제7강. 군집분석(2)

- R 계층적 군집분석 사례
- 파이썬 계층적 군집분석 사례
- K-평균 군집분석
- K-평균 군집분석 사례 : R
- K-평균 군집분석 사례 : 파이썬

1. R 군집분석 사례분석 2

1) 데이터 설명

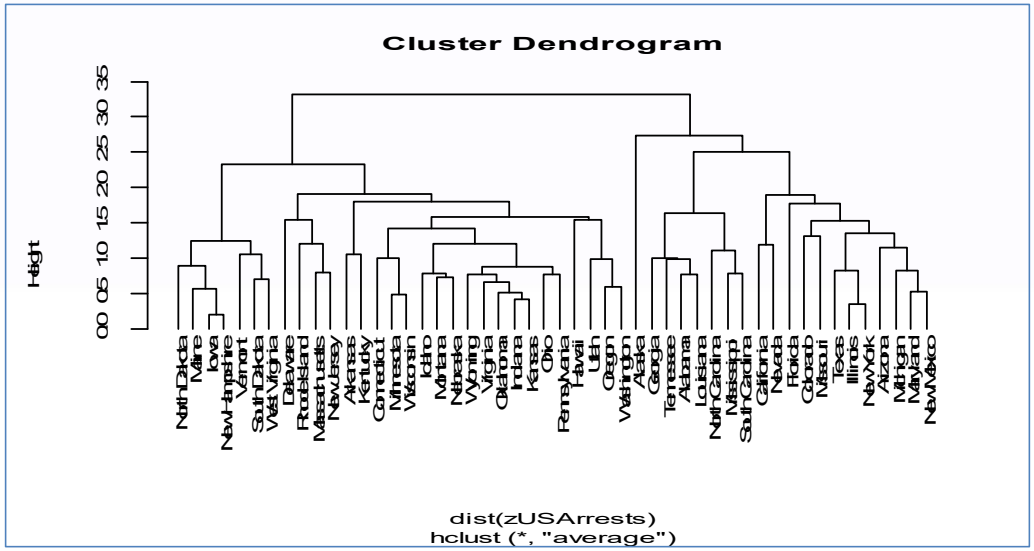
- 미국 50개 주별로 인구 10만명당 각종 범죄로 인한 체포자의 수가 기록된 데이터 (USArrests)
- R 내장데이터

```
> head(USArrests)
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
> summary(USArrests)
     Murder          Assault         UrbanPop          Rape
 Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30
 1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07
 Median : 7.250   Median :159.0   Median :66.00   Median :20.10
 Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23
 3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18
 Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00
```

1. R 군집분석 사례분석 2

2) 계층적 군집분석 실행하기 – 평균연결법

```
> zUSArrests=scale(USArrests)
> hc_a = hclust(dist(zUSArrests), method="average")
> hc_a
Call:
hclust(d = dist(zUSArrests), method = "average")
Cluster method   : average
Distance         : euclidean
Number of objects: 50
> plot(hc_a, hang=-1)
```



1. R 군집분석 사례분석 2

3) 소속 군집 알기

```
> hcmember <- cutree(hc_a, k=5)
> hcmember
     Alabama       Alaska      Arizona     Arkansas   California
           1            1            1            2            1
    Colorado  Connecticut     Delaware      Florida      Georgia
           2            3            1            4            2
      Hawaii        Idaho     Illinois      Indiana         Iowa
           3            3            1            3            3
      Kansas     Kentucky    Louisiana        Maine     Maryland
           3            3            1            3            1
Massachusetts     Michigan    Minnesota  Mississippi     Missouri
           2            1            3            1            2
     Montana     Nebraska       Nevada New Hampshire   New Jersey
           3            3            1            3            2
  New Mexico     New York North Carolina North Dakota         Ohio
           1            1            4            3            3
    Oklahoma       Oregon Pennsylvania Rhode Island South Carolina
           2            2            3            2            1
South Dakota    Tennessee        Texas         Utah      Vermont
           3            2            2            3            3
    Virginia   Washington West Virginia    Wisconsin      Wyoming
           2            2            3            3            2
>
```

1. R 군집분석 사례분석 2

4) 각 군집별 중심점 찾기

```
> data_combined = cbind(USArrests, hcmember)
> aggregate(.~hcmember, data_combined, mean)
  hcmember     Murder   Assault  UrbanPop     Rape
1        1 14.671429 251.28571 54.28571 21.685714
2        2 10.000000 263.00000 48.00000 44.500000
3        3 10.883333 256.91667 78.33333 32.250000
4        4  5.530435 129.43478 68.91304 17.786957
5        5  2.700000  65.14286 46.28571  9.885714
```

