# 07. R 정형데이터 분석 03
# 비지도학습

성현곤

충북대학교 도시공학과
Dept.of URBAN ENGINEERING

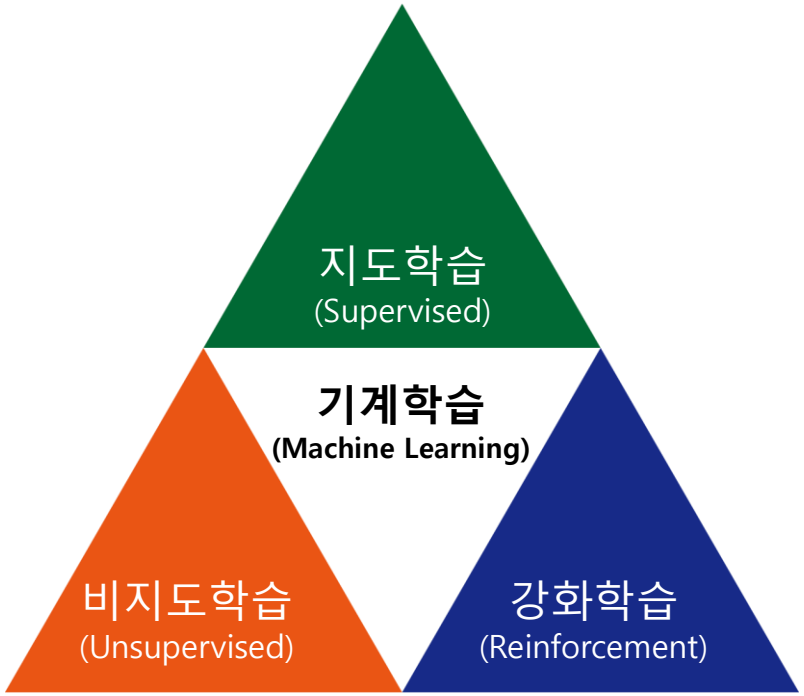
# 목차


- 분류와 머신러닝
  - 머신러닝: 지도학습과 비지도 학습
  - 통계모델과 머신러닝의 차이
  - 분류 알고리즘의 종류
  - 분류 모델의 평가
  - 교차타당성(cross validation)
- 분류와 예측
  - 로지스틱 회귀모델
    - 이항 로지스틱 모델
    - 일반화 가법 모델(gam)
  - 판별분석
    - 선형 판별모델
    - 비선형 판별모델
- 분류와 기계학습
  - K-근접 이웃(KNN) 분류
  - 결정트리 학습법
  - 앙상블 학습
    - 배깅과 랜덤 포레스트
    - 부스팅과 XGBoost
- 비지도학습
  - 비지도학습의 개념
  - 주성분 분석
    - 주성분 분석
  - 군집(클러스터링) 분석
    - K-평균 클러스터링
    - 계층적 클러스터링

# 비지도 학습의 개념

- 기계학습(Machine Learning)
  - 데이터를 이용해서 컴퓨터를 학습시키는 방법론
  - 규칙기반 프로그래밍에 기반하지 않고 데이터로 부터 학습하는 알고리즘
- 기계학습 알고리즘의 종류
  - 지도 학습(Supervised Learning)
    - 데이터에 대한 **레이블(Label)-명시적인 정답-이 주어진 상태에서 컴퓨터를 학습시키는 방법**
    - 학습된 알고리즘으로 얼마나 예측(Prediction)하는 데 사용
      - 예측하는 결과값이 **discrete value(이산값)면 classification(분류) 문제**
      - 결과값이 **continuous value(연속값)면 regression(회귀) 문제**
  - 비지도 학습(Unsupervised Learning)
    - 데이터에 대한 **레이블(Label)-명시적인 정답-**이 주어지지 상태에서 컴퓨터를 학습시키는 방법론
      - 예: **클러스터링(Clustering)**
    - **데이터의 숨겨진(Hidden) 특징(Feature)이나 구조를 발견**하는데 사용
  - 강화 학습(Reinforcement Learning)
    - **에이전트가 주어진 환경(state)에 대해 어떤 행동(action)을 취하고 이로부터 어떤 보상(reward)을 얻으면서 학습**을 진행하는 방법
    - 에이전트는 보상(reward)을 최대화(maximize)하도록 학습
    - 일종의 동적인 상태(dynamic environment)에서 데이터를 수집하는 과정까지 포함되어 있는 알고리즘
      - 예: Q-Learning, Deep-Q-Network(DQN) , 알파고

- Labeled data
- Direct feedback
- Predict outcome/future




- No labels
- No feedback
- "Find hidden structure

- Decision process
- Reward system
- Learn series of actions

출처: http://solarisailab.com/archives/1785

# 비지도 학습

## 주성분 분석

- 주성분분석(Principal Component Analysis)의 개념
  - 차원축소(dimensionality reduction)와 변수추출(feature extraction) 기법
  - 데이터의 분산(variance)을 최대한 보존하면서 서로 직교하는 새 기저(축)를 찾아, 고차원 공간의 표본들을 선형 연관성이 없는 저차원 공간으로 변환하는 기법
    - 데이터 프레임의 총 변동을 대부분 설명할 수 있는 변수 선형 조합을 찾아내는 것
    - 직교 관계의 표준선형 결합 집합
  - 서로 연관되어 있는 변수들의 정보를 최대한 확보하면서 적은 수의 새로운 변수들(주성분)을 생성하는 방법
  - 여러 개인 변수들의 변이(Variation)을 결정하는 데이들 상관 구조를 활용하여 더 낮은 차원의 상호 독립적 요인, 즉 주성분(principle component)를 찾아냄
- 차원축소의 방법
  - 특징선택(Feature Selection)
    - 일부 중요 변수만을 추출
  - 특징추출(Feature Extraction)
    - 기존 변수를 조합해 새로운 변수를 만드는 기법
    - 예: PCA
- 기타 차원축소 방법들
  - 요인분석(FA)
  - 독립성분분석(ICA)
  - 다차원 척도법(MDA)
  - 비선형 차원 축소법 등

출처1: https://datacookbook.kr/35
출처2: https://ratsgo.github.io/machine%20learning/2017/04/24/PCA/
출처3: https://rpubs.com/Evan_Jung/pca

비지도 학습

# 주성분 분석

- 주성분분석(Principal Component Analysis)의 목적
  - 어떤 한 대상을 설명하는 데 있어 많은 변수들을 사용하기 보다 정보의 손실을 최소화하면서 이들의 차원을 몇개의 중요한 성분으로 축소하여 달리 표현하고자 할 때 사용
    - 정보의 손실을 최소화하면서 중요한 주요 정보를 활용
    - 여러 개의 변수들에서 존재하는 오차와 잡음을 제거하면서 공통된 주요 정보만 추출
  - 적은 수의 특징(주성분)만으로 특정 현상을 설명하고자 할 때 사용
- 주성분 분석의 활용
  - 회귀분석에서 설명변수의 축소
    - 다중공선성이 존재할 경우 해결 방법 중의 하나가 바로 상관도가 높은 변수들을 하나의 주성분 혹은 요인으로 축소하여 모형개발에 활용
  - 인자분석의 전초작업(즉, 인자를 구하는 방법으로 이용)
  - 군집분석의 전초작업(즉, 입력변수로 이용)
    - 원을 축소한 후에 군집분석을 수행하면 군집화 결과, 연산속도 개선
  - 고차원으로 인해 야기될 수 있는 모델의 성능 저하를 저차원으로 변환하여 모델의 성능을 강화
    - 기계에서 나오는 다수의 센서데이터를 주성분분석이나 요인분석을 하여 차원을 축소한 후에 시계열로 분포나 추세의 변화를 분석하면 기계의 고장(fatal failure) 징후를 사전에 파악하는데 활용

출처1: https://datacookbook.kr/35
출처2: https://ratsgo.github.io/machine%20learning/2017/04/24/PCA/
출처3: https://rpubs.com/Evan_Jung/pca

비지도 학습

# 주성분 분석

- 주성분분석(Principal Component Analysis)
  - 직교 관계의 표준선형 결합 집합
    - 변수들간 상호 상관성이 없는, 즉 독립적인 선형 결합(변환)

정방행렬 A에 대하여 다음이 성립하는 0이 아닌 벡터 x가 존재할 때

$Ax = \lambda x$ (상수 $\lambda$)

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

**상수 λ**를 행렬 A의 **고유값(eigenvalue)**,
**x**를 이에 대응하는 **고유벡터(eigenvector)** 라고 함

- 고유벡터(eigenvector)
  - 원 데이터의 분산을 최대화하는 직교행렬(주성분)
    - 서로 다른 고유벡터끼리는 서로 직교(orthogonal)하게 되어 상관성이 없음

출처: https://rfriend.tistory.com/61

비지도 학습

# 주성분 분석

- 주성분 개수의 결정
  - 총 분산에 대한 공헌도
    - Cumulative Proportion의 값이 80% 이상
  - 개별 고유값의 크기
    - 분산값 즉 표준편차(Standard deviation)의 제곱의 수치가 1이상인 경우
    - 즉, 주성분이 원래 변수들의 1개 이상의 분산을 설명하는 경우
  - 스크리 그래프(검정)을 통한 판단 :
    - 수평축에 주성분을 놓고 수직축에 해당 주성분에 대응하는 고유값을 연결한 차트임
    - 그래프가 완만해 지는 부분이전 까지의 주성분을 선택
  - 위의 세 개 기준에서 연구자가 임의로 판단 가능

