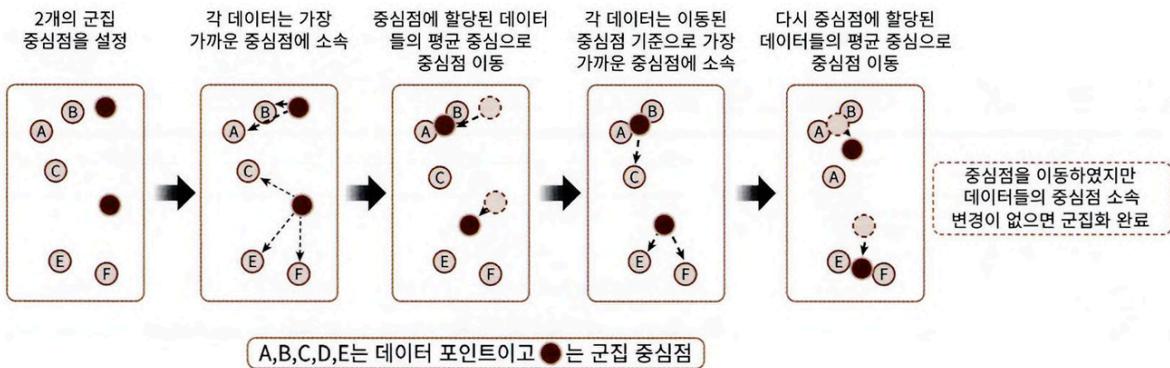


# chap7 군집화

## K-평균 알고리즘 이해

: 군집 중심점(centroid)이라는 특정한 임의의 지점을 선택해 해당 중심에 가장 가까운 포인트들을 선택하는 군집화 기법



- 먼저 군집화의 기준이 되는 중심을 구성하려는 군집화 개수만큼 임의의 위치에 가져다 놓는다.
- 각 데이터는 가장 가까운 곳에 위치한 중심점에 소속된다.
- 소속이 결정되면 군집 중심점을 소속된 데이터의 평균 중심으로 이동한다.
- 중심점이 이동했기 때문에 각 데이터는 기존에 속한 중심점보다 더 가까운 중심점이 있다면 해당 중심점으로 다시 소속을 변경한다.
- 다시 중심을 소속된 데이터의 평균 중심으로 이동한다.
- 중심점을 이동했는데 데이터의 중심점 소속 변경이 없으면 군집화 종료 / 아니라면 4번 과정 거치기

### K-평균의 장점

- 일반적인 군집화에서 가장 많이 사용되는 알고리즘이다. 알고리즘이 쉽고 간결하다.

### K-평균의 단점

- 거리 기반 알고리즘으로 속성의 개수가 매우 많을 경우 군집화 정확도가 떨어진다.
- 반복 수행할 때 반복 횟수가 많을 경우 수행 시간이 매우 느려진다.
- 몇 개의 군집을 선택해야 할지 가이드하기가 어렵다.

## K-평균을 구현하기 위한 사이킷런 KMeans 클래스

```
class sklearn.cluster.KMeans(n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001,  
                             precompute_distances='auto', verbose=0, random_state=None,  
                             copy_x=True, n_jobs=1, algorithm='auto')
```

- n\_clusters: 군집화할 개수, 즉 군집 중심점의 개수
- init: 초기에 군집 중심점의 좌표를 설정할 방식
- max\_iter: 최대 반복 횟수 / 이 횟수 이전에 모든 데이터의 중심점 이동이 없으면 종료
- labels\_: 각 데이터 포인트가 속한 군집 중심점 레이블
- cluster\_centers\_: 각 군집 중심점 좌표

## 군집화 알고리즘 테스트를 위한 데이터 생성

- make\_blobs() API : 개별 군집의 중심점과 표준 편차 제어 기능이 추가되어 있음
  - 호출하면 피처 데이터 세트와 타깃 데이터 세트가 튜플(Tuple)로 반환
  - 파라미터
    - n\_samples : 생성할 총 데이터의 개수
    - n\_features : 데이터의 피처 개수
    - centers : int 값
    - cluster\_std : 생성될 군집 데이터의 표준 편차를 의미
- make\_classification() API : 노이즈를 포함한 데이터를 만드는 데 유용하게 사용할 수 있음
- make\_circle(), make\_moon() API : 중심 기반의 군집화로 해결하기 어려운 데이터 세트 만드는 데 사용

