데이터 마이닝

* 의미 있는 패턴이나 법칙을 찾기 위해 다량의 데이터를 탐구하거나 분석하는 과정
* 대량의 데이터로부터 그 안에 숨어있는 새롭고, 가치 있고 의사결정에 유용한 정보를 찾는 작업
* 소비자를 더 잘 이해함으로써 판매량과 이윤 그리고 회사의 가치와 경쟁력을 높이는 것
* 새로운 서비스를 창출
* 데이터로부터 비즈니스에 관련 있는 중요한 정보/함의를 유도하는 과정

데이터 분석과정

* 데이터 소스 (내/외부 데이터, 미디어)
* 수집 (수동, 자동 크롤링)
* 저장 (정형/비정형 하드디스크 저장)
* 처리 (일괄 처리, 실시간&배치, 분산 병렬 처리)
* 분석 (전처리)
* 표현

독립변수

* 예측 모델의 입력변수

종속변수

* 출력변수, 목표변수, 타깃변수

예측/분류

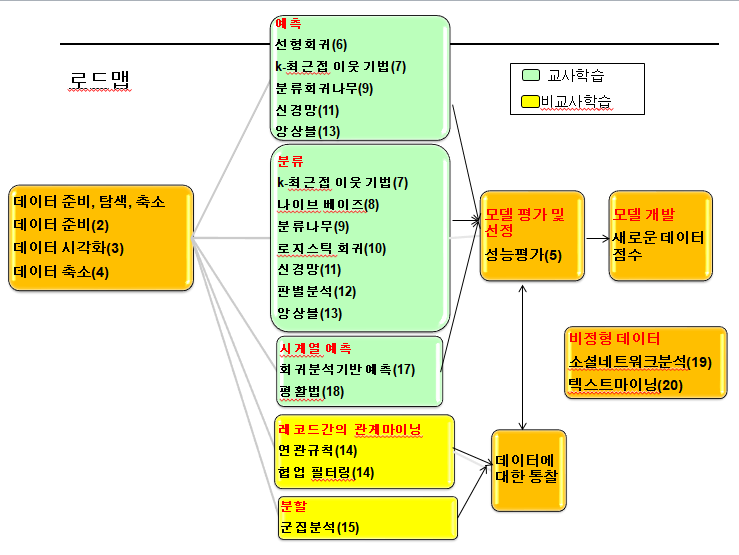
* 목표변수의 추정된 값

데이터 분할

* 훈련데이터
* 검증데이터
* 평가데이터

점수

* 개발된 모델을 이용하여 새로운 데이터의 결과를 예측한 값 혹은 분류된 계층



예측

* 독립변수 또는 예측변수로 종속(목표)변수를 예측
* 예측변수(X)는 연속적이거나 또는 범주적일 수 있음
* **선형 회귀분석, k-근접 이웃기법, 신경망**
* 제품의 품질은? / 다음 주 주가 = f(최근 가격, 경제지표)

분류

* 범주형 종속변수(목표변수, Y)를 가진 회귀분석
* 학습 데이터, 평가 데이터, 일반화
* **의사결정나무, CBR, 신경망**
* 대출 승인 = f(개인정보) / 증권가격 이동 방향 = f(최근 가격, 경제지표)

의사결정나무

* 포털 사용자 중에서 서비스 사용자 예측

장바구니 분석(연관 분석)

* 어떤 항목들이 함께 구매되는가?

협력적 필터링

* 사용자 기반: 패턴이 유사한 고객들이 선호한 상품(아이템)을 추천하는 시스템
* 항목(아이템) 기반: 아이템 간의 유사도를 고려하여 추천하는 방법 (아마존 쇼핑몰, 논문)

군집화

* 비슷한 대상들의 무리
* 비교사, 탐색적 지식 발견
* C-means, SOM
* 타깃 마케팅을 위한 시장 세분화 / 각 각 다른 체형들을 위한 유니폼 사이즈 수 감소

Social Network Analysis (사회 연결망 네트워크 분석)

* 그래프 레이아웃, 새로운 연결 추천 (페이스 북, 링크드인 등), 개체해석
* 중요한 페이지나 노드 탐색 (pageRank 알고리즘을 이용하여 노드의 중요도 계산)

데이터마이닝

* 기법: 통계학 + 기계학습(인공지능) + 데이터베이스

실제 성공하기 위한 핵심 요소

* 문제 이해, 기법 이해

R의 다양한 기능

* 구문보다는 데이터 기반 문제를 해결의 절차 위주로 생각하고 그에 맞는 라이브러리를 활용
* 쉬운 편집/개발 환경
* 데이터의 쉬운 조작
* 풍부한 실습용 데이터 세트
* 주변에 산재된 사회와 자연현상의 데이터
* 풍부한 라이브러리

패키지와 라이브러리의 차이

* 패키지는 R함수들을 모아 놓은 컬렉션이며, 라이브러리는 R패키지가 저장되는 폴더를 의미하며, R세션으로 불러오는 함수가 library() 이다.

데이터마이닝에서의 핵심 아이디어

* 분류(Classification)
* 예측(Prediction)
* 연관규칙 및 추천시스템(Association Rules, Recommender)
* 데이터 축소(Data Reduction) – 레코드 감소
* 차원 축소(Dimension Reduction) – 예측변수 감소
* 데이터 검색
* 시각화

교사학습 (목표변수가 존재함)

* 하나의 목표 또는 결과 변수를 예측
* 독립(입력) 변수와의 관계를 통해 모델링
* 새로운 입력데이터와 목표변수 값 분류 혹은 예측 (각 행은 케이스, 각 열은 변수)

분류: 범주형(categorical) 목표(결과) 변수 예측 (구매/비구매, 사기/정상, 신용OK/신용NO)

예측: 수치형(numerical) 목표(결과) 변수 예측 (판매량, 수익, 실적)

**분류와 예측을 결합하여 예측적 분석(predictive analytics)**

* 학습데이터 (Training data): 목표 변수가 취하는 것이 알려짐/모델구축에 사용되는 데이터
* 검증데이터 (Validation data): 목표 변수가 취하는 값이 알려짐/모델의 성능을 평가/비교

비교사 학습(목표변수가 존재하지 않음)

* 데이터를 의미 있는 부분들로 분할 (데이터 내에 있는 패턴 발견)
* 예측하거나 분류할 목표(결과)변수가 없음
* 연관규칙, 데이터 축소와 탐색, 시각화, 협력적 필터링을 통한 추천

연관 규칙: 구매 항목 간의 연관성을 분석하고 탐색 (X가 구매되면, Y 또한 구매된다.)

* 각 행은 매매거래, 추천 시스템에 사용됨. **유사성 분석(affinity analysis) 이라고도 함.**



Milk 🡪 Coke인 경우

신뢰도: 3/4 = 75% (장바구니에서 Milk를 산 사람 중, Coke까지 구입을 한 경우)

지지도: 3/5 = 60% (장바구니에 얼마나 나와있는지)

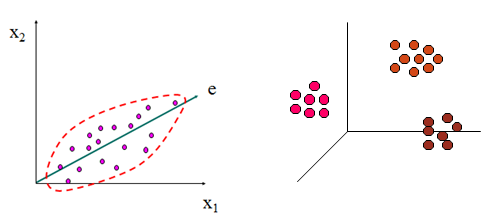
데이터 축소: 복잡한/다량의 데이터를 보다 간단한/적은 데이터로 증류

* <왼쪽 그림> 변수/열의 수를 축소 (주성분 분석)

x1, x2의 2차원 정보를 e 한 축을 기준 1차원으로 볼 때, 두 경우는 결과는 동일

* <오른쪽 그림> 레코드/행의 수를 축소 (군집화)

그룹화하여 대표 값(평균 값)을 정한다. 빨, 주, 보 – 7, 9, 7 🡪 1, 1, 1 이런 식으로

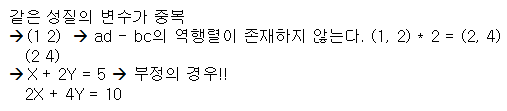


데이터 탐색: 데이터의 전반적 패턴과 특이 패턴 탐색 (차원축소, 시각화 기법 사용)

* 데이터 세트는 대게 대용량, 복잡함
* 문제 해결을 위해 데이터를 검토가 필요

같은 성질의 변수가 중복 / 이상치 탐지 / 데이터 요약

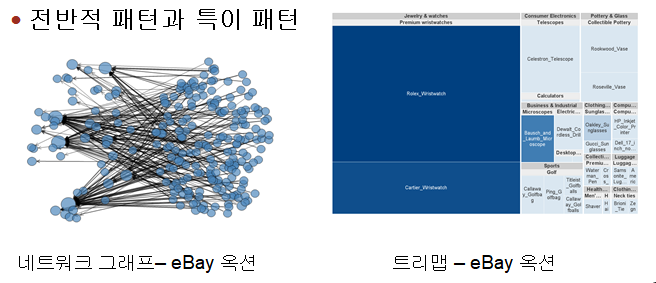
같은 성질의 변수가 중복



이상치 탐지: 정상적인 데이터로 보기에는 어려운 경우

데이터 시각화: 데이터의 그래프와 plot, 히스토그램, boxplots, bar 차트, 산점도

* 변수 쌍 사이의 관계를 조사하는데 특히 유용
* 수치형인 경우 그 변수의 분포를 파악하고 이상치 탐색 등



트리맵: 각 사각형 크기가 의미하는 것은 ‘제품의 가격’ / 색이 진할수록 불평이 많음.

협업 필터링 (Collaborative Filtering)

* 개개인의 과거 구매정보와 다른 사람의 구매 정보 이용

개개인의 구매 성향을 예측하는 추천시스템 / 맞춤형 추천이 필요한 분야에 사용

* 나와 비슷한 행동을 한 사람들을 통해 추천해 주는 시스템 (사용자 기반 추천시스템)
* 비슷한 아이템을 통해 추천해 주는 시스템 (아이템 기반 추천시스템)

데이터마이닝 관계

* 데이터마이닝 목적을 설정 (정의/이해)
* 데이터 획득
* 데이터의 탐색, 정제, 전처리

적절한 수치인가? 결측치는? 이상치는?

* 데이터 축소

불필요한 변수 제거, 새로운 변수 생성, dummy variable 생성

* 데이터마이닝 문제로 결정 (분류, 예측, 군집 등)
* 데이터 분할 (지도학습인 경우) : 훈련데이터, 검증데이터, 평가데이터
* 기법 선택 (회귀 분석, 인공신경망, 분류와 회귀나무 등)
* 실행 반복과 모델 개선
* 결과 평가 – 모델 비교
* 가장 좋은 모델 적용

데이터분석 사전 단계

* 데이터 획득: 샘플링

데이터마이닝은 대량의 데이터베이스를 취급

알고리즘과 모델은 DB에서 샘플링하여 일부에 적용

최종 모델 선택 및 개발 (더 큰 DB에 적용하여 목표변수 값 계산)

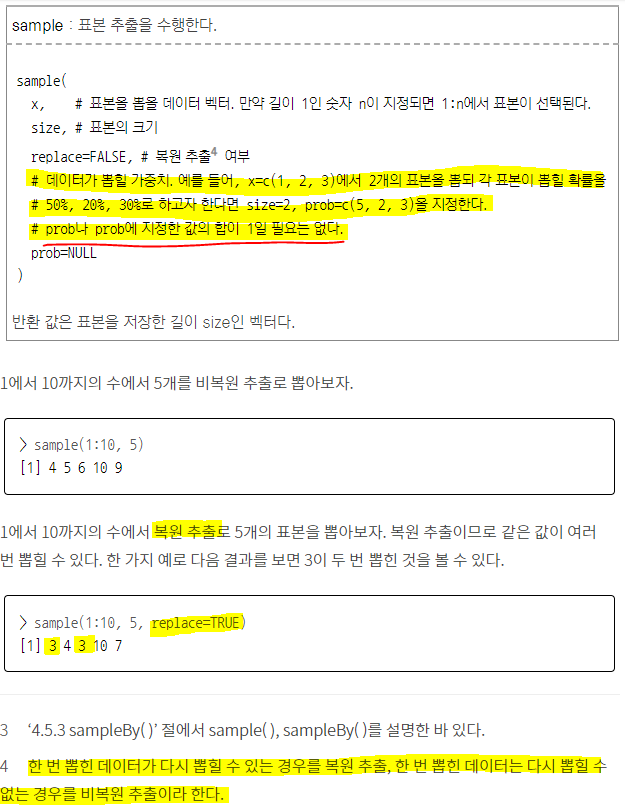
* 희소 사건에 대한 오버샘플링

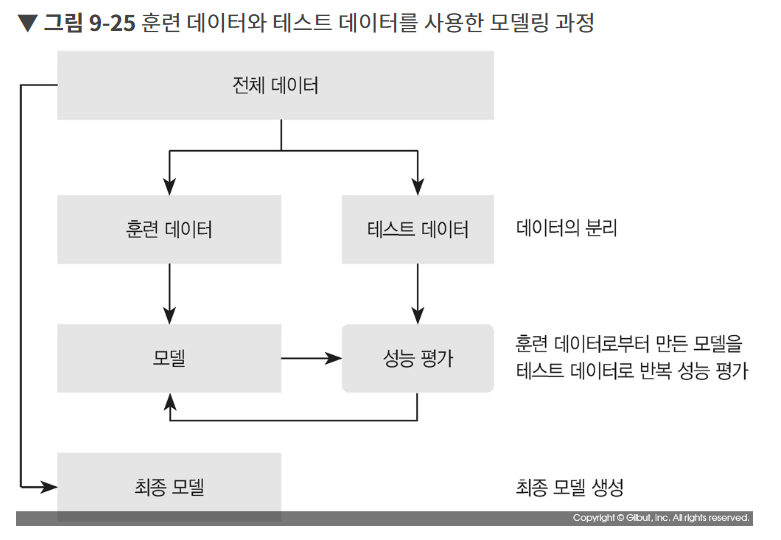
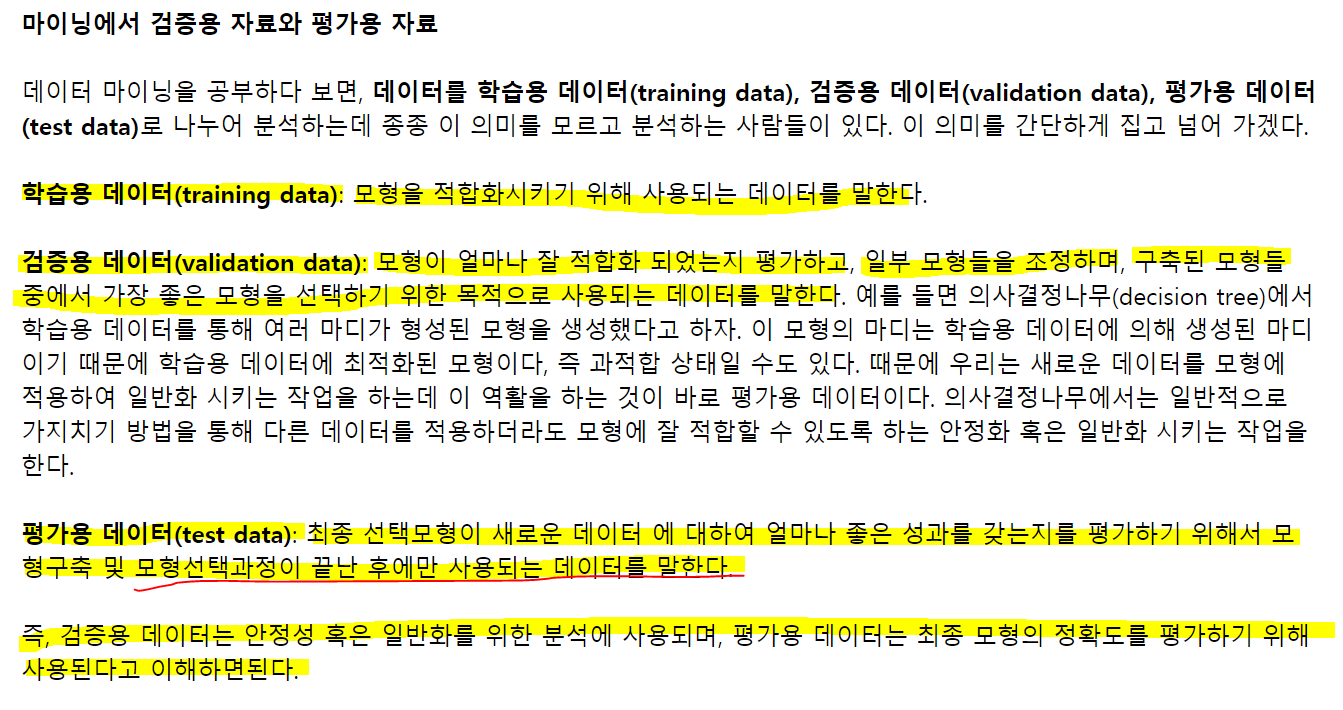
때때로 관심 있는 사건이 희소한 경우가 존재 (우편물에 대한 응답, 세금 사기 등)

