             if (NROW(entries)) {
##                 if (is.matrix(entries) && ncol(entries) == 2L)
```

```
##                   db <- rbind(db, cbind(packageName, dirname(path),
##                     entries))
##                 else warning(gettextf("data index for package %s is invalid and will be ignor
ed",
##                   sQuote(packageName)), domain = NA, call. = FALSE)
##             }
##         }
##         colnames(db) <- c("Package", "LibPath", "Item", "Title")
##         footer <- if (missing(package))
##             paste0("Use ", sQuote(paste("data(package =", ".packages(all.available = TRU
E))")),
##                 "\n", "to list the data sets in all *available* packages.")
##         else NULL
##         y <- list(title = "Data sets", header = NULL, results = db,
##             footer = footer)
##         class(y) <- "packageIQR"
##         return(y)
##     }
##     paths <- file.path(paths, "data")
##     for (name in names) {
##         found <- FALSE
##         for (p in paths) {
##             tmp_env <- if (overwrite)
##                 envir
##             else new.env()
##             if (file_test("-f", file.path(p, "Rdata.rds"))) {
##                 rds <- readRDS(file.path(p, "Rdata.rds"))
##                 if (name %in% names(rds)) {
##                   found <- TRUE
##                   if (verbose)
##                     message(sprintf("name=%s:\t found in Rdata.rds",
##                       name), domain = NA)
##                   thispkg <- sub(".*/([^/]*)/data$", "\\1", p)
##                   thispkg <- sub("_.*$", "", thispkg)
##                   thispkg <- paste0("package:", thispkg)
##                   objs <- rds[[name]]
##                   lazyLoad(file.path(p, "Rdata"), envir = tmp_env,
##                     filter = function(x) x %in% objs)
##                   break
##                 }
##                 else if (verbose)
##                   message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s
\n",
##                     name, paste(names(rds), collapse = ",")),
##                     domain = NA)
##             }
##             if (file_test("-f", file.path(p, "Rdata.zip"))) {
##                 warning("zipped data found for package ", sQuote(basename(dirname(p))),
##                   ".\nThat is defunct, so please re-install the package.",
##                   domain = NA)
##                 if (file_test("-f", fp <- file.path(p, "filelist")))
##                   files <- file.path(p, scan(fp, what = "", quiet = TRUE))
```

```
##                 else {
##                   warning(gettextf("file 'filelist' is missing for directory %s",
##                     sQuote(p)), domain = NA)
##                   next
##                 }
##             }
##             else {
##                 files <- list.files(p, full.names = TRUE)
##             }
##             files <- files[grep(name, files, fixed = TRUE)]
##             if (length(files) > 1L) {
##                 o <- match(fileExt(files), dataExts, nomatch = 100L)
##                 paths0 <- dirname(files)
##                 paths0 <- factor(paths0, levels = unique(paths0))
##                 files <- files[order(paths0, o)]
##             }
##             if (length(files)) {
##                 for (file in files) {
##                   if (verbose)
##                     message("name=", name, ":\t file= ...", .Platform$file.sep,
##                       basename(file), "::\t", appendLF = FALSE,
##                       domain = NA)
##                   ext <- fileExt(file)
##                   if (basename(file) != paste0(name, ".", ext))
##                     found <- FALSE
##                   else {
##                     found <- TRUE
##                     zfile <- file
##                     zipname <- file.path(dirname(file), "Rdata.zip")
##                     if (file.exists(zipname)) {
##                       Rdatadir <- tempfile("Rdata")
##                       dir.create(Rdatadir, showWarnings = FALSE)
##                       topic <- basename(file)
##                       rc <- .External(C_unzip, zipname, topic,
##                         Rdatadir, FALSE, TRUE, FALSE, FALSE)
##                       if (rc == 0L)
##                         zfile <- file.path(Rdatadir, topic)
##                     }
##                     if (zfile != file)
##                       on.exit(unlink(zfile))
##                     switch(ext, R = , r = {
##                       library("utils")
##                       sys.source(zfile, chdir = TRUE, envir = tmp_env)
##                     }, RData = , rdata = , rda = load(zfile,
##                       envir = tmp_env), TXT = , txt = , tab = ,
##                       tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##                       txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##                         header = TRUE, as.is = FALSE), envir = tmp_env),
##                       CSV = , csv = , csv.gz = , csv.bz2 = ,
##                       csv.xz = assign(name, my_read_table(zfile,
##                         header = TRUE, sep = ";", as.is = FALSE),
##                         envir = tmp_env), found <- FALSE)
```

