2강 세그멘테이션
(다차원 척도법, 군집분석)

서울대학교 통계학과

김용대 교수

1. 다차원척도법
Multidimensional Scaling (MDS)

1.1 다차원척도법 개요

1.1 다차원척도법 개요

다차원 척도법 Multidimensional Scaling (MDS)

다차원 관측값 또는 개체들 간의 거리(distance) 또는 비유사성(dissimilarity)을 이용하여 개체들을 원래의 차원보다 낮은 차원(2차원 또는 3차원)의 공간상의 점으로 표현(spatial configuration)하는 통계적 분석방법

목적 → 차원의 축소를 통해 개체들 사이의 관계를 쉽게 파악

예) 정치 후보자, 소비자 제품들의 성향에 대한 구조를 파악하고자 할 때, 이들 개체들의 특성을 측정한 후에 개체들의 거리 또는 비유사성을 구한 뒤, 이들 개체들을 2차원 또는 3차원 공간상에 표현하여 개체들 사이의 관계를 파악하는데 이용

1.1 다차원척도법 개요

군집분석과 다차원척도법

군집분석 개체들 간의 비유사성을 이용하여 동일한 그룹들로 분류하고자 하는 분석방법

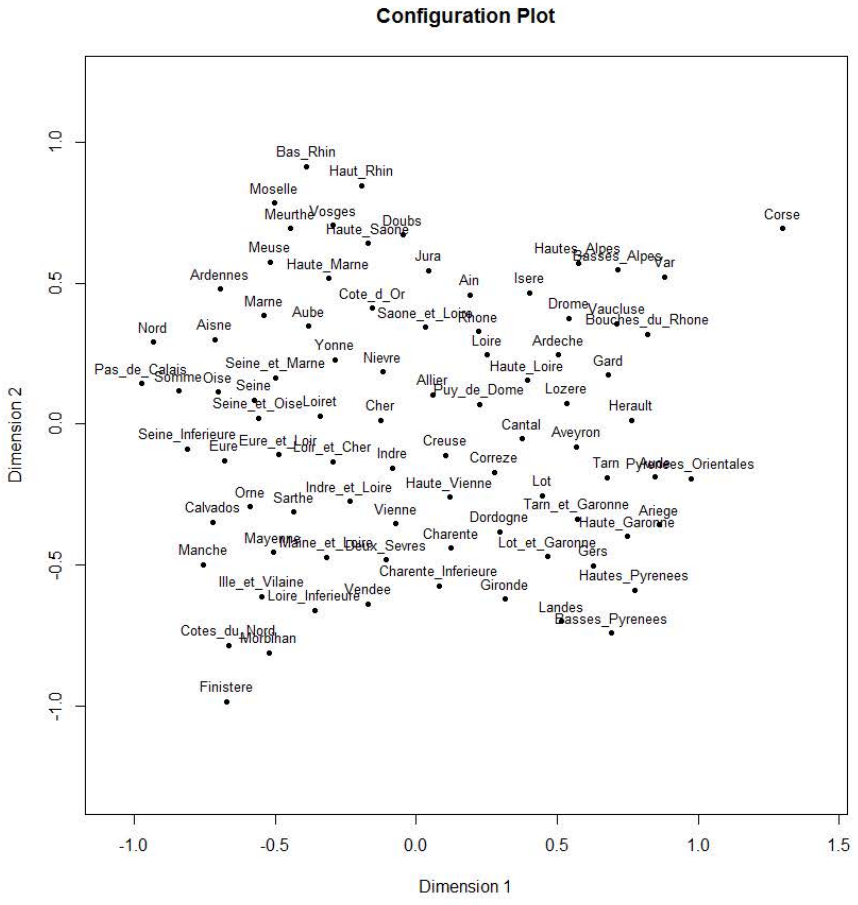
다차원척도법 개체들의 비유사성을 이용하여 공간상에 표시함으로써 개체들 간의 상대적인 위치를 표시하고, 이를 이용 유사한 개체들을 파악하는 방법

1.1 다차원척도법 개요

예) 1830년도의 프랑스 내의 86개 정부부처들간의 비행거리에 대한 자료가 다음과 같이 주어졌다고 하자. 이를 이용하여 MDS를 이용한 거리지도를 그려보면 다음과 같다.

```
> library(smacof)
> library(RgoogleMaps)
> data(Guerry)
> head(Guerry)
                  Ain   Aisne   Allier Basses_Alpes Hautes_Alpes  Ardeche Ardennes
Ain            0.0000 236.127  97.4827     136.4560     102.8580  96.9111 227.8940
Aisne        236.1270   0.000 204.5250     372.5810     338.5050 312.9950  46.0778
Allier        97.4827 204.525   0.0000     203.8090     179.4800 120.2220 216.0980
Basses_Alpes 136.4560 372.581 203.8090       0.0000      36.5306  94.5906 362.8050
Hautes_Alpes 102.8580 338.505 179.4800      36.5306       0.0000  85.8828 327.3240
Ardeche       96.9111 312.995 120.2220      94.5906      85.8828   0.0000 313.6630

> fit.guerry <- mds(Guerry)
> op <- par(mfrow = c(1,2))
> plot(fit.guerry)
> theta <- 82*pi/180 ## degrees to radians
> rot <- matrix(c(cos(theta), sin(theta), -sin(theta), cos(theta)), ncol = 2)
> configs82 <- fit.guerry$conf %*% rot ## rotated configurations
> francemap1 <- GetMap(destfile="mypic1.png", zoom = 6, center = c(46.55, 3.05),
+ maptype = "satellite")
> PlotOnStaticMap(francemap1)
> text(configs82*280, labels = rownames(configs82), col = "white", cex = 0.7)
> par(op)
```

1.1 다차원척도법 개요


Configuration Plot
Dimension 1
Dimension 2
-1.0
-0.5
0.0
0.5
1.0
1.5
Bas_Rhin
Haut_Rhin
Moselle
Meurthe
Vosges
Haute_Saone
Doubs
Corse
Meuse
Haute_Marne
Jura
Hautes_Alpes
Basses_Alpes
Var
Ardennes
Ain
Isere
Marne
Cote_d_Or
Drome
Vaucluse
Aube
Saone_et_Loire
Bouches_du_Rhone
Nord
Aisne
Rhone
Loire
Ardeche
Yonne
Nievre
Seine_et_Marne
Haute_Loire
Gard
Pas_de_Calais
Somme
Oise
Seine
Allier
Puy_de_Dome
Lozere
Seine_et_Oise
Loiret
Cher
Herault
Cantal
Aveyron
Seine_Inferieure
Eure
Eure_et_Loir
Loir_et_Cher
Indre
Creuse
Correze
Tarn
Pyrenees_Orientales
Orne
Sarthe
Indre_et_Loire
Haute_Vienne
Lot
Calvados
Vienne
Tarn_et_Garonne
Ariege
Dordogne
Haute_Garonne
Mayenne
Maine_et_Loire
Deux_Sevres
Charente
Lot_et_Garonne
Manche
Gers
Charente_Inferieure
Hautes_Pyrenees
Ille_et_Vilaine
Vendee
Gironde
Loire_Inferieure
Landes
Basses_Pyrenees
Cotes_du_Nord
Morbihan
Finistere


Pas_de_Calais
Nord
Somme
Seine_Inferieure
Aisne
Ardennes
Oise
Manche
Calvados
Eure
Seine
Marne
Meuse
Moselle
Cotes_du_Nord
Orne
Seine_et_Oise
Finistere
Seine_et_Marne
Meurthe
Bas_Rhin
Ille_et_Vilaine
Mayenne
Eure_et_Loir
Aube
Morbihan
Sarthe
Haute_Marne
Vosges
Loiret
Yonne
Haut_Rhin
Loire_Inferieure
Maine_et_Loire
Loir_et_Cher
Haute_Saone
Indre_et_Loire
Cote_d_Or
Cher
Nievre
Doubs
Vendee
Indre
Deux_Sevres
Vienne
Saone_et_Loire
Jura
Allier
Creuse
Ain
Charente_Inferieure
Haute_Vienne
Rhone
Charente
Puy_de_Dome
Loire
Correze
Isere
Dordogne
Haute_Loire
Cantal
Gironde
Ardeche
Lot
Drome
Hautes_Alpes
Lot_et_Garonne
Lozere
Aveyron
Landes
Tarn_et_Garonne
Basses_Alpes
Gard
Vaucluse
Gers
Tarn
Herault
Bouches_du_Rhone
Basses_Pyrenees
Haute_Garonne
Var
Hautes_Pyrenees
Aude
Ariege
Pyrenees_Orientales
Google
Imagery ©2016 TerraMetrics

1.1 다차원척도법 개요

MDS 구분 (자료 특성에 따라)

1. **메트릭 MDS (metric MDS)** : 등간척도나 비율척도 자료에 근거하여 비유사성 이루어지는 경우
2. **넌메드릭 MDS (nonmetric MDS)** : 순서척도 자료에 근거하여 비유사성 측정 되는 경우

적합성

Kruskal의 STRESS or S-STRESS

: 공간상의 표현이 주어진 비유사성에 어느 정도 적합한가를 측정하는 기준

1.2 메트릭 MDS와 넌메트릭 MDS

최적모형의 적합

✓ **부적합도** : STRESS or S-STRESS 이용 각 개체들을 공간상에 표현

$$STRESS = \sqrt{\frac{\sum_{i<j}\left(S_{ij} - \widehat{S_{ij}}\right)^2}{\sum_{i<j} S_{ij}^2}}$$

$$S - STRESS = \sqrt{\frac{\sum_{i<j}\left(S_{ij} - \widehat{S_{ij}}\right)^2}{\sum_{i<j}\left(S_{ij}^2\right)^2}}$$

$\widehat{S_{ij}}$: 측정 모형에서 구한 S_{ij} 의 적합값

✓ 부적합 도를 최소로 하는 방법으로 반복알고리즘을 이용하게 적합

✓ 부적합도 값 일정한 수준이하로 될 때 최종적으로 적합된 모형으로 제시

1.2 메트릭 MDS와 넌메트릭 MDS

최적모형의 적합

✓ 부적합도 값은 0과 1 사이의 값을 취함
0 으로 작아질수록 적합된 모형이 적절하다고 판단

Stress	적합도 수준
0	완벽 (Perfect)
0.05 이내	매우 좋음 (Excellent)
0.05 – 0.10	만족 (Satisfactory)
0.10 – 0.15	보통 (Acceptable, but doubt)
0.15 이상	나쁨 (Poor)

⇒ STRESS≥0.10 : STRESS의 크기가 적정 수준이 될 때까지 차원을 높임

✓ STRESS는 표현 공간이 커질수록 작아짐 그러나 표현공간이 클수록 결과의 해석이 복잡
⇒ 일반적으로 2차원 또는 3차원 정도가 이용

1.3 다차원 척도 분석 실습

1.3 다차원 척도 분석 실습

2일차 4.3 인자분석의 실습에서 이용하였던 시리얼 자료(cereal.csv) 를 이용하여 다차원 척도분석을 실시해보고자 한다. 자료의 정리작업은 4.3과 동일하며 74개의 관측치와 9개의 변수로 이루어진 자료를 만든다.

```
> cereal=read.csv("C:\\Users\\Sunghyun Sun Cho\\Desktop\\Insight\\cereals.csv")
> cereal=cereal[,c("name","calories","protein","fat","sodium","fiber","carbo","sugars","potass","vitamins")]
> cereal[!complete.cases(cereal),]
                    name calories protein fat sodium fiber carbo sugars potass vitamins
5          Almond_Delight      110       2   2    200   1.0    14       8      NA       25
21 Cream_of_Wheat_(Quick)      100       3   0     80   1.0    21       0      NA        0
58         Quaker_Oatmeal      100       5   2      0   2.7    NA      NA     110        0
> cereal=cereal[-c(5,21,58),]
> rownames(cereal)=cereal["name"]
> cereal=cereal[,-1]
> head(cereal)
                         calories protein fat sodium fiber carbo sugars potass vitamins
100%_Bran                      70       4   1    130  10.0   5.0       6    280       25
100%_Natural_Bran             120       3   5     15   2.0   8.0       8    135        0
All-Bran                       70       4   1    260   9.0   7.0       5    320       25
All-Bran_with_Extra_Fiber      50       4   0    140  14.0   8.0       0    330       25
Apple_Cinnamon_Cheerios       110       2   2    180   1.5  10.5      10     70       25
Apple_Jacks                   110       2   0    125   1.0  11.0      14     30       25
```

1.3 다차원 척도 분석 실습

(1) 다차원척도분석 모형 적합

cmdscale(d, k= , eig= , add=)

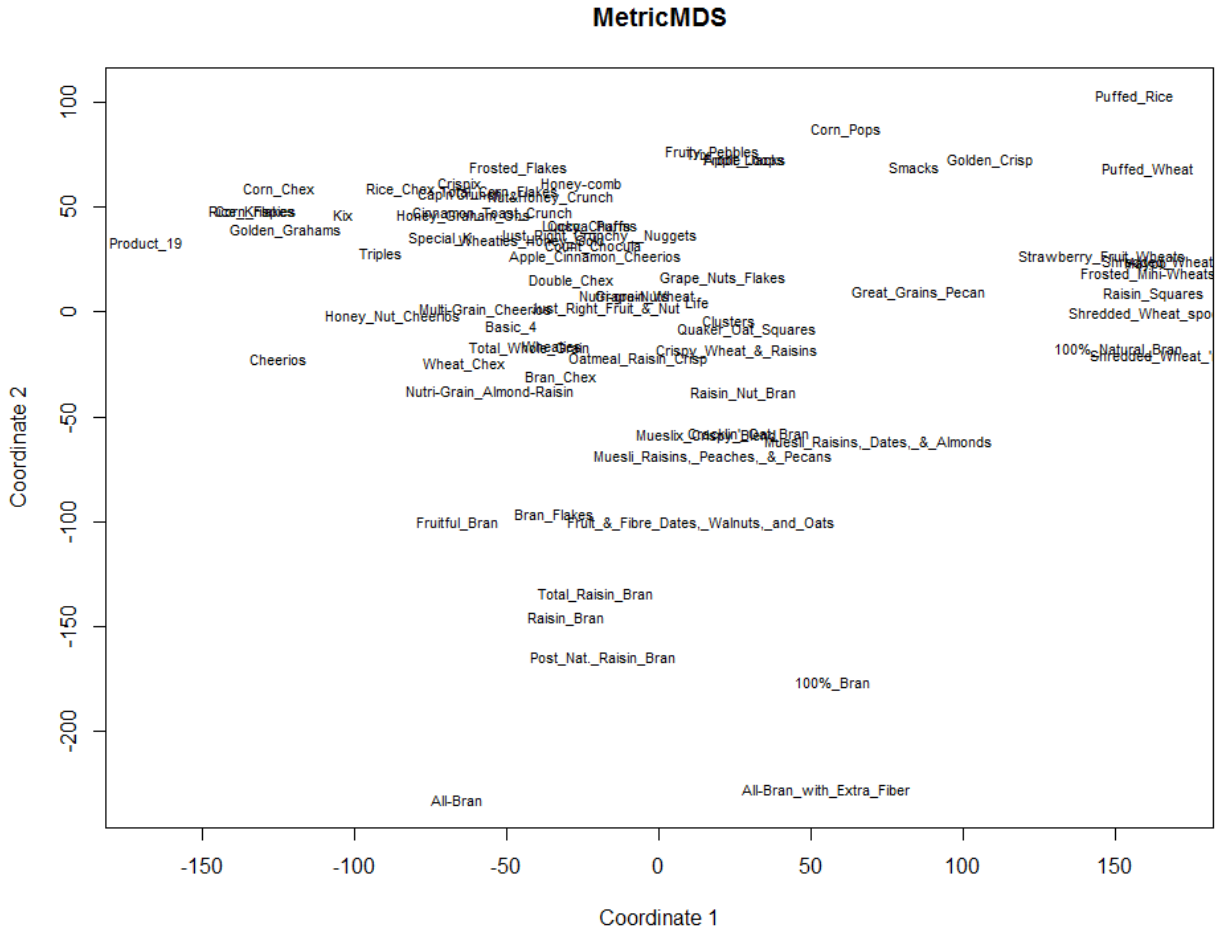
d : dist 자료 (대칭형태의 비유사성 자료)
k : 자료의 형태를 표현하고자하는 최대 차원
eig : 고유값의 출력 여부
add : 추가적인 상수의 사용여부
(TRUE인 경우, 수정된 비유사성 자료를 만들어 n-1 차원으로 낮춤)

```
> cereal.d <- dist(cereal) # euclidean distances between the rows
> fit <- cmdscale(cereal.d,eig=TRUE, k=2, add=TRUE) # k is the number of dim
> fit # view results
$points
                                 [,1]        [,2]
100%_Bran                  57.445184 -176.676670
100%_Natural_Bran         151.459613  -17.697962
All-Bran                  -65.746679 -232.202850
All-Bran_with_Extra_Fiber  55.208483 -227.718276
Apple_Cinnamon_Cheerios   -20.630234   26.149925
Apple_Jacks                28.636167   72.581875
Basic_4                   -48.164938   -6.998693
Bran_Chex                 -32.144004  -31.003491
Bran_Flakes               -34.018209  -96.756330
Cap'n'Crunch              -65.099984   56.121492
```

