

---

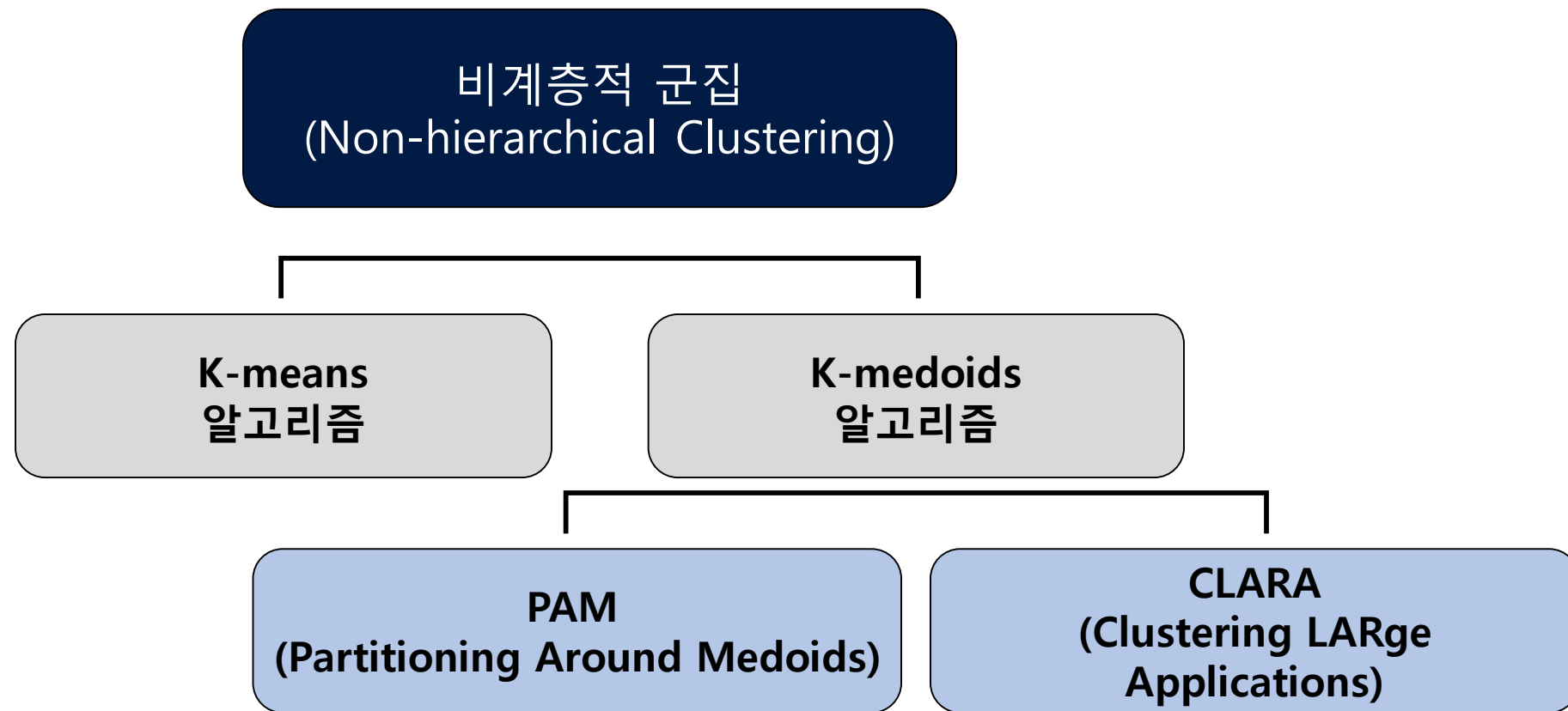
# Wk13-3 : 군집분석

## - 비계층적 군집분석 -

---

# 1. 비계층적 군집분석

- 사전에 군집 수  $k$ 를 정한 후 각 객체를  $k$ 개 중 하나의 군집에 배정



## 2. k-means 군집분석

- k-means 군집분석은 비계층적 군집분석 중 가장 널리 사용

- k개 군집의 중심좌표를 고려하여 각 객체를 가장 가까운 군집에 배정하는 것을 반복

[단계 0] (초기 객체 선정)  
k개 객체 좌표를 초기 군집 중심좌표로 선정.



