宋 一 城 教授指導

碩士學位 請求論文

# Excel VBA를 이용한 군집분석의 구현

2006

誠信女子大學校 大學院

統 計 學 科

尹 珉 貞

# Excel VBA를 이용한
# 군집분석의 구현

宋 一 城 敎授指導

이 論文을 碩士學位 論文으로 提出함

2005年 11月

誠信女子大學校 大學院
統 計 學 科
尹 珉 貞

# 認 准 書

尹珉貞의 碩士學位 論文으로 認准함.

審査委員 _____ ㊞

審査委員 _____ ㊞

審査委員 _____ ㊞

誠信女子大學校 大學院

# 논문개요

본 논문에서 사용하고자 하는 군집분석 (Cluster Analysis)은 다양한 특성을 지닌 관찰대상을 비유사성 행렬 (dissimilarity matrix) 또는 유사성 행렬 (similarity matrix)에 의해 동질적인 집단으로 분류하는데 쓰이는 다변량분석 방법중의 하나이다.

군집분석은 SAS 또는 SPSS를 비롯한 기존의 여러 통계 패키지를 이용하여 분석이 가능하다. 본 논문에서는 일반인들이 보편적으로 사용하는 Excel에서 군집 분석을 이용한 통계분석이 가능하도록 Excel VBA를 이용하여 군집분석 프로그램을 구현하였다. 그리고 구현된 프로그램의 사용방법의 이해와 실제 적용을 위해 실제 자료를 이용하여 군집분석의 실행과정과 결과를 보여주고 그 결과를 SAS를 이용한 분석결과와 비교하였다.

# 목 차

# 제1장 서론

분류는 인간의 기본적인 개념적 활동이라고 할 수 있다. 우리는 환경으로부터 대상을 구별하고 분류하고 그것들에 대한 용어를 배우고 관계를 이해하는 것으로 모든 것을 배우기 시작한다고 볼 수 있다. 과학분야나 사회분야에서 보면 분류체계는 이론발전에 필요한 개념 형성에 있어서 중요한 과정이 되며 관찰이나 실험을 통해 얻은 개체들을 분류하는 것이 연구 목표가 되기도 한다.

1939년 Tryon이 'Cluster analysis'라는 용어를 처음으로 사용하였으며 그 이후 군집화하기 위한 다양한 방법과 알고리즘이 개발되었다. 1963년 생물학자인 Robert Sokal과 Peter Sneath가 쓴 "Principles of Numerical Taxonomy"는 군집화 방법개발에 중요한 자극이 되었다. Sokal과 Sneath는 생물학적 분류를 위해 생명체에 대한 유사성을 측정하여 유사성이 큰 것들은 동일한 군집을 형성하며 군집의 유형이 인식된 후에는 새로운 개체를 유형을 통해 분류할 수 있다고 가정하였다. 그 이후로 과학분야에서 군집분석 응용 결과들이 많이 나왔으며 컴퓨터의 발달과 과학에서의 분류 중요성 증가 등으로 인하여 군집분석에 대한 연구가 더욱 증가하게 되었다. (김재희 (2005))

사회과학 분야에서도 군집분석에 대한 관심은 크게 증가되고 있다. 예를 들어 인류학분야에서 데이터에 근거한 인류학적 분류, 심리학분야에서 심리시험결과에 의거한 집단분류, 사회학에서 사회경제활동지표를 근거로한 계급분류 등 통계적 분류분석에 대해 관심이 증가하고 있다.

군집분석에서는 군집의 개수나 구조에 대한 가정없이 다변량 데이터로부터 거리 기준에 의해 자발적인 군집화를 유도한다. 군집분석의 첫 번째 목적은 적절한 군집으로 나누는 것이고 두 번째 목적은 각 군집의 특성, 군집

간의 차이등에 대한 탐색적 연구를 하는 것이다.

군집분석이 널리 사용되고 있는 만큼 현재 보편적으로 사용되고 있는 SAS나 SPSS등으로 군집분석이 가능하지만 통계에 대한 비전문가들이 익히고 사용하기에는 어려움이 있다. 또한 제시된 결과를 쉽게 이해하고 해석하는데 번거로움이 있다. 본 논문에서는 군집분석에 대한 알고리즘을 정리하고 일반인들이 보편적으로 사용하는 Excel의 VBA로 군집분석을 쉽게 할 수 있도록 프로그램을 구현하는 것을 시도하였다. 구현된 프로그램은 분석할 자료만 입력하면 수행하도록 하였다.

본 논문의 구성은 다음과 같다. 제1장 서론에서는 군집분석의 배경과 연구의 목적과 범위를 서술하였으며, 제2장에서는 군집분석에 대한 정의 및 이론에 대해서 살펴보고, 제3장에서는 Excel VBA로 구현한 군집분석의 알고리즘과 사용방법에 대해 설명하고, 제4장에서는 응용사례를 들어 SAS의 결과와 본 프로그램과의 결과를 비교하여 보았다. 그리고 마지막 제5장에서는 결론 및 차후연구과제를 논의하였다.

# 제2장 군집분석

## 2.1 군집분석의 정의

군집분석(Cluster Analysis)이란 주어진 많은 수의 관측개체나 대상들에 잠재되어 있는 속성들에 의하여 그룹 내적으로는 동질적이고 그룹 외적으로는 이질적인 성향을 갖는 집단으로 묶어주는 다변량분석 기법중에 하나로서 사회과학과 인문과학을 비롯한 많은 분야에서 응용되어 쓰여지고 있다. 예를 들어 재무제표에 따른 기업의 분류, 성질에 따른 세포의 분류를 하는데 응용이 되어 쓰일 수가 있고, 소비자들을 그들의 나이, 성, 소득수준, 라이프스타일 등을 기준으로 몇 개의 그룹으로 나눈 뒤 각 그룹을 겨냥한 제품 개발 및 홍보에 군집분석을 활용할 수 있다. (허명회 (2002))

군집분석의 목적은 주어진 많은 수의 관측개체를 몇 개의 군집으로 나눔으로써 대상이나 객체집단을 이해하고 군집을 효율적으로 활용하고자 함에 있다. 또한 군집의 개수, 내용, 구조를 모르는 상태에서 개체간의 유사성에 근거하여 군집을 형성하고 군집의 특성을 파악하여 군집간의 관계를 분석하고자 하는 것이다.

군집분석에 대한 기본 가정은 같은 군집에 속한 개체들끼리 밀접한 유사성이, 다른 군집에 속한 개체들끼리는 비유사성이 존재한다는 것이다. 본 논문에서는 군집분석을 위한 자료가 n개의 개체와 p개의 변수로 주어졌다고 가정한다. 즉 자료행렬 X는 다음과 같이 크기 n×p 로 구성된다.

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} \qquad (2.1)$$

두 개체를 $X_i' = (X_{i1}, X_{i2}, \cdots, X_{ip})$와 $X_j' = (X_{j1}, X_{j2}, \cdots, X_{jp})$로 표현하고 i와 j사이의 거리를 $d_{ij}$로 정의한다.

## 2.2 비유사성의 정의

비유사성(또는 유사성)은 여러개의 개체를 군집화하는 기본자료가 된다. 즉, 군집분석을 위한 자료의 형태는 비유사성(또는 유사성)의 행렬의 형태를 가져야 한다는 것이다. 다른 다변량분석에서는 주로 상관성 척도를 사용하지만 대부분의 군집분석방법들은 비유사성 행렬을 구성하기 위해서 개체들간의 '거리'를 이용한다. 그리고 군집분석은 비유사성을 어떻게 정의하느냐에 따라서도 달라진다. 거리척도의 유사성은 개체들간의 근접도를 나타내고 '거리'가 가까울수록 개체들간에는 연관성이 높다는 것을 의미한다. (강이중(2001), 김기영・전명식 (1997)) 따라서 자료가 원자료(raw data)로 주어진 경우에는 이 자료들에 대해서 각 개체 벡터들간의 거리를 구하고 구해진 거리들로 구성된 비유사성 행렬(dissimilarity matrix)을 이용하여 군집분석을 해야한다. 그러나 자료의 형태에 따라서 비유사성 행렬은 자료 자체에서 비유사성의 개념을 포함하고 있으면 그대로 사용되는 경우도 있다.

