군집 (Clustering)

Kyungsik Han

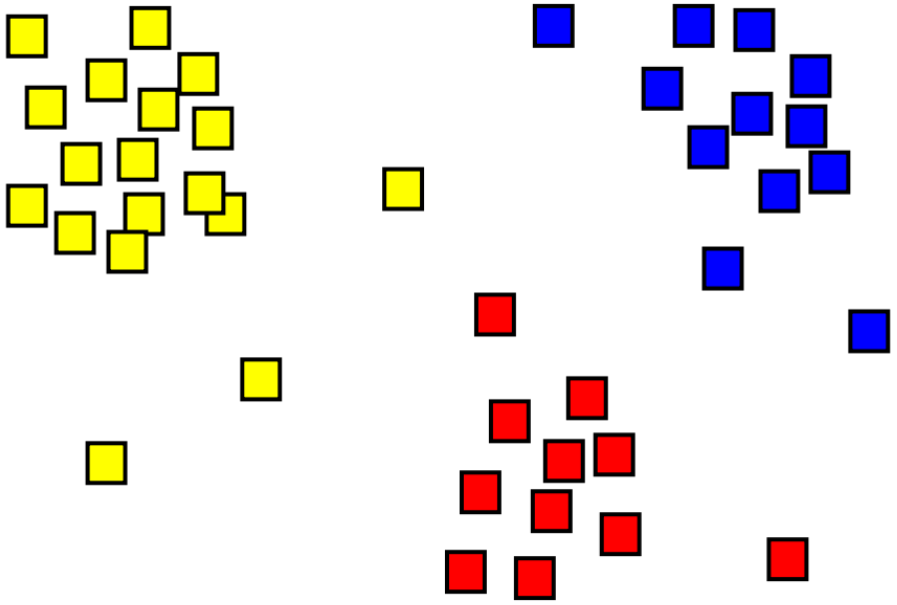
본 영상에서 배울 내용

CODING

- 비지도학습 군집(clustering) 개념, 알고리즘

군집 분석

Goal: Organize similar items into groups



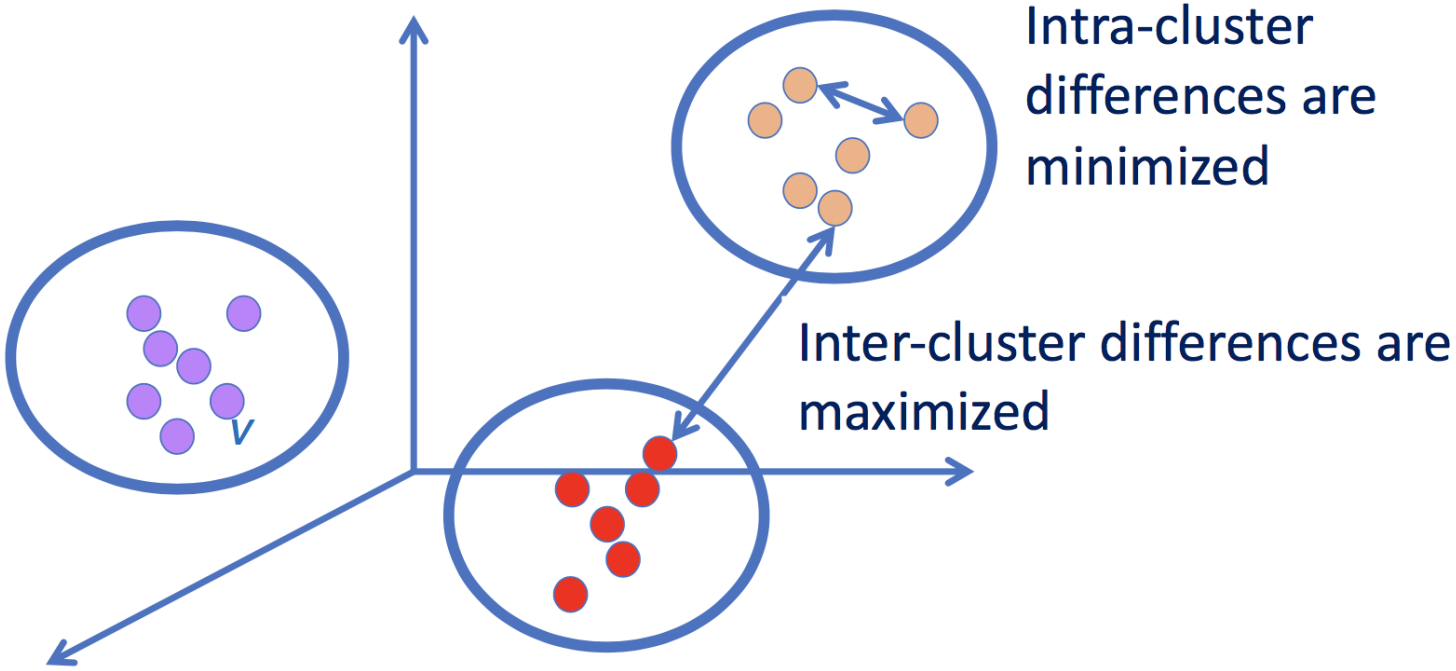
군집 분석 예제

CODING

- 소비자 군집화 (Segment customer base into groups)
- 지역에 따른 날씨 패턴 군집화 (Characterize different weather patterns for a region)
- 뉴스 기사를 토픽별로 군집화 (Group news articles into topics)
- 범죄 지역 군집화 (Discover crime hot spots)

