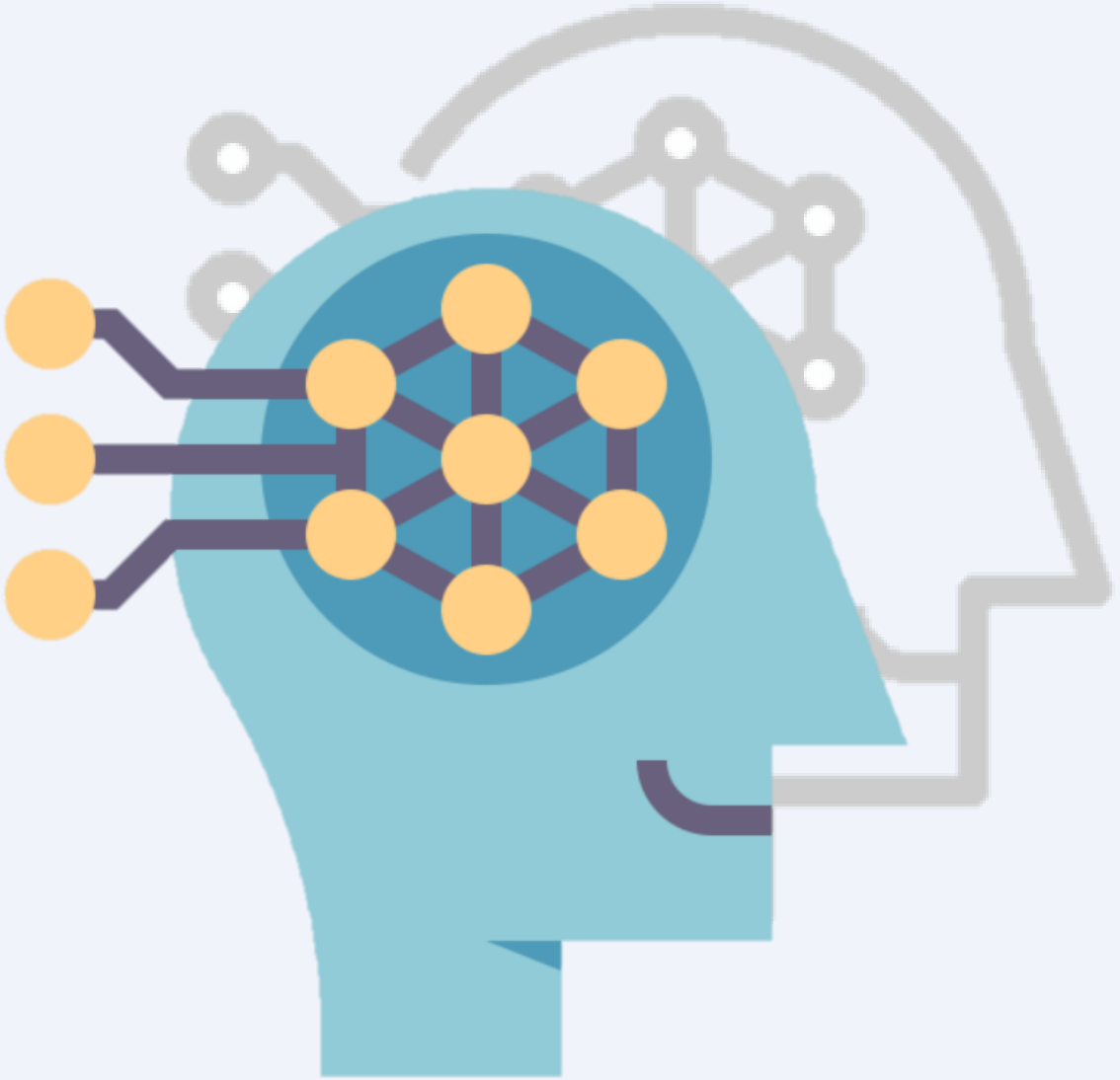


7주차
Clustering & K-NN algorithm

5기 김하은

목차

01. 군집분석

01. 군집분석

군집 분석이란?

관측된 여러 개의 변수로부터 **유사성에 기초하여**
n개의 군집으로 집단화하여 집단의 특성을 분석하는 다변량 분석

변수들이 속한 모집단 또는 범주에 대한 사전정보가 없는 경우
관측값들 사이의 **거리 (유사성)을 이용하여** 개체들을 **n개의 군집으로** 나누는 분석

01. 군집분석

군집 (clustering)

: 군집분석은 **비지도 학습** 방법으로
군집의 수, 속성 등의 정보가 사전에 **알려지지 않을 때** 사용하는 분석방법

따라서 데이터 간의 유사도를 정의하고 그 유사도에 가까운 것부터 합쳐가는 방법

분류 (classification)

군집 (clustering) ≠ 분류 (classification)

: 분류는 **지도 학습** 방법으로
각 개체 별 그룹의 정보가 사전에 **알려져 있을 때** 사용하는 분석방법

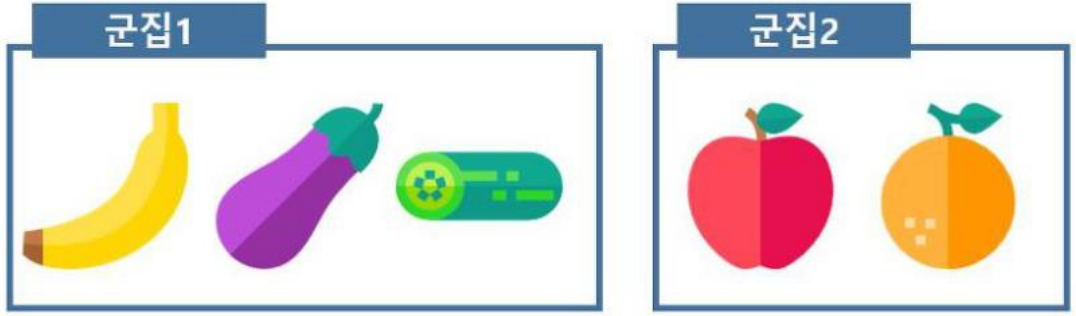
따라서 데이터의 범주를 파악하고 새롭게 관측된 데이터의 범주를 스스로 판별하는 과정
즉, 군집 분석과 달리 각 개체가 어떤 그룹에 들어갈까 예측하는 기법

01. 군집분석

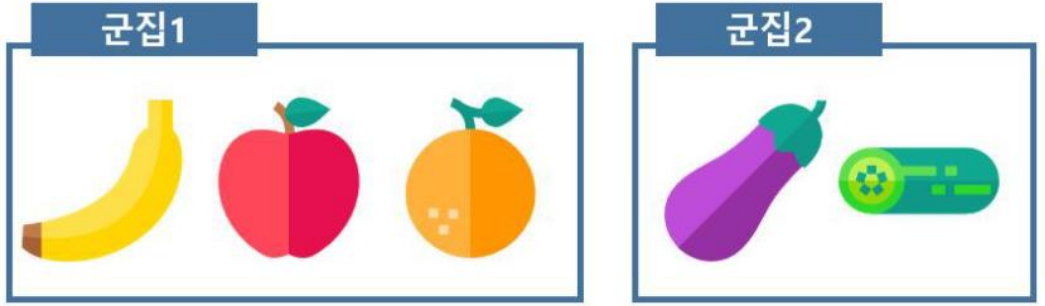
군집분석의 point

같은 군집 내에서는 동질적이고, 다른 군집 간에는 서로 이질적이어야 한다.

