

R 기반 의학통계 및 머신러닝

박 승



충북대학교
CHUNGBUK NATIONAL UNIVERSITY

CHAPTER

05

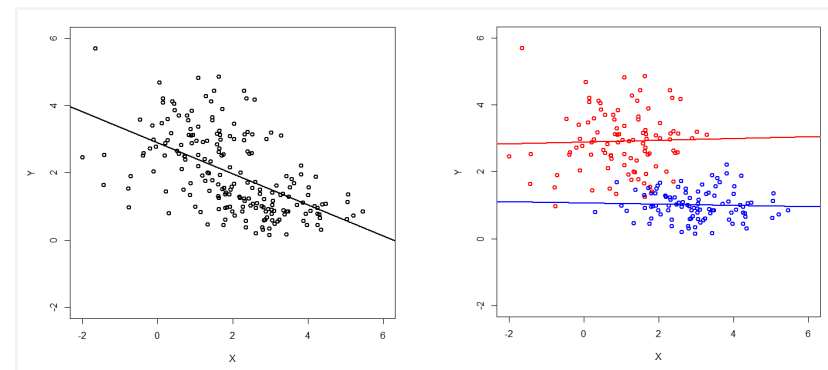
군집화를 통한 환자 그룹 분류

■ 1. 군집화(Clustering)란?

- 학습 시 결과 변수(종속 변수)가 있었던 로지스틱 회귀 분석 같은 지도 학습(Supervised learning)과 달리, 정답(Label) 없이 데이터를 비슷한 그룹으로 나누는 비지도 학습 방법의 하나
(본 강의의 경우 강의를 위해 예시를 들고 임의로 정답을 설정한 후 군집화 결과와 비교할 것)

* 활용 사례

- 환자 유형별 그룹화 (ex : 중증도 기반 환자 클러스터링)
- 질병 유형의 패턴 분석 (ex : 유사한 증상을 가진 환자 그룹화)



데이터 기반 맞춤형 치료 전략을 수립하고, 이상 탐지와 연계해 의료 리소스의 효율적 배분이 목적
절대적인 성능의 기준이 없으며 연구자가 연구 목적에 맞게 군집화 방법을 채택 가능

■ 2. 왜 군집화가 필요한가?

- 의료 데이터는 관련 변수가 많아 매우 복잡하기 때문에, 모든 환자를 한 두개의 기준으로 분류하기 어려움
- 군집화를 통해 비슷한 패턴을 가진 환자들을 자동으로 그룹화 하여 더 나은 의료서비스를 제공 가능

(1) 개별 환자 맞춤형 의료 (Personalized Medicine)

- 환자마다 질병의 원인과 반응이 다르기 때문에, 맞춤형 치료가 필요
- 군집화를 통해 유사한 환자군을 찾아 최적의 치료 전략을 수립

(2) 질병의 세부 유형 (subtypes) 분석

- 동일한 질병이라도 환자별로 증상, 속도, 반응이 다를 수 있음
- 군집화를 통해 기존에 정의되지 않은 세부 유형을 발견할 수 있음



05 군집화를 통한 환자 그룹 분류

■ 군집화 예시

```
set.seed(42)
a1<- rnorm(100, 1,1)
a2<- rnorm(100, 3,1) #데이터 a 생성
a<-cbind(a1,a2)
b1<- rnorm(100, 3,1)
b2<- rnorm(100, 1,0.5) #데이터 b 생성
b<-cbind(b1,b2)

data<-data.frame(rbind(a,b))
colnames(data)<-c("X", "Y")

plot(data, xlim=c(-2, 6), ylim=c(-2, 6), lwd=2) #통합 데이터 산점도
abline(lm(Y~X, data=data), lwd=2) #통합 데이터 회귀 적합선
#이미 그려진 그림에 직선을 추가하는 함수

plot(a, xlim=c(-2, 6), ylim=c(-2, 6), xlab="X", ylab="Y", #데이터 a 산점도
      col="red", lwd=2) #x축 제목 #y축 제목
lines(b, col="blue", type="p", lwd=2) # 또는 points(b, col="blue") #데이터 b 산점도
#이미 그려진 그림에 점이나 선을 추가하는 함수 #선이 아닌 점을 찍는 옵션

abline(lm(a2~a1,data=data.frame(a)), col="red", lwd=2) #각 데이터 회귀 적합선
abline(lm(b2~b1,data=data.frame(b)), col="blue", lwd=2)
```

우리가 데이터프레임의 이름으로 설정한 data와 함수 내부의 옵션의 data항목은 다른 항목.
함수 이름은 덮어쓸 수 있지만 함수 내부의 옵션 항목은 덮어쓸 수 없음.
R의 문법의 7,8번 항목 참조

#점 또는 선의 두께

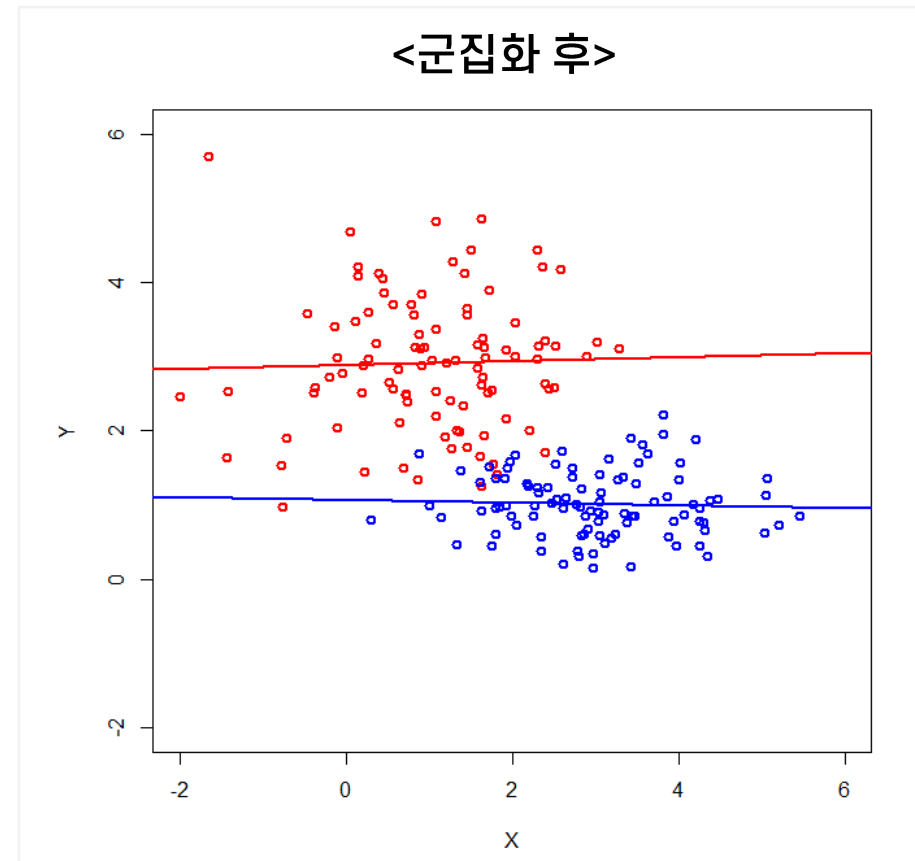
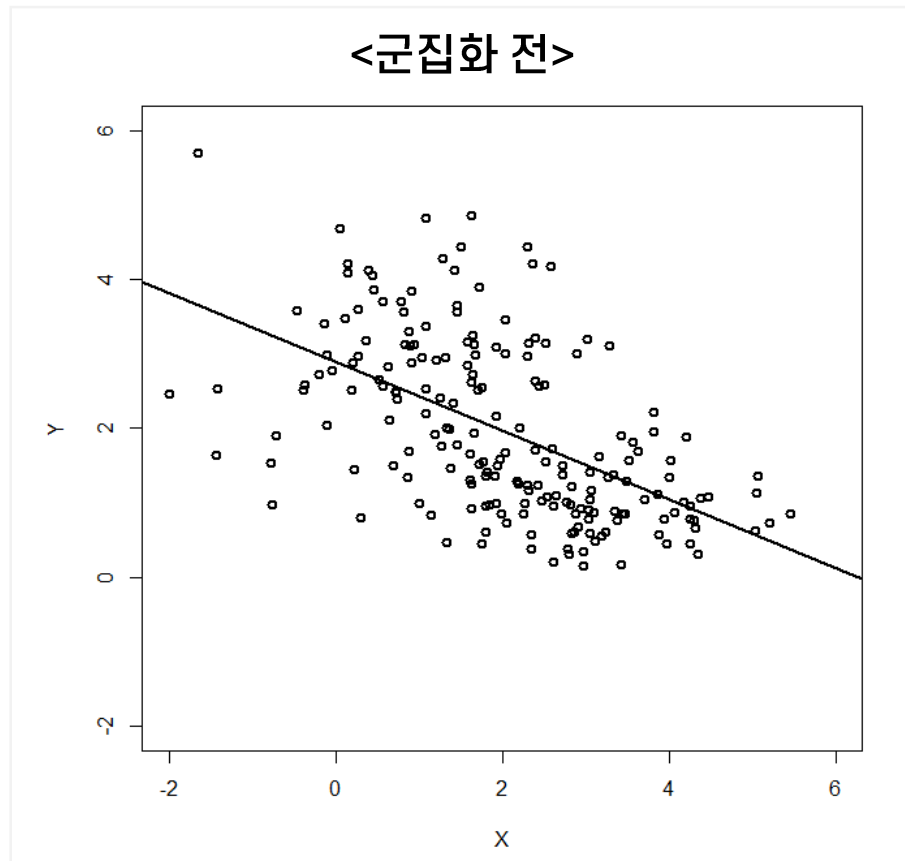
#그림의 x축 길이 #그림의 y축 길이

