Assignment week 2: Cluster Analysis

Rawinan Soma

Cluster analysis is the unsupervised machine learning algorithm for classified unknown data into clusters that have similarities and distinguish between other clusters. This assignment is one of the example for understanding clustering methods like k-means algorithm, and other interesting method. For now, I'm going to use "processed.cleveland.data" dataset about heart disease for clustering.

I'm starting with setting your working directory and loading the dataset into your computer.

I have to convert provide data file into .txt file by notepad, for importing purpose.

```
setwd("D:/Work-BHI/ML and Data mining/assignment2")
library(readr)
data <- read_csv("cleveland.txt", col_names = FALSE)
```

Here are the first 5 rows of your dataset.

```
head(data)
```

```
## # A tibble: 6 x 14
##      X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11 X12
X13
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
<chr>
## 1    63     1     1   145   233     1     2   150     0   2.3     3 0.0
6.0
## 2    67     1     4   160   286     0     2   108     1   1.5     2 3.0
3.0
## 3    67     1     4   120   229     0     2   129     1   2.6     2 2.0
7.0
## 4    37     1     3   130   250     0     0   187     0   3.5     3 0.0
3.0
## 5    41     0     2   130   204     0     2   172     0   1.4     1 0.0
3.0
## 6    56     1     2   120   236     0     0   178     0   0.8     1 0.0
3.0
## # ... with 1 more variable: X14 <dbl>
```

You'll see our dataset has no columns name LOL, but don't worry we find the metadata in this link http://archive.ics.uci.edu/ml/datasets/Heart+Disease and replace the columns name

```
library(tidyverse)
old_colNames <- colnames(data)
new_colNames <- c('age', 'sex', 'cp', 'trestbp', 'chol', 'fbs', 'restecg',
```

```
'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num')
data <- data %>%
  rename_at(vars(old_colNames), ~new_colNames)
```

The dataset should have the columns name by now

```
head(data)
```

```
## # A tibble: 6 x 14
##     age   sex    cp trestbp  chol   fbs restecg thalach exang oldpeak
slope
##   <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl>   <dbl>   <dbl> <dbl>   <dbl>
<dbl>
## 1    63     1     1     145   233     1       2     150     0     2.3
3
## 2    67     1     4     160   286     0       2     108     1     1.5
2
## 3    67     1     4     120   229     0       2     129     1     2.6
2
## 4    37     1     3     130   250     0       0     187     0     3.5
3
## 5    41     0     2     130   204     0       2     172     0     1.4
1
## 6    56     1     2     120   236     0       0     178     0     0.8
1
## # ... with 3 more variables: ca <chr>, thal <chr>, num <dbl>
```

The cluster analysis is the unsupervised learning. So, we don't need the class attribute or "num" column

```
data <- data %>% select(-num)
```

Before clustering, we should make sure the data is clean enough for entering the model such as no missing value, no outlier.

```
summary(data)
```

```
##       age             sex               cp           trestbp
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0
##  Median :56.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :54.44   Mean   :0.6799   Mean   :3.158   Mean   :131.7
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##       chol            fbs             restecg          thalach
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :241.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.7   Mean   :0.1485   Mean   :0.9901   Mean   :149.6
##  3rd Qu.:275.0   3rd Qu.:0.0000   3rd Qu.:2.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
```

```
##      exang           oldpeak         slope            ca
##  Min.   :0.0000   Min.   :0.00   Min.   :1.000   Length:303
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   Class :character
##  Median :0.0000   Median :0.80   Median :2.000   Mode  :character
##  Mean   :0.3267   Mean   :1.04   Mean   :1.601
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000
##  Max.   :1.0000   Max.   :6.20   Max.   :3.000
##      thal
##  Length:303
##  Class :character
##  Mode  :character
##
##
##
```