1.3 다차원 척도 분석 실습

(2) 모형결과

```
x <- fit$points[,1]
y <- fit$points[,2]
plot(x, y, xlab="Coordinate 1", ylab="Coordinate 2", main="Metric     MDS",    type="n")
text(x, y, labels = row.names(cereal), cex=.7)
```



1.3 다차원 척도 분석 실습

직접해보기 (RockHard [smacof package])

다음의 자료는 2013년도에 독일의 헤비메탈 잡지인 RockHard Magazine에서 수행한 설문조사의 결과이다. 자료는 매 달 약 50개 내외의 밴드들을 대상으로 14명의 기자들이 점수를 0에서부터 10까지의 부여하였다. 데이터를 불러오는 방법은 다음과 같다.

```
library(smacof)
data(RockHard)
```

```
> head(RockHard)
  Year Month              Band                     Album Gotz Thomas Frank
1 2013     1             Attic           The Invocation  8.5    8.0   8.0
2 2013     1          Paradox       Tales Of The Weird  7.5    7.0   8.0
3 2013     1              Zuul        To The Frontlines  8.0    7.0   7.5
4 2013     1 Chapel Of Disease      Summoning Black Gods  8.0    7.5   8.0
5 2013     1  Dropkick Murphys Signed And Sealed In Blood  7.0    7.5   7.5
6 2013     1          Saturnus       Saturn In Ascension  6.0    7.0   6.5
  Bjorn Jan Boris Himmelstein Michael Jens Ronny Felix Jakob Marcus Jenny
1   9.0 7.0   8.5         8.5     7.0   NA    NA    NA   8.5    7.5   7.0
2   9.0 7.0   7.5         7.0     6.5   NA    NA    NA   7.5    8.0   7.5
3   8.5 7.0   8.5         8.0     6.5   NA    NA    NA   8.0    7.0   6.5
4   8.0 7.5   8.0         8.0     6.5   NA    NA    NA   8.0    6.5   6.0
5   6.0 6.5   7.0         8.5     8.5   NA    NA    NA   7.5    7.0   6.5
6   8.0 7.0   7.5         7.0     6.0   NA    NA    NA   8.5    7.5   7.0
```

1.3 다차원 척도 분석 실습

1. 결측치가 있는 경우에는 cmdscale() 의 이용이 어렵다. 따라서 결측치가 있는 기자의 자료는 제외하고 cmdscale()을 이용하여 차원이 2인 메트릭 다차원 척도 분석을 수행하고 결과를 그림으로 표현하시오.

2. (선택) smacof 패키지 안에있는 새로운 함수인 mds()를 이용하면 결측치가 있는 자료도 mds를 수행할 수가 있게된다. 이 함수를 이용하여 결측치를 포함한 전체 자료에 대해서 다차원 척도분석을 수행하고 결과를 그림으로 표현하시오.

2. 군집분석
Cluster Analysis

2.1 군집분석의 정의

2.1 군집분석의 정의

군집분석이란?

군집분석은 모집단 또는 범주에 대한 사전 정보가 없는 경우에
주어진 관측값들 사이의 거리 또는 유사성을 이용하여
전체를 몇 개의 집단으로 그룹화하여
각 집단의 성격을 파악함으로써
데이터 전체의 구조에 대한 이해를 돕고자 하는 분석법이다.

2.1 군집분석의 정의

군집화

1) 군집화의 기준

동일한 군집에 속하는 개체 (또는 개인) 는 여러 속성이 비슷하고 서로 다른 군집에 속한 관찰치는 그렇지 않도록 군집을 구성

2) 군집화를 위한 변수 : 전체 개체(개인)의 속성을 판단하기 위한 기준

예) 고객세분화

인구통계적 변인 (성별, 나이, 거주지, 직업, 소득, 교육, 종교, …)

구매패턴 변인 (상품, 주기, 거래액, …)

생활패턴 변인 (라이프스타일, 성격, 취미, 가치관, …)

2.2 군집분석의 활용

2.2 군집분석의 활용

고객세분화

고객이 기업의 수익에 기여하는 정도를 통한 고객 세분화

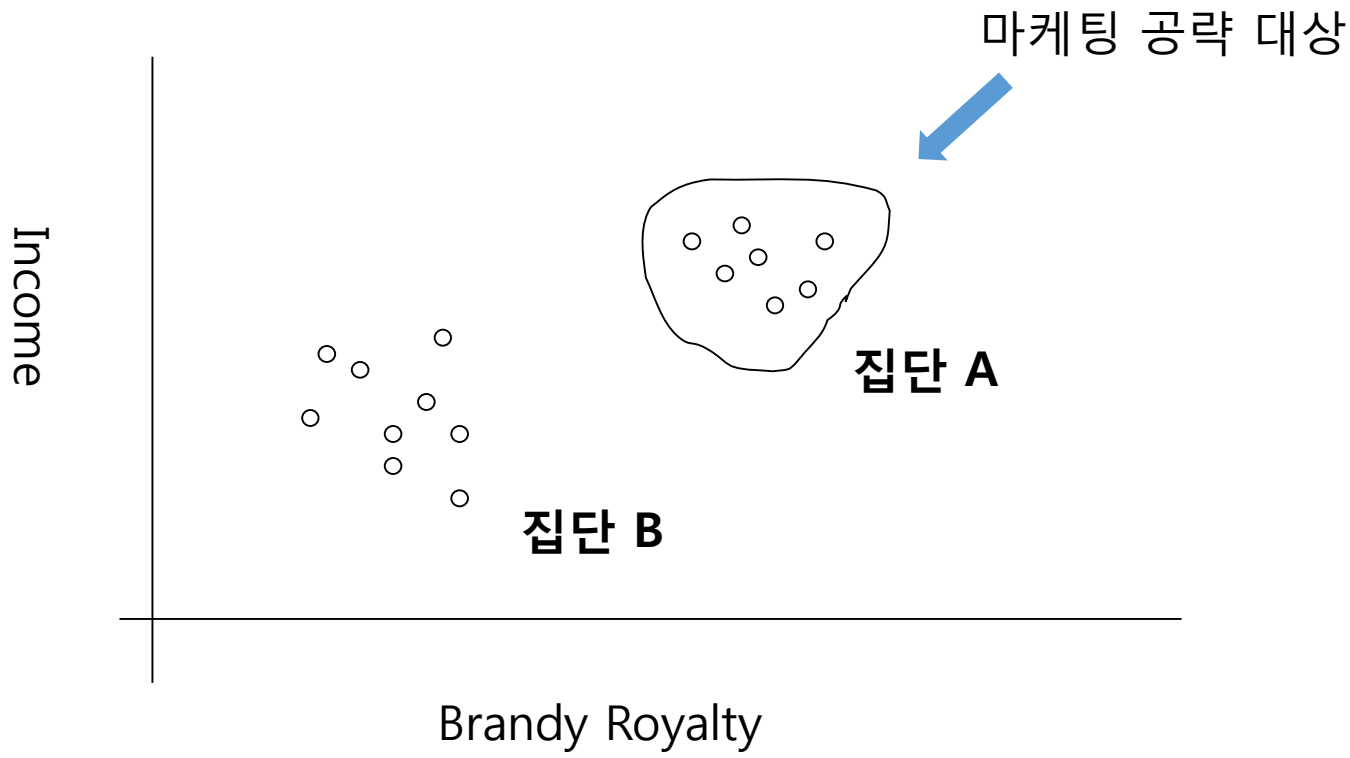
=> 우수고객의 인구통계적 요인, 생활패턴 파악
=> 개별고객에 대한 맞춤 관리

고객의 구매패턴에 따른 고객세분화

=> 신상품 판촉, 교차판매를 위한 표적집단 구성

2.2 군집분석의 활용

2.2 군집분석의 활용



2.3 비유사성의 척도 : 거리

2.3 비유사성의 척도 : 거리

비유사성의 척도 : 거리

군집분석에서는 관측값들이 서로 얼마나 유사한지 또는 유사하지 않은지를 측정할 수 있는 측도가 필요하다.

군집분석에서는 보통 유사성(similarity)보다는 비유사성(dissimilarity)를 기준으로 하며 거리(distance)를 사용한다.

거리의 정의 : 두 점 x와 y의 거리 $d(x,y)$ 는 다음을 만족한다

$$d(x,y) = \geq x = y$$
$$d(x,y) \geq 0$$
$$d(x,y) = d(y,x)$$
$$d(x,y) \leq d(x,z) + d(z,y) \text{ (triangular inequality)}$$

2.3 비유사성의 척도 : 거리

거리 측도의 종류들

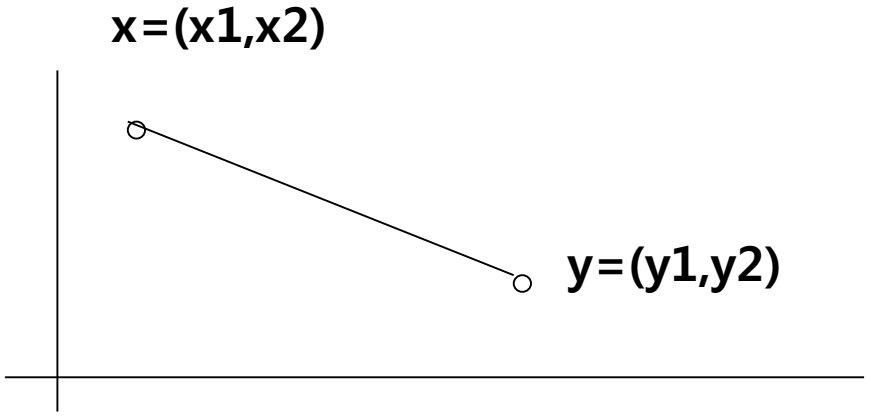
1) 유클리드(Euclid) 거리

P차원 공간에서 주어진 두 점 $x = (x_1, \dots, x_p)$ 와 $y = (y_1, \dots, y_p)$ 사이의 유클리드 거리 $d(x, y)$ 는

$$\mathrm{d(x,y)} = \left(\sum_{i=1}^{p} (x_i - y_i)^2 \right)^{1/2}$$

$p = 2$ 인 경우

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$



2.3 비유사성의 척도 : 거리

2) Minkowski 거리

$$d(x,y) = \left(\sum_{i=1}^{p} (x_i - y_i)^m \right)^{1/m}$$

3) 표준화거리

$$d(x,y) = \left(\sum_{i=1}^{p} \left(\frac{(x_i - y_i)}{s_i} \right)^2 \right)^{1/2}$$

4) Mahalanobis 거리

$$d(x,y) = \chi' \Sigma^{-1} y$$

2.3 비유사성의 척도 : 거리

5) 범주형 자료의 거리 : 불일치 항목 수

예)

개체	성별	학력	출신지역
A	남자	고졸	경기
B	여자	고졸	전남
C	남자	대졸	경기

$d(A,B) = 2,\ d(A,C) = 1,\ d(B,C) = 3$

2.3 비유사성의 척도 : 거리

6) Symbolic String 사이의 거리

예) $x = \{table\}, y = \{tale\}$

Hamming distance

ordered set : 불일치 bit 수
unordered set : $\max\{ n(x), x(y)\} - n(x^* y) = \max\{7,6\} - \{6\} = 1$

Edit distance

$= \min\{ replacement + insertion + deletion \} = 1$

Feature distance

$\max\{ n(x), n(y)\} - Nd \ (d = 2)$
$= 7 - n(\ \{t, ta, ab, bl, le, e\} * \{t, ta, al, le, e\}) = 3$

2.4 군집분석의 유형 및 특징

2.4 군집분석의 유형 및 특징

2.4.1 군집분석의 유형

1) 상호배반적(disjoint)군집

-각 관찰치가 상호배반적인 여러 군집 중 오직 하나에만 속함
-예) 한국인, 중국인, 일본인

2) 계보적(hierarchical)군집

- 한 군집이 다른 군집의 내부에 포함되는 형태로 군집간의 중복은 없으며, 군집들이 매 단계 계층적인 (나무)구조를 이룬다.
- 예) 전자제품 -> 주방용 -> 냉장고

2.4 군집분석의 유형 및 특징

3) 중복(overlapping) 군집

-두 개 이상의 군집에 한 관찰자가 동시에 포함되는 것을 허용

4) 퍼지(fuzzy) 군집

- 관찰치가 소속되는 특정한 군집을 표현하는 것이 아니라, 각 군집에 속할 가능성을 표현
- $\Pr($ 개체가 군집A에 속함$) = 0.7$, $\Pr($ 개체가 군집B에 속함$) = 0.3$

2.4 군집분석의 유형 및 특징

2.4.2 군집분석의 특징

군집분석은 그 기준의 설정, 즉 유사성이나 혹은 비유사성의 정의나 군집의 형태 등 매우 다양한 방법이 있다.

군집분석은 자료의 사전정보 없이 자료를 파악하는 방법으로, 분석자의 주관에 결과가 달라질 수 있다.

따라서, 군집분석은 한번에 분석이 끝나는 것이 아니고, 매회 결과를 잘 관찰하여 의미 있는 정보요약을 얻어내야 한다.

특이값을 갖는 개체의 발견, 결측값의 보정 등에 군집분석이 사용될 수 있다.

군집분석에서 군집을 분석하는 중요한 변수의 선택이 중요하다.

2.5 계층적 군집분석

2.5 계층적 군집분석

2.5.1 계층적 군집분석의 개요

가까운 관측값들 끼리 묶는 병합(agglomeration)방법과 먼 관측값들을 나누어가는 분할(division) 방법으로 나눌 수 있다.

계층적 군집분석에서는 주로 병합 방법이 주로 사용된다.

계층적 군집분석의 결과는 나무구조인 덴드로그램(dendrogram)을 통해 간단하게 나타낼 수 있고, 이를 이용하여 전체 군집들간의 구조적 관계를 쉽게 살펴볼 수 있다.

2.5 계층적 군집분석

2.5.2 병합방법

처음에 n개의 자료를 각각 하나의 군집으로 생각한다. 즉 군집의 수는 n이다.

이 n개의 군집 중 가장 거리가 가까운 두 개의 군집을 병합하여 n-1개의 군집으로 군집을 줄인다.

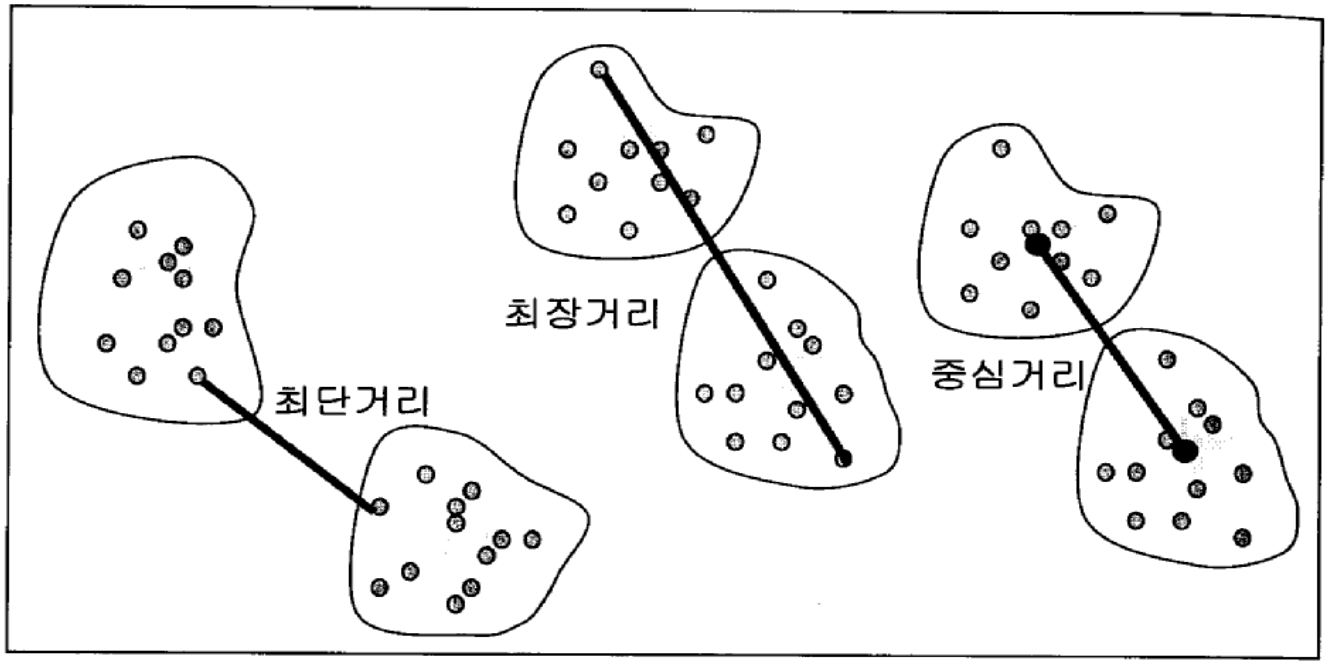
그 다음, n-1개의 군집 중 가장 가까운 두 군집을 병합하여 군집을 n-2개로 줄인다.