### 2.2.1 유클리드 제곱거리 (squared Euclidean distance)

유클리드 제곱거리(squared Euclidean distance)방법은 유클리드 거리(Euclidean distance)방법에 제곱을 한 것으로 계산이 편리하고 군집분석(Cluster Analysis)에서 가장 많이 쓰이는 방법으로 다음과 같이 정의된다.

$$d_{ij} = \sum_{k=1}^{p} (X_{ik} - X_{jk})^2 \qquad (2.2)$$

### 2.2.2 표준화된 유클리드 제곱거리(standardized squared Euclidean distance)

유클리드 제곱거리를 사용하여 구해진 거리는 자료의 척도에 따라서 거리 순위에 영향을 미치기 때문에 실제 거리들과 다른 결과를 유도할 수 있다.

이러한 단점을 보완하기 위해서는 각 변수를 표준편차로 나눈 표준화 변수를 고려해서 거리를 계산해야 하고 이러한 것을 표준화된 유클리드 제곱거리(standardized squared Euclidean distance)라 한다. 그리고 다음과 같이 정의된다.

$$d_{ij} = \sum_{k=1}^{p} (X_{ik} - X_{jk})^2 / S_k^2 \qquad (2.3)$$

### 2.2.3 마하라노비스 거리(Mahalanobis distance)

마하라노비스 거리(Mahalanobis distance)는 위에서 말한 변수들의 표준화 문제뿐만 아니라 변수들 간의 상관관계(correlation)가 존재하는 경우에 상관관계를 고려한 방법으로 쓰인다. 그리고 다음과 같이 정의된다.

$$d_{ij} = \sqrt{(X_i - X_j)' S^{-1} (X_i - X_j)} \qquad (2.4)$$

여기서 S는 표본공분산 행렬이다.

### 2.2.4 민코우스키 (Minkowski)

민코우스키 (Minkowski) 방법은 다음과 같이 정의 될 수 있다.

$$d_{ij} = \left\{ \sum_{l=1}^{p} |X_{il} - X_{jl}|^k \right\}^{1/k} \quad (2.5)$$

특히 k=1일 때의 민코우스키 거리는 “도시블럭” 거리라고 한다. 그리고 k=2일 때는 유클리드 거리가 된다. 이 식은 거리를 재는 일반식으로서 지수 k는 여러 가지로 변할 수 있어서 다양한 방식의 거리를 구하는 데에 이용된다.

## 2.3 군집방법

군집화 방법은 크게 두 가지로 나누어 진다. 하나는 계층적 방법(hierachical method)이고 다른 하나는 비계층적 방법 (nonhierachical method)이다. 계층적 군집화 방법은 군집의 형성에 위계가 있어서 일단 한 군집에 속하게 된 두 개체는 다시 흩어지지 않는다. 그에 비해 비계층적 군집화 방법은 군집이 형성된 이후에도 일정기준에 따라 개체들이 재결합 될 수 있다. (허명회 · 양경숙 (2001))

계층적 군집화 방법은 각 대상들 사이의 거리를 계산하여 군집화 하는데 그 거리계산 방식에 따라 최단연결법(nearest neighbor method), 최장연결법(furthest neighbor method), 군 평균법(group average method), 중심연결법(centroid clustering method), 중위수 연결법(median method), 와드법(Ward method), 가변법(flexible method)등이 있다.

### 2.3.1 최단연결법 (nearest neighbor method)

최단연결법(nearest neighbor method)은 최소한의 거리를 기준으로 하는 방법이다. 크기 n×n 거리행렬에서 가장 가까운 거리에 있는 두 개체가 a와

b라고 했을 때 이 두 대상을 먼저 군집화 하고 그 군집과 개체 (여기서는 c) 또는 다른 군집과의 거리를 비교할 때 군집내의 대상들 중 가장 가까운 거리를 선정하여 군집화 하는 방법이다. 즉 다음과 같다.

$$d_{(a,b)c} = \min(d_{ab}, d_{bc}) \tag{2.6}$$

### 2.3.2 최장연결법 (furthest neighbor method)

최장연결법(furthest neighbor method)은 군집내에서 가장 멀리있는 대상과의 거리를 기준으로 군집화 하는 방법이다. 즉, 군집 (a, b) 와 개체 c 사이의 거리를 다음식과 같이 정의하여 군집화한다.

$$d_{(a,b)c} = \max(d_{ab}, d_{bc}) \tag{2.7}$$

### 2.3.3 군 평균 연결법 (group average method)

군 평균 연결법 (group average method) 은 하나의 군집내의 모든 개체들과 다른 군집내의 모든 개체들 간의 평균거리를 기준으로 가까이 있는 개체 또는 군집을 묶어주는 방법이이다. 즉, 두 군집간의 거리를 각 군집에서 하나의 개체를 뽑아 만든 모든 가능한 $n_{(a,b)}n_c$쌍의 거리의 평균으로 다음과 같이 정의한다.

$$d_{(a,b)c} = \frac{\sum_i \sum_j d_{ij}}{n_{(a,b)}n_c} \tag{2.8}$$

예를 들어, 군집 $C_1$에는 a, b라는 개체가 포함되어 있고 군집 $C_2$에는 c라는

개체가 포함되어 있을 경우 군집$C_1$과 군집$C_2$ 사이의 거리는 (a, c), (b, c)의 거리의 평균에 의해 결정된다.

### 2.3.4 중심연결법 (centroid clustering method)

두 군집 사이의 거리를 정할 때 군집내의 모든 변수들의 중심을 기준으로 정의된다. 일반적으로 중심은 평균벡터가 사용이 되고 새로운 개체는 군집의 중심과의 거리가 가장 가까운 군집에 개체를 포함시킨다.

### 2.3.5 중위수 연결법 (median method)

중심연결법(centroid clustering method)에서 군집간 표본크기의 차가 큰 경우에 작은 군집의 특성이 무시되는 경향이 발생하는데 이럴 때 평균대신 중위수를 사용하는 중위수연결법을 사용할 수 있다.

### 2.3.6 와드법 (Ward method)

군집 중심간 거리에 가중치를 부여하여 군집간의 거리를 계산한다. 군집을 만들어 가는 각 단계마다 두 군집의 병합으로 발생하는 총 군집내 (제곱) 거리의 오차제곱합이 최소화 하도록 군집들을 병합한다. 비슷한 크기의 군집끼리 병합하는 경향이 있다.

### 2.3.7 가변법 (flexible method)

위의 6가지 각 방법에 있어서 비유사도의 식은 모수 $a_p, a_q, \beta, r$을 사용하여 다음과 같은 공통의 1개의 식으로 표현된다.

$$d_{tr} = a_p d_{pr} + a_q d_{qr} + \beta d_{pq} + r|d_{pr} - d_{qr}| \qquad (2.9)$$

Lance and Williams(1967)은 식(2.9)를 이용한 방법을 제안하고 이 식에 근거한 방법을 편성적 방법이라고 불렀다.

또 모수 $a_p, a_q, \beta, r$에 대해 $a_p + a_q + \beta = 1$, $a_p = a_q$, $\beta < 1$, $\gamma = 0$인 조건을 만족시키는 범위에서 임의의 값을 사용하는 방법을 가변법으로서 제안하였고 [표 2.1]과 같이 해당되는 상수에 특별한 값을 줌으로써 다른 여러 방법들이 사용한 거리척도 사이의 관계를 찾을 수 있다.