비슷한 모양끼리 군집화



과일끼리, 채소끼리 군집화



01. 군집분석

군집 분석의 종류

① 계층적 군집분석

가까운 개체끼리 묶거나 멀리 떨어진 개체를 차례로 분리해 가는 군집분석
이때 한번 병합된 개체는 다시 분리되지 **않음**
군집의 개수 unknown

② 비계층적 군집분석

다변량 자료의 **산포**를 나타내는 여러 측도를 이용하여 판정 기준을 최적화하는 방식으로 나누는 군집분석
한번 분리된 개체도 반복적으로 시행하는 과정에서 **재분류 가능**
군집의 개수가 사전에 정해짐
그러나, 일반적으로 군집의 개수를 판단할 수 있는 다양한 기준을 적용하여 군집의 개수를 최종 결정

02. 거리 측정 방법

02. 거리 측정 방법

군집 간 거리 계산 = 군집 간 유사성 확인

거리가 멀다 = 유사성 낮다
거리가 가깝다 = 유사성 높다

거리는 다양한 방법으로 정의 될 수 있지만, 일반적으로 다음의 성질을 충족시켜야 함

d_{ij} **관측치 i와 j 사이의 거리**

양의 성질 : $d_{ij} \geq 0$

자기 근접성 : $d_{ii} = 0$ **-> 자기 자신과의 거리는 0**

대칭성 : $d_{ij} = d_{ji}$

삼각 부등식 : $d_{ij} \leq d_{ik} + d_{kj}$

02. 거리 측정 방법 – 연속형 변수

연속형 변수의 거리

① 수학적 거리 : 기하적 거리, 통계적 개념 내포 X

a. 유클리드 거리 – 기하학적 최단 거리

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

b. 맨해튼 거리 (시티 블록 거리) – 두 위치 차이의 절대값

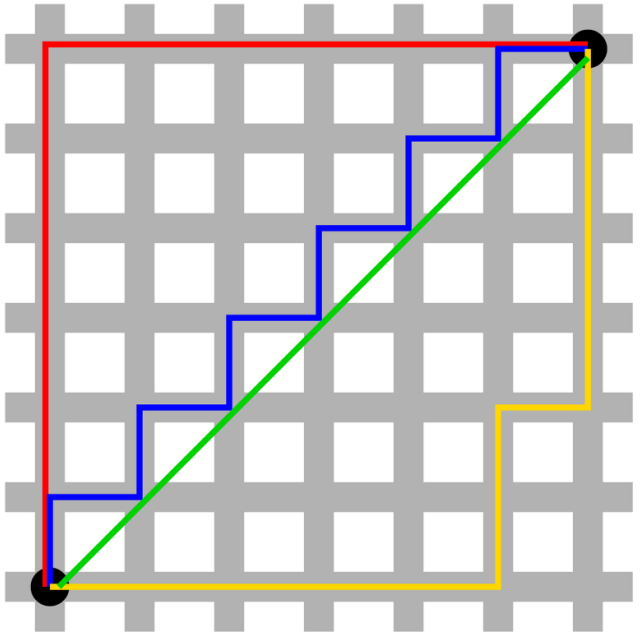
$$d_{ij} = \sum_{m=1}^{p} |x_{im} - x_{jm}|$$

c. 최대 좌표 거리

$$d_{ij} = \max |x_{im} - x_{jm}|$$

d. 민코프스키 거리 – 유클리드+맨해튼 일반화

$$d_{ij} = (\sum_{m=1}^{n} |x_{im} - x_{jm}|^p)^{1/p}$$



02. 거리 측정 방법 – 연속형 변수

② 통계적 거리 : 통계적 개념이 내포 O -> 척도의 차이, 분산의 차이로 인한 왜곡 방지 가능

a. **상관계수**에 기초한 유사도

$$d_{ij} = 1 - r_{ij}^2 \qquad r_{ij} = \frac{\sum_{m=1}^{p}(x_{im} - \bar{x}_m)(x_{jm} - \bar{x}_m)}{\sqrt{(x_{im} - \bar{x}_m)^2 + (x_{jm} - \bar{x}_m)^2}}$$

b. **표준화** 거리 – 표준화한 뒤 계산한 유클리드 거리 (s는 j의 표준편차, D는 표본분산의 대각행렬)

$$d_{ij} = \sqrt{\frac{\sum_{m=1}^{n}(x_{im} - x_{jm})^2}{s_{jj}}} = \sqrt{(x_i - x_j)^T D^{-1}(x_i - x_j)}$$

c. **마할라노비스** 거리 – 표준화 거리에 상관성까지 동시에 고려한 거리

$$d_{ij} \equiv \sqrt{(x_i - x_j)^T \sum^{-1} (x_i - x_j)}$$

$\sum^{-1}$ =표본 공분산 행렬

02. 거리 측정 방법 – 범주형 변수

범주형 변수의 거리

ID	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
i	0	1	0	0	0	1	1	1	1	1
j	1	1	0	1	1	0	1	1	1	1

i=j	1=1	5개
	0=0	1개
i≠j	1≠0	1개
	0≠1	3개

i와 j의 거리
=Hamming Distance

$$= \frac{4}{10}$$

유사도=1-(거리)=0.6

<Hamming Distance 의 문제점>
1=1 과 0=0 을 모두 동일하게 반영 (만약 1=1이 0=0보다 유사성이 크다면 바람직하지 않음)

따라서 가중값이 부여된 새로운 유사도 계수가 필요해짐.

02. 거리 측정 방법 – 범주형 변수

수정된 범주형 변수의 거리

A: 1=1	C: 0≠1
B: 1≠0	D: 0=0

Jaccard = $\frac{A}{A+B+C}$

Simple Matching= $\frac{A+D}{A+B+C+D}$

Soerensen-Dice = $\frac{2A}{2A+B+C}$

Rogers and Tanimoto= $\frac{A+D}{A+D+2(B+C)}$

Anderberg= $\frac{A}{A+2(B+C)}$

Phi-Coefficient= $\frac{AD-BC}{\sqrt{(A+B)(C+D)(A+C)(B+D)}}$

Ochiai= $\frac{A}{\sqrt{(A+B)(A+C)}}$

Russel and Rao= $\frac{A}{A+B+C+D}$

02. 거리 측정 방법 – 혼합형 데이터

★ 주의 ★

연속형 변수와 범주형 변수 (이진형 변수) 가 함께 존재하는 혼합형 데이터의 경우,
군집분석에서 개체 간 거리 측정이 모호함.

다시 말해, 연속형 변수와 명목형 변수 사이의 거리를 측정한다는 것이 모호함.
거리를 측정하였다 한들 해석이 쉽지 않음.

Gower의 유사도 : 각 변수를 [0,1] 스케일로 조정 후 각 변수의 거리를 가중평균 한 것

$$s_{ij} = \frac{\sum_{m=1}^{p} w_{ijm} s_{ijm}}{\sum_{m=1}^{p} w_{ijm}}$$

이 외에도 가변수 변환 방식, Eskin 등의 방법이 존재함.