이를 반복하여 계속하여 군집의 수를 줄여 나간다.

이 과정은 시작부분에는 군집의 크기는 작고 동질적이며, 끝부분에서는 군집의 크기는 커지고 이질적이 된다.

2.5 계층적 군집분석

군집들간의 거리를 측정하는 방법에 따라 다양한 종류의 방법이 있다.
최단거리, 최장거리, 평균거리 방법 등이 있다.



[그림5-1] 군집 사이의 거리

2.5 계층적 군집분석

1) 최단연결법 (Single Linkage Method)

두 군집 사이의 거리를 각 군집에서 하나씩 관측값을 뽑았을 때 나타날 수 있는 거리의 최소값으로 측정

유리 위에 떨어진 물방울들이 서로 뭉치는 현상과 비슷

같은 군집에 속하는 관측치는 다른 군집에 속하는 관측치에 비하여 거리가 가까운 변수를 적어도 하나는 갖고있다.

군집이 고리형태로 연결되어 있는 경우에는 부적절한 결과를 제공한다.

고립된 군집을 찾는데 중점을 둔 방법이다.

2.5 계층적 군집분석

Ex) 최단연결법 예제

다음에 주어진 5개의 관측값에 대한 거리 행렬 (비유사성 행렬)에 대하여 최단연결법으로 군집을 얻고 덴드로그램으로 나타내보자.

1	0				
2	7	0			
3	1	6	0		
4	9	3	8	0	
5	8	5	7	4	0

2.5 계층적 군집분석

1단계

거리 행렬에서 $d(1,3) = 1$이 최소이므로 관측값 1과 3을 묶어 군집 (1,3)을 만든다.

군집 (1,3)과 관측값 2,4,5와의 거리를 구하여 다음과 같은 거리행렬을 만든다

$$d((2),(1,3)) = \min\{d(2,1), d(2,3)\} = d(2,3) = 6$$
$$d((4),(1,3)) = \min\{d(4,1), d(4,3)\} = d(4,3) = 8$$
$$d((5),(1,3)) = \min\{d(5,1), d(5,3)\} = d(5,3) = 7$$

(1,3)	0			
2	6	0		
4	8	3	0	
5	7	5	4	0

2.5 계층적 군집분석

2단계

다음의 거리 행렬에서 $d(2,4) = 3$이 최소값을 가지므로 관측값 2와 4를 묶어 군집 (2,4)를 만든다.

군집 (2,4)와 군집 (1,3), (5)와의 거리를 구한 후 거리 행렬을 다시 다음과 같이 만든다.

$$d((2,4),(1,3)) = \min\{d((2),(1,3)), d((4),(1,3))\} = d((2),(1,3)\} = 6$$
$$d((5),(2,4)) = \min\{d(5,2), d(5,4)\} = d(5,4) = 4,$$

(1,3)	0		
(2,4)	6	0	
5	7	4	0

2.5 계층적 군집분석

3단계

$d((5),(2,4)) = 4$이 최소값을 가지므로 군집 (2,4)와 (5)를 묶어 (2,4,5)를 만든 후 $d((1,3),(2,4,5)) = d(2,3) = 6$을 이용하여 다음의 거리행렬을 얻는다.

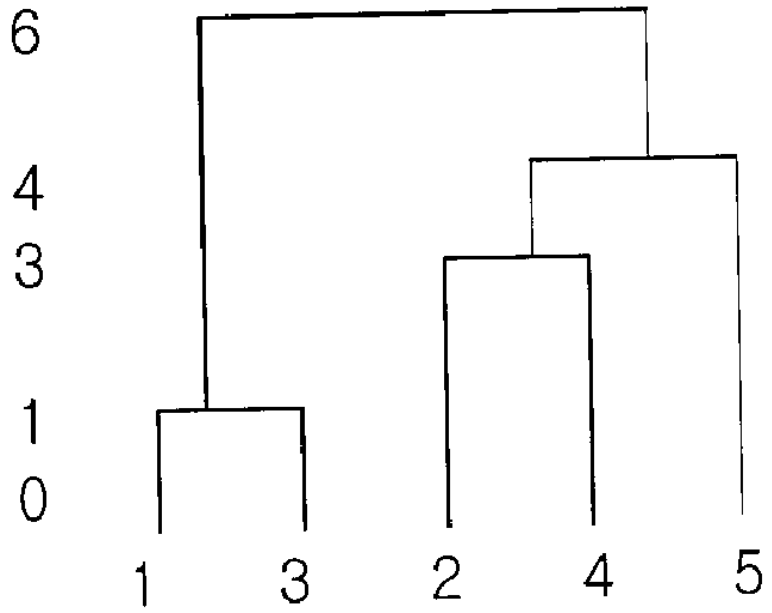
(1,3)	0	
(2,4,5)	6	0

4단계

마지막 단계로 전체가 하나의 군집을 이룬다.

2.5 계층적 군집분석

덴드로그램을 그리면 다음과 같다.



2.5 계층적 군집분석

2) 최장(완전)연결법 (Complete Linkage Method)

두 군집 사이의 거리를 각 군집에서 하나씩 관측값을 뽑았을 때 나타날 수 있는 거리의 최대값으로 측정

같은 군집에 속하는 관측치는 알려진 최대 거리보다 짧다.

군집들의 내부 응집성에 중점을 둔다.

2.5 계층적 군집분석

Ex) 최장연결법 예제

다음에 주어진 5개의 관측값에 대한 거리 행렬 (비유사성 행렬)에 대하여 최장연결법으로 군집을 얻고 덴드로그램으로 나타내보자.

1	0				
2	7	0			
3	1	6	0		
4	9	3	8	0	
5	8	5	7	4	0

2.5 계층적 군집분석

1단계

1단계는 최단연결법의 1단계와 같다. 즉, 관측값 1과 3이 최단거리에 위치하고 이를 묶어서 새로운 군집 (1,3)을 만든다.

군집 (1,3)과 관측값 2,4,5와의 거리를 구하여 다음과 같은 거리 행렬을 만든다.

$$d((2),(1,3)) = \max\{d(2,1), d(2,3)\} = d(2,1) = 7$$
$$d((4),(1,3)) = \max\{d(4,1), d(4,3)\} = d(4,1) = 9$$
$$d((5),(1,3)) = \max\{d(5,1), d(5,3)\} = d(5,1) = 8$$

(1,3)	0			
2	7	0		
4	9	3	0	
5	8	5	4	0

2.5 계층적 군집분석

2단계

다음의 거리 행렬에서 $d(2,4) = 3$이 최소값을 가지므로 관측값 2와 4를 묶어 군집 (2,4)를 만든다.

군집 (2,4)와 군집 (1,3), (5)와의 거리를 구한 후 거리 행렬을 다시 다음과 같이 만든다.

$$d((2,4),(1,3)) = \max\{d((2),(1,3)), d((4),(1,3))\} = d((4),(1,3)\} = 9$$
$$d((5),(2,4)) = \max\{d(5,2), d(5,4)\} = d(2,4) = 5,$$

(1,3)	0		
(2,4)	9	0	
5	7	5	0

2.5 계층적 군집분석

3단계

$d((5),(2,4)) = 5$이 최소값을 가지므로 군집 (2,4)와 (5)를 묶어 (2,4,5)를 만든 후 $d((1,3),(2,4,5)) = d(2,3) = 9$을 이용하여 다음의 거리행렬을 얻는다.

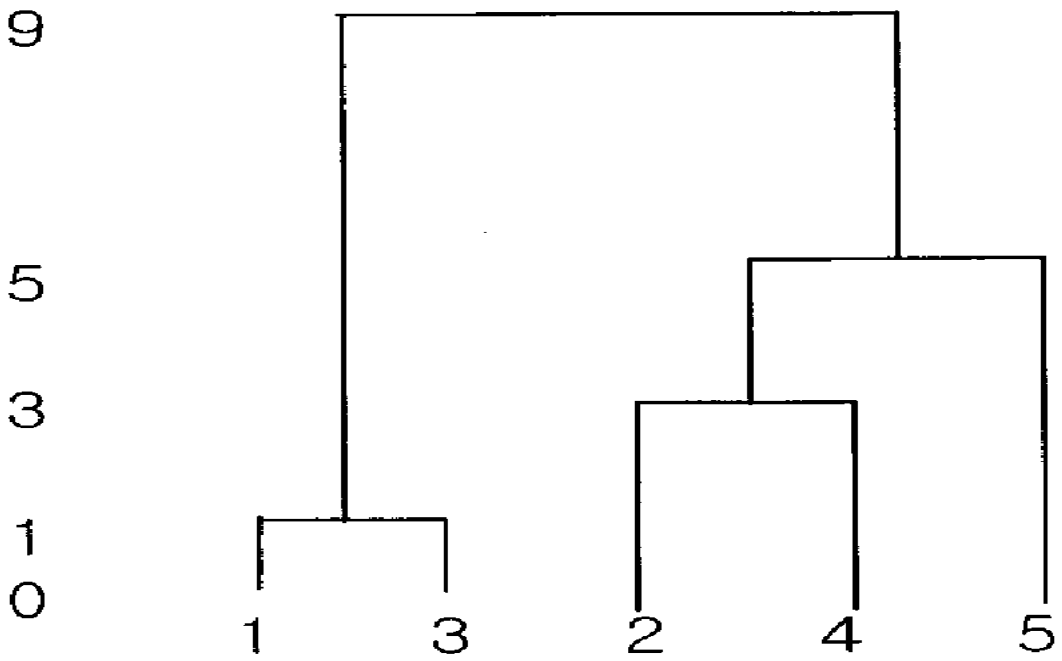
(1,3)	0	
(2,4,5)	9	0

4단계

마지막 단계로 전체가 하나의 군집을 이룬다.

2.5 계층적 군집분석

덴드로그램을 그리면 다음과 같다.



2.6 비계층적 군집분석

2.6 비계층적 군집분석

2.6.1 비계층적 군집분석의 개요

비계층적 군집분석에는 흔히 관측값들을 몇 개의 군집으로 나누기 위하여 주어진 판정기준을 최적화 한다.

따라서, 최적분리 군집분석이라고 한다.

대표적인 비계층적 군집분석 방법이 k- 평균 방법이다.

2.6 비계층적 군집분석

2.6.2 K-평균 군집방법

K-평균 군집방법

사전에 결정된 군집수 k에 기초하여

전체 데이터를 상대적으로 유사한 k개의 군집으로 구분한다.

k-평균 군집법은 계보적 군집법에 비하여 계산량이 적다.

따라서, 대용량 데이터를 빠르게 처리할 수 있다.

2.6 비계층적 군집분석

1) K-평균 군집방법의 알고리즘

군집수 k를 결정한다.

초기 k개 군집의 중심을 선택한다.

각 관찰치를 그 중심과 가장 가까운 거리에 있는 군집에 할당한다.

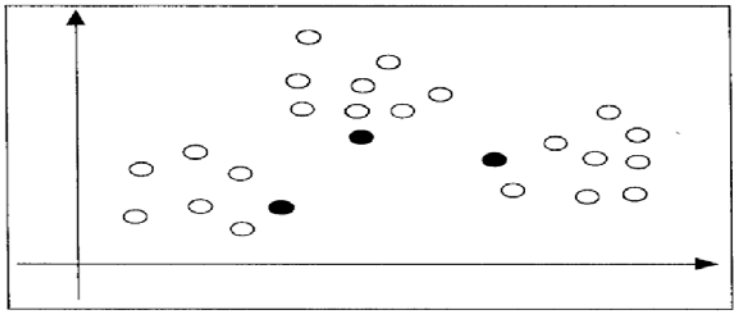
위의 과정을 기존의 중심과 새로운 중심의 차이가 없을 때까지 반복한다.

2.6 비계층적 군집분석

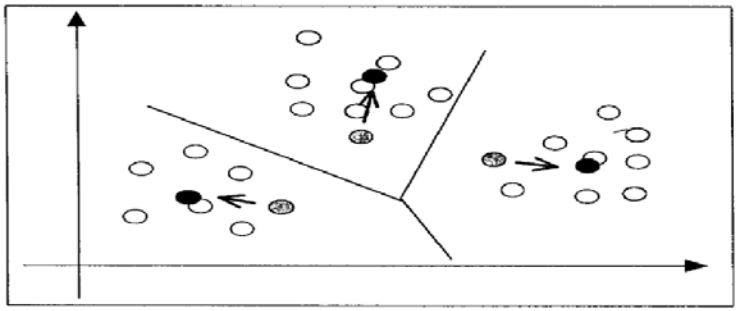
<u>알고리즘 설명</u>

다음의 그림을 생각하자.

〈그림 7-3〉
k-평균 군집법
: 초기 군집의 중심



〈그림 7-4〉
k-평균 군집법
: 중심의 이동



그림은 거리를 표현하는 변수가 x1,x2로 하여 2차원으로 표현한 각 관찰치의 위치이다.

그림을 보고 먼저, 군집수를 3으로 결정한다.

자료를 임의로 3등분하여 각각의 평균을 각 군집의 평균으로 정한다.

그림 7-4와 같이 주어진 군집의 중심을 기준으로 관찰치를 가장 가까운 군집에 할당한다.

새로운 군집의 생기면 각 새로운 군집의 중심을 각 군집의 평균으로 갱신한다. 그림 7-4에서 회색점이 과거 중심이고 검은점이 갱신된 중심이다.

2.6 비계층적 군집분석

2) 초기군집수의 결정

K-평균 군집분석법의 결과는 초기 군집수 k의 결정에 민감하게 반응한다.

실제 자료의 분석에서는 여러 가지의 k값을 선택하여 군집분석을 수행한 후 가장 좋다고 생각되는 k값을 이용한다.

여러 개의 군집분석 결과 중 어떤 결과가 좋은가 하는 문제는 관측값 간의 평균거리와 군집간의 평균거리를 비교함으로써 수행한다.

가장 좋은 방법은 자료의 시각화를 통한 최적 군집수의 결정인데, 자료의 시각화를 위하여는 차원의 축소가 필수적이고, 이를 위하여 주성분 분석방법이 널리 사용된다.

시각화가 어려운 경우에는 , 여러 가지 통계량을 사용하는데, 예를 들면, 각 그룹의 산포행렬의 행렬식을 최소로 하는 군집수를 찾는다.

2.6 비계층적 군집분석

3) K-평균방법의 변형

k- 평균방법의 단점으로는

군집이 겹치는 경우에 좋지 않고

이상치에 민감하며

각 관찰치가 할당된 군집에 속하지 않을 불확실성에 대한 측정치가 없다.

이러한 단점을 극복한 군집방법이 **가우스 혼합모형 (Gaussian mixture model)**이 있다.

2.6 비계층적 군집분석

가우스 혼합모형 (Gaussian mixture model)

주어진 군집수 k에 대하여, 각 군집의 관측치의 분포가 미지의 평균과 분산을 따르는 정규분포라고 가정한다.

자료를 가장 잘 분리할 수 있는 최적의 평균과 분산을 추정과 최대화의 두 단계를 반복하여 구한다.

결과물로는 각 관찰치에 대하여 그 관찰치가 각 군집에 속할 확률이 계산된다.

마지막으로, 각 관찰치는 가장 높은 확률을 갖는 집단으로 할당된다.

이러한 방법을 소프트 군집화 (soft clustering)이라 한다.

2.7 단위변환, 가중치 부여

2.7 단위변환, 가중치 부여

2.7.1 단위변환

군집분석은 자료 사이의 거리를 이용하여 수행되기 때문에, 각 자료의 단위가 결과에 큰 영향을 미친다.

예를 들면, (x,y,z) 세 개의 변수가 어떤 거리를 측정하였다고 했을 때, 그 단위가 x는 야드, y는 센티미터, z는 마일로 측정되었다면, 그 거리의 계산에 유의 하여야 한다.

z의 단위 1의 차이는 y의 단위 185,200의 차이와 같고, x의 2,025와 같다.

만약, 서로 다른 종류의 측정치로 자료가 구성되어 있으면, 위와 같은 상대적인 평가도 불가능 하다.