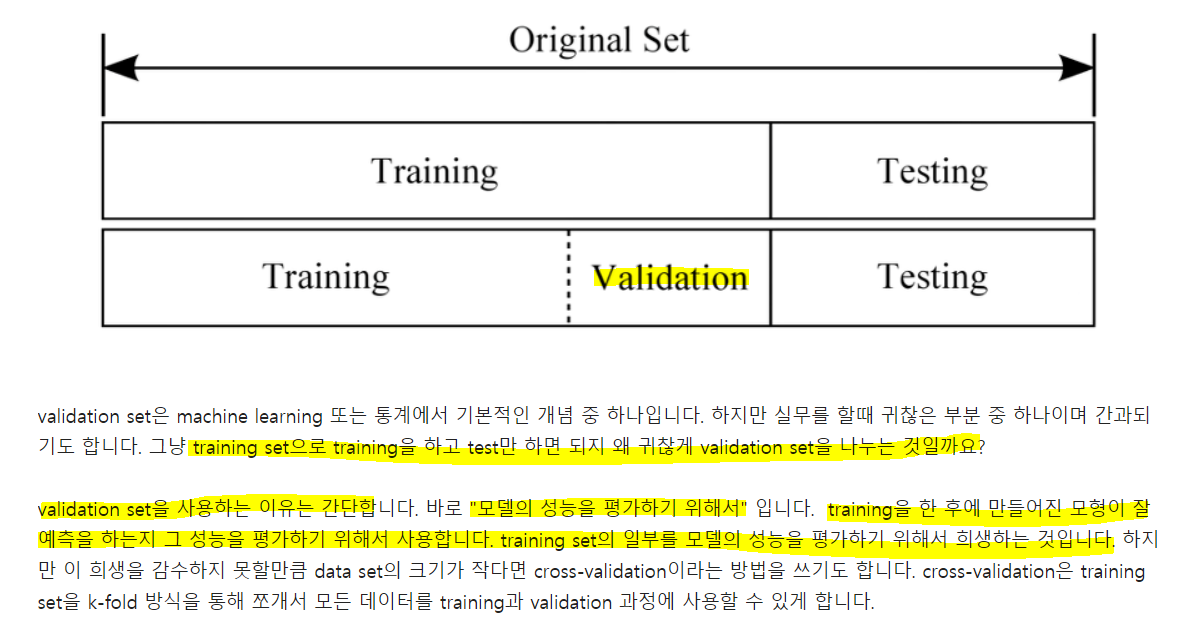
이 경우 샘플링은 “관심 있는” 케이스를 너무 적게 샘플링함 (단점)

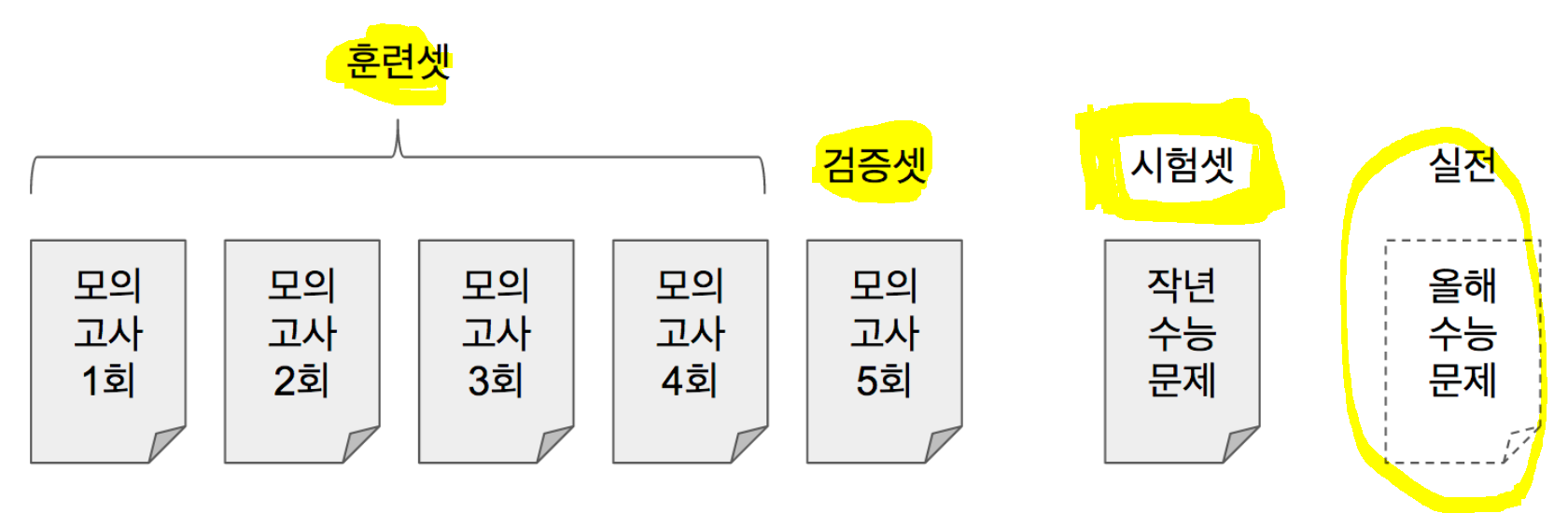
일반적인 해법: 좀 더 균형 잡힌 학습 데이터를 얻도록 희소 케이스를 오버샘플링

오버샘플링 결과 조정 필요









Ex) <https://tykimos.github.io/2017/03/25/Dataset_and_Fit_Talk/>

변수의 종류

* 범주형(Categorical)



나이브 베이즈는 범주형변수를 그래도 사용.

대부분의 다른 알고리즘에서는 **이진 가변수**를 만들어야 함 (가변수의 수 = 범주의 수 - 1)

순위가 있는 경우 (낮음, 중간, 높음): ordinal

순위가 없는 경우 (남성, 여성): nominal

* 수치형(Numeric)

대부분의 알고리즘은 수치 데이터를 처리할 수 있음.

때때로 범주형 변수로 변환할 경우가 있음.

연속형(Continuous)

정수형(Integer)

이상치 발견(Detecting Outliers)

* 이상치는 **극단적인**, 대부분의 데이터에서 멀리 떨어진 관측치
* 모델에 불균형한 영향을 줄 수 있다.
* 데이터 전처리에서 중요한 단계
* 일단 발견되면, 그것이 오류인지 아니면 실제로 극단적인지 결정 (실무지식 요구)
* 이상치는 찾는 것이 데이터 마이닝 실행의 목적. Anomaly detection
* 신용카드 불법/사용
* Airport security screening
* 비정상적인 물품이 있는 것, 네트워크 침입을 찾는 것, 스팸메일을 찾는 것

결측 데이터(Missing Data)처리

* 대부분의 알고리즘은 결측치가 있는 레코드를 처리하지 않음.
* Default로 이러한 레코드는 삭제함.
* 해법1: 삭제

적은 수의 레코드가 결측치를 갖는다면, 삭제 OK

많은 레코드가 작은 변수 집합에서 결측치라면, 이 변수들을 뺼 수 있음.

많은 레코드가 결측치를 갖는다면, 삭제는 알맞지 않다.

* 해법2: 대체

결측치를 타당한 예측치로 대체

평균(열 변수의 평균)

중앙값(열 변수의 중앙 값)

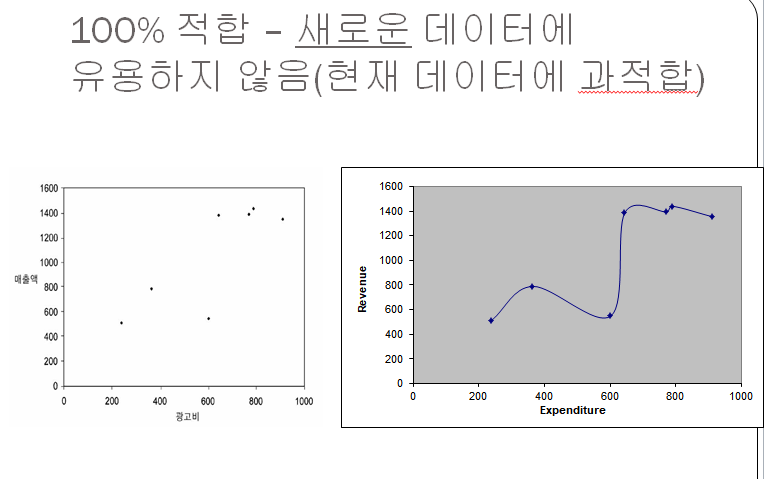
회귀식(예측변수를 통해 예측을 하는 방법)

데이터의 표준화/정규화

* 변수가 큰 값을 가지면서 예측 결과에 주된 영향을 미치고 왜곡할 경우에 사용
* 평균을 빼고 표준편차로 나눔: Z-SCORE
* 최소값을 빼고 범위로 나눔으로써 0-1 척도로 사용

과적합(Overfitting)의 문제

* 통계적 모델은 변수들 사이의 관계에 대해 고도로 복잡한 모델로 설명할 수 있음
* 복잡한 모델일수록 주어진 데이터에 잘 적합
* BUT) 새로운 데이터가 사용될 때, 고도로 복잡한 모델은 잘 맞지 않을 수 있음.
* EX) 광고기 지출이 매출에 끼치는 영향



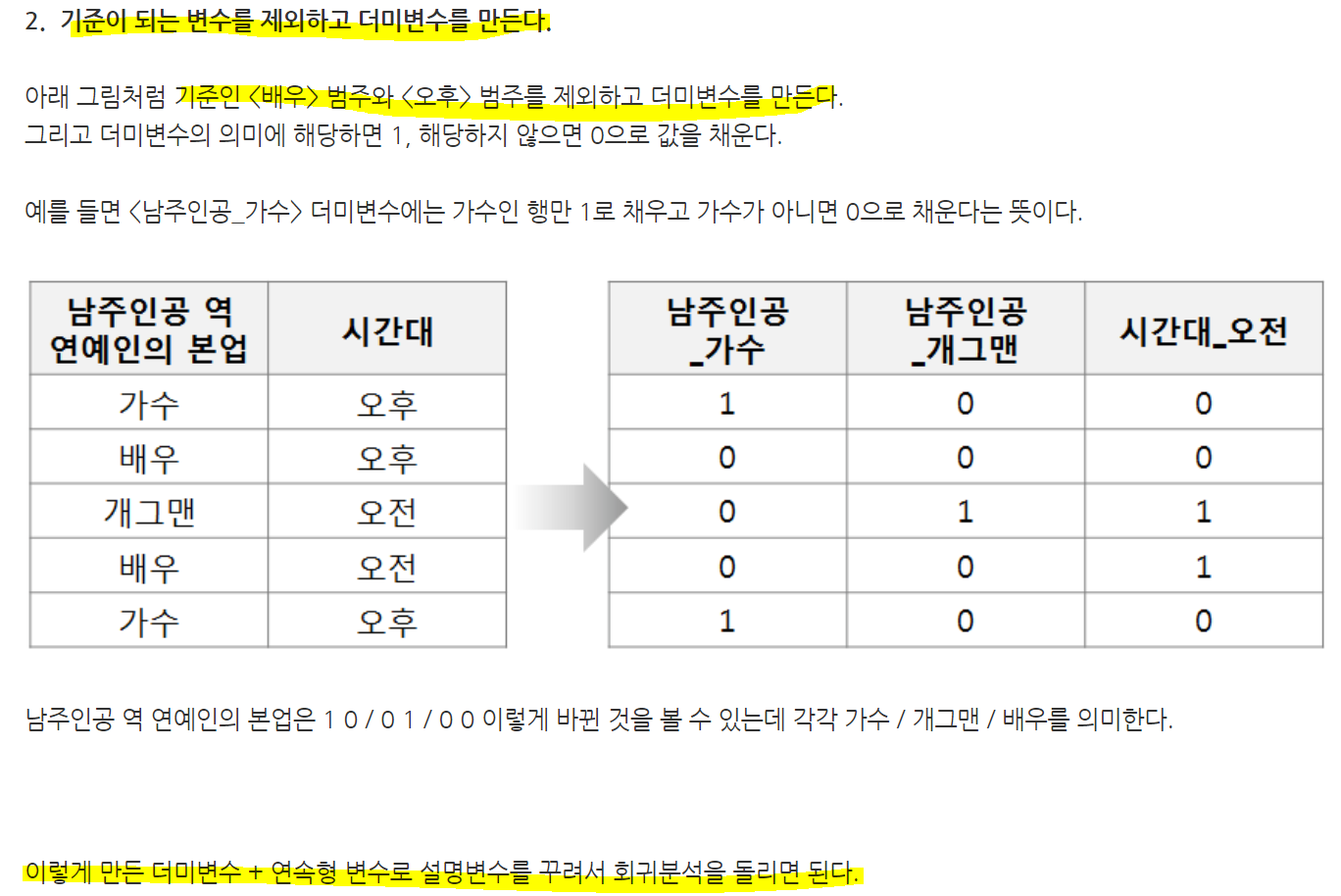
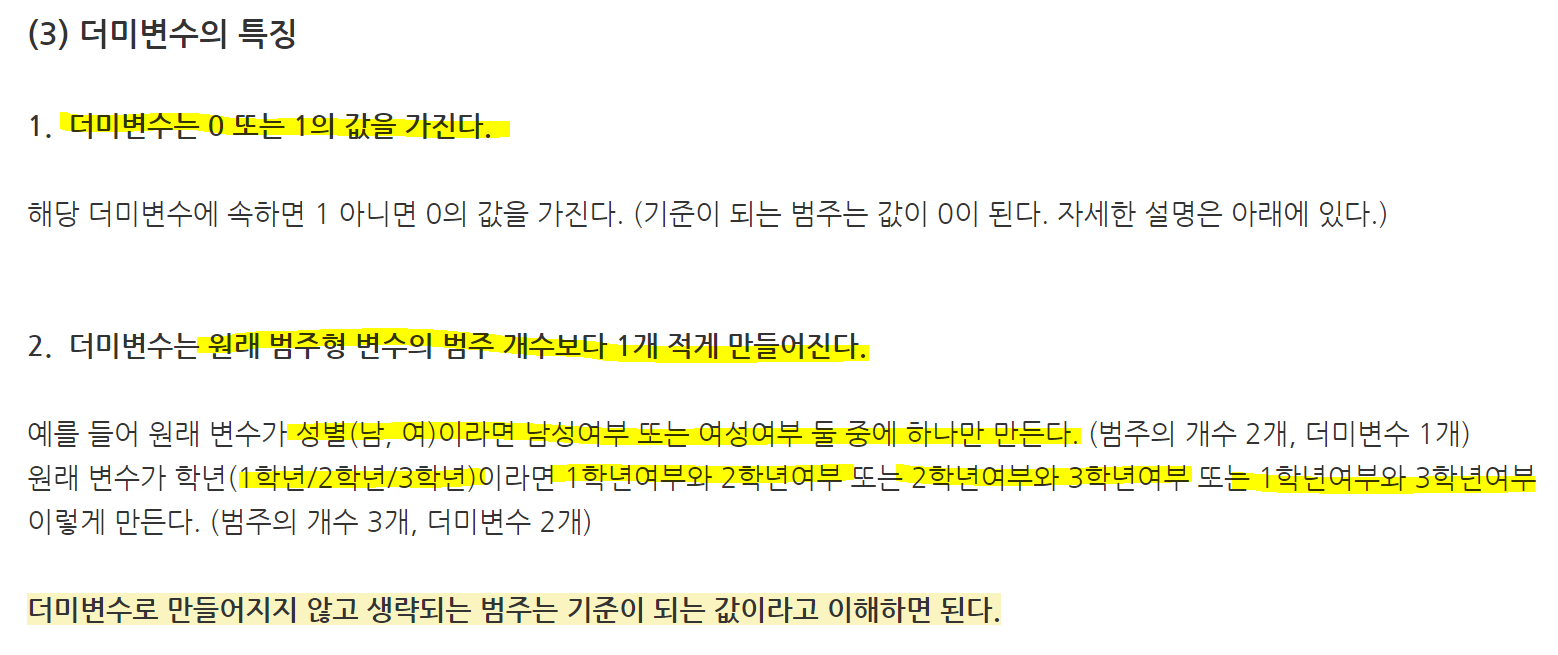
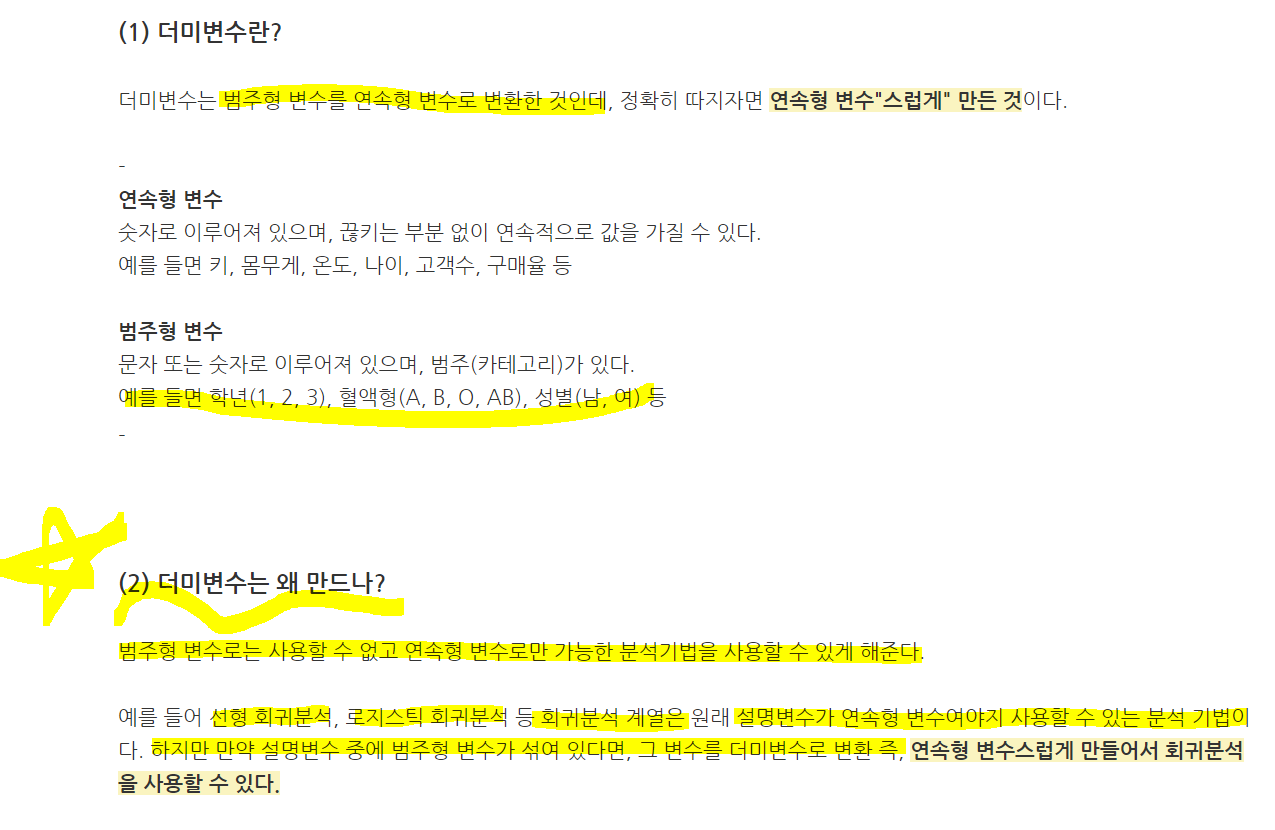
* 7개의 점 – 6차 방정식
* 예측변수는 ‘광고비’ 하나.
* 하나를 위해 파라미터 수(6개)가 너무 많다. – 과적합