02. 두 군집 사이 거리 측정 방법

군집 사이의 거리

① 최단거리 : 각 군집에 속한 A와 B 사이의 거리 중 가장 가까운 거리로 정의

② 최장거리 : 각 군집에 속한 A와 B 사이의 거리 중 가장 먼 거리로 정의

③ 평균거리 : 군집 A에 속한 레코드와 군집 B에 속한 레코드 사이의 가능한 모든 거리의 평균으로 정의

④ 중심거리 : 각 군집에서 구한 중심점 사이의 거리로 정의

03. 계층적 군집분석 알고리즘

03. 계층적 군집분석 알고리즘

- 계층적 군집분석
  - 합병
    - 단일 (최단) 연결법 $d(U, V) = \min[d(x, y) | x \in U, y \in V]$
    - 완전 (최장) 연결법 $d(U, V) = \max[d(x, y) | x \in U, y \in V]$
    - 평균 연결법 $d(U, V) = 1/(N_1 N_2) \sum_i \sum_j d_{ij}$
    - 중심 연결법 $d(U, V) = P(\overline{X}_1, \overline{X}_2)$
    - WARD 연결법 – 정보의 손실을 고려한 방식 / (군집 내 오차제곱합) 이용
    - 중앙값 연결법
  - 분할 – 다이아나 방법
    - 거의 진행 X

03. 계층적 군집분석 알고리즘

계층적 군집분석의 단계

1. Distance Measure 결정 : 거리 측도를 어떻게 할 것인가
2. Clustering Algorithm 결정 : 군집이 만들어지는 단계마다 거리를 어떻게 측정할 것인지에 관한 알고리즘 결정
3. 군집 개수 결정
4. 분석의 타당성 검토

계층적 군집 분석의 한계점

1. 큰 데이터 셋에서의 계산속도 저하
2. 군집의 수정 불가능
3. 안정성이 낮음
4. 거리계산 방식에 영향이 큼 (결과가 달라짐)
5. 이상치에 민감

04. 비계층적 군집분석 알고리즘

04. 비계층적 군집분석 알고리즘

K-평균 군집화 알고리즘 (K-Means clustering algorithm)

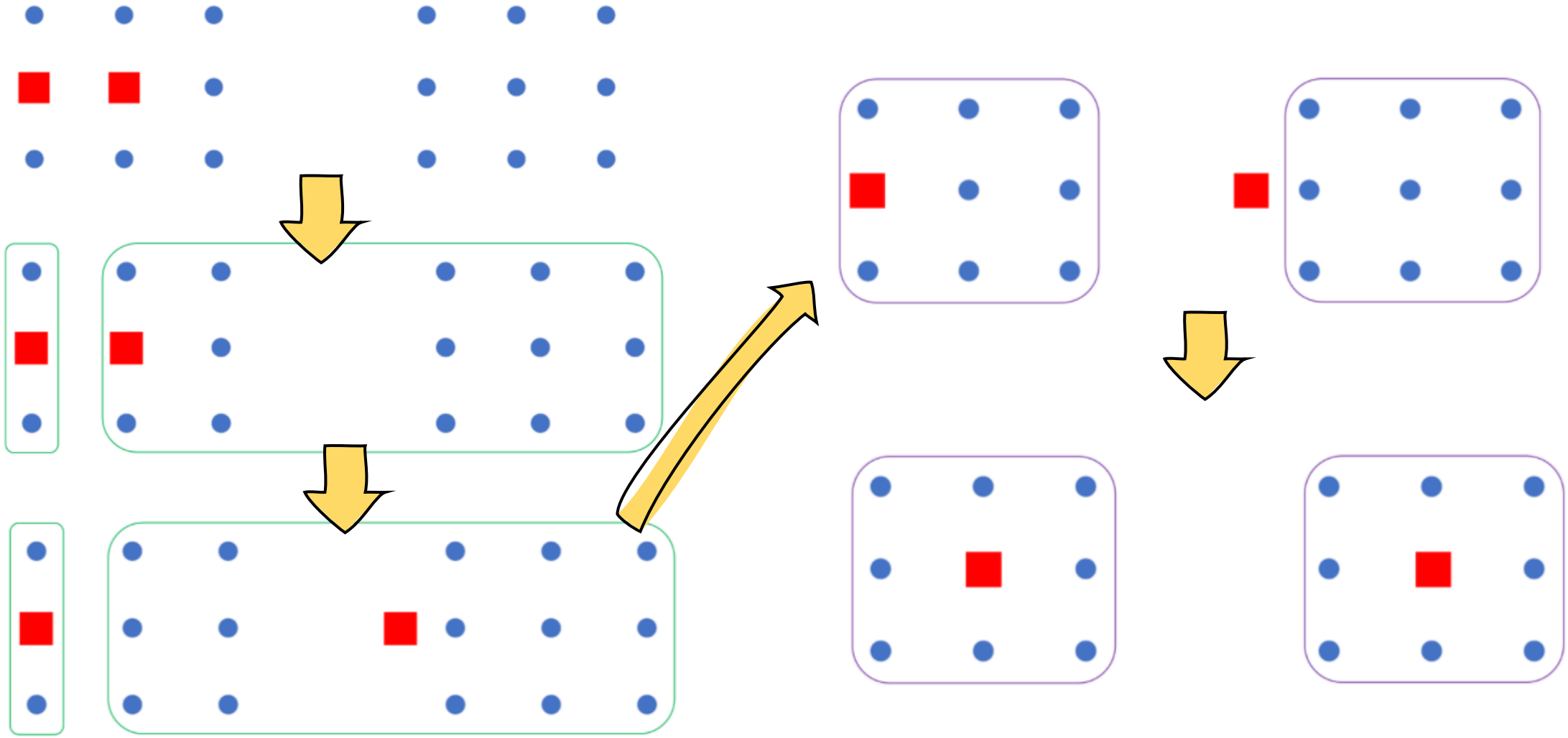
: **연속형 자료**에 적용하는 군집화 기법

K-평균 군집화 알고리즘의 단계

1. 초기 군집 수를 k로 잡는다.
2. 매 단계에서 각 레코드는 중심이 가장 가까운 군집으로 재할당된다.
3. 관측치가 빠지거나 추가되는 군집의 중심을 다시 계산하고, 단계 2를 반복한다.
4. 더 이상의 레코드 이동이 없으면 중지한다. 혹은 사용자가 반복 수 지정.

이때 k-평균 군집화 알고리즘은 **EM 알고리즘**을 기반으로 진행된다.

04. 비계층적 군집분석 알고리즘



04. 비계층적 군집분석 알고리즘

K-Modes clustering algorithm

: **범주형 자료**에 적용하는 군집화 기법

범주형 자료의 거리의 값을 구하는 데 어려움이 있기 때문에
비유사도(dissimilarity)의 개념을 활용

K-Prototypes clustering algorithm

: **혼합형 자료**에 적용하는 군집화 기법

연속형 자료는 유클리드 거리를 구하고, 범주형 자료는 비유사도를 구한 다음,
비유사도에 가중치를 부여하여 둘을 합한 것을 거리로 정의하는 알고리즘