1번 군집의 경우, Murder'의 범죄체포수가 가장 많은 군집

5번 군집의 경우 모든 변수에서 범죄체포수가 가장 적은 군집

2. 파이썬 군집분석 사례분석 : beer 데이터

1) 데이터 읽기

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# 데이터 읽기
beer = pd.read_csv("c:/data/mva/beerbrand.csv", index_col='name')
beer.head()
Out[6]:
             calories  sodium  alcohol  cost
name
BUDWEISER         144      15      4.7  0.43
SCHLITZ           151      19      4.9  0.43
LOWENBRAU         157      15      4.9  0.48
KRONENBOURG       170       7      5.2  0.73
HEINEKEN          152      11      5.0  0.77
# 기술통계량 구하기
beer.describe()
```

2. 파이썬 군집분석 사례분석 : beer 데이터

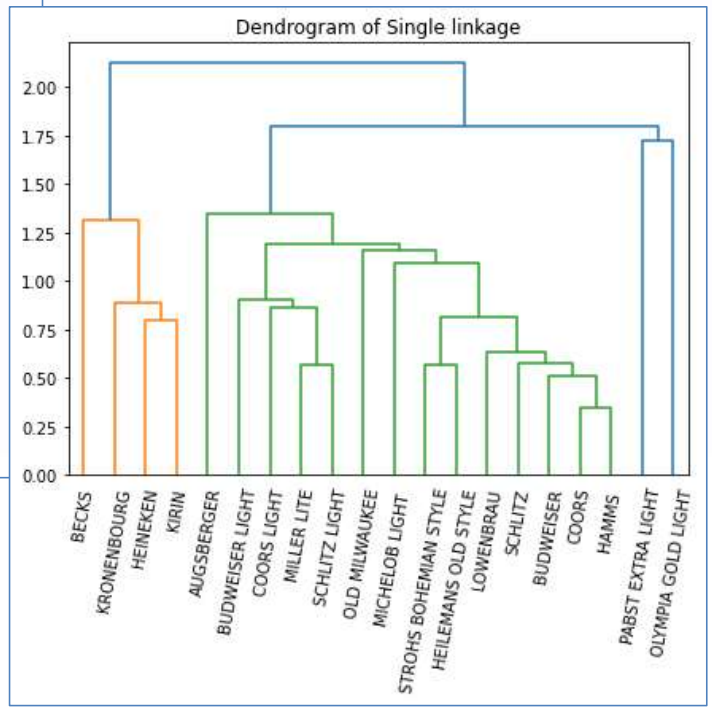
2) 데이터 표준화

```
# 표준화 패키기 불러오기
from sklearn.preprocessing import StandardScaler
# 표준화 시행
zbeer = StandardScaler().fit_transform(beer)
type(zbeer)
Out[12]: numpy.ndarray
zbeer_frame = pd.DataFrame(zbeer)
zbeer_frame.columns = beer.columns
zbeer_frame.describe()
Out[17]:
              calories         sodium        alcohol           cost
count  2.000000e+01  2.000000e+01  2.000000e+01  2.000000e+01
mean  -1.110223e-16  7.771561e-17 -1.634803e-15 -3.747003e-16
std    1.025978e+00  1.025978e+00  1.025978e+00  1.025978e+00
min   -2.183691e+00 -1.395248e+00 -2.889782e+00 -1.545138e+00
25%   -7.510676e-01 -8.496201e-01 -3.240877e-01 -4.746037e-01
50%    3.933356e-01  7.794680e-03  2.835767e-01 -4.032347e-01
75%    6.052621e-01  6.313691e-01  6.211680e-01 -8.207432e-02
max    1.444491e+00  1.878518e+00  1.431387e+00  2.094679e+00
```

2. 파이썬 군집분석 사례분석 : beer 데이터

3) 계층적 군집분석 - 최단연결법

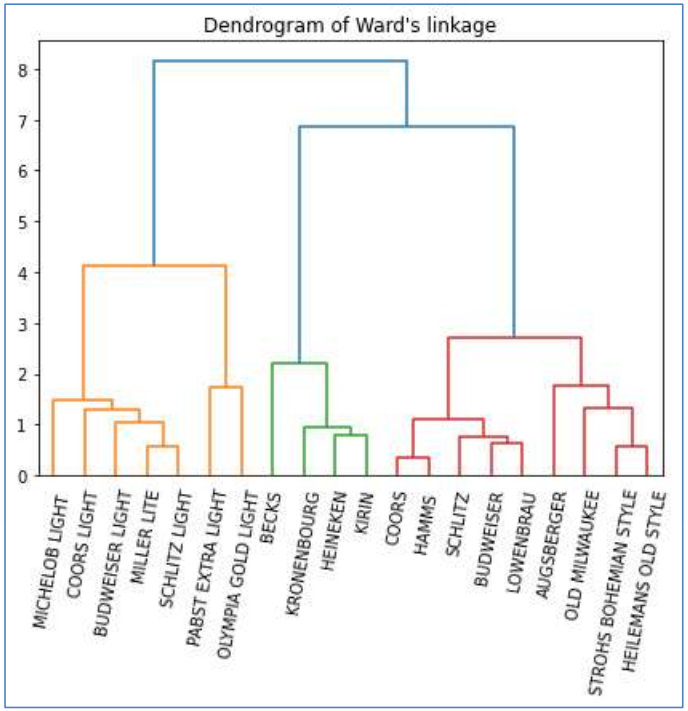
```
# 패키지 불러오기
import scipy.cluster.hierarchy as sch
# 계층적 군집분석 시행하기: 최단연결법
slink = sch.linkage(zbeer, 'single')
# method = 'single', 'complete', 'average', 'median', 'ward'