```
##                   }
##                   if (found)
##                     break
##                 }
##                 if (verbose)
##                   message(if (!found)
##                     "*NOT* ", "found", domain = NA)
##             }
##             if (found)
##                 break
##         }
##         if (!found) {
##             warning(gettextf("data set %s not found", sQuote(name)),
##                 domain = NA)
##         }
##         else if (!overwrite) {
##             for (o in ls(envir = tmp_env, all.names = TRUE)) {
##                 if (exists(o, envir = envir, inherits = FALSE))
##                   warning(gettextf("an object named %s already exists and will not be overwri
tten",
##                     sQuote(o)))
##                 else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##                   envir = envir)
##             }
##             rm(tmp_env)
##         }
##     }
##     invisible(names)
## }
## <bytecode: 0x000001c9a4374de0>
## <environment: namespace:utils>
```

```
location_table <- table(temp_data$cluster, temp_data$Location)
names(dimnames(location_table)) <- c("Cluster", "Location")
location_table <- addmargins(location_table)
location_table
```

```
##        Location
## Cluster CANADA FRANCE GERMANY IRELAND SWITZERLAND UK US Sum
##     1        0      0       0       0           0  1  3   4
##     2        0      0       0       0           1  2  5   8
##     3        1      0       1       0           0  0  1   3
##     4        0      0       0       0           0  0  1   1
##     5        0      1       0       1           0  0  3   5
##     Sum      1      1       1       1           1  3 13  21
```

```
#Location of firm headquarter's breakdown of clusters based on the merged
data
```

```
## function (..., list = character(), package = NULL, lib.loc = NULL,
##     verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
## {
##     fileExt <- function(x) {
##         db <- grepl("\\.[^.]+\\.(gz|bz2|xz)$", x)
##         ans <- sub(".*\\.", "", x)
##         ans[db] <- sub(".*\\.([^.]+\\.)(gz|bz2|xz)$", "\\1\\2",
##             x[db])
##         ans
##     }
##     my_read_table <- function(...) {
##         lcc <- Sys.getlocale("LC_COLLATE")
##         on.exit(Sys.setlocale("LC_COLLATE", lcc))
##         Sys.setlocale("LC_COLLATE", "C")
##         read.table(...)
##     }
##     stopifnot(is.character(list))
##     names <- c(as.character(substitute(list(...))[-1L]), list)
##     if (!is.null(package)) {
##         if (!is.character(package))
##             stop("'package' must be a character vector or NULL")
##     }
##     paths <- find.package(package, lib.loc, verbose = verbose)
##     if (is.null(lib.loc))
##         paths <- c(path.package(package, TRUE), if (!length(package)) getwd(),
##             paths)
##     paths <- unique(normalizePath(paths[file.exists(paths)]))
##     paths <- paths[dir.exists(file.path(paths, "data"))]
##     dataExts <- tools:::.make_file_exts("data")
##     if (length(names) == 0L) {
##         db <- matrix(character(), nrow = 0L, ncol = 4L)
##         for (path in paths) {
##             entries <- NULL
##             packageName <- if (file_test("-f", file.path(path,
##                 "DESCRIPTION")))
##                 basename(path)
##             else "."
##             if (file_test("-f", INDEX <- file.path(path, "Meta",
##                 "data.rds"))) {
##                 entries <- readRDS(INDEX)
##             }
##             else {
##                 dataDir <- file.path(path, "data")
##                 entries <- tools::list_files_with_type(dataDir,
##                   "data")
##                 if (length(entries)) {
##                   entries <- unique(tools::file_path_sans_ext(basename(entries)))
##                   entries <- cbind(entries, "")
##                 }
##             }
##             if (NROW(entries)) {
##                 if (is.matrix(entries) && ncol(entries) == 2L)
```