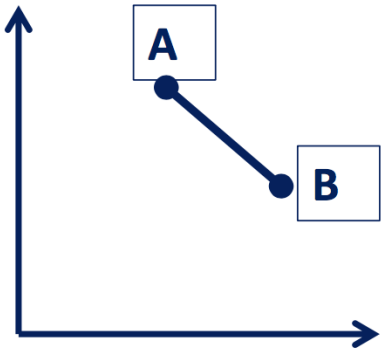
군집 분석

- 데이터를 군집으로 나눔
- 비슷한 데이터는 모임

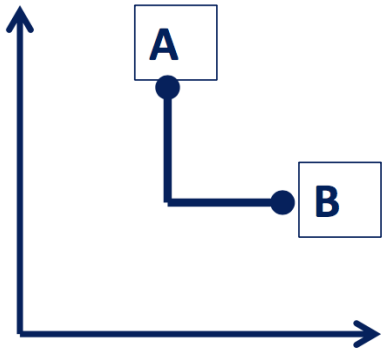


거리, 유사도

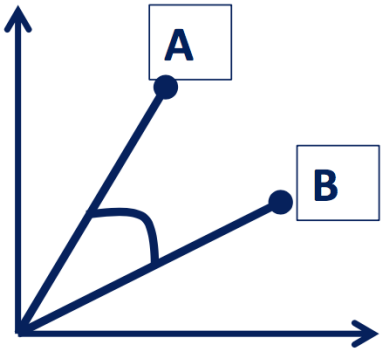
CODING



Euclidean Distance



Manhattan Distance



Cosine Similarity

군집 분석

Unsupervised

There is no ‘correct’ clustering

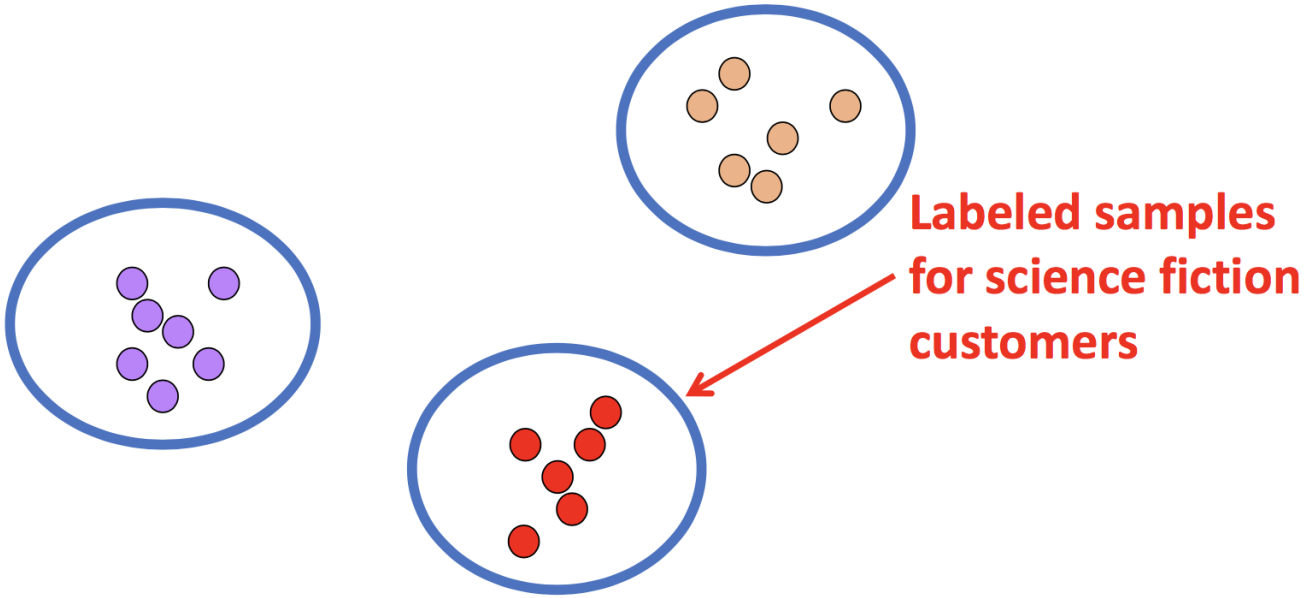
Clusters don’t come with labels

Interpretation and analysis required to make sense of clustering results!

