Riding with BK: BK 교수와 함께하는 R 통계분석

이 병 관
한양대학교 광고홍보학과 교수

2017년 2월 28일

차 례

제 1 장

Market Segmentation: Cluster Analysis

시장세분화를 위한 군집분석

시장세분화(market segmentation)는 가장 기본적인 전략적 마케팅 개념이다. 최상의 세분화가 수행될 때 그 조직은 시장에서의 성공을 보장받을 수 있다. 시장을 세분화 한다는 것은 조직의 잠재 소비자를 다음과 같은 기준을 통해 하위집단으로 나누는 것을 말한다:

- 동일한 그룹의 소비자는 주어진 특성 집합과 유사하다.
- 상이한 그룹에 속한 소비자는 동일한 특성 집합과 관련하여 유사하지 않다.

군집분석(cluster analysis)은 이러한 시장세분화를 위해 사용할 수 있는 유용한 분석방법 중 하나이다. 군집분은 각 케이스를 다음과 같은 그룹으로 분류하는데 사용되는 통계 기법이다.

- 그룹 내에서는 비교적 동질하고,
- 그룹 간에는 서로 이질적이며,
- 동질성(유사성)과 이질성(차이점)은 정의된 변수를 통해 측정된다

일반적으로 군집분석은 다음과 같은 목적을 위해 사용된다.

- 개체를 그룹으로 분류하거나 세분화하여 각 그룹 내의 개체가 다양한 변수에 대해 서로 유사하게 만든다.
- 세그먼트 내에서는 유사성이 높고, 세그먼트 간에서는 가능한 차이가 커지도록 개체를 분류하거나 세분화한다.
- 특정 변수를 종속변수로 지정하지 않고 여러 변수 중에서 자연 그룹 또는 세그먼트를 찾는다.

군집분석을 사용한 시장 세분화의 예는 다음과 같다.

- 경쟁 제품과 비교하여 제품을 조사하는 신제품 연구
- 테스트 마케팅을 위해 지역을 동종 클러스터로 그룹화하는 테스트 마케팅
- 비슷한 선택 기준을 가진 유사한 구매자 그룹을 식별하는 구매자 행동
- 지리적, 인구통계학적, 심리학적 및 행동적 변수를 기반의 명확한 시장 세크먼트를 개발하기 위한 시장 세분화

예제 1: European Protein Consumption

25개 유럽 국가(n = 25 units)의 단백질 섭취(퍼센트)의 주요 공급원(p = 9)에 대한 데이터를 가정해보자. 예를 들어, 오스트리아는 붉은 고기에서 8.9%, 우유에서 19.9% 등의 단백질을 섭취한다. 이때, 25개국을 몇개의 클러스터로 분리할 수 있는지 알아 보는 것이 중요하다. 지중해 국가들은 북유럽 및 독일어권 국가에서 선호하는 음식과 다른 특정 식품 카테고리에서 단백질 섭취량을 얻을지도 모른다. 오리지널 데이터는 Hand et al.(1994)을 사용하였다.

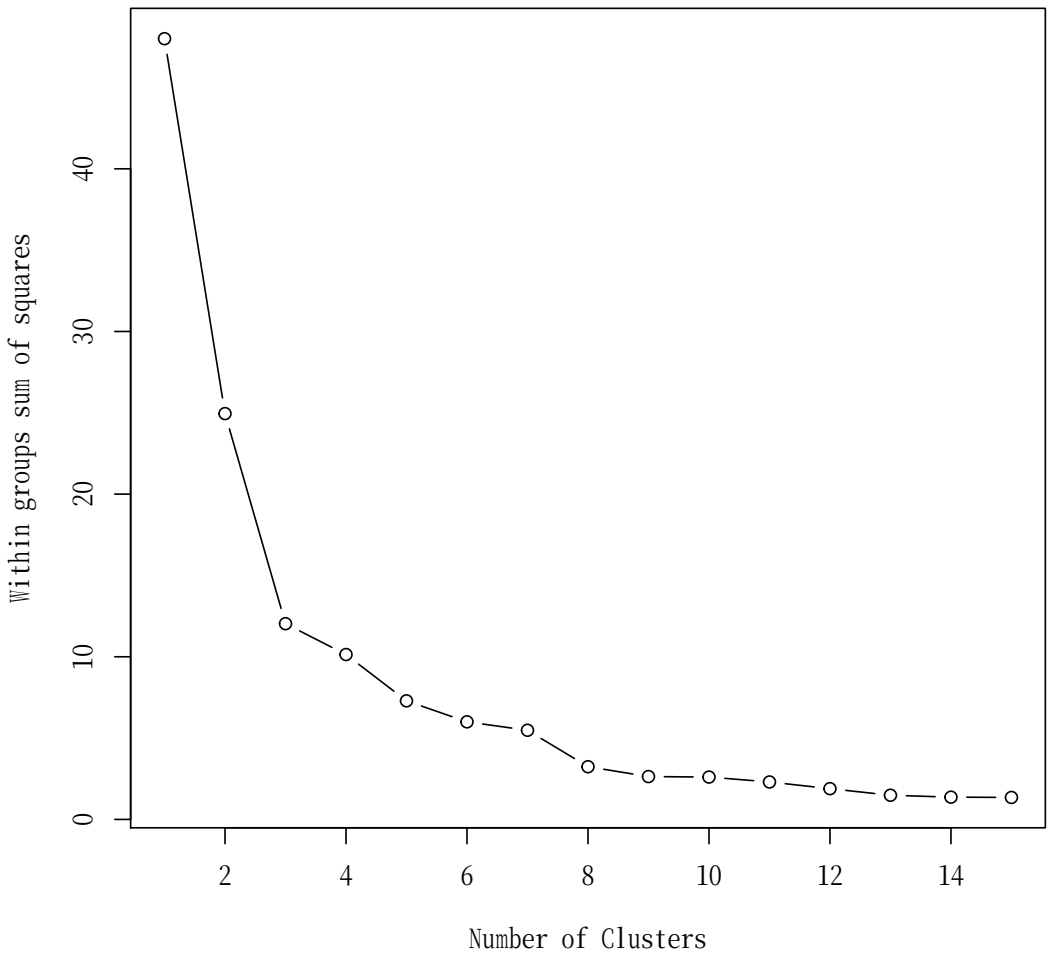
먼저, 처음 두 개의 특성(붉은색과 흰색 육류로 부터의 단백질 섭취)에 대해 군집화해보자. 이를 위해 25개국을 세 그룹으로 묶는다. 붉은 육류에서 단백질을 섭취하는 클러스터와 흰 육류에서 단백질을 섭취하는 클러스터의 스캐터 플롯을 생성한다. 국가별 k-means 클러스터 연합은 색으로 나타낸다. 이렇게 하면 세 개의 클러스터를 효과적으로 시각화할 수 있다. 클러스터는 각각의 클러스터 중심에 대한 유클리드 거리를 최소화함으로써 형성된다.

```
> setwd("~/Google Drive/furious lion/king/Big Mac/R Book/clustering/data")
> food <- read.csv("protein.csv")
> head(food)

         Country RedMeat WhiteMeat Eggs Milk Fish Cereals Starch Nuts
1        Albania    10.1       1.4  0.5  8.9  0.2    42.3    0.6  5.5
2        Austria     8.9      14.0  4.3 19.9  2.1    28.0    3.6  1.3
3        Belgium    13.5       9.3  4.1 17.5  4.5    26.6    5.7  2.1
4       Bulgaria     7.8       6.0  1.6  8.3  1.2    56.7    1.1  3.7
5 Czechoslovakia     9.7      11.4  2.8 12.5  2.0    34.3    5.0  1.1
6        Denmark    10.6      10.8  3.7 25.0  9.9    21.9    4.8  0.7
  Fr.Veg
1    1.7
2    4.3
3    4.0
4    4.2
5    4.0
6    2.4
```