The dataset is clean enough for analysis but, I will rename the columns for better understanding.

```
### change variables name
data <- data %>%
  rename(chest_pain_type = cp,
         rest_bp = trestbp,
         max_hr = thalach,
         exercise_angina = exang,
         stdep = oldpeak,
         num_vessel = ca)
### add index (ID) column
data <- data %>%
  mutate(id = row_number()) %>%
  relocate(id)
### Make new table name "cleveland_kmm"
cleveland_kmm <- data
```

K-mean is one of methods for clustering by divide all data point into *k* groups. At first, the centroids of this dataset will randomly generates. After that, the distance between centroids to all data point will be calculated and assigned data point to cluster that has the shortest distance, then new centroid will be evaluated from mean of the cluster objects, At last, object will be reassigned by distance of the new centroid. This process will iterated until the centroid stopped moving or some criteria.

```
### Let us apply k = 3 clusters
set.seed(99)
kmeans(cleveland_kmm, centers = 3)

## Warning in storage.mode(x) <- "double": NAs introduced by coercion

## Error in do_one(nmeth): NA/NaN/Inf in foreign function call (arg 1)
```

No!!, you got errors. I found some explanation in Stackoverflow tell that there was some of columns are not numeric type and some of them has NA. So, I tried to change them.

```
cleveland_kmm <- cleveland_kmm %>%
  mutate(thal = replace(thal, thal =='3.0', 3)) %>%
  mutate(thal = replace(thal, thal =='6.0', 6)) %>%
  mutate(thal = replace(thal, thal =='7.0', 7)) %>%
  mutate(thal = replace(thal, thal =='?', 3))

cleveland_kmm <- cleveland_kmm %>%
  mutate(num_vessel = replace(num_vessel, num_vessel =='?','0')) %>%
  mutate(num_vessel = as.numeric(num_vessel))

cleveland_kmm <- cleveland_kmm %>%
  mutate(num_vessel = as.numeric(num_vessel)) %>%
  mutate(thal = as.numeric(thal))
```