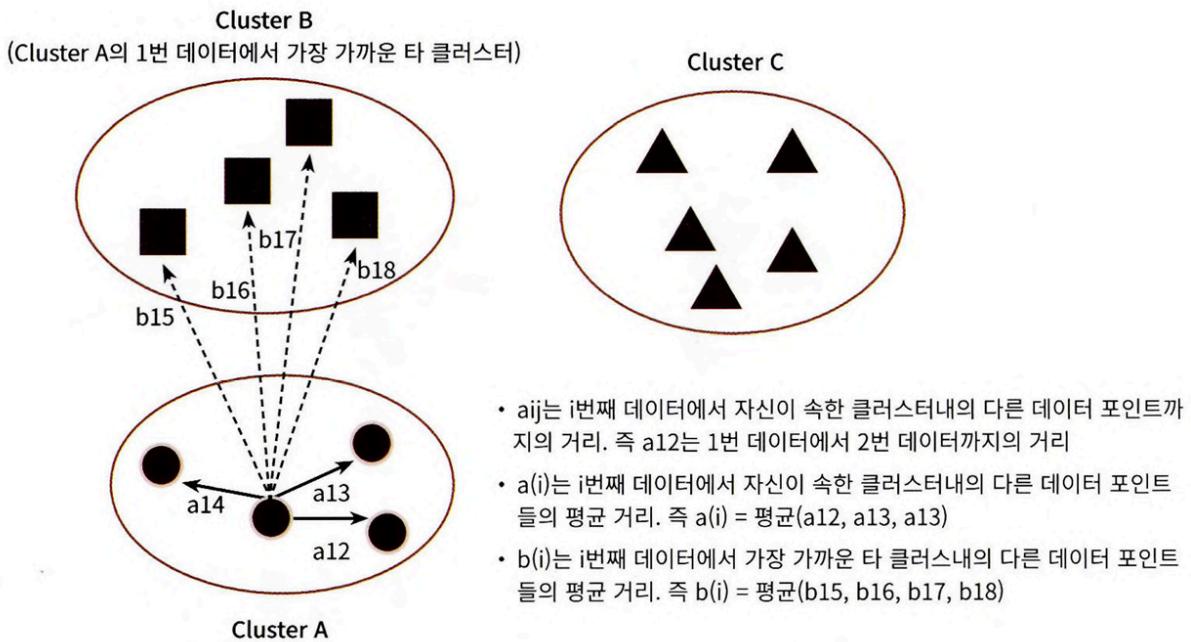
## 군집 평가 (Cluster Evaluation) - 실루엣 분석

: 각 군집 간의 거리가 얼마나 효율적으로 분리돼 있는지.

-효율적으로 잘 분리되어 있다 = 다른 군집과의 거리는 떨어져 있고, 동일 군집끼리의 데이터는 서로 가깝게 잘 뭉쳐 있다., 군집화가 잘될수록 개별 군집은 비슷한 정도의 여유공간을 가지고 떨어져 있다.

실루엣 계수: 개별 데이터가 가지는 군집화 지표로,

개별 데이터가 가지는 실루엣 계수는 해당 데이터가 같은 군집 내의 데이터와 얼마나 가깝게 군집화되어 있고, 다른 군집에 있는 데이터와는 얼마나 멀리 분리되어 있는지를 나타내는 지표



$$s(i) = \frac{(b(i) - a(i))}{(\max(a(i), b(i)))}$$

$s(i) = i$ 번째 데이터 포인트의 실루엣 계수 ( $-1 < s(i) < 1$ )

\*1과 가까울수록 근처의 군집과 더 멀리 떨어져있고,

0에 가까울수록 근처의 군집과 가깝고,  
-값은 아예 다른 군집에 데이터 포인트가 할당되었음

### 사이킷런 메서드

- `sklearn.metrics.silhouette_samples(X, labels, metric='euclidean', **kwds)` : 인자로 X feature 데이터 세트와 각 피처 데이터 세트가 속한 군집 레이블 값인 `labels` 데이터를 입력해주면 각 데이터 포인트의 실루엣 계수를 계산해 반환한다.
- `sklearn.metrics.silhouette_score(X, labels, metric='euclidean', sample_size=None, **kwds)`: 인자로 X feature 데이터 세트와 각 피처 데이터 세트가 속한 군집 레이블 값인 `labels` 데이터를 입력해주면 전체 데이터의 실루엣 계수 값을 평균해 반환합니다. 즉, `np.mean(silhouette_samples())`입니다. 일반적으로 이 값이 높을수록 군집화가 어느정도 잘 됐다고 판단할 수 있지만 무조건 이 값이 높다고 해서 군집화가 잘 됐다고 판단할 수는 없습니다.