군집 결과 사용

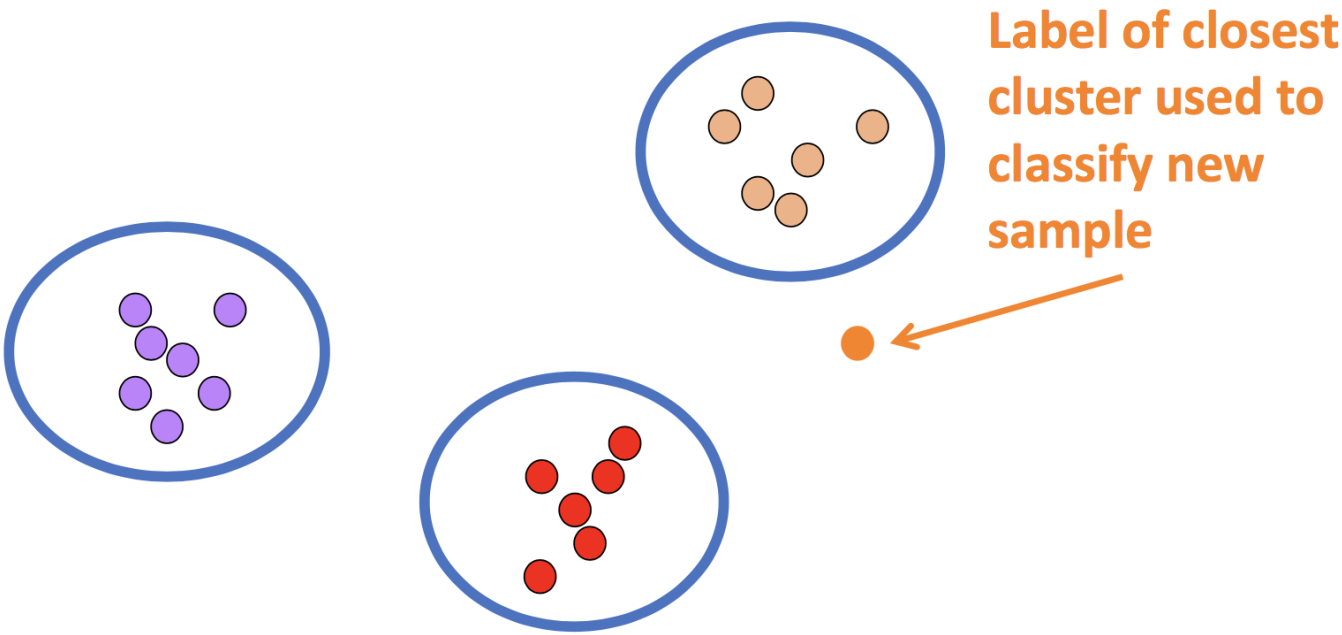
CODING

- 비지도학습으로 군집된 결과를 지도학습의 정답 데이터로도 사용할 수 있음



군집 결과 사용

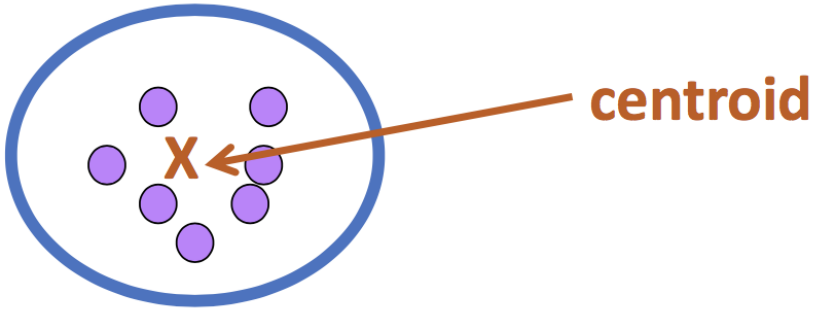
- 새로운 데이터는 어느 군집으로 속하게 되는지?
  - 일반적으로 가까이 존재하는 군집으로 속함



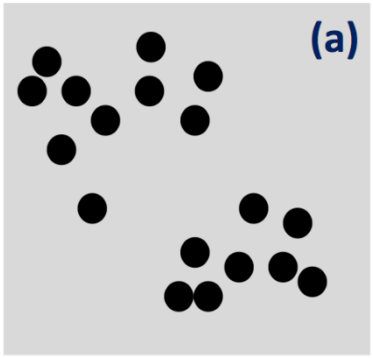
k-Means Clustering

k-Means 군집

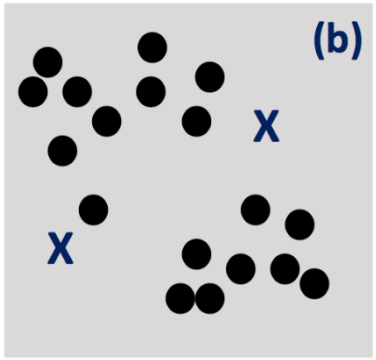
- k 개의 초기 센트로이드 값(initial centroids)을 설정
- 반복 작업
  - 각 샘플을 가까운 센터로 군집화함
  - 새로운 센터를 찾음
- 만족할만한 결과가 나올 때까지, 군집화 반복적 진행