#이미 그려진 그림에 직선을 추가하는 함수

#선이 아닌 점을 찍는 옵션

05 군집화를 통한 환자 그룹 분류

■ 군집화 예시



****군집화에 따라 접근방식이나 결과의 해석이 상이해지므로 적절한 군집화의 선행이 매우 중요**

■ 3. 데이터 전처리

정규화 : 군집화 알고리즘은 데이터의 스케일에 민감

- 군집화는 유사성을 기반으로 데이터를 분류하는 기법으로, 극단값이 잘못된 유사성 관계 유발 가능
- 단위의 문제 자체는 군집화 성능에 직접적으로 영향을 주지는 않음
- 극단값(outlier)이 존재할 경우에는 스케일링(Scaling), 표준화를 이용한 중심화(Centering)가 도움이 됨

```
data<-data.frame(data)
data$real<-c(rep(1,100),rep(2,100))

data_km <- kmeans(data, centers = 2)
data$kmcluster<-data_km$cluster
data$kmcluster<-ifelse(data$kmcluster==1,2,1)

data_scale_km <- kmeans(scale(data), centers =2)
data$scale_kmcluster<-data_scale_km$cluster
data$scale_kmcluster<-ifelse(data$scale_kmcluster==1,2,1)

table(data$real,data$kmcluster)

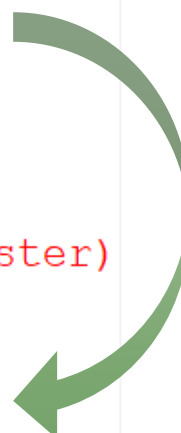
table(data$real,data$scale_kmcluster)
```

```
> table(data$real,
+       data$kmcluster)

      1  2
1  96  4
2   2 98

>
> table(data$real,
+       data$scale_kmcluster)

      1  2
1  97  3
2   2 98
```



■ 4. 군집화 방법

(1) K-means Clustering

- 데이터 포인트를 K개의 군집으로 나누는 알고리즘
- 초기값을 뿌린 뒤, 군집의 중심(centroid)를 찾고, 포인트를 가장 가까운 중심으로 할당하여 갱신
- 중심을 매 step 마다 업데이트하여 최적의 군집을 형성

군집 중심 업데이트

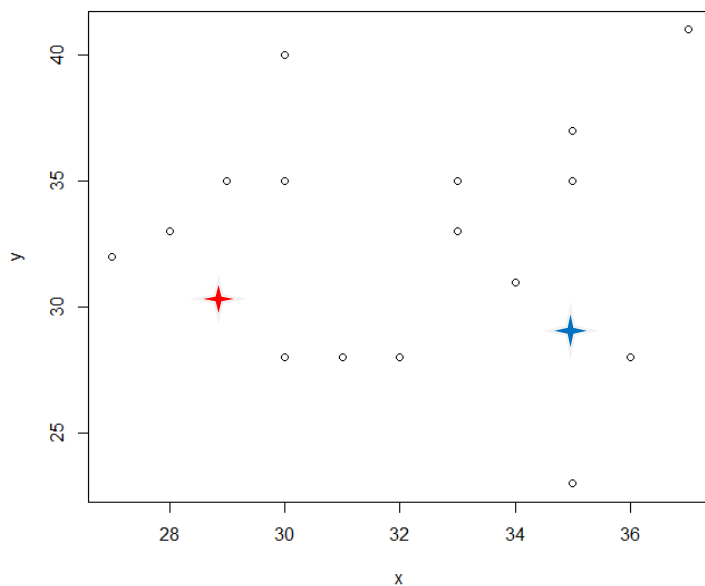
$$C_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$$

군집 내
샘플의 평균

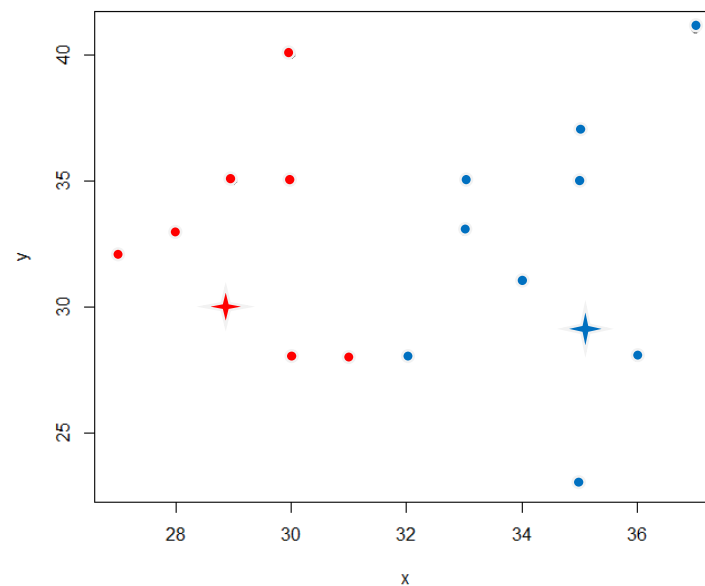
- x_i 는 각 샘플
- S_j 는 군집 j 에 속한 데이터 샘플의 집합
- C_j 는 군집 j 의 중심점
- 군집 내 모든 데이터의 평균(Mean)을 중심으로 업데이트

4. 군집화 방법

(1) K-means Clustering (k=2)