```
##                   db <- rbind(db, cbind(packageName, dirname(path),
##                     entries))
##                 else warning(gettextf("data index for package %s is invalid and will be ignor
ed",
##                   sQuote(packageName)), domain = NA, call. = FALSE)
##             }
##         }
##         colnames(db) <- c("Package", "LibPath", "Item", "Title")
##         footer <- if (missing(package))
##             paste0("Use ", sQuote(paste("data(package =", ".packages(all.available = TRU
E))")),
##                 "\n", "to list the data sets in all *available* packages.")
##         else NULL
##         y <- list(title = "Data sets", header = NULL, results = db,
##             footer = footer)
##         class(y) <- "packageIQR"
##         return(y)
##     }
##     paths <- file.path(paths, "data")
##     for (name in names) {
##         found <- FALSE
##         for (p in paths) {
##             tmp_env <- if (overwrite)
##                 envir
##             else new.env()
##             if (file_test("-f", file.path(p, "Rdata.rds"))) {
##                 rds <- readRDS(file.path(p, "Rdata.rds"))
##                 if (name %in% names(rds)) {
##                   found <- TRUE
##                   if (verbose)
##                     message(sprintf("name=%s:\t found in Rdata.rds",
##                       name), domain = NA)
##                   thispkg <- sub(".*/([^/]*)/data$", "\\1", p)
##                   thispkg <- sub("_.*$", "", thispkg)
##                   thispkg <- paste0("package:", thispkg)
##                   objs <- rds[[name]]
##                   lazyLoad(file.path(p, "Rdata"), envir = tmp_env,
##                     filter = function(x) x %in% objs)
##                   break
##                 }
##                 else if (verbose)
##                   message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s
\n",
##                     name, paste(names(rds), collapse = ",")),
##                     domain = NA)
##             }
##             if (file_test("-f", file.path(p, "Rdata.zip"))) {
##                 warning("zipped data found for package ", sQuote(basename(dirname(p))),
##                   ".\nThat is defunct, so please re-install the package.",
##                   domain = NA)
##                 if (file_test("-f", fp <- file.path(p, "filelist")))
##                   files <- file.path(p, scan(fp, what = "", quiet = TRUE))
```

```
##                 else {
##                   warning(gettextf("file 'filelist' is missing for directory %s",
##                     sQuote(p)), domain = NA)
##                   next
##                 }
##             }
##             else {
##                 files <- list.files(p, full.names = TRUE)
##             }
##             files <- files[grep(name, files, fixed = TRUE)]
##             if (length(files) > 1L) {
##                 o <- match(fileExt(files), dataExts, nomatch = 100L)
##                 paths0 <- dirname(files)
##                 paths0 <- factor(paths0, levels = unique(paths0))
##                 files <- files[order(paths0, o)]
##             }
##             if (length(files)) {
##                 for (file in files) {
##                   if (verbose)
##                     message("name=", name, ":\t file= ...", .Platform$file.sep,
##                       basename(file), "::\t", appendLF = FALSE,
##                       domain = NA)
##                   ext <- fileExt(file)
##                   if (basename(file) != paste0(name, ".", ext))
##                     found <- FALSE
##                   else {
##                     found <- TRUE
##                     zfile <- file
##                     zipname <- file.path(dirname(file), "Rdata.zip")
##                     if (file.exists(zipname)) {
##                       Rdatadir <- tempfile("Rdata")
##                       dir.create(Rdatadir, showWarnings = FALSE)
##                       topic <- basename(file)
##                       rc <- .External(C_unzip, zipname, topic,
##                         Rdatadir, FALSE, TRUE, FALSE, FALSE)
##                       if (rc == 0L)
##                         zfile <- file.path(Rdatadir, topic)
##                     }
##                     if (zfile != file)
##                       on.exit(unlink(zfile))
##                     switch(ext, R = , r = {
##                       library("utils")
##                       sys.source(zfile, chdir = TRUE, envir = tmp_env)
##                     }, RData = , rdata = , rda = load(zfile,
##                       envir = tmp_env), TXT = , txt = , tab = ,
##                       tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##                       txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##                         header = TRUE, as.is = FALSE), envir = tmp_env),
##                       CSV = , csv = , csv.gz = , csv.bz2 = ,
##                       csv.xz = assign(name, my_read_table(zfile,
##                         header = TRUE, sep = ";", as.is = FALSE),
##                         envir = tmp_env), found <- FALSE)
```

```
##                   }
##                   if (found)
##                     break
##                 }
##                 if (verbose)
##                   message(if (!found)
##                     "*NOT* ", "found", domain = NA)
##             }
##             if (found)
##                 break
##         }
##         if (!found) {
##             warning(gettextf("data set %s not found", sQuote(name)),
##                 domain = NA)
##         }
##         else if (!overwrite) {
##             for (o in ls(envir = tmp_env, all.names = TRUE)) {
##                 if (exists(o, envir = envir, inherits = FALSE))
##                   warning(gettextf("an object named %s already exists and will not be overwri
tten",
##                     sQuote(o)))
##                 else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##                   envir = envir)
##             }
##             rm(tmp_env)
##         }
##     }
##     invisible(names)
## }
## <bytecode: 0x000001c9a4374de0>
## <environment: namespace:utils>
```