예를 들면, 대지면적, 수입, 건평 등으로 이루어진 자료는 각 변수가 서로 비교될 수 가 없다.

2.7 단위변환, 가중치 부여

이러한 문제를 해결하기 위하여, 가장 널리 쓰이는 방법이 표준화 방법이다.

표준화 방법이란 각 변수의 관찰값으로부터 그 변수의 평균을 빼고, 그 변수의 표준편차로 나누는 것이다.

표준화된 자료는 모든 변수가 평균이 0이고 표준편차가 1이 된다.

표준화된 자료의 유클리드거리는 표준화거리와 같다.

2.7 단위변환, 가중치 부여

2.7.2 가중치 부여

자료의 분석 전에 각 변수의 중요도가 같지 않음을 안다면, 적절한 가중치를 이용하여 각 변수의 중요도를 조절할 수 있다.

예를 들면, 같은 수입을 가지는 두 가족이 같은 대지면적을 가지는 두 가족보다 공통점이 많다고 생각이 드는 경우가 있다.

이 경우에는, 수입 변수에 높은 가중치를 주고 대지면적 변수에 낮은 가중치를 줌으로써 해결한다.

가중치는 대부분의 경우 단위변환(표준화)를 수행한 후 부여한다.

가중치에 대한 군집의 영향을 평가 하기 위하여는 여러 가지의 가중치에 대하여 군집분석의 결과를 구하고 이 결과들을 비교한다.

2.8 군집평가 및 변수선택

2.8 군집평가 및 변수선택

2.8.1 군집평가

군집분석에는 분석 전에 정해야 하는 사항이 많다 (예: 초기군집수, 가중치 등)

분석자의 주관에 의하여 결정되는 이러한 사항들이 군집분석의 결과에 어떻게 영향을 미치는 가를 알아보기 위하여는, 군집분석 결과의 평가가 필수적이다.

좋은 결과는 각 군집 안에서의 분산이 최소로 되는 것이다.

또는, 사용되어진 거리의 측도를 이용하여 군집내의 거리의 평균과 군집간의 거리의 평균을 비교할 수 있다.

즉, 군집내의 거리의 평균이 군집간의 거리의 평균 보다 작으면 좋은 결과이다.

2.8 군집평가 및 변수선택

2.8.2 변수선택

찾아진 각 군집은 어떠한 변수에 의하여 군집이 형성됐는가를 파악하는 것을 목적으로 한다.

각 변수에 대한 그룹내의 거리의 평균과 그룹간의 거리의 평균을 측정한다.

그룹내의 거리가 그룹간의 거리에 비하여 아주 작은 변수가, 그 군집을 형성하는데 크게 기여하는 변수이다.

예를 들면, 군집분석을 수행한 결과 특정한 군집에는 소득이 비슷한 사람들이 많이 모여 있음을 알 수 있다. 이를 통하여, 소득이 자료의 패턴에 큰 영향을 주는 것을 확인 할 수 있다.

2.9 자기영상 군집분석

2.9 자기영상 군집분석

군집분석에서 오직 하나만의 군집이 존재하는 경우에 아주 유용하게 사용될 수 있다.

예를 들면, 모터제조 공장에서 모터의 불량원인을 알고자 한다.

이 경우, 정상적인 모터의 자료를 이용하여 군집분석을 수행하면, 하나의 군집이 찾아진다.

새로운 모터와 이 군집과의 거리가 크면, 이 새로운 모터를 불량모터라고 의심한다.

또 다른 예로는, 위조지폐 탐지가 있다.

2.10 군집분석의 장단점

2.10 군집분석의 장단점

장점:

탐색적인 기법: 자료의 내부구조에 대한 사전정보 없이 의미 있는 자료구조를 찾아낼 수 있다.

다양한 형태의 데이터에 적용가능: 거리만 잘 정의되면, 모든 종류의 자료에 적용할 수 있다.
예를 들면, 신문기사와 같은 텍스트 자료도 그 거리만 잘 정의하면 얼마든지 군집분석을 사용할 수 있다.

분석방법의 적용 용이성: 자료의 사전정보를 필요로 하지 않아서 누구나 쉽게 분석가능

단점:

가중치와 거리 정의: 가중치와 거리를 어떻게 정의하는가에 따라 분석의 결과가 민감하게 반응.

초기 군집수 k의 결정이 쉽지 않다.

결과의 해석이 어렵다. 특히, 찾아진 군집이 무엇을 의미 하는지 데이터만으로는 알 수 없음.

2.11 군집분석의 실습

2.11 군집분석 실습

시리얼 자료 (cereals.csv)
: 다음의 자료는 미국에서 판매되는 77가지 시리얼의 영양성분과 위치에 관한 자료이다. 자료의 세부 정보는 다음과 같다.

1. Name : 시리얼의 이름
2. Mfr : 시리얼 제조사
   A : American Home Food Products, G: General Mills, K : Kelloggs, N : Nabisco,
   P : Post , Q : Quaker Oats, R : Ralston Purina
3. type : 차갑게 먹는가 따듯하게 먹는가 (cold / hot)
4. Calories : 1회 제공량 당 칼로리
5. Protein : 단백질 함량(그램)
6. Fat : 지방 함량(그램)
7. Sodium : 소금 함량 (밀리그램)
8. fiber : 식이섬유 함량(그램)
9. carbo : 복합탄수화물 함량 (그램)
10. sugars : 설탕 함량 (그램)
11. potass : 칼륨 함량 (밀리그램)
12. vitamins : FDA 기준치 대비 비타민, 미네랄 함량 %
13. shelf : 진열대 위치 (바닥부터 1,2,3층)
14. weight : 1회 제공량 당 무게 (온스)
15. cups : 1회 제공량 당 컵 단위 (ex. 1.5 컵, 0.9컵 등)
16. rating : 소비자 조사에 의한 시리얼 평점

2.11 군집분석의 실습

(1) 자료의 형태 확인 및 필요한 자료 선택

: 칼로리, 단백질, 지방, 소금, 식이섬유, 복합 탄수화물, 설탕, 칼륨, 그리고 비타민 함량의 9개의 변수를 선택하여 이용하기로 한다. Missing value가 있는 자료는 삭제하기로 한다.

```
> cereal=read.csv("C:\\Users\\Sunghyun Sun Cho\\Desktop\\Insight\\cereals.csv")
> cereal=cereal[,c("name","calories","protein","fat","sodium","fiber","carbo","sugars","potass","vitamins")]
> cereal[!complete.cases(cereal),]
                   name calories protein fat sodium fiber carbo sugars potass vitamins
5         Almond_Delight      110       2   2    200   1.0    14       8     NA       25
21 Cream_of_Wheat_(Quick)      100       3   0     80   1.0    21       0     NA        0
58         Quaker_Oatmeal      100       5   2      0   2.7    NA      NA    110        0
> cereal=cereal[-c(5,21,58),]
> rownames(cereal)=cereal[,"name"]
> cereal=cereal[,-1]
> head(cereal)
                      calories protein fat sodium fiber carbo sugars potass vitamins
100%_Bran                   70       4   1    130  10.0   5.0       6    280       25
100%_Natural_Bran          120       3   5     15   2.0   8.0       8    135        0
All-Bran                    70       4   1    260   9.0   7.0       5    320       25
All-Bran_with_Extra_Fiber   50       4   0    140  14.0   8.0       0    330       25
Apple_Cinnamon_Cheerios    110       2   2    180   1.5  10.5      10     70       25
Apple_Jacks                110       2   0    125   1.0  11.0      14     30       25
```

따라서 74개의 관측치와 9개의 변수로 이루어진 자료를 만들었다

2.11 군집분석의 실습

산포도 확인

```
> plot(cereal)
```

2.11 군집분석의 실습

(2) 계층적 군집분석

계층적 군집 분석을 결정짓는 두 요인 : 거리 측정 & 연결방법

a. 거리측정 : **dist ()** - 거리행렬을 계산한다

선택 가능한 거리측정 방법 : Euclidean distance, Maximum, Manhattan distance, canberra, binary, minkowski 등 사용 가능

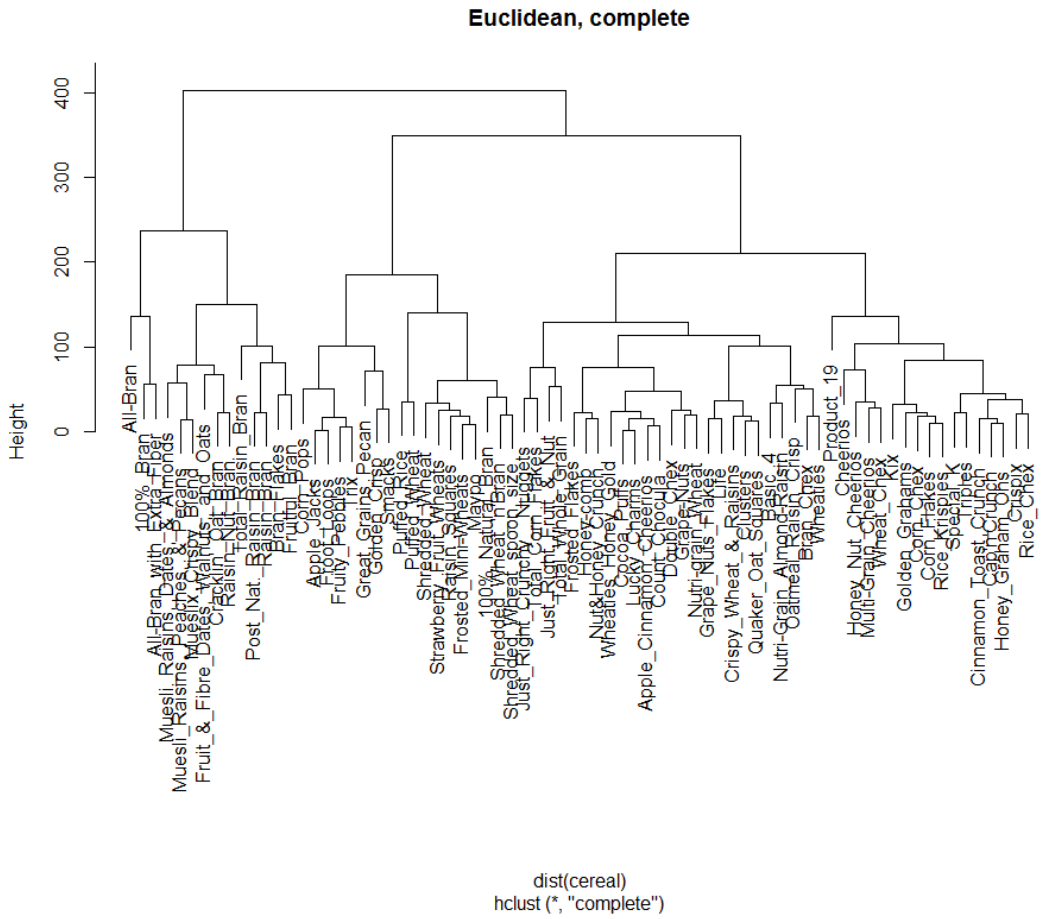
b. 군집분석 : **hclust (distance, method)**

distance 는 각 개체간의 거리를 행렬로 나타낸 것을 입력한다.
Method 는 연결방법을 지정한다. 기본으로 complete가 사용된다

2.11 군집분석의 실습

(2) 계층적 군집분석 : Euclidean distance, complete linkage

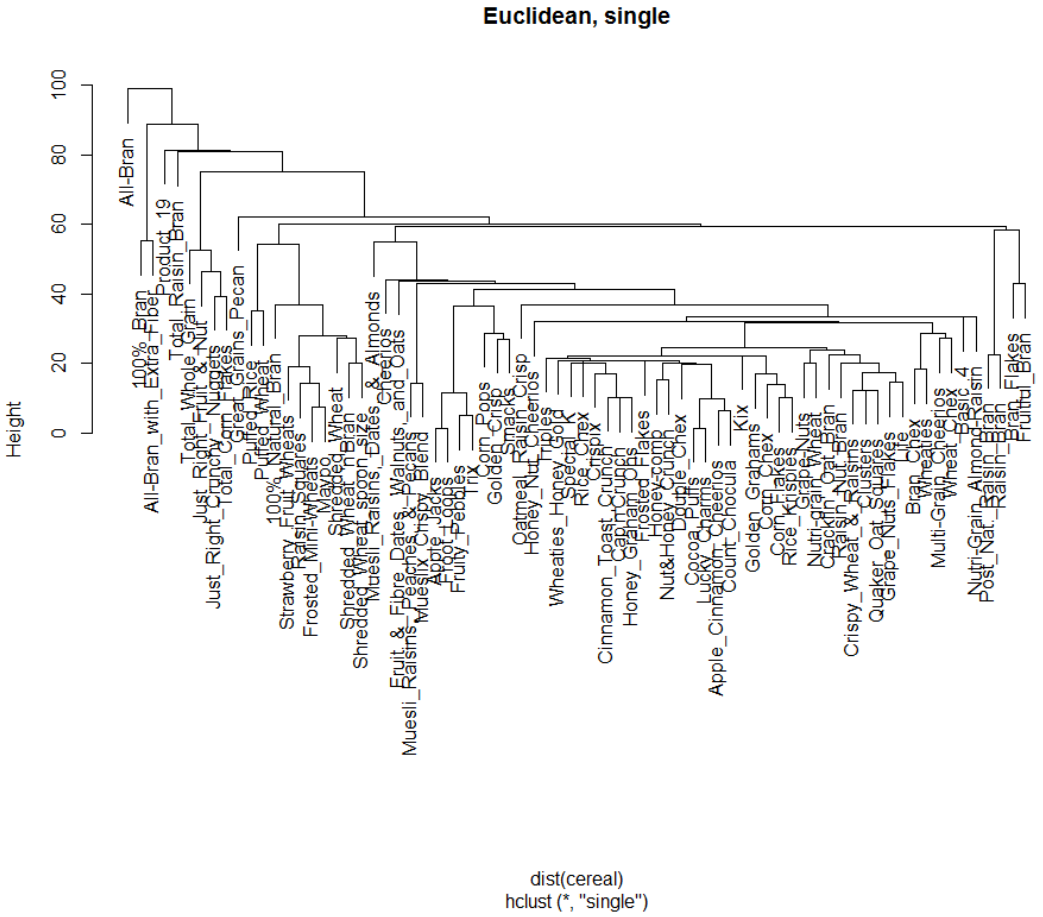
```
### Euclidean distance, complete linkage
clustering.EC <- hclust(dist(cereal))
plot(clustering.EC, main="Euclidean, complete")
```



2.11 군집분석의 실습

(2) 계층적 군집분석 : Euclidean distance, single linkage

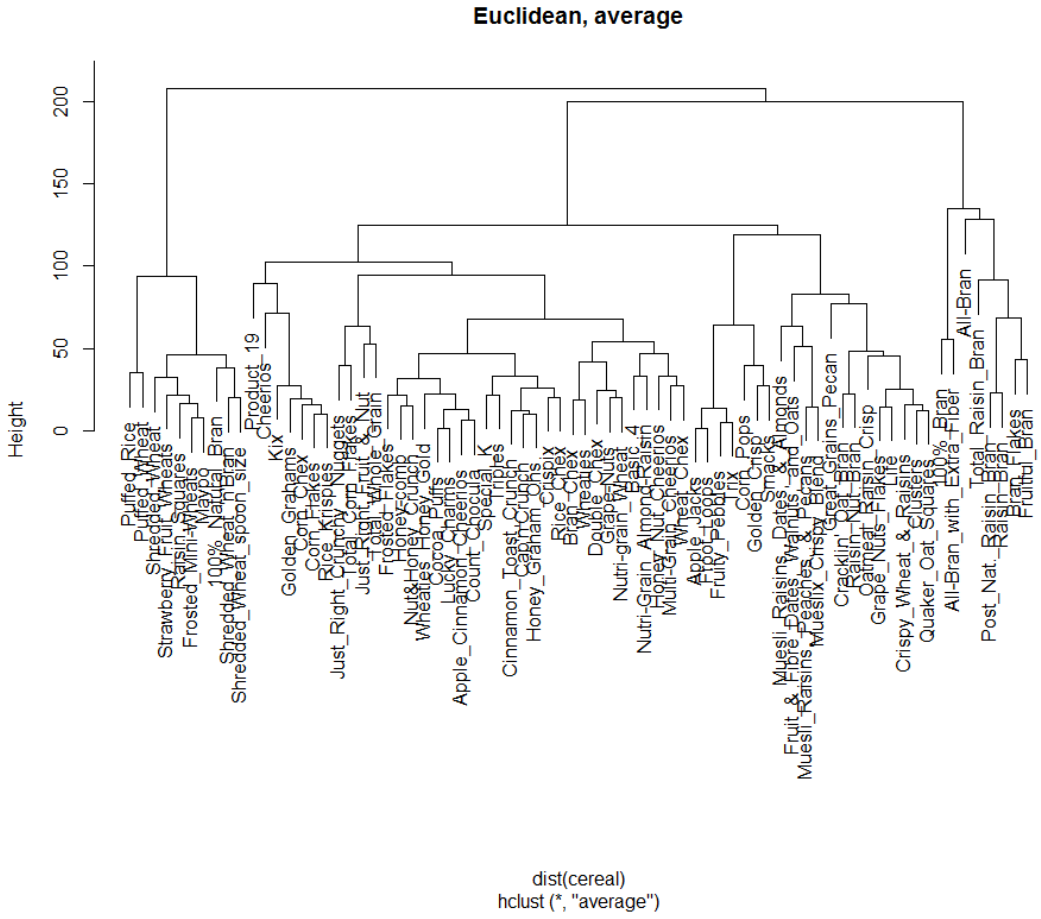
```
### Euclidean distance, single linkage
clustering.ES = hclust(dist(cereal), method="single")
plot(clustering.ES, main="Euclidean, single")
```