Step1) 랜덤하게 중심점을 설정

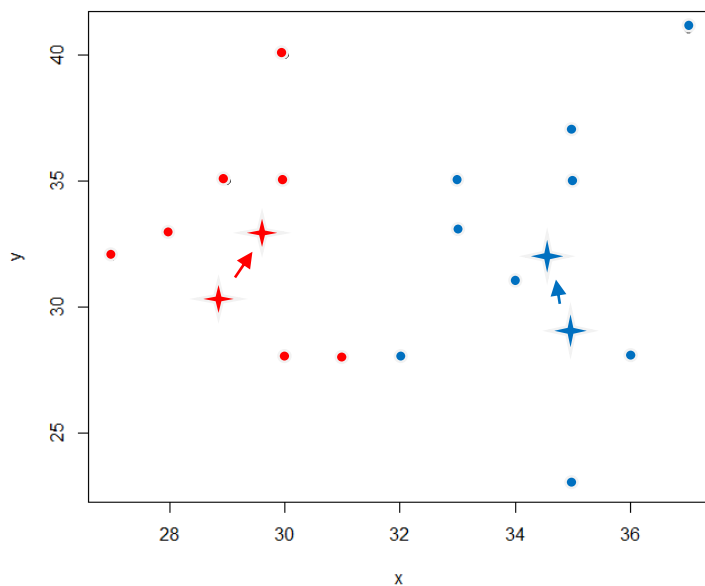


Step2) 각 중심점과의 유클리드 거리를
계산해서 군집 형성

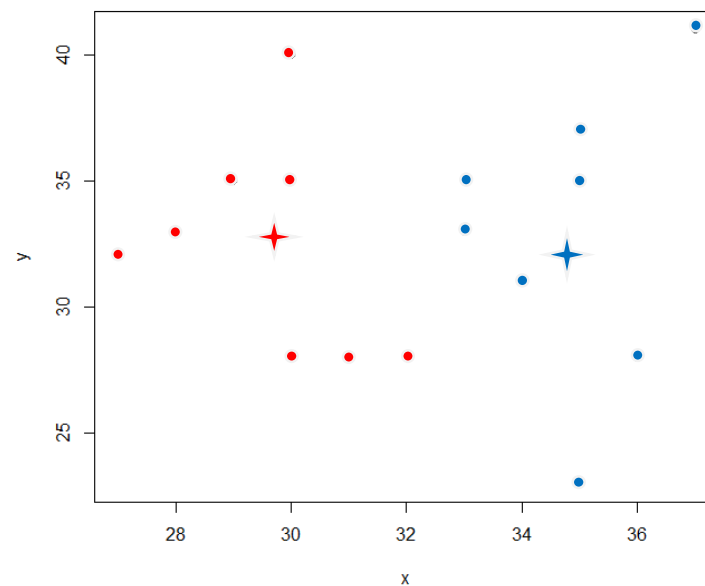
[K-means 알고리즘 영상]

4. 군집화 방법

(1) K-means Clustering (k=2)