```
location_table <- table(temp_data$cluster, temp_data$Location)
names(dimnames(location_table)) <- c("Cluster", "Location")
location_table <- addmargins(location_table)
location_table
```

```
##        Location
## Cluster CANADA FRANCE GERMANY IRELAND SWITZERLAND UK US Sum
##     1        0      0       0       0           0  1  3   4
##     2        0      0       0       0           1  2  5   8
##     3        1      0       1       0           0  0  1   3
##     4        0      0       0       0           0  0  1   1
##     5        0      1       0       1           0  0  3   5
##     Sum      1      1       1       1           1  3 13  21
```

```
#summarizing the stock exchange values for each cluster
#creating a data frame for the merged data and initializing the exchange
table
```

```
## function (..., exclude = if (useNA == "no") c(NA, NaN), useNA = c("no",
##     "ifany", "always"), dnn = list.names(...), deparse.level = 1)
## {
##     list.names <- function(...) {
##         l <- as.list(substitute(list(...)))[-1L]
##         if (length(l) == 1L && is.list(..1) && !is.null(nm <- names(..1)))
##             return(nm)
##         nm <- names(l)
##         fixup <- if (is.null(nm))
##             seq_along(l)
##         else nm == ""
##         dep <- vapply(l[fixup], function(x) switch(deparse.level +
##             1, "", if (is.symbol(x)) as.character(x) else "",
##             deparse(x, nlines = 1)[1L]), "")
##         if (is.null(nm))
##             dep
##         else {
##             nm[fixup] <- dep
##             nm
##         }
##     }
##     miss.use <- missing(useNA)
##     miss.exc <- missing(exclude)
##     useNA <- if (miss.use && !miss.exc && !match(NA, exclude,
##         nomatch = 0L))
##         "ifany"
##     else match.arg(useNA)
##     doNA <- useNA != "no"
##     if (!miss.use && !miss.exc && doNA && match(NA, exclude,
##         nomatch = 0L))
##         warning("'exclude' containing NA and 'useNA' != \"no\"' are a bit contradicting")
##     args <- list(...)
##     if (length(args) == 1L && is.list(args[[1L]])) {
##         args <- args[[1L]]
##         if (length(dnn) != length(args))
##             dnn <- paste(dnn[1L], seq_along(args), sep = ".")
##     }
##     if (!length(args))
##         stop("nothing to tabulate")
##     bin <- 0L
##     lens <- NULL
##     dims <- integer()
##     pd <- 1L
##     dn <- NULL
##     for (a in args) {
##         if (is.null(lens))
##             lens <- length(a)
##         else if (length(a) != lens)
##             stop("all arguments must have the same length")
##         fact.a <- is.factor(a)
##         if (doNA)
##             aNA <- anyNA(a)
```