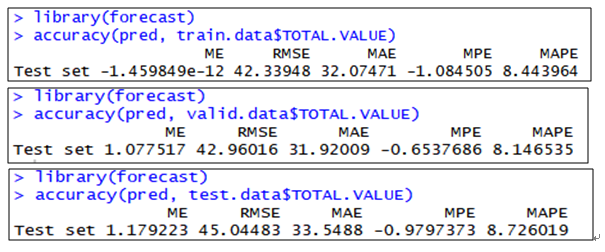
과적합 원인/결과

* 너무 많은 예측변수들
* 너무 많은 파라미터들을 가진 모델
* 많은 다양한 모델들을 시도
* 완전히 새로운 데이터에 사용된 모델은 기대만큼 잘 작동하지 않음.
* 현재의 결과를 위해 많은 예측변수를 투입시키지 말고 데이터마이닝의 목적은 미래를 예측할 수 있어야 한다. 많은 예측변수를 투입시킴으로써 ‘과적합’이 발생한다.

**데이터 분할**

* 학습데이터: 모델 구축 시 사용되는 데이터
* 검증데이터: 학습데이터로부터 구축된 여러 모델의 성능을 비교할 때 사용
* 평가데이터: 구축된 모델이 미래에 수집될 새로운 데이터데 대해 얼마만큼 예측/분류 성능을 보일지 평가할 때 쓰이는 데이터
* 모델이 학습 데이터(Training Data)에서 개발될 때, 학습데이터를 과적합 할 수 있다. 따라서 Validation 에서 조정이 필요하다.
* 하지만, 검증 데이터(Validation Data)에서 다량의 모델을 조정한다면 검증 데이터(Validation Data) 또한 과적합 할 수 있다. (물론 검증데이터를 이용해서 오류를 최소화 할 수 있는 K값을 결정한다.) 검증데이터를 이용해서 모델을 결정해야 할 경우에는 평가데이터 (Test Data)가 필요하다.
* 따라서, 최종 선택된 모델에 평가 분할을 적용함으로써 새로운 데이터에 적용했을 때 오차가 작은 추정치를 제공한다.
* **모델 성능 평가에 사용된 검증데이터가 우연히 모델에 잘 맞아 우수한 성능을 보였을 수도 있기 때문에 검증데이터로부터 선택한 모델은 때때로 일반화 관점에서 과대평가 될 수 있다. 즉 미래의 새로운 데이터에도 꼭 좋은 성능을 내리라는 보장이 없다. 따라서 모델 평가에 있어 가장 객관적인 방법은 모델구축에 전혀 사용되지 않는 평가데이터를 통한 검증이라고 할 수 있다. (미래 값에 대한 정확한 예측이 목적이 아닌 경우에는 학습데이터와 검증데이터로도 충분하다.)**





ME : 평균오차 (잔차들의 단순 평균값)

* 실제값과 예측값의 차이가 양수, 음수가 될 수 있기 때문에 거의 0에 가깝다.
* 이론적으로는 잔차의 합(혹은 평균)은 0이다.
* 따라서 단순 평균값으로는 오차가 얼마나 큰지 알 수 없다.

RMSE : 제곱근 평균 제곱근 오차

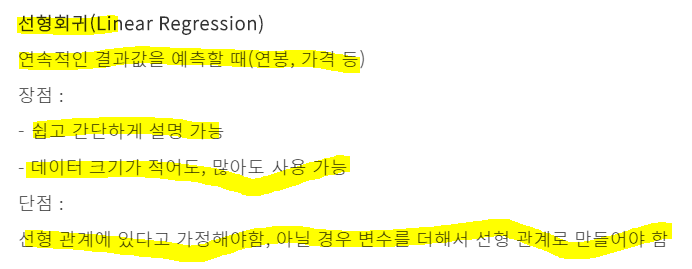
* 평균 제곱 오차 값에 제곱근을 취함

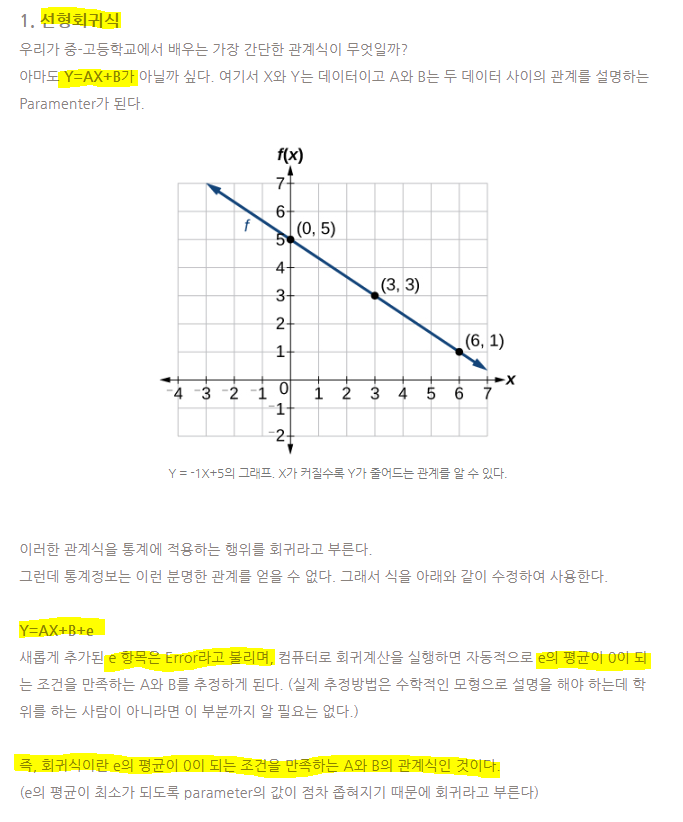
**오차 메트릭은 실제 값에서 예측 값의 평균 편차로 회귀 모델의 예측 성능을 측정한다. 오차 값이 낮을수록 모델의 예측이 더 정확함을 의미한다. 전체 오차 메트릭이 0이라는 것은 모델이 데이터에 완벽하게 적합하다는 의미이다.**

**즉 훈련데이터에서 예측값의 정확도가, 오차 값이 가장 작으므로 모델예측이 검증데이터와 평가데이터보다 더 정확함을 알 수 있다.**

선형회귀분석 장점 (Linear Regression)

* 변수선택방법의 하나인 최상 부분집합(Best subset)기법을 써서 예측에 중요한 변수만을 선택할 수 있다. 이와 같은 방법으로 선정된 가장 좋은 모델을 새로운 데이터에 적용해본다.





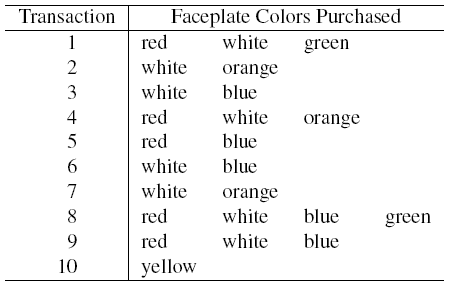
**연관규칙**이란

* 어떤 항목이 어떤 항목을 동반하는가에 대한 규칙 생성
* **거래 기반**(transaction-based), **사건 기반**(event-based)
* **장바구니 분석**(market basket analysis), **연관성 분석**(affinity analysis)
* 고객의 구매 항목들(items) 사이의 연관성을 찾도록 **고객 거래 데이터베이스를 학습하던 것**으로부터 유래
* 많은 추천 시스템에서 사용됨(Amazon)
* IF-THEN 구문 형식 (IF: 조건부(antecedent) / THEN: 결론부(consequent))
* Item sets: 조건부와 결론부를 구성하는 항목들의 집합

항목들 사이의 모든 가능한 규칙들을 조사

(가장 의존적인 관계를 보일 것 같은 규칙을 선정)

* **조건부와 결론부는 disjoint 하다. 즉 공통항목이 존재하지 않는다.**



* Transaction data (0~5 선호도 데이터 / 0 or 1 구매? 비구매?)
* **예를 들어 Transaction 1에 대해 다음과 같은 규칙이 가능하다.**
* If 빨간색, then 하얀색 (만약 빨간 케이스를 사면, 하얀 것도 산다.)
* If 하얀색, then 빨간색
* If 빨간색과 하얀색, then 초록색
* 그 외 다수..

연관성규칙의 첫 번째 단계

* 항목들 간의 연관성을 나타내기 위한 **후보가 되는 모든 규칙들을 생성**
* **항목이 늘어나면**, 이러한 모든 조합들을 생성하기 위한 **계산시간은 기하급수적으로 증가**
* 해결책은 DB내에서 **빈도수(frequent)가 높은 조합만을 고려** (**이를 빈발 아이템세트 [frequent item sets] 이라고 부른다.**)

빈발 항목집합 (Frequent Item Sets)

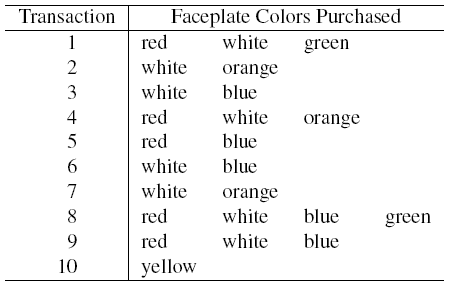
* 발생하는 빈도 수가 높은 항목 조합
* 단일 빈발항목, 2개 빈발항목, 3개 빈발항목 …

단일 빈발항목을 토대로 2개 3개 .. 빈발항목을 만든다.

* 빈발 아이템세트 기준: 지지도(Support)

**조건부와 결론부 모두를 포함하는** 거래(transaction)의 개수 / 전체거래 중 퍼센트

**P(조건부 항목집합 AND 결론부 항목집합)**



* If 빨간색, then 하얀색
* 지지도는 전체 거래 중 {빨간색, 하얀색} 항목집합을 만족하는 거래의 수
* 4/10 \* 100 = 40% or 4개

Apriori 알고리즘 (k개 제품에 대해)

* 빈발 아이템 세트 생성하는 알고리즘
* **사용자는 최소 지지도 기준 지정**
* **지지도 기준을 만족시키는 단일-아이템세트의 리스트 생성 (빈발 단일 아이템세트)**

**지지도 기준을 충족하는 빈발 단일 아이템세트 리스트를 이용하여 2개 아이템세트의 리스트를 생성 (지지도 기준을 충족하는 빈발 2개 아이템세트 생성)**

빈발 2개 아이템세트 리스트 사용하여 빈발 3개 아이템세트 생성..