05.K-NN Algorithm

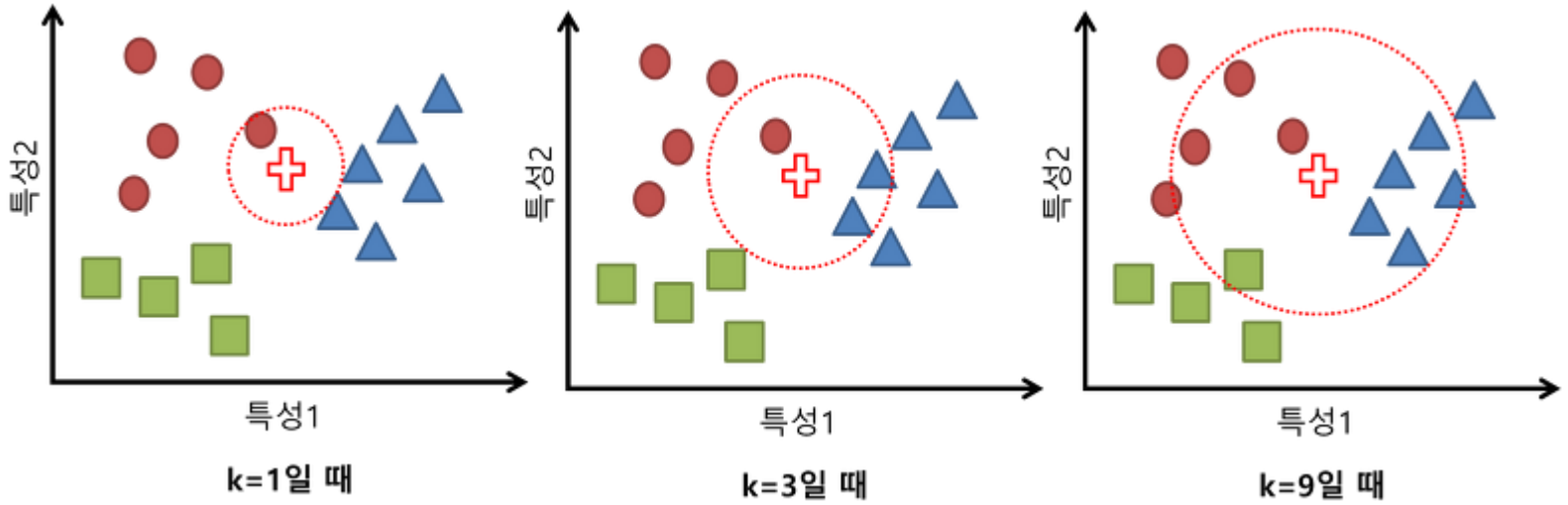
05. K-NN Algorithm

K 최근접 이웃 알고리즘 (K-Nearest Neighbors Algorithm)

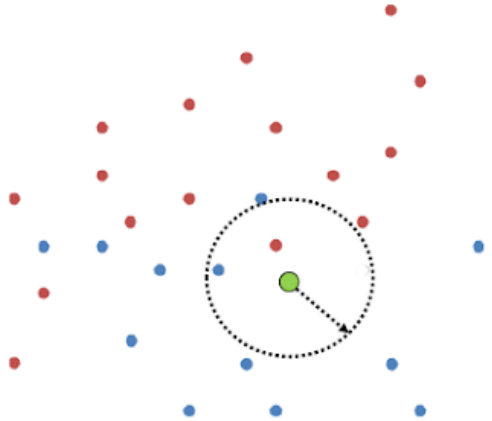
: 별도의 모델 생성 없이 인접 데이터를 분류 / 예측하는데 사용하는 지도학습 알고리즘

기존 관측치인 Y값 (class) 가 존재한다는 점에서 비지도 학습인 군집분석과 차이를 보임

K-NN 알고리즘은 데이터로부터 가까운 K개의 다른 데이터 레이블을 참조하여 분류
이때 거리 측정은 '유클리드' 거리 계산법을 이용



05. K-NN Algorithm



K의 개수는 홀수로 하는 것이 좋음
=>동점 시 분류할 수 없는 상황이 발생하기 때문

K-NN의 장/단점

장점 : 단순하고 효율적임
훈련 단계가 빠름
수치기반 데이터 분류 작업에서 성능이 우수함
단점 : 모델을 생성하지 않아 특징과 클래스 간 관계를 이해하는 데 제한적
적절한 K의 선택이 필요
데이터가 많아지면 분류단계가 느림

06. 실습

06. 실습

미국의 공공 전력 회사에 관한 데이터

목표 : 22개의 전력회사를 유사한 전력회사끼리 묶어 군집화하기

	Company	Fixed_charge	RoR	Cost	Load_factor	Demand_growth	Sales	Nuclear	Fuel_Cost
1	Arizona	1.06	9.2	151	54.4	1.6	9077	0.0	0.628
2	Boston	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
3	Central	1.43	15.4	113	53.0	3.4	9212	0.0	1.058
4	Commonwealth	1.02	11.2	168	56.0	0.3	6423	34.3	0.700
5	NY	1.49	8.8	192	51.2	1.0	3300	15.6	2.044
6	Florida	1.32	13.5	111	60.0	-2.2	11127	22.5	1.241
7	Hawaiian	1.22	12.2	175	67.6	2.2	7642	0.0	1.652
8	Idaho	1.10	9.2	245	57.0	3.3	13082	0.0	0.309
9	Kentucky	1.34	13.0	168	60.4	7.2	8406	0.0	0.862
10	Madison	1.12	12.4	197	53.0	2.7	6455	39.2	0.623
11	Nevada	0.75	7.5	173	51.5	6.5	17441	0.0	0.768
12	New England	1.13	10.9	178	62.0	3.7	6154	0.0	1.897
13	Northern	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
14	Oklahoma	1.09	12.0	96	49.8	1.4	9673	0.0	0.588
15	Pacific	0.96	7.6	164	62.2	-0.1	6468	0.9	1.400
16	Puget	1.16	9.9	252	56.0	9.2	15991	0.0	0.620
17	San Diego	0.76	6.4	136	61.9	9.0	5714	8.3	1.920
18	Southern	1.05	12.6	150	56.7	2.7	10140	0.0	1.108
19	Texas	1.16	11.7	104	54.0	-2.1	13507	0.0	0.636
20	Wisconsin	1.20	11.8	148	59.9	3.5	7287	41.1	0.702
21	United	1.04	8.6	204	61.0	3.5	6650	0.0	2.116
22	Virginia	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

company	회사명
Fixed_charge	고정 비용 부담률 (수익/부채)
RoR	투자 수익률
Cost	KW당 생산비용
Load_factor	연간 부하량
Demand_growth	1974~75년까지 최대 전력 수요의 증가율
Sales	전력 판매 매출액
Nuclear	원자력 발전 비율
Fuel Cost	총 연료비

06. 실습 - 계층적 군집분석

```
> head(utilities)
       Company Fixed_charge  RoR Cost Load_factor Demand_growth Sales Nuclear Fuel_Cost
1      Arizona         1.06  9.2  151        54.4            1.6  9077     0.0     0.628
2       Boston         0.89 10.3  202        57.9            2.2  5088    25.3     1.555
3      Central         1.43 15.4  113        53.0            3.4  9212     0.0     1.058
4 Commonwealth         1.02 11.2  168        56.0            0.3  6423    34.3     0.700
5           NY         1.49  8.8  192        51.2            1.0  3300    15.6     2.044
6      Florida         1.32 13.5  111        60.0           -2.2 11127    22.5     1.241
```

```
> row.names(utilities)<-utilities[,1]
> utilities<-utilities[,-1]
> head(utilities)
             Fixed_charge  RoR Cost Load_factor Demand_growth Sales Nuclear Fuel_Cost
Arizona              1.06  9.2  151        54.4            1.6  9077     0.0     0.628
Boston               0.89 10.3  202        57.9            2.2  5088    25.3     1.555
Central              1.43 15.4  113        53.0            3.4  9212     0.0     1.058
Commonwealth         1.02 11.2  168        56.0            0.3  6423    34.3     0.700
NY                   1.49  8.8  192        51.2            1.0  3300    15.6     2.044
Florida              1.32 13.5  111        60.0           -2.2 11127    22.5     1.241
```