plt.figure(figsize=(7,5))
sch.dendrogram(slink,
        leaf_rotation=80,
        leaf_font_size=10,
        labels = beer.index
)
plt.title("Dendrogram of Single linkage")
plt.show()
```



2. 파이썬 군집분석 사례분석 : beer 데이터

4) 계층적 군집분석 - 와드연결법

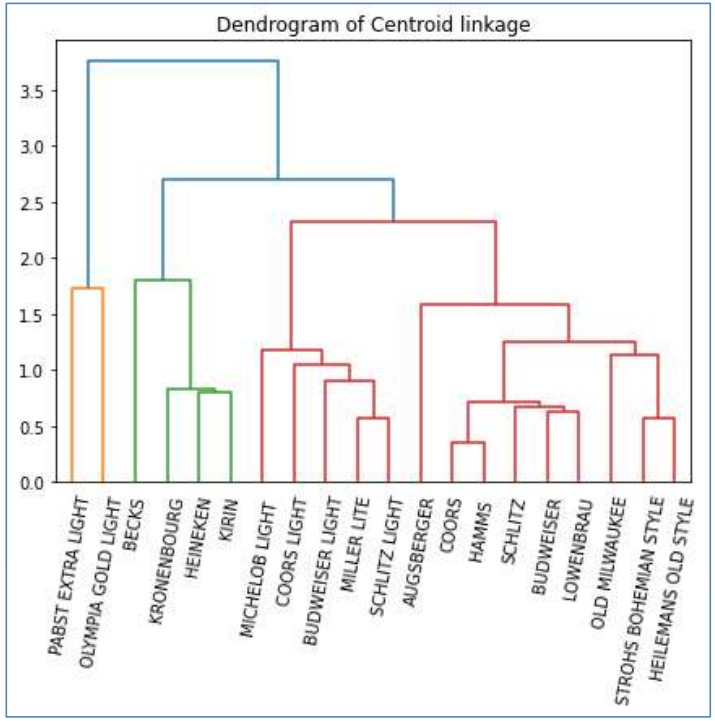
```
# 계층적 군집분석 시행: 와드의 방법
wlink = sch.linkage(zbeer, 'ward')

plt.figure(figsize=(7,5))
sch.dendrogram(wlink,
 leaf_rotation=80,
 leaf_font_size=10,
 labels = beer.index
)
plt.title("Dendrogram of Ward's linkage")
plt.show()
```



2. 파이썬 군집분석 사례분석 : beer 데이터

5) 계층적 군집분석 - 중심연결법

```
# 계층적 군집분석: 중심연결법
clink = sch.linkage(zbeer, 'centroid')

plt.figure(figsize=(7,5))
sch.dendrogram(clink,
 leaf_rotation=80,
 leaf_font_size=10,
 labels = beer.index
)
plt.title("Dendrogram of Centroid linkage")
plt.show()
```



2. 파이썬 군집분석 사례분석 : beer 데이터

6) 소속 군집 알기

```
from sklearn.cluster import AgglomerativeClustering
# help(AgglomerativeClustering): 클래스 코드 보기

# 계층적 군집분석: ward 방법
wcluster = AgglomerativeClustering(n_clusters=4, affinity='euclidean',
 linkage='ward')

# linkage: {"ward", "complete", "average", "single"}, default="ward"

# 소속군집
member = wcluster.fit_predict(zbeer)
member
Out[31]:
array([0, 0, 0, 2, 2, 0, 0, 0, 3, 3, 0, 3, 3, 2, 2, 1, 0, 0, 1, 3],
    dtype=int64)
```

2. 파이썬 군집분석 사례분석 : beer 데이터

7) 군집별 평균계산