**K – 아이템세트까지 재귀적으로 계속 진행**

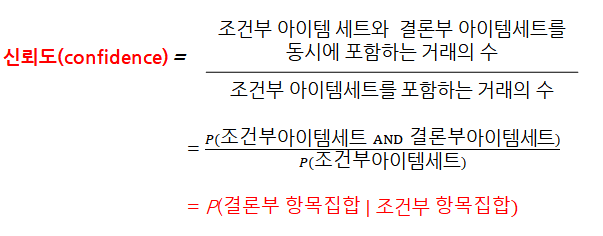
* **즉 연관규칙은 빈발아이템 세트를 이용하여 생성하는 것이다.**

연관규칙 평가 측정

* 연관성의 강도를 측정하기 위한 측정도구

신뢰도(Confidence)

신뢰도가 높을수록 강한 연관규칙



* 슈퍼마켓에 100,000개 거래 기록

오렌지 주스와 감기약이 함께 [조건부] 있는 거래 2,000개

이 중 수프 [결론부] 가 포함된 거래 800개

(연관 규칙 가정) [조건부] If 오렌지 주스, [결론부] 감기약 then 수프

**지지도: 800/100,000 = 0.8% / 신뢰도: 800/2,000 = 40%**

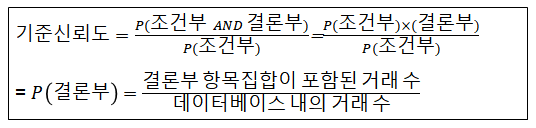
향상도(Lift Ratio) = 신뢰도 / 기준 신뢰도

향상도가 높을수록 강한 연관규칙

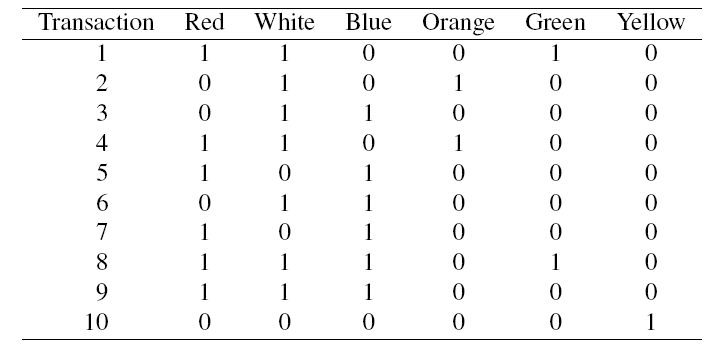
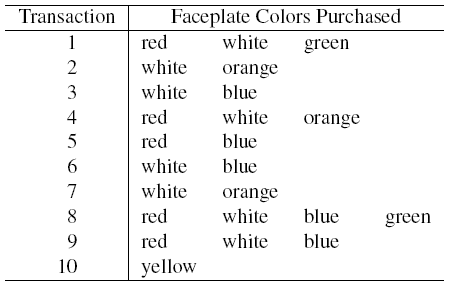
향상도가 1보다 크면, 현재 규칙이 결론부 항목집합을 찾는데 유용한 규칙임을 나타냄.

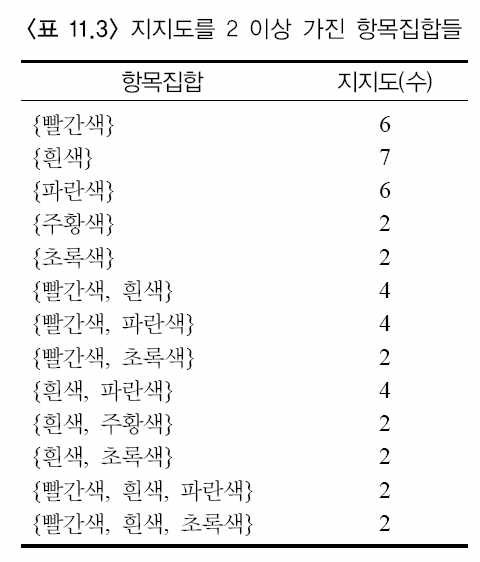
[기준 신뢰도]

거래 내의 결론부 항목집합이 각 규칙의 조건부와 독립적이라고 가정하여 얻어진 것.



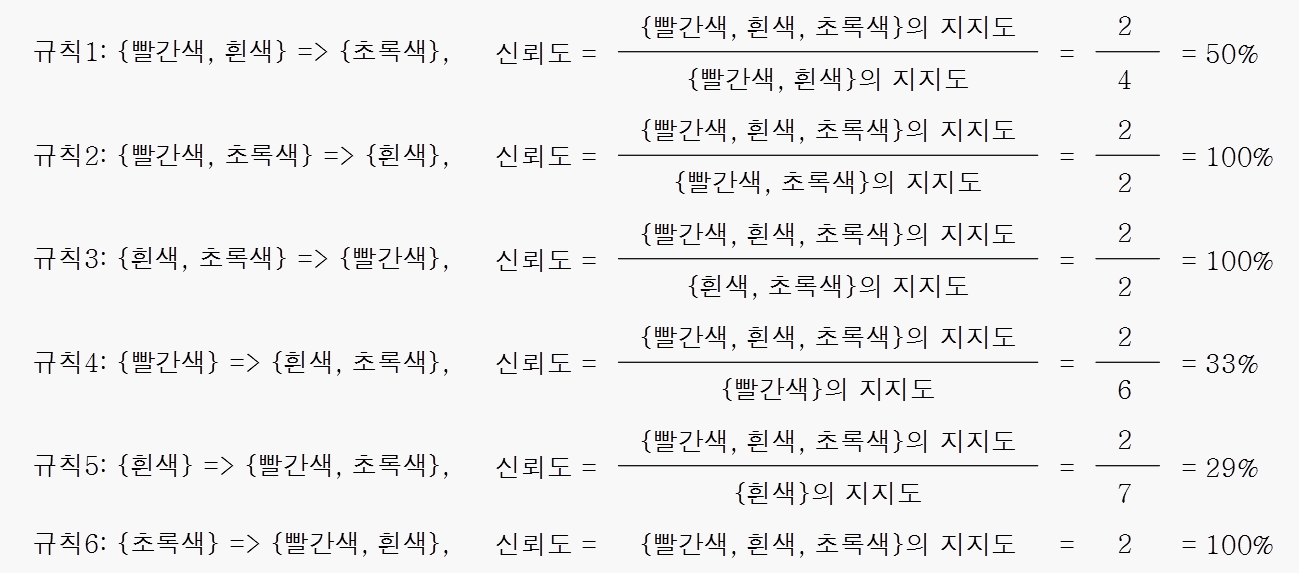
If A, then B 일 때 B가 누구나 다 하는 상황이라면 **향상도를 계산해야 한다.**



🡪 transaction data 🡪 binary matrix (이진행렬)

🡪 이진행렬을 보고 판단 가능

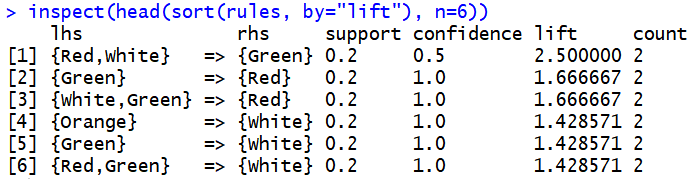
🡪 이러한 빈발 아이템세트로부터 특정한 신뢰도를 충족하는 연관 규칙 생성 가능.



* {빨간색, 흰색, 초록색} 연관규칙에 대해 알고자 할 때, 만약 신뢰도 기준이 70%라면 최소 신뢰도 70% 이상인 규칙2, 규칙3, 규칙6만 선택된다.

연관규칙

* arules in R package 사용.



* [4] 만일 오렌지색이 구매되면 100% 신뢰도로 흰색도 구매된다. 이 규칙의 향상비는 1.43이다.

**지지도가 높을수록 굉장히 영향력이 크다.**

**향상도는 임의 선택과 비교해서 결론부를 찾는데 얼마나 규칙이 효과적인지 보여준다.**

**신뢰도는 어느 정도로 결론부가 발견될 비율을 보여준다. (얼마나 믿을 만한 규칙인지 판단가능)**

**보다 많은 레코드에 기반한 규칙일수록 결론이 좀 더 견고하다.**

**(아마도 동일한 아이템들을 다루는 여러 규칙들을 합하여) 더 많은 분명한 규칙들을 세밀하게 고려할수록 적어도 일부가 우연한 표본추출의 결과에 근거할 가능성이 더 크다.**

**한 사람이 동전을 10번 던져서 10번의 앞면이 나왔다면 매우 놀라울 것이다.**

**만일 1000명이 동전을 10번씩 던져서 그 중 10번의 앞면이 나온 사람이 있다면 그다지 놀랍지 않을 수 있다.**

개인화 추천기술의 필요성

* 컨텐츠의 증가로 인하여 사용자가 원하는 컨텐츠를 선택하기가 어려워짐
* 사용자의 취향에 맞는 콘텐츠를 필터링하여 자동으로 제공
* 인터넷 쇼핑몰에서의 상품 추천 (아마존, 넷플릭스, Spotify, Pandora음악)
* 디지털 TV 환경에서의 프로그램 추천 (IPTV)
* 웹 환경에서의 웹 사이트, 페이지 추천, SNS 상 추천 (구글, 유튜브,, 페이스북, 린크드인)

개인화된 추천을 제공하기 위해 고려할 수 있는 정보

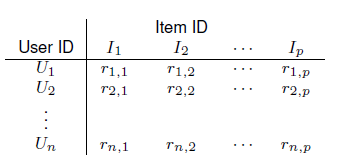
* 사용자의 인구통계학적 속성 정보 (나이, 성별 등)
* 컨텐츠의 속성 정보 (제목, 설명문)
* 사용자의 과거 이용 정보 (상품 구매, 서비스 이용 등)
* 상품의 과거 구매 정보 (구매 패턴 분석)

개인화 추천기술의 종류

* 내용기반 추천 (Content-based Recommendation): 사용자가 이전에 이용한 컨텐츠와 내용 정보가 유사한 다른 컨텐츠를 추천
* 인구통계학적 추천 (Demographic Recommendation): 사용자의 일반적인 정보 이용 (나이, 성별, 거주지 등) 규칙기반 (30대 직장인에게 노트북 추천) 연관규칙 (a를 사면 b추천)
* **협력적 추천 (Collaborative Recommendation or Social Filtering)**

**User based methods / Item based methods**

Item-user matrix: 항목에 대한 사용자 선호도를 측정값

  **(Uu , Ii , rui)**

Ratings: 선호도

Binary: 샀는지 안 샀는지? / 클릭을 했는지 안 했는지? / 좋아요를 눌렀는지 안 눌렀는지?

**사용자기반 협업 필터링 (User based methods) / people like you**

* 사용자와 성향이 비슷한 사용자들이 공통적으로 좋아하는 컨텐츠를 추천
* 과거에 이용한(구매한) 컨텐츠가 비슷하면 사용자 성향이 비슷하다고 판단
* 협력적 추천, CF 추천, 협력적 여과, 사회적 여과 등으로 불린다.

A

C

B

A

B

**High**

**Correlation**

**Like**

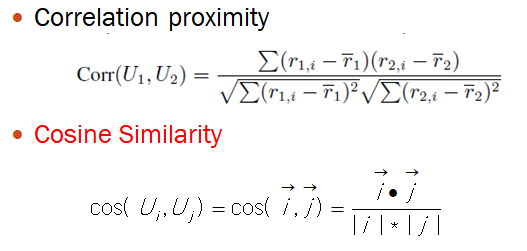
**Like**

**Recommend**

* 구매하지 않은 항목 C를 추천한다.
* 선호도 벡터에 근거하여, 사용자와 가장 비슷한 사람을 찾는다.
* **Measuring Proximity (유사도 계산)**

**유클리디언 거리는 적절하지 않음**

**피어슨 상관계수 유사도가 더 좋음: co-rated 된 항목들을 찾아 그러한 항목들에 대한 그들의 선호도 벡터 사이의 상관계수를 계산한다.**



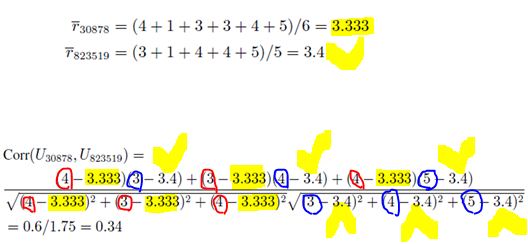
신규 고객이나 신규 아이템일 경우, Correlation 과 Cosine 값을 알 수 없을 때를 Cold start라고 한다.



**선호도 벡터.**

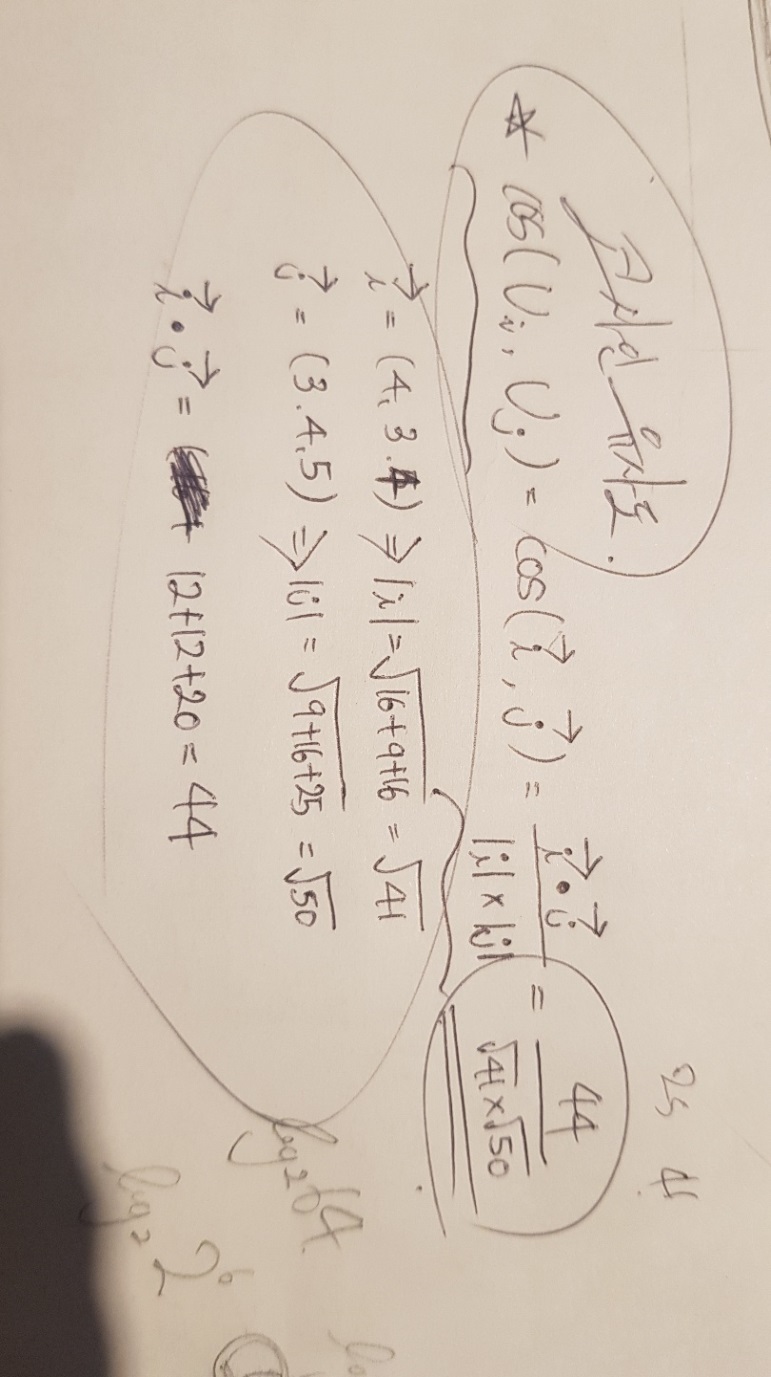
**Co-rated 된 것.**

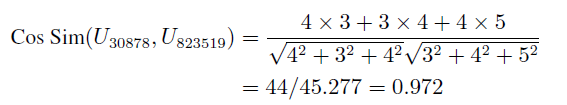
**U1 = {4, 3, 4} / U2 = {3, 4, 5}**



**Correlation = 0.34**

**양의 값으로 클수록 서로 유사한 것이다.**

****

****

**0 (no similarity) ~ Cos Sim ~ 1 (perfect match)**

**Cos Sim = 0 (두 벡터의 이루는 각도가 90도)**

**즉 서로 반대로 움직이는 관계**

**누가 선호하면 다른 사람은 선호하지 않는다.**

관심대상인 어떠한 사용자에 대해, 상관계수, 코사인 유사도, 또는 다른 측정방법을 사용하여 데이터베이스에 있는 다른 사용자들과의 유사도를 각각 계산한다.

그 다음 2단계에서는 **k-최근접 사용자들을 관찰하고,** 그들이 평가하고 구매한 다른 모든 항목들 중에서 최고 항목을 선정하여 관심대상의 사용자에게 추천한다.