[표 2.1] 편성적 방법의 거리척도를 통한 계층적 군집방법들의 비교

| | $\alpha_p$ | $\alpha_q$ | $\beta$ | $\gamma$ |
|---|---|---|---|---|
| 최단연결법 | 1/2 | 1/2 | 0 | -1/2 |
| 최장연결법 | 1/2 | 1/2 | 0 | 1/2 |
| 군 평균연결법 | $n_p/n_t$ | $n_q/n_t$ | 0 | 0 |
| 중심연결법 | $n_p/n_t$ | $n_p/n_t$ | $-n_p n_q/n_t^2$ | 0 |
| 중위수연결법 | 1/2 | 1/2 | -1/4 | 0 |
| Ward연결법 | $(n_p+n_r)/(n_t+n_r)$ | $(n_q+n_r)/(n_t+n_r)$ | $-n_r/(n_t+n_r)$ | 0 |
| 가변법 | $(1-\beta)/2$ | $(1-\beta)/2$ | $\beta < 1$ | 0 |

# 제3장 Excel VBA로 구현한 군집분석

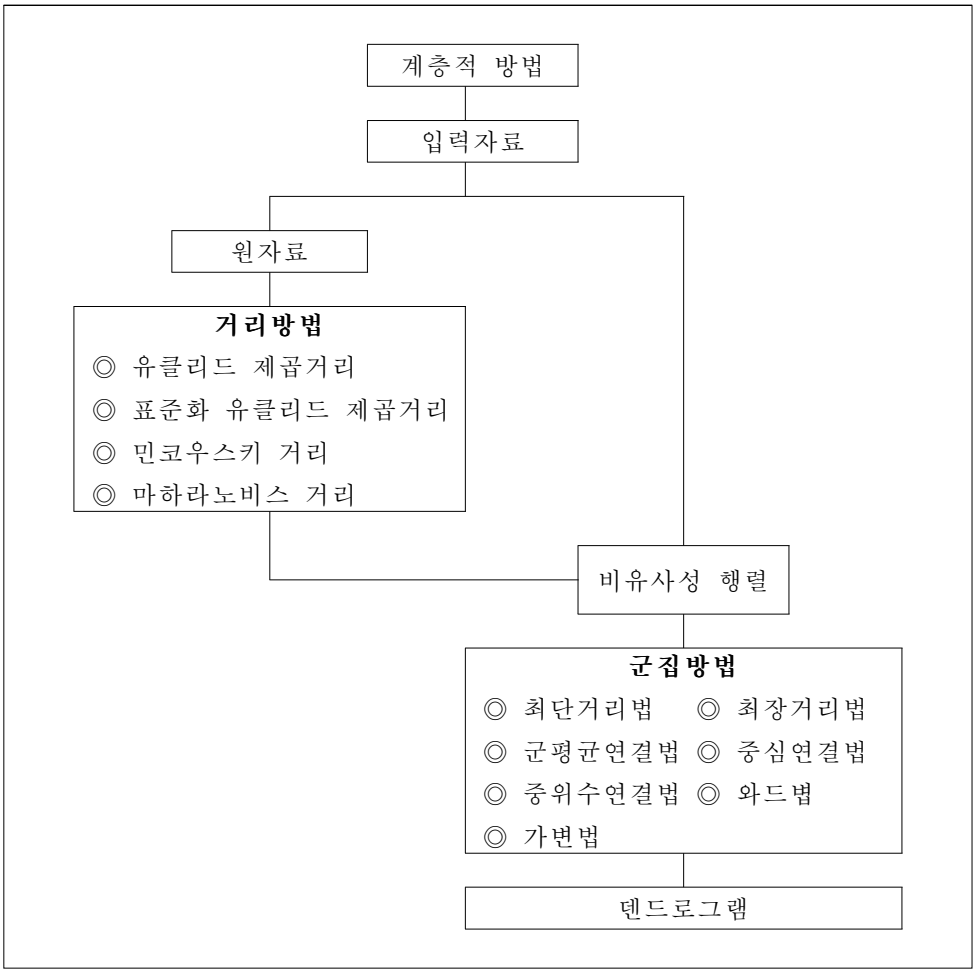
본 논문에서는 보다 편리하고 쉬운 통계적 자료분석을 위한 소프트웨어의 한 시도로서 군집분석을 위한 프로그램을 Excel VBA로 개발하였다. 분석결과를 유저폼(userform)이나 sheet창에 나타냄으로써 시각적으로 볼 수 있도록 구현하였다. Excel VBA는 Excel에서 작동되는 것으로 사용자에게 친숙하여 전문가 뿐만이 아니라 초보자도 쉽게 프로그램을 사용할 수 있도록 되어있고 결과를 쉽게 이해할 수 있도록 도와준다.

## 3.1 군집분석방법 프로그램의 구조와 이용

본 논문에서는 군집방법중에 계층적 방법에 대해서 Excel VBA로 프로그램을 구현하였고 앞으로 이 프로그램을 '계층적 방법'이라 하겠다. '계층적 방법'에 사용된 프로그램의 구조는 [그림 3.1]과 같다.

구현한 프로그램은 부록에 첨부하였다. 계층적 방법을 사용하기 위해서 [그림 3.2]와 같이 Excel의 주 메뉴로 추가되어 사용가능 하도록 하였다. 메뉴는 '계층적 방법', '결과 폼 보기'로 이루어진다. 분석수행 버튼을 누르면 프로그램이 실행이 되는데 메뉴의 '결과 폼 보기'를 이용하여 결과 폼은 다시 볼 수 있다. (권혁욱 (2004), 박재영 (2003), 신봉섭 (2004), 이재현외 (2004), 오양환 (2004))

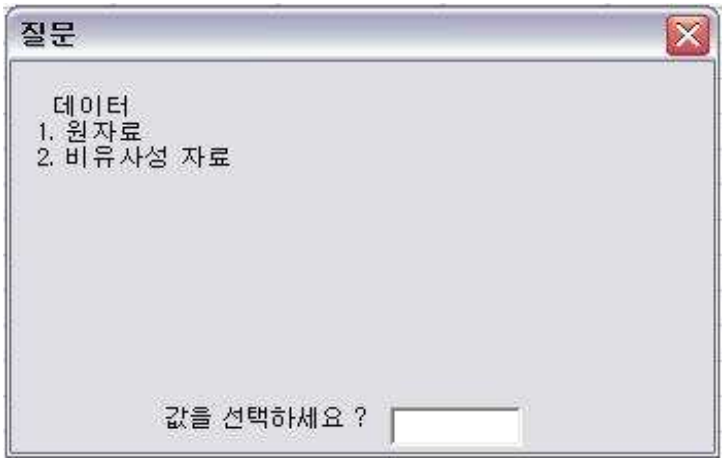
[그림 3.1] 계층적 방법의 프로그램 구조




[그림 3.2] 메뉴에 추가된 계층적 방법

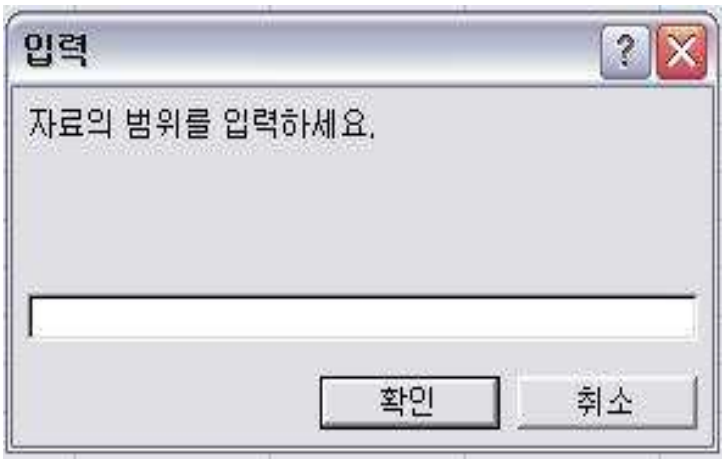
분석하고자 하는 자료파일을 열고 '계층적 방법'을 실행하면 분석할 자료의 데이터 형태를 묻는 질문의 창이 [그림 3.3]과 같이 나타나고 우리는 원자료(raw data)인지 아니면 비유사성 행렬(dissimilarity matrix)자료 자체인지를 파악하여 선택하게 된다. 형태를 선택하면 범위를 묻는 InputBox가 [그림 3.4]와 같이 나타난다. 여기서 분석을 위한 자료시트를 선택하고 자료의 범위를 지정해준다. InputBox에 자료의 범위를 입력한 후에는 입력자료에 대한 개체수와 변수에 대한 확인 창이 나타나게 된다.