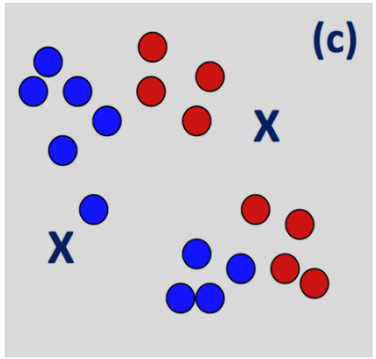
k-Means 군집



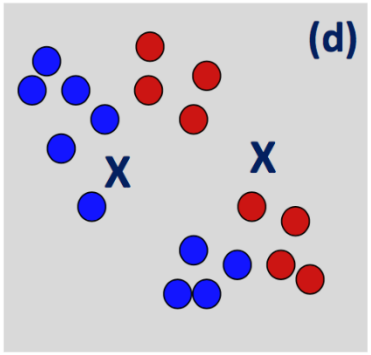
Original samples



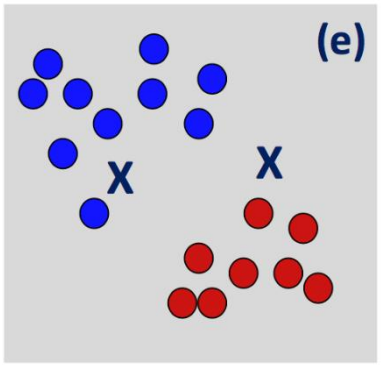
Initial centroids



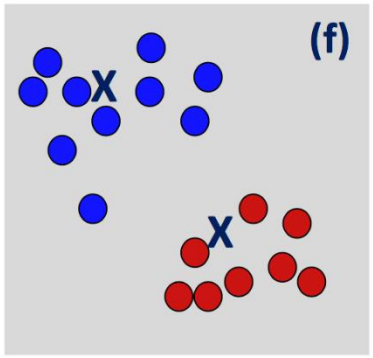
Assign samples



Re-calculate centroids



Assign samples



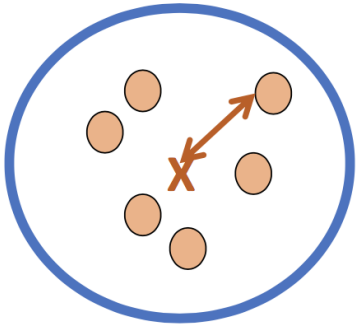
Re-calculate centroids

초기 센터값 사용

CODING

- 이슈
  - 최종적으로 결정되는 클러스터는 초기 센터(센트로이드)값에 의존되는 경우가 많음
- 해결 방안
  - 다른 초기 센터 값을 사용하여 k-means 를 여러 번 수행 후, 가장 최적의 결과를 사용

결과 평가



Error = 샘플과 센트로이드의 거리

Squared Error = $Error^2$

모든 샘플과 센트로이드 사이의 squared error 총 합

모든 클러스터에 대한 합 → **WSSE**

Within-Cluster Sum of Squared Error

WSSE 사용

CODING

$WSSE_1 < WSSE_2$ → WSSE1 is better numerically

하지만 이것이,

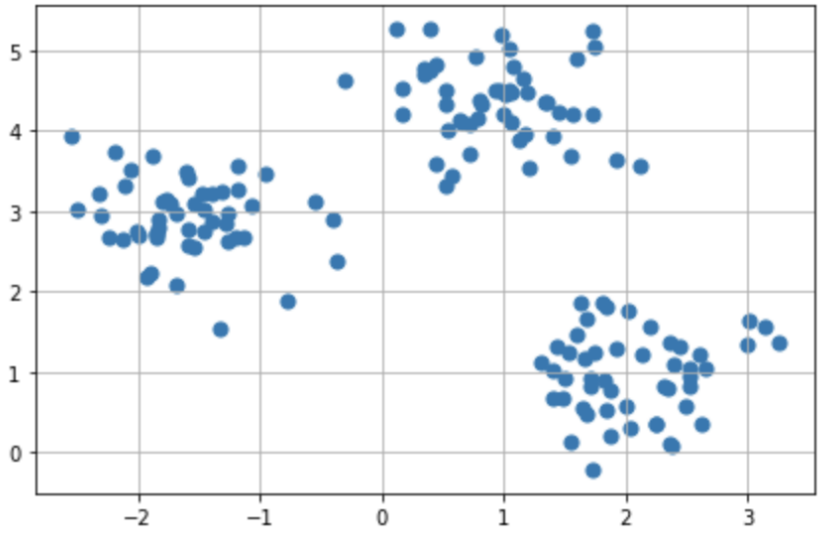
- Cluster set 1 이 Cluster set 2 보다 항상 낫다고 할 수는 없음
- k 값이 크다면 WSSE 값은 작아지는 경향이 있음

k-Means Code

CODING ✖

In [2]:
```
from sklearn.datasets import make_blobs

X, y = make_blobs(n_samples=150,
                  n_features=2,
                  centers=3,
                  cluster_std=0.5,
                  shuffle=True,
                  random_state=0)
```