**무엇이 가장 좋은 추천일까?**

이진화의 구매데이터의 경우 가장 많이 구매된 항목이다. 평가 점수 데이터의 경우 가장 좋게 평가된, 가장 많이 평가된, 혹은 이 둘의 가중치의 항목일 수 있다.

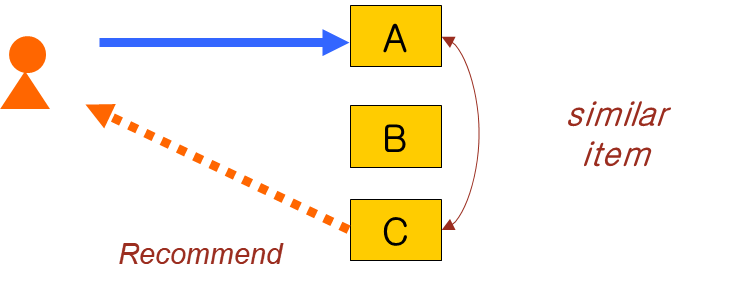
**최근접이웃 방법은** 규모가 큰 사용자들의 데이터베이스를 가지고 있는 경우에는 계산량이 많아질 수 있다. 이 문제점의 한 가지 해결방법은 **군집 방법론**을 적용하여 사용자들의 선호에 따라 동종의 군집으로 그룹화하고, **각각의 군집들과** **관심대상 사용자간의 거리를 측정하는 것이다.**

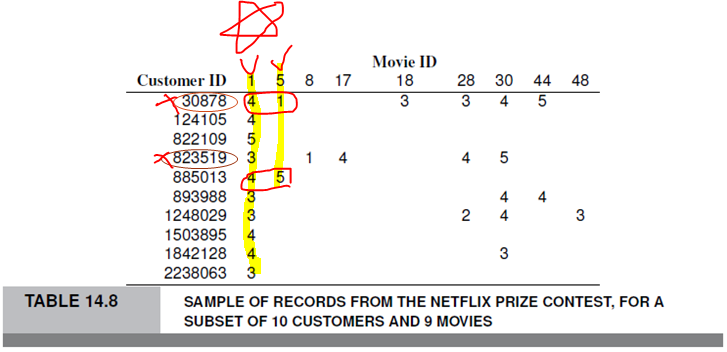
이는 해당 사용자와 각각의 군집들을 동시에 비교함으로써 더 효율적이며 빠르다.

**군집화로 인한 단점은** 가장 가까운 군집의 멤버들이 해당 사용자와 가장 비슷한 것은 아니기 때문에 비교적 덜 정확한 추천들을 생성할 수 있다는 점이다.

**항목기반 협업 필터링 (Item based methods)**

* 사용자의 수가 항목들의 수보다 훨씬 큰 경우, 비슷한 사용자들보다 비슷한 항목들을 찾는 것이 계산적으로 효율적이고 빠르다.
* 상관계수, Cos Sim 사용
* 임의의 사용자가 관심을 가지는 항목과 공동으로 평가 혹은 구매한 항목들을 찾는다.
* 비슷한 항목들 중에서 가장 대중적이거나 상관관계가 높은 항목을 추천한다.



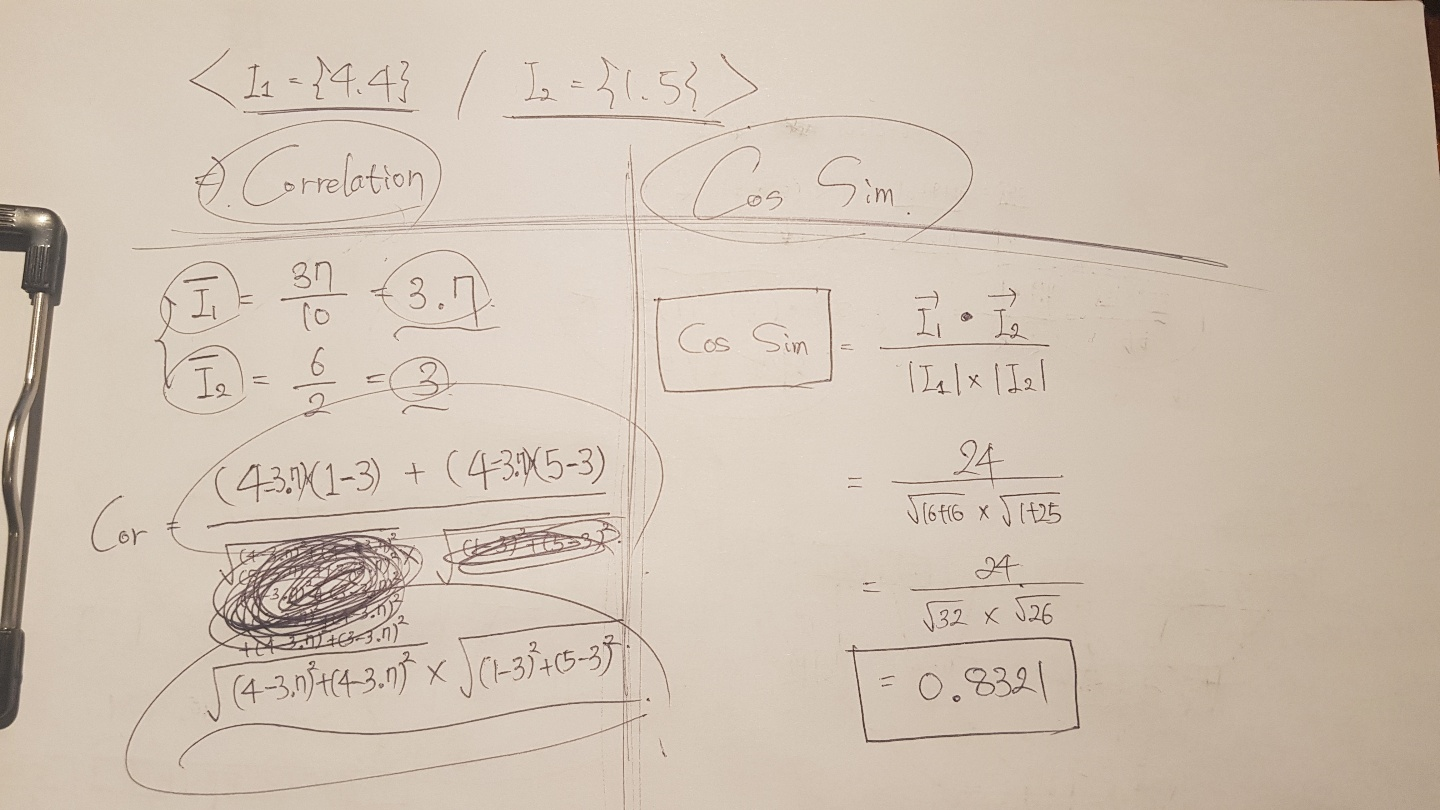


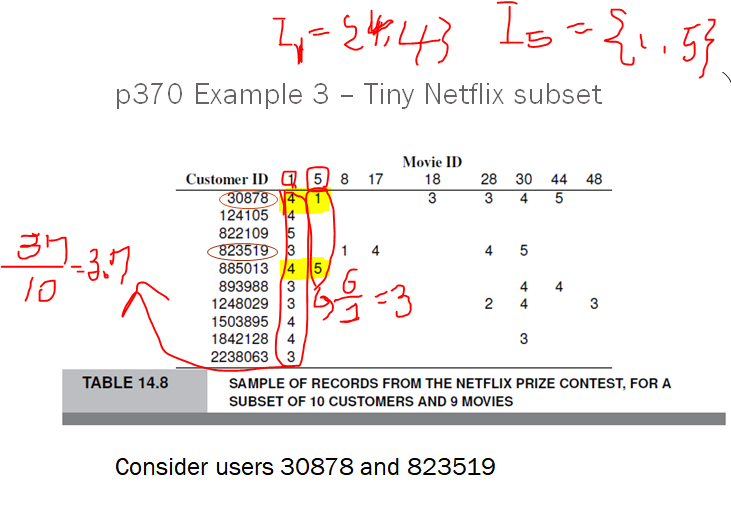
**선호도 벡터.**

**Co-rated 된 것.**

**I1 = {4, 4} / I2 = {1, 5}**







연관규칙 (유사성 분석, 장바구니 분석)은 매매거래 데이터베이스에서 항목들 사이의 연관성에 대한 규칙을 생성한다.

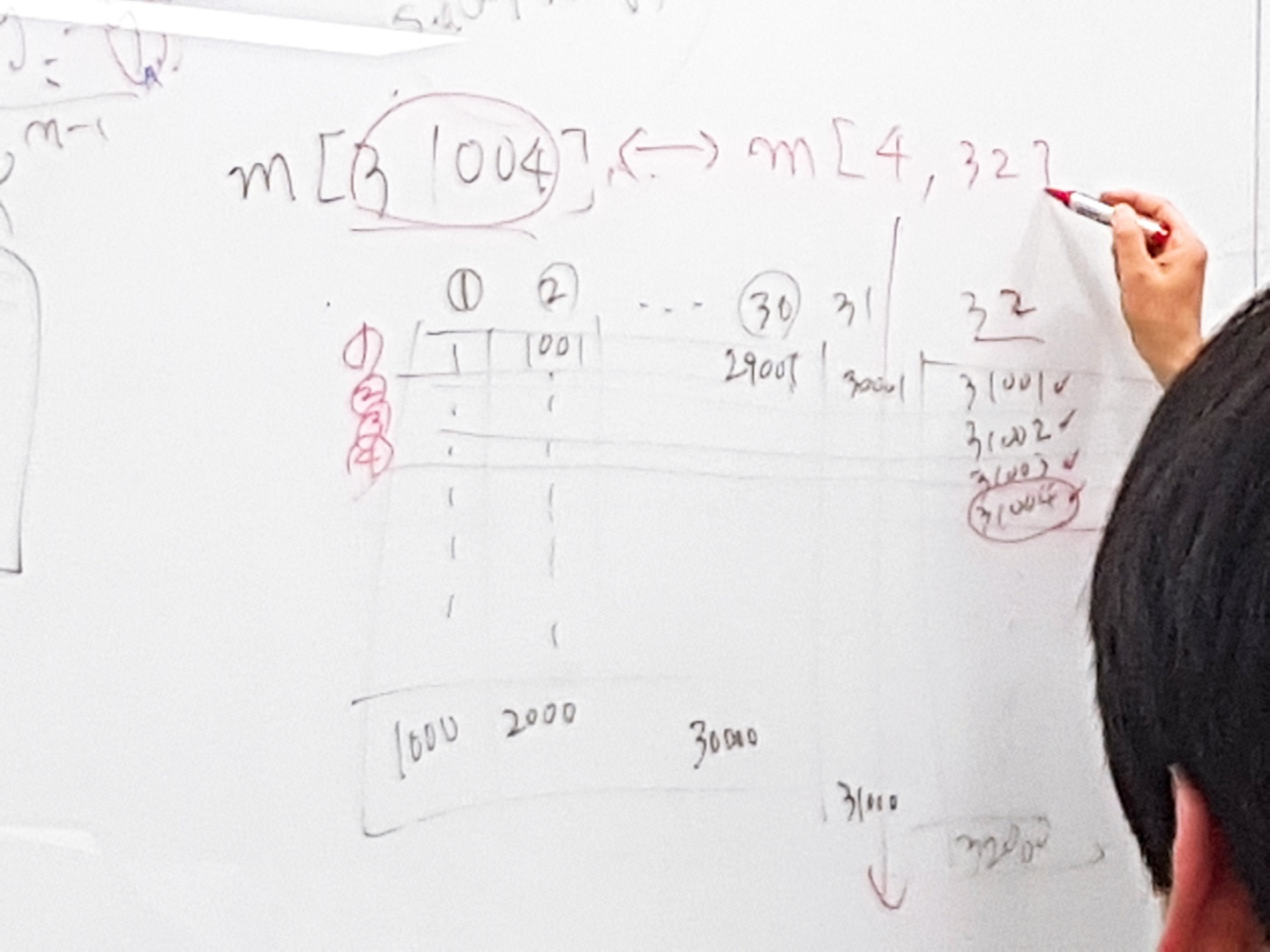
**추천 시스템**에 광범위하게 사용된다.

가장 유명한 방법은 Apriori 알고리즘이다.

**계산을 줄이기 위해, 오직 “빈발” 항목집합 (=지지도) 만 고려한다.**

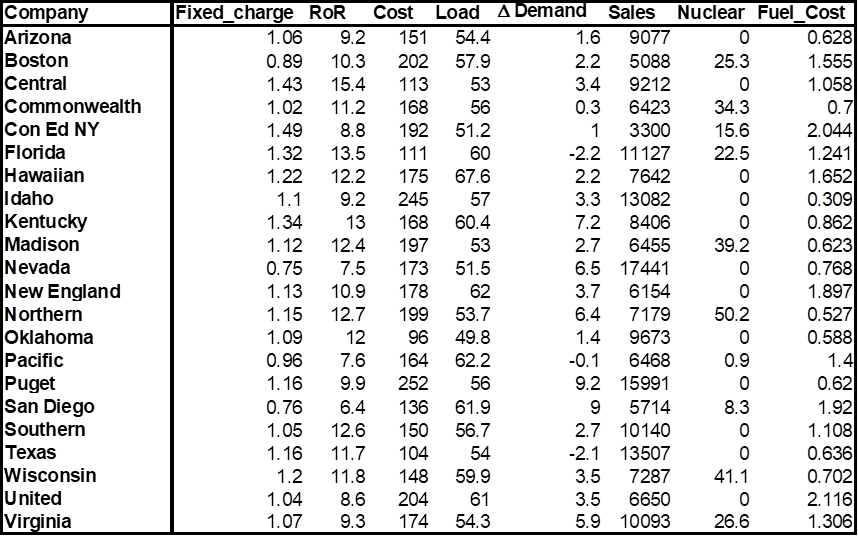
**평가는 신뢰도와 향상도로 측정된다.**

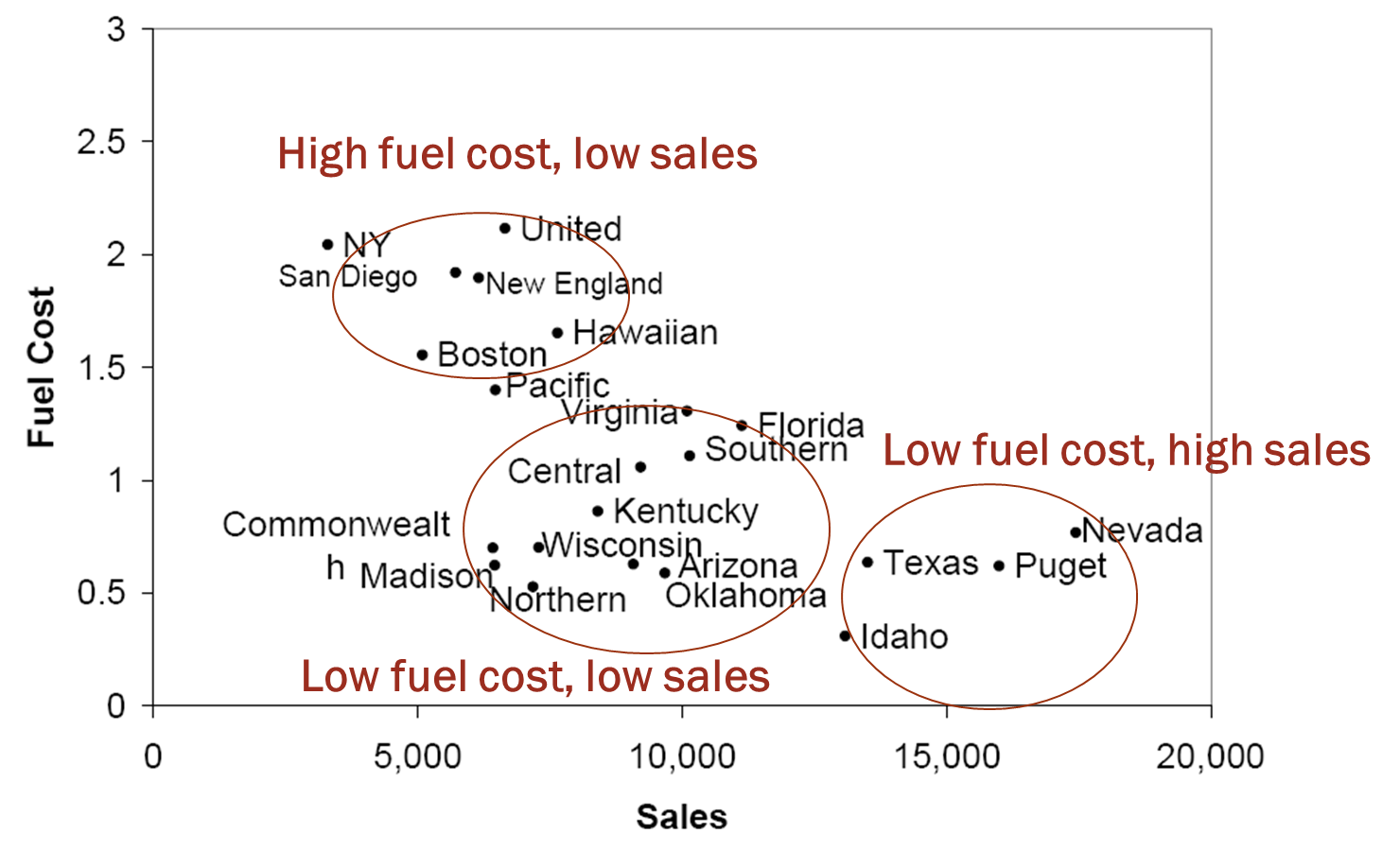
규칙을 너무 많이 생성할 수 있으므로 유용한 규칙을 찾고, 반복되는 것을 축소하도록 검토가 요구된다.