[그림 3.3] 자료의 형태 질문 창

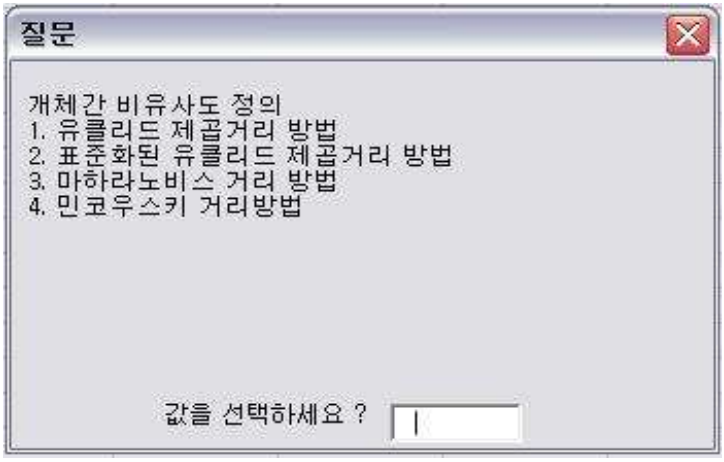



입력자료 확인이 끝난 후에는 데이터의 형태가 원자료(raw data)였다면 거리구하는 방법을 선택하는 창이 [그림 3.5]와 같이 나타나고 다음단계로 군집방법를 선택하는 창이 [그림 3.6]과 같이 나타난다.

[그림 3.4] 자료 범위 입력

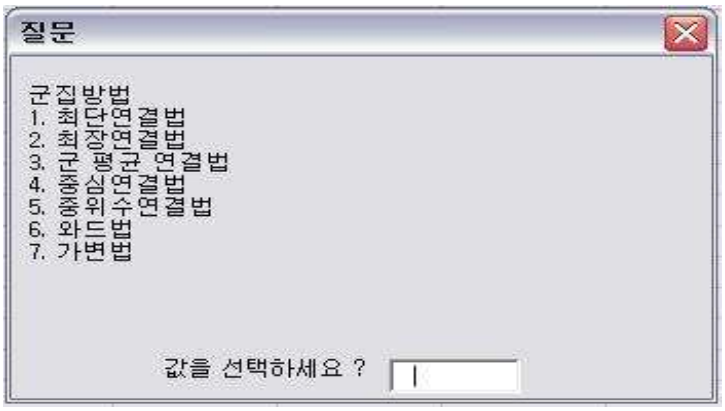



[그림 3.5] 거리방법 입력 창

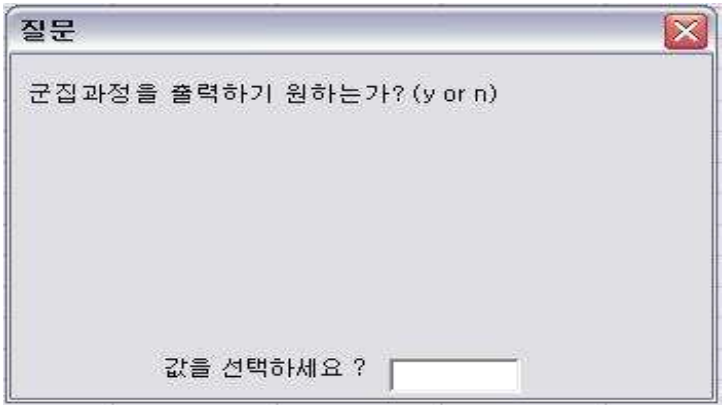



만약 데이터의 형태가 비유사성 행렬(dissimilarity matrix)이라면 [그림 3.5]를 건너뛰고 [그림 3.6]이 나타난다. 위의 선택들을 하고 나면 [그림 3.7]과 같이 군집과정의 출력여부를 묻는 창이 나타나고 [그림 3.8]과 같이 덴드로그램의 출력여부를 묻는 창이 나타난다.

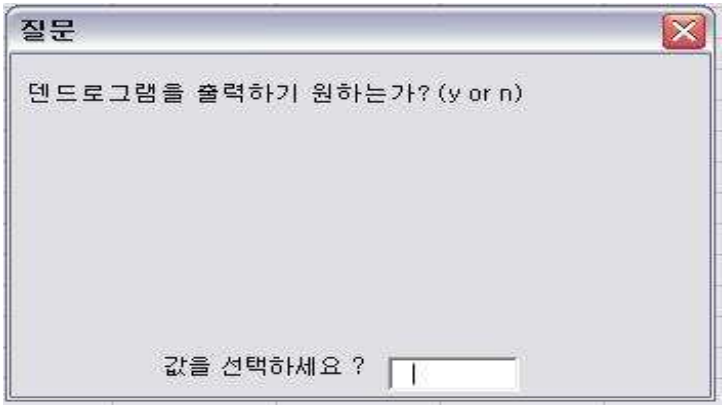
[그림 3.6] 군집방법 입력 창



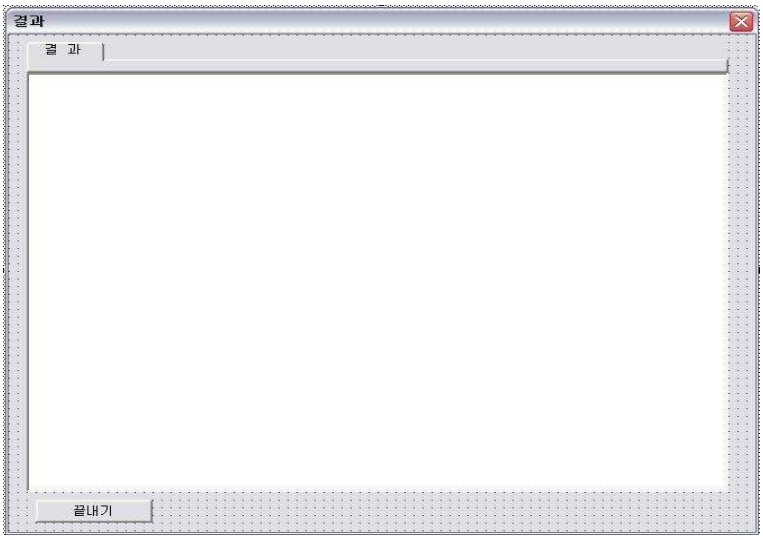

[그림 3.7] 군집과정 출력여부 창




[그림 3.8] 덴드로그램 출력여부 창



[그림 3.9] 결과 폼




위의 입력과정이 모두 끝나면 분석결과가 시트(sheet)에 나타난다. 각 시트에 나타난 결과에 대해서는 제4장에서 응용사례를 통해 자세히 설명하겠다. 또한 메뉴창에서 '결과 폼 보기'를 실행하게 되면 군집분석에 대한 수행결과가 [그림 3.9]와 같은 유저폼에 나타나게 된다.

# 제4장 응용사례

## 4.1 Excel VBA를 이용한 군집분석

군집분석 프로그램 '계층적 방법'을 이용하여 실제 자료를 분석하여 프로그램 '계층적 방법'의 수행과정을 자세히 보이고자 한다. 당뇨병 환자의 특성비교를 위해 환자들로부터 관련변수를 측정하여 15명의 환자 데이터(Rencher, A. C. (1998))를 얻었으며 [표 4.1]에 자료에 대한 설명이 나타나 있다.