Determine the optimal number of clusters

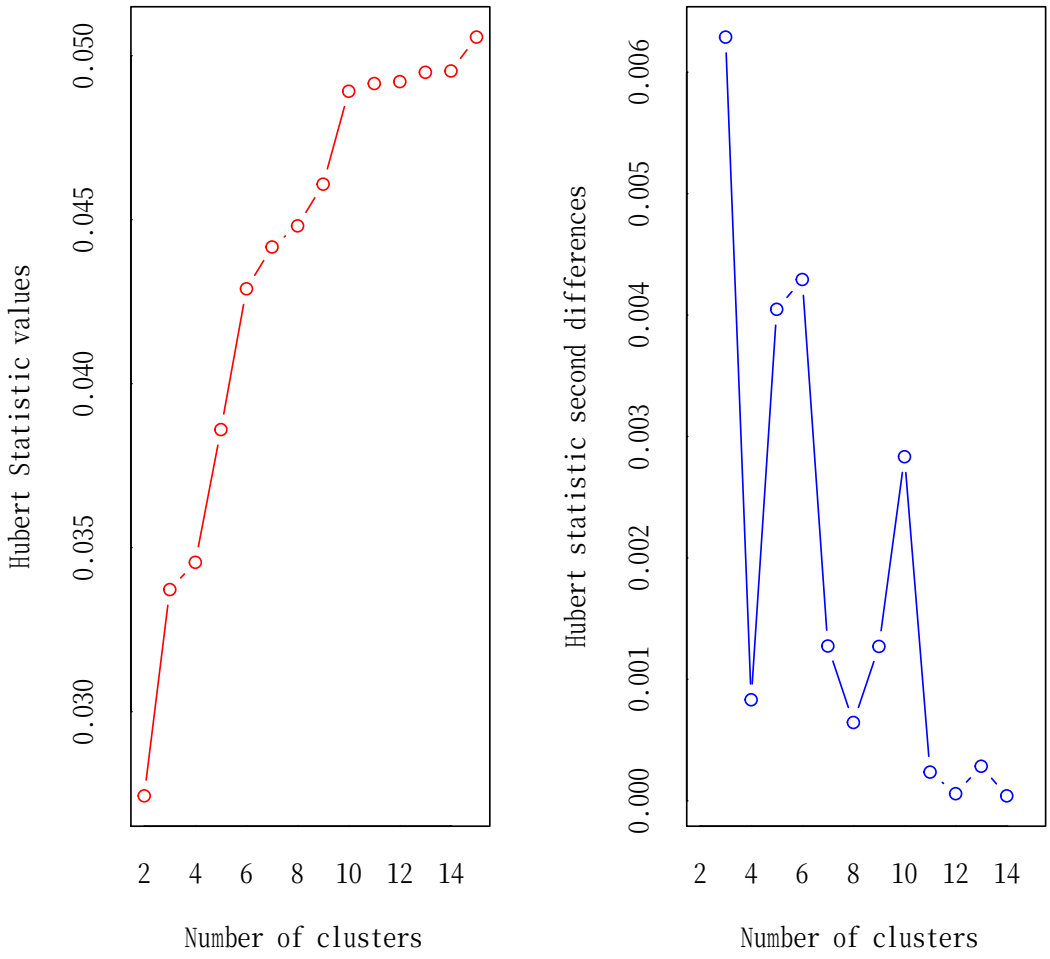
1. Scree plot

```
> source('~/Google Drive/furious lion/king/Big Mac/R script/my scripts/my functions/wssplot.R')
> df <- scale(food[, 2:3])
> wssplot(df)
```

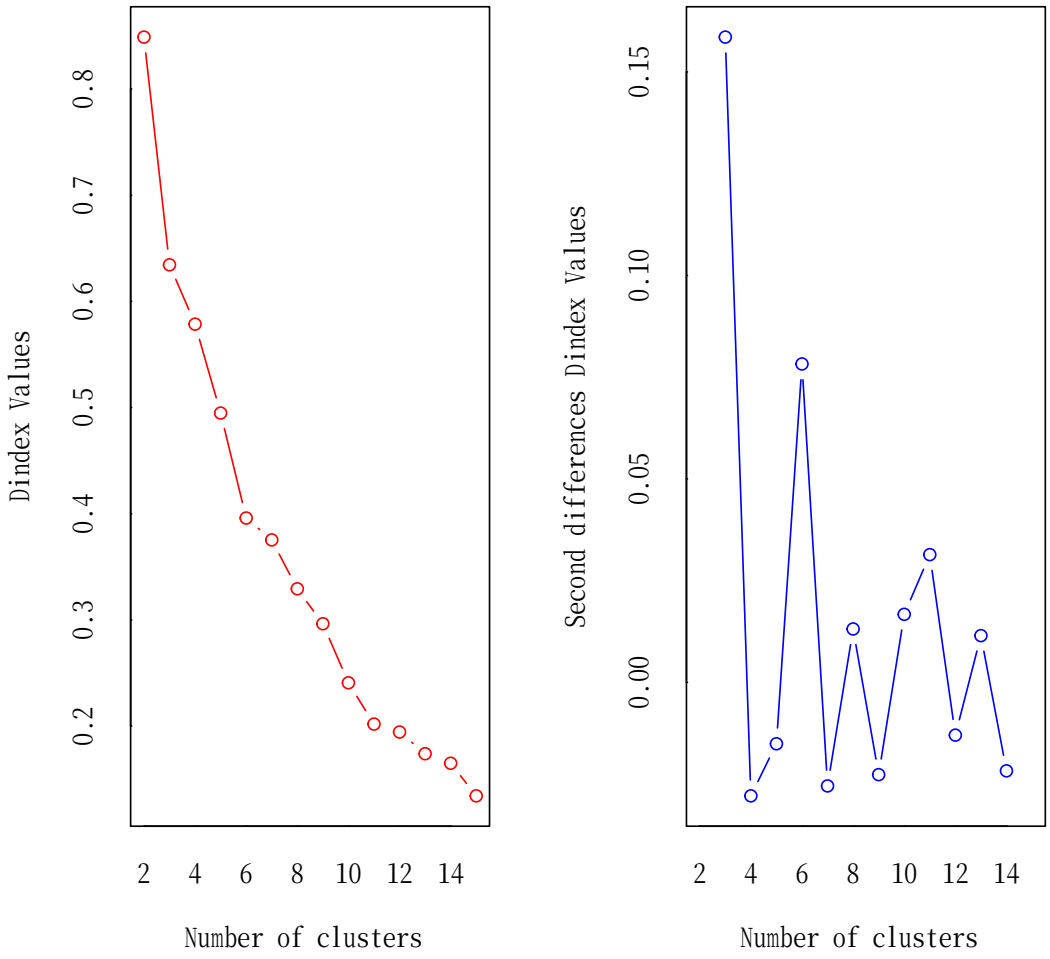


2. NbClust package

```
> library(NbClust)
> nc <- NbClust(df, min.nc=2, max.nc=15, method="kmeans")
```




```
*** : The Hubert index is a graphical method of determining the number of clusters.
                In the plot of Hubert index, we seek a significant knee that corresponds to a
                significant increase of the value of the measure i.e the significant peak in H
                index second differences plot.
```




```
*** : The D index is a graphical method of determining the number of clusters.
                In the plot of D index, we seek a significant knee (the significant peak in Dindex
                second differences plot) that corresponds to a significant increase of the value of
                the measure.

*******************************************************************
* Among all indices:
* 3 proposed 2 as the best number of clusters
* 7 proposed 3 as the best number of clusters
* 3 proposed 6 as the best number of clusters
```

```
* 1 proposed 8 as the best number of clusters
* 1 proposed 10 as the best number of clusters
* 2 proposed 11 as the best number of clusters
* 1 proposed 13 as the best number of clusters
* 1 proposed 14 as the best number of clusters
* 4 proposed 15 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  3


*******************************************************************
```

따라서,붉은색 육류와 흰색 육류(p = 2)에 대해 k = 3 클러스터 추출

```
> set.seed(2017)
> grpMeat <- kmeans(food[,c("WhiteMeat")], centers=3, nstart=25)
> # nstart=25 will generate 25 initial configurations.
> grpMeat

K-means clustering with 3 clusters of sizes 6, 12, 7

Cluster means:
       [,1]
1 12.583333
2  4.483333
3  9.728571

Clustering vector:
 [1] 2 1 3 2 1 3 1 2 3 2 1 3 2 1 2 3 2 2 2 3 3 2 2 1 2

Within cluster sum of squares by cluster:
[1]  5.448333 21.456667  5.514286
 (between_SS / total_SS =  90.1 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
```