출처1: https://datacookbook.kr/35
출처2: https://ratsgo.github.io/machine%20learning/2017/04/24/PCA/
출처3: https://rpubs.com/Evan_Jung/pca

비지도 학습

# 주성분 분석

- 실습: 아파트 거래가격 특성변수 차원 축소

데이터 불러오기

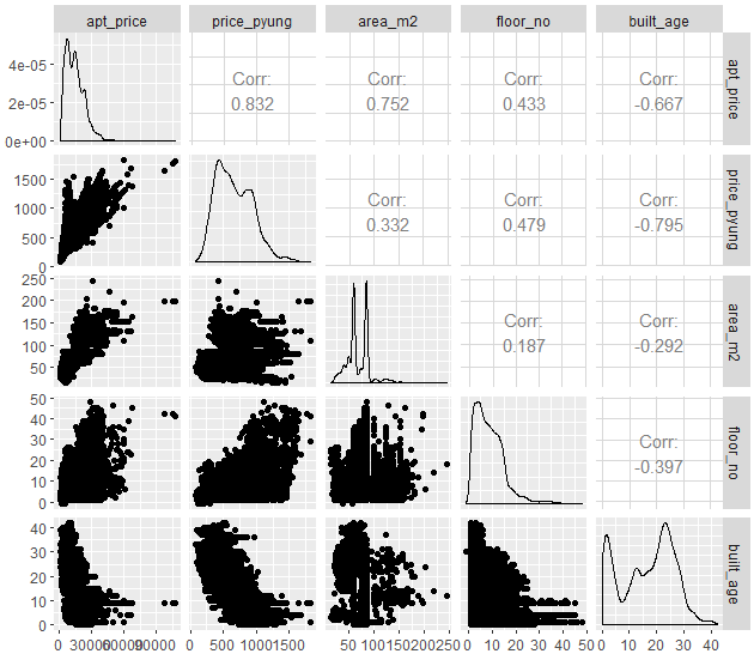
데이터 탐색하기

주성분 분석

주성분 데이터 산출

결과 활용:
분류, 예측, 군집

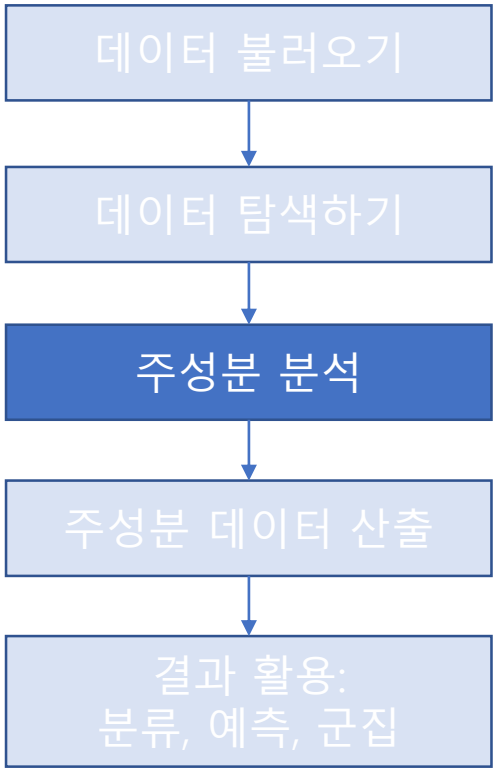
```
> ## 주성분 분석
> ### 사용하게 될 패키지와 작업 폴더 파일 확인
> library(dplyr)
> library(ggplot2)
> library(GGally)
> setwd("K:\\기타\\2019년2학기\\수치해석\\실습데이터") # 실습데이터가 있는 폴더로 작업폴더 변경
> ### 1. 데이터 불러오기와 탐색
> df.apt <- read.csv("apt3.csv") # csv파일 불러오기
> df2.apt <- df.apt %>%
+   select(apt_price, # 데이터 열 선택: 아파트 실거래가격(만원)
+          price_pyung, # 평당가격(만원)
+          area_m2, # 전용면적(m2)
+          floor_no, # 층수
+          built_age) # 건축연령
> round(cov(df2.apt), 3) # 공분산 행렬: 대각선은 분산,이외는 공분산
             apt_price price_pyung    area_m2  floor_no  built_age
apt_price   76584900.79 1986474.962 153445.221 24734.022 -57675.964
price_pyung  1986474.96   74366.382   2112.101   851.773  -2142.107
area_m2       153445.22    2112.101    543.913    28.487    -67.267
floor_no       24734.02     851.773     28.487    42.595    -25.621
built_age     -57675.96   -2142.107    -67.267   -25.621     97.544
> round(cor(df2.apt), 3) # 상관 행렬
            apt_price price_pyung area_m2 floor_no built_age
apt_price       1.000        0.832   0.752    0.433    -0.667
price_pyung     0.832        1.000   0.332    0.479    -0.795
area_m2         0.752        0.332   1.000    0.187    -0.292
floor_no        0.433        0.479   0.187    1.000    -0.397
built_age      -0.667       -0.795  -0.292   -0.397     1.000
> ?ggpairs() # A ggplot2 generalized pairs plot
> ggpairs(df2.apt) # 객체 전체 데이터 산점도 매트릭스 그래프 작성
```



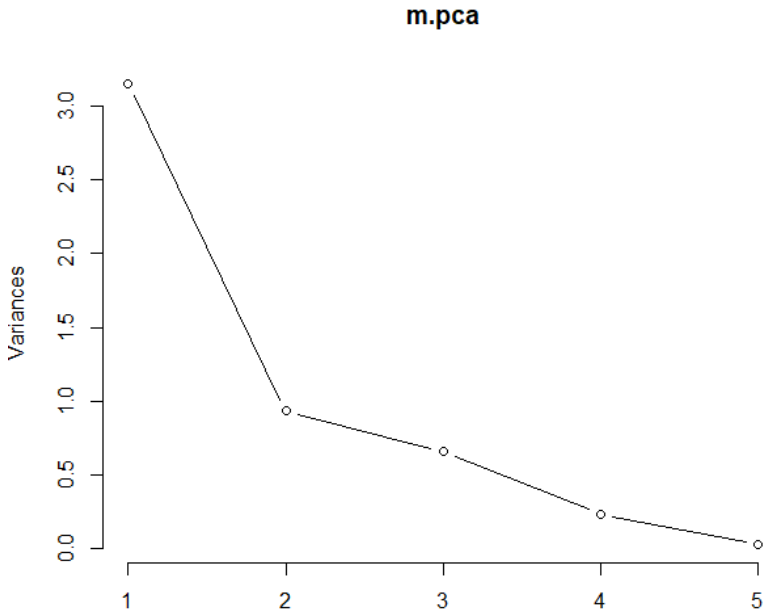
비지도 학습

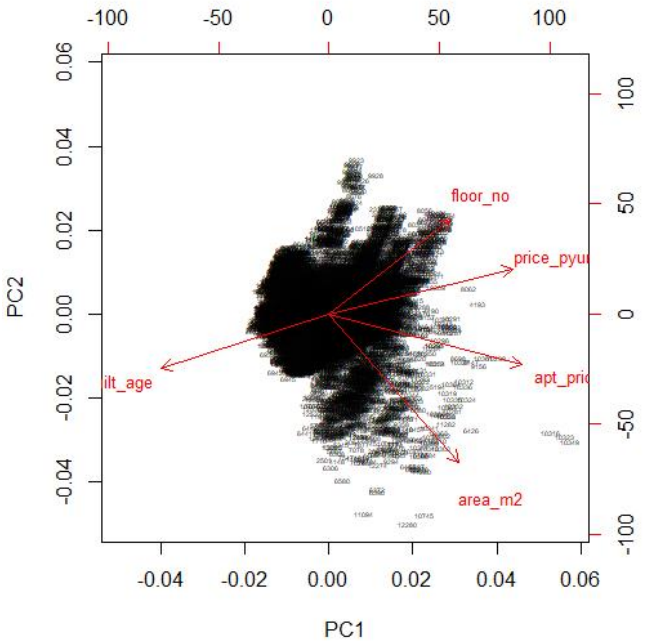
# 주성분 분석

- 실습: 아파트 거래가격 특성변수 차원 축소




```
> ### 2. 차원 축소: 주성분 분석
> ?prcomp() # Principal Components Analysis
> m.pca <- prcomp(df2.apt, # df2.apt 데이터 전체 변수들
+                 center = TRUE, # 평균 중심화: 평균을 0으로
+                 scale = TRUE) # 분산을 1로..
> summary(m.pca) # 결과요약
Importance of components:
                          PC1    PC2    PC3     PC4     PC5
Standard deviation     1.7751 0.9648 0.8105 0.48139 0.17152
Proportion of Variance 0.6302 0.1862 0.1314 0.04635 0.00588
Cumulative Proportion  0.6302 0.8164 0.9478 0.99412 1.00000
> print(m.pca) # 주성분 점수 출력
Standard deviations (1, .., p=5):
[1] 1.7751491 0.9647875 0.8104776 0.4813927 0.1715169

Rotation (n x k) = (5 x 5):
                PC1        PC2         PC3        PC4         PC5
apt_price    0.5334942 -0.2600447  0.01850555 -0.3215244 -0.73759090
price_pyung  0.5082991  0.2304954  0.28887995 -0.5614359  0.53836984
area_m2      0.3573046 -0.7550549 -0.26506966  0.2636248  0.40307015
floor_no     0.3373521  0.4818351 -0.80241759  0.1002419  0.01030014
built_age   -0.4642661 -0.2774449 -0.44952196 -0.7084246  0.05954781
> plot(m.pca, # 주성분 결과 스크리 검정 그래프
+      type="l") # 선의 유형 = line
> ?biplot() # Biplot of Multivariate Data
> biplot(m.pca, # 주성분 2개 축 그래프 작성
+        cex = c(0.3, 0.8)) # 점의 크기와 텍스트 크기 비율
```