회사명을 row 이름으로 변경

06. 실습 - 계층적 군집분석

유클리드 거리 계산

```
> distance<-dist(utilities,method="euclidean")
> distance
```

	Arizona	Boston	Central	Commonwealth	NY	Florida	Hawaiian	Idaho	Kentucky	Madison	Nevada	New England	Northern	Oklahoma	Pacific	Puget	San Diego	Southern	Texas	Wisconsin	United
Boston	3989.40808																				
Central	140.40286	4125.04413																			
Commonwealth	2654.27763	1335.46650	2789.75967																		
NY	5777.16767	1788.06803	5912.55291	3123.15322																	
Florida	2050.52944	6039.68908	1915.15515	4704.36310	7827.42921																
Hawaiian	1435.26502	2554.28716	1571.29540	1219.56001	4342.09380	3485.67156															
Idaho	4006.10419	7994.15599	3872.25763	6659.53457	9782.15818	1959.73108	5440.46178														
Kentucky	671.27635	3318.27656	807.92079	1983.31435	5106.09415	2721.70630	764.08319	1676.63838													
Madison	2622.69900	1367.09063	2758.55966	43.64889	3155.09559	4672.82929	1187.94114	5627.29178	1951.62858												
Nevada	8364.03105	12353.06270	8229.22328	11018.05781	14141.02258	6314.35909	9799.01555	3359.59960	9035.00749	10986.09801											
New England	2923.13610	1066.57943	3058.70743	271.45273	2854.09948	4973.50684	1488.01491	5928.32617	2252.02672	304.27703	11287.00691										
Northern	1899.27982	2091.16049	2035.44152	756.83195	3879.16746	3949.09232	466.55912	5903.39545	1228.43633	724.09618	10262.15729	1026.48299									
Oklahoma	598.55663	4586.30256	461.34167	3250.98459	6373.74325	1454.29260	2032.61425	3412.26397	1269.10210	3219.82511	7768.38479	3519.97756	2496.63889								
Pacific	2609.04536	1380.74996	2744.50285	56.64463	3168.17746	4659.35626	1174.07562	5614.49924	1938.02656	53.30140	10973.01095	314.35403	713.66505	3205.74888							
Puget	6914.74206	10903.14646	6780.43031	9568.43443	12691.15511	4866.11165	8349.36644	2909.01468	7585.46729	9536.24219	1452.16201	9837.28183	8812.30356	6319.93384	9523.41350						
San Diego	3363.06163	629.76075	3498.11301	710.29296	2414.69876	5413.09300	1928.44148	1368.81544	2692.21236	744.25367	11727.06629	442.13276	1466.99195	3959.24075	754.61209	10277.66038					
Southern	1063.00907	5052.33167	928.74925	3717.20296	6840.15029	988.04456	2498.14902	2943.53557	1734.10330	3685.51009	7301.04086	3986.10243	2961.83475	470.16479	3672.03540	5851.89331	4426.04189				
Texas	4430.25159	8419.61054	4295.01469	7084.37284	10207.39263	2380.12497	5865.44719	447.82867	5101.41414	7052.72388	3934.61752	7353.37915	6328.91795	3834.01226	7039.26207	2488.43222	7793.08395	3367.31887			
Wisconsin	1790.48565	2199.72167	1925.77256	864.27315	3987.33596	3840.22794	358.47629	5795.95881	1119.94001	833.47299	10154.11879	1134.14501	119.98126	2386.94275	820.16430	8704.72128	1573.40838	2853.29878	6220.29673		
United	2427.58887	1562.21081	2563.63736	232.47687	3350.07312	4478.02887	992.45325	5432.13220	1756.37897	199.22840	10791.04927	496.68741	531.47633	3024.95235	186.38865	9341.12661	938.52273	3490.42292	6857.73586	640.78677	
Virginia	1016.61769	5005.08126	883.53546	3670.01819	6793.03530	1035.98148	2451.18516	2989.96398	1687.23603	3638.09755	7348.04902	3939.10035	2914.20499	428.06526	3625.11887	5898.57696	4379.21182	59.32529	3414.83146	2806.16571	3443.24097

데이터 표준화

```
> utilities.norm<-sapply(utilities,scale) #전체 데이터 정규화(표준화)를 한번에 할 수 있음
> row.names(utilities.norm)<-row.names(utilities)
> head(utilities.norm)
```

	Fixed_charge	RoR	Cost	Load_factor	Demand_growth	Sales	Nuclear	Fuel_Cost
Arizona	-0.2931579	-0.6846390	-0.417122002	-0.5777152	-0.52622751	0.04590290	-0.7146294	-0.85367545
Boston	-1.2145113	-0.1944537	0.821002037	0.2068363	-0.33381191	-1.07776413	0.7920476	0.81329670
Central	1.7121407	2.0782236	-1.339645796	-0.8915357	0.05101929	0.08393124	-0.7146294	-0.08043055
Commonwealth	-0.5099470	0.2066070	-0.004413989	-0.2190631	-0.94312798	-0.70170610	1.3280197	-0.72420189
NY	2.0373243	-0.8628882	0.578232617	-1.2950193	-0.71864311	-1.58142837	0.2143888	1.69263800
Florida	1.1159709	1.2315399	-1.388199680	0.6775672	-1.74485965	0.62337028	0.6253007	0.24864810

06. 실습 - 계층적 군집분석

표준화한 데이터로 다시 유클리드 거리 계산

```
> d.norm<-dist(utilities.norm,method="euclidean") #euclidean=유클리드
> d.norm
```

	Arizona	Boston	Central	Commonwealth	NY	Florida	Hawaiian	Idaho	Kentucky	Madison	Nevada	New England	Northern	Oklahoma	Pacific	Puget	San Diego	Southern	Texas	Wisconsin	United
Boston	3.096154																				
Central	3.679230	4.916465																			
Commonwealth	2.462149	2.164213	4.107079																		
NY	4.123129	3.852850	4.468735	4.127368																	
Florida	3.606269	4.218804	2.992760	3.201836	4.600183																
Hawaiian	3.901898	3.448346	4.217769	3.969367	4.596261	3.352919															
Idaho	2.737407	3.892524	4.990876	3.692949	5.155516	4.913953	4.364509														
Kentucky	3.253851	3.957125	2.752623	3.753627	4.489900	3.730814	2.796298	3.594824													
Madison	3.099116	2.705330	3.934935	1.491427	4.045276	3.829058	4.506512	3.673884	3.572023												
Nevada	3.491163	4.792640	5.902882	4.864730	6.460986	6.004557	5.995814	3.462587	5.175240	5.081469											
New England	3.223138	2.432568	4.031434	3.498769	3.603863	3.738824	1.660047	4.059770	2.735861	3.942171	5.208504										
Northern	3.959637	3.434878	4.385973	2.577003	4.758059	4.554909	5.010221	4.140607	3.658647	1.407032	5.309741	4.496249									
Oklahoma	2.113490	4.323825	2.742000	3.230069	4.818803	3.469268	4.914949	4.335241	3.816443	3.610272	4.315584	4.335484	4.385649								
Pacific	2.593481	2.501195	5.156977	3.190250	4.255251	4.065764	2.930142	3.849872	4.113606	4.264133	4.735659	2.328833	5.103646	4.239522							
Puget	4.033051	4.837051	5.264442	4.967244	5.816715	5.842268	5.042444	2.201457	3.627307	4.531420	3.429962	4.617791	4.406173	5.169314	5.175157						
San Diego	4.396680	3.623588	6.356548	4.893679	5.628591	6.099456	4.577294	5.426511	4.901037	5.484537	4.751387	3.497555	5.606577	5.558002	3.399659	5.559320					
Southern	1.877248	2.904409	2.723954	2.651532	4.338150	2.853942	2.949006	3.237409	2.428533	3.070750	3.945595	2.451935	3.780942	2.301050	2.998784	3.973815	4.426129				
Texas	2.410434	4.634878	3.179392	3.464171	5.133791	2.581208	4.515428	4.107966	4.109049	4.130120	4.522319	4.414578	5.010864	1.876051	4.030721	5.232256	6.089597	2.473696			
Wisconsin	3.174488	2.997481	3.733274	1.816465	4.385852	2.912401	3.541931	4.094283	2.948021	2.054393	5.352136	3.430937	2.226493	3.744430	3.782111	4.823711	4.866540	2.922392	3.903723		
United	3.453407	2.318451	5.088018	3.884260	3.644137	4.628341	2.675404	3.977130	3.742680	4.361961	4.883977	1.384124	4.937119	4.926966	2.097150	4.568885	3.095002	3.185250	4.972551	4.145222	
Virginia	2.509287	2.421916	4.109321	2.578463	3.771757	4.026935	4.000096	3.239374	3.208932	2.559945	3.436927	2.995066	2.739910	3.512207	3.352644	3.457129	3.628061	2.548060	3.967618	2.618050	3.012264