```
##         if (!fact.a) {
##             a0 <- a
##             op <- options(warn = 2)
##             on.exit(options(op))
##             a <- factor(a, exclude = exclude)
##             options(op)
##         }
##         add.na <- doNA
##         if (add.na) {
##             ifany <- (useNA == "ifany")
##             anNAc <- anyNA(a)
##             add.na <- if (!ifany || anNAc) {
##                 ll <- levels(a)
##                 if (add.ll <- !anyNA(ll)) {
##                   ll <- c(ll, NA)
##                   TRUE
##                 }
##                 else if (!ifany && !anNAc)
##                   FALSE
##                 else TRUE
##             }
##             else FALSE
##         }
##         if (add.na)
##             a <- factor(a, levels = ll, exclude = NULL)
##         else ll <- levels(a)
##         a <- as.integer(a)
##         if (fact.a && !miss.exc) {
##             ll <- ll[keep <- which(match(ll, exclude, nomatch = 0L) ==
##                 0L)]
##             a <- match(a, keep)
##         }
##         else if (!fact.a && add.na) {
##             if (ifany && !aNA && add.ll) {
##                 ll <- ll[!is.na(ll)]
##                 is.na(a) <- match(a0, c(exclude, NA), nomatch = 0L) >
##                   0L
##             }
##             else {
##                 is.na(a) <- match(a0, exclude, nomatch = 0L) >
##                   0L
##             }
##         }
##         nl <- length(ll)
##         dims <- c(dims, nl)
##         if (prod(dims) > .Machine$integer.max)
##             stop("attempt to make a table with >= 2^31 elements")
##         dn <- c(dn, list(ll))
##         bin <- bin + pd * (a - 1L)
##         pd <- pd * nl
##     }
##     names(dn) <- dnn
```

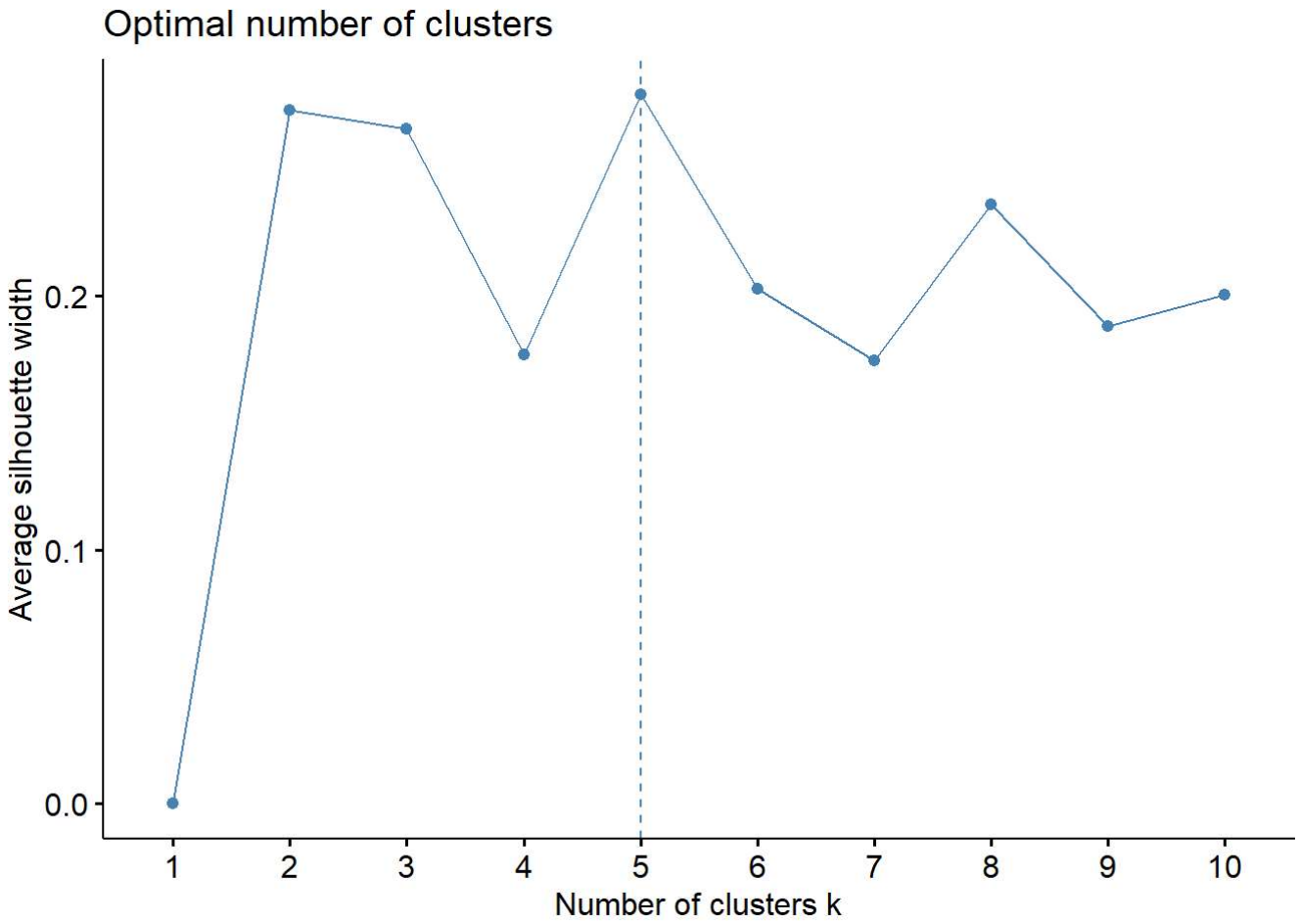
```
##     bin <- bin[!is.na(bin)]
##     if (length(bin))
##         bin <- bin + 1L
##     y <- array(tabulate(bin, pd), dims, dimnames = dn)
##     class(y) <- "table"
##     y
## }
## <bytecode: 0x000001c9a3f59a40>
## <environment: namespace:base>
```