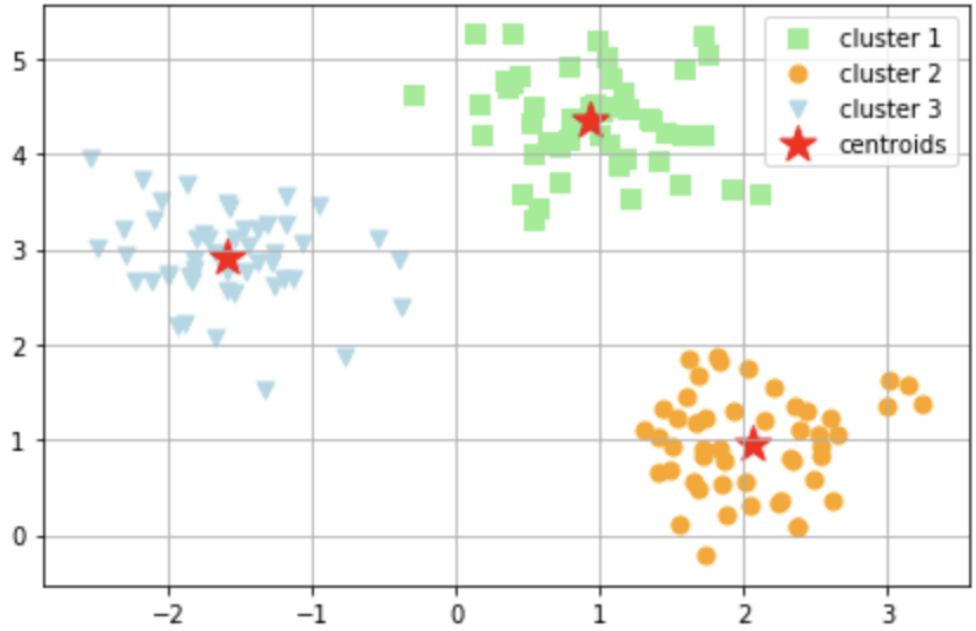
In [10]:
```
import matplotlib.pyplot as plt

plt.scatter(X[:, 0], X[:, 1], marker='o', s=50)
plt.grid()
plt.tight_layout()
plt.show()
```