06. 실습 - 계층적 군집분석

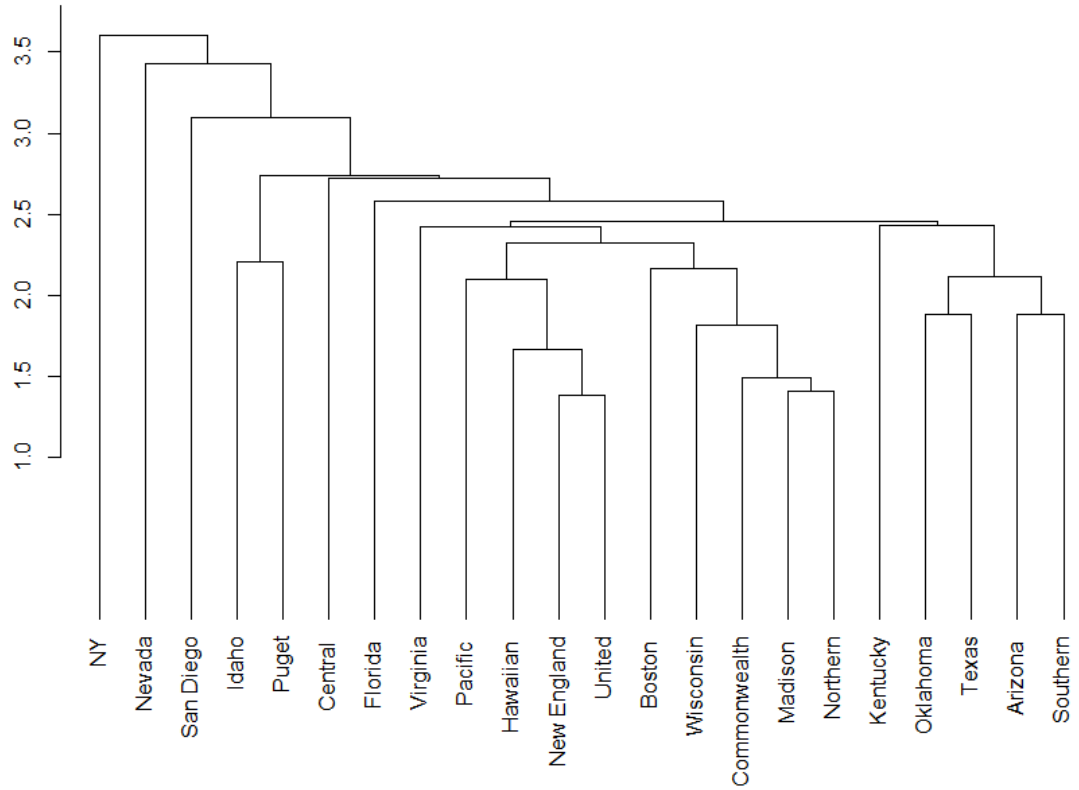
계층적 군집화 진행

```
#계층적 군집화 :hclust()
#"single","complete","average","median","centroid","ward.D"
#단일계산법(최단)
hc.s<-hclust(d.norm,method="single")
plot(hc.s,hang=-1,ann=FALSE)
#평균연결법
hc.a<-hclust(d.norm,method="average")
plot(hc.a,hang=-1,ann=FALSE)
```

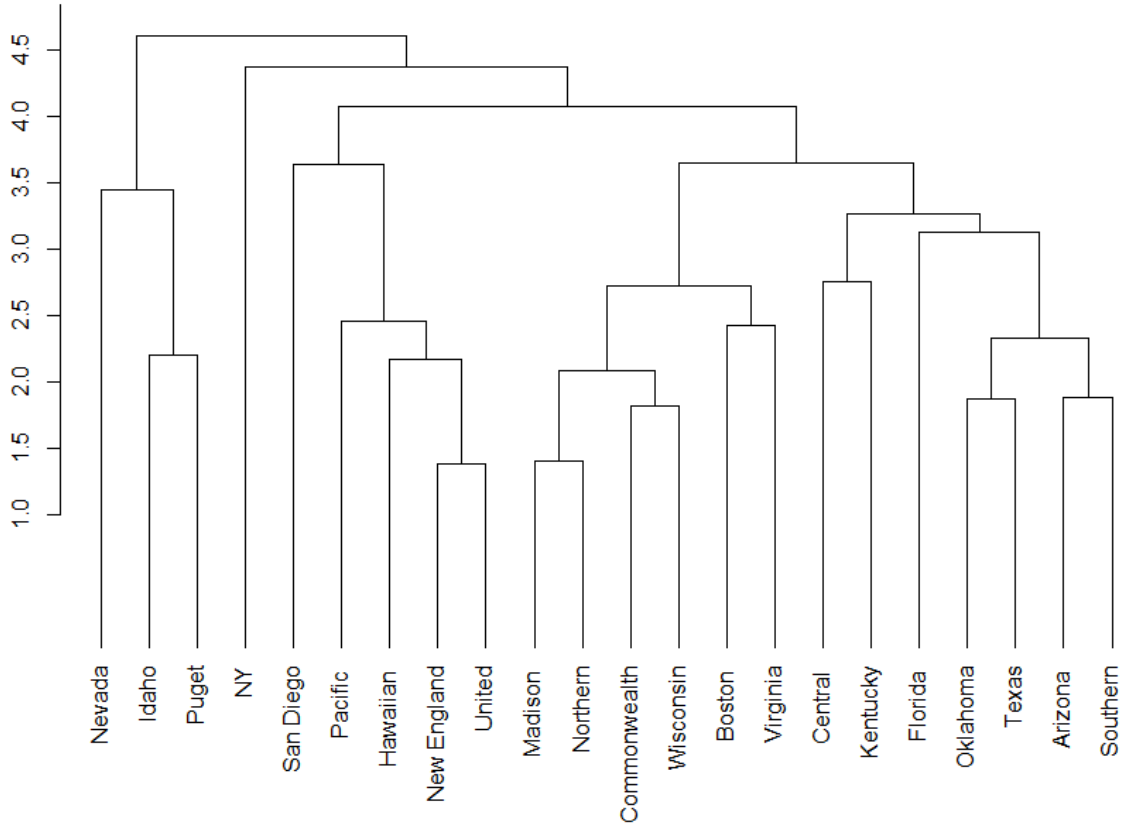
```
#군집수 결정
memb.s<-cutree(hc.s,k=6)
memb.s
  Arizona       Boston      Central  Commonwealth           NY     Florida     Hawaiian
        1            1            2            1            3           1            1
    Idaho     Kentucky     Madison       Nevada  New England     Northern     Oklahoma
        4            1            1            5            1           1            1
  Pacific        Puget    San Diego     Southern        Texas    Wisconsin       United
        1            4            6            1            1           1            1
  Virginia
        1
memb.a<-cutree(hc.s,k=6)
memb.a
  Arizona       Boston      Central  Commonwealth           NY     Florida     Hawaiian
        1            1            2            1            3           1            1
    Idaho     Kentucky     Madison       Nevada  New England     Northern     Oklahoma
        4            1            1            5            1           1            1
  Pacific        Puget    San Diego     Southern        Texas    Wisconsin       United
        1            4            6            1            1           1            1
  Virginia
        1
```