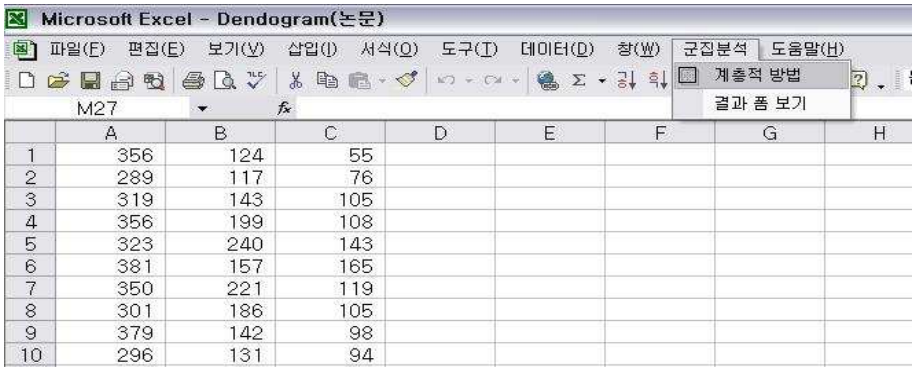
[표 4.1] 당뇨병 환자의 관련변수 자료

| 변수 | 변수설명 |
|---|---|
| $X_1$ | 글루코오즈 한계량 |
| $X_2$ | 인슐린 반응성 |
| $X_3$ | 인슐린 저항성 |

분석에 사용된 자료는 모두 15개의 개체와 3개의 변수로 되어있고 변수명은 편의상 ID, X1, X2, X3 로 명명하였다. 이 자료의 수행 절차는 다음에 있는 [그림 4.1], [그림 4.2], [그림 4.3]과 같다.

[그림 4.1] '계층적 방법'의 수행 및 데이터 형태·범위 입력

✷ '계층적 방법' 프로그램 시작



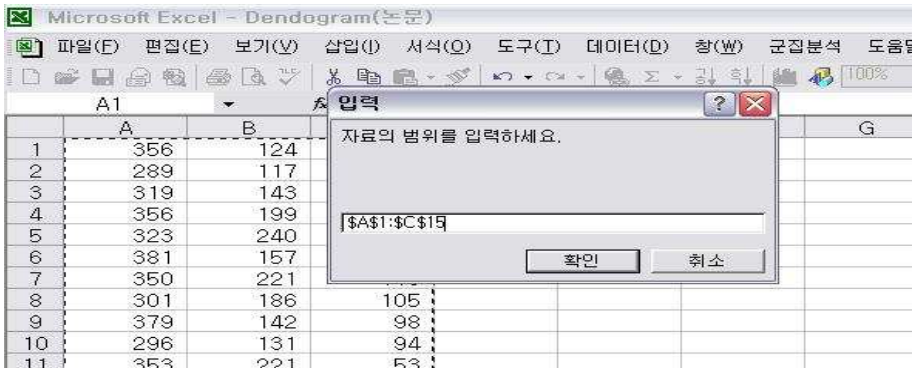

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | 356 | 124 | 55 | | | | | |
| 2 | 289 | 117 | 76 | | | | | |
| 3 | 319 | 143 | 105 | | | | | |
| 4 | 356 | 199 | 108 | | | | | |
| 5 | 323 | 240 | 143 | | | | | |
| 6 | 381 | 157 | 165 | | | | | |
| 7 | 350 | 221 | 119 | | | | | |
| 8 | 301 | 186 | 105 | | | | | |
| 9 | 379 | 142 | 98 | | | | | |
| 10 | 296 | 131 | 94 | | | | | |

✷ 데이터 형태 입력


| | A | B |
|---|---|---|
| 1 | 356 | 124 |
| 2 | 289 | 117 |
| 3 | 319 | 143 |
| 4 | 356 | 199 |
| 5 | 323 | 240 |
| 6 | 381 | 157 |
| 7 | 350 | 221 |
| 8 | 301 | 186 |
| 9 | 379 | 142 |
| 10 | 296 | 131 |
| 11 | 353 | 221 |
| 12 | 306 | 178 |

| | |
|---|---|
| 10 | 94 |
| 11 | 53 |
| 12 | 66 |

✷ 데이터 범위 입력



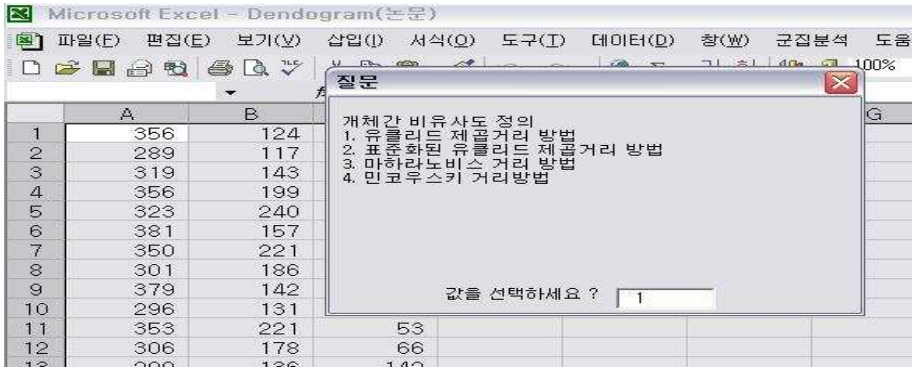

| | A | B |
|---|---|---|
| 1 | 356 | 124 |
| 2 | 289 | 117 |
| 3 | 319 | 143 |
| 4 | 356 | 199 |
| 5 | 323 | 240 |
| 6 | 381 | 157 |
| 7 | 350 | 221 |
| 8 | 301 | 186 |
| 9 | 379 | 142 |
| 10 | 296 | 131 |
| 11 | 353 | 221 |