k-Means Code

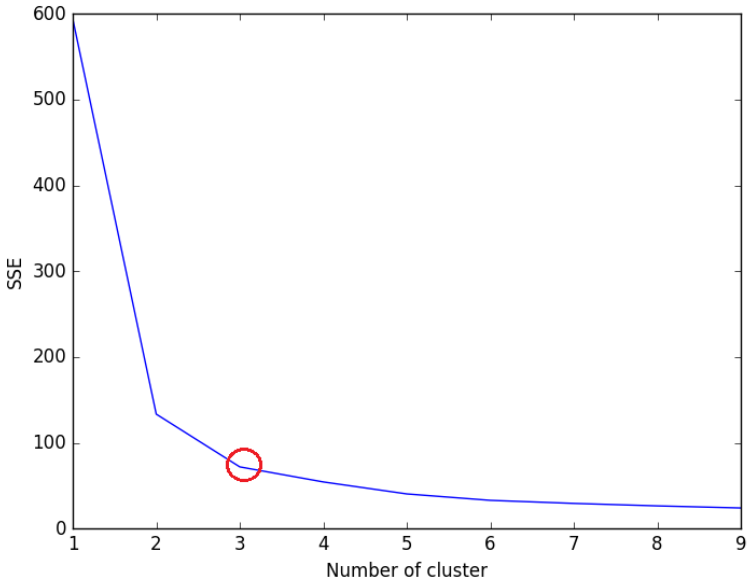
CODING ✖

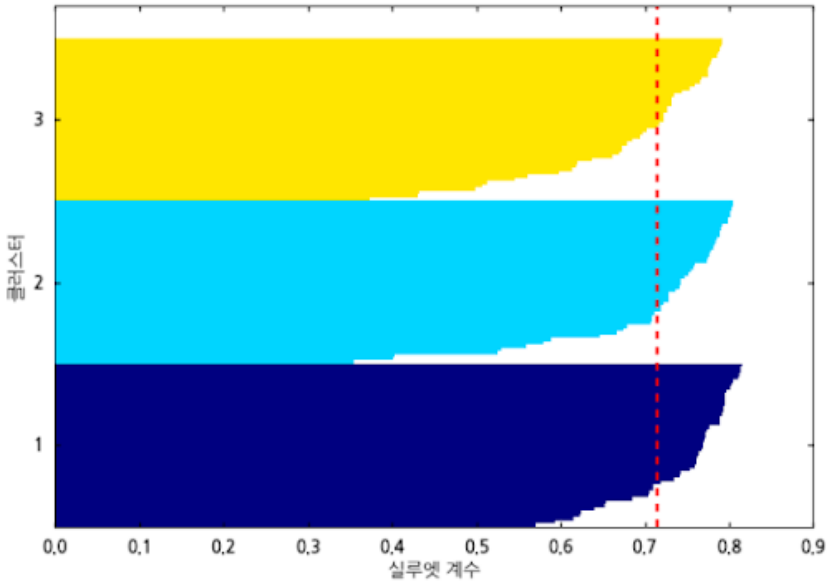
In [12]:
```
from sklearn.cluster import KMeans

km = KMeans(n_clusters=3, # 클러스터 갯수
            init='random',
            n_init=10, # 임의의 센트로이드 10번 독립적인 k-means 알고리즘 실행
            max_iter=300, # 개별 실행을 위한 최대 반복 횟수
            tol=1e-04, # 허용 오차
            random_state=0)
y_km = km.fit_predict(X)
```