06. 실습 - 계층적 군집분석

단일계산법으로 군집화



평균연결법으로 군집화



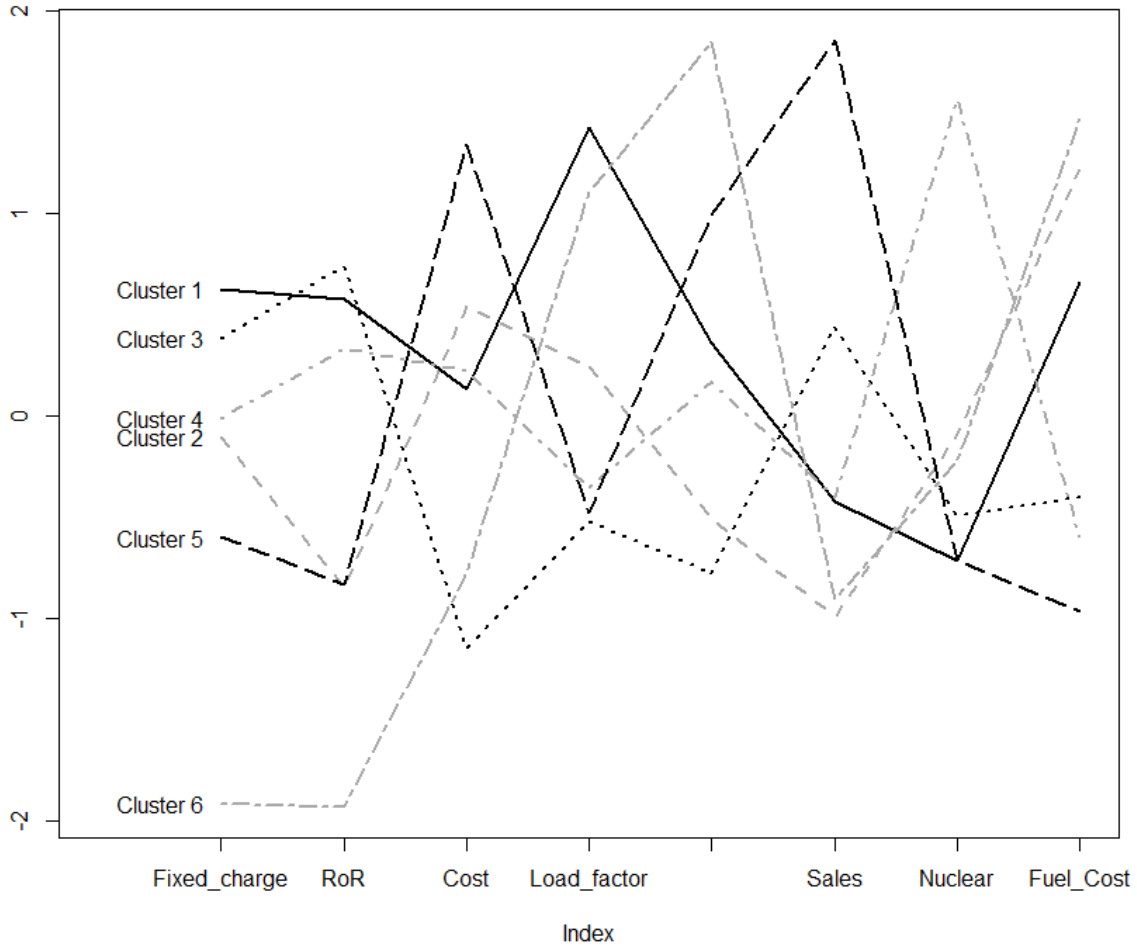
06. 실습 - 비계층적 군집분석

앞서 진행한 데이터 표준화 이후 유클리드 거리 계산까지 모두 동일하게 진행
이후 K-Means 군집화 진행

```
> set.seed(2)
> km<-kmeans(utilities.norm,6)
> km
K-means clustering with 6 clusters of sizes 3, 4, 6, 5, 3, 1

Cluster means:
  Fixed_charge        RoR       Cost Load_factor Demand_growth      Sales    Nuclear  Fuel_Cost
1   0.62819552  0.5779595  0.1331553   1.4247590     0.3610222 -0.4263058 -0.7146294  0.6610454
2  -0.10346750 -0.8517476  0.5418172   0.2460639    -0.5101929 -0.9964962 -0.0923063  1.2156538
3   0.38430785  0.7413546 -1.1494764  -0.5216758    -0.7827816  0.4343553 -0.4913077 -0.4068118
4  -0.01133215  0.3313815  0.2189339  -0.3580408     0.1664686 -0.4018738  1.5650384 -0.5954476
5  -0.60027572 -0.8331800  1.3389101  -0.4805802     0.9917178  1.8565214 -0.7146294 -0.9657660
6  -1.91907572 -1.9323833 -0.7812761   1.1034665     1.8468982 -0.9014253 -0.2203441  1.4696557

Clustering vector:
    Arizona      Boston     Central Commonwealth          NY     Florida    Hawaiian       Idaho
          3           2           3            4           2           3           1           5