```
exchange_table <- table(temp_data$cluster, temp_data$Exchange)
names(dimnames(exchange_table)) <- c("Cluster", "Exchange")
exchange_table <- addmargins(exchange_table)
exchange_table
```

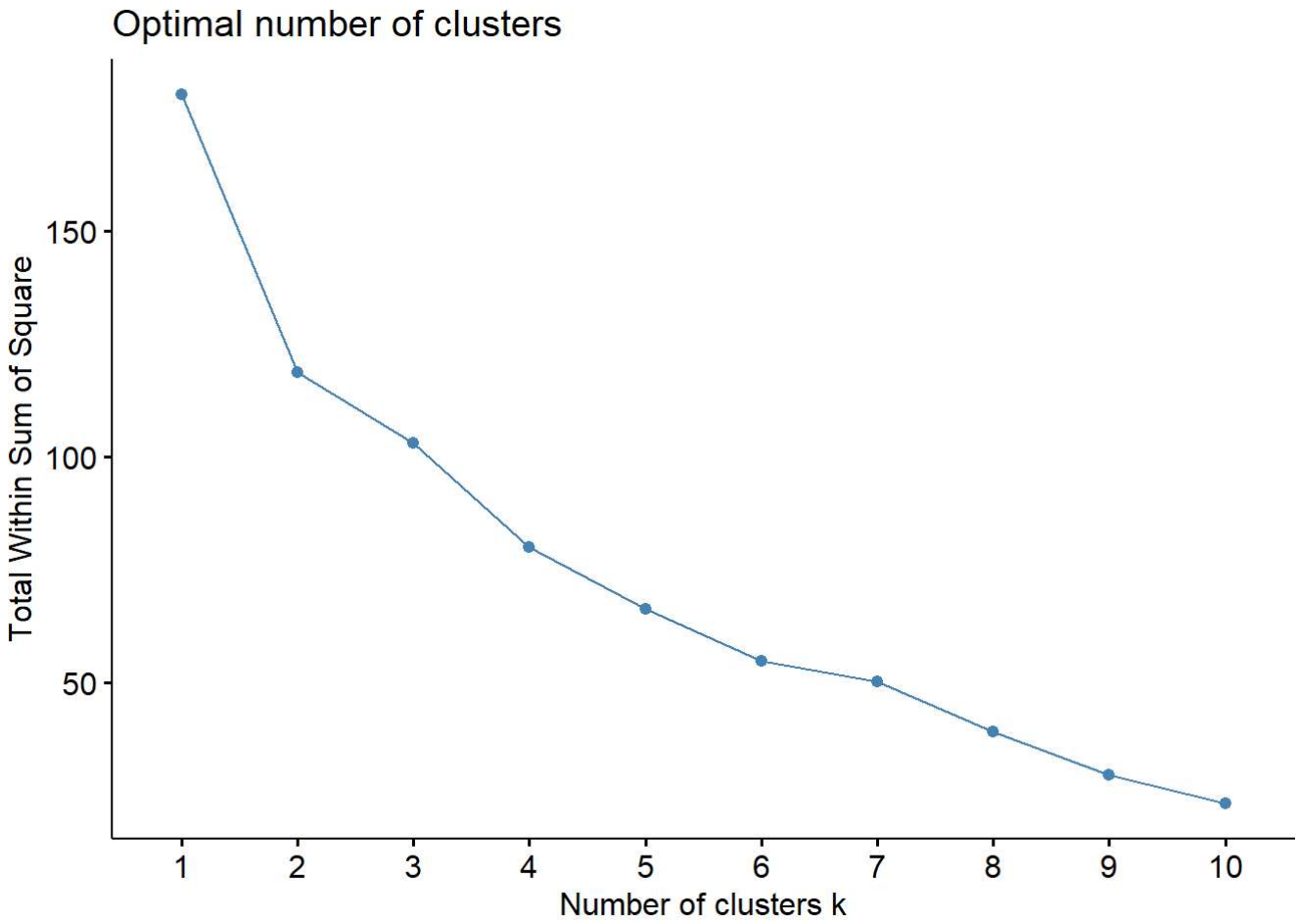
```
##        Exchange
## Cluster AMEX NASDAQ NYSE Sum
##     1      0      0    4   4
##     2      0      0    8   8
##     3      0      0    3   3
##     4      0      1    0   1
##     5      1      0    4   5
##     Sum    1      1   19  21
```

##There are 21 companies overall, divided into 1 Amex, 1 Nasdaq, and 19 NYSE. Cluster 5 just has the NYSE. All three are in Cluster 2. clusters 1,3,4 all contains only NYSE. ##Question d - Provide an appropriate name for each cluster using any or all of the variables in the dataset. Answer :- Cluster 1: - The cluster 3 can be named as “Small_Net_Profit_Margin-High PE ratio”. All are NAM companies. Cluster 2: The cluster number 2 can be named “Low_Market_Cap & Less_ROA” - Hold or Buy exchanges Cluster 3: The cluster 4 can be named “High Market Cap - more RoE - more RoA- High Asset Turnover- more NetProfitMargin” - All are the Hold or Buy US companies that are part of NYSE Cluster 4: The cluster 4 can be named as “least PE ratio & low RoE & Minimum Asset Turnover- High revenue growth - mixed recommendation. All are US or European companies that belongs to NYSE. Cluster 5: The cluster number 5 can be named as “Least_Revenue_growth”. It mostly comprised of US companies and all are NYSE. ##Also Trying or Exploring the other algorithms whether they can perform better clustering or not? -