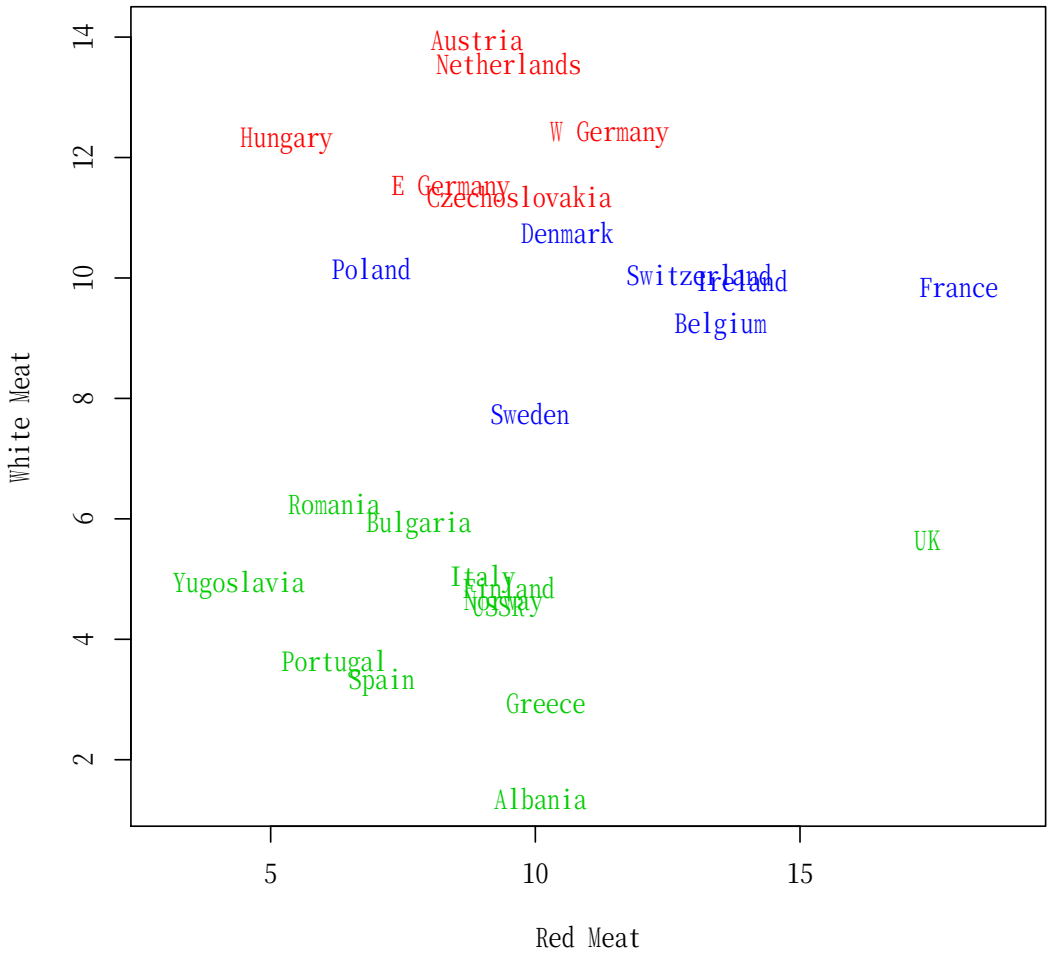
```
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

클러스터 할당 리스트하기

```
> o=order(grpMeat$cluster)
> data.frame(food$Country[o],grpMeat$cluster[o])

   food.Country.o. grpMeat.cluster.o.
1          Austria                  1
2   Czechoslovakia                  1
3        E Germany                  1
4          Hungary                  1
5      Netherlands                  1
6        W Germany                  1
7          Albania                  2
8         Bulgaria                  2
9          Finland                  2
10          Greece                  2
11           Italy                  2
12          Norway                  2
13        Portugal                  2
14         Romania                  2
15           Spain                  2
16              UK                  2
17            USSR                  2
18      Yugoslavia                  2
19         Belgium                  3
20         Denmark                  3
21          France                  3
22         Ireland                  3
23          Poland                  3
24          Sweden                  3
25     Switzerland                  3
```

plotting cluster assignments on Red and White meat scatter plot

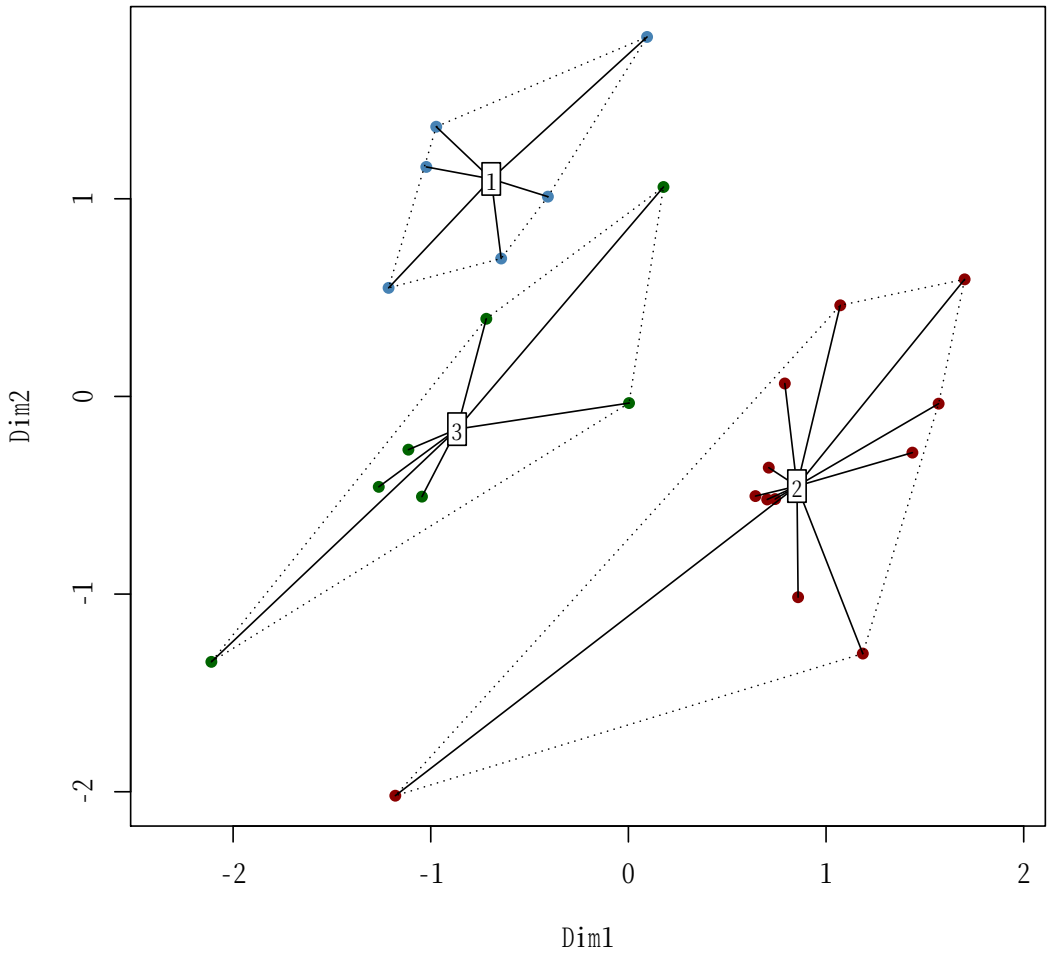
```
> plot(food$Red, food$White, type="n", xlim=c(3,19), xlab="Red Meat", ylab="White Meat")
> text(x=food$Red, y=food$White, labels=food$Country, col=grpMeat$cluster+1)
```



Plotting by "vegan" package

```
> require(vegan)
> food_dist <- dist(df) # distance matrix
> cmd <- cmdscale(food_dist) # Multidimensional scaling
> groups <- levels(factor(grpMeat$cluster)) # plot MDS, with colors by groups from kmeans
> ordiplot(cmd, type = "n")
> cols <- c("steelblue", "darkred", "darkgreen")
> for(i in seq_along(groups)){
+   points(cmd[factor(grpMeat$cluster) == groups[i], ], col = cols[i], pch = 16)
+ }
```