```
# 군집별 평균계산
member1 = pd.DataFrame(member, columns=['cluster'], index=beer.index)
data_combined = beer.join(member1)
data_combined.groupby('cluster').mean()
Out[32]:
         calories     sodium  alcohol      cost
cluster
0          149.00  20.444444    4.800  0.415556
1           70.00  10.500000    2.600  0.420000
2          155.25  10.750000    4.975  0.762500
3          109.20  10.200000    4.100  0.460000
```

3. 대용량 자료의 군집분석

- 계층적 군집분석은 관찰치의 수가 적은 경우에 적당
- 관찰치의 수가 많은 경우에는 관찰치들 사이의 유사성/거리 행렬을 구하는 것이 매우 번거롭고 방대함
- 예. 관찰치수가 10개인 경우 유사성행렬은 9*10/2 = 45
  (혹은 1+2+3+... +9=45) 개의 원소를 가지나, 관찰치수가 1000개인 경우는 999*1000/2 = 499,500 개의 원소를 가짐
- 대용량의 데이터에 적합한 비계층적 군집분석 방법인 K-평균 군집분석 사용

4. 비계층적 군집분석

1) K-평균 군집분석의 절차

1. 군집의 수 K를 정한다.
2. 임의의 K개 관찰치를 K개 각 군집에 임의로 지정한다.
   이를 K개 각 군집의 중심으로 이용한다.
3. 모든 관찰치를 군집중심으로 부터 유클리디안 거리가 최소인 군집에
   귀속시킨다.
4. 각 군집에 속한 관찰치들을 이용하여 군집중심을 새로 계산한다.
5. 변화(군집간 관찰치이동)가 없을 때까지 단계3 과 단계4 를 반복한다.

4. 비계층적 군집분석

2) K-평균 군집분석의 예

- 7개의 관찰치와 2개의 변수가 있는 데이터를 가정

관찰치	변수1	변수2
1	-1	0
2	0	1
3	0	-1
4	2	0
5	3	1
6	3	-1
7	3	0

1. 먼저 군집의 수는 K=2로 정하기로 한다.
2. 관찰치 4와 7을 임의의 초기값으로 설정한다. 이를 각 군집의 중심점으로 이용한다.

4. 비계층적 군집분석

2) K-평균 군집분석의 예

3. 모든 관찰치에 대해서 각 군집 중심점까지의 유클리디안 거리를 측정한다.

관찰치	군집1까지의 거리	군집2까지의 거리	군집할당 결과
1	$\sqrt{(2-(-1))^2+(0-0)^2}=3$	$\sqrt{(3-(-1))^2+(0-0)^2}=4$	1
2	$\sqrt{(2-0)^2+(0-1)^2}=2.24$	$\sqrt{(3-0)^2+(0-1)^2}=3.16$	1
3	$\sqrt{(2-0)^2+(0-(-1))^2}=2.24$	$\sqrt{(3-0)^2+(0-(-1))^2}=3.16$	1
4	$\sqrt{(2-2)^2+(0-0)^2}=0$	$\sqrt{(3-2))^2+(0-0)^2}=1$	1
5	$\sqrt{(2-3)^2+(0-1)^2}=1.41$	$\sqrt{(3-3)^2+(0-1)^2}=1$	2
6	$\sqrt{(2-3)^2+(0-(-1))^2}=1.41$	$\sqrt{(3-3)^2+(0-(-1))^2}=1$	2
7	$\sqrt{(2-3)^2+(0-0)^2}=1$	$\sqrt{(3-3)^2+(0-0)^2}=0$	2

초기 중심값에 의한 군집분석 결과 관찰치 1,2,3,4는 군집1로 할당되고, 관찰치 5,6,7은 군집2로 할당되었다.

4. 비계층적 군집분석

2) K-평균 군집분석의 예

4. 군집에 속한 관찰치들을 이용하여 새로운 군집의 중심점을 계산한다.

군집	변수1	변수2
1	$\frac{-1+0+0+2}{4}=0.25$	$\frac{0+1-1+0}{4}=0$
2	$\frac{3+3+3}{3}=3$	$\frac{1-1+0}{3}=0$

4. 비계층적 군집분석

2) K-평균 군집분석의 예

5. 모든 관찰치에 대해서 새로운 군집 중심점까지의 유클리디안 거리를 다시 측정.

관찰치	군집1까지의 거리	군집2까지의 거리	군집할당
1	$\sqrt{(0.25-(-1))^2+(0-0)^2}=1.25$	$\sqrt{(3-(-1))^2+(0-0)^2}=4$	1
2	$\sqrt{(0.25-0)^2+(0-1)^2}=1.03$	$\sqrt{(3-0)^2+(0-1)^2}=3.16$	1
3	$\sqrt{(0.25-0)^2+(0-(-1))^2}=1.03$	$\sqrt{(3-0)^2+(0-(-1))^2}=3.16$	1
4	$\sqrt{(0.25-2)^2+(0-0)^2}=1.75$	$\sqrt{(3-2))^2+(0-0)^2}=1$	*2*
5	$\sqrt{(0.25-3)^2+(0-1)^2}=2.93$	$\sqrt{(3-3)^2+(0-1)^2}=1$	2
6	$\sqrt{(0.25-3)^2+(0-(-1))^2}=2.93$	$\sqrt{(3-3)^2+(0-(-1))^2}=1$	2
7	$\sqrt{(0.25-3)^2+(0-0)^2}=2.75$	$\sqrt{(3-3)^2+(0-0)^2}=0$	2

수정된 중심값에 의한 군집분석 결과 관찰치 1,2,3은 군집1로 할당되고, 관찰치 4,5,6,7은 군집2로 할당되었다. 관찰치 4에 대한 군집 할당 결과가 바뀐 것을 알 수 있다.

4. 비계층적 군집분석

2) K-평균 군집분석의 예

6. 군집에 속한 관찰치들을 이용하여 새로운 군집의 중심점을 계산한다.

군집	변수1	변수2