```
fviz_nbclust(range_pharma_data, FUN = kmeans, method = "silhouette")
```




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



Plotting the K means and clusters

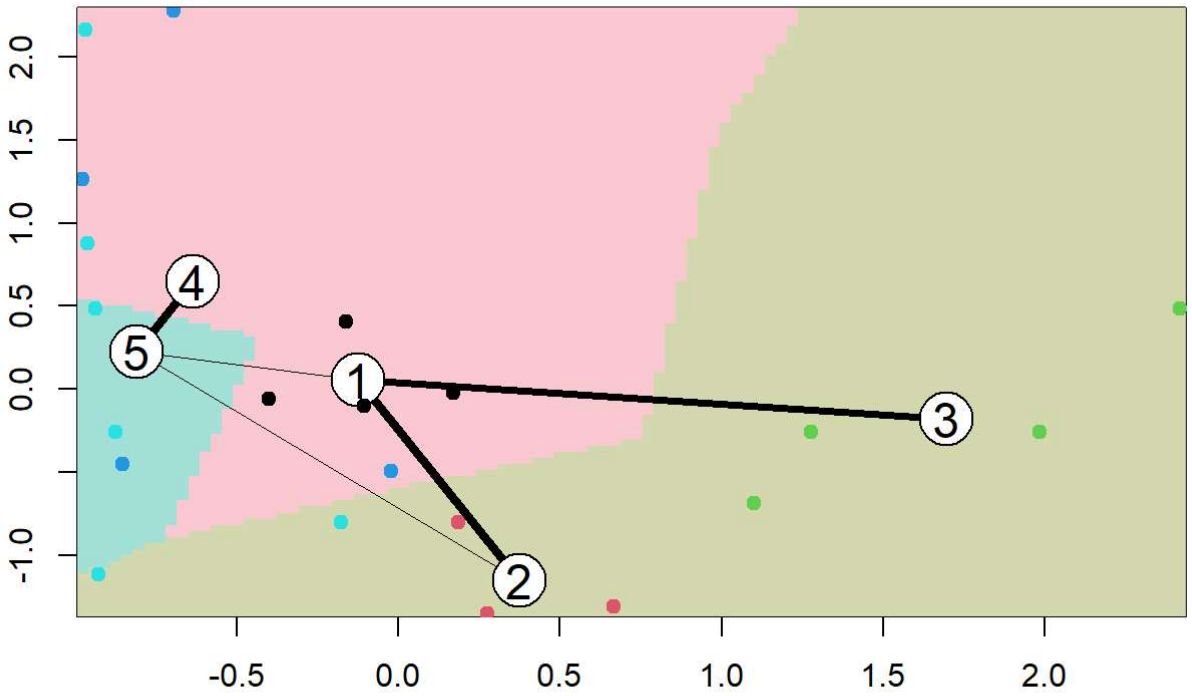
```
kmeans_2 = kcca(Scaled_pharma_data, k=5, kccaFamily("kmeans"))
kmeans_2
```

```
## kcca object of family 'kmeans'
##
## call:
## kcca(x = Scaled_pharma_data, k = 5, family = kccaFamily("kmeans"))
##
## cluster sizes:
##
## 1 2 3 4 5
## 4 3 4 4 6
```