군집 분석 (Cluster Analysis)

* 데이터 세트에 대한 통찰력을 생성할 목적으로 비슷한 레코드들의 그룹(군집) 형성
* 비슷한 고객 그룹으로 **시장을 세분화하는데** 사용됨.
* 원소 주기율표, 생물 종 분류, 주식 포트폴리오 그룹화, 경제 구조분석을 위한 회사 그룹화, 육군 군복 크기, 마케팅(고객 정보를 이용한 시장 세분화), 인터넷 검색엔진(사용자들이 검색한 검색어를 군집화하여 검색 알고리즘 향상)





* 유사한 전력회사들의 군집 찾기 (22개 회사, 8개 변수, 연속형 데이터)

2차원 이상으로 확장

* 위 예의 군집화는 눈으로 볼 수 있는 정도이다.
* 다차원의 경우, 거리 측정 방법과 군집을 형성하는 데에서 거리를 이용하는 방법을 사용
* 계층적, 비계층적 알고리즘

계층적 방법

* 응집 방법

가장 유사한 것들끼리 묶어 나가는 것

n-군집으로 시작 (각각의 레코드를 자신의 군집으로)

하나의 군집이 남을 때까지 레코드들을 군집 안으로 합함 (완전한 전체 데이터 세트)

가장 많이 쓰인다.

* 분할 방법

정보의 손실이 가장 적도록 분할 하는 것

전체를 포함하는 하나의 군집으로 시작

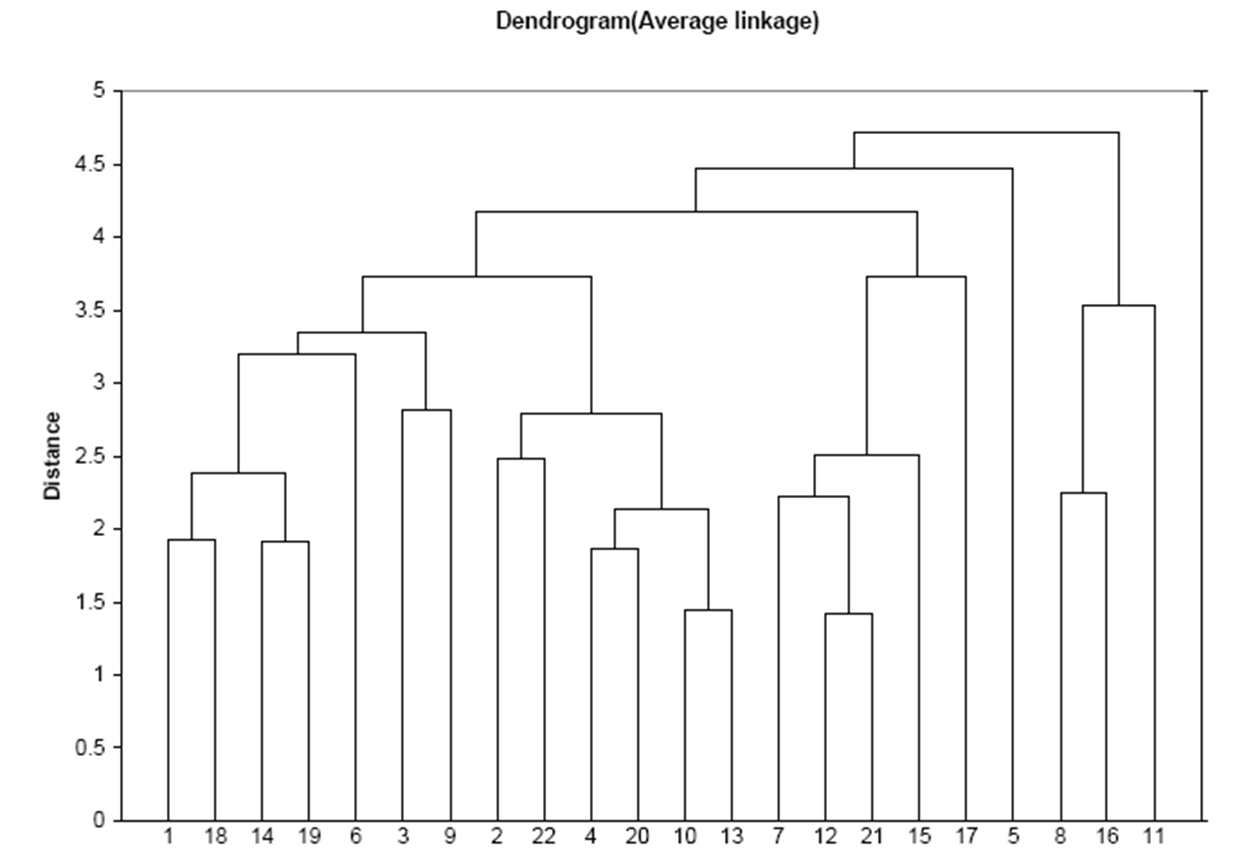
더 작은 군집들로 계속 분할

비계층적 방법

* K-평균 군집화 방법

K개의 군집이 미리 정해져 있음

계산 집약적이지 않으므로 대용량 데이터에서 선호

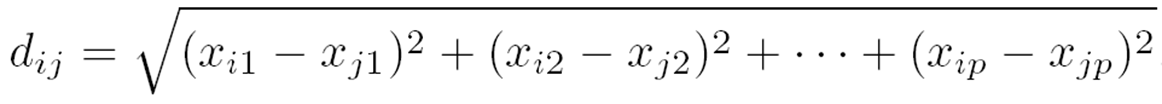


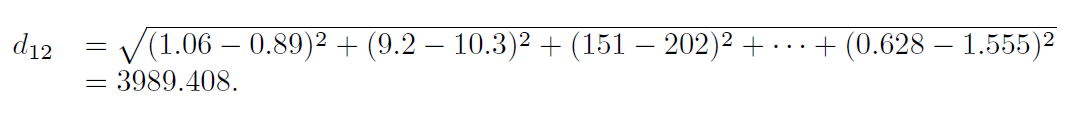
* 댄드로그램은 군집 계층을 보여준다.

거리 측정

* 레코드들 간 / 군집들 간
* 2개의 레코드 간의 거리

유클리드 거리가 가장 유명





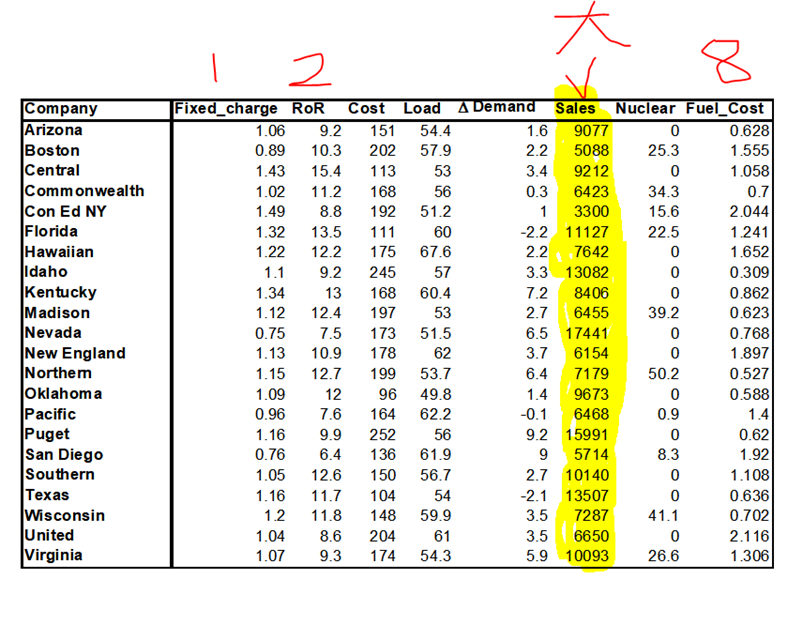
* 애리조나와 보스턴 공공 전력회사 간의 유클리드 거리

정규화

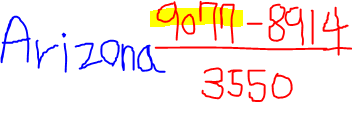
* 원시 거리 측정은 측정치의 단위에 크게 영향을 받는다.
* 먼저 데이터를 정규화(표준화)하므로 해결할 수 있다.

평균을 빼고, 표준편차로 나눈다.

Z-SCORES라고도 부른다. (표준 정규 분포)



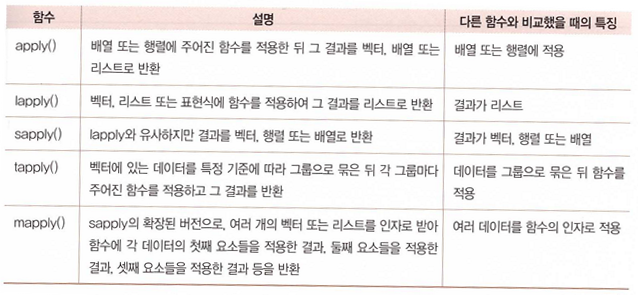
* Sales에 의해 값이 좌지우지 되므로 측정치의 값을 정확하게 하기 위해 정규화 작업을 진행하는 것이다.
* Sales 평균 = 8,914 / 표준편차 = 3,550





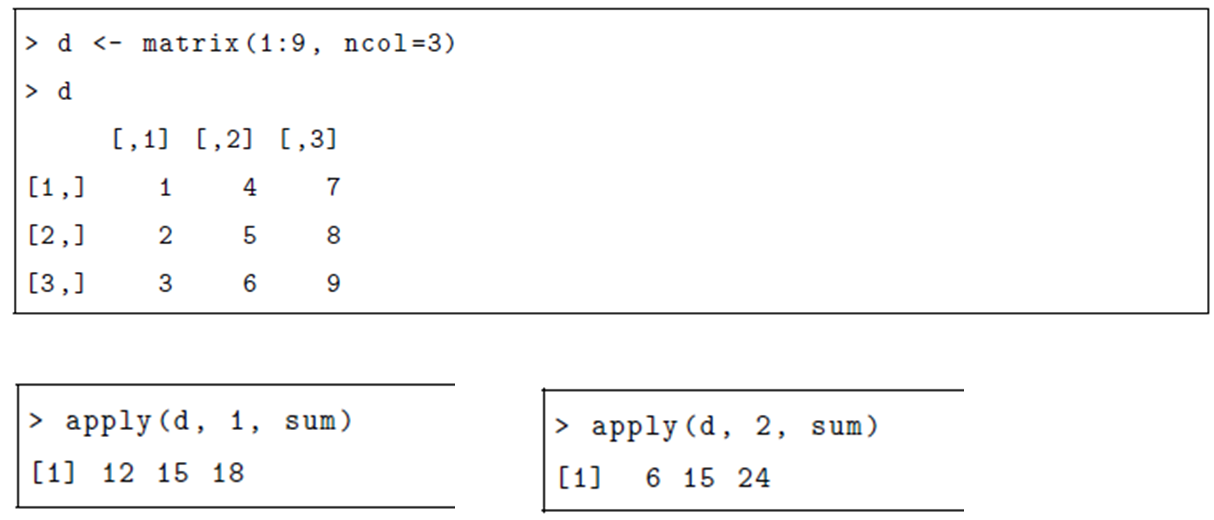
* 표준정규분포는 평균이 0이고 표준편차가 1인 정규분포이다.

즉 평균에 가까운 애들은 거의 0에 가까운 값이 나오게 되고, 조금 떨어진 애들은 +,-1이 나오게 되는 것이다.



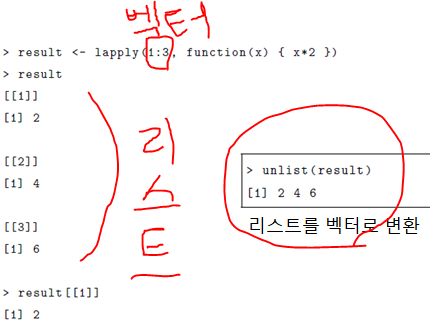
apply(행렬, 방향, 함수)

* 벡터에는 적용 안됨
* 방향은 1이면 행 / 2이면 열



lapply(X, 함수)

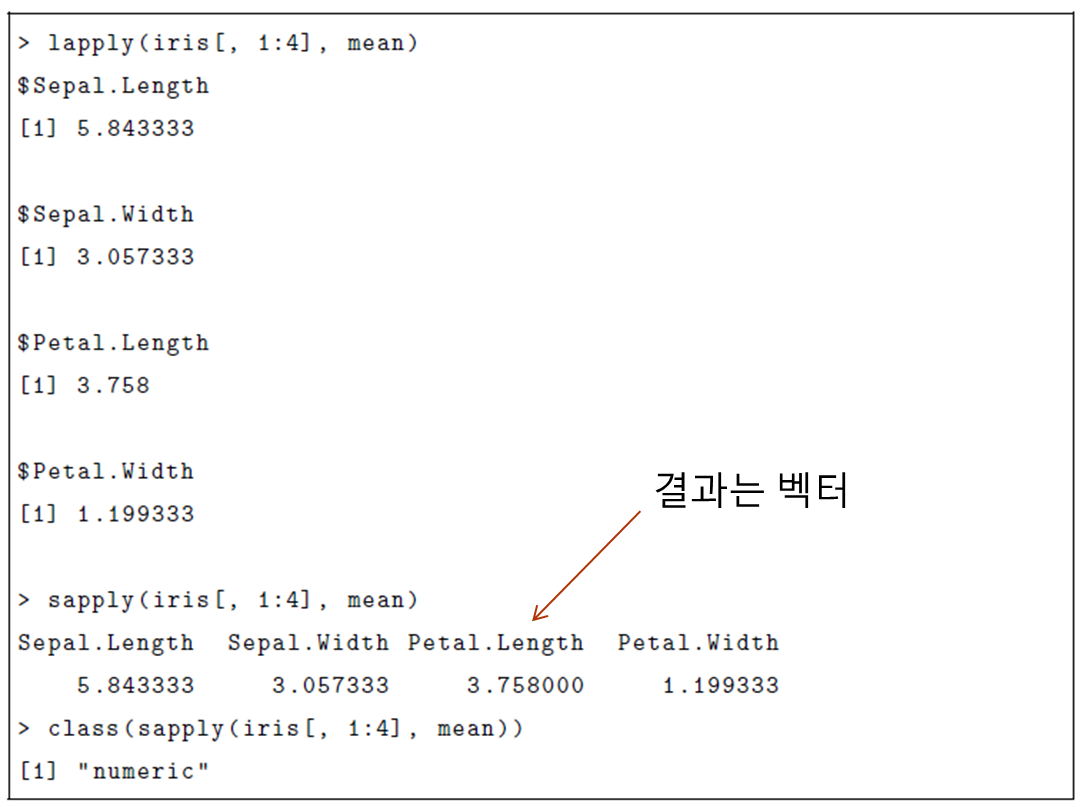
* X는 벡터 또는 리스트, 데이터 프레임 등 가능
* 결과는 list 형태