```
> # add spider and hull
> ordispider(cmd, factor(grpMeat$cluster), label = TRUE)
> ordihull(cmd, factor(grpMeat$cluster), lty = "dotted")
```



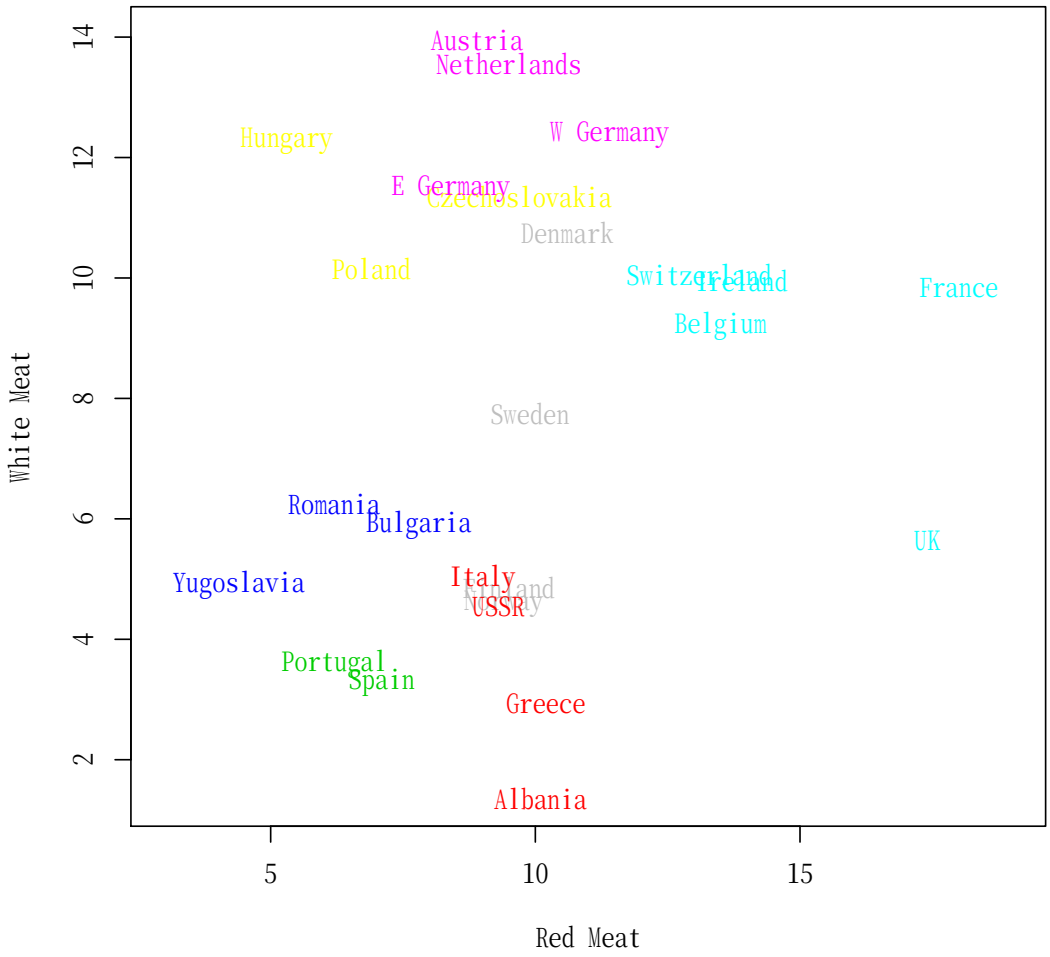
다음으로 모든 9개의 단백질 그룹을 군집화하고 7개의 클러스터를 만들어보자. 결과적으로 붉은 육류와 흰 육류의 산점도에 색상으로 표시되는 클러스터는 실제로 많은 의미를 가진다. 가까운 지리적 근접성을 가진 국가들은 같은 그룹으로 군집화 된다. 같은 분석이지만, 이제는 모든 단백질 그룹의 클러스터링으로 클러스터 수를 7개로 바꾸어 군집화 해보자.

```
> set.seed(2017)
> grpProtein <- kmeans(food[,-1], centers=7, nstart=25)
```

```
> grpProtein

K-means clustering with 7 clusters of sizes 4, 2, 3, 5, 4, 3, 4

Cluster means:
    RedMeat WhiteMeat     Eggs      Milk       Fish  Cereals   Starch
1  9.650000  3.525000 2.075000 14.200000  3.1250000 41.10000 2.825000
2  6.650000  3.550000 2.100000  6.750000 10.6000000 28.10000 5.800000
3  6.133333  5.766667 1.433333  9.633333  0.9333333 54.06667 2.400000
4 15.180000  9.000000 3.980000 21.440000  3.8000000 25.72000 4.840000
5  9.550000 12.925000 3.925000 18.300000  3.3500000 23.40000 4.875000
6  7.300000 11.333333 2.800000 13.833333  1.7666667 36.83333 4.966667
7  9.850000  7.050000 3.150000 26.675000  8.2250000 22.67500 4.550000
      Nuts   Fr.Veg
1 5.250000 4.450000
2 5.300000 7.550000
3 4.900000 3.400000
4 2.380000 4.320000
5 1.350000 3.850000
6 2.833333 4.933333
7 1.175000 2.125000

Clustering vector:
 [1] 1 5 4 3 6 7 5 7 4 1 6 4 1 5 7 6 2 3 2 7 4 4 1 5 3

Within cluster sum of squares by cluster:
[1] 150.48000  38.62000  47.00000 119.86400 148.48250  98.74667 131.77750
 (between_SS / total_SS =  86.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"

> o=order(grpProtein$cluster)
> data.frame(food$Country[o],grpProtein$cluster[o])

   food.Country.o. grpProtein.cluster.o.
```

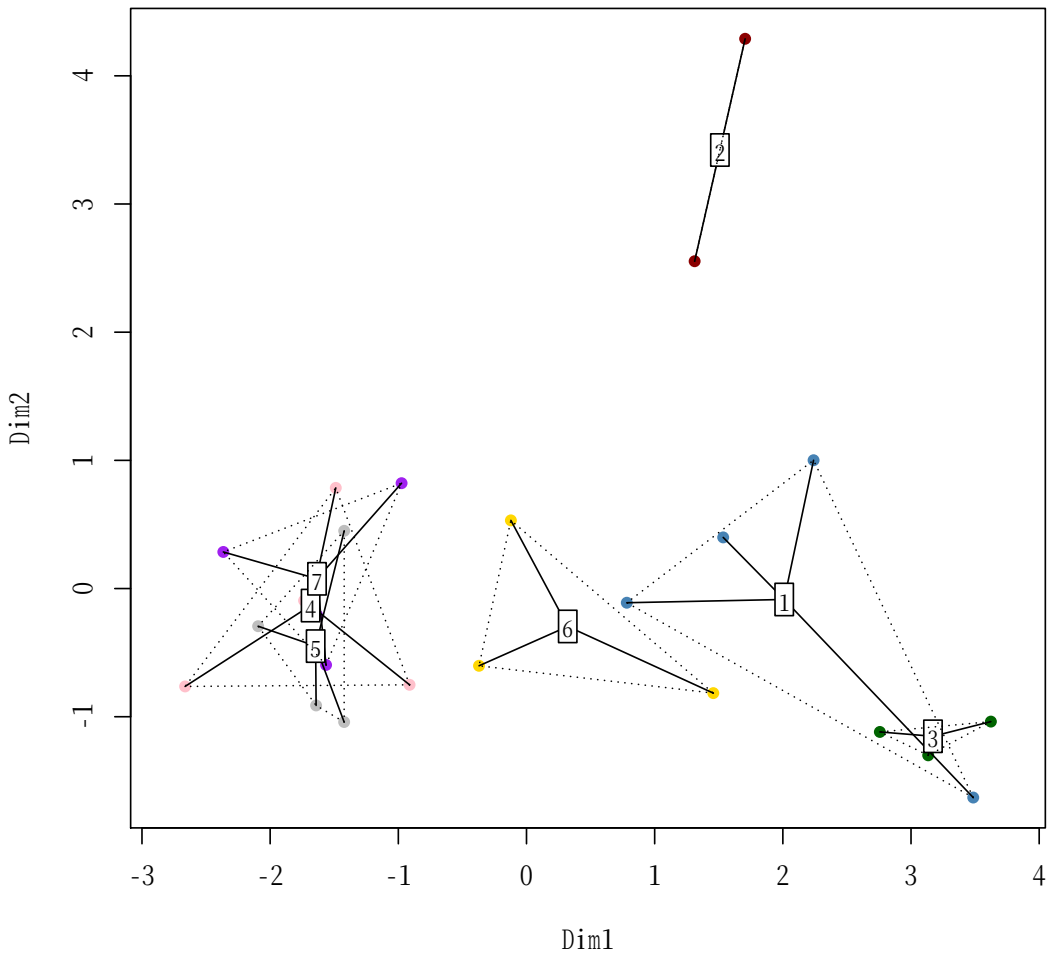
```
1          Albania                    1
2           Greece                    1
3            Italy                    1
4             USSR                    1
5         Portugal                    2
6            Spain                    2
7         Bulgaria                    3
8          Romania                    3
9       Yugoslavia                    3
10         Belgium                    4
11          France                    4
12         Ireland                    4
13     Switzerland                    4
14              UK                    4
15         Austria                    5
16       E Germany                    5
17     Netherlands                    5
18       W Germany                    5
19  Czechoslovakia                    6
20         Hungary                    6
21          Poland                    6
22         Denmark                    7
23         Finland                    7
24          Norway                    7
25          Sweden                    7
```