And try again

```
set.seed(99)
kmeans(cleveland_kmm[,-1], centers = 3)
```

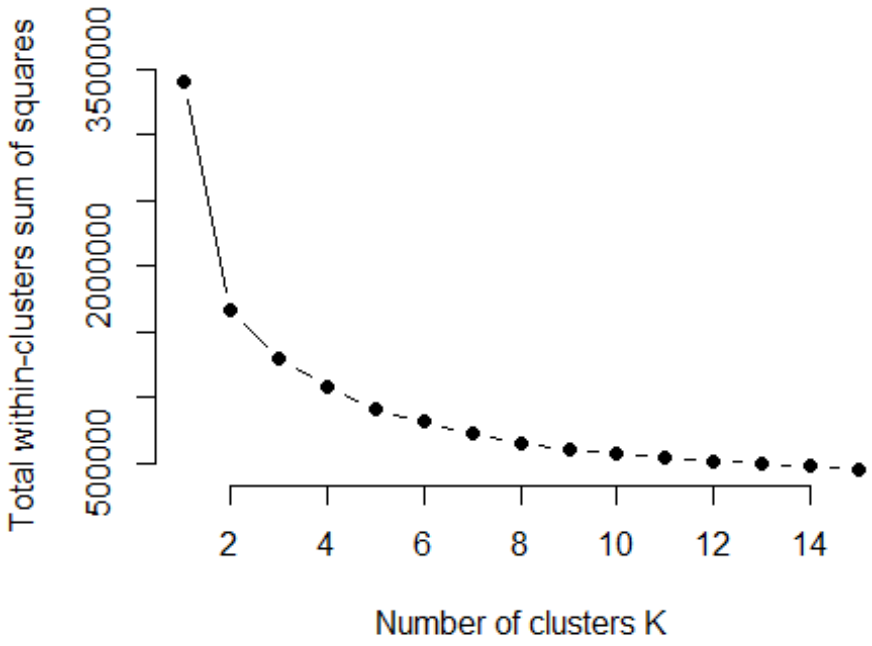
```
## K-means clustering with 3 clusters of sizes 133, 109, 61
##
## Cluster means:
##        age       sex chest_pain_type  rest_bp     chol       fbs   restecg
## 1 55.71429 0.7142857        3.105263 132.4211 252.3759 0.1578947 1.1278195
## 2 51.84404 0.7339450        3.128440 127.7615 197.6972 0.1284404 0.7522936
## 3 56.29508 0.5081967        3.327869 137.1148 321.8525 0.1639344 1.1147541
##     max_hr exercise_angina     stdep    slope num_vessel     thal
## 1 147.9323       0.3458647 1.0323308 1.676692  0.7067669 4.812030
## 2 150.8257       0.2844037 0.9972477 1.541284  0.4678899 4.605505
## 3 151.0820       0.3606557 1.1311475 1.540984  0.9180328 4.737705
##
## Clustering vector:
##   [1] 1 1 1 1 2 1 1 3 1 2 2 3 1 1 2 2 1 1 1 1 2 1 1 2 2 2 3 1 1 2 1 1 3 1
1 2 2
##  [38] 1 3 1 1 2 3 2 3 1 2 1 3 2 2 2 3 2 1 1 1 2 1 2 3 2 2 3 2 1 2 1 3 1 1
1 1 1
##  [75] 2 3 1 3 1 1 2 1 3 1 3 1 1 2 1 1 3 2 1 2 1 1 1 1 2 2 1 2 3 1 2 3 2 1
1 2 3
## [112] 1 2 3 1 2 2 2 3 1 1 3 2 2 1 1 3 1 2 2 1 2 2 1 2 1 2 1 2 1 2 3 2 3 1
1 3 1
## [149] 3 3 3 1 3 3 1 3 3 3 3 1 2 3 2 1 1 2 2 3 1 2 1 1 1 3 2 1 1 2 3 1 1 3
1 1 3
## [186] 2 1 1 3 1 2 3 1 3 2 3 1 1 1 1 1 3 2 3 2 3 1 2 1 1 2 1 2 1 1 2 2 1 3
1 1 1
## [223] 2 1 1 2 2 1 2 2 2 3 2 1 2 1 1 1 1 3 1 3 1 1 2 1 1 1 2 2 2 2 1 3 3 2
2 2 1
## [260] 1 1 3 1 2 2 3 2 2 2 2 2 1 3 2 2 1 1 2 1 2 3 1 2 2 2 3 1 2 2 1 2 3 2
2 2 2
## [297] 2 1 1 2 2 1 2
##
```

```
## Within cluster sum of squares by cluster:
## [1] 165163.8 143623.7 163792.7
##  (between_SS / total_SS =  56.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

You'll find this data divide into 3 clusters of 133, 109, and 61 points. In details, group 2 has lower average age, blood pressure, and cholesterol level. This cluster analysis is quite explainable between clusters with btw_ss/total_ss = *56.6%*

The *k* is the number of appropriate clusters to divide the data point which considered from clustering pattern, explainable cluster, and elbow method. The elbow method looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster did not give much better model.

```
set.seed(99)
k_max <- 15
wss <- sapply(1:k_max,
              function(k){kmeans(cleveland_kmm,
                                 k, nstart=50,
                                 iter.max = 15 )$tot.withinss})
plot(1:k_max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```