2.11 군집분석의 실습

(2) 계층적 군집분석 : Euclidean distance, average linkage

```
### Euclidean distance, average linkage
clustering.EA = hclust(dist(cereal), method="average")
plot(clustering.EA, main="Euclidean, average")
```



2.11 군집분석의 실습

(2) 계층적 군집분석 – 1. 군집 할당

-군집분석을 이용하여 각 개체를 군집에 할당한다.
-덴드로그램의 높이나 군집수를 조정하여 군집을 분할할 수 있다.

a.군집 수를 조절하면서 분할

```
> cutree(clustering.EC, k=2:5)
                           2 3 4 5
100%_Bran                  1 1 1 1
100%_Natural_Bran          2 2 2 2
All-Bran                   1 1 1 1
All-Bran_with_Extra_Fiber  1 1 1 1
Apple_Cinnamon_Cheerios    2 3 3 3
Apple_Jacks                2 2 2 2
Basic_4                    2 3 3 3
Bran_Chex                  2 3 3 3
Bran_Flakes                1 1 4 4
Cap'n'Crunch               2 3 3 5
Cheerios                   2 3 3 5
Cinnamon_Toast_Crunch      2 3 3 5
Clusters                   2 3 3 3
Cocoa_Puffs                2 3 3 3
Corn_Chex                  2 3 3 5
Corn_Flakes                2 3 3 5
```

2.11 군집분석의 실습

(2) 계층적 군집분석 – 1. 군집 할당

-군집분석을 이용하여 각 개체를 군집에 할당한다.
-덴드로그램의 높이나 군집수를 조정하여 군집을 분할할 수 있다.

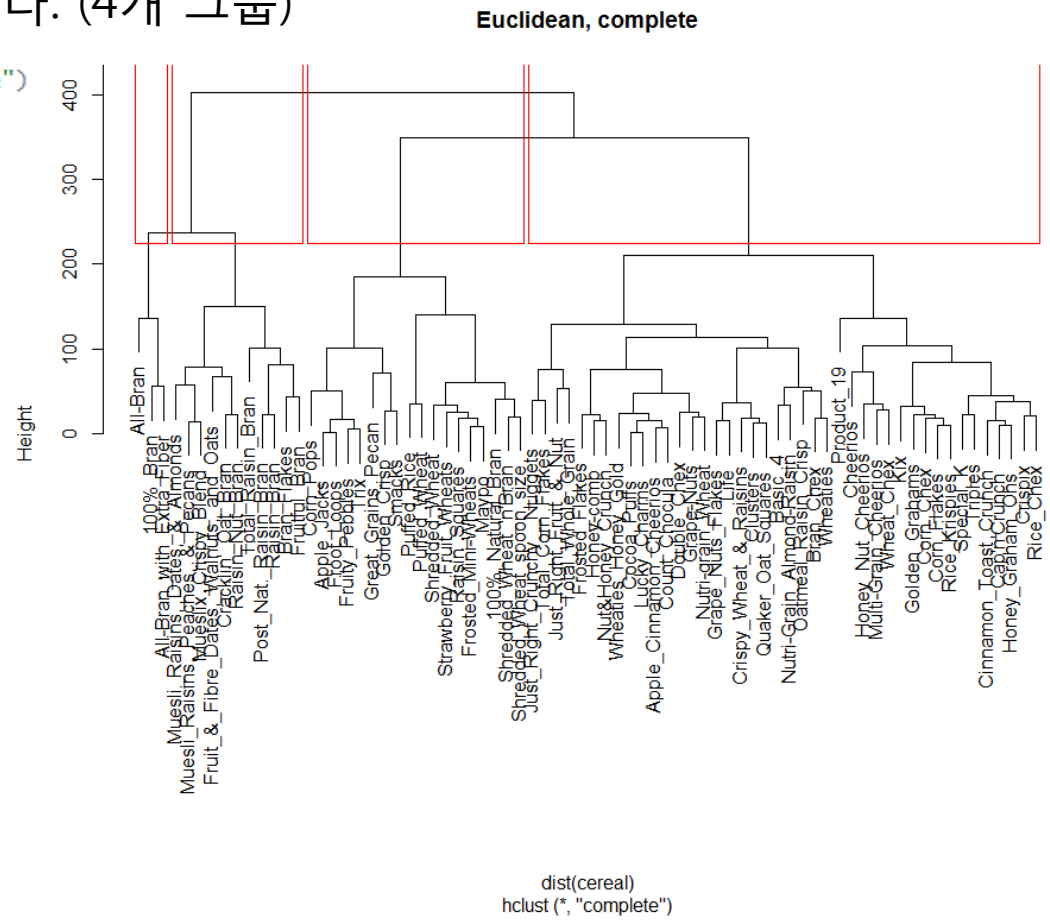
b.높이를 조절하면서 분할

```
> cutree(clustering.EC, h=100)
                         100%_Bran                    100%_Natural_Bran
                                 1                                    2
                          All-Bran          All-Bran_with_Extra_Fiber
                                 3                                    1
          Apple_Cinnamon_Cheerios                          Apple_Jacks
                                 4                                    5
                           Basic_4                        Bran_Chex
                                 6                                    6
                       Bran_Flakes                      Cap'n'Crunch
                                 7                                    8
                          Cheerios                 Cinnamon_Toast_Crunch
                                 9                                    8
                          Clusters                         Cocoa_Puffs
                                10                                    4
                         Corn_Chex                        Corn_Flakes
                                 8                                    8
                         Corn_Pops                      Count_Chocula
                                 5                                    4
              Cracklin'_Oat_Bran                             Crispix
                                11                                    8
           Crispy_Wheat_&_Raisins                       Double_Chex
                                10                                    4
```

2.11 군집분석의 실습

(2) 계층적 군집분석 – 2. 군집 할당
: 덴드로그램에서 군집을 나눈다. (4개 그룹)

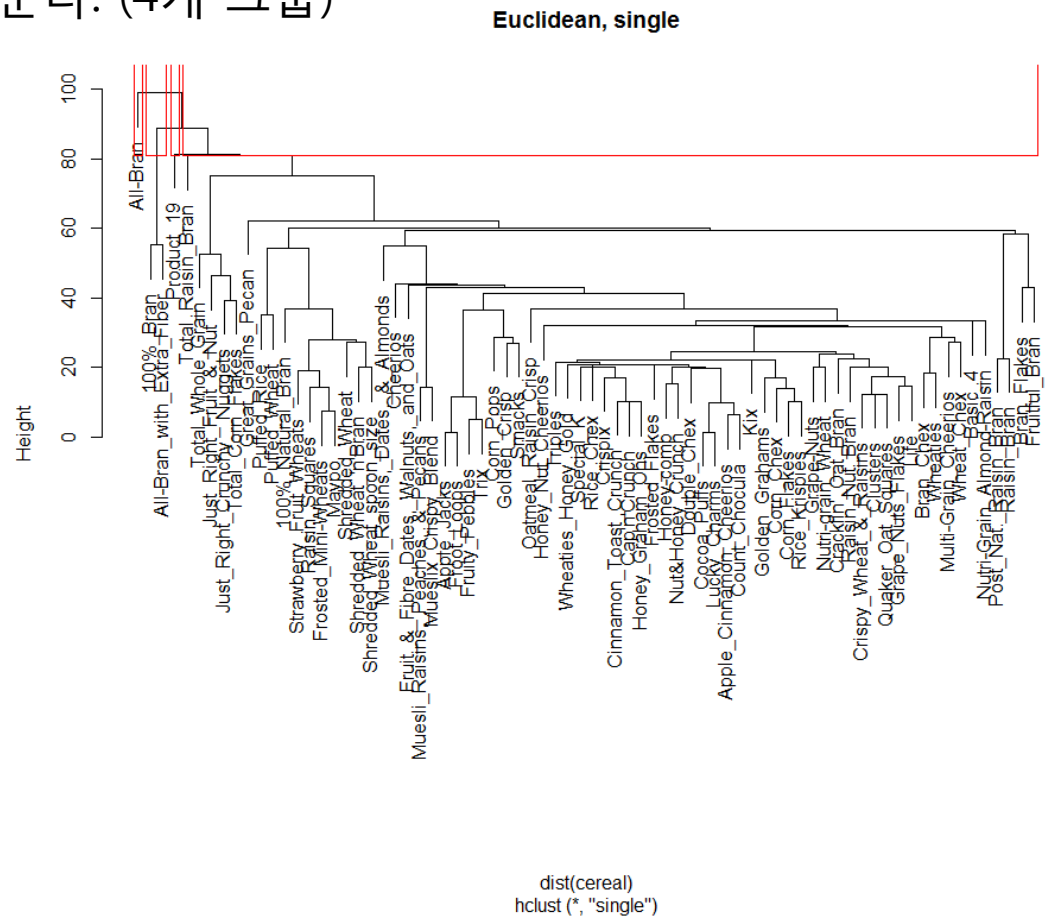
```
plot(clustering.EC, main="Euclidean, complete")
rect.hclust(clustering.EC, k=4, border="red")
```



2.11 군집분석의 실습

(2) 계층적 군집분석 – 2. 군집 할당
: 덴드로그램에서 군집을 나눈다. (4개 그룹)

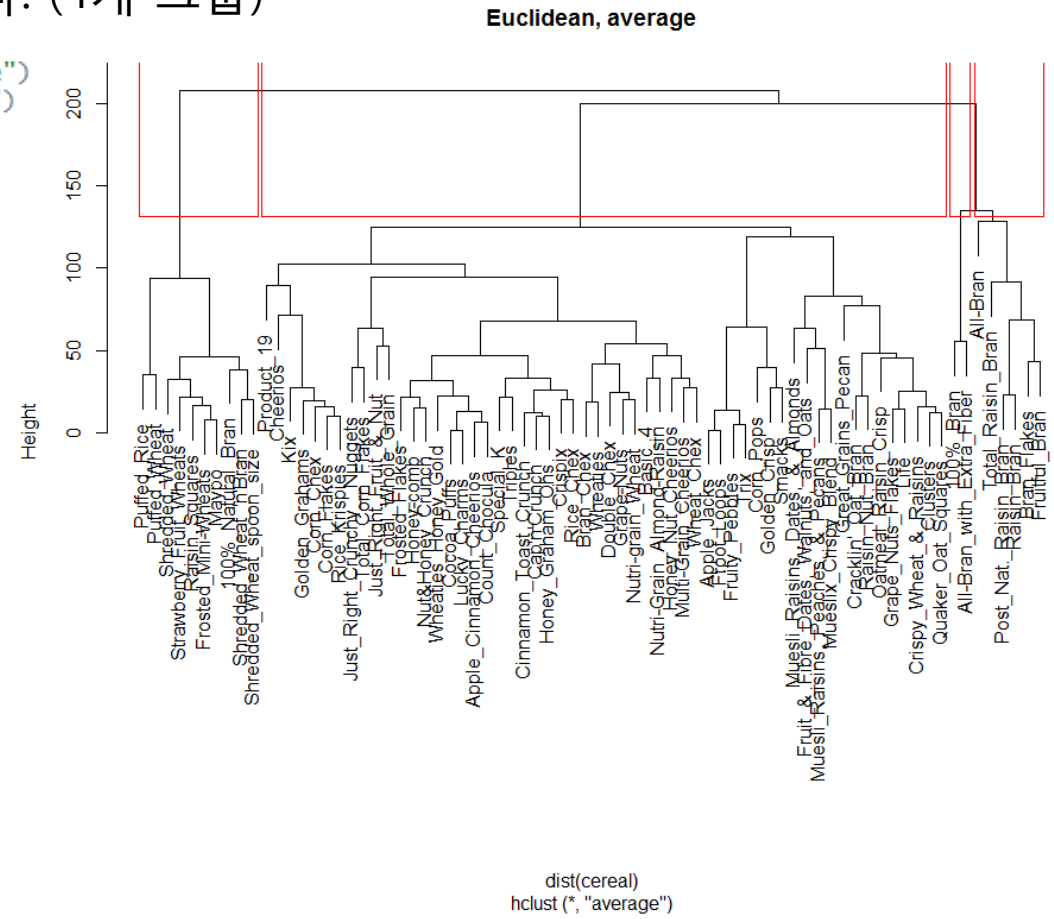
```
plot(clustering.ES, main="Euclidean, single")
rect.hclust(clustering.ES, k=4, border="red")
```



2.11 군집분석의 실습

(2) 계층적 군집분석 – 2. 군집 할당
: 덴드로그램에서 군집을 나눈다. (4개 그룹)

```
plot(clustering.EA, main="Euclidean, average")
rect.hclust(clustering.EA, k=4, border="red")
```



2.11 군집분석의 실습

(2) 계층적 군집분석 – 3. 군집 내 비교 (4개 그룹) :
Euclidean distance, Complete linkage

칼로리 유사

```
> summary(cereal[cutree(clustering.EC,k=4)==1,])
    calories        protein          fat            sodium          fiber            carbo          sugars          potass         vitamins
 Min.   :50.00   Min.   :4   Min.   :0.0000   Min.   :130.0   Min.   : 9.0   Min.   :5.000   Min.   :0.000   Min.   :280   Min.   :25
 1st Qu.:60.00   1st Qu.:4   1st Qu.:0.5000   1st Qu.:135.0   1st Qu.: 9.5   1st Qu.:6.000   1st Qu.:2.500   1st Qu.:300   1st Qu.:25
 Median :70.00   Median :4   Median :1.0000   Median :140.0   Median :10.0   Median :7.000   Median :5.000   Median :320   Median :25
 Mean   :63.33   Mean   :4   Mean   :0.6667   Mean   :176.7   Mean   :11.0   Mean   :6.667   Mean   :3.667   Mean   :310   Mean   :25
 3rd Qu.:70.00   3rd Qu.:4   3rd Qu.:1.0000   3rd Qu.:200.0   3rd Qu.:12.0   3rd Qu.:7.500   3rd Qu.:5.500   3rd Qu.:325   3rd Qu.:25
 Max.   :70.00   Max.   :4   Max.   :1.0000   Max.   :260.0   Max.   :14.0   Max.   :8.000   Max.   :6.000   Max.   :330   Max.   :25
> summary(cereal[cutree(clustering.EC,k=4)==2,])
    calories        protein          fat            sodium          fiber            carbo          sugars          potass         vitamins
 Min.   : 50.00   Min.   :1.000   Min.   :0.0000   Min.   :  0.00   Min.   :0.000   Min.   : 8.00   Min.   : 0.00   Min.   : 15   Min.   : 0.00
 1st Qu.: 90.00   1st Qu.:2.000   1st Qu.:0.0000   1st Qu.:  0.00   1st Qu.:0.250   1st Qu.:11.00   1st Qu.: 0.75   1st Qu.: 30   1st Qu.: 0.00
 Median :100.00   Median :2.000   Median :0.0000   Median : 15.00   Median :1.000   Median :13.00   Median : 6.50   Median : 70   Median :25.00
 Mean   : 96.67   Mean   :2.167   Mean   :0.7222   Mean   : 46.39   Mean   :1.556   Mean   :13.33   Mean   : 7.00   Mean   : 70   Mean   :16.67
 3rd Qu.:110.00   3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.: 86.25   3rd Qu.:3.000   3rd Qu.:15.00   3rd Qu.:12.00   3rd Qu.:100   3rd Qu.:25.00
 Max.   :120.00   Max.   :4.000   Max.   :5.0000   Max.   :140.00   Max.   :4.000   Max.   :20.00   Max.   :15.00   Max.   :140   Max.   :25.00
```