### 좋은 군집화가 되기 위한 조건

1. 사이킷런의 `silhouette_score()`값은 0~1 사이의 값을 가져야하고, 1에 가까울수록 좋음
2. 개별 군집의 실루엣 계수 평균값이 전체 실루엣 계수의 평균값에서 크게 벗어나지 않아야 한다.

## 평균이동

### 평균이동(Mean Shift)의 개요

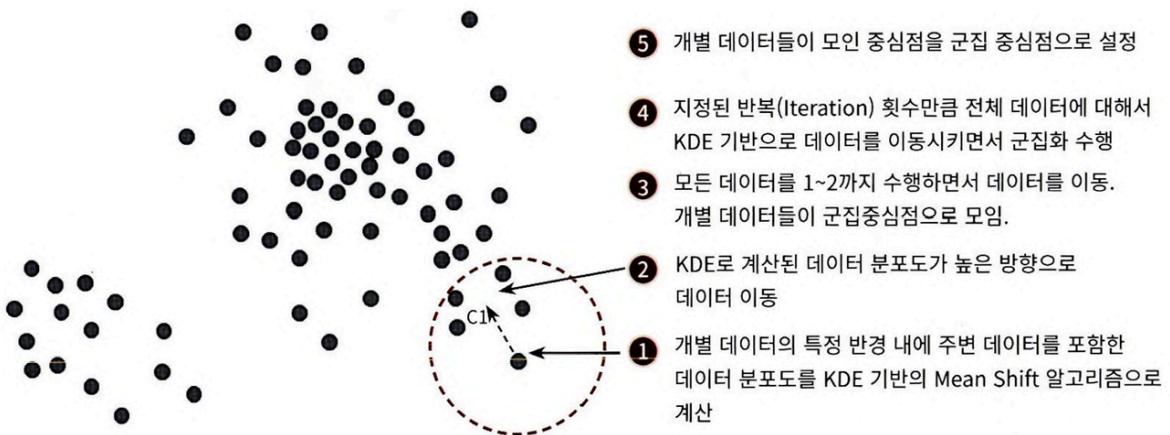
평균 이동은 K-평균과 유사하게 중심을 군집의 중심으로 지속적으로 움직이면서 군집화를 수행

- 평균 이동

K-평균	평균 이동
중심에 소속된 데이터의 평균 거리 중심	데이터가 모여 있는 밀도가 가장 높은 곳

평균 이동 군집화 : 특정 데이터를 반경 내의 데이터 분포 확률 밀도가 가장 높은 곳으로 이동하기 위해 주변 데이터와의 거리 값을 KDE 함수 값으로 입력한 뒤 그 반환 값을 현재 위치에서 업데이트하면서 이동