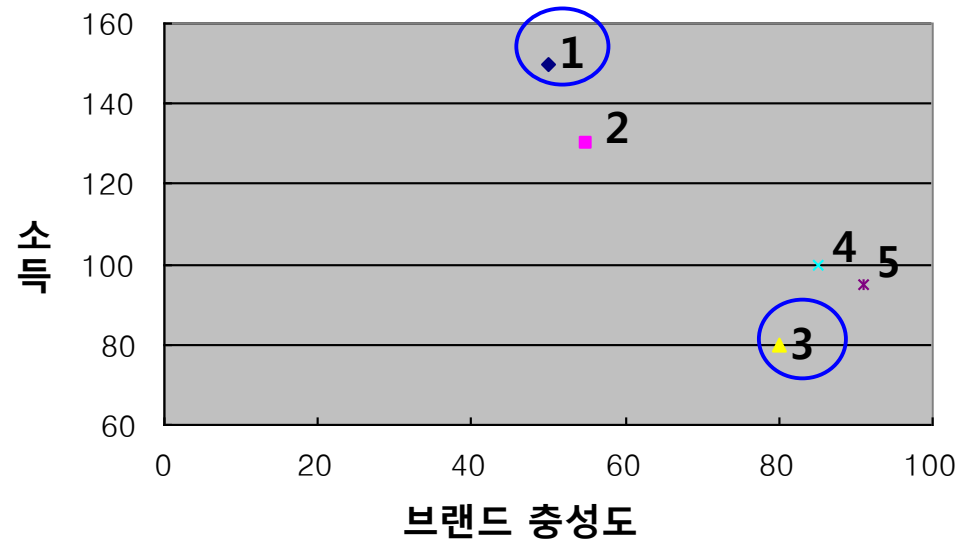
[단계 1] (객체 군집 배정)  
각 객체와 k개 중심좌표와의 거리 산출 후, 가장 가까운 군집에 객체 배정.

[단계 2] (군집 중심좌표 산출)  
새로운 군집의 중심좌표 산출.

[단계 3] (수렴 조건 점검)  
새로 산출된 중심 좌표값과 이전 좌표값을 비교.  
수렴 조건 내에 들면 종료, 그렇지 않으면 단계 1 반복.

# 3. k-means 군집분석 예제

- k-means 알고리즘을 적용 (군집 수  $k=2$ 라 가정)



## Step 0. 초기 객체 선정

- 임의의 두 객체 Obs1, Obs3 선정

## Step 1. 객체 군집 배정

ID	1	3
1	0.0	76.2
2	<u>20.6</u> < 55.9	
3	76.2	0.0
4	61.0 > <u>20.6</u>	
5	68.6 > <u>18.6</u>	

C1
Obs1, Obs2

C2
Obs3, Obs4, Obs5

# 3. k-means 군집분석 예제

- k-means 알고리즘을 적용 (군집 수 k=2라 가정)

## Step 2. 군집 중심좌표 산출

	C1	C2
객체	Obs1, Obs2	Obs3, Obs4, Obs5
중심좌표	$(\frac{50+55}{2}, \frac{150+130}{2})$ = (52.5, 140)	$(\frac{80+85+91}{3}, \frac{80+100+95}{3})$ = (85.33, 91.67)

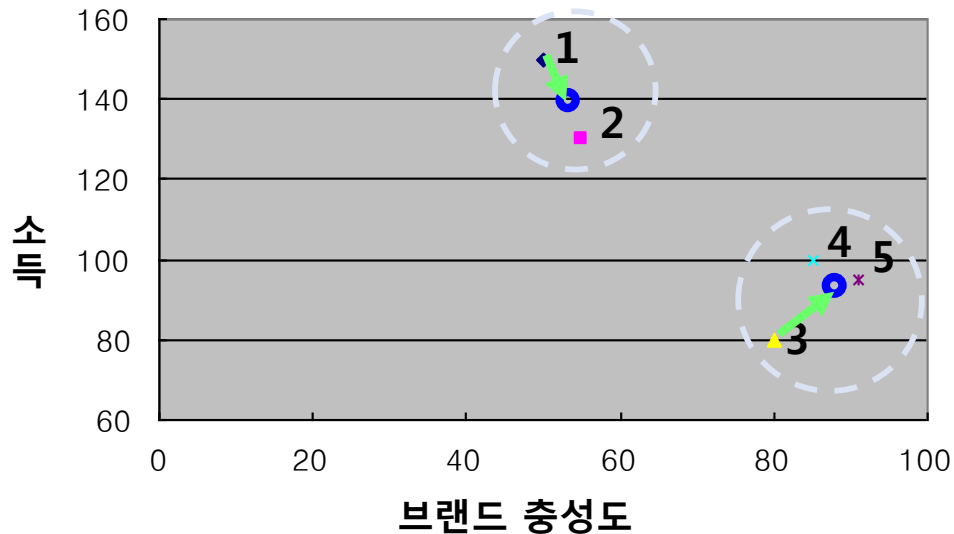
## Step 3. 수렴 조건 점검

ID	C1	C2
1	<u>10.3</u> < 68.2	
2	<u>15.2</u>	48.9
3	66.0	<u>12.8</u>
4	51.5 > <u>8.3</u>	
5	59.2	<u>6.6</u>

C1
Obs1, Obs2

C2
Obs3, Obs4, Obs5

이전 군집결과와 변화 없으므로,  
본 해가 최종 군집해!