Step3) 할당된 군집에 따라 중심을 다시 계산

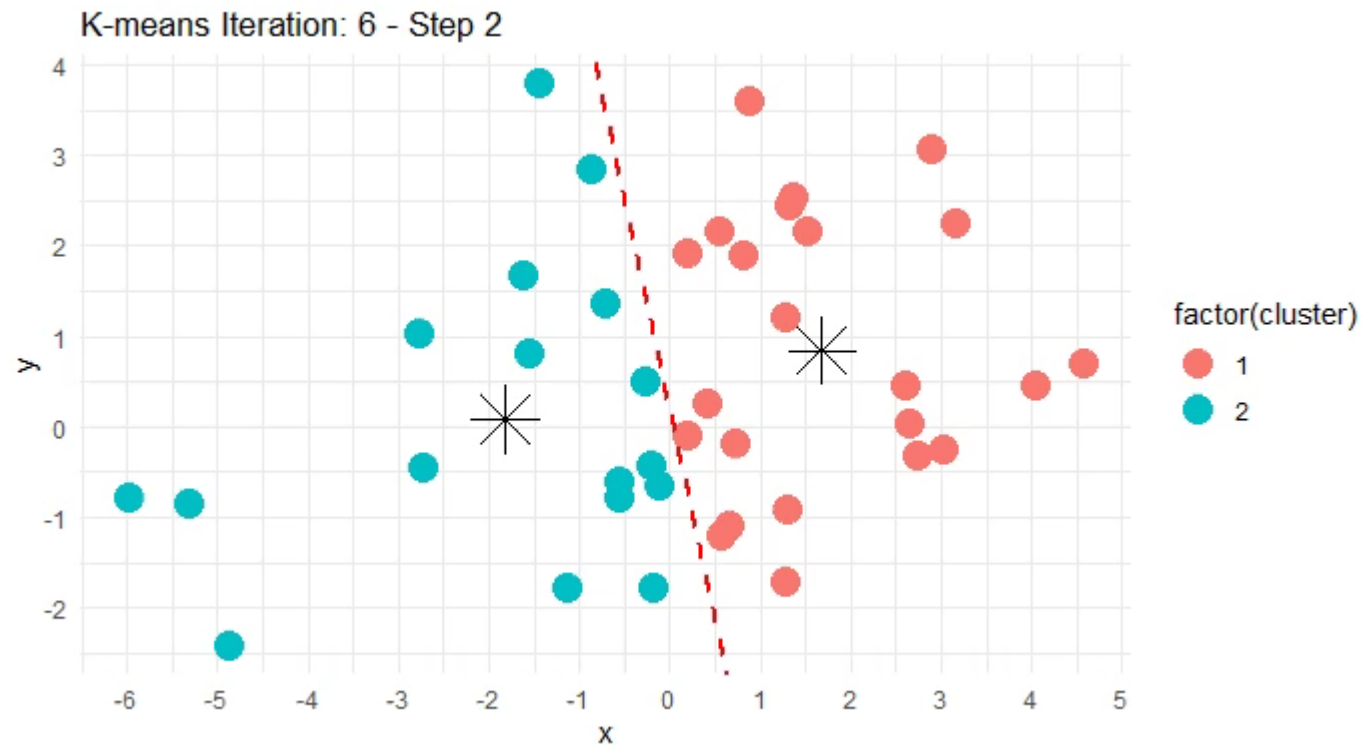


Step4) 다시 계산된 중심점에 따라 군집을 다시 할당

****Step3~4를 수렴할 때 까지 반복**

■ 4. 군집화 방법

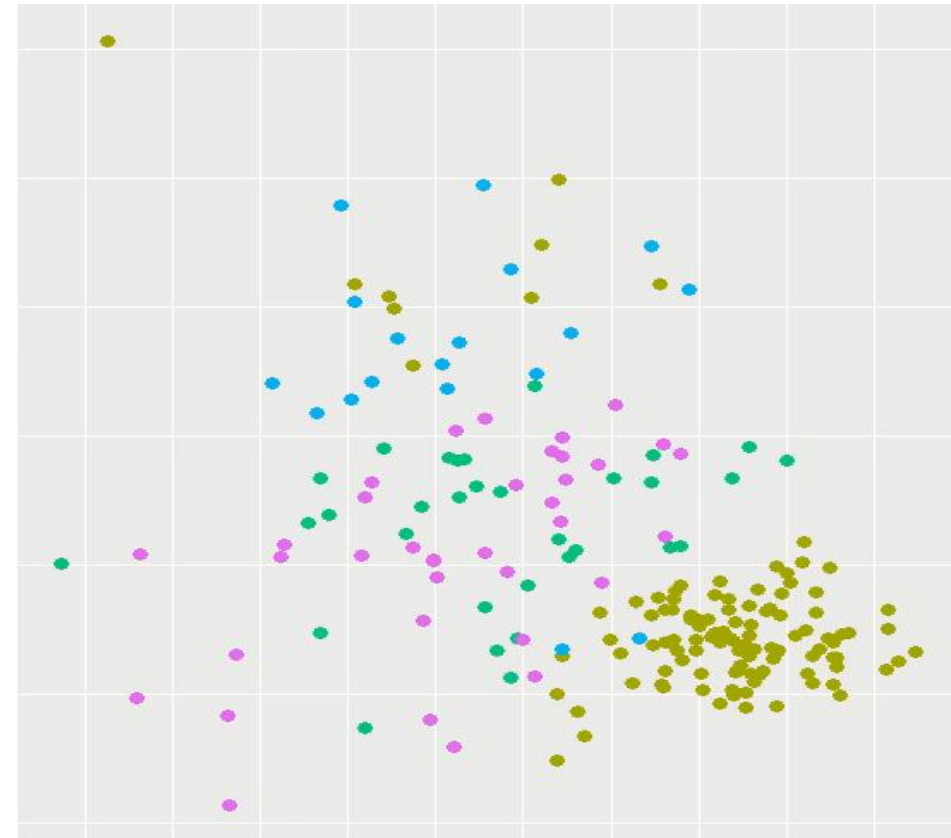
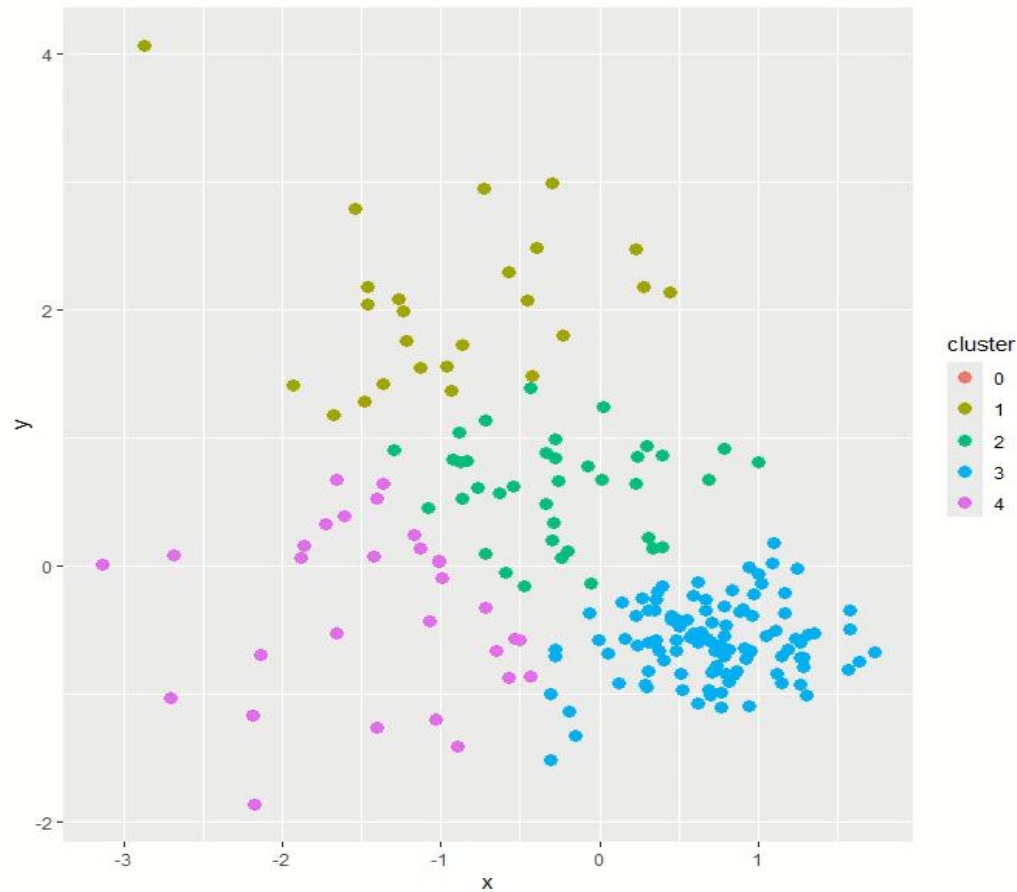
(1) K-means Clustering (k=2)



[K-means 알고리즘 영상]

4. 군집화 방법

(1) K-means Clustering(영상)



■ 4. 군집화 방법

(1) K-means Clustering

```
data<-data.frame(data)  
data$real<-c(rep(1,100),rep(2,100))
```

```
data_km <- kmeans(data, centers = 2)  
data$kmcluster<-data_km$cluster  
data$kmcluster<-ifelse(data$kmcluster==1,2,1)
```

```
data_scale_km <- kmeans(scale(data), centers =2)  
data$scale_kmcluster<-data_scale_km$cluster  
data$scale_kmcluster<-ifelse(data$scale_kmcluster==1,2,1)
```

```
table(data$real,data$kmcluster)
```

```
table(data$real,data$scale_kmcluster)
```

```
> table(data$real,  
+       data$scale_kmcluster)
```

	1	2
1	97	3
2	2	98



■ 4. 군집화 방법

(2) 계층적 군집분석

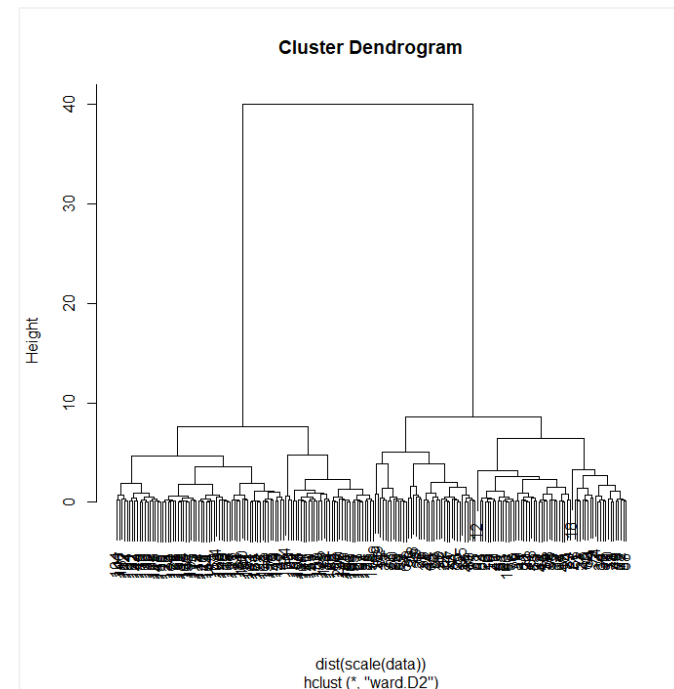
- 각 데이터 간 거리 기반으로 계층적 구조를 형성
- 덴드로그램(Dendrogram)으로 시각화
- 각 데이터를 개별 클러스터로 시작하여, 가장 가까운 두개의 클러스터를 병합
이 과정을 하나의 클러스터가 남을 때까지 반복

군집 간 거리 계산 방법

- 최단 연결, 최장 연결, 평균 연결 등의 방법이 있으나 Ward's 방법이 널리 쓰임
- Ward's method : 클러스터 내 분산을 최소화하도록 거리 계산
- 군집 C_i 와 C_j 가 있을 때, 군집간 거리 $d(i, j)$ 정의

$$d(i, j) = \frac{|C_i||C_j|}{|C_i| + |C_j|} \left\| \bar{x}_i - \bar{x}_j \right\|^2$$