칼로리 유사

```
> summary(cereal[cutree(clustering.EC,k=4)==3,])
    calories        protein          fat            sodium          fiber            carbo          sugars          potass         vitamins
 Min.   : 90.0   Min.   :1.000   Min.   :0.0000   Min.   :135.0   Min.   :0.000   Min.   :10.50   Min.   : 1.000   Min.   : 25.00   Min.   : 25.00
 1st Qu.:100.0   1st Qu.:2.000   1st Qu.:0.0000   1st Qu.:180.0   1st Qu.:0.000   1st Qu.:13.12   1st Qu.: 3.000   1st Qu.: 41.25   1st Qu.: 25.00
 Median :110.0   Median :2.000   Median :1.0000   Median :200.0   Median :1.000   Median :16.00   Median : 6.000   Median : 62.50   Median : 25.00
 Mean   :109.8   Mean   :2.381   Mean   :0.9762   Mean   :208.7   Mean   :1.298   Mean   :16.23   Mean   : 6.476   Mean   : 71.19   Mean   : 33.93
 3rd Qu.:110.0   3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:230.0   3rd Qu.:2.000   3rd Qu.:19.50   3rd Qu.: 9.750   3rd Qu.: 98.75   3rd Qu.: 25.00
 Max.   :140.0   Max.   :6.000   Max.   :3.0000   Max.   :320.0   Max.   :4.000   Max.   :23.00   Max.   :13.000   Max.   :130.00   Max.   :100.00
> summary(cereal[cutree(clustering.EC,k=4)==4,])
    calories        protein          fat            sodium          fiber            carbo          sugars          potass         vitamins
 Min.   : 90.0   Min.   :3.000   Min.   :0.000   Min.   : 95.0   Min.   :2.500   Min.   :10.0   Min.   : 5.00   Min.   :140.0   Min.   : 25.00
 1st Qu.:115.0   1st Qu.:3.000   1st Qu.:1.000   1st Qu.:145.0   1st Qu.:3.000   1st Qu.:11.5   1st Qu.: 9.00   1st Qu.:165.0   1st Qu.: 25.00
 Median :120.0   Median :3.000   Median :2.000   Median :160.0   Median :4.000   Median :14.0   Median :11.00   Median :190.0   Median : 25.00
 Mean   :125.5   Mean   :3.182   Mean   :1.636   Mean   :171.4   Mean   :4.136   Mean   :13.5   Mean   :10.64   Mean   :191.8   Mean   : 31.82
 3rd Qu.:145.0   3rd Qu.:3.000   3rd Qu.:2.500   3rd Qu.:205.0   3rd Qu.:5.000   3rd Qu.:15.5   3rd Qu.:12.50   3rd Qu.:215.0   3rd Qu.: 25.00
 Max.   :160.0   Max.   :4.000   Max.   :3.000   Max.   :240.0   Max.   :6.000   Max.   :17.0   Max.   :14.00   Max.   :260.0   Max.   :100.00
```

2.11 군집분석의 실습

(2) 계층적 군집분석 – 3. 군집 내 비교 (4개 그룹) :
Euclidean distance, Single linkage

```
> summary(cereal[cutree(clustering.ES,k=4)==1,])
    calories        protein           fat            sodium          fiber           carbo           sugars          potass          vitamins
 Min.   :50   Min.   :4   Min.   :0.00   Min.   :130.0   Min.   :10   Min.   :5.00   Min.   :0.0   Min.   :280.0   Min.   :25
 1st Qu.:55   1st Qu.:4   1st Qu.:0.25   1st Qu.:132.5   1st Qu.:11   1st Qu.:5.75   1st Qu.:1.5   1st Qu.:292.5   1st Qu.:25
 Median :60   Median :4   Median :0.50   Median :135.0   Median :12   Median :6.50   Median :3.0   Median :305.0   Median :25
 Mean   :60   Mean   :4   Mean   :0.50   Mean   :135.0   Mean   :12   Mean   :6.50   Mean   :3.0   Mean   :305.0   Mean   :25
 3rd Qu.:65   3rd Qu.:4   3rd Qu.:0.75   3rd Qu.:137.5   3rd Qu.:13   3rd Qu.:7.25   3rd Qu.:4.5   3rd Qu.:317.5   3rd Qu.:25
 Max.   :70   Max.   :4   Max.   :1.00   Max.   :140.0   Max.   :14   Max.   :8.00   Max.   :6.0   Max.   :330.0   Max.   :25
> summary(cereal[cutree(clustering.ES,k=4)==2,])
    calories        protein           fat            sodium          fiber           carbo           sugars          potass          vitamins
 Min.   : 50.0   Min.   :1.000   Min.   :0.000   Min.   :  0.0   Min.   :0.000   Min.   : 8   Min.   : 0.000   Min.   : 15.00   Min.   :  0.00
 1st Qu.:100.0   1st Qu.:2.000   1st Qu.:0.000   1st Qu.:135.0   1st Qu.:0.000   1st Qu.:12   1st Qu.: 3.000   1st Qu.: 40.00   1st Qu.: 25.00
 Median :110.0   Median :2.000   Median :1.000   Median :180.0   Median :1.750   Median :15   Median : 7.000   Median : 90.00   Median : 25.00
 Mean   :109.0   Mean   :2.443   Mean   :1.029   Mean   :159.5   Mean   :1.814   Mean   :15   Mean   : 7.314   Mean   : 90.21   Mean   : 28.21
 3rd Qu.:117.5   3rd Qu.:3.000   3rd Qu.:1.750   3rd Qu.:210.0   3rd Qu.:3.000   3rd Qu.:17   3rd Qu.:11.000   3rd Qu.:118.75   3rd Qu.: 25.00
 Max.   :160.0   Max.   :6.000   Max.   :5.000   Max.   :290.0   Max.   :6.000   Max.   :23   Max.   :15.000   Max.   :260.00   Max.   :100.00
> summary(cereal[cutree(clustering.ES,k=4)==3,])
    calories        protein           fat            sodium          fiber           carbo           sugars          potass          vitamins
 Min.   :70   Min.   :4   Min.   :1   Min.   :260   Min.   :9   Min.   :7   Min.   :5   Min.   :320   Min.   :25
 1st Qu.:70   1st Qu.:4   1st Qu.:1   1st Qu.:260   1st Qu.:9   1st Qu.:7   1st Qu.:5   1st Qu.:320   1st Qu.:25
 Median :70   Median :4   Median :1   Median :260   Median :9   Median :7   Median :5   Median :320   Median :25
 Mean   :70   Mean   :4   Mean   :1   Mean   :260   Mean   :9   Mean   :7   Mean   :5   Mean   :320   Mean   :25
 3rd Qu.:70   3rd Qu.:4   3rd Qu.:1   3rd Qu.:260   3rd Qu.:9   3rd Qu.:7   3rd Qu.:5   3rd Qu.:320   3rd Qu.:25
 Max.   :70   Max.   :4   Max.   :1   Max.   :260   Max.   :9   Max.   :7   Max.   :5   Max.   :320   Max.   :25
> summary(cereal[cutree(clustering.ES,k=4)==4,])
    calories        protein           fat            sodium          fiber           carbo           sugars          potass          vitamins
 Min.   :100   Min.   :3   Min.   :0   Min.   :320   Min.   :1   Min.   :20   Min.   :3   Min.   :45   Min.   :100
 1st Qu.:100   1st Qu.:3   1st Qu.:0   1st Qu.:320   1st Qu.:1   1st Qu.:20   1st Qu.:3   1st Qu.:45   1st Qu.:100
 Median :100   Median :3   Median :0   Median :320   Median :1   Median :20   Median :3   Median :45   Median :100
 Mean   :100   Mean   :3   Mean   :0   Mean   :320   Mean   :1   Mean   :20   Mean   :3   Mean   :45   Mean   :100
 3rd Qu.:100   3rd Qu.:3   3rd Qu.:0   3rd Qu.:320   3rd Qu.:1   3rd Qu.:20   3rd Qu.:3   3rd Qu.:45   3rd Qu.:100
 Max.   :100   Max.   :3   Max.   :0   Max.   :320   Max.   :1   Max.   :20   Max.   :3   Max.   :45   Max.   :100
```

2.11 군집분석의 실습

(2) 계층적 군집분석 – 3. 군집 내 비교 (4개 그룹) :
Euclidean distance, Average linkage

```
> summary(cereal[cutree(clustering.EA,k=4)==1,])
    calories        protein           fat             sodium           fiber           carbo           sugars          potass          vitamins
 Min.   :50   Min.   :4   Min.   :0.00   Min.   :130.0   Min.   :10   Min.   :5.00   Min.   :0.0   Min.   :280.0   Min.   :25
 1st Qu.:55   1st Qu.:4   1st Qu.:0.25   1st Qu.:132.5   1st Qu.:11   1st Qu.:5.75   1st Qu.:1.5   1st Qu.:292.5   1st Qu.:25
 Median :60   Median :4   Median :0.50   Median :135.0   Median :12   Median :6.50   Median :3.0   Median :305.0   Median :25
 Mean   :60   Mean   :4   Mean   :0.50   Mean   :135.0   Mean   :12   Mean   :6.50   Mean   :3.0   Mean   :305.0   Mean   :25
 3rd Qu.:65   3rd Qu.:4   3rd Qu.:0.75   3rd Qu.:137.5   3rd Qu.:13   3rd Qu.:7.25   3rd Qu.:4.5   3rd Qu.:317.5   3rd Qu.:25
 Max.   :70   Max.   :4   Max.   :1.00   Max.   :140.0   Max.   :14   Max.   :8.00   Max.   :6.0   Max.   :330.0   Max.   :25
> summary(cereal[cutree(clustering.EA,k=4)==2,])
    calories        protein          fat            sodium          fiber           carbo           sugars          potass          vitamins
 Min.   : 50.0   Min.   :1.0   Min.   :0.0   Min.   :  0   Min.   :0.00   Min.   : 8.00   Min.   :0.00   Min.   : 15.00   Min.   :  0
 1st Qu.: 82.5   1st Qu.:2.0   1st Qu.:0.0   1st Qu.:  0   1st Qu.:1.25   1st Qu.:13.25   1st Qu.:0.00   1st Qu.: 91.25   1st Qu.:  0
 Median : 90.0   Median :2.5   Median :0.0   Median :  0   Median :2.50   Median :15.00   Median :1.50   Median : 97.50   Median :  0
 Mean   : 86.0   Mean   :2.5   Mean   :0.6   Mean   :  3   Mean   :2.10   Mean   :14.60   Mean   :2.90   Mean   : 95.00   Mean   : 10
 3rd Qu.: 97.5   3rd Qu.:3.0   3rd Qu.:0.0   3rd Qu.:  0   3rd Qu.:3.00   3rd Qu.:16.00   3rd Qu.:5.75   3rd Qu.:117.50   3rd Qu.: 25
 Max.   :120.0   Max.   :4.0   Max.   :5.0   Max.   : 15   Max.   :4.00   Max.   :20.00   Max.   :8.00   Max.   :140.00   Max.   : 25
> summary(cereal[cutree(clustering.EA,k=4)==3,])
    calories        protein          fat            sodium          fiber           carbo           sugars          potass          vitamins
 Min.   : 70.0   Min.   :3.000   Min.   :0.0000   Min.   :190.0   Min.   :4.000   Min.   : 7.00   Min.   : 5.00   Min.   :190.0   Min.   : 25.0
 1st Qu.: 97.5   1st Qu.:3.000   1st Qu.:0.2500   1st Qu.:202.5   1st Qu.:5.000   1st Qu.:11.50   1st Qu.: 6.75   1st Qu.:200.0   1st Qu.: 25.0
 Median :120.0   Median :3.000   Median :1.0000   Median :210.0   Median :5.000   Median :13.50   Median :12.00   Median :235.0   Median : 25.0
 Mean   :110.0   Mean   :3.167   Mean   :0.6667   Mean   :218.3   Mean   :5.667   Mean   :12.33   Mean   :10.33   Mean   :238.3   Mean   : 37.5
 3rd Qu.:120.0   3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:232.5   3rd Qu.:5.750   3rd Qu.:14.00   3rd Qu.:13.50   3rd Qu.:255.0   3rd Qu.: 25.0
 Max.   :140.0   Max.   :4.000   Max.   :1.0000   Max.   :260.0   Max.   :9.000   Max.   :15.00   Max.   :14.00   Max.   :320.0   Max.   :100.0
> summary(cereal[cutree(clustering.EA,k=4)==4,])
    calories        protein          fat            sodium          fiber           carbo           sugars          potass          vitamins
 Min.   : 90.0   Min.   :1.000   Min.   :0.000   Min.   : 45.0   Min.   :0.000   Min.   : 9.00   Min.   : 1.000   Min.   : 20.00   Min.   : 25.0
 1st Qu.:100.0   1st Qu.:2.000   1st Qu.:0.000   1st Qu.:140.0   1st Qu.:0.000   1st Qu.:12.00   1st Qu.: 3.000   1st Qu.: 38.75   1st Qu.: 25.0
 Median :110.0   Median :2.000   Median :1.000   Median :180.0   Median :1.000   Median :15.00   Median : 8.000   Median : 62.50   Median : 25.0
 Mean   :112.1   Mean   :2.393   Mean   :1.125   Mean   :185.8   Mean   :1.464   Mean   :15.30   Mean   : 7.661   Mean   : 76.79   Mean   : 31.7
 3rd Qu.:110.0   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:220.0   3rd Qu.:2.625   3rd Qu.:17.25   3rd Qu.:11.000   3rd Qu.:106.25   3rd Qu.: 25.0
 Max.   :160.0   Max.   :6.000   Max.   :3.000   Max.   :320.0   Max.   :5.000   Max.   :23.00   Max.   :15.000   Max.   :200.00   Max.   :100.0
```

2.11 군집분석의 실습

(2) 계층적 군집분석 – 4. 군집 비교 (4개 그룹)

```
> Clu.Ave=cutree(clustering.EA,k=4)
> Clu.Sig=cutree(clustering.ES,k=4)
> Clu.Com=cutree(clustering.EC,k=4)
> table(Clu.Ave, Clu.Sig)
       Clu.Sig
Clu.Ave  1  2  3  4
      1  2  0  0  0
      2  0 10  0  0
      3  0  5  1  0
      4  0 55  0  1
> table(Clu.Ave, Clu.Com)
       Clu.Com
Clu.Ave  1  2  3  4
      1  2  0  0  0
      2  0 10  0  0
      3  1  0  0  5
      4  0  8 42  6
> table(Clu.Sig, Clu.Com)
       Clu.Com
Clu.Sig  1  2  3  4
      1  2  0  0  0
      2  0 18 41 11
      3  1  0  0  0
      4  0  0  1  0
```