— 이 방식을 전체 데이터에 반복 적용하여 데이터의 군집 중심점을 찾아냄

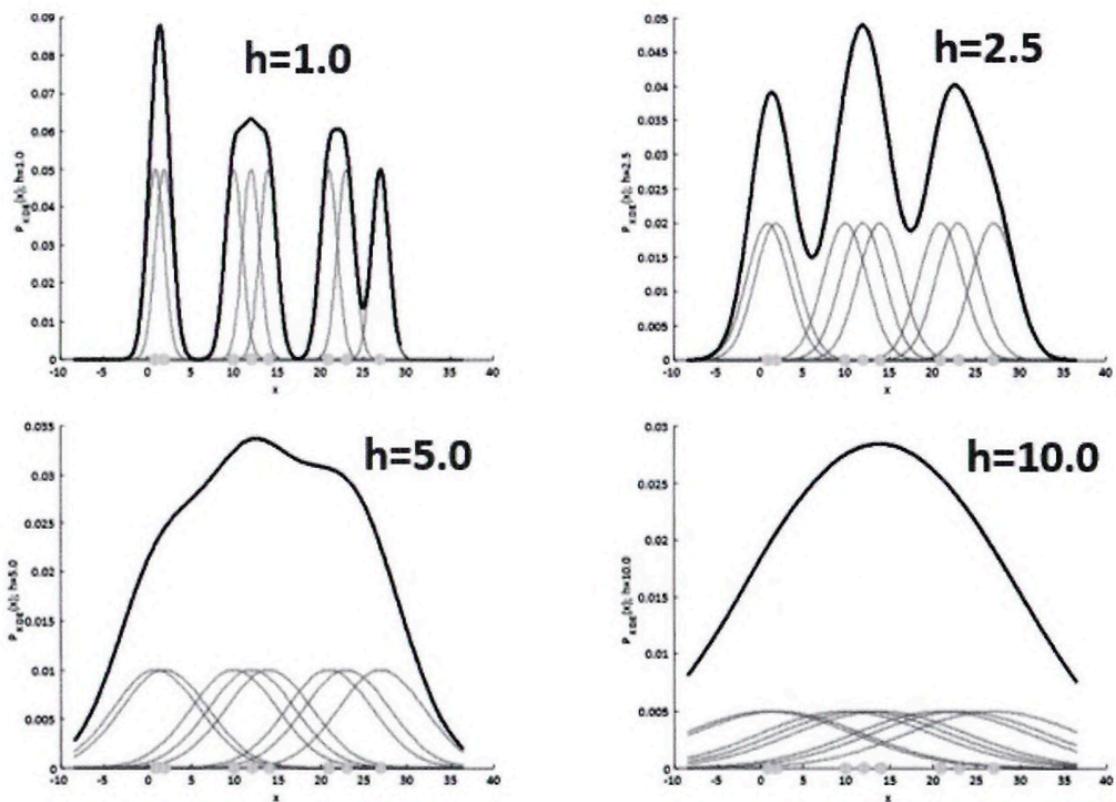


KDE(kernel density estimation):  $\sum(\text{관측된 데이터에 커널 함수 적용값}) / \text{데이터 건수} = \text{확률 밀도 함수}$

$$\text{KDE} = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

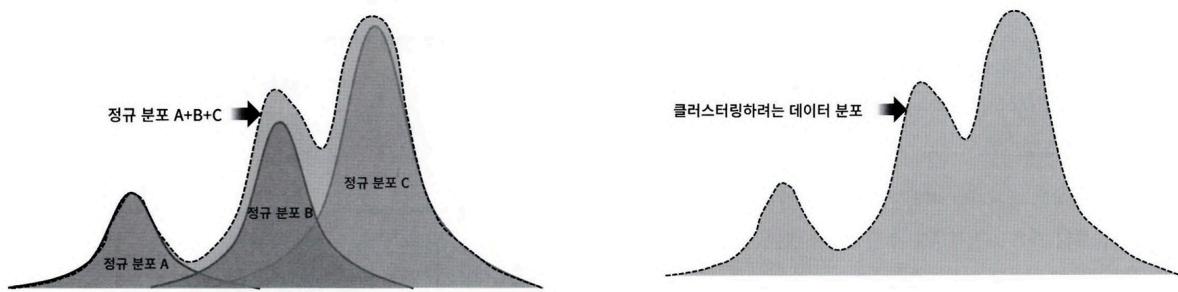
- 대표적인 커널 함수: 가우시안 분포 함수

<b><math>h = 1.0</math> (작은 값)</b>	<b><math>h = 10</math> (큰 값)</b>
좁고 뾰족한 KDE	과도하게 평활화(smoothing)된 KDE
변동성이 커, 과적합하기 쉬움	과소적합하기 쉬움