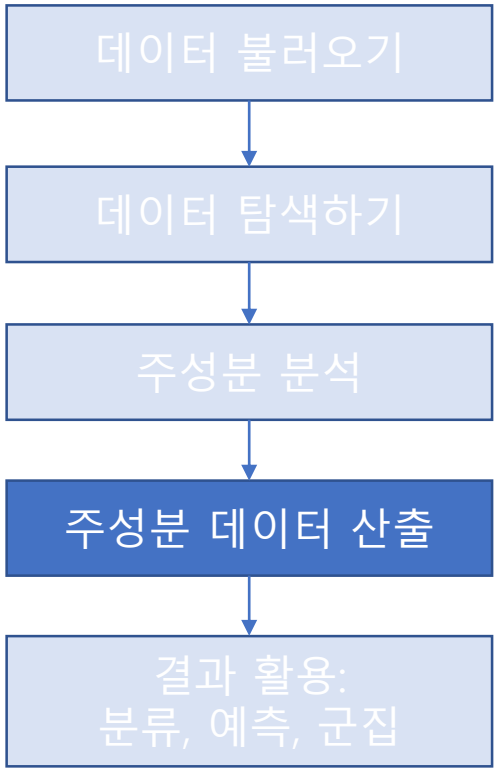


# 비지도 학습

## 주성분 분석

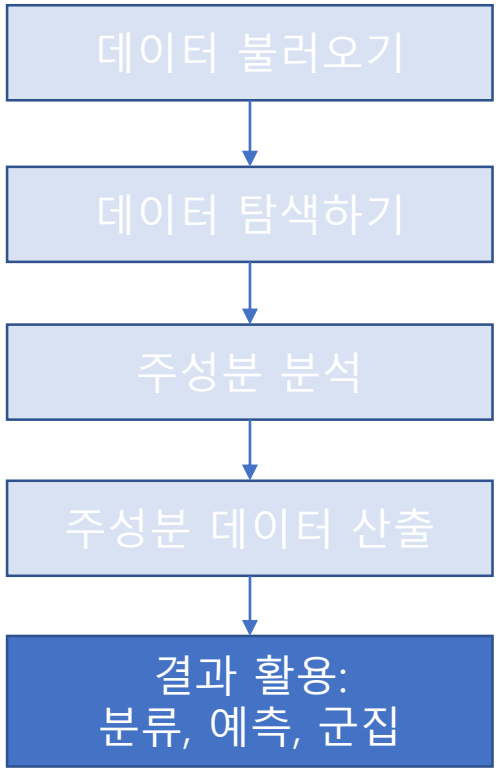
```
> ### 3. 주성분 점수 산출
> # 원래 데이터와 선형계수를 메트릭스 곱으로 관측치별 선형조합 주성분 값 산출
> pr.c <- as.matrix(df2.apt) %*% # 행렬로 변환하고 행렬 곱 연산(%*%)
+                 m.pca$rotation # m.pca의 Rotation 값
> head(pr.c) # 데이터 일부 확인
          PC1         PC2       PC3         PC4         PC5
[1,] 2768.216 -1191.3936 182.2034 -1770.947 -3308.857
[2,] 2596.873 -1117.1863 166.2564 -1660.123 -3100.930
[3,] 2309.391  -996.2378 144.2250 -1475.985 -2754.444
[4,] 2310.066  -995.2741 142.6201 -1475.785 -2754.423
[5,] 2308.041  -998.1652 147.4347 -1476.386 -2754.485
[6,] 2309.391  -996.2378 144.2250 -1475.985 -2754.444
> pr.c.df <- as.data.frame(pr.c) # 데이터 프레임으로 변환
> str(pr.c.df) # 데이터 구조 확인
'data.frame':   13309 obs. of  5 variables:
 $ PC1: num  2768 2597 2309 2310 2308 ...
 $ PC2: num  -1191 -1117 -996 -995 -998 ...
 $ PC3: num  182 166 144 143 147 ...
 $ PC4: num  -1771 -1660 -1476 -1476 -1476 ...
 $ PC5: num  -3309 -3101 -2754 -2754 -2754 ...
> df3.apt <- cbind(df.apt, # 데이터 열 결합
+                  pr.c.df)
> summary(df3.apt) # 요약 통계
       X           apt_price        area_m2          floor_no        year_built      ym_sale          day_sale      urban      price_pyung
 Min.   :    1   Min.   :  1250   Min.   : 14.98   Min.   :-1.000   Min.   :1977   Min.   :201809   Min.   : 1.00   동:8848   Min.   :  85.48
 1st Qu.: 3328   1st Qu.:  7700   1st Qu.: 58.93   1st Qu.: 4.000   1st Qu.:1995   1st Qu.:201811   1st Qu.: 8.00   면:1332   1st Qu.: 444.44
 Median : 6655   Median : 13500   Median : 60.00   Median : 8.000   Median :2001   Median :201903   Median :15.00   읍:3129   Median : 627.35
 Mean   : 6655   Mean   : 14553   Mean   : 68.92   Mean   : 8.835   Mean   :2003   Mean   :201874   Mean   :15.63             Mean   : 666.15
 3rd Qu.: 9982   3rd Qu.: 19500   3rd Qu.: 84.89   3rd Qu.:12.000   3rd Qu.:2012   3rd Qu.:201906   3rd Qu.:23.00             3rd Qu.: 866.05
 Max.   :13309   Max.   :108000   Max.   :244.07   Max.   :48.000   Max.   :2019   Max.   :201908   Max.   :31.00             Max.   :1825.47
 yr_built2    season      urban2      built_age          PC1                PC2                PC3                PC4                PC5
 New:7097   가을:3419   농촌:4461   Min.   : 0.00   Min.   :  721.9   Min.   :-27799.2   Min.   :  18.21   Min.   :-35690.8   Min.   :-78605.4
 old:6212   겨울:3004   도시:8848   1st Qu.: 7.00   1st Qu.: 4380.2   1st Qu.: -4960.9   1st Qu.: 243.40   1st Qu.: -6780.3   1st Qu.:-13940.1
            봄  :3510               Median :18.00   Median : 7493.0   Median : -3385.1   Median : 406.35   Median : -4667.1   Median : -9528.8
            여름:3376               Mean   :16.39   Mean   : 8122.4   Mean   : -3683.2   Mean   : 429.02   Mean   : -5045.6   Mean   :-10346.5
                                    3rd Qu.:24.00   3rd Qu.:10942.7   3rd Qu.: -1927.6   3rd Qu.: 574.35   3rd Qu.: -2749.6   3rd Qu.: -5383.8
                                    Max.   :42.00   Max.   :58617.1   Max.   :  -337.7   Max.   :2432.13   Max.   :  -467.6   Max.   :  -848.7
```



# 비지도 학습

## 주성분 분석

- 실습: 아파트




```
> ### 4. 주성분 결과 활용: 랜덤포레스트 학습모델링
> library(randomForest) # 미설치시 install.packages("randomForest")
> m.rf.apt <- randomForest(urban ~ # 분류항목 urban 종속변수
+                          PC1 + PC2 + season, # 설명변수 PC1, PC2, season
+                          data=df3.apt, # 사용할 데이터 객체
+                          importance = TRUE) # 변수 중요도 측정 여부
> m.rf.apt # 모델 훈련에 사용되지 않은 데이터를 사용한 에러 추정치가 'OOBOut of Ba

Call:
 randomForest(formula = urban ~ PC1 + PC2 + season, data = df3.apt,
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 30.66%
Confusion matrix:
     동 면  읍 class.error
동 8417  1 430  0.04871157
면 1196 21 115  0.98423423
읍 2338  0 791  0.74720358
> pred.y <- predict(m.rf.apt, # 학습모델 예측
+                   data=df3.apt # 사용할 데이터
+                   , type = "class") # 분류항목으로 예측
> table(pred.y, # 빈도분포표: 예측치
+       df3.apt$urban)  # 실측치

pred.y   동   면   읍
    동 8417 1196 2338
    면    1   21    0
    읍  430  115  791
> library(caret) # caret 패키지의 혼동행렬
> confusionMatrix(pred.y, # 혼동 행렬: 예측치
+                 df3.apt$urban) # 시험데이터 실측치
Confusion Matrix and Statistics

          Reference
Prediction   동   면   읍
        동 8417 1196 2338
        면    1   21    0
        읍  430  115  791
> library(caret) # caret 패키지의 혼동행렬
> confusionMatrix(pred.y, # 혼동 행렬: 예측치
+                 df3.apt$urban) # 시험데이터 실측치
Confusion Matrix and Statistics

          Reference
Prediction   동   면   읍
        동 8417 1196 2338
        면    1   21    0
        읍  430  115  791

Overall Statistics

               Accuracy : 0.6934
                 95% CI : (0.6855, 0.7013)
    No Information Rate : 0.6648
    P-Value [Acc > NIR] : 9.731e-13

                  Kappa : 0.1917

 Mcnemar's Test P-Value : < 2.2e-16
```

# 연습문제 01

- "df.seoul.worker.csv" 데이터를 불러들여, 서울 동남권 거주자(area.living == 5)와 아래에 해당하는 변수들만을 추출하여 df2에 할당하라는 명령문을 작성하시오.

age, # 만나이
edu.level, # 교육수준
hh.income, # 총가구 소득
commuting.time, # 통근시간(분)
commuting.area, # 통근 목적지
feel.commuting) # 통근 만족도

# 연습문제 02

- 연습문제 01에서의 df2객체를 활용하여 표준화(center, scale)한 데이터로 주성분 분석을 실행하시오.