| | |
|---|---|
| 8 | 105 |
| 9 | 98 |
| 10 | 94 |
| 11 | 53 |

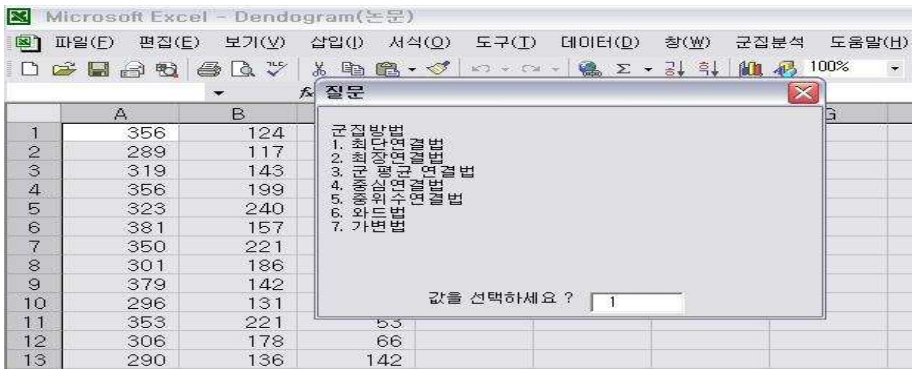
[그림 4.2] '계층적 방법'의 자료확인 및 거리·군집방법의 선택

✷ 자료확인


✷ 거리방법 입력

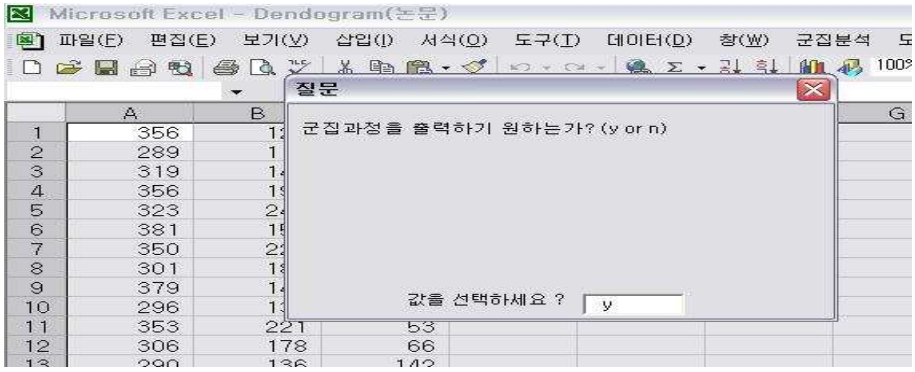



✷ 군집방법 입력

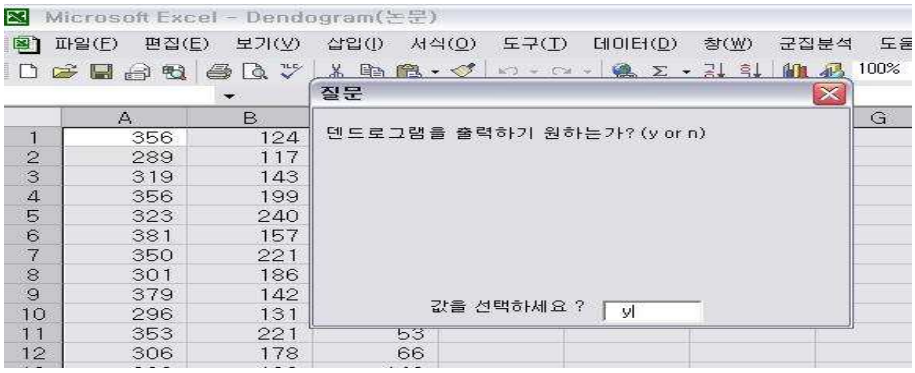


[그림 4.3] '계층적 방법'의 군집과정 · 덴드로그램 출력

✷ 군집과정 출력




✷ 덴드로그램 출력



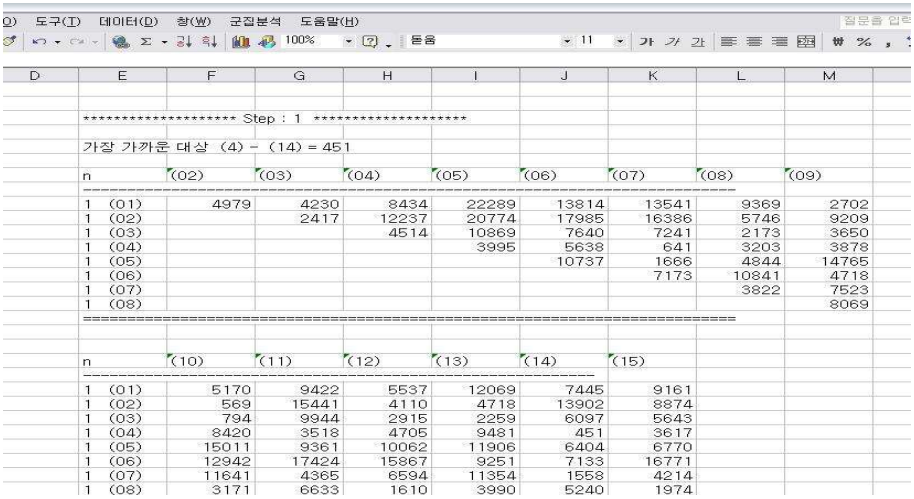

[그림 4.1]의 두 번째 화면에서는 데이터의 형태를 묻는다. 여기서는 갖고 있는 자료가 원자료(raw data)이므로 원자료를 선택한다. 형태를 선택하면 범위를 묻는 InputBox가 [그림 4.1]의 세 번째 화면과 같이 나타난다. 여기서 분석을 위한 자료시트를 선택하고 자료의 범위를 지정해준다. InputBox에 자료의 범위를 입력한 후에는 입력자료에 대한 결과로 [그림 4.2]의 첫 번째 화면과 같이 개체수 15개, 변수3개로 나타나게 된다.

계속해서 비유사도 거리와 분석하고 싶은 군집방법(여기서는 유클리드 제

곱거리 방법과 최단연결법)을 선택하고 군집과정의 출력과 덴드로그램의 출력을 원하는지 선택하게 된다. 모든 것이 수행이 되면 <result> 시트에 [그림 4.4]와 같은 결과가 나타나게 된다. 군집이 형성되어 가는 과정을 보게 되면 첫 군집으로 4번째와 14번째 환자가 이루어지는 것을 볼 수 있다. 이렇게 해서 모든 개체가 한 그룹으로 통합될 때까지 군집화가 계속된다.

[그림 4.4] '계층적 방법'의 군집과정 출력

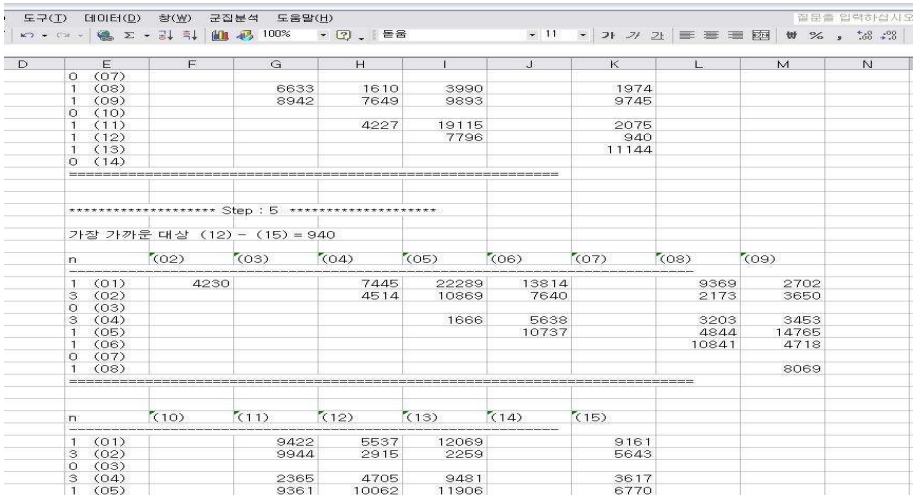
✷ 군집과정 처음출력



| D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|
| | ********** Step : 1 ********** | | | | | | | | |
| | 가장 가까운 대상 (4) - (14) = 451 | | | | | | | | |
| | n | (02) | (03) | (04) | (05) | (06) | (07) | (08) | (09) |
| | 1 (01) | 4979 | 4230 | 8434 | 22289 | 13814 | 13541 | 9369 | 2702 |
| | 1 (02) | | 2417 | 12237 | 20774 | 17985 | 16386 | 5746 | 9209 |
| | 1 (03) | | | 4514 | 10869 | 7640 | 7241 | 2173 | 3650 |
| | 1 (04) | | | | 3995 | 5638 | 641 | 3203 | 3878 |
| | 1 (05) | | | | | 10737 | 1666 | 4844 | 14765 |
| | 1 (06) | | | | | | 7173 | 10841 | 4718 |
| | 1 (07) | | | | | | | 3822 | 7523 |
| | 1 (08) | | | | | | | | 8069 |

| | n | (10) | (11) | (12) | (13) | (14) | (15) |
|---|---|---|---|---|---|---|---|
| | 1 (01) | 5170 | 9422 | 5537 | 12069 | 7445 | 9161 |
| | 1 (02) | 569 | 15441 | 4110 | 4718 | 13902 | 8874 |
| | 1 (03) | 794 | 9944 | 2915 | 2259 | 6097 | 5643 |
| | 1 (04) | 8420 | 3518 | 4705 | 9481 | 451 | 3617 |
| | 1 (05) | 15011 | 9361 | 10062 | 11906 | 6404 | 6770 |
| | 1 (06) | 12942 | 17424 | 15867 | 9251 | 7133 | 16771 |
| | 1 (07) | 11641 | 4365 | 6594 | 11354 | 1558 | 4214 |
| | 1 (08) | 3171 | 6633 | 1610 | 3990 | 5240 | 1974 |