   Kentucky     Madison      Nevada New England    Northern    Oklahoma     Pacific      Puget
          1           4           5            1           4           3           2           5
  San Diego    Southern       Texas   Wisconsin      United    Virginia
          6           3           3            4           2           4

Within cluster sum of squares by cluster:
[1]  6.019991 15.565144 19.353940 10.177094  9.533522  0.000000
 (between_SS / total_SS =  63.9 %)

Available components:

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

06. 실습 - 비계층적 군집분석

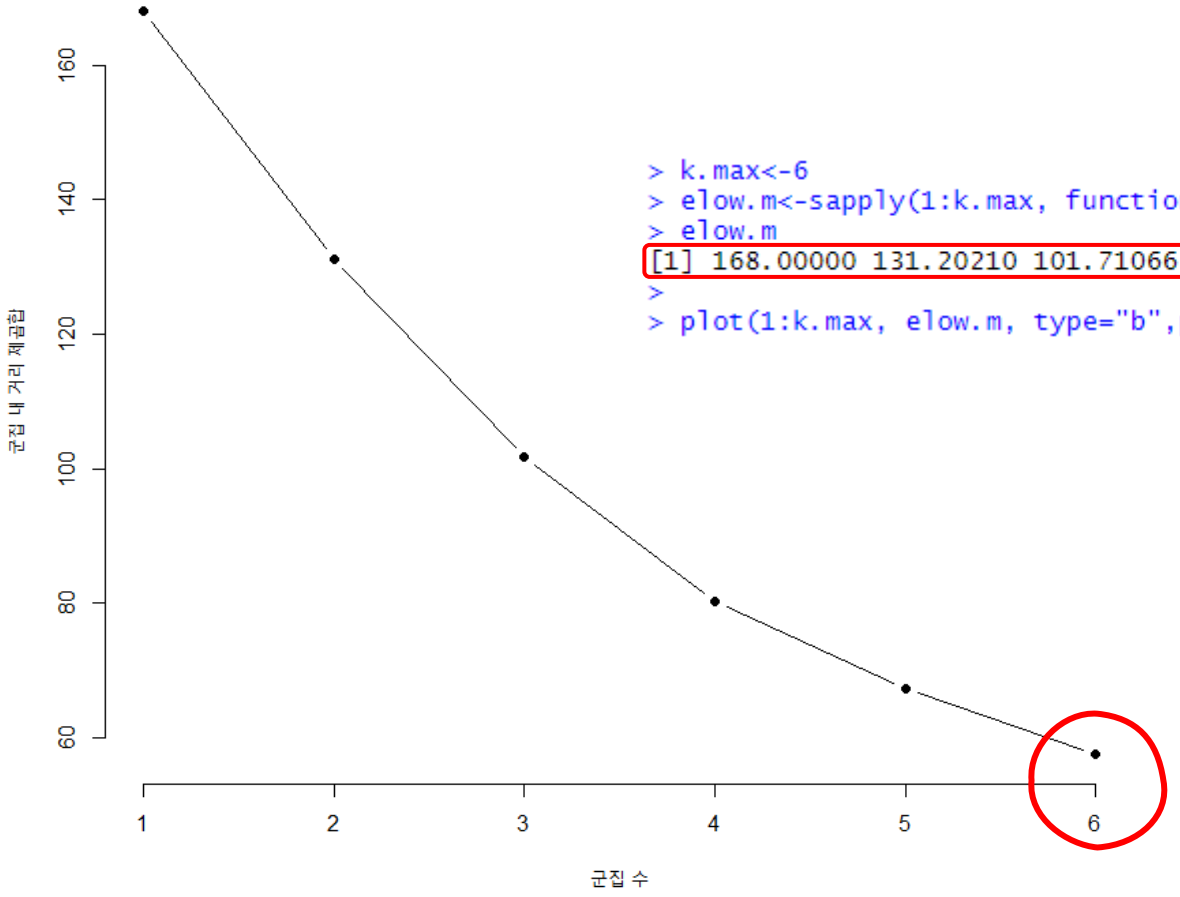
군집 중심에 대한 시각적 표현 - 군집별 해석 가능



06. 실습 - 비계층적 군집분석

Elbow Chart

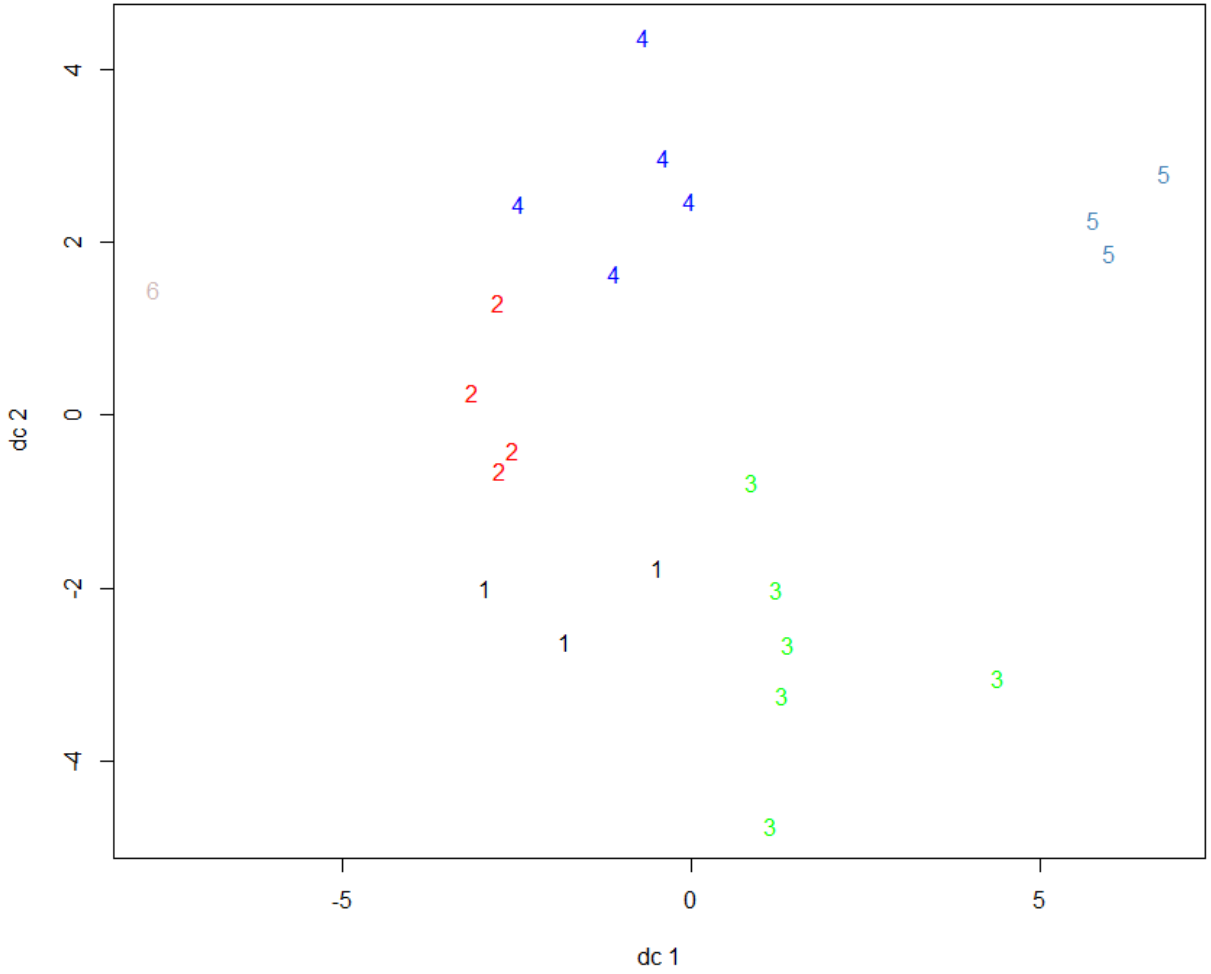
: 더 많은 군집이 추가되면서 군집 이산성이 감소하는 것을 확인할 수 있다.




```
> k.max<-6
> elow.m<-sapply(1:k.max, function(k){kmeans(utilities.norm,k,nstart=50, iter.max=15)$tot.withinss})
> elow.m
[1] 168.00000 131.20210 101.71066  80.38320  67.40636  57.65863
>
> plot(1:k.max, elow.m, type="b",pch=19, frame=FALSE, xlab="군집 수", ylab="군집 내 거리 제곱합")
```

06. 실습 - 비계층적 군집분석

K-Means 비계층적 군집분석 최종 시각화



감사합니다