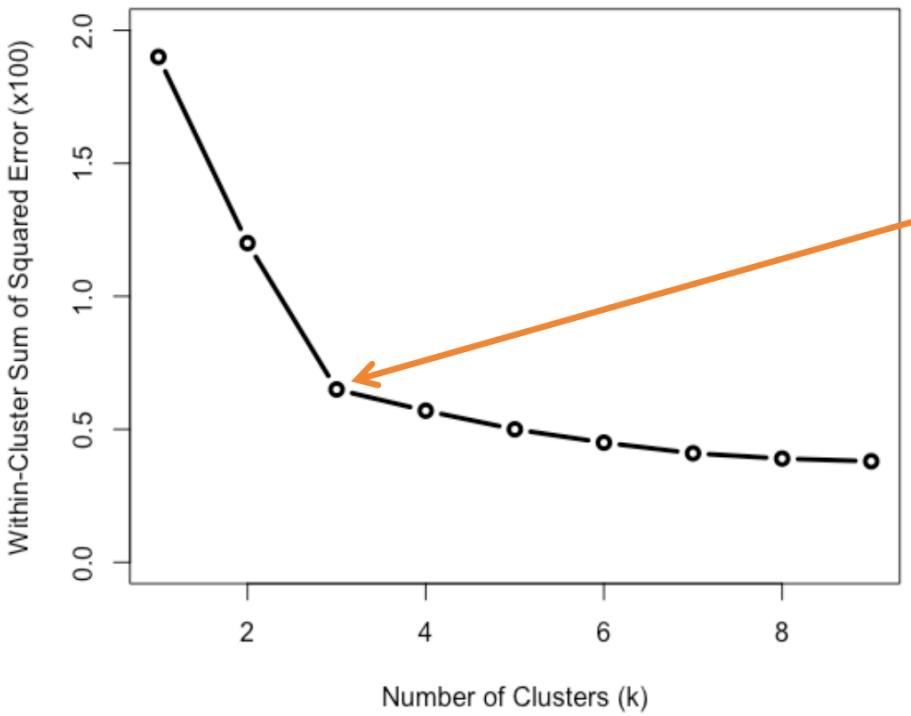
k 값 고르기

- Elbow 기법
- Silhouette 기법





Elbow Method



"Elbow" suggests value for k should be 3

Elbow Method

In [14]:
```
distortions = []
for i in range(1, 11):
    km = KMeans(n_clusters=i,
                init='k-means++',
                n_init=10,
                max_iter=300,
                random_state=0)
    km.fit(X)
    distortions.append(km.inertia_)
plt.plot(range(1, 11), distortions, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Distortion')
plt.tight_layout()
#plt.savefig('./figures/elbow.png', dpi=300)
plt.show()
```