✷ 군집과정 중간출력

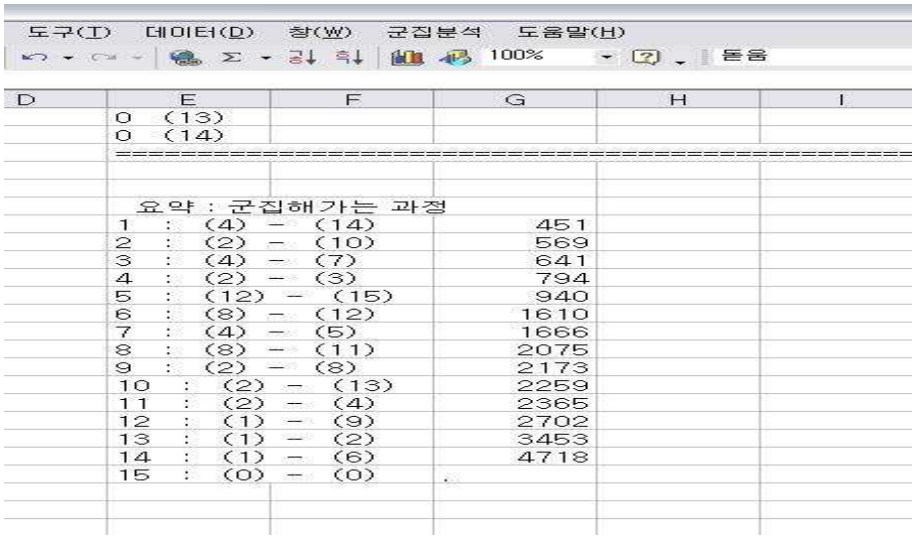


| D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 (07) | | | | | | | | | |
| | 1 (08) | | 6633 | 1610 | 3990 | | 1974 | | | |
| | 1 (09) | | 8942 | 7649 | 9893 | | 9745 | | | |
| | 0 (10) | | | | | | | | | |
| | 1 (11) | | | 4227 | 19115 | | 2075 | | | |
| | 1 (12) | | | | 7796 | | 940 | | | |
| | 1 (13) | | | | | | 11144 | | | |
| | 0 (14) | | | | | | | | | |
| | ********** Step : 5 ********** | | | | | | | | | |
| | 가장 가까운 대상 (12) - (15) = 940 | | | | | | | | | |
| | n | (02) | (03) | (04) | (05) | (06) | (07) | (08) | (09) | |
| | 1 (01) | 4230 | | 7445 | 22289 | 13814 | | 9369 | 2702 | |
| | 3 (02) | | | 4514 | 10869 | 7640 | | 2173 | 3650 | |
| | 0 (03) | | | | | | | | | |
| | 3 (04) | | | | 1666 | 5638 | | 3203 | 3453 | |
| | 1 (05) | | | | | 10737 | | 4844 | 14765 | |
| | 1 (06) | | | | | | | 10841 | 4718 | |
| | 0 (07) | | | | | | | | | |
| | 1 (08) | | | | | | | | 8069 | |

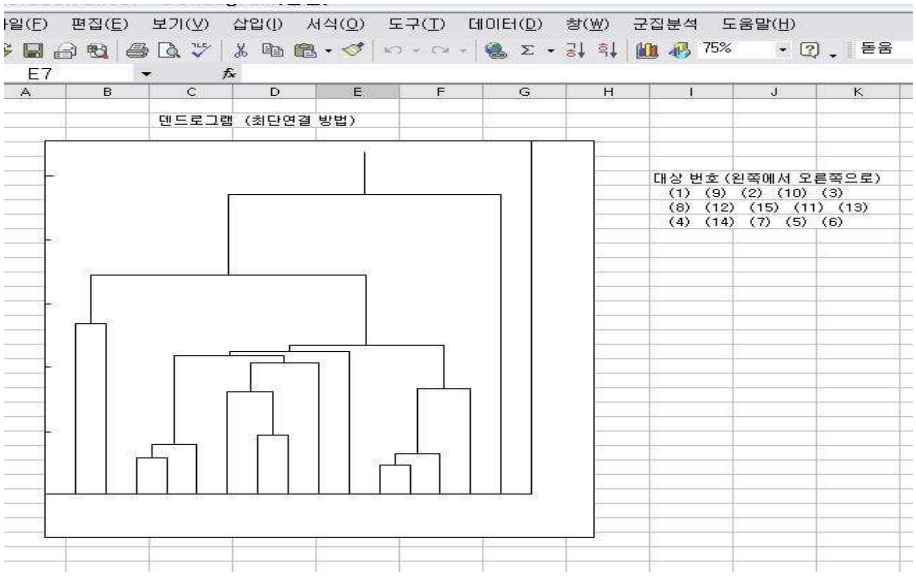
| | n | (10) | (11) | (12) | (13) | (14) | (15) |
|---|---|---|---|---|---|---|---|
| | 1 (01) | | 9422 | 5537 | 12069 | | 9161 |
| | 3 (02) | | 9944 | 2915 | 2259 | | 5643 |
| | 0 (03) | | | | | | |
| | 3 (04) | | 2365 | 4705 | 9481 | | 3617 |
| | 1 (05) | | 9361 | 10062 | 11906 | | 6770 |

[그림 4.5] '계층적 방법' 군집의 요약표 및 덴드로그램 출력

✷ 군집의 요약표 출력



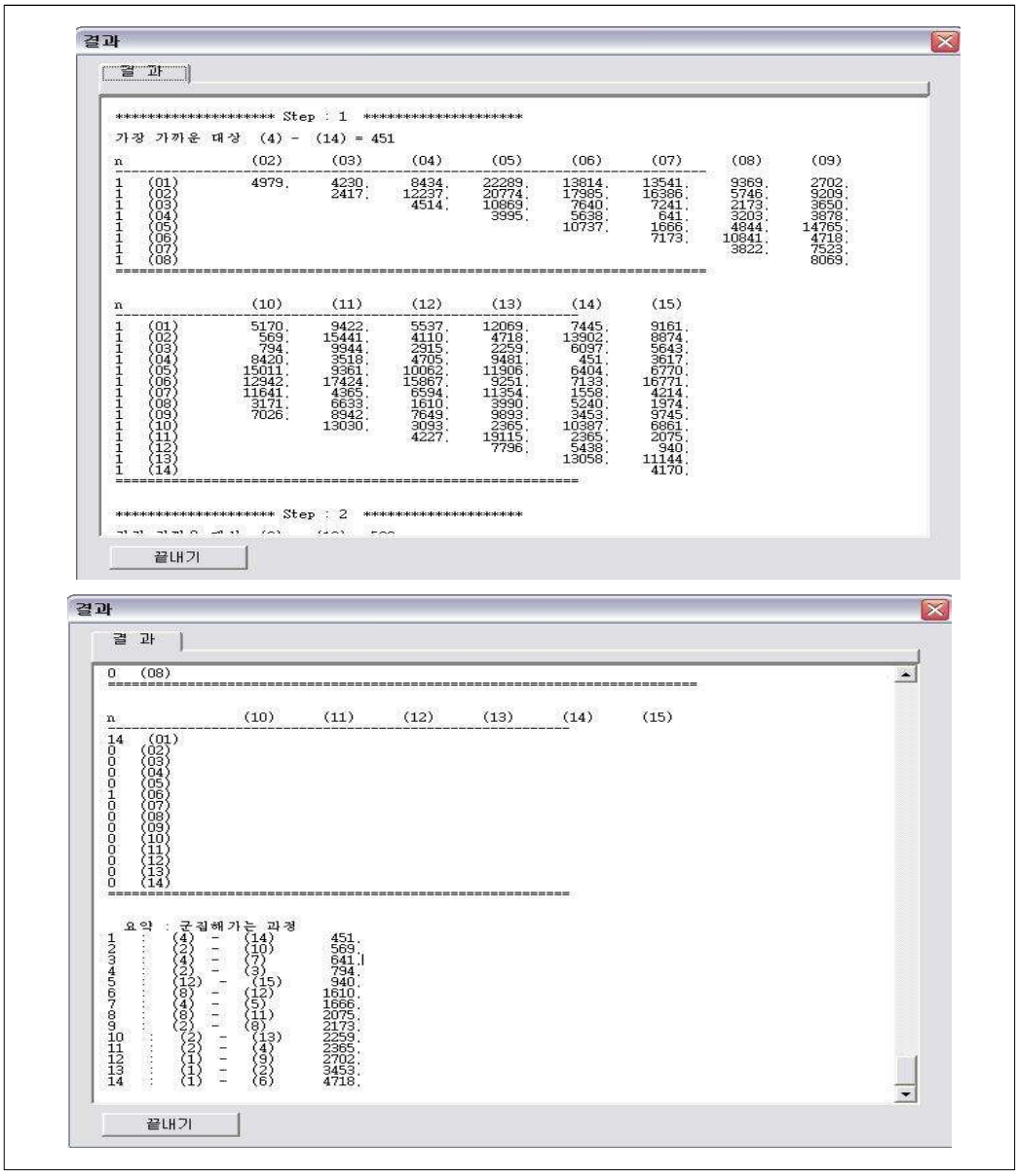

✷ 덴드로그램 출력




[그림 4.5]의 첫 번째 화면에서는 이제까지의 군집과정이 요약되어 나타나 있는 것을 볼 수 있고 그 군집을 바탕으로 [그림4.5]의 두 번째 화면같이 덴