# 4. k-means 군집분석

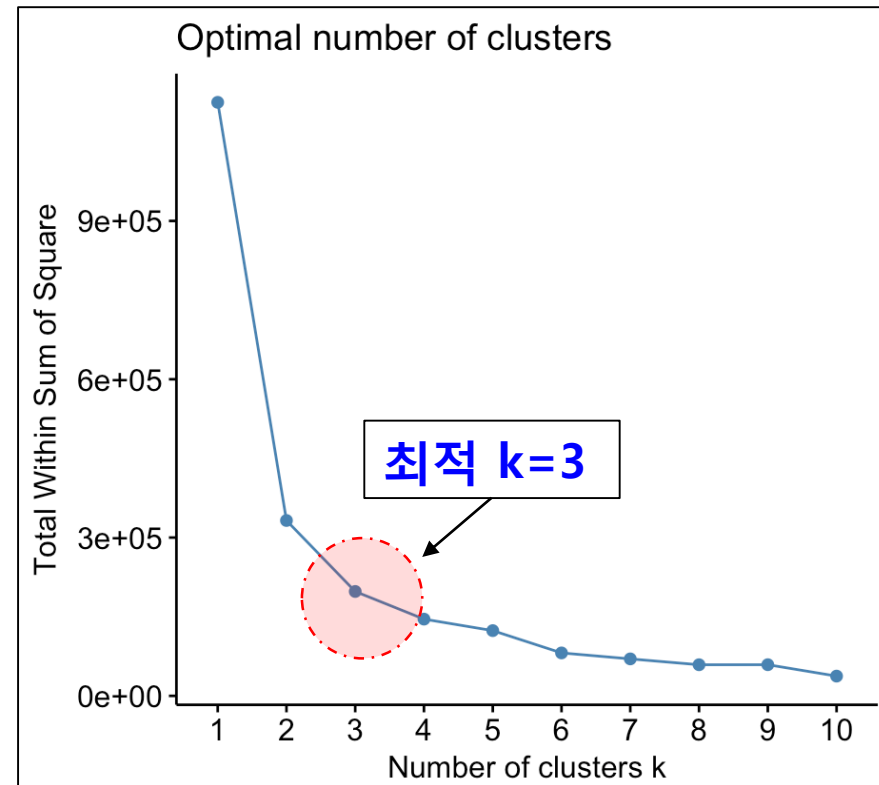
## • 데이터 불러오기 및 군집수 k 결정

```
#lec13_3_clus.R
# Clustering
# Non-hierarchical clustering

# install package & set library
# install.packages("DAAG")
library(DAAG)

# preprocessing
dat1<-wages1833
dat1<-na.omit(dat1)
head(dat1, n=5)

# to choose the optimal k
install.packages("factoextra")
library(factoextra)
fviz_nbclust(dat1, kmeans, method = "wss")
```



- 최적 군집수에 대한 시각화
- 최적값은 "silhouette", "gap\_stat", "wss(그룹내합계제곱)" 으로 산출
- 그래프가 완만해지는 지점을 k의 값으로 추정

# 4. k-means 군집분석

## • k-means (k=3)

```
# compute kmeans
set.seed(123)
km <- kmeans(dat1, 3, nstart = 25)

# visualize
fviz_cluster(km, data = dat1,
  ellipse.type="convex",
  repel = TRUE)
```

*k* (points to 3)  
*random set의 수 (nstart)* (points to 25)

- Kmeans 결과 시각화
- Convex 모양으로 구역 표시
- Repel을 통해 관측치 표기

```
> km
K-means clustering with 3 clusters of sizes 13, 10, 28
```

Cluster means:

	age	mnum	mwage	fnum	fwage
1	16.0	187.2308	96.36154	225.23077	71.0000
2	55.5	6.9000	178.99000	0.00000	0.0000
3	36.5	43.2500	241.73214	31.21429	107.9643



# 5. K-medoids 군집분석

- K-medoids 군집분석은 각 군집의 대표 객체(medoid)를 고려
  - 군집의 대표 객체란, 군집 내 다른 객체들과의 거리가 최소가 되는 객체
  - 즉, K-medoids 군집분석은 객체들을 K개의 군집으로 구분하는데,
  - 객체와 속하는 군집의 대표 객체와의 거리 총합을 최소로 하는 방법
    - **PAM 알고리즘**: 모든 객체에 대하여 대표 객체가 변했을 때 발생하는 거리 총합의 변화를 계산.  
데이터 수가 많아질수록 연산량이 크게 증가함.
    - **CLARA 알고리즘**: 적절한 수의 객체를 샘플링 한 후, PAM 알고리즘을 적용하여 대표 객체 선정.  
샘플링을 여러 번 한 후 가장 좋은 결과를 택함.  
편향된 샘플링은 잘못된 결과값을 도출할 수 있음.

참고문헌: 전치혁, 데이터마이닝 기법과 응용, 2012, 한나래출판사



# 6. PAM (Partitioning Around Medoids) 알고리즘

## • PAM (k=3)

```
# compute PAM  
library("cluster")
```

```
pam_out <- pam(dat1, 3)  
pam_out
```

```
# freq of each cluster  
table(pam_out$clustering)
```

```
# visualize  
fviz_cluster(pam_out, data = dat1,  
              ellipse.type="convex",  
              repel = TRUE)
```

```
> table(pam_out$clustering)
```

```
1  2  3  
13 28 10
```

```
> pam_out <- pam(dat1, 3)  
> pam_out  
Medoids:  
   ID age mnum mwage fnum fwage  
7   7  16  204  83.5  256   72  
31  31  40   38 243.5   15  104  
45  45  54   12 174.0    0    0
```

대표 객체