From the elbow plot, we'll see significant reduction of variance start at k = 4. So, we'll select 4 cluster for next analysis. This analysis better explanation with btw_ss/total_ss = *62.5%*

```
kmeans(cleveland_kmm[,-1], centers = 4)

## K-means clustering with 4 clusters of sizes 85, 59, 99, 60
##
## Cluster means:
##        age       sex chest_pain_type  rest_bp     chol       fbs   restecg
## 1 53.15294 0.7411765        3.176471 128.2941 191.9176 0.1529412 0.8235294
## 2 56.50847 0.5084746        3.169492 135.0678 322.4407 0.1525424 1.1525424
## 3 51.24242 0.6868687        3.000000 128.6970 239.3939 0.1414141 0.8888889
## 4 59.50000 0.7500000        3.383333 138.1167 261.8500 0.1500000 1.2333333
##     max_hr exercise_angina     stdep    slope num_vessel     thal
## 1 146.1529       0.3294118 1.1352941 1.611765  0.5058824 4.776471
## 2 153.9322       0.3220339 1.0389831 1.491525  0.7627119 4.525424
## 3 164.6768       0.2020202 0.7161616 1.484848  0.5454545 4.303030
## 4 125.3833       0.5333333 1.4383333 1.883333  0.9833333 5.533333
##
## Clustering vector:
##   [1] 3 4 4 3 1 3 3 2 4 1 1 2 4 3 1 1 3 3 4 3 1 2 2 3 1 3 2 4 3 1 3 3 2 3
3 3 1
##  [38] 4 2 4 4 1 2 1 2 3 1 4 2 1 1 1 2 3 4 4 3 1 4 1 2 1 1 2 1 4 1 3 2 3 4
3 4 3
##  [75] 1 2 4 2 3 4 1 4 2 4 2 3 3 1 3 3 2 1 3 1 3 3 3 3 1 3 3 1 2 4 1 2 1 3
4 1 2
```

```
## [112] 3 1 2 4 1 1 1 2 4 3 2 3 1 2 3 4 4 3 1 3 3 3 3 3 3 1 4 1 3 3 2 3 2 3
3 4 3
## [149] 2 2 2 4 2 4 4 2 2 2 2 4 1 2 1 4 3 1 3 2 4 1 4 4 4 2 1 4 3 1 2 3 3 2
3 4 2
## [186] 1 3 4 2 4 1 4 3 4 1 2 4 3 3 4 3 2 1 2 1 2 4 1 3 3 3 3 3 3 3 1 1 3 2
3 3 3
## [223] 1 4 3 3 1 3 1 1 1 2 1 4 1 4 4 3 3 2 3 2 3 3 1 4 3 4 3 1 1 4 4 2 2 3
3 1 4
## [260] 4 3 2 3 3 1 2 1 1 3 1 1 4 2 1 1 3 4 3 3 1 2 3 1 1 1 2 3 1 3 3 1 2 1
1 1 1
## [297] 1 4 4 1 1 3 1
##
## Within cluster sum of squares by cluster:
## [1] 105424.46 154838.39  75958.91  71809.33
##  (between_SS / total_SS =  62.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

In the recent example, you will be curious about average value of sex: it means half man half woman? *K-means Clustering* has some weakness, it require euclidean distance for calculate disimilarity. So, it cannot clustering the dataset that contains both numerical and categorical data For example, I tried to run k-means with mixed dataset.

```
cleveland_mix <- data %>%
  mutate(across(c(3,4,7,8,10,12,14), ~as.factor(.)))

kmeans(cleveland_mix[,-1], centers = 4)

## Warning in storage.mode(x) <- "double": NAs introduced by coercion

## Error in do_one(nmeth): NA/NaN/Inf in foreign function call (arg 1)
```

Therefore, we need another clustering algorithm for mixed dataset. Partitioning around medoids (PAM) is the one of solution, it required dissimilarity matrix for clustering instead of euclidean distance.

```
cleveland_mix <- data %>%
  mutate(thal = replace(thal, thal =='?', 3)) %>%
  mutate(num_vessel = replace(num_vessel, num_vessel =='?','0')) %>%
  mutate(num_vessel = as.numeric(num_vessel)) %>%
  mutate(across(c(3,4,7,8,10,12,14), ~as.character(.)))