```
clusters(kmeans_2)
```

```
##  [1] 2 4 5 1 5 4 1 4 5 2 3 5 3 5 3 2 3 4 1 5 1
```

```
#Applying the predict() function
clusters_index <- predict(kmeans_2)
image(kmeans_2)
points(Scaled_pharma_data, col=clusters_index, pch=19, cex=1.0)
```



##Here, we execute a kmeans cluster on k = 5 using the kcca algorithm rather than the kmeans function from base R. The clustering has the same size but different assignment between points compared to base R approach. The clustering graph demonstrates that the clustering is not as clear-cut as we would like, particularly between clusters 1, 2 and 3

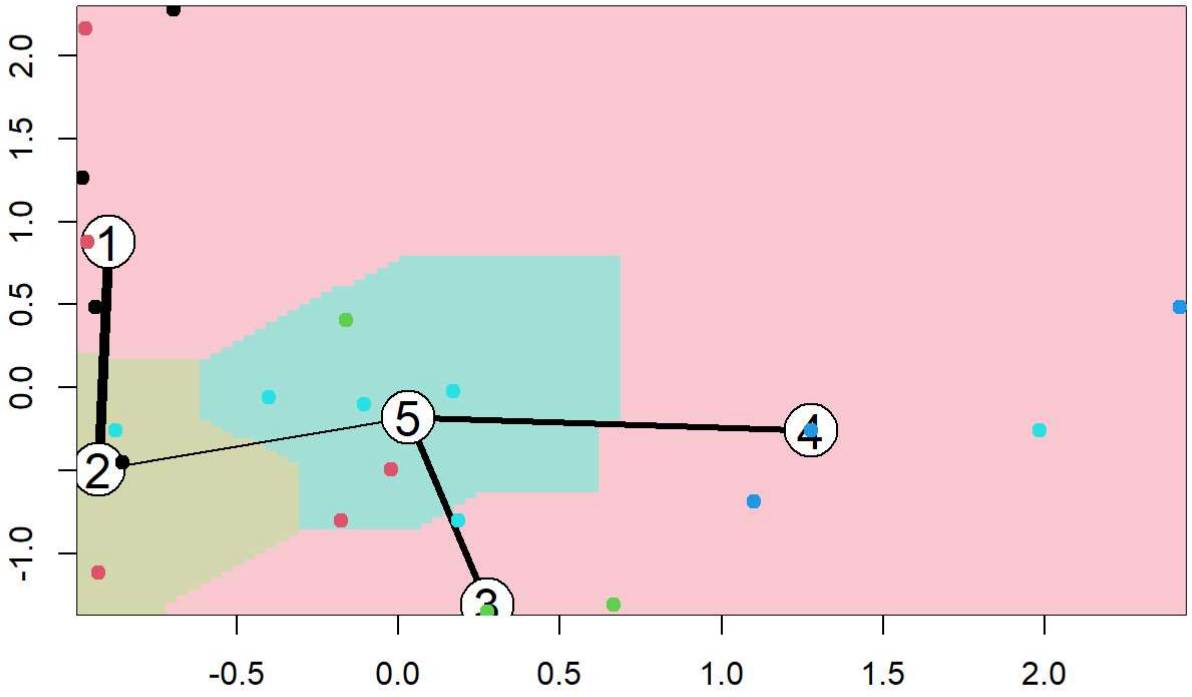
```
kmeans_2 = kcca(Scaled_pharma_data, k=5, kccaFamily("kmedians"))
kmeans_2
```

```
## kcca object of family 'kmedians'
##
## call:
## kcca(x = Scaled_pharma_data, k = 5, family = kccaFamily("kmedians"))
##
## cluster sizes:
##
## 1 2 3 4 5
## 4 5 3 3 6
```

```
clusters(kmeans_2) #clusteing
```

```
##  [1] 5 1 5 5 2 1 5 1 2 3 4 1 5 2 4 3 4 2 5 2 3
```

```
clusters_index <- predict(kmeans_2)
image(kmeans_2)
points(Scaled_pharma_data, col=clusters_index, pch=19, cex=1.0)
```



if we change from kmeans to kmedian in kcca, the five clusters' sizes are 4, 5, 3, 3, and 6. Yet, the clustering is not as obvious. We are investigating the extra data to determine if there are any better techniques or tools we can use to enhance the visual cluster, but it is unclear whether a better cluster actually exists.