```
> plot(food$Red, food$White, type="n", xlim=c(3,19), xlab="Red Meat", ylab="White Meat")
> text(x=food$Red, y=food$White, labels=food$Country, col=grpProtein$cluster+1)
```



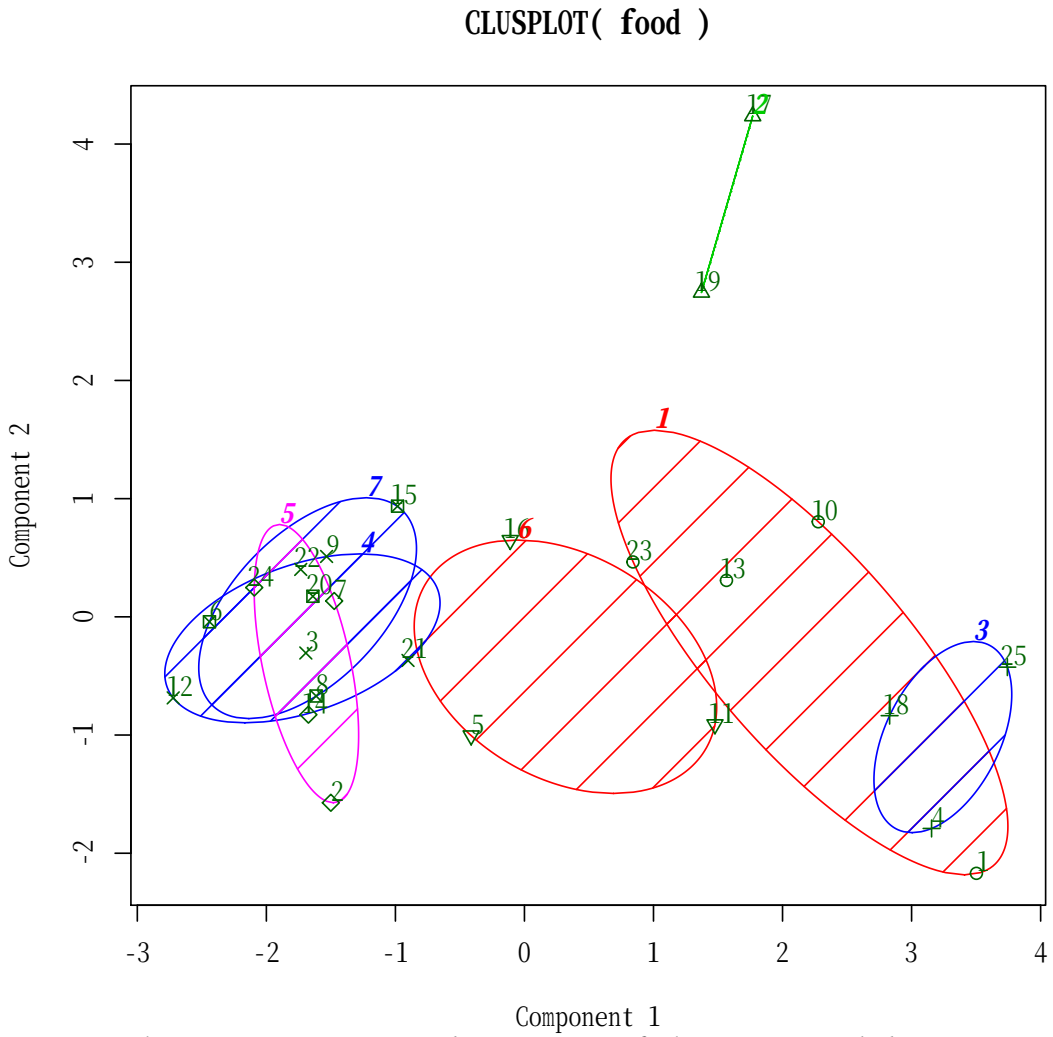

```
> food.scale <- scale(food[,-1])
> food_dist <- dist(food.scale) # distance matrix
> cmd <- cmdscale(food_dist) # Multidimensional scaling
> groups <- levels(factor(grpProtein$cluster)) # plot MDS, with colors by groups from kmeans
> ordiplot(cmd, type = "n")
> cols <- c("steelblue", "darkred", "darkgreen", "pink", "grey", "gold", "purple")
> for(i in seq_along(groups)){
+   points(cmd[factor(grpProtein$cluster) == groups[i], ], col = cols[i], pch = 16)
+ }
> # add spider and hull
```

```
> ordispider(cmd, factor(grpProtein$cluster), label = TRUE)
> ordihull(cmd, factor(grpProtein$cluster), lty = "dotted")
```



Plotting by using fpc package:

```
> library(fpc)
> library(cluster)
> clusplot(food, grpProtein$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```



예제 2: Customer Segmentation

```
> setwd("~/Google Drive/furious lion/king/Big Mac/R Book/clustering/data")
> offers<-read.csv("offers.csv", header=T)
> head(offers)

  OfferID Campaign           Varietal MinimumQt Discount      Origin
1       1  January             Malbec        72       56      France
2       2  January         Pinot Noir        72       17      France
```

```
3       3 February          Espumante       144       32      Oregon
4       4 February          Champagne        72       48      France
5       5 February Cabernet Sauvignon       144       44 New Zealand
6       6    March           Prosecco       144       86       Chile
  PastPeak
1    FALSE
2    FALSE
3     TRUE
4     TRUE
5     TRUE
6    FALSE

> transactions<-read.csv('transactions.csv', header=T)
> head(transactions)

  CustomerLastName OfferID
1            Smith       2
2            Smith      24
3          Johnson      17
4          Johnson      24
5          Johnson      26
6         Williams      18
```

Step 1: Organizing the information

분석을 위해 두 개의 데이터 세트가 있다: 하나는 오퍼에 대한 것이고 다른 하나는 트랜잭션에 대한 것이다. 먼저 해야 할 일은 트랜잭션 매트릭스를 만드는 것인데, 다시 말해 각 고객의 거래 내역 옆에 우송한 오퍼를 넣어야 한다. 이는 피벗 테이블을 사용하여 쉽게 수행할 수 있다.

```
> library(reshape)
> pivot <- melt(transactions[1:2])
> ## Using CustomerLastName as id variables
> pivot <- (cast(pivot, value~CustomerLastName,fill=0,
+                fun.aggregate=function(x) length(x)))
> pivot <- cbind(offers, pivot[-1])
> cluster.data <- pivot[,8:length(pivot)]
> cluster.data <- t(cluster.data)
> head(cluster.data)
```

```
         1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Adams    0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
Allen    0 0 0 0 0 0 0 0 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Anderson 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
Bailey   0 0 0 0 0 0 1 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Baker    0 0 0 0 0 0 1 0 0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
Barnes   0 0 0 0 0 0 0 0 0  1  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0
         26 27 28 29 30 31 32
Adams     0  0  0  1  1  0  0
Allen     0  1  0  0  0  0  0
Anderson  1  0  0  0  0  0  0
Bailey    0  0  0  0  1  0  0
Baker     0  0  0  0  0  1  0
Barnes    0  0  0  0  0  1  0
```

클러스터링 데이터 세트에서 행은 customers를 나타내고 열은 와인 브랜드/유형이다.