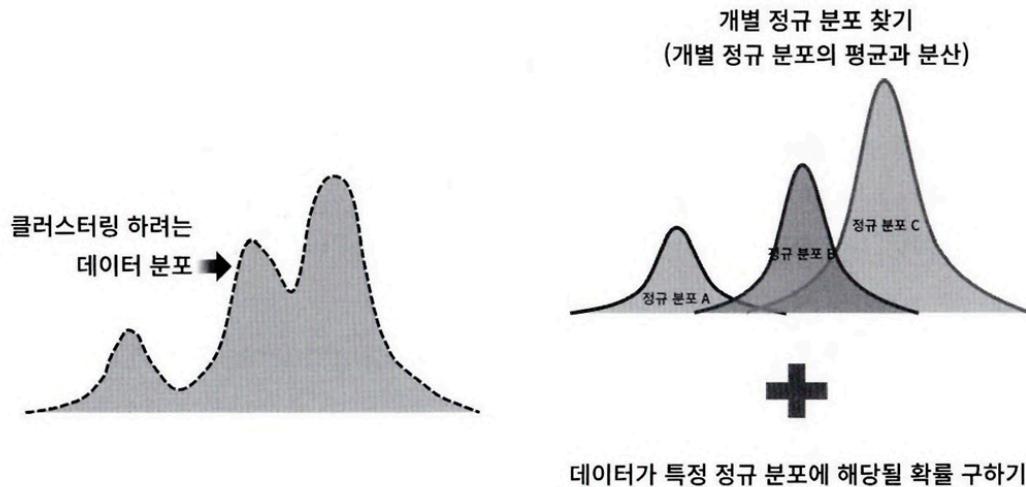


## GMM(Gaussian Mixture Model)

: 군집화를 적용하고자 하는 데이터가 여러 개의 가우시안 분포를 가진 데이터 집합들이 섞여서 생성된 것 이라는 가정하에 군집화를 수행하는 방식



: 서로 다른 정규 분포에 기반해 군집화를 수행하는 것



: 모수 추정

- 개별 정규 분포의 평균과 분산
- 각 데이터가 어떤 정규 분포에 해당되는지의 확률

을 추정하는 것

## DBSCAN

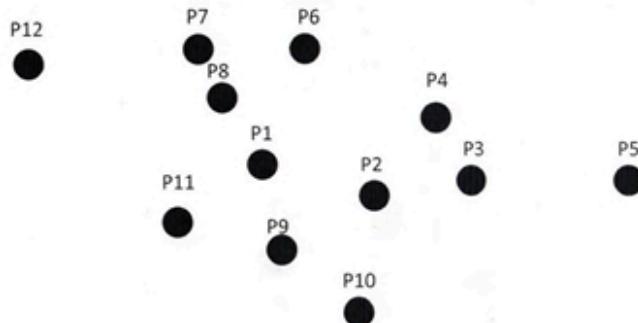
: Density Based Spatial Clustering of Applications with Noise

중요한 파라미터

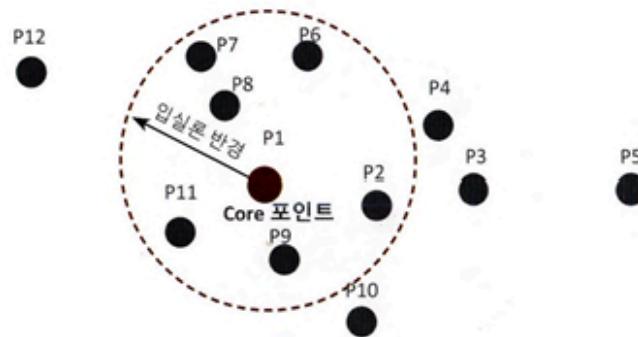
- 입실론 주변 영역 (epsilon) : 개별 데이터를 중심으로 입실론 반경을 가지는 원형의 영역
- 최소 데이터 개수 (min points) : 개별 데이터의 입실론 주변 영역에 포함되는 타 데이터의 개수

- **핵심 포인트(Core Point)**: 주변 영역 내에 최소 데이터 개수 이상의 타 데이터를 가지고 있을 경우 해당 데이터를 핵심 포인트라고 합니다.
- **이웃 포인트(Neighbor Point)**: 주변 영역 내에 위치한 타 데이터를 이웃 포인트라고 합니다.
- **경계 포인트(Border Point)**: 주변 영역 내에 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않지만 핵심 포인트를 이웃 포인트로 가지고 있는 데이터를 경계 포인트라고 합니다.
- **잡음 포인트(Noise Point)**: 최소 데이터 개수 이상의 이웃 포인트를 가지고 있지 않으며, 핵심 포인트도 이웃 포인트로 가지고 있지 않는 데이터를 잡음 포인트라고 합니다.