  - 이 때, 첫번째 주성분의 고유값은 얼마인가요?

  - 주성분 분석 결과로 볼 때, 몇 개의 주성분으로 차원을 축소하는 것이 좋을까요?

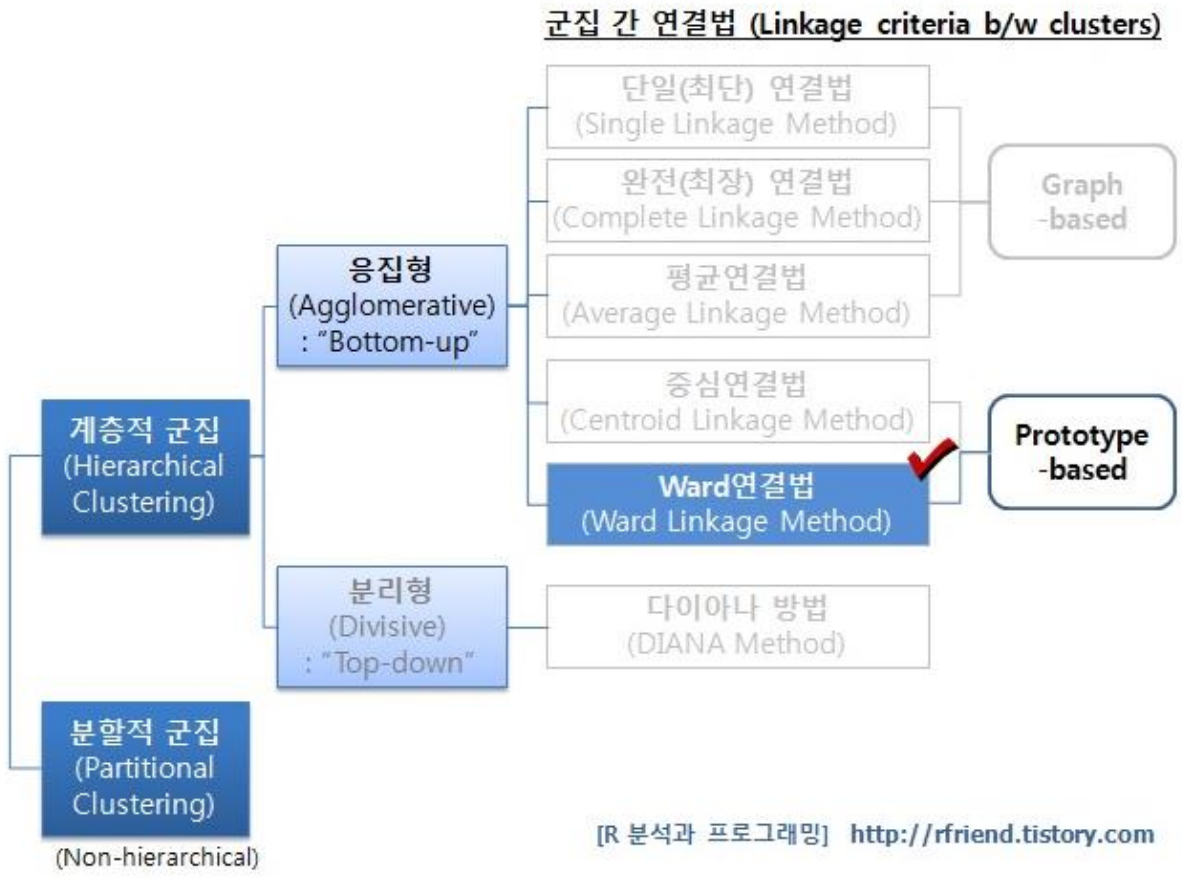
# 군집분석

- 군집분석의 개념
  - 주어진 데이터 내에 특징별로 군집화하는 비지도(unsupervised) 기법
  - 다른 그룹에 속한 다른 관찰치들에 비해 서로 보다 유사한 관찰치들의 그룹
- 군집분석의 활용분야
  - 생물학, 행동과학, 마케팅 및 의학 등 다양한 분야 활용
    - 우울증 환자의 중상과 인구학적인 특징에 관한 데이타로 군집분석
    - 고객의 인구학적인 특징 및 소비 성향의 유사성에 근거해 몇개의 군집으로 나눔으로써 맞춤형 마케팅을 하는데 활용
    - 유전자 데이터를 이용하여 유사한 표현형 및 공통되는 생물학적 경로를 가지는 유전자와 단백질들을 grouping하는데 활용

# 비지도 학습

## 군집분석

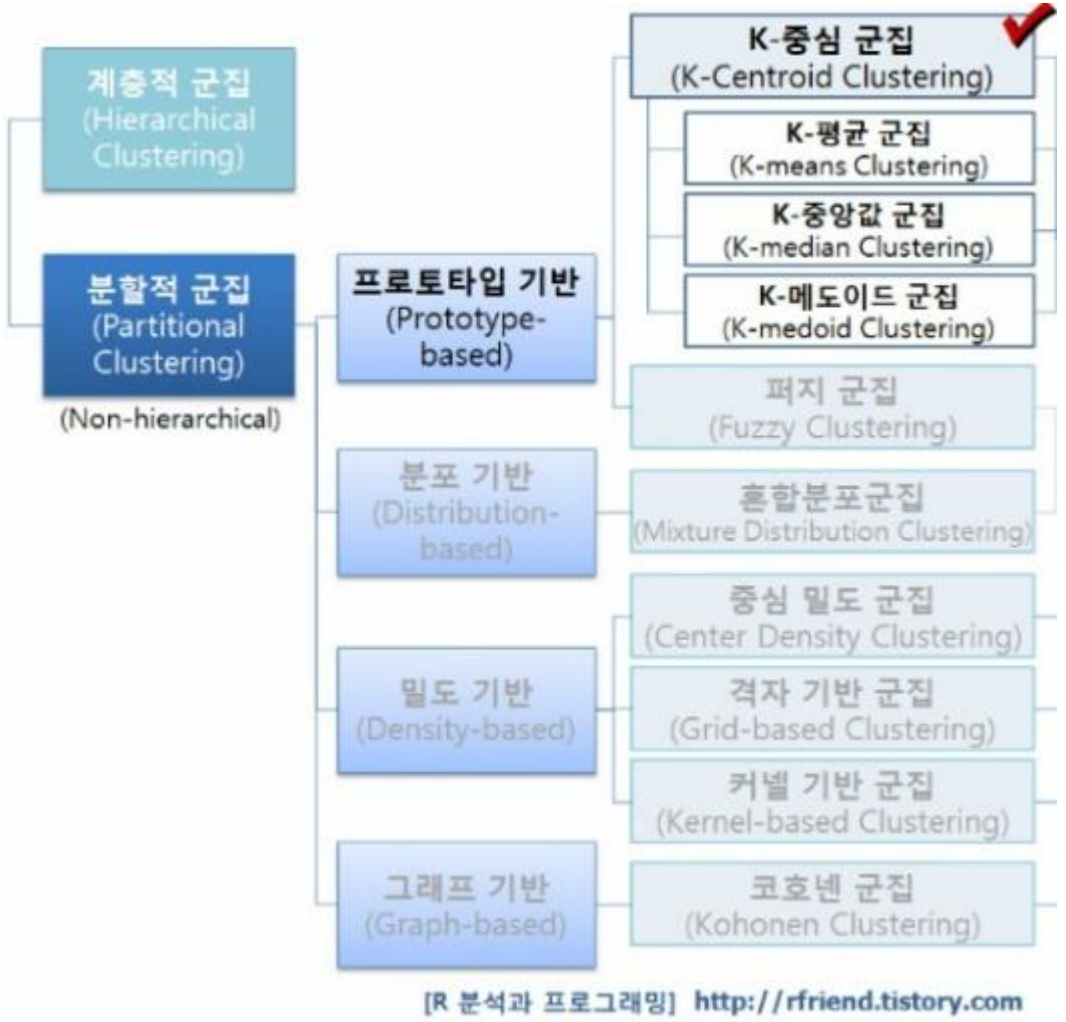
- 군집분석의 종류
  - 응집형 계층적 군집화(Agglomerative Hierarchical Clustering)
    - 각각의 객체에서 시작해서 유사성 척도(proximity measure)에 의거해 유사한 (거리가 짧은) 객체들을 Tree 형태의 계층적 군집으로 차근 차근 묶어가는 기법
    - '유클리드 제곱거리(euclidean squared distance) 기반: 단일연결법, 완전연결법, 평균연결법, 중심연결법
    - **오차 제곱합(ESS : error sum of squares)** 증가분의 거리기반: Ward연결법
      - Ward는 이 방법을 제시했던 학자인 Joe H. Ward 의 이름을 딴 것
  - 분할적 군집화(Partitional Clustering) 기법