1	$\frac{-1+0+0}{3}=-0.33$	$\frac{0+1-1}{3}=0$
2	$\frac{3+3+3+2}{4}=2.75$	$\frac{1-1+0+0}{4}=0$

4. 비계층적 군집분석

2) K-평균 군집분석의 예

7. 모든 관찰치에 대해서 새로운 군집 중심점까지의 유클리디안 거리를 다시 측정.

관찰치	군집1까지의 거리	군집2까지의 거리	군집할당 결과
1	$\sqrt{(-0.33-(-1))^2+(0-0)^2}=0.67$	$\sqrt{(2.75-(-1))^2+(0-0)^2}=3.75$	1
2	$\sqrt{(-0.33-0)^2+(0-1)^2}=1.05$	$\sqrt{(2.75-0)^2+(0-1)^2}=2.93$	1
3	$\sqrt{(-0.33-0)^2+(0-(-1))^2}=1.05$	$\sqrt{(2.75-0)^2+(0-(-1))^2}=2.93$	1
4	$\sqrt{(-0.33-2)^2+(0-0)^2}=2.33$	$\sqrt{(2.75-2))^2+(0-0)^2}=0.75$	2
5	$\sqrt{(-0.33-3)^2+(0-1)^2}=4.19$	$\sqrt{(2.75-3)^2+(0-1)^2}=1.03$	2
6	$\sqrt{(-0.33-3)^2+(0-(-1))^2}=4.19$	$\sqrt{(2.75-3)^2+(0-(-1))^2}=1.03$	2
7	$\sqrt{(-0.33-3)^2+(0-0)^2}=3.33$	$\sqrt{(2.75-3)^2+(0-0)^2}=0.25$	2

4. 비계층적 군집분석

2) K-평균 군집분석의 예

8. 더 이상 군집간 관찰치 이동이 없으므로 종료한다.
   최종적인 K-평균 군집분석의 결과로는 군집 1에 관찰치 (1,2,3) 이 속하며,
   군집 2에 관찰치 (4,5,6,7) 이 속한다고 결론 내린다.

4. 비계층적 군집분석

3) 기타 고려사항

(1) 초기값의 설정

① 처음에 지정된 군집중심(초기값)에 의하여 최종 군집결과에 차이가 발생할 수도 있음

② 데이터의 수가 소량일수록 초기값의 설정은 더욱 중요한 문제

③ 초기값을 설정하는 대표적 방법은 데이터 내 임의의 K개 관찰치를 각 군집의 초기값으로 설정하는 것

④ 계층적 군집분석을 먼저 수행하여 구하여진 K개 군집의 중심점을 K-평균 군집분석의 초기값으로 사용하는 방식도 가능

⑤ 데이터가 대용량이기 때문에 계층적 군집분석을 사용하여 초기값을 구하기 어려운 경우에는 원데이터로 부터 적은 표본을 추출하여 계층적 군집분석을 수행한 후 초기값을 구할 수도 있음

4. 비계층적 군집분석

3) 기타 고려사항

(2) 군집의 수 결정

① K-평균 군집분석에서 군집수 K를 얼마로 하느냐에 의하여도 군집결과가 달라질 수 있음

② 군집의 수를 증가시켜가면서 K-평균 군집분석을 반복하여 수행한 후 이러한 결과중 가장 좋은 결과를 보이는 군집의 수를 결정하는 방법

③ 군집분석의 수행 이전에 주성분 분석을 먼저 수행하고 상위 2개의 주성분을 이용하여 군집의 개수를 미리 그림으로 확인해 보는 방법도 있음

④ 계층적 군집분석을 먼저 수행하여 군집의 수를 덴드로그램을 통하여 미리 정한 후 비계층적 군집분석을 수행하는 방법

4. 비계층적 군집분석

3) 기타 고려사항

(3) 실용적 고려사항

① 빠른 연산으로 인하여 대규모의 데이터에도 손쉽게 군집분석 결과를 구할 수 있다는 장점. 실무적으로 아주 활용도가 높음

② 검증: 분석 데이터를 훈련데이터와 검증데이터로 분리하여 각 데이터에 K-평균 군집분석을 수행하고 생성된 K개 군집의 중심점을 비교해 보는 방법

③ 검증: 군집분석의 군집할당 결과를 이용하여 판별분석을 수행해 보는 방법이 있음. 판별분석의 결과가 우수하다면 군집분석의 결과를 신뢰할 수 있음

5. k-평균 군집분석 사례

1) 데이터 설명

- ❖ 20개 맥주상표를 대상으로 가격, 칼로리, 염분, 알코올농도 등을 측정한 자료
- ❖ 각 변수의 관측단위가 다르기 때문에 각 변수들을 표준화하여 이용하는 것이 바람직함

변수명	의미
BEER	맥주이름
X1: CALORIES	12온스당 칼로리량
X2: SODIUM	12온스당 염분량(mg)
X3: ALCOHOL	알코올농도(%)
X4: COST	12온스당 가격($)

5. k-평균 군집분석 사례

1) 데이터 설명

	calories	sodium	alcohol	cost
BUDWEISER	144	15	4.7	0.43
SCHLITZ	151	19	4.9	0.43
LOWENBRAU	157	15	4.9	0.48
KRONENBOURG	170	7	5.2	0.73
HEINEKEN	152	11	5.0	0.77
OLD MILWAUKEE	145	23	4.6	0.28
AUGSBERGER	175	24	5.5	0.40
STROHS BOHEMIAN STYLE	149	27	4.7	0.42
MILLER LITE	99	10	4.3	0.43
BUDWEISER LIGHT	113	8	3.7	0.44
COORS	140	18	4.6	0.44
COORS LIGHT	102	15	4.1	0.46
MICHELOB LIGHT	135	11	4.2	0.50
BECKS	150	19	4.7	0.76
KIRIN	149	6	5.0	0.79
PABST EXTRA LIGHT	68	15	2.3	0.38
HAMMS	136	19	4.4	0.43
HEILEMANS OLD STYLE	144	24	4.9	0.43
OLYMPIA GOLD LIGHT	72	6	2.9	0.46
SCHLITZ LIGHT	97	7	4.2	0.47

5. k-평균 군집분석 사례

2) 데이터 읽기