여기서, \bar{x}_i 는 군집 C_i 의 중심

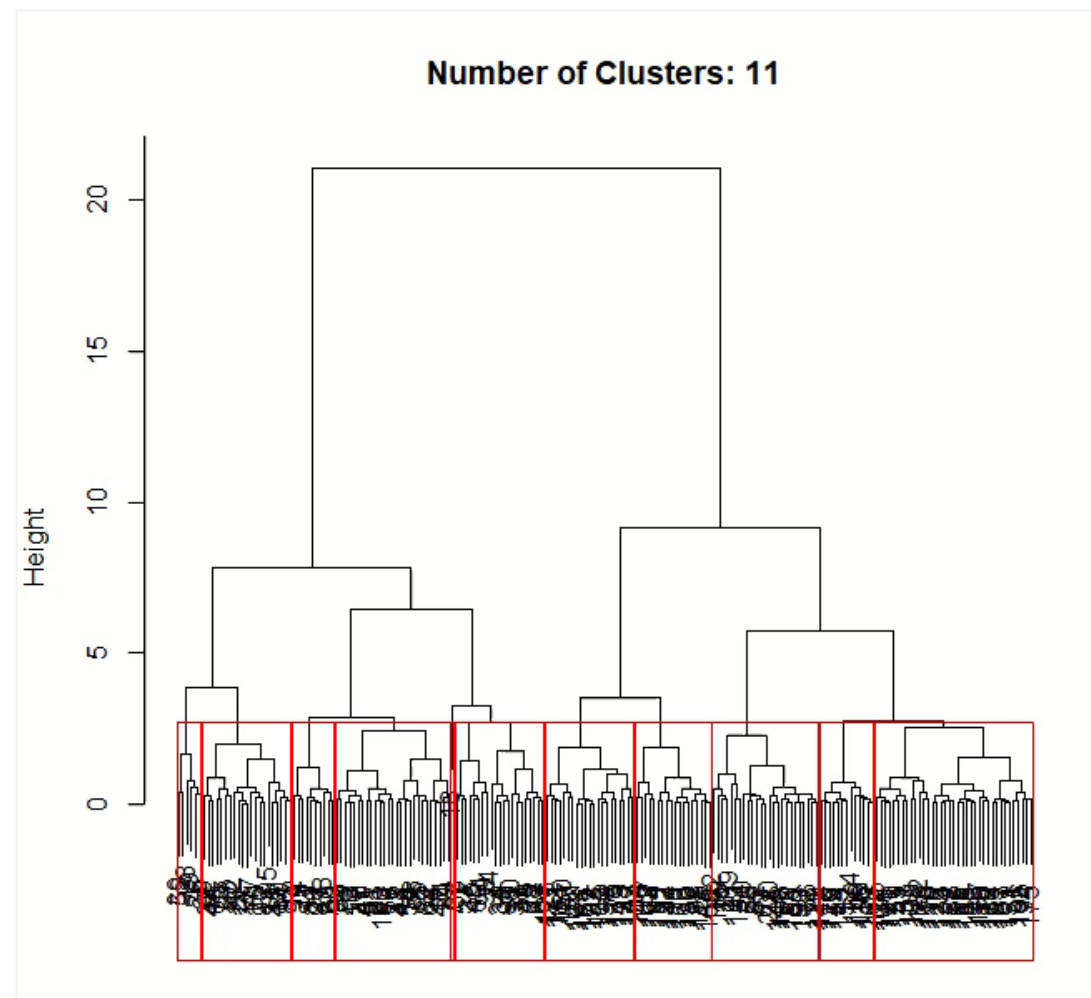


■ 4. 군집화 방법

(2) 계층적 군집분석(영상)

dist() : 개체간 거리를 측정하는 함수

```
hc <- hclust(dist(scale(data)),  
             method = "ward.D2")  
plot(hc)
```



[계층적 군집분석 알고리즘 영상]

■ 4. 군집화 방법

(3) DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

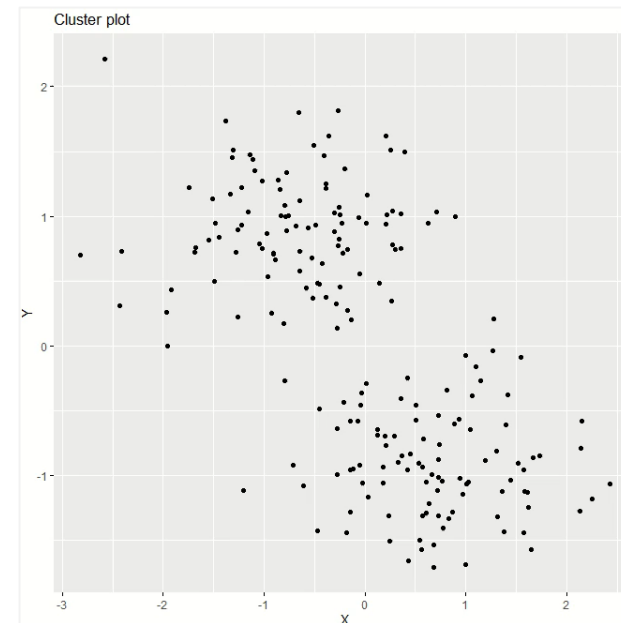
- 밀도(Density) 기반 군집화
- 데이터의 밀집된 영역을 클러스터로 인식하고, 밀도가 낮은 영역을 노이즈(Outlier)로 처리

알고리즘

- 반경 ϵ 내의 데이터 개수를 확인하여 핵심 포인트(core point) 판별
- 핵심 포인트를 기준으로 클러스터 확장, 밀도가 낮은 부분은 노이즈로 분류

$d(x_i, x_j) \leq \epsilon$ 즉, 데이터 x_i 와 x_j 의 거리가 ϵ 이하일때 x_i 가 핵심포인트가 되려면
 $|\{x_j | d(x_i, x_j) \leq \epsilon\}| \geq MinPts$

여기서 $MinPts$ = 최소 데이터 개수



[DBSCAN 알고리즘 영상]

■ 4. 군집화 방법

(3) DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

```
library(dbSCAN)
db <- dbSCAN(scale(data),
              eps = 0.5, minPts = 5)
data$dbcluster <- db$cluster
table(data$real, data$dbcluster)
```

```
fviz_cluster(db, data=scale(data), geom="point")
```

#dbSCAN()이 반환하는 개체에는 원본 데이터가 포함되지 않기 때문에 data옵션이 필요

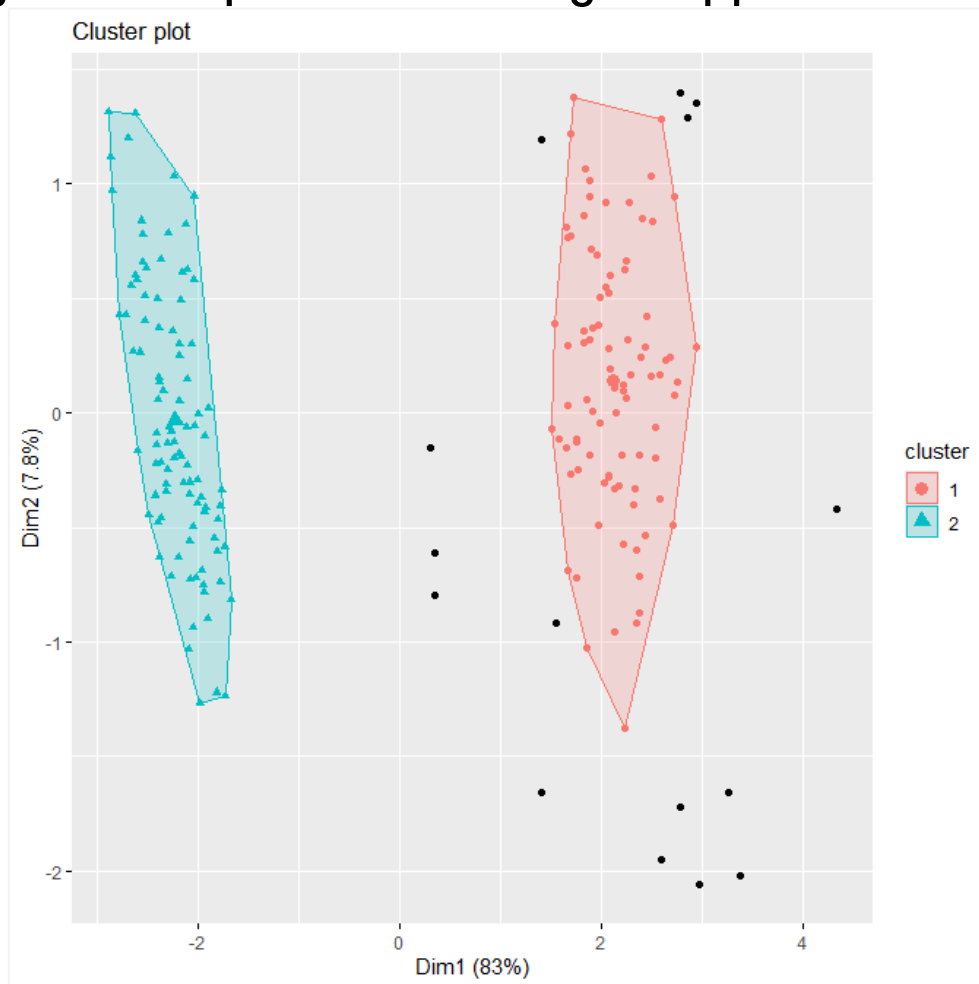
```
> table(data$real, data$dbcluster)
```