출처: https://rfriend.tistory.com/227 [R, Python 분석과 프로그래밍의 친구 (by R Friend)]

# 군집분석

- 군집분석의 종류
  - 응집형 계층적 군집화(Agglomerative Hierarchical Clustering)
  - 분할적 군집화(Partitional Clustering) 기법
    - 객체가 하나의 군집에 exclusive하게 속하도록 군집을 형성
    - 프로토타입 기반(Prototype-based), 분포 기반(distribution-based), 밀도 기반(Density-based), 그래프 기반(Graph-based) 기법




출처: https://rfriend.tistory.com/227 [R, Python 분석과 프로그래밍의 친구 (by R Friend)]

# 군집분석

- 군집분석의 유사성 = 거리(distance)
  - 거리 계산의 유형
    - **Euclidean distance** : Usual square distance between the two vectors (2 norm).

$$d(x,y)= (sum_{j=1}^{d} (x_j - y_j)^2)^(1/2)$$

    - **Maximum distance**: Maximum distance between two components of **x** and **y** (supremum norm).

$$d(x,y)= sup|x_j - y_j|, 1≤ j ≤ d$$

    - **Manhattan distance** : Absolute distance between the two vectors (1 norm).

$$d(x,y)= sum_{j=1}^{d}|x_j - y_j|$$

    - **Canberra distance** : Terms with zero numerator and denominator are omitted from the sum and treated as if the values were missing.

$$sum_{j=1}^{d}|x_j - y_j|) / (|x_j|+|y_j|)$$

    - **Binary distance** : The vectors are regarded as binary bits, so non-zero elements are "on" and zero elements are "off". The distance is the proportion of bits in which only one is on amongst those in which at least one is on.

    - **Minkowski distance** : The **p** norm, the $p^{th}$ root of the sum of the $p^{th}$ powers of the differences of the components.

$$d(x,y)= (sum_{j=1}^{d} |x_j - y_j|^p)^(1/p)$$

# 비지도 학습

## 군집분석

- 군집분석의 유사성 = 거리(d
  - 군집 방법론

- **Kmeans** : This method is said to be a reallocation method. Here is the general principle:
  1. Select as many points as the number of desired clusters to create initial centers.
  2. Each observation is then associated with the nearest center to create temporary clusters.
  3. The gravity centers of each temporary cluster is calculated and these become the new clusters centers.
  4. Each observation is reallocated to the cluster which has the closest center.
  5. This procedure is iterated until convergence.

- **Ward** : Ward method minimizes the total within-cluster variance. At each step the pair of clusters with minimum cluster distance are merged. To implement this method, at each step find the pair of clusters that leads to minimum increase in total within-cluster variance after merging. Two different algorithms are found in the literature for Ward clustering. The one used by option "ward.D" (equivalent to the only Ward option "ward" in R versions <= 3.0.3) does not implement Ward's (1963) clustering criterion, whereas option "ward.D2" implements that criterion (Murtagh and Legendre 2013). With the latter, the dissimilarities are squared before cluster updating.

- **Single** : The distance *D_ij* between two clusters *C_i* and *C_j* is the minimum distance between two points *x* and *y*, with *x in C_i, y in C_j*.

$$D\_ij = min\ d(x,y),\ x\ in\ C\_i\ and\ y\ in\ C\_j$$

  A drawback of this method is the so-called chaining phenomenon: clusters may be forced together due to single elements being close to each other, even though many of the elements in each cluster may be very distant to each other.

- **Complete** : The distance *D_ij* between two clusters *C_i* and *C_j* is the maximum distance between two points *x* and *y*, with *x in C_i, y in C_j*.

$$D\_ij = max\ d(x,y),\ x\ in\ C\_i,\ y\ in\ C\_j$$

- **Average** : The distance *D_ij* between two clusters *C_i* and *C_j* is the mean of the distances between the pair of points x and y, where *x in C_i, y in C_j*.

$$D\_ij = sum\ d(x,y)/(n\_i * n\_j),\ x\ in\ C\_i\ and\ y\ in\ C\_j$$

  where *n_i* and *n_j* are respectively the number of elements in clusters *C_i* and *C_j*. This method has the tendency to form clusters with the same variance and, in particular, small variance.

- **McQuitty** : The distance between clusters *C_i* and *C_j* is the weighted mean of the between-cluster dissimilarities:

$$D\_ij = (D\_ik + D\_il)/2$$

  where cluster *C_j* is formed from the aggregation of clusters *C_k* and *C_l*.

- **Median** : The distance *D_ij* between two clusters *C_i* and *C_j* is given by the following formula:

$$D\_ij = (D\_ik + D\_il)/2) - (D\_kl/4)$$

  where cluster *C_j* is formed by the aggregation of clusters *C_k* and *C_l*.

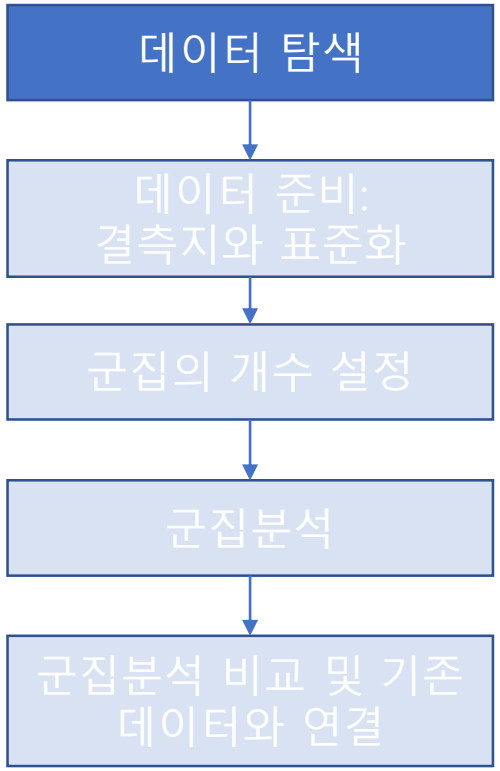
- **Centroid** : The distance *D_ij* between two clusters *C_i* and *C_j* is the squared euclidean distance between the gravity centers of the two clusters, i.e. between the mean vectors of the two clusters, $\bar{x_i}$ and $\bar{x_j}$ respectively.

$$D\_ij = ||\bar{x\_i} - \bar{x\_j}||^{2}$$

# 비지도 학습

## 군집분석

- 실습: 시군구별 특성에 따른 군집분석




```
> ### 1. 데이터 불러오기와 탐색
> library(readxl)
> df <- read_excel("data_by_sigungu_2018.xlsx", 1) # 엑셀 파일 불러오기
> df2 <- df %>%  # 필요한 변수 추출하기
+   select(id, # 시군구 id
+          area, # 행정구역 면적
+          pop_density, # 인구밀도
+          pop_tot, # 총인구수
+          pop_female, # 여성인구수
+          age_average, # 평균 나이
+          r_aged, # 65세 이상 인구 비율
+          pop_net_move, # 순인구 이동수
+          housing) # 주택수
> df2$age_average <- as.numeric(df2$age_average) # 평균 연령을 숫자형 변수로 변환
Warning message:
NAs introduced by coercion
> summary(df2) # 요약 통계
       id          area          pop_density        pop_tot          pop_female      age_average       r_aged      
 Min.   :  1   Min.   :   2.83   Min.   :   19.89   Min.   :   9975   Min.   :  4534   Min.   :35.90   Min.   : 7.09  
 1st Qu.: 66   1st Qu.:  54.71   1st Qu.:   96.58   1st Qu.:  62973   1st Qu.: 31607   1st Qu.:40.88   1st Qu.:13.15  
 Median :131   Median : 443.62   Median :  483.69   Median : 191992   Median : 95197   Median :43.00   Median :17.47  
 Mean   :131   Mean   : 438.26   Mean   : 3929.54   Mean   : 233514   Mean   :116859   Mean   :44.37   Mean   :19.78  
 3rd Qu.:196   3rd Qu.: 693.09   3rd Qu.: 5824.04   3rd Qu.: 341337   3rd Qu.:169035   3rd Qu.:48.23   3rd Qu.:26.18  
 Max.   :261   Max.   :1820.14   Max.   :27445.51   Max.   :1202628   Max.   :596793   Max.   :56.40   Max.   :38.63  
               NA's   :33        NA's   :33                                          NA's   :1       NA's   :33     
  pop_net_move       housing       
 Min.   :-2555   Min.   :  2996  
 1st Qu.: -204   1st Qu.: 26230  
 Median :  -15   Median : 65244  
 Mean   :    0   Mean   : 78746  
 3rd Qu.:   61   3rd Qu.:109767  
 Max.   : 3829   Max.   :365222  
 NA's   :32
```