2.11 군집분석의 실습

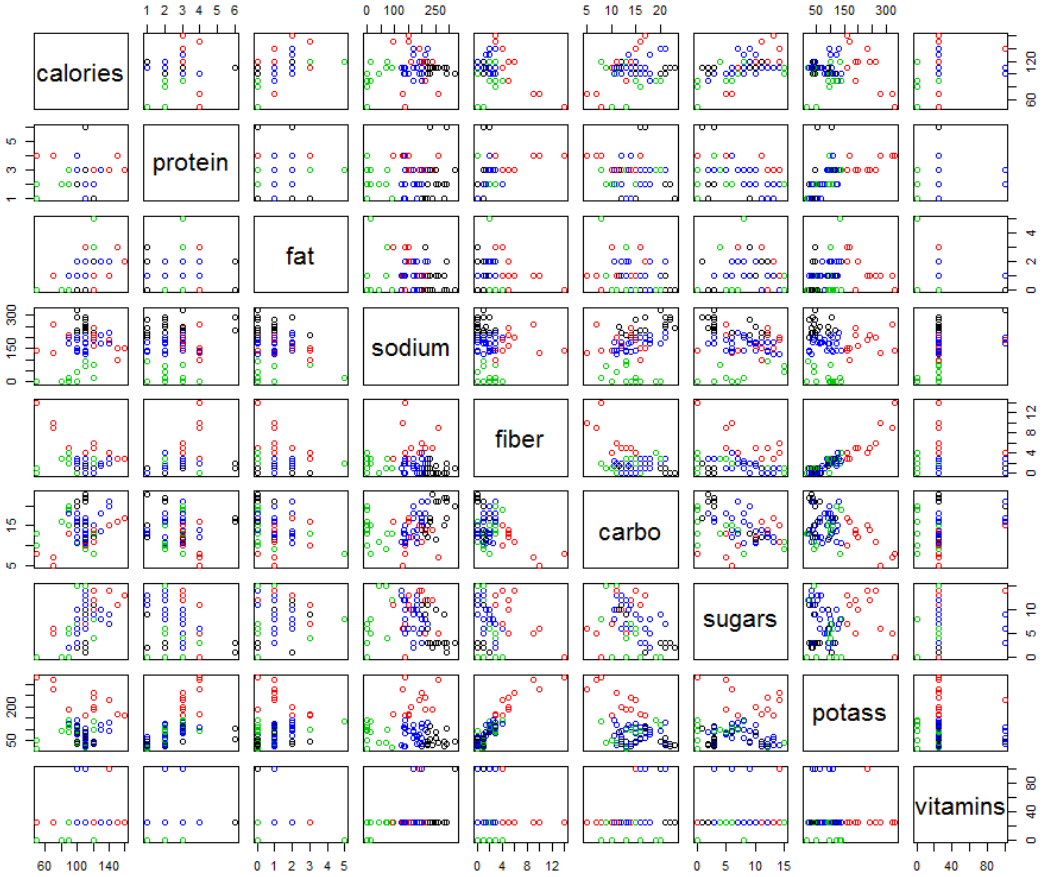
(3) K-means 군집분석 – 1. K-means 결과

```
> library(cluster)
> clustering.K4 <- kmeans(cereal, 4)
> clustering.K4
K-means clustering with 4 clusters of sizes 18, 13, 14, 29

Cluster means:
   calories  protein       fat    sodium     fiber    carbo   sugars    potass vitamins
1 109.44444 2.277778 0.8333333 250.00000 0.8055556 17.41667 5.388889  52.77778 29.16667
2 113.07692 3.384615 1.3846154 175.00000 5.8461538 12.15385 9.230769 223.07692 30.76923
3  92.85714 2.357143 0.7142857  22.14286 1.8571429 13.71429 5.357143  82.14286 14.28571
4 109.65517 2.344828 1.0689655 170.00000 1.5344828 14.70690 8.068966  78.96552 35.34483

Clustering vector:
                          100%_Bran                   100%_Natural_Bran                           All-Bran
                                  2                                   3                                  2
          All-Bran_with_Extra_Fiber               Apple_Cinnamon_Cheerios                        Apple_Jacks
                                  2                                   4                                  4
                            Basic_4                          Bran_Chex                        Bran_Flakes
                                  4                                   4                                  2
                       Cap'n'Crunch                           Cheerios                Cinnamon_Toast_Crunch
                                  1                                   1                                  1
                           Clusters                         Cocoa_Puffs                          Corn_Chex
                                  4                                   4                                  1
                         Corn_Flakes                           Corn_Pops                      Count_Chocula
                                  1                                   3                                  4
                  Cracklin'_Oat_Bran                             Crispix                Crispy_Wheat_&_Raisins
                                  2                                   1                                  4
                        Double_Chex                          Froot_Loops                      Frosted_Flakes
                                  4                                   4                                  1
                Frosted_Mini-Wheats Fruit_&_Fibre_Dates,_Walnuts,_and_Oats                     Fruitful_Bran
                                  3                                   2                                  2
                     Fruity_Pebbles                        Golden_Crisp                       Golden_Grahams
                                  4                                   3                                  1
                   Grape_Nuts_Flakes                           Grape-Nuts                    Great_Grains_Pecan
                                  4                                   4                                  3

Within cluster sum of squares by cluster:
[1] 40148.14 88980.00 45411.86 80466.07
 (between_SS / total_SS =  72.7 %)

Available components:

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

2.11 군집분석의 실습

(3) K-means 군집분석 – 1. K-means 결과

```
plot(cereal, col=clustering.K4$cluster)
```



2.11 군집분석의 실습

(3) K-means 군집분석 – 2. 초기값에 따른 군집분석의 차이

Set.seed(1)과 set.seed(2)로 초기값을 다르게 주고 두 결과를 비교한다.

```
> set.seed(1)
> clustering1 <- kmeans(cereal,4)
> set.seed(2)
> clustering2 <- kmeans(cereal,4)
> table(clustering1$cluster, clustering2$cluster)

   1  2  3  4
1  0 16  0  5
2  4  0  0 26
3  0  0 14  0
4  9  0  0  0
```

2.11 군집분석의 실습

(3) K-means 군집분석 – 2. 초기값에 따른 군집분석의 차이

Set.seed(1)과 set.seed(2)로 초기값을 다르게 주고 두 결과를 비교한다.

```
> clustering1
K-means clustering with 4 clusters of sizes 21, 30, 14, 9

Cluster means:
   calories  protein       fat    sodium     fiber    carbo   sugars    potass vitamins
1 110.00000 2.238095 0.8571429 242.38095 0.7380952 17.40476 5.571429  51.66667 32.14286
2 113.66667 2.533333 1.3000000 162.50000 1.8833333 14.45000 8.533333  93.83333 32.50000
3  92.85714 2.357143 0.7142857  22.14286 1.8571429 13.71429 5.357143  82.14286 14.28571
4 100.00000 3.333333 0.7777778 193.33333 7.0000000 11.00000 8.666667 248.88889 33.33333


> clustering2
K-means clustering with 4 clusters of sizes 13, 16, 14, 31

Cluster means:
   calories  protein       fat    sodium    fiber    carbo   sugars    potass vitamins
1 113.07692 3.384615 1.3846154 175.00000 5.846154 12.15385 9.230769 223.07692 30.76923
2 110.62500 2.250000 0.8125000 253.12500 0.593750 17.59375 5.500000  46.56250 29.68750
3  92.85714 2.357143 0.7142857  22.14286 1.857143 13.71429 5.357143  82.14286 14.28571
4 109.03226 2.354839 1.0645161 173.54839 1.596774 14.79032 7.838710  80.48387 34.67742
```

2.11 군집분석의 실습

(4) 계층적 군집분석과 K-means 군집분석의 결과 비교

계층 (complete)