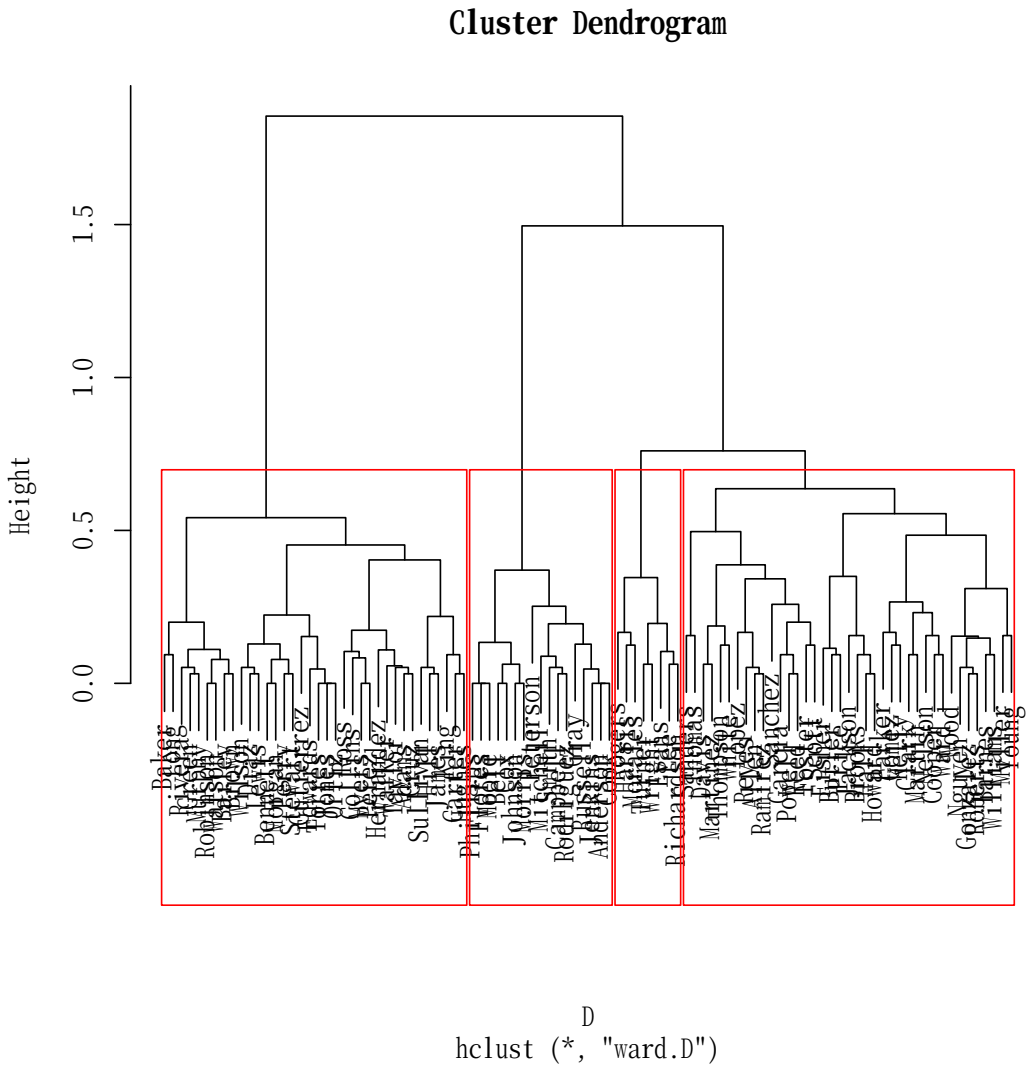
Step 2: Distances and Clusters

우리는 k = 4를 사용할 것이다. 이것은 다소 임의적이지만 선택할 클러스터의 수는 비즈니스로 처리할 수 있는 세그먼트 수를 나타내야 한다. 따라서 전자 메일 마케팅 캠페인에는 100개의 세그먼트를 모두 사용할 수 없지 않은가. 먼저 각 고객이 클러스터의 평균으로부터 얼마나 떨어져 있는지 계산해야 한다.

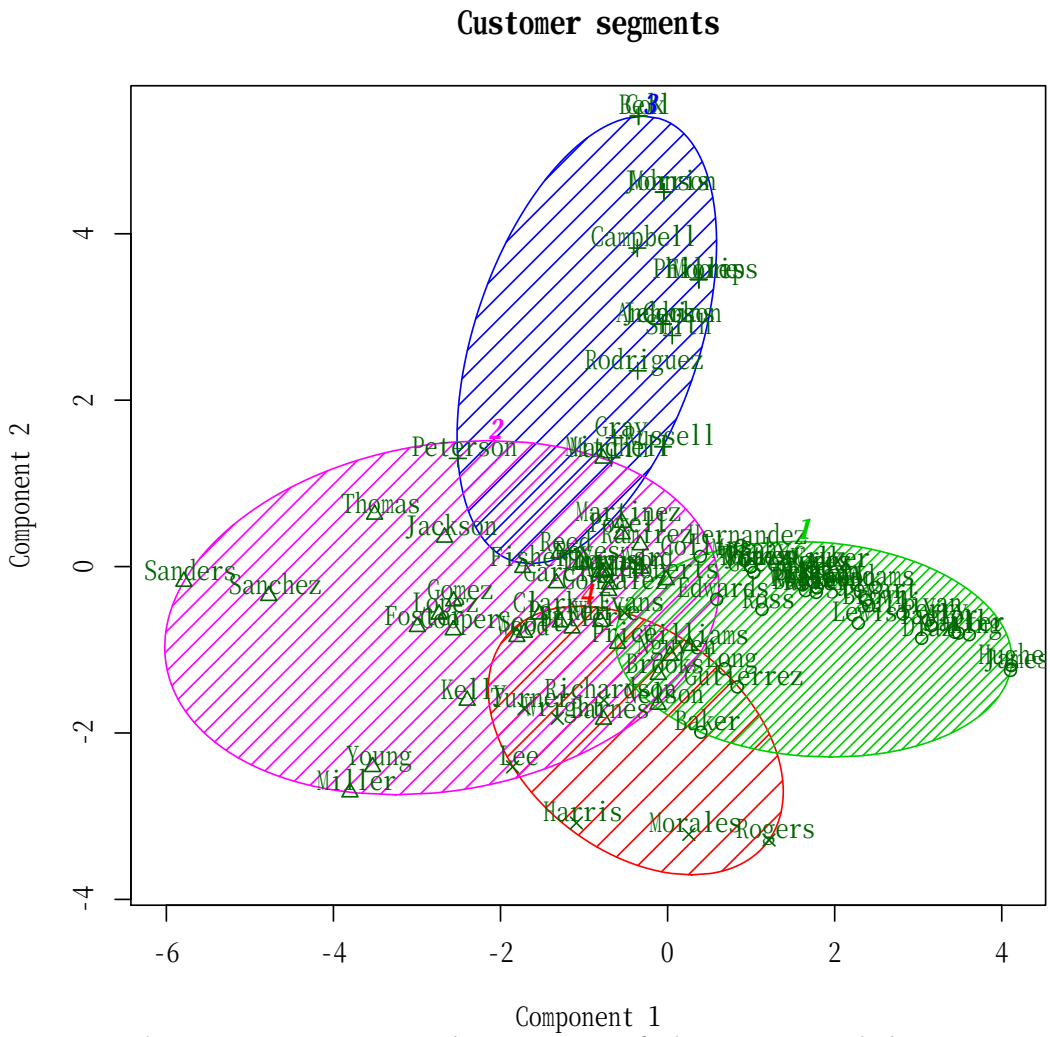
```
> library(cluster)
> D=daisy(cluster.data, metric='gower')
```

거리 행렬(distance matrix)을 생성한 후 Ward의 계층적 클러스터링(hierarchical clustering) 프로세스를 구현한다.

```
> H.fit <- hclust(D, method="ward.D")
> plot(H.fit) # display dendrogram
> groups <- cutree(H.fit, k=4) # cut tree into 4 clusters
> # draw dendogram with red borders around the 4 clusters
> rect.hclust(H.fit, k=4, border="red")
```




```
> # 2D representation of the Segmentation:
> clusplot(cluster.data, groups, color=TRUE, shade=TRUE,
+          labels=2, lines=0, main= 'Customer segments')
```



These two components explain 20.26 % of the point variability.

최대 거래를 파악하기 위해 약간의 데이터를 손 볼 필요가 있다. 먼저 클러스터와 트랜잭션을 결합해야 한다. 특히 트랜잭션과 클러스터의 테이블의 길이가 다르기 때문에 데이터를 병합할 필요가 있다. *merge()* 함수를 사용하여 열에 이름을 부여한다.

```
> # Merge Data
> cluster.deals<-merge(transactions[1:2],groups,by.x = "CustomerLastName",
+                      by.y = "row.names")
> colnames(cluster.deals)<-c("Name","Offer","Cluster")
> head(cluster.deals)

      Name Offer Cluster
1    Adams    18       1
```

```
2    Adams    29       1
3    Adams    30       1
4    Allen     9       2
5    Allen    27       2
6 Anderson    24       3
```

그런 다음 각 클러스터의 총 트랜잭션 수를 계산하기 위해 피벗 프로세스를 반복한다. 일단 피벗 테이블을 만들면 그것을 오퍼 데이터 테이블과 합친다.

```
> # Get top deals by cluster
> cluster.pivot <- melt(cluster.deals,id=c("Offer","Cluster"))
> cluster.pivot <- cast(cluster.pivot,Offer~Cluster,fun.aggregate=length)
> cluster.topDeals <- cbind(offers,cluster.pivot[-1])
> head(cluster.topDeals)

  OfferID Campaign           Varietal MinimumQt Discount      Origin
1       1  January             Malbec        72       56      France
2       2  January         Pinot Noir        72       17      France
3       3 February           Espumante       144       32      Oregon
4       4 February           Champagne        72       48      France
5       5 February Cabernet Sauvignon       144       44 New Zealand
6       6    March            Prosecco       144       86       Chile
  PastPeak 1 2 3 4
1    FALSE 0 8 2 0
2    FALSE 0 3 7 0
3     TRUE 1 2 0 3
4     TRUE 0 8 0 4
5     TRUE 0 4 0 0
6    FALSE 1 5 0 6

> #### And finally we can export the data in excel format with the command:
> #### write.csv(file="topdeals.csv", cluster.topDeals, row.names=F)
```

예제 3: Finding teen market segments using k-means clustering

전 세계적으로 Facebook이나 Instagram과 같은 소셜 네트워킹 사이트에서 친구들과 상호작용하는 것은 십대들에겐 너무나 자연스런 현상이다. 비교적 높은 구매력을 지닌 이 청소년들의 검색 정보는 스낵, 음료, 전자 제품, 위생 용품 등을 판매하는 사업자들에겐 유용한 정보가 된다.