# 비지도 학습

## 군집분석

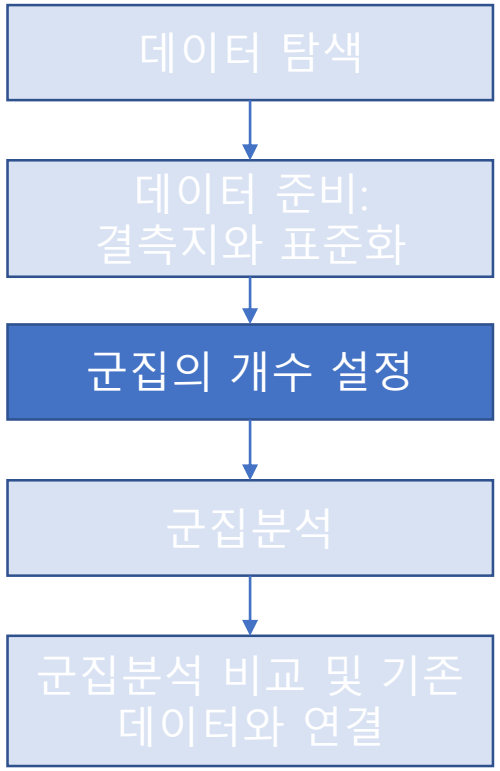
- 실습: 시군구별 특성에 따른 군집분석


```
> ### 2. 데이터 준비: 결측치 제거와 정규화
> df3 <- df2 %>% select(-id) # 분석에 사용될 데이터만 추출(id 제외)
> df3 <- na.omit(df3) # listwise deletion of missing
> df3 <- scale(df3) # standardize variables
> summary(df3) # 데이터 요약
      area             pop_density         pop_tot           pop_female        age_average          r_aged          pop_net_move
 Min.   :-1.14341   Min.   :-0.6264   Min.   :-0.9791   Min.   :-0.9809   Min.   :-1.8174   Min.   :-1.5963   Min.   :-4.539571
 1st Qu.:-1.00437   1st Qu.:-0.6140   1st Qu.:-0.7768   1st Qu.:-0.7746   1st Qu.:-0.7747   1st Qu.:-0.8395   1st Qu.:-0.351553
 Median : 0.01023   Median :-0.5514   Median :-0.3087   Median :-0.3014   Median :-0.2641   Median :-0.2762   Median :-0.009091
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.000000
 3rd Qu.: 0.66225   3rd Qu.: 0.2803   3rd Qu.: 0.5119   3rd Qu.: 0.5108   3rd Qu.: 0.8424   3rd Qu.: 0.8035   3rd Qu.: 0.125575
 Max.   : 3.60889   Max.   : 3.8323   Max.   : 4.5051   Max.   : 4.4439   Max.   : 2.4383   Max.   : 2.3654   Max.   : 6.847273
    housing
 Min.   :-1.0683
 1st Qu.:-0.7679
 Median :-0.3426
 Mean   : 0.0000
 3rd Qu.: 0.4898
 Max.   : 4.1947
```

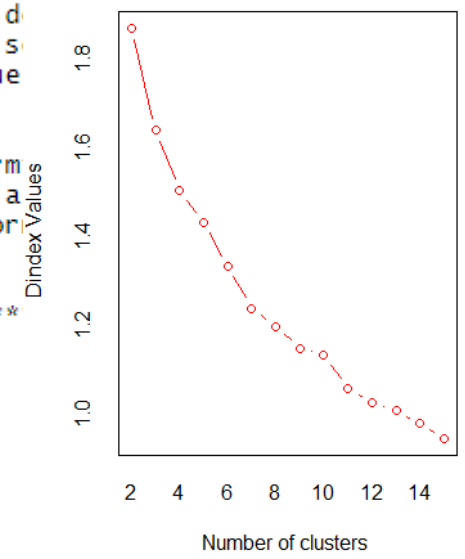
# 비지도 학습

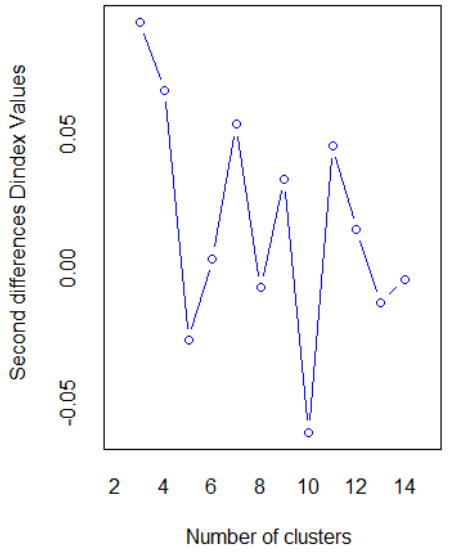
## 군집분석

- 실습: 시군구별 특성에 따른 군집분석



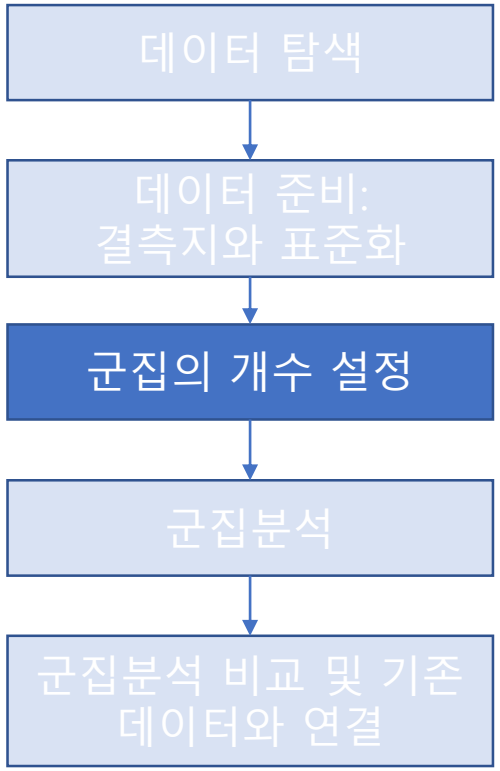

```
> ### 3. 군집의 갯수 설정Determine number of clusters
> library(NbClust) # install.packages("NbClust")
> ?NbClust() # determining the best number of clusters
> nc1 <- NbClust(df3,  # 군집화할 데이터
+                min.nc=2, # 최소군집의 수
+                max.nc=15, # 최대군집의 수
+                distance = "euclidean", # 거리계산
+                method="kmeans") # 군집 방법론
*** : The Hubert index is a graphical method of d
                In the plot of Hubert index, we s
                significant increase of the value
                index second differences plot.

*** : The D index is a graphical method of determ
                In the plot of D index, we seek a
                second differences plot) that cor
                the measure.

*******************************************************************
* Among all indices:
* 5 proposed 2 as the best number of clusters
* 8 proposed 3 as the best number of clusters
* 2 proposed 4 as the best number of clusters
* 1 proposed 5 as the best number of clusters
* 1 proposed 9 as the best number of clusters
* 2 proposed 10 as the best number of clusters
* 1 proposed 12 as the best number of clusters
* 3 proposed 15 as the best number of clusters

                ***** Conclusion *****

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



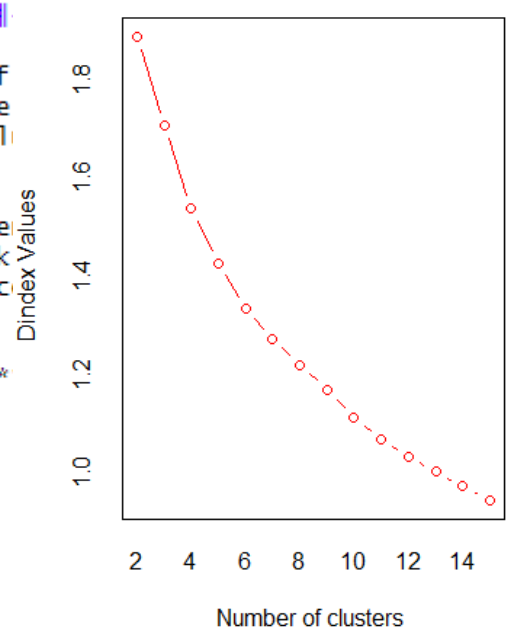



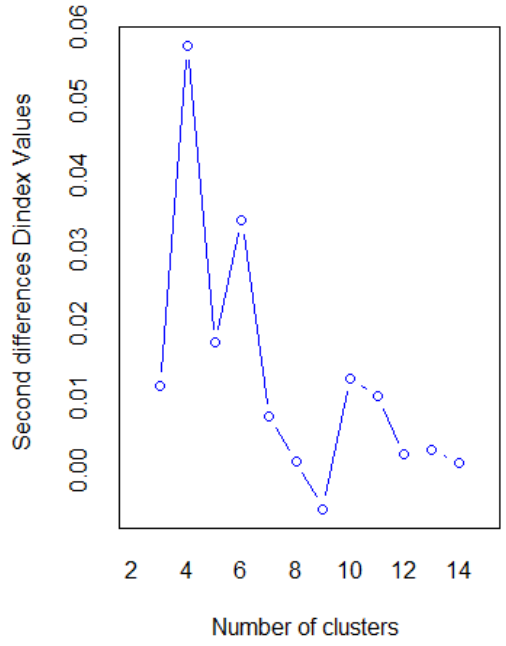
# 군집분석

- 실습: 시군구별 특성에 따른 군집분석



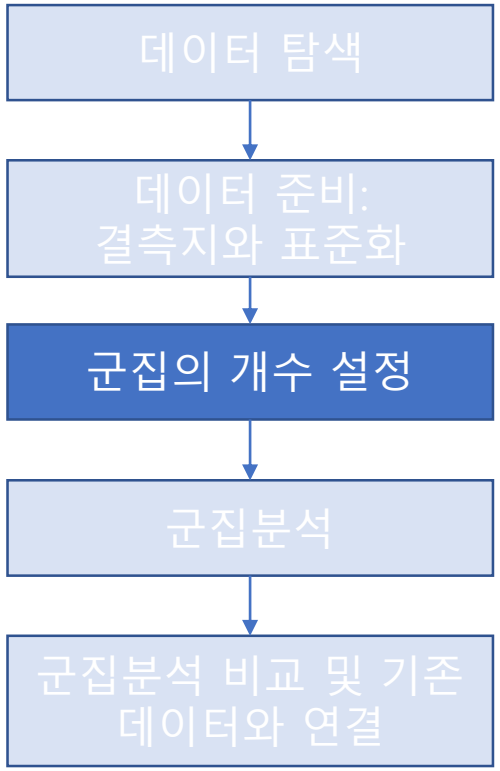