```
> rownames(cereal[cutree(clustering.EC,k=4)==1,])
[1] "100%_Bran"                "All-Bran"                 "All-Bran_with_Extra_Fiber"
> rownames(cereal[cutree(clustering.EC,k=4)==2,])
 [1] "100%_Natural_Bran"       "Apple_Jacks"              "Corn_Pops"                "Froot_Loops"              "Frosted_Mini-Wheats"
 [6] "Fruity_Pebbles"          "Golden_Crisp"             "Great_Grains_Pecan"       "Maypo"                    "Puffed_Rice"
[11] "Puffed_Wheat"            "Raisin_Squares"           "Shredded_Wheat"           "Shredded_Wheat_'n'Bran"   "Shredded_Wheat_spoon_size"
[16] "Smacks"                  "Strawberry_Fruit_Wheats"  "Trix"
> rownames(cereal[cutree(clustering.EC,k=4)==3,])
 [1] "Apple_Cinnamon_Cheerios"  "Basic_4"                  "Bran_Chex"                "Cap'n'Crunch"             "Cheerios"
 [6] "Cinnamon_Toast_Crunch"    "Clusters"                 "Cocoa_Puffs"              "Corn_Chex"                "Corn_Flakes"
[11] "Count_Chocula"            "Crispix"                  "Crispy_Wheat_&_Raisins"   "Double_Chex"              "Frosted_Flakes"
[16] "Golden_Grahams"           "Grape_Nuts_Flakes"        "Grape-Nuts"               "Honey_Graham_Ohs"         "Honey_Nut_Cheerios"
[21] "Honey-comb"               "Just_Right_Crunchy__Nuggets" "Just_Right_Fruit_&_Nut" "Kix"                      "Life"
[26] "Lucky_Charms"             "Multi-Grain_Cheerios"     "Nut&Honey_Crunch"         "Nutri-Grain_Almond-Raisin" "Nutri-grain_Wheat"
[31] "Oatmeal_Raisin_Crisp"     "Product_19"               "Quaker_Oat_Squares"       "Rice_Chex"                "Rice_Krispies"
[36] "Special_K"                "Total_Corn_Flakes"        "Total_Whole_Grain"        "Triples"                  "Wheat_Chex"
[41] "Wheaties"                 "Wheaties_Honey_Gold"
> rownames(cereal[cutree(clustering.EC,k=4)==4,])
 [1] "Bran_Flakes"              "Cracklin'_Oat_Bran"       "Fruit_&_Fibre_Dates,_Walnuts,_and_Oats"
 [4] "Fruitful_Bran"            "Muesli_Raisins,_Dates,_&_Almonds" "Muesli_Raisins,_Peaches,_&_Pecans"
 [7] "Mueslix_Crispy_Blend"     "Post_Nat._Raisin_Bran"    "Raisin_Bran"
[10] "Raisin_Nut_Bran"          "Total_Raisin_Bran"
```

군집

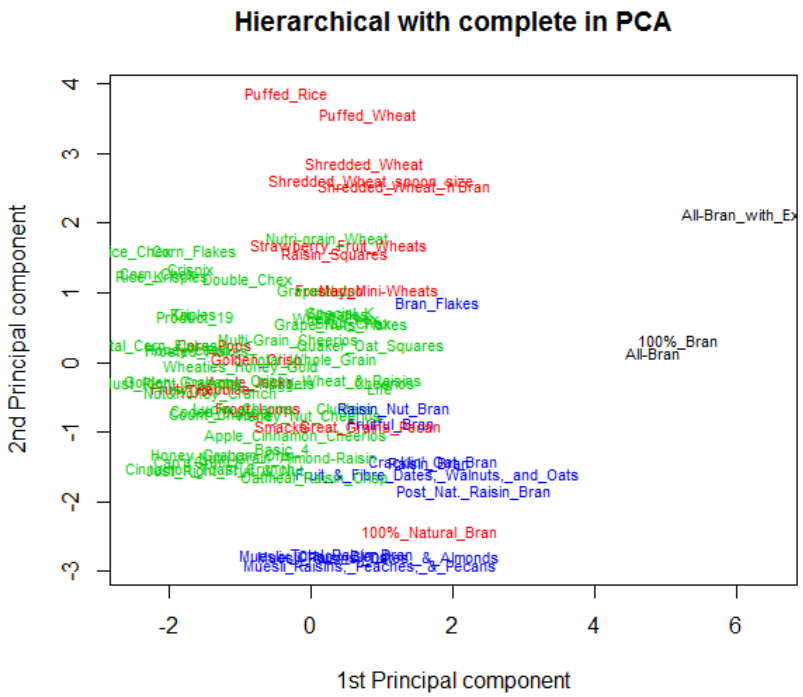
```
> rownames(cereal[clustering.K4$cluster==1,])
 [1] "Cap'n'Crunch"             "Cheerios"                 "Cinnamon_Toast_Crunch"    "Corn_Chex"                "Corn_Flakes"              "Crispix"
 [7] "Frosted_Flakes"           "Golden_Grahams"           "Honey_Graham_Ohs"         "Honey_Nut_Cheerios"       "Kix"                      "Multi-Grain_Cheerios"
[13] "Product_19"               "Rice_Chex"                "Rice_Krispies"            "Special_K"                "Triples"                  "Wheat_Chex"
> rownames(cereal[clustering.K4$cluster==2,])
 [1] "100%_Bran"                "All-Bran"                 "All-Bran_with_Extra_Fiber"
 [4] "Bran_Flakes"              "Cracklin'_Oat_Bran"       "Fruit_&_Fibre_Dates,_Walnuts,_and_Oats"
 [7] "Fruitful_Bran"            "Muesli_Raisins,_Dates,_&_Almonds" "Muesli_Raisins,_Peaches,_&_Pecans"
[10] "Mueslix_Crispy_Blend"     "Post_Nat._Raisin_Bran"    "Raisin_Bran"
[13] "Total_Raisin_Bran"
> rownames(cereal[clustering.K4$cluster==3,])
 [1] "100%_Natural_Bran"        "Corn_Pops"                "Frosted_Mini-Wheats"      "Golden_Crisp"             "Great_Grains_Pecan"
 [6] "Maypo"                    "Puffed_Rice"              "Puffed_Wheat"             "Raisin_Squares"           "Shredded_Wheat"
[11] "Shredded_Wheat_'n'Bran"   "Shredded_Wheat_spoon_size" "Smacks"                  "Strawberry_Fruit_Wheats"
> rownames(cereal[clustering.K4$cluster==4,])
 [1] "Apple_Cinnamon_Cheerios"  "Apple_Jacks"              "Basic_4"                  "Bran_Chex"                "Clusters"
 [6] "Cocoa_Puffs"              "Count_Chocula"            "Crispy_Wheat_&_Raisins"   "Double_Chex"              "Froot_Loops"
[11] "Fruity_Pebbles"           "Grape_Nuts_Flakes"        "Grape-Nuts"               "Honey-comb"               "Just_Right_Crunchy__Nuggets"
[16] "Just_Right_Fruit_&_Nut"   "Life"                     "Lucky_Charms"             "Nut&Honey_Crunch"         "Nutri-Grain_Almond-Raisin"
[21] "Nutri-grain_Wheat"        "Oatmeal_Raisin_Crisp"     "Quaker_Oat_Squares"       "Raisin_Nut_Bran"          "Total_Corn_Flakes"
[26] "Total_Whole_Grain"        "Trix"                     "Wheaties"                 "Wheaties_Honey_Gold"
```

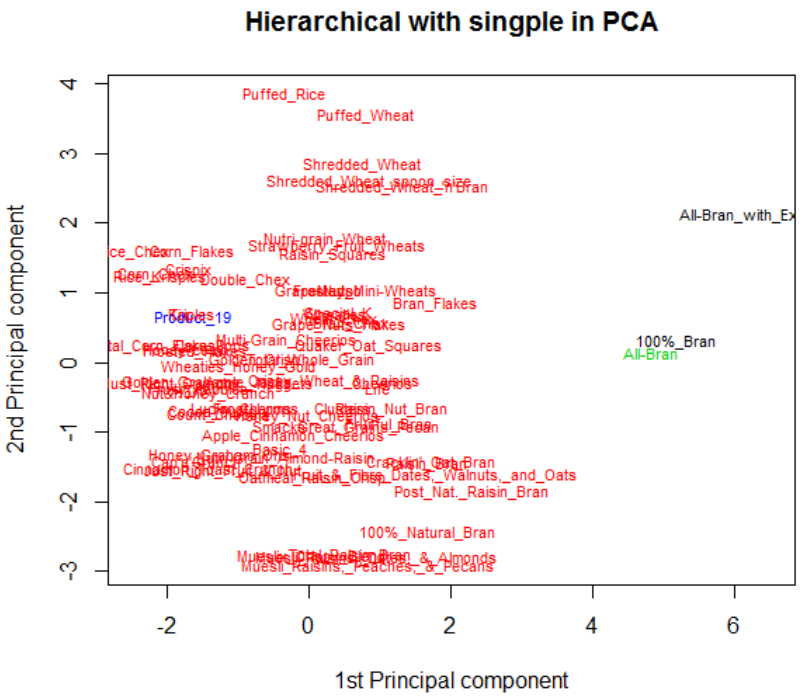
2.11 군집분석의 실습

(5) 계층적 군집분석과 K-means 군집분석 결과를 PCA 평면상에서 확인

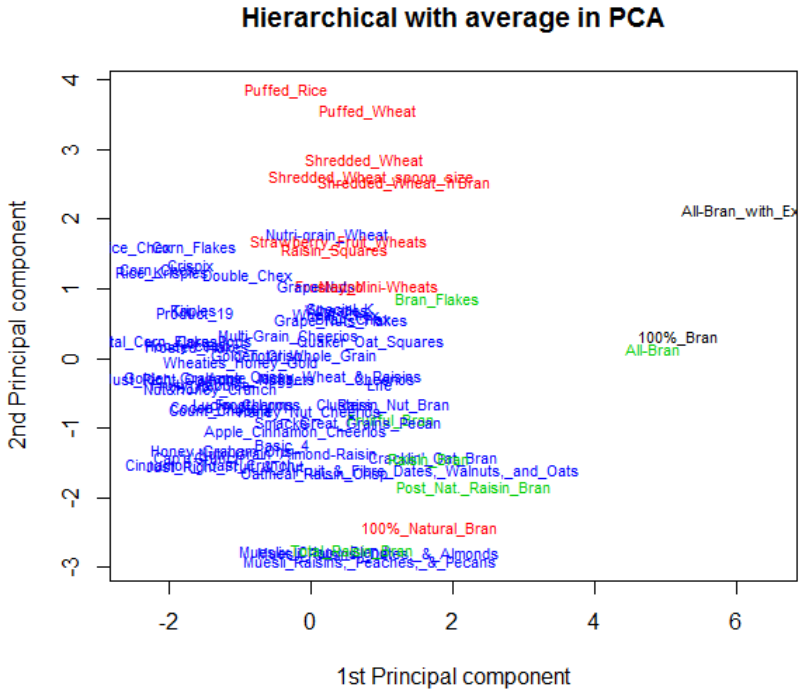
```
> fit <- princomp(cereal, cor=TRUE)
> x=fit$scores[,1]
> y=fit$scores[,2]
> #1. Hierarchical
> plot(x, y, xlab="1st Principal component", ylab="2nd Principal component", main="Hierarchical with
 complete in PCA",       type="n")
> text(x, y, labels = row.names(cereal), cex=.7, col=cutree(clustering.EC, k=4))
> plot(x, y, xlab="1st Principal component", ylab="2nd Principal component", main="Hierarchical with
 singple in PCA",        type="n")
> text(x, y, labels = row.names(cereal), cex=.7, col=cutree(clustering.ES, k=4))
> plot(x, y, xlab="1st Principal component", ylab="2nd Principal component", main="Hierarchical with
 average in PCA",        type="n")
> text(x, y, labels = row.names(cereal), cex=.7, col=cutree(clustering.EA, k=4))
> #2. K-means
> plot(x, y, xlab="1st Principal component", ylab="2nd Principal component", main="K-means in PCA",
       type="n")
> text(x, y, labels = row.names(cereal), cex=.7, col=clustering.K4$cluster)
```

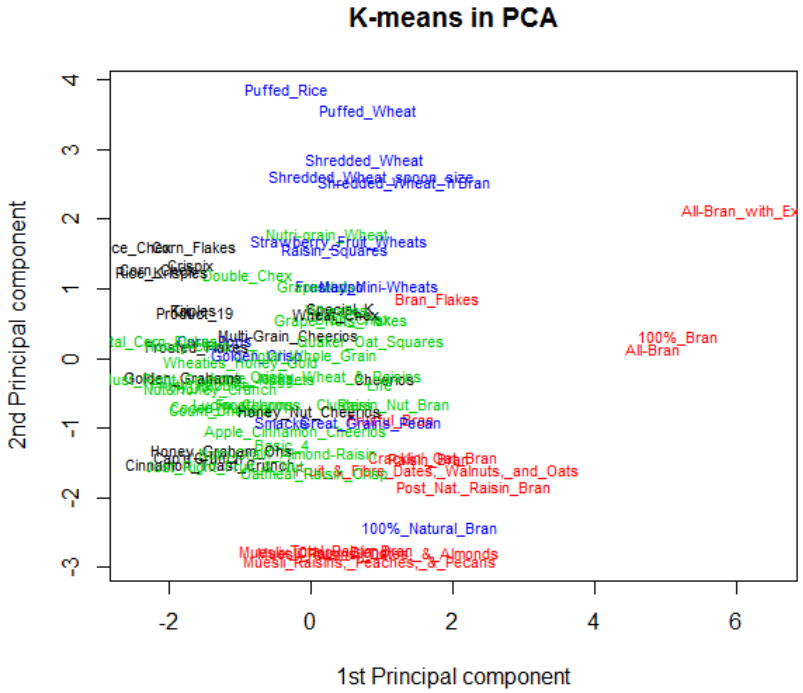
2.11 군집분석의 실습


Hierarchical with complete in PCA
2nd Principal component
1st Principal component
Puffed_Rice
Puffed_Wheat
Shredded_Wheat
Shredded_Wheat_spoon_size
Shredded_Wheat_n'Bran
All-Bran_with_Ex
Nutri-grain_Wheat
Raisin_Squares
Bran_Flakes
100%_Bran
All-Bran
Multi-Grain_Cheerios
Quaker_Oat_Squares
Wheaties_Honey_Gold
Apple_Cinnamon_Cheerios
Basic_4
Post_Nat._Raisin_Bran
100%_Natural_Bran


Hierarchical with singple in PCA
2nd Principal component
1st Principal component
Puffed_Rice
Puffed_Wheat
Shredded_Wheat
Shredded_Wheat_spoon_size
Shredded_Wheat_n'Bran
All-Bran_with_Ex
Nutri-grain_Wheat
Raisin_Squares
Bran_Flakes
100%_Bran
All-Bran
Multi-Grain_Cheerios
Quaker_Oat_Squares
Wheaties_Honey_Gold
Apple_Cinnamon_Cheerios
Basic_4
Post_Nat._Raisin_Bran
100%_Natural_Bran

2.11 군집분석의 실습


Hierarchical with average in PCA
2nd Principal component
1st Principal component
Puffed_Rice
Puffed_Wheat
Shredded_Wheat
Shredded_Wheat_spoon_size
Shredded_Wheat_n'Bran
All-Bran_with_Ex
Nutri-grain_Wheat
Raisin_Squares
Double_Chex
Bran_Flakes
Multi-Grain_Cheerios
Quaker_Oat_Squares
100%_Bran
All-Bran
Wheaties_Honey_Gold
Honey_Nut_Cheerios
Apple_Cinnamon_Cheerios
Basic_4
Post_Nat._Raisin_Bran
100%_Natural_Bran

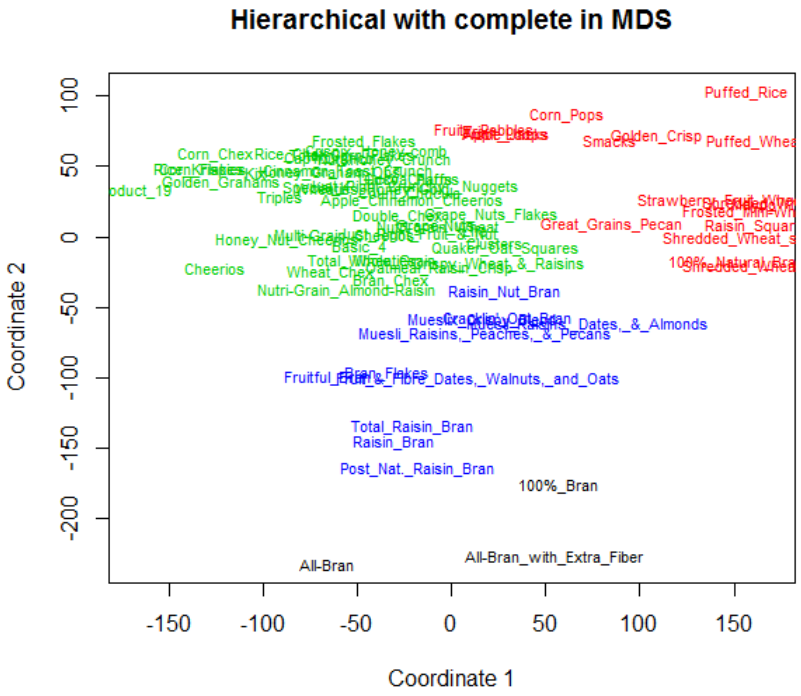

K-means in PCA
2nd Principal component
1st Principal component
Puffed_Rice
Puffed_Wheat
Shredded_Wheat
Shredded_Wheat_spoon_size
Shredded_Wheat_n'Bran
All-Bran_with_Ex
Nutri-grain_Wheat
Raisin_Squares
Double_Chex
Bran_Flakes
Multi-Grain_Cheerios
Quaker_Oat_Squares
100%_Bran
All-Bran
Wheaties_Honey_Gold
Honey_Nut_Cheerios
Apple_Cinnamon_Cheerios
Basic_4
Post_Nat._Raisin_Bran
100%_Natural_Bran

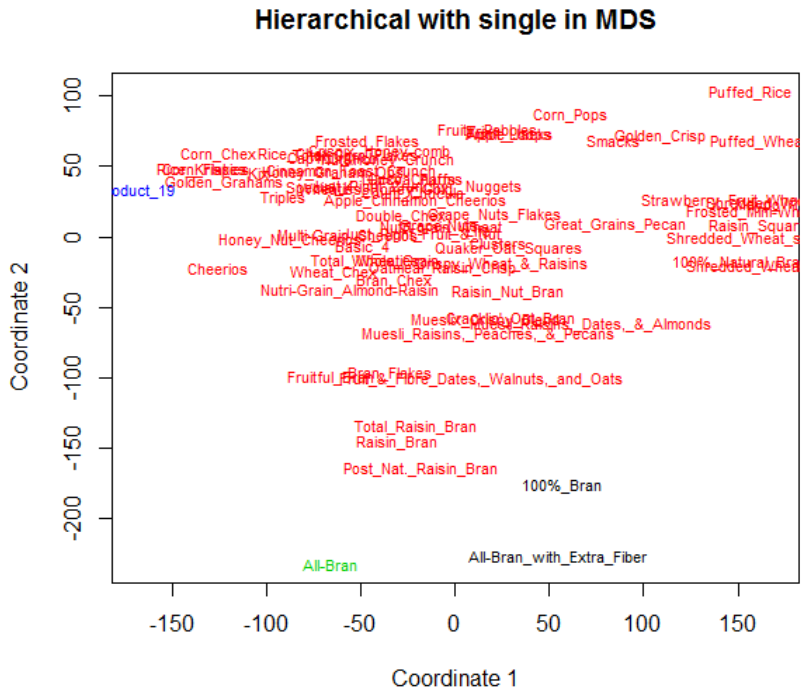
2.11 군집분석의 실습

(6) 계층적 군집분석과 K-means 군집분석 결과를 MDS 평면상에서 확인

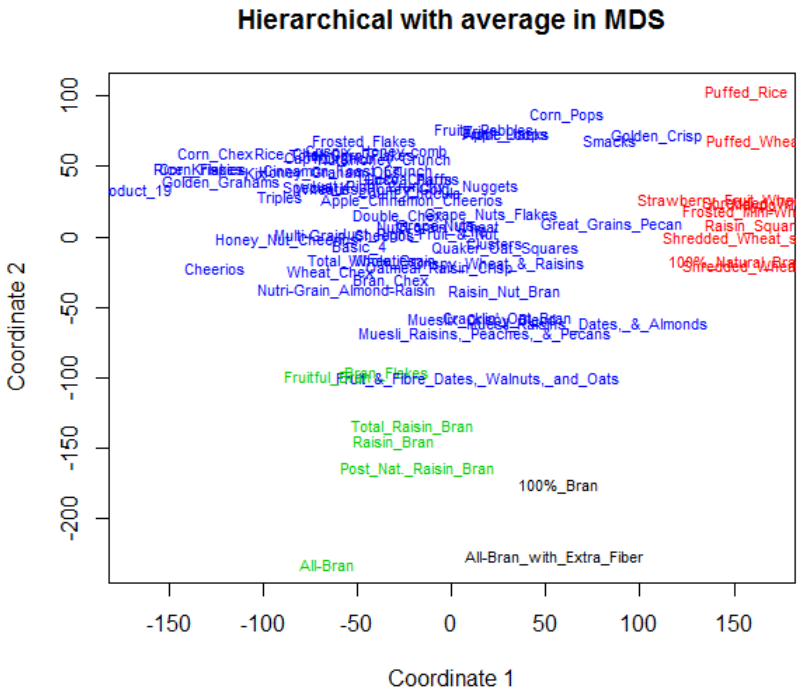
```
> cereal.d <- dist(cereal)
> fit <- cmdscale(cereal.d,eig=TRUE, k=2, add=TRUE)
> x <- fit$points[,1]
> y <- fit$points[,2]
> #1. Hierarchical
> plot(x, y, xlab="Coordinate 1", ylab="Coordinate 2", main="Hierarchical with complete in MDS",
       type="n")
> text(x, y, labels = row.names(cereal), cex=.7, col=cutree(clustering.EC, k=4))
> plot(x, y, xlab="Coordinate 1", ylab="Coordinate 2", main="Hierarchical with single in MDS",  type
="n")
> text(x, y, labels = row.names(cereal), cex=.7, col=cutree(clustering.ES, k=4))
> plot(x, y, xlab="Coordinate 1", ylab="Coordinate 2", main="Hierarchical with average in MDS", type
="n")
> text(x, y, labels = row.names(cereal), cex=.7, col=cutree(clustering.EA, k=4))
> #2. K-means
> plot(x, y, xlab="Coordinate 1", ylab="Coordinate 2", main="K-means in MDS",   type="n")
> text(x, y, labels = row.names(cereal), cex=.7, col=clustering.K4$cluster)
```

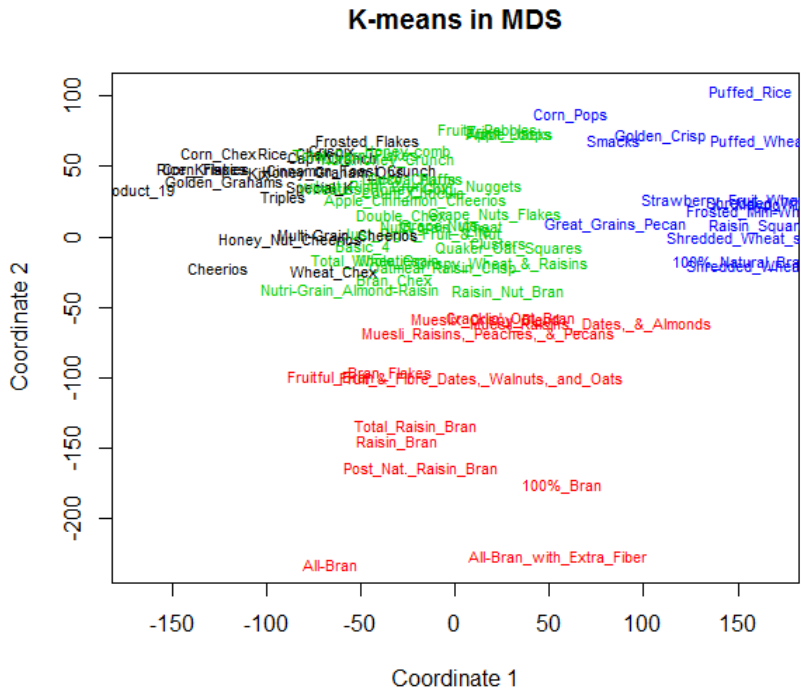
2.11 군집분석의 실습


Hierarchical with complete in MDS
Coordinate 2
Coordinate 1
Puffed_Rice
Corn_Pops
Smacks
Golden_Crisp
Puffed_Wheat
Frosted_Flakes
Corn_Chex
Triples
Apple_Cinnamon_Cheerios
Double_Chex
Grape_Nuts_Flakes
Great_Grains_Pecan
Raisin_Squares
Shredded_Wheat_s
Honey_Nut_Cheerios
Basic_4
Quaker_Oat_Squares
Cheerios
Wheat_Chex
Oatmeal_Raisin_Crisp
Bran_Chex
Nutri-Grain_Almond-Raisin
Raisin_Nut_Bran
Muesli_Raisins,_Peaches,_&_Pecans
Fruitful_Bran
Dates,_Walnuts,_and_Oats
Total_Raisin_Bran
Raisin_Bran
Post_Nat._Raisin_Bran
100%_Bran
All-Bran
All-Bran_with_Extra_Fiber


Hierarchical with single in MDS
Coordinate 2
Coordinate 1
Puffed_Rice
Corn_Pops
Smacks
Golden_Crisp
Frosted_Flakes
Corn_Chex
Triples
Apple_Cinnamon_Cheerios
Great_Grains_Pecan
Honey_Nut_Cheerios
Basic_4
Quaker_Oat_Squares
Cheerios
Wheat_Chex
Bran_Chex
Nutri-Grain_Almond-Raisin
Raisin_Nut_Bran
Muesli_Raisins,_Peaches,_&_Pecans
Fruitful_Bran
Dates,_Walnuts,_and_Oats
Total_Raisin_Bran
Raisin_Bran
Post_Nat._Raisin_Bran
100%_Bran
All-Bran
All-Bran_with_Extra_Fiber

2.11 군집분석의 실습


Hierarchical with average in MDS
Coordinate 2
Coordinate 1
Puffed_Rice
Corn_Pops
Smacks
Golden_Crisp
Frosted_Flakes
Corn_Chex
Triples
Apple_Cinnamon_Cheerios
Double_Chex
Grape_Nuts_Flakes
Great_Grains_Pecan
Honey_Nut_Cheerios
Basic_4
Quaker_Oats_Squares
Cheerios
Wheat_Chex
Bran_Chex
Nutri-Grain_Almond-Raisin
Raisin_Nut_Bran
Muesli_Raisins,_Peaches,_&_Pecans
Fruitful_Bran
Fruit_&_Fibre_Dates,_Walnuts,_and_Oats
Total_Raisin_Bran
Raisin_Bran
Post_Nat._Raisin_Bran
100%_Bran
All-Bran
All-Bran_with_Extra_Fiber


K-means in MDS
Coordinate 2
Coordinate 1
Puffed_Rice
Corn_Pops
Smacks
Golden_Crisp
Frosted_Flakes
Triples
Apple_Cinnamon_Cheerios
Great_Grains_Pecan
Honey_Nut_Cheerios
Cheerios
Wheat_Chex
Nutri-Grain_Almond-Raisin
Raisin_Nut_Bran
Muesli_Raisins,_Peaches,_&_Pecans
Fruitful_Bran
Fruit_&_Fibre_Dates,_Walnuts,_and_Oats
Total_Raisin_Bran
Raisin_Bran
Post_Nat._Raisin_Bran
100%_Bran
All-Bran
All-Bran_with_Extra_Fiber