	0	1	2
1	13	87	0
2	2	0	98

#노이즈로 판정

■ 4. 군집화 방법

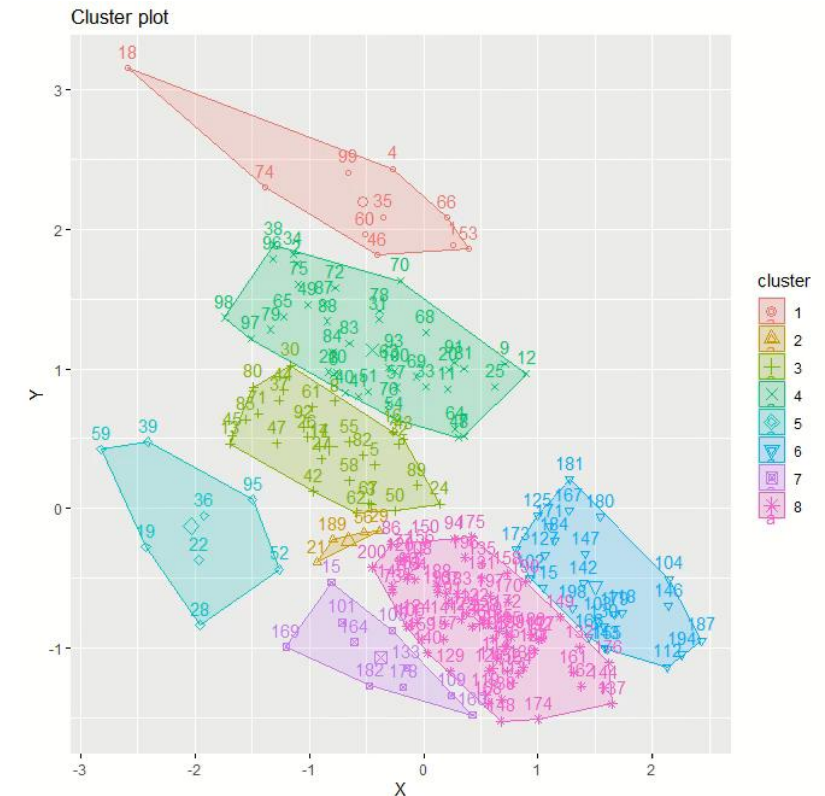
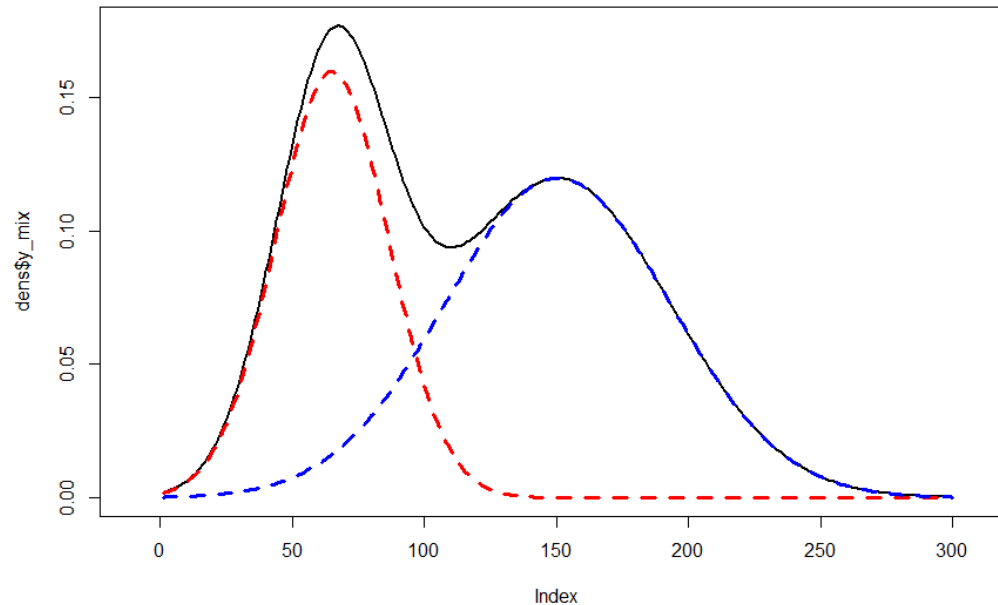
(3) DBSCAN (Density-Based Spatial Clustering of Applications with Noise)



4. 군집화 방법

(4) GMM (Gaussian Mixture Model)

- 데이터를 여러 개의 가우시안분포(정규분포)로 표현하는 확률 모델
- K-means와 달리 확률 기반 군집화로, 데이터가 여러개의 정규분포의 혼합으로 이루어졌다고 가정 각 클러스터에 속할 확률을 계산



[GMM 알고리즘 영상]

■ 4. 군집화 방법

(4) GMM (Gaussian Mixture Model)

EM 알고리즘

- 초기 가우시안 모델 파라미터 (π_k, μ_k, Σ_k) 설정
- E-step(Expectation Step) : 각 데이터가 클러스터 k 에 속할 확률 계산

$$\omega_{ik} = \frac{\pi_j N(x_i | \mu_j, \Sigma_j)}{\sum_{j=1}^K \pi_j N(x_i | \mu_j, \Sigma_j)}$$

- M-step(Maximization Step) : 가우시안 분포의 평균, 분산, 가중치 업데이트

$$\mu_k = \frac{\sum_{i=1}^n \omega_{ik} x_i}{\sum_{i=1}^n \omega_{ik}}, \quad \Sigma_k = \frac{\sum_{i=1}^n \omega_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^n \omega_{ik}}$$

- 수렴할 때 까지 반복

■ 4. 군집화 방법

(4) GMM (Gaussian Mixture Model)

```
library(mclust)
gmm <- Mclust(scale(data), G=2)
data$gmmcluster <- gmm$classification

table(data$real, data$gmmcluster)
```

```
> table(data$real, data$gmmcluster)
```