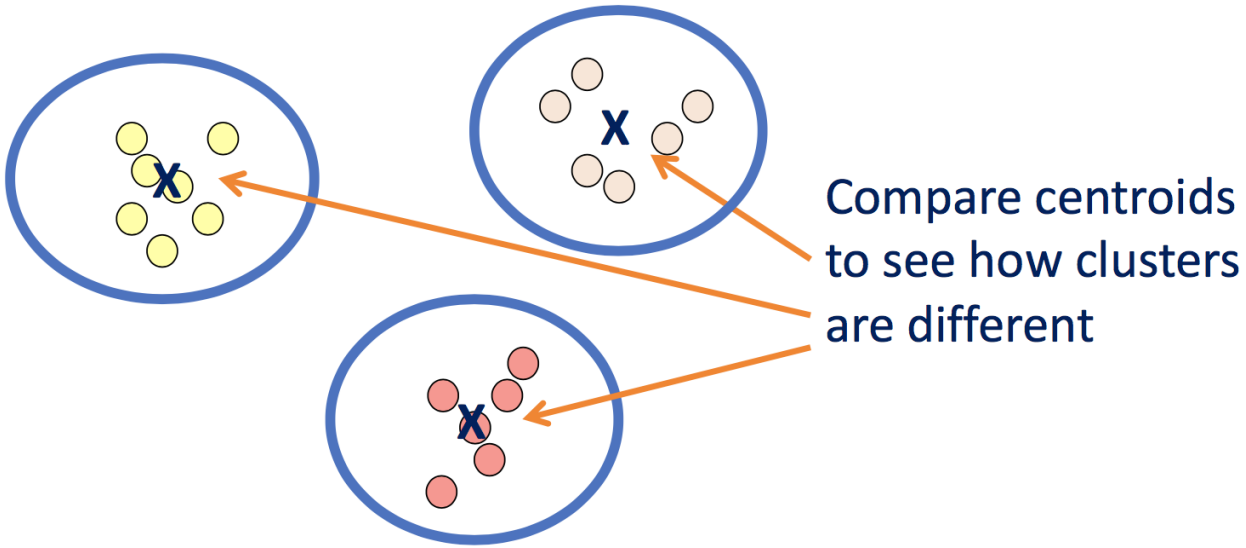
종료 조건

CODING

- 센트로이드 변화가 없을 경우 (No changes to centroids)
- 기준 조건 보다 클러스터 갯수가 더 이상 작아지지 않는 경우 (Number of samples changing clusters is below threshold)

결과 해석

- Examine cluster centroids
  - How are clusters different?



k-Means 요약

CODING

- 클러스터 분석을 위한 전통적인 알고리즘 (Classic algorithm for cluster analysis)
- 쉽게 이해/구현할 수 있고, 효과적임 (Simple to understand and implement and is efficient)
- k 값이 명시 되어야 함 (Value of k must be specified)
- 초기 센트로이드 값에 따라 클러스터 결정이 달라질 수 있음 (Final clusters are sensitive to initial centroids)

다음 영상에서 배울 내용

CODING X

- 비지도학습 군집(clustering) 실습

수고하셨습니다