```
> beer.data = read.table("c:/data/mva/beerbrand.csv", header=T, sep=",")
> head(beer.data)
              calories sodium alcohol cost
BUDWEISER          144     15     4.7 0.43
SCHLITZ            151     19     4.9 0.43
LOWENBRAU          157     15     4.9 0.48
KRONENBOURG        170      7     5.2 0.73
HEINEKEN           152     11     5.0 0.77
OLD MILWAUKEE      145     23     4.6 0.28
> summary(beer.data)
    calories         sodium         alcohol          cost       
 Min.   : 68.0   Min.   : 6.00   Min.   :2.30   Min.   :0.2800  
 1st Qu.:110.2   1st Qu.: 9.50   1st Qu.:4.20   1st Qu.:0.4300  
 Median :144.0   Median :15.00   Median :4.65   Median :0.4400  
 Mean   :132.4   Mean   :14.95   Mean   :4.44   Mean   :0.4965  
 3rd Qu.:150.2   3rd Qu.:19.00   3rd Qu.:4.90   3rd Qu.:0.4850  
 Max.   :175.0   Max.   :27.00   Max.   :5.50   Max.   :0.7900  
> 
```

head(beer.data) 명령은 처음 6개의 케이스를 출력

5. k-평균 군집분석 사례

3) 자료 표준화