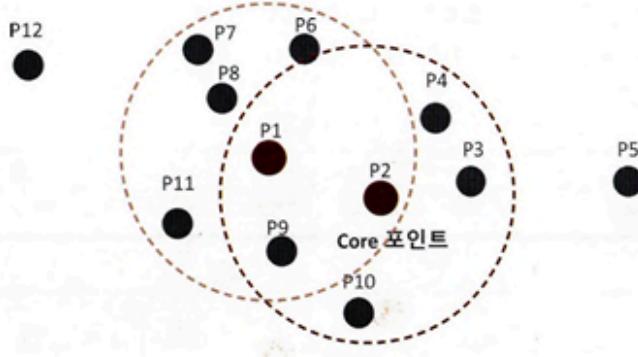
1. 다음 그림과 같이 P1에서 P12까지 12개의 데이터 세트에 대해서 DBSCAN 군집화를 적용하면서 주요 개념을 설명하겠습니다 특정 입실론 반경 내에 포함될 최소 데이터 세트를 6개로(자기 자신의 데이터를 포함) 가정하겠습니다.



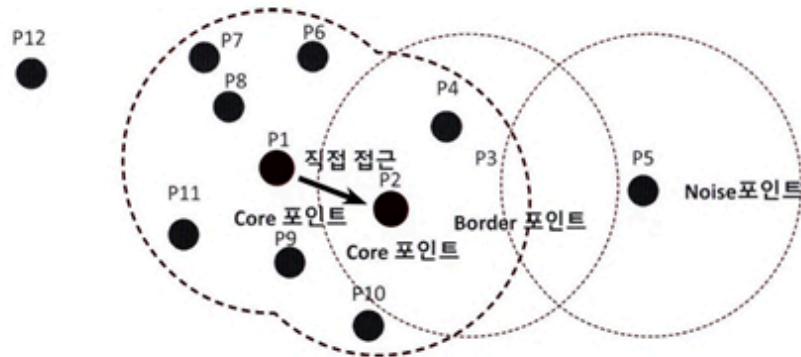
2. P1 데이터를 기준으로 입실론 반경 내에 포함된 데이터가 7개(자신은 P1, 이웃 데이터 P2, P6, P7, P8, P9, P11)로  
최소 데이터 5개 이상을 만족하므로 P1 데이터는 핵심 포인트(Core Point)입니다.



3. 다음으로 P2 데이터 포인트를 살펴보겠습니다. P2 역시 반경 내에 6개의 데이터(자신은 P2, 이웃 데이터 P1, P3, P4, P9, P10)를 가지고 있으므로 핵심 포인트입니다.



7. 다음 그림의 P5와 같이 반경 내에 최소 데이터를 가지고 있지도 않고, 핵심 포인트 또한 이웃 데이터로 가지고 있지 않는 데이터를 잡음 포인트(Noise Point)라고 합니다.



DBSCAN은 이처럼 입실론 주변 영역의 최소 데이터 개수를 포함하는 밀도 기준을 충족시키는 데이터  
간 핵심 포인트를 연결하면서 군집화를 구성하는 방식입니다.

나이키런은 DBSCAN 클래스를 통해 DBSCAN 알고리즘을 지원합니다. DBSCAN 클래스는 다음과  
같은 주요한 초기화 파라미터를 가지고 있습니다.

- eps: 입실론 주변 영역의 반경을 의미합니다.
- min\_samples: 핵심 포인트가 되기 위해 입실론 주변 영역 내에 포함돼야 할 데이터의 최소 개수를 의미합니다(자신의 데이터를 포함합니다. 위에서 설명한 min points + 1).

## 고객 세그먼테이션 (Customer Segmentation)

주요 목표: 타깃 마케팅 - 고객을 여러 특성에 맞게 세분화해서 그 유형에 따라 맞춤형 마켓팅이나 서비스를 제공하는 것

- RFM 기법
  - RECENTY (R) : 가장 최근 상품 구입 일에서 오늘까지의 기간
  - FREQUENCY (F) : 상품 구매 횟수
  - MONETARY VALUE (M) : 총 구매 금액