```
> nc2 <- NbClust(df3, # 군집화할 데이터
+                min.nc=2, # 최소군집의 수
+                max.nc=15, # 최대군집의 수
+                distance = "euclidean", # 거리계
+                method="ward.D") # 군집 방법론
*** : The Hubert index is a graphical method of
                In the plot of Hubert index, we
                significant increase of the val
                index second differences plot.

*** : The D index is a graphical method of dete
                In the plot of D index, we seek
                second differences plot) that c
                the measure.

*******************************************************
* Among all indices:
* 5 proposed 2 as the best number of clusters
* 9 proposed 3 as the best number of clusters
* 2 proposed 12 as the best number of clusters
* 1 proposed 13 as the best number of clusters
* 2 proposed 14 as the best number of clusters
* 3 proposed 15 as the best number of clusters

                ***** Conclusion *****

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



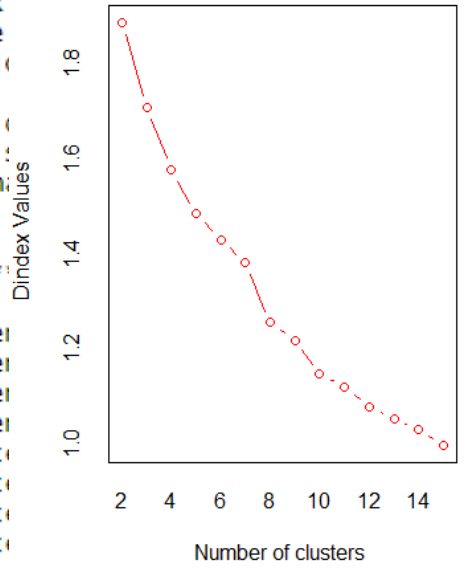



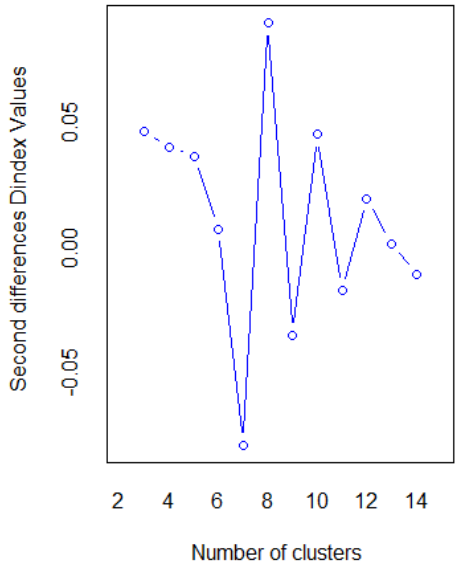
# 군집분석

- 실습: 시군구별 특성에 따른 군집분석




```
> nc3 <- NbClust(df3, # 군집화할 데이터
+                min.nc=2, # 최소군집의 수
+                max.nc=15,  # 최대군집의 수
+                distance =  "manhattan", # 거리계산
+                method="ward.D2") # 군집 방법론
*** : The Hubert index is a graphical method of determining the numb
                In the plot of Hubert index
                significant increase of the
                index second differences pl

*** : The D index is a graphical method of
                In the plot of D index, we
                second differences plot) th
                the measure.

*******************************************************************
* Among all indices:
* 7 proposed 2 as the best number of cluste
* 3 proposed 3 as the best number of cluste
* 4 proposed 4 as the best number of cluste
* 2 proposed 6 as the best number of cluste
* 2 proposed 10 as the best number of clust
* 1 proposed 12 as the best number of clust
* 1 proposed 14 as the best number of clust
* 3 proposed 15 as the best number of clust

                ***** Conclusion *****

* According to the majority rule, the best number of clusters is  2
```

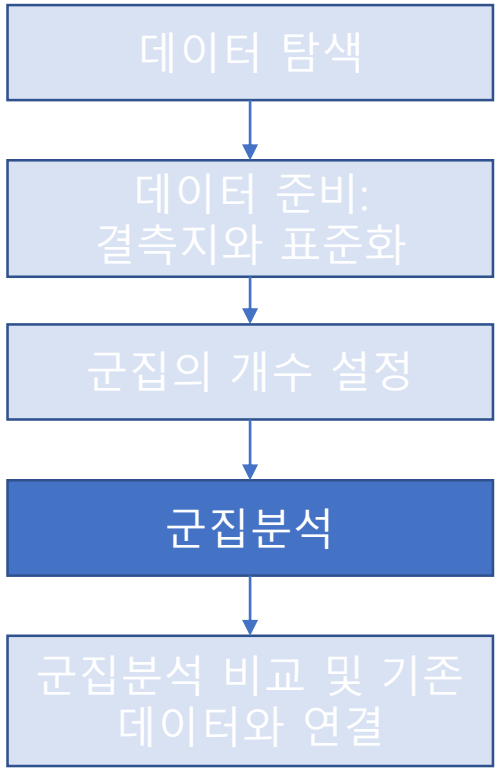


# 비지도 학습

## 군집분석

- 실습: 시군구별




```
> ### 4-1. K-Means Cluster Analysis
> fit <- kmeans(df3, 3) # 3 cluster solution
> aggregate(df3, # get cluster means
+           by = list(fit$cluster), # 군집별로
+           FUN = mean) # 특성변수들의 평균 산출
  Group.1       area pop_density    pop_tot pop_female age_average     r_aged pop_net_move    housing
1       1  0.6684298 -0.58342602 -0.6956719 -0.6959296   0.8649096  0.8960196    0.07110533 -0.7102093
2       2 -0.0479303  0.05633585  1.9445510  1.9256852  -1.0069485 -1.0037689    1.32283275  1.8196746
3       3 -0.6978054  0.60569704  0.2860694  0.2907257  -0.6833909 -0.7171250   -0.38273952  0.3304995
> # append cluster assignment
> df4 <- data.frame(df3, # 군집 결과를 기존 데이터와 합치기
+                   fit$cluster)
> summary(df4) # 요약통계
      area            pop_density         pop_tot           pop_female       age_average          r_aged
 Min.   :-1.14341   Min.   :-0.6264   Min.   :-0.9791   Min.   :-0.9809   Min.   :-1.8174   Min.   :-1.5
 1st Qu.:-1.00437   1st Qu.:-0.6140   1st Qu.:-0.7768   1st Qu.:-0.7746   1st Qu.:-0.7747   1st Qu.:-0.8
 Median : 0.01023   Median :-0.5514   Median :-0.3087   Median :-0.3014   Median :-0.2641   Median :-0.2
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0
 3rd Qu.: 0.66225   3rd Qu.: 0.2803   3rd Qu.: 0.5119   3rd Qu.: 0.5108   3rd Qu.: 0.8424   3rd Qu.: 0.8
 Max.   : 3.60889   Max.   : 3.8323   Max.   : 4.5051   Max.   : 4.4439   Max.   : 2.4383   Max.   : 2.3
    housing          fit.cluster
 Min.   :-1.0683   Min.   :1.000
 1st Qu.:-0.7679   1st Qu.:1.000
 Median :-0.3426   Median :2.000
 Mean   : 0.0000   Mean   :1.974
 3rd Qu.: 0.4898   3rd Qu.:3.000
 Max.   : 4.1947   Max.   :3.000
> ### 4-1. K-Means Cluster Analysis
> fit <- kmeans(df3, 3) # 3 cluster solution
> aggregate(df3, # get cluster means
+           by = list(fit$cluster), # 군집별로
+           FUN = mean) # 특성변수들의 평균 산출
  Group.1       area pop_density     pop_tot  pop_female age_average     r_aged pop_net_move    housing
1       1 -0.8911728   1.4620737  1.34487610  1.36408523  -0.6620074 -0.7668388   -0.75169820  1.0712459
2       2 -0.2717282  -0.1809863  0.07722019  0.06607357  -0.7557387 -0.7191723    0.37358748  0.2497745
3       3  0.6740655  -0.5574439 -0.72599659 -0.72564776   0.9853792  1.0047073    0.04112986 -0.7430168
> # append cluster assignment
> df4 <- data.frame(df3, # 군집 결과를 기존 데이터와 합치기
+                   fit$cluster)
> df4 <- df4 %>% rename(cluster.kmeans = fit.cluster)
> table(df4$cluster.kmeans) # 군집 빈도분포

 1  2  3
47 84 96
```