	1	2
1	100	0
2	2	98

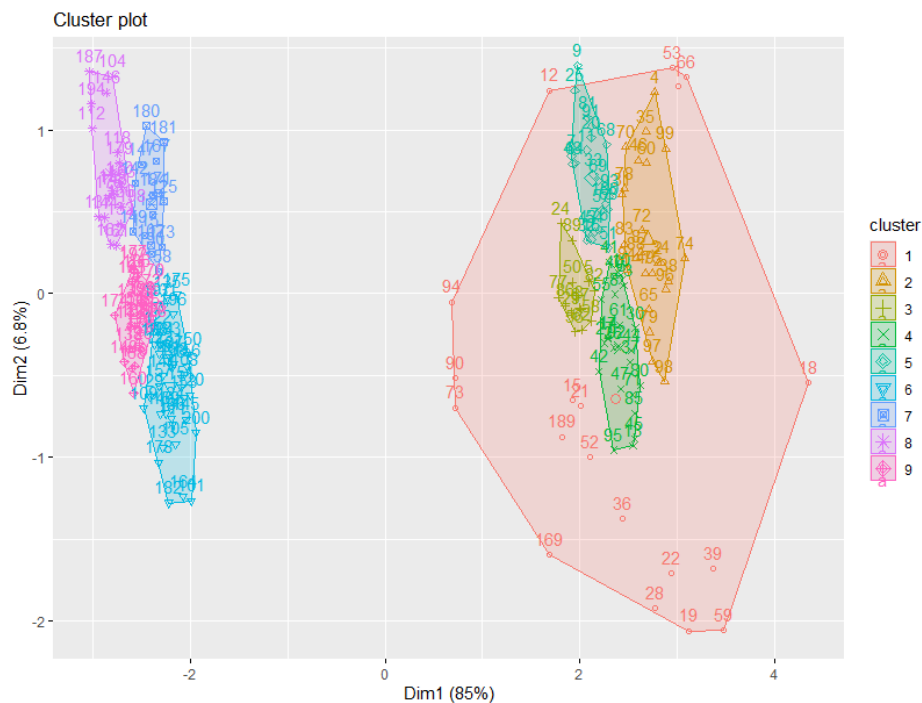
#애초에 정규분포 기반으로 생성된 자료이므로 GMM 방법의 성능이 뛰어남

05 군집화를 통한 환자 그룹 분류

4. 군집화 방법

(4) GMM (Gaussian Mixture Model)

```
fviz_cluster(Mclust(scale(data)))
```



```
fviz_cluster(Mclust(scale(data), G=2))
```



■ 4. 군집화 방법

(5) 실습

이전에 사용했던 데이터 불러오기

```
data<-read.csv("heart_disease_uci.csv")
data$target<-ifelse(data$num>0,1,0)
data2<-filter(data, if_all(everything(), ~!is.na(.) & .!=""))
data2sub<-subset(data2, select=c(-id, -num))
colnames(data2sub)
```

군집화 진행하기

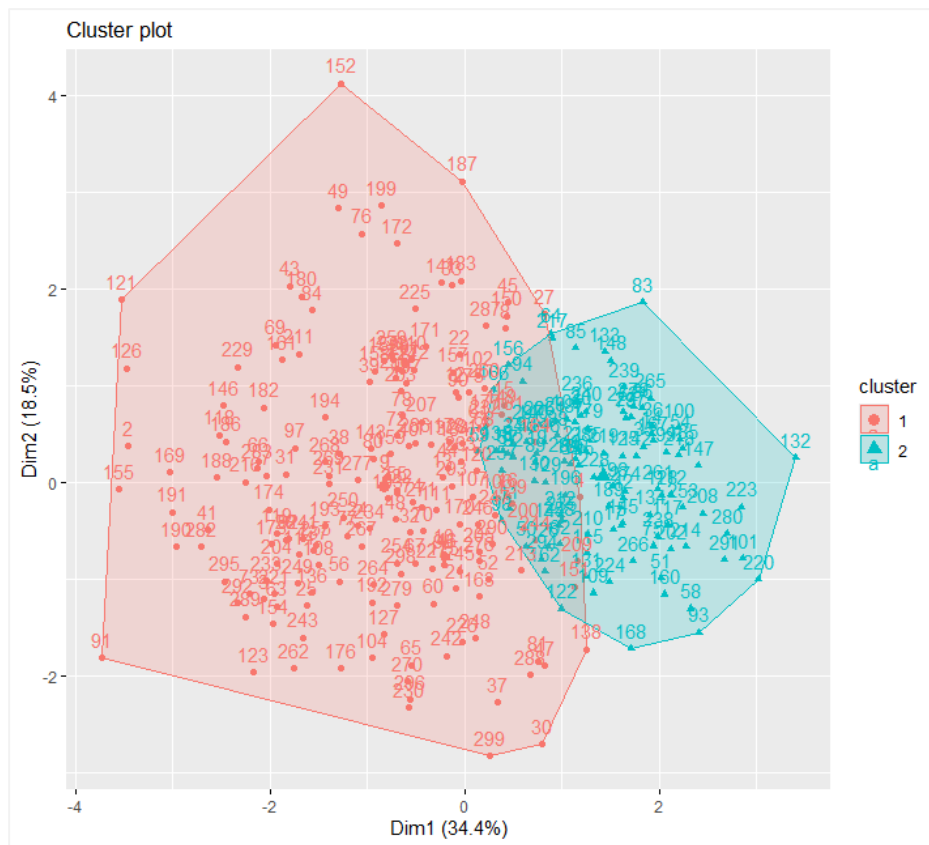
```
str(data2sub)
data2sub2<-subset(data2sub, select=c(
    age, trestbps, chol, thalch, oldpeak, ca))#연속형 변수에 대해서 군집화 진행
fviz_cluster(Mclust(data2sub2, G=2))

Mclust(data2sub, G=2)$classification
data2sub$class<-Mclust(data2sub2, G=2)$classification #군집 변수를 데이터에 포함

model3<-glm(target ~ ., family="binomial", data=data2sub) #모형 적합
summary(step(model3))
```


4. 군집화 방법

(5) 실습



Call:

```
glm(formula = target ~ sex + cp + trestbps + fbs + exang + slope +
    ca + thal + class, family = "binomial", data = data2sub)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-2.33571	2.08454	-1.120	0.262503
sexMale	1.64807	0.49840	3.307	0.000944 ***
cpatypical angina	-0.88887	0.56700	-1.568	0.116960
cpnon-anginal	-1.90525	0.49426	-3.855	0.000116 ***
cptypical angina	-2.27332	0.66596	-3.414	0.000641 ***
trestbps	0.01825	0.01078	1.693	0.090510 .
fbsTRUE	-0.81672	0.57588	-1.418	0.156128
exangTRUE	0.76872	0.43418	1.770	0.076644 .
slopeflat	0.30836	0.75574	0.408	0.683256
slopeupsloping	-1.20329	0.76755	-1.568	0.116952
ca	1.12810	0.26868	4.199	2.68e-05 ***
thalnormal	0.05075	0.76798	0.066	0.947314
thalreversible defect	1.44217	0.75692	1.905	0.056738 .
class	-1.13945	0.47048	-2.422	0.015439 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 413.03 on 298 degrees of freedom
 Residual deviance: 194.65 on 285 degrees of freedom
 AIC: 222.65

■ 4. 군집화 방법

(5) 실습

군집 별 회귀분석 시행

```
model3_1<-glm(target ~ .-class , family="binomial",  
               data=filter(data2sub,data2sub$class==1))  
summary(step(model3_1))
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-5.40649	2.11750	-2.553	0.010672	*
sexMale	2.00535	0.53148	3.773	0.000161	***
cpatypical angina	-0.72028	0.82113	-0.877	0.380391	
cpnon-anginal	-1.61051	0.60029	-2.683	0.007299	**
cptypical angina	-2.54706	0.73407	-3.470	0.000521	***
trestbps	0.03313	0.01282	2.583	0.009796	**
fbTRUE	-0.94379	0.59087	-1.597	0.110205	
exangTRUE	1.37525	0.54375	2.529	0.011433	*
slopeflat	0.56922	0.90696	0.628	0.530257	
slopeupsloping	-1.39978	0.94232	-1.485	0.137422	
ca	1.08635	0.27835	3.903	9.51e-05	***