set.seed(99)
library(cluster)
pam(cleveland_mix[,-1], k=4, diss = FALSE)
```

```
## Medoids:
##       ID age sex chest_pain_type rest_bp chol fbs restecg max_hr
## [1,] 140  51   2               3     125  245   2       3    166
## [2,] 207  58   2               4     128  259   1       3    130
## [3,]  50  53   2               3     130  197   2       3    152
## [4,] 262  58   1               2     136  319   2       3    152
##      exercise_angina stdep slope num_vessel thal
## [1,]               1   2.4     2          0    2
## [2,]               2   3.0     2          2    4
## [3,]               1   1.2     3          0    2
## [4,]               1   0.0     1          2    2
## Clustering vector:
##   [1] 1 2 2 1 3 1 1 4 2 3 3 4 2 1 3 3 1 1 2 1 3 4 4 1 3 3 4 2 1 3 1 1 4 1
1 1 3
##  [38] 2 4 2 2 3 4 3 4 1 3 2 4 3 3 3 4 1 2 2 1 3 2 3 4 3 3 4 3 2 3 1 4 1 2
1 2 1
##  [75] 3 4 2 4 1 2 3 2 4 2 4 1 1 3 1 2 4 3 1 3 1 1 2 1 3 1 1 3 4 2 3 4 3 1
2 3 4
## [112] 2 3 4 2 3 3 3 4 2 1 4 3 3 1 1 4 2 1 3 2 1 3 1 3 1 3 2 3 1 3 4 3 4 1
1 2 1
## [149] 4 4 4 2 4 4 2 4 4 4 4 2 3 4 3 2 1 3 1 4 1 3 2 2 2 4 3 2 1 3 4 1 1 4
1 2 4
## [186] 3 1 2 4 2 3 4 2 2 3 2 2 1 1 2 1 4 3 4 3 4 2 3 1 1 3 1 3 1 1 3 3 1 4
1 1 1
## [223] 3 2 1 1 3 1 3 3 3 4 3 2 3 2 2 2 1 4 1 4 1 1 3 2 1 2 3 3 3 2 2 4 4 3
3 3 2
## [260] 2 1 4 1 1 3 4 3 3 1 3 3 2 4 3 3 3 2 3 1 3 4 1 3 3 3 4 3 3 1 1 3 4 3
3 3 3
## [297] 3 2 2 3 3 1 3
## Objective function:
##    build     swap
## 32.86880 32.10439
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

Lastly, here are the summary of characteristic for each cluster from PAM method

	Clusters (n)			
	1 (81)	2 (63)	3 (99)	4 (60)
Age (median)	51	59	53	57
Sex				
male	58 (72%)	48 (76%)	70 (70%)	30 (50%)
Chest pain type				
Typical	7 (9%)	5 (8%)	8 (8%)	3 (5%)
Atypical	18 (22%)	5 (8%)	15 (15%)	12 (20%)
Non-anginal	27 (33%)	14 (22%)	32 (32%)	13 (22%)
Asymptomatic	29 (36%)	39 (62%)	44 (45%)	32 (53%)
Resting SBP (median)	130	130	128	133
Cholesterol level (median)	240	261	200	309
FBS				
>120 mg%	11 (14%)	10 (16%)	15 (15%)	9 (15%)
Resting ECG				
ST-T abnormal	0	0	2 (2%)	2 (3%)
LV hypertrophy	36 (44%)	41 (65%)	38 (38%)	33 (55%)
normal	45 (56%)	22 (35%)	59 (60%)	25 (42%)
Max HR (median)	165	128	150	157
Exercise-induced angina				
yes	15 (19%)	32 (51%)	31 (31%)	21 (35%)
ST depression (median)	0.2	1	0.8	0.9
Slope				
upslope	50 (62%)	14 (22%)	46 (46%)	32 (53%)
downslope	6 (7%)	5 (8%)	7 (8%)	3 (5%)
flat	25 (31%)	44 (70%)	46 (46%)	25 (42%)
Number of vessel occlusion				
0	53 (65%)	25 (40%)	68 (69%)	34 (57%)
1	12 (15%)	25 (40%)	17 (17%)	11 (18%)
2	13 (16%)	7 (10%)	10 (10%)	8 (13%)
3	3 (4%)	6 (10%)	4 (4%)	7 (12%)