이러한 사이트를 검색하는 수십만 명의 청소년들은 점차 경쟁이 치열해지는 시장에서 우위를

점하기 위해 애쓰고있는 마케터들의 관심을 끌기에 충분하다. 시장에서 우위를 점할 수 있는 한 가지 방법은 비슷한 취향을 공유하는 십대들의 세그먼트를 식별하여 청소년 고객이 원하지 않는 제품은 그들을 대상으로 광고를 하지 않는 것이다. 예를 들어, 스포츠 음료는 스포츠에 관심이 없는 10대들에게 광고하는 것은 의미가 없다.

청소년의 SNS 페이지의 텍스트를 통해 스포츠, 종교, 음악과 같은 공통 관심사를 공유하는 그룹을 식별할 수 있다. 클러스터링은 특정 모집단에서 자연적 세그먼트(natural segments)를 파악하는 프로세스를 자동화할 수 있다. 그러나 흥미로운 클러스터를 추출하는 것과 이 결과를 실제 마케팅에서 어떻게 사용할지는 결국 마케터의 역량이라는 것을 잊지 말자.

0.0.1 Step 1 – collecting data

이 분석을 위해 2006년 유명 SNS에서 프로필을 보유한 미국 고등학생 30,000명을 무작위로 추출한 데이터 세트를 사용한다. 자동화된 웹 크롤러를 사용하여 SNS 프로필의 전체 텍스트를 다운로드하고 각 청소년의 성별, 나이 및 SNS 친구 수를 기록했다. 텍스트 마이닝 도구를 사용하여 나머지 SNS 페이지 콘텐츠를 단어로 나누었다. 모든 페이지에 나타나는 상위 500 단어에서 36개의 단어가 과외 활동, 패션, 종교, 로맨스, 반사회적 행동과 같은 관심 분야를 나타내기 위해 선택되었다. 36단어는 축구, 섹시, 키스, 성경, 쇼핑, 죽음, 마약과 같은 용어를 포함한다. 최종 데이터 세트는 각 사람에 대해 그 사람의 SNS 프로필에 각 단어가 몇 번이나 등장했는지를 보여준다.

0.0.2 Step 2 – exploring and preparing the data

```
> setwd("~/Google Drive/furious lion/king/Big Mac/R Book/clustering/data")
> teens <- read.csv("snsdata.csv")
> str(teens)

'data.frame': 30000 obs. of  40 variables:
 $ gradyear    : int  2006 2006 2006 2006 2006 2006 2006 2006 2006 2006 ...
 $ gender      : Factor w/ 2 levels "F","M": 2 1 2 1 NA 1 1 2 1 1 ...
 $ age         : num  19 18.8 18.3 18.9 19 ...
 $ friends     : int  7 0 69 0 10 142 72 17 52 39 ...
 $ basketball  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ football    : int  0 1 1 0 0 0 0 0 0 0 ...
 $ soccer      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ softball    : int  0 0 0 0 0 0 0 1 0 0 ...
 $ volleyball  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ swimming    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ cheerleading: int  0 0 0 0 0 0 0 0 0 0 ...
 $ baseball    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ tennis      : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
 $ sports      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ cute        : int  0 1 0 1 0 0 0 0 0 1 ...
 $ sex         : int  0 0 0 0 1 1 0 2 0 0 ...
 $ sexy        : int  0 0 0 0 0 0 0 1 0 0 ...
 $ hot         : int  0 0 0 0 0 0 0 0 0 1 ...
 $ kissed      : int  0 0 0 0 5 0 0 0 0 0 ...
 $ dance       : int  1 0 0 0 1 0 0 0 0 0 ...
 $ band        : int  0 0 2 0 1 0 1 0 0 0 ...
 $ marching    : int  0 0 0 0 0 1 1 0 0 0 ...
 $ music       : int  0 2 1 0 3 2 0 1 0 1 ...
 $ rock        : int  0 2 0 1 0 0 0 1 0 1 ...
 $ god         : int  0 1 0 0 1 0 0 0 0 6 ...
 $ church      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ jesus       : int  0 0 0 0 0 0 0 0 0 2 ...
 $ bible       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hair        : int  0 6 0 0 1 0 0 0 0 1 ...
 $ dress       : int  0 4 0 0 0 1 0 0 0 0 ...
 $ blonde      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mall        : int  0 1 0 0 0 0 2 0 0 0 ...
 $ shopping    : int  0 0 0 0 0 2 1 0 0 0 1 ...
 $ clothes     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hollister   : int  0 0 0 0 0 0 0 2 0 0 0 ...
 $ abercrombie : int  0 0 0 0 0 0 0 0 0 0 ...
 $ die         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ death       : int  0 0 1 0 0 0 0 0 0 0 ...
 $ drunk       : int  0 0 0 0 0 1 1 0 0 0 0 ...
 $ drugs       : int  0 0 0 0 0 1 0 0 0 0 0 ...

> table(teens$gender, useNA = "ifany") #To include the NA values (if there are any), we simply need

    F     M  <NA>
22054  5222  2724

> summary(teens$age)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  3.086  16.310  17.290  17.990  18.260 106.900    5086
```

최소값과 최대값을 보면 3살 또는 106살이 나타난다. 그러나 3살 또는 106살이 고등학교에 다니는 일은 거의 없다. 이러한 극단치를 해결하기 위해서 데이터 클리닝을 해야 한다. 사실, 고등

학생의 적당한 연령대는 13세 이상 20세 미만이다. 이 범위를 벗어나는 연령값은 신뢰할 수 없으며, 따라서 누락된 데이터와 동일하게 취급하여야 한다. *ifelse()*함수를 사용하여 age변수를 다시 코딩하자. 즉, 나이가 13세 이상 20년 미만인 경우가 아니면 NA 값을 부여한다.

```
> teens$age <- ifelse(teens$age >= 13 & teens$age < 20, teens$age, NA)
> summary(teens$age)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  13.03   16.30   17.26   17.25   18.22   20.00    5523
```

성별과 같은 범주형 데이터의 결측치에 대한 대안은 결측치를 별도의 범주로 취급하는 것이다.

```
> teens$female <- ifelse(teens$gender == "F" & !is.na(teens$gender), 1, 0)
> teens$no_gender <- ifelse(is.na(teens$gender), 1, 0)
```

또한 연령 변수에는 5,523개의 결측치가 있다. 연령은 숫자 변수이므로 'unknown'으로 카테고리를 만드는 것이 의미가 없다. 대신, imputation을 사용하자.

```
> aggregate(data = teens, age ~ gradyear, mean, na.rm = TRUE)

  gradyear      age
1     2006 18.65586
2     2007 17.70617
3     2008 16.76770
4     2009 15.81957

> ave_age <- ave(teens$age, teens$gradyear, FUN =
+                   function(x) mean(x, na.rm = TRUE))
> # we can use the ave() function, which returns a vector with the group means repeated such
> teens$age <- ifelse(is.na(teens$age), ave_age, teens$age)
> summary(teens$age)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  13.03   16.28   17.24   17.24   18.21   20.00
```

0.0.3 Step 3 – training a model on the data

십대들의 SNS 프로필에 나타난 몇개의 관심사만을 고려하여 클러스터 분석을 해보자. 편의상 이들 관심사만 포함하는 데이터 프레임을 만들어 보자.

```
> interests <- teens[5:40]
```