■ 4. 군집화 방법

(5) 실습

군집 별 회귀분석 시행(오류 발생)

```
model3_2<-glm(target ~ .-class , family="binomial",  
               data=filter(data2sub,data2sub$class==2))
```

``contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]])`에서 다음과 같은 에러가 발생했습니다:
contrasts는 오로지 2 또는 그 이상의 level들을 가진 요인들에만 적용할 수 있습니다

filter 함수로 인해 일부 케이스가 제외되면서, 범주가 하나만 남는 변수가 생기는 경우 해당 오류 발생함

■ 4. 군집화 방법

(5) 실습

데이터 재탐색

```
filter(data2sub, data2sub$class==2) %>% str #파이프함수
filter(data2sub, data2sub$class==2) %>% summary
```

```
'data.frame': 110 obs. of 16 variables:
 $ age      : int  41 56 57 44 48 54 48 49 50 43 ...
 $ sex      : chr   "Female" "Male" "Male" "Male" ...
 $ dataset  : chr   "Cleveland" "Cleveland" "Cleveland" "Cleveland" ...
 $ cp       : chr   "atypical angina" "atypical angina" "asymptomatic" "atypical angina" ...
 $ trestbps : int   130 120 140 120 110 140 130 130 120 150 ...
 $ chol     : int   204 236 192 263 229 239 275 266 219 247 ...
 $ fbs      : logi   FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ restecg  : chr   "lv hypertrophy" "normal" "normal" "normal" ...
 $ thalch   : int   172 178 148 173 168 160 139 171 158 171 ...
 $ exang     : logi   FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ oldpeak  : num    1.4 0.8 0.4 0 1 1.2 0.2 0.6 1.6 1.5 ...
 $ slope    : chr   "upsloping" "upsloping" "flat" "upsloping" ...
 $ ca       : int    0 0 0 0 0 0 0 0 0 0 ...
 $ thal     : chr   "normal" "normal" "fixed defect" "reversable defect" ...
 $ target   : num    0 0 0 0 1 0 0 0 0 0 ...
 $ class    : num    2 2 2 2 2 2 2 2 2 2 ...
```

#단일범주가
의심되는 변수들

■ 4. 군집화 방법

(5) 실습

데이터 재탐색

```
table(filter(data2sub, data2sub$class==2) $dataset)
table(filter(data2sub, data2sub$class==2) $fbs)
table(filter(data2sub, data2sub$class==2) $exang)
```

```
> table(filter(data2sub, data2sub$class==2) $dataset)
```

```
Cleveland
      110
```

```
> table(filter(data2sub, data2sub$class==2) $fbs)
```

```
FALSE  TRUE
    103     7
```

```
> table(filter(data2sub, data2sub$class==2) $exang)
```

```
FALSE  TRUE
    94    16
```

#자료가 필터링 되면서
dataset 변수가 단일범주화됨을 확인

4. 군집화 방법

(5) 실습

모델 재설정

문제가 되는 변수를 제거했지만, 잘 돌아갈까?

```
model3_2<-glm(target ~ .-class-dataset , family="binomial",
               data=filter(data2sub,data2sub$class==2) )
```

`contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]])에서 다음과 같은 에러가 발생했습니다:
contrasts는 오로지 2 또는 그 이상의 level들을 가진 요인들에만 적용할 수 있습니다

그럼 이건 어떨까?

```
filtered<-filter(data2sub,data2sub$class==2)
filtered<-select(filtered, c(-dataset))
model3_2<-glm(target ~ ., family="binomial",
               data=filtered)
```

#문제가 되는 해당 변수를 미리 제외

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-20.26236	2191.07935	-0.009	0.992622	
cpatypical angina	-2.31941	0.96788	-2.396	0.016558 *	
cpnon-anginal	-3.56046	1.40016	-2.543	0.010994 *	
cptypical angina	-0.24299	1.96823	-0.123	0.901744	
chol	0.01706	0.01197	1.425	0.154096	
exangTRUE	-2.17230	1.16917	-1.858	0.063172 .	
ca	4.36450	1.29290	3.376	0.000736 ***	
thalnormal	13.48056	2191.07836	0.006	0.995091	
thalreversible defect	17.57049	2191.07821	0.008	0.993602	

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' ' 1

■ 4. 군집화 방법

(5) 실습

원인은 R의 근본적인 연산절차때문?

formula 선언 및 해석과는 별개로 전체 data에
대해 model.frame() 과정에서 contrasts()설정이
진행되기 때문인 것으로 추측

해당 오류가 지속적으로 발생하는 경우에는

1)데이터셋 설정 단계에서 변수를 삭제하고
모형 적합

2)변수의 factor level을 변경한 후 적합
(추후에 설명)

```
> glm
function (formula, family = gaussian, data, weights, subset,
  na.action, start = NULL, etastart, mustart, offset, control = list(...),
  model = TRUE, method = "glm.fit", x = FALSE, y = TRUE, singular.ok = TRUE,
  contrasts = NULL, ...)
{
  cal <- match.call()
  if (is.character(family))
    family <- get(family, mode = "function", envir = parent.frame())
  if (is.function(family))
    family <- family()
  if (is.null(family$family)) {
    print(family)
    stop("'family' not recognized")
  }
  if (missing(data))
    data <- environment(formula)
  mf <- match.call(expand.dots = FALSE)
  m <- match(c("formula", "data", "subset", "weights", "na.action",
    "etastart", "mustart", "offset"), names(mf), 0L)
  mf <- mf[c(1L, m)]
  mf$drop.unused.levels <- TRUE
  mf[[1L]] <- quote(stats::model.frame)
  mf <- eval(mf, parent.frame())
  if (identical(method, "model.frame"))
    return(mf)
  if (!is.character(method) && !is.function(method))
    stop("invalid 'method' argument")
  :
  :
```

4. 군집화 방법

(5) 결과 비교

Class : 1

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-5.40649	2.11750	-2.553	0.010672	*
sexMale	2.00535	0.53148	3.773	0.000161	***
cptypical angina	-0.72028	0.82113	-0.877	0.380391	
cpnon-anginal	-1.61051	0.60029	-2.683	0.007299	**
cptypical angina	-2.54706	0.73407	-3.470	0.000521	***
trestbps	0.03313	0.01282	2.583	0.009796	**
fbsTRUE	-0.94379	0.59087	-1.597	0.110205	
exangTRUE	1.37525	0.54375	2.529	0.011433	*
slopeflat	0.56922	0.90696	0.628	0.530257	
slopeupsloping	-1.39978	0.94232	-1.485	0.137422	
ca	1.08635	0.27835	3.903	9.51e-05	***

Class : 2

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-20.26236	2191.07935	-0.009	0.992622	
cptypical angina	-2.31941	0.96788	-2.396	0.016558	*
cpnon-anginal	-3.56046	1.40016	-2.543	0.010994	*
cptypical angina	-0.24299	1.96823	-0.123	0.901744	
chol	0.01706	0.01197	1.425	0.154096	
exangTRUE	-2.17230	1.16917	-1.858	0.063172	.
ca	4.36450	1.29290	3.376	0.000736	***
thalnormal	13.48056	2191.07836	0.006	0.995091	
thalreversible defect	17.57049	2191.07821	0.008	0.993602	

Class 1 에서는 Class 2에 비해 Sex, trestbps 변수가 유의미한 것으로 확인됨

⇒ 임상 Background를 통해 적절한 군집을 나누고, 비교 분석하여 결과를 해석하는 역량이 중요

감사합니다

Q&A



충북대학교
CHUNGBUK NATIONAL UNIVERSITY