비지도 학습

# 군집분석

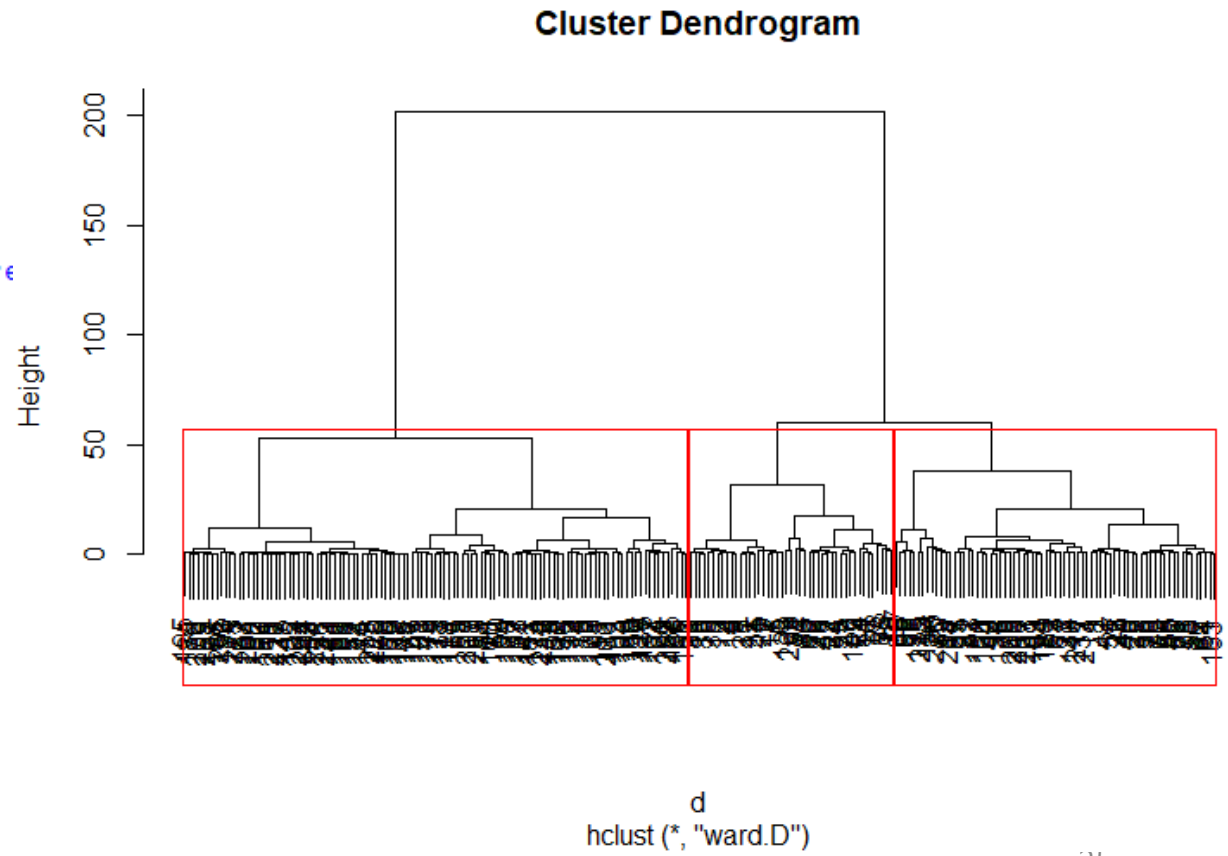
- 실습: 시군구별 특성에 따른 군집분석

```
> ### 4-2. Ward Hierarchical Clustering
> ?dist() # Distance Matrix Computation
> d <- dist(df3, method = "euclidean") # distance matrix 설정
> ?hclust() # Hierarchical Clustering
> fit2 <- hclust(d,  # distance matrix
+                method="ward.D") # 군집 방법
> par(mfrow = (c(1,1))) # 그래픽 환경 설정: 한 화면에 한 개의 그래프
> plot(fit2) # display dendogram
> groups <- cutree(fit2, k=3) # cut tree into 3 clusters
> rect.hclust(fit2, k=3, border="red") # # draw dendogram with re
```

군집의 개수 설정

군집분석

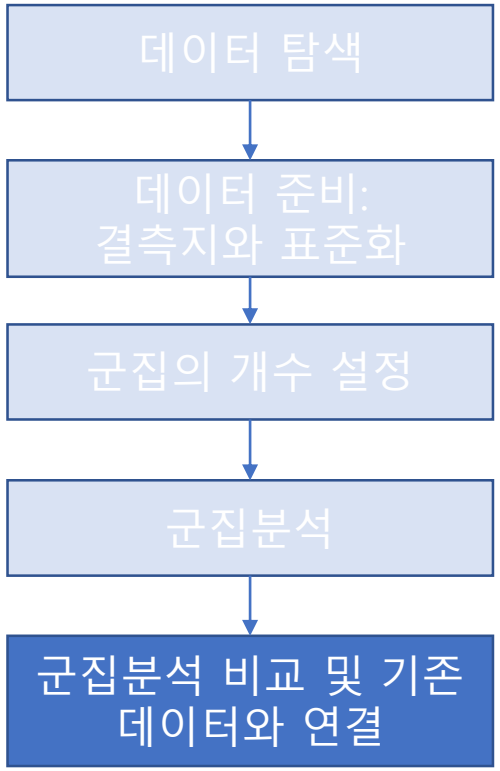
군집분석 비교 및 기존 데이터와 연결



비지도 학습

# 군집분석

- 실습: 시군구별 특성에 따른 군집분석




```
> ### 5. 군집 결과 비교 및 기존 데이터와 연결
> df4 <- data.frame(df4, # 군집 결과를 기존 데이터와 합치기
+                   fit$cluster)
> table(df4$fit.cluster) # 계층적 군집 결과

 1  2  3
47 84 96
> table(df4$cluster.kmeans, # 이원빈도분포표: kmeans 군집결과
+       df4$fit.cluster) # 계층적 군집 결과

    1  2  3
  1 47  0  0
  2  0 84  0
  3  0  0 96
> summary(df4)
      area            pop_density          pop_tot          pop_female       age_average          r_aged
 Min.   :-1.14341   Min.   :-0.6264   Min.   :-0.9791   Min.   :-0.9809   Min.   :-1.8174   Min.   :-1.5963
 1st Qu.:-1.00437   1st Qu.:-0.6140   1st Qu.:-0.7768   1st Qu.:-0.7746   1st Qu.:-0.7747   1st Qu.:-0.8395
 Median : 0.01023   Median :-0.5514   Median :-0.3087   Median :-0.3014   Median :-0.2641   Median :-0.2762
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.66225   3rd Qu.: 0.2803   3rd Qu.: 0.5119   3rd Qu.: 0.5108   3rd Qu.: 0.8424   3rd Qu.: 0.8035
 Max.   : 3.60889   Max.   : 3.8323   Max.   : 4.5051   Max.   : 4.4439   Max.   : 2.4383   Max.   : 2.3654
    housing         cluster.kmeans   fit.cluster
 Min.   :-1.0683   Min.   :1.000   Min.   :1.000
 1st Qu.:-0.7679   1st Qu.:2.000   1st Qu.:2.000
 Median :-0.3426   Median :2.000   Median :2.000
 Mean   : 0.0000   Mean   :2.216   Mean   :2.216
 3rd Qu.: 0.4898   3rd Qu.:3.000   3rd Qu.:3.000
 Max.   : 4.1947   Max.   :3.000   Max.   :3.000
```

# 연습문제 03

- 시군구별 데이터인 "data_by_sigungu_2018.xlsx"를 불러들여, 다음과 같은 변수들만 추출하여 군집분석을 실시하고자 한다.

pop_density, # 인구밀도
r_aged, # 65세 이상 인구 비율
pop_net_move, # 순인구 이동수
apt_sale_price, # 아파트 거래가격지수
accident_per_1000car) # 1000대 차량당 사고건수

- 최적의 군집의 갯수를 몇 개로 하는 것이 가장 바람직한 지 논의하시오.

# 요약

- 분류와 머신러닝
  - 머신러닝: 지도학습과 비지도 학습
  - 통계모델과 머신러닝의 차이
  - 분류 알고리즘의 종류
  - 분류 모델의 평가
  - 교차타당성(cross validation)
- 분류와 예측
  - 로지스틱 회귀모델
    - 이항 로지스틱 모델
    - 일반화 가법 모델(gam)
  - 판별분석
    - 선형 판별모델
    - 비선형 판별모델
- 분류와 기계학습
  - K-근접 이웃(KNN) 분류
  - 결정트리 학습법
  - 앙상블 학습
    - 배깅과 랜덤 포레스트
    - 부스팅과 XGBoost
- 비지도학습
  - 비지도학습의 개념
  - 주성분 분석
    - 주성분 분석
  - 군집(클러스터링) 분석
    - K-평균 클러스터링
    - 계층적 클러스터링

# 끝

- 질의와 토의(Question & Discussion)
  - 이번 강의 내용을 시청하고, 실행하면서 궁금한 점이나 어려운 점에 대하여 토의해봅시다.

- 다음 차 강의주제
  - **비정형 데이터 마이닝: 텍스트**