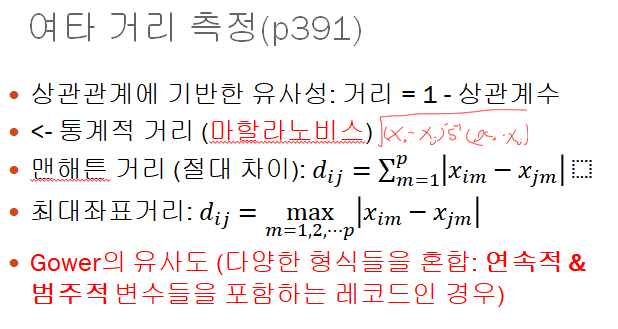


unlist (리스트를 벡터로) 🡪 matrix (벡터를 행렬로) 🡪 as.data.frame (행렬을 데이터프레임으로) 벡터를 직접 데이터프레임으로 바꾸면 1column 프레임으로 변환됨

sapply()

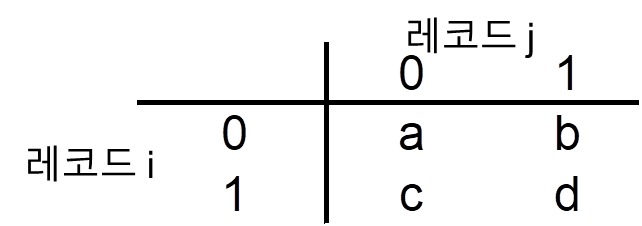
* lapply()와 유사하지만 list 대신 행렬, 벡터 등으로 결과를 반환
* 입력: 벡터, 리스트 등





범주형 데이터의 유사성

* 이진형 변수(0/1)로 구성된 2개 레코드 간의 거리를 측정하기 위해, 도수 표 생성



* a: 두 레코드간 속성이 존재하지 않는 예측변수 개수
* d: 두 레코드간 속성이 존재하는 예측변수 개수
* b, c: 두 레코드간 속성이 불일치 하는 개수
* 매칭 계수: (a+d) / (a+b+c+d)
* 자카드 계수: d / (b+c+d)

1매칭이 0매칭보다 유사성이 훨씬 드러나는 경우에 사용 (스포츠카 소유: 1이 0보다 두 레코드간의 유사성에 더 큰 의미가 있음)

군집 간의 거리 측정

* 최단 거리 (minimum distance)

단일연결법 (single linkage)이라고도 함

두 군집 간의 거리는 가장 가까운 레코드 쌍 사이의 거리이다.

* 최대 거리 (maximum distance)

완전연결법 (complete linkage)이라고도 함

두 군집 사이의 거리는 서로 가장 먼 레코드 쌍 사이의 거리이다.

* 평균 거리 (average distance)

평균연결법 (average linkage)이라고도 함

두 군집 간의 거리는 모든 가능한 거리 쌍의 평균이다.

* 중심 거리 (centroid distance)

두 군집 간의 거리는 두 개의 군집 중심점(centroid linkage)사이의 거리이다.

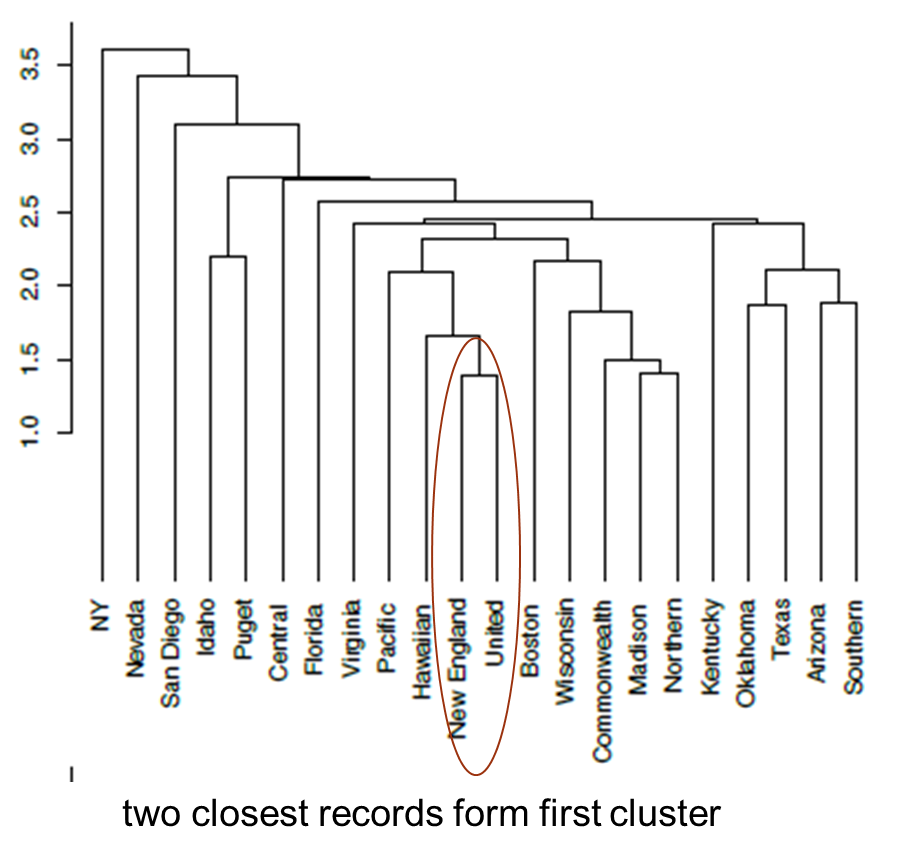
중심점은 군집 내 모든 레코드들을 변수 별로 평균한 벡터이다.

* median: 각 중간 값을 구해서 median 사이의 거리를 구한다.
* ward.D: 레코드가 군집화하면서 발생하는 ‘정보의 손실’을 고려한다. 정보손실을 측정하기 위해서 워드방법은 ‘오차제곱합(ESS)’을 이용한다. 군집의 수가 적은 쪽으로 움직이는 워드 방법은 정보의 손실이 가장 적은 군집화를 선택한다.

**계층적 군집화 단계 (응집 방법 사용) 사용되는 함수는 hclust()**

1. N 군집으로 시작 (각각의 레코드는 자신의 군집)
2. 가장 가까운 레코드 2개를 하나의 군집으로 병합
3. 각각의 연속적 단계에서, 서로 가장 가까운 2개의 군집이 병합된다.

**덴드로그램 읽는 방법**



* 군집화 과정 보기

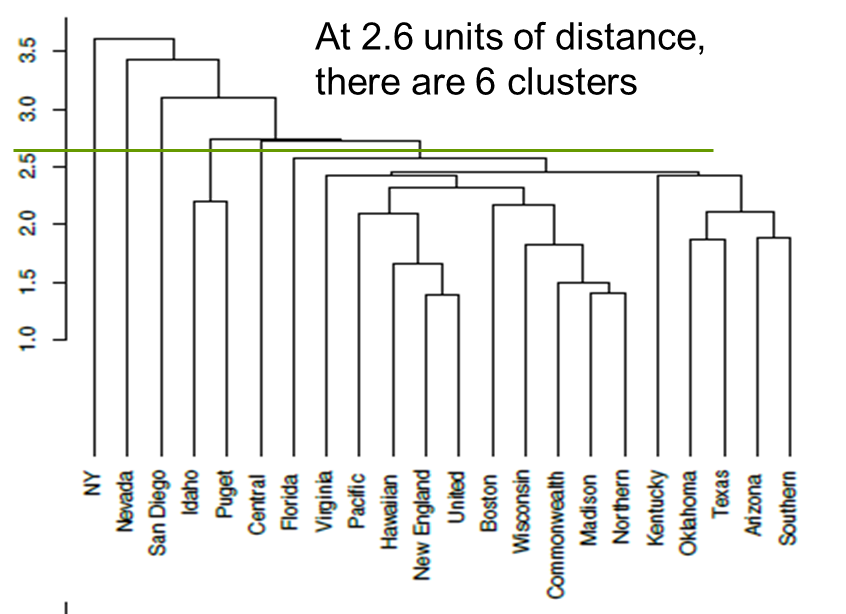
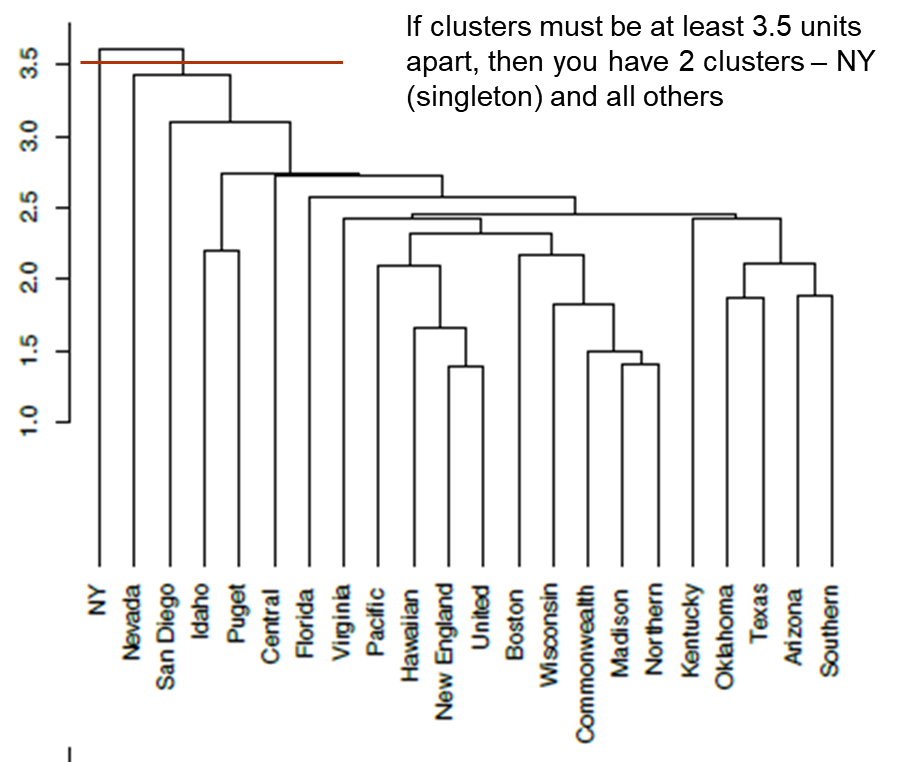
수평선을 그렸을 때, 아래쪽에 연결된 선들이 먼저 병합된다.

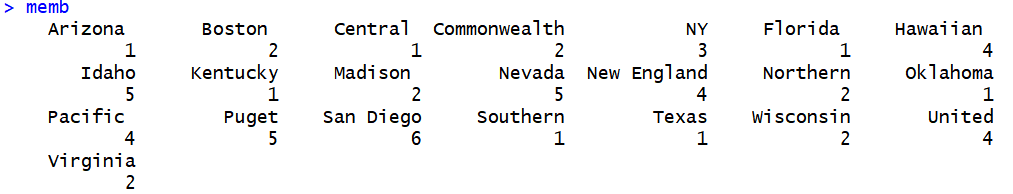
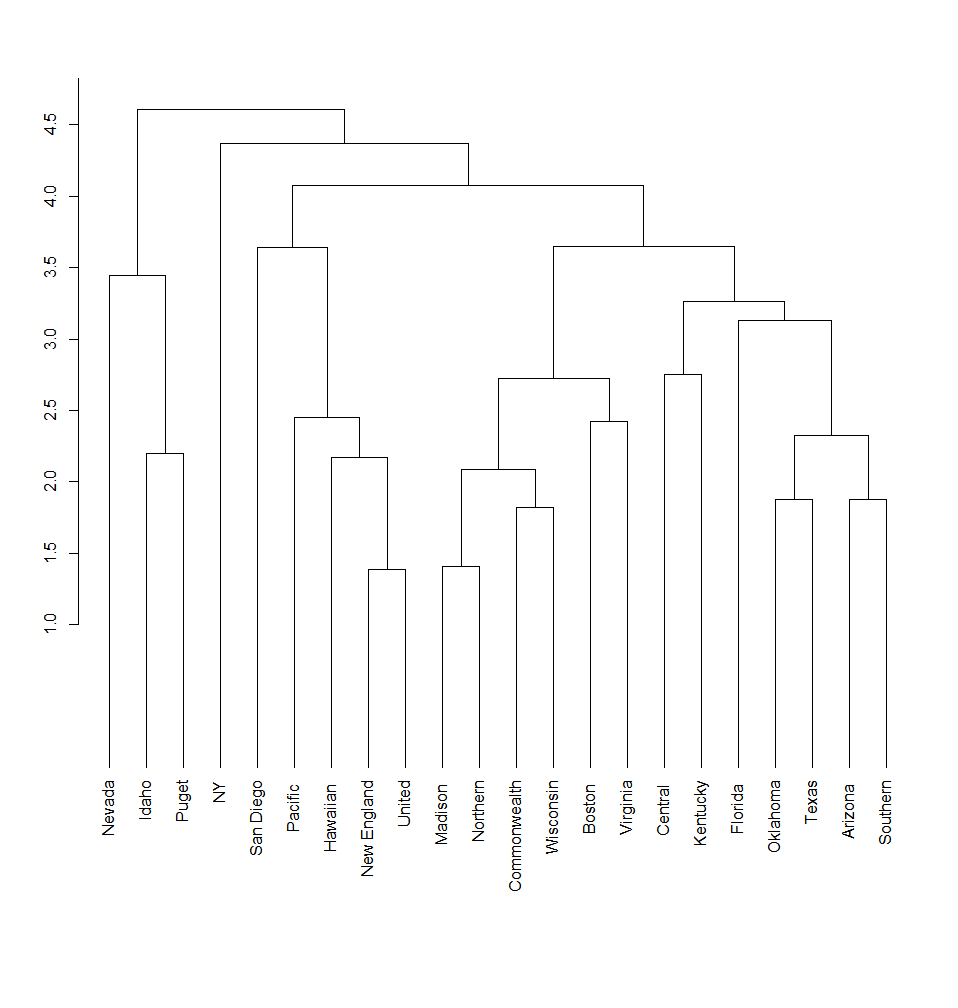
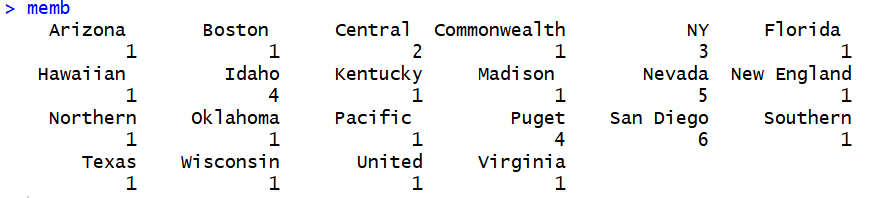
New England와 United가 가장 먼저 병합된다. Y축 수치상으로 가장 낮음

그 다음, Madison와 Northern 병합

* **군집의 개수 결정**

주어진 “군집 간의 거리”에서 수평선은 군집을 생성하기 위해 멀리 떨어진 군집들을 교차한다.





군집 검증

* 의미 있고 유용한 군집을 얻는 것이 목표

군집의 바람직한 특성들

* Cluster Interpretability (군집의 해석 가능성)

군집에 사용된 각 변수들의 요약통계량(평균값, 최솟값, 최댓값)을 구한다.

또한 군집화에 사용되지 않은 변수 측면에서 군집을 검토한다.