```
> zbeer = scale(beer)
> round(apply(zbeer, 2, mean), 3)
calories   sodium  alcohol     cost
       0        0        0        0
> round(apply(zbeer, 2, sd), 3)
calories   sodium  alcohol     cost
       1        1        1        1
>
```

```
> # 0-1 변환
> library(caret)
> z01_beer = preProcess(beer,
  method='range')
> z01_model = preProcess(beer,
  method='range')
> z01_beer = predict(z01_model, beer)
> summary(z01_beer)
```

```
> # 0-1 변환 (2)
> maxX = apply(beer, 2, max)
> minX = apply(beer, 2, min)
> z01X = scale(beer, center=minX,
  scale=maxX-minX)
> summary(z01X)
```

5. k-평균 군집분석 사례

4) K-평균 군집분석 실행하기

```
> kmc = kmeans(zbeer, centers=2)
> kmc
K-means clustering with 2 clusters of sizes 14, 6
Cluster means:
    calories     sodium    alcohol       cost
1  0.5745921  0.3114899  0.4832236  0.1684391
2 -1.3407148 -0.7268097 -1.1275218 -0.3930246

Clustering vector:
           BUDWEISER              SCHLITZ           LOWENBRAU
                   1                    1                   1
         KRONENBOURG             HEINEKEN        OLD MILWAUKEE
                   1                    1                   1
  AUGSBERGER STROHS BOHEMIAN STYLE          MILLER LITE
                   1                    1                   2
     BUDWEISER LIGHT               COORS        COORS LIGHT
                   2                    1                   2
 ...
Within cluster sum of squares by cluster:
[1] 34.328491  9.515432
 (between_SS / total_SS =  42.3 %)
```

군집 수를 2개로

군집별 평균값

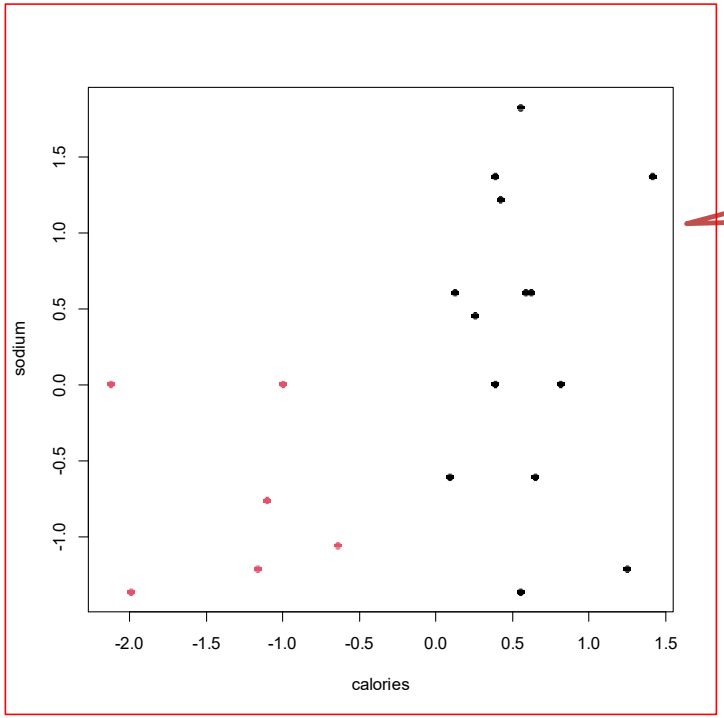
소속 군집 알기

5. k-평균 군집분석 사례

5) K-평균 소속 군집 산점도

산점도의 점을
색상으로 채움

```
> plot(zbeer, col=kmc$cluster, pch=16)
```



소속 군집 산점도

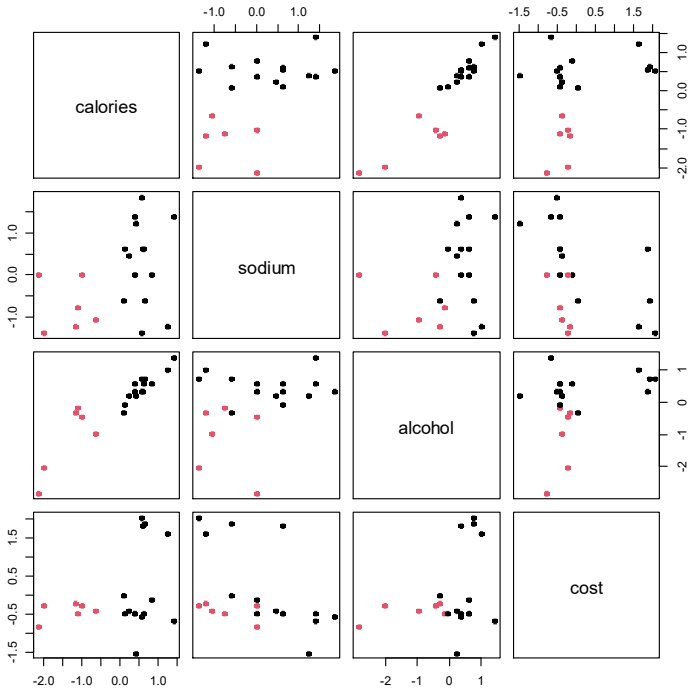
처음 두 개 변수
사용됨

5. k-평균 군집분석 사례

6) K-평균 소속 군집 산점도