거리 계산을 사용하여 분석을 수행하기 전에 표준화 또는 z-score로 표준화하여 각 관찰치를 동일한 스케일로 표준화한다. 사실, 어떤 관심사는 다른 관심사보다 큰 값을 갖을 수 있기 때문에, 이렇게 함으로써 일부 관심사가 너무 지배적으로 나타나는 문제를 피할 수 있다.

```
> interests_z <- as.data.frame(lapply(interests, scale))
```

분석을 하기 전에 적당한 클러스터를 결정해보자. 선행연구에 따르면, 청소년의 관심사는 5개의 패턴으로 구성된다. 따라서 5개의 클러스터를 사용하도록 한다. 물론 *NbClust* 패키지를 사용하여 클러스터의 수를 결정할 수도 있다. 그러나 큰 데이터에 이 방법을 사용하면 많은 메모리를 사용해야할 경우가 생긴다는 것을 잊지말자.

```
> set.seed(123)
> teen_clusters <- kmeans(interests_z, 5)
```

Step 4 – evaluating model performance

```
> teen_clusters$size

[1]  2583  4477 21263   990   687

> teen_clusters$centers

    basketball    football       soccer    softball  volleyball
1  1.366079417  1.17529316  0.476361860  1.20618908  1.11332011
2 -0.006342877  0.07044371  0.070298133 -0.06211143 -0.03378958
3 -0.187154596 -0.18680193 -0.079356506 -0.14162901 -0.13481522
4  0.366956761  0.38410687  0.139266026  0.12755805  0.09991111
5  0.168842670  0.35013806  0.006283266  0.06937640  0.06293187
     swimming cheerleading    baseball      tennis      sports       cute
1  0.08300583  -0.08312542  1.11304300  0.11564208  1.12464282 -0.0250709
2  0.31984029  -0.07834280 -0.06498358  0.12271196 -0.05672321  0.7425203
3 -0.09285629  -0.15200307 -0.13712560 -0.04384667 -0.16074457 -0.1843897
4  0.27537789   0.07744270  0.26159478  0.11438259  0.78356954  0.4901231
5  0.08071351   5.41605043  0.10577029 -0.04222980 -0.01284514  0.2561155
            sex         sexy           hot      kissed       dance
1 -0.0402756042 -0.004192134  0.0008016923 -0.09392197 -0.01172372
2  0.0030847772  0.259154797  0.5431886244 -0.01229063  0.61606628
```

```
3 -0.0945818898 -0.082923098 -0.1366202488 -0.13354891 -0.15462573
4  2.1224136296  0.562625023  0.3070830791  3.14496638  0.43189169
5  0.0001842334  0.082659861  0.2431118756  0.03458439  0.19270407
         band    marching       music        rock         god     church
1 -0.05808894 -0.05915918  0.05137051  0.12117303  0.02050123  0.1104702
2  0.31178917  0.23670165  0.36842838  0.16079934  0.39956811  0.5492738
3 -0.07841503 -0.04460347 -0.13874129 -0.10961423 -0.10842311 -0.1407903
4  0.46442486  0.08727753  1.20220515  1.26491434  0.41687142  0.1687260
5 -0.05571803 -0.06536590 -0.03241782  0.06633699  0.07405633  0.1195675
         jesus        bible         hair       dress      blonde
1  0.009671393 -0.015331112 -0.005461765 -0.07720504  0.03038766
2  0.322264538  0.287591989  0.362556940  0.60493309  0.02645834
3 -0.072995552 -0.062419592 -0.200722203 -0.14498022 -0.02885309
4  0.102543501  0.073445701  2.610594596  0.51050013  0.37030392
5 -0.024997356  0.009559632  0.108308415  0.09964249  0.07271788
         mall    shopping      clothes   hollister abercrombie         die
1 -0.01100531  0.02576077  0.003624177 -0.06415495 -0.06531511 -0.07843801
2  0.71776455  0.93294127  0.602627610  0.63973379  0.59918903  0.09219892
3 -0.18665726 -0.22878033 -0.188617303 -0.15441295 -0.14955061 -0.09070794
4  0.62747827  0.26986108  1.220278810  0.34322167  0.42038245  1.75771036
5  0.23680450  0.51539377  0.138527854  0.35679356  0.36369087 -0.03141111
         death       drunk       drugs
1 -0.030192694 -0.07102059 -0.09086939
2  0.176206267  0.04486769 -0.04432093
3 -0.077193727 -0.08606936 -0.10943422
4  0.934049802  1.84219543  2.80112648
5  0.008403465 -0.01616748 -0.01903205
```

결과의 행(1에서 5까지 번호가 매겨진)은 클러스터를 나타내고, 숫자는 열의 맨 위에 나열된 관심의 평균값을 나타낸다. 값은 표준화된 z점수이므로 음수 값은 모든 학생의 전체 평균보다 작은 값, 양수 값은 평균보다 높은 것을 의미한다. 예를 들어, 운동과 관련한 8가지 관심사로 클러스터의 특성을 추측해보자. 클러스터 4는 치어리딩이 약간 낮은 것을 제외한 모든 스포츠에서 평균 이상의 점수를 보였으며, 이 그룹에는 운동 선수가 포함될 수 있음을 암시한다. 클러스터 1은 치어 리딩을 가장 많이 언급하며 축구 관심도는 평균 이상이다. 이같은 해석을 통해 얻어진 최종 결론은 다음과 같다.

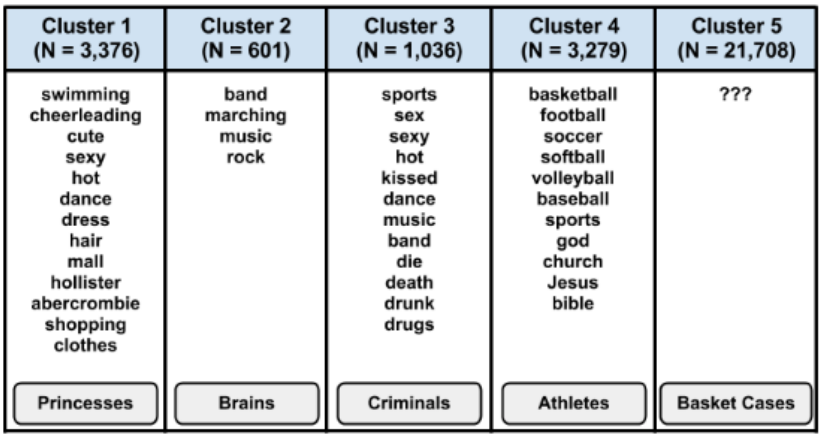


그림 1.1: 클러스터 정보(예)

0.0.4 Step 5 – improving model performance

k-means 클러스터가 수행되면 함수에는 샘플의 30,000명의 모든 사람에 대한 클러스터 할당이 포함된 *teens$cluster*라는 구성 요소가 저장된다. 다음 명령을 사용하여 이것을 원본 데이터 프레임에 열로 추가 할 수 있다.

```
> teens$cluster <- teen_clusters$cluster
> teens[1:5, c("cluster", "gender", "age", "friends")]

  cluster gender    age friends
1       3      M 18.982       7
2       2      F 18.801       0
3       3      M 18.335      69
4       3      F 18.875       0
5       4   <NA> 18.995      10
```

전에 사용했던 *aggregate()* 함수를 사용하여 클러스터 전체의 인구통계적 특성을 볼 수도 있다. 결과에 따르면 평균 연령이 클러스터에 따라 크게 다르지 않은 것으로 나타났다.

```
aggregate(data = teens, age ~ cluster, mean)

##   cluster      age
## 1       1 17.03618
## 2       2 17.10679
## 3       3 17.30350
## 4       4 17.11253
```

```
## 5       5 16.97600
```

반면에, 집단에 의한 여성의 비율에는 주목할만한 차이가 있다. 전체적으로 SNS 사용자의 약 74%가 여성이라는 점을 고려할 때, Cluster 5는 거의 약 91%가 여성이며, Cluster 1과 3은 여성이 약 70%에 불과하다.

```
> aggregate(data = teens, female ~ cluster, mean)

  cluster    female
1       1 0.6879597
2       2 0.8767031
3       3 0.7024879
4       4 0.7959596
5       5 0.9126638
```

또한 클러스터에 따라 SNS 사용자의 친구 수를 예측할 수도 있다.

```
> aggregate(data = teens, friends ~ cluster, mean)

  cluster  friends
1       1 35.25474
2       2 37.50681
3       3 27.68664
4       4 31.07576
5       5 39.20961
```