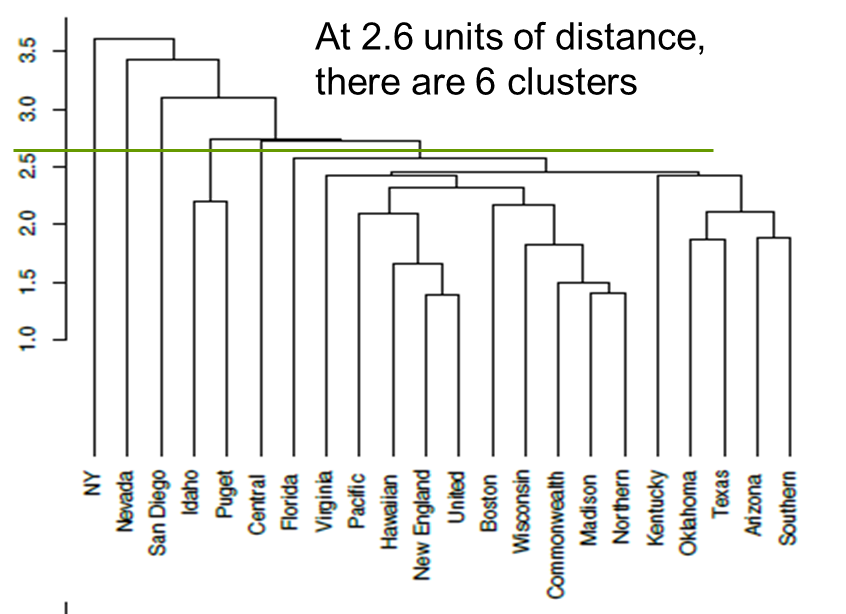
군집에 적절한 라벨을 붙인다.

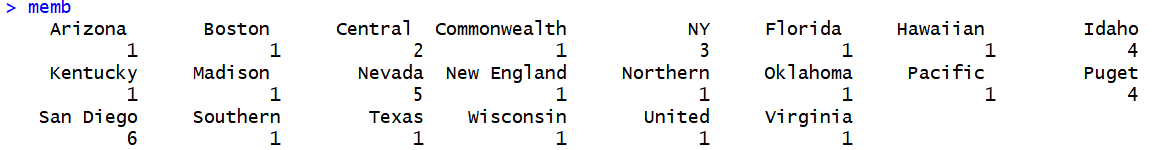
* Cluster Stability (군집의 안정성)

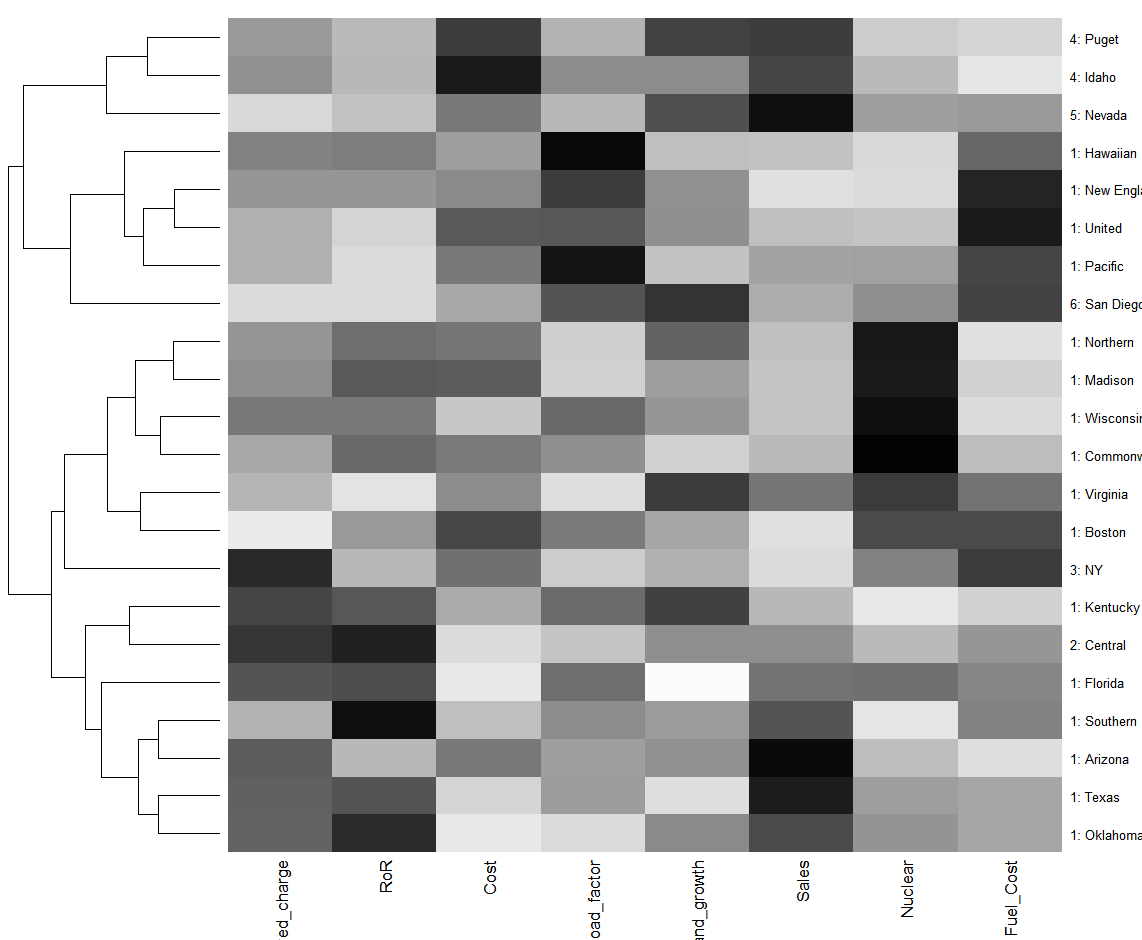
입력값들이 약간 변경되면 군집결과가 유의하게 변하는가? 안정성을 검토하는 또 다른 방법은 데이터를 분할한 후, 한쪽 데이터에 근거하여 형성된 군집이 나머지 데이터에 얼마나 잘 적용되는지 살펴보는 것이다.

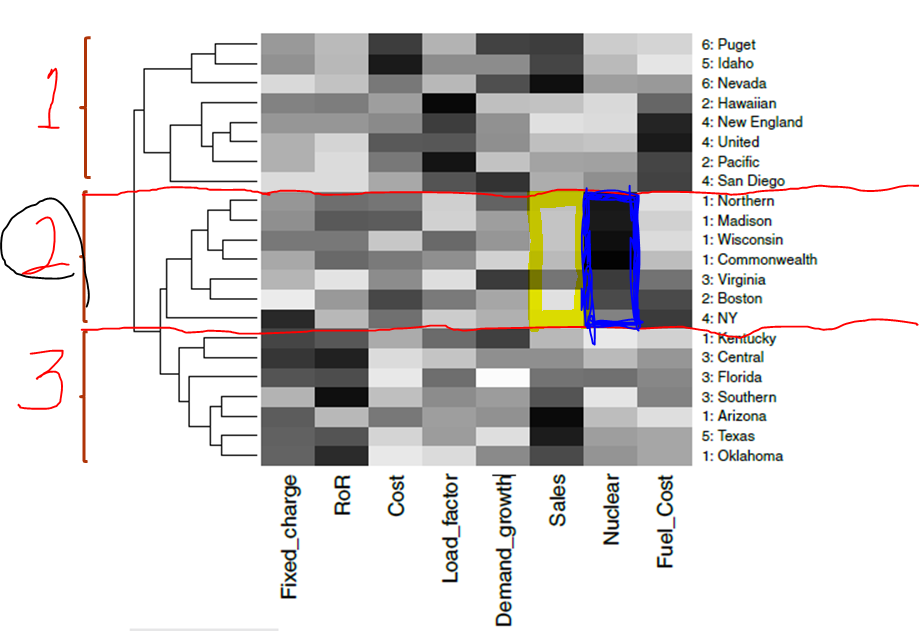
* Cluster Separation (군집의 분리)

군집의 분리가 합리적인지 살펴보기 위해, 군집 내 변동에 대한 군집 간 변동의 비를 검토한다. (높을수록 좋다.)







오른쪽 숫자는 cutree 6으로 나눈 것이다. 왼쪽 군집 그림은 초기 응집 결과가 아니라, 생성된 히트맵을 기준으로 색에 따라 군집화를 다시 한 그림이다. 

* 높은 값일수록 어두운 색 (1) / 낮은 값일수록 밝은 색 (99) 을 나타낸다.
* 3개의 클러스터를 고려해본 결과 ‘2’번의 군집은 Sales는 낮고 Nuclear는 높은 값인 것을 알 수 있다.
* 또한, ‘3’번의 군집은 나머지 2개의 군집보다 Fixed\_charge와 RoR이 높은 값인 것을 알 수 있다.

계층적 군집화의 제한

* 오직 N x N 거리행렬을 계산하고 저장해야 한다. 따라서 매우 큰 데이터세트에서는, 계산비용이 많이 들고 계산속도가 느려질 수 있다.
* 계층적 군집화를 하기 위한 알고리즘은 데이터를 한 번 통과시킨다. 이것은 분석 과정 초기 단계에서 레코드가 잘못된 군집에 속하게 되면, 그 이후에 다시 수정할 수 없다는 것을 말한다.
* 안정성이 낮은 경향이 있다. 데이터를 재정렬하거나 몇 개의 레코드를 제외시킬 경우, 전혀 다른 군집결과가 나타날 수 있다.
* 이상치에 민감하다.

비계층적 군집 (k-평균 군집화 알고리즘)

1. 원하는 군집의 개수 k를 선택
2. k개의 군집으로 분할하면서 시작
3. 매 단계마다, 각각의 레코드를 가장 가까운 중심점의 군집으로 이동
4. 중심점 재계산, 3단계 반복
5. 레코드 이동이 군집 내 분산을 증가시킬 때 중단

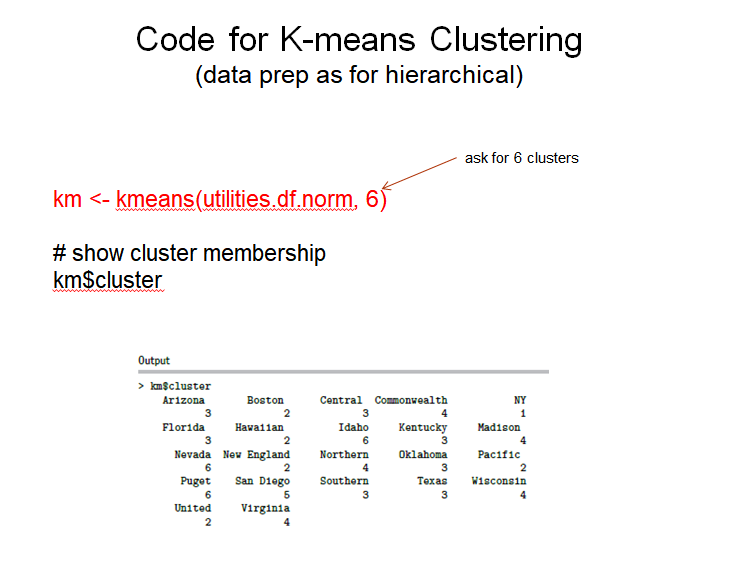
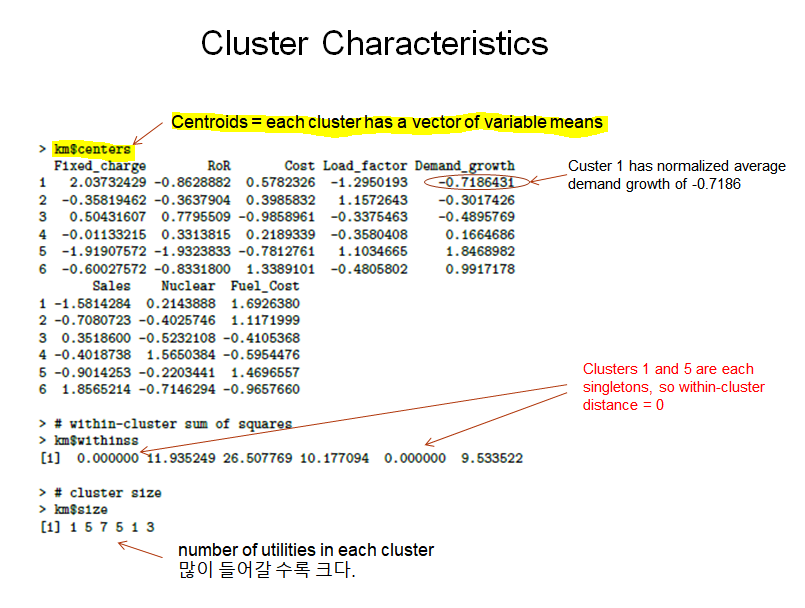
k-평균 알고리즘

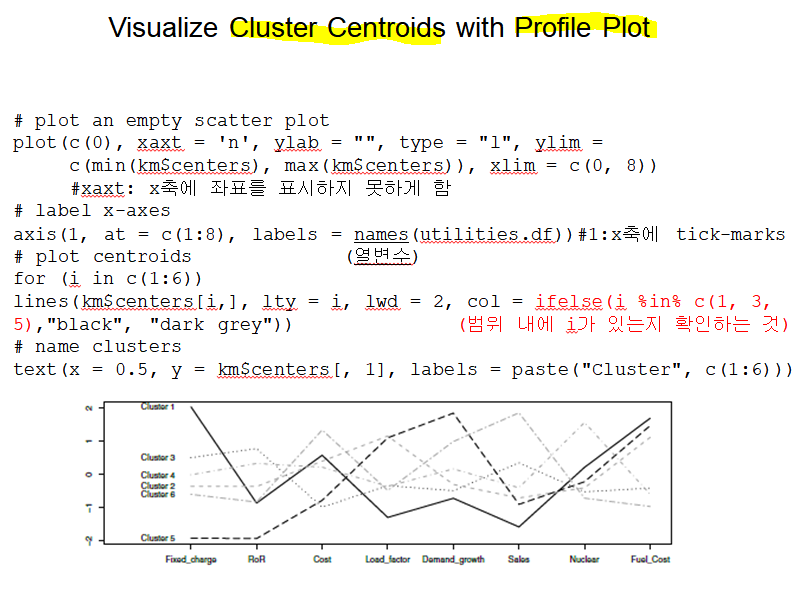
* k를 선택하고 초기 분할

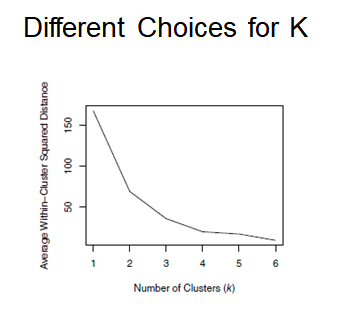
어떠한 결과가 사용될 것인가에 기반하여 k를 선택

또한, 군집으로의 초기 분할은 임의적이거나 domain 지식에 근거할 수 있다.

**만약 임의 분할이라면**, 다른 임의 분할로 과정을 반복하라.





군집거리에 따라 어떻게 되는가?

군집의 수가 증가함에 따라, 각각의 군집은 서로 가까워 진다.

* K가 1에서 2로 이동할 때 군집이 상당히 단단해진다. (군집 내 거리가 크게 감소)
* 군집의 개수를 4 이상으로 추가하게 되면 군집의 동질성 향상이 덜하게 된다.
* 군집 3과 4는 서로 가장 가까운 것을 볼 수 있다. 군집 1과 5는 서로 가장 멀리 떨어짐

요약

* 군집 분석은 설명적 도구이다. 오직 의미 있는 군집을 생산할 때만 유용하다.
* 계층적 군집화는 다른 수준의 군집화에 대한 시각적 묘사를 보여준다.

하지만 비반복적 속성 때문에, 불안정하고 설정에 대한 의존이 매우 다양하기 때문에 계산적으로 비용이 많이 든다.

* 비계층적 군집화는 계산하기에 좀더 싸고 안정적이다. 사용자가 k를 설정해야 한다.
* 양쪽 방법을 모두 쓸 수 있다.
* 기회결과에 주의하라
* 데이터는 확정적 “실제” 군집을 갖지 않을 수도 있다.