```
> pairs(zbeer, col=kmc$cluster, pch=16, cex.labels=1.5)
```

모든 변수 사용한 산점도



6. 파이썬 k-평균 군집분석 : beer 데이터

1) 데이터 읽기

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# 데이터 읽기
beer = pd.read_csv("c:/data/mva/beerbrand.csv", index_col='name')
beer.head()
Out[6]:
            calories  sodium  alcohol  cost
name
BUDWEISER        144      15      4.7  0.43
SCHLITZ          151      19      4.9  0.43
LOWENBRAU        157      15      4.9  0.48
KRONENBOURG      170       7      5.2  0.73
HEINEKEN         152      11      5.0  0.77
# 기술통계량 구하기
beer.describe()
```

6. 파이썬 k-평균 군집분석 : beer 데이터

2) 데이터 표준화

```
# 표준화 패키기 불러오기
from sklearn.preprocessing import StandardScaler
# 표준화 시행
zbeer = StandardScaler().fit_transform(beer)
type(zbeer)
Out[12]: numpy.ndarray
zbeer_frame = pd.DataFrame(zbeer)
zbeer_frame.columns = beer.columns
zbeer_frame.describe()
Out[17]:
```

	calories	sodium	alcohol	cost
count	2.000000e+01	2.000000e+01	2.000000e+01	2.000000e+01
mean	-1.110223e-16	7.771561e-17	-1.634803e-15	-3.747003e-16
std	1.025978e+00	1.025978e+00	1.025978e+00	1.025978e+00
min	-2.183691e+00	-1.395248e+00	-2.889782e+00	-1.545138e+00
25%	-7.510676e-01	-8.496201e-01	-3.240877e-01	-4.746037e-01
50%	3.933356e-01	7.794680e-03	2.835767e-01	-4.032347e-01
75%	6.052621e-01	6.313691e-01	6.211680e-01	-8.207432e-02
max	1.444491e+00	1.878518e+00	1.431387e+00	2.094679e+00

6. 파이썬 k-평균 군집분석 : beer 데이터

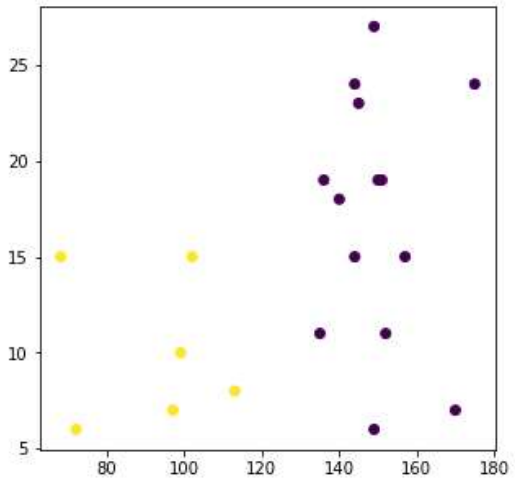
3) K-평균 군집분석 실행하기

```
# K-means 군집분석
from sklearn.cluster import KMeans

# k-평균 군집분석: 군집수 = 2
kmc = KMeans(n_clusters=2)
kmc.fit(zbeer)
Out[4]: KMeans(n_clusters=2)

# 군집 중심 알기
kmc.cluster_centers_
Out[5]:
array([[ 0.58951901,  0.31958187,  0.49577698,  0.17281486],
       [-1.37554436, -0.74569103, -1.15681296, -0.40323468]])

# 소속군집 알기
kmc.labels_
Out[6]: array([0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1])
```

6. 파이썬 k-평균 군집분석 : beer 데이터

4) K-평균 소속 군집 산점도

```
# 소속 군집 산점도
plt.figure(figsize=(5,5))
plt.scatter(x=beer['calories'], y=beer['sodium'], c=kmc.labels_)
plt.show()
```



다음시간에는

8강 다차원척도법

수고했습니다