드로그램이 그려진 것을 알 수 있다. 몇 개의 군집으로 나눌지는 결정을 하면 되는데 만약 2개의 군집으로 나눈다고 하면 (6) (1, 9, 4, 14, 2, 10, 7, 3, 12, 15, 8, 5, 11, 13) 로 묶이게 된다. 결과를 폼에 실행시켜서 보게 되면 [그림 4.6]과 같다.

[그림 4.6] '계층적 방법' 의 유저폼에 의한 결과



## 4.1 SAS 분석결과와 비교

'계층적 방법'의 프로그램에 대한 신뢰성을 보이고자 같은 예제를 가지고 SAS에서 cluster 처리절차를 사용하여 사용한 결과와 비교하려 한다. 계층적 군집방법을 적용하기 위한 SAS의 proc cluster과 proc tree를 이용한 프로그램이 [표 4.2]와 같다.

[표 4.2] SAS를 이용한 군집분석 프로그램

```
data diabetes;
  input id x1 x2 x3;
  cards;
1 356.00      124.00  55.00
2 289.00      117.00  76.00
          ⋮
14 371.00     200.00  93
15 312.00     208.00  68


proc cluster data=diabetes simple method=single out=result;
  var id x1 x2 x3;
  id id;
run;
proc tree data=result nclusters=2 out=cluster2;
  id id;
  copy x1 x2 x3;
run;
proc print data=cluster2;
  var cluster id x1 x2 x3;
run;
```

[표 4.3] SAS를 이용한 군집분석의 상관행렬 고유값 결과

The CLUSTER Procedure
Single Linkage Cluster Analysis

| Variable | Mean | Std Dev | Skewness | Kurtosis | Bimodality |
|---|---|---|---|---|---|
| id | 8.0000 | 4.4721 | 0 | -1.2000 | 0.3892 |
| x1 | 332.1 | 32.9975 | 0.1351 | -1.5749 | 0.4640 |
| x2 | 173.5 | 40.2703 | 0.1125 | -1.4526 | 0.4371 |
| x3 | 99.3333 | 33.1505 | 0.4190 | -0.4325 | 0.3523 |

Eigenvalues of the Covariance Matrix

| | Eigenvalue | Difference | Proportion | Cumulative |
|---|---|---|---|---|
| 1 | 1785.70447 | 685.06633 | 0.4663 | 0.4663 |
| 2 | 1100.63814 | 174.65728 | 0.2874 | 0.7537 |
| 3 | 925.98086 | 908.81862 | 0.2418 | 0.9955 |
| 4 | 17.16224 | | 0.0045 | 1.0000 |

Root-Mean-Square Total-Sample Standard Deviation = 30.94142
Mean Distance Between Observations = 82.84277

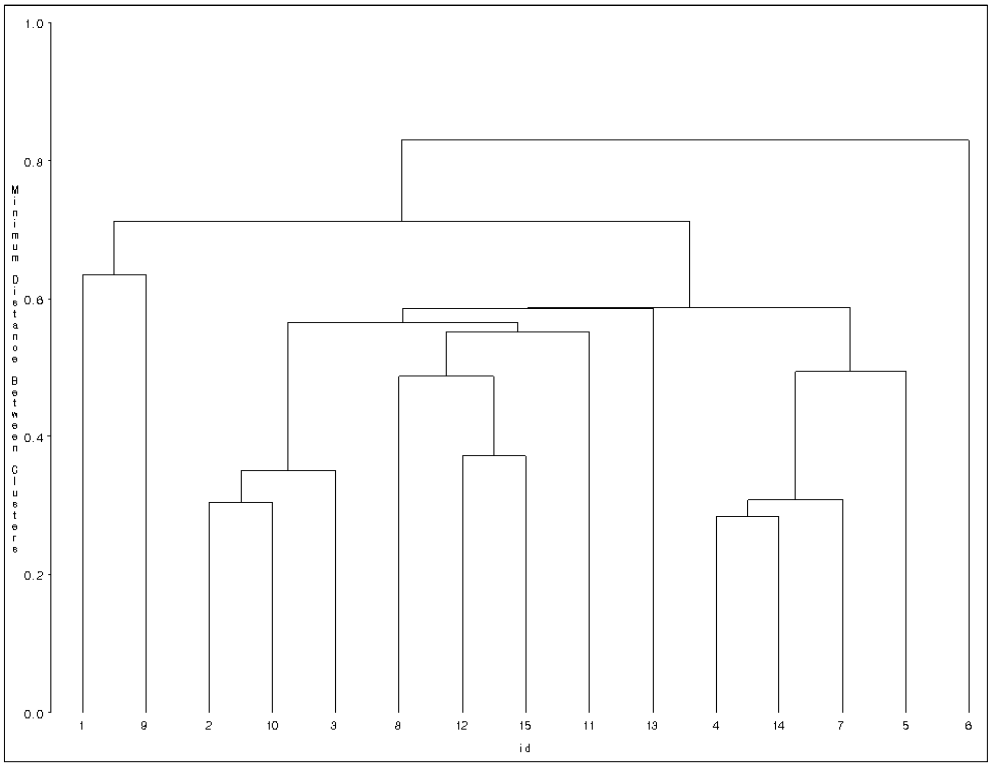
[표 4.3]의 출력결과를 보면 우선 각 변수별 기본통계량과 공분산행렬의 고유값이 주어진다. 그리고 나서 [표 4.4]같이 각 단계별 군집과정을 볼 수 있다. 그리고 SAS출력 결과에 근거하여 그린 덴드로그램도 확인할 수 있다.

[표 4.4] SAS를 이용한 군집을 형성해가는 과정 결과

Cluster History

| NCL | Clusters Joined | | FREQ | Norm Min Dist | Time |
|---|---|---|---|---|---|
| 14 | 4 | 14 | 2 | 0.2833 | |
| 13 | 2 | 10 | 2 | 0.3037 | |
| 12 | CL14 | 7 | 3 | 0.3078 | |
| 11 | CL13 | 3 | 3 | 0.3505 | |
| 10 | 12 | 15 | 2 | 0.3719 | |
| 9 | 8 | CL10 | 3 | 0.4867 | |
| 8 | CL12 | 5 | 4 | 0.4933 | |
| 7 | CL9 | 11 | 4 | 0.552 | |
| 6 | CL11 | CL7 | 7 | 0.5659 | |
| 5 | CL6 | 13 | 8 | 0.5863 | |
| 4 | CL5 | CL8 | 12 | 0.5881 | |
| 3 | 1 | 9 | 2 | 0.6349 | |
| 2 | CL3 | CL4 | 14 | 0.7119 | |
| 1 | CL2 | 6 | 15 | 0.8299 | |

[표 4.5] SAS를 이용한 덴드로그램 결과




[표 4.4]를 보았을 때 첫 군집으로 4번째와 14번째 환자가 이루어지는 것을 볼 수 있다. 그리고 다음 군집으로 2번째와 10번째 환자가 이루어지는 것을 볼 수 있다. 위의 모든 결과를 보았을 때 SAS로 구한 예제 자료에 대한 모든 값이 4.1절에서 언급한 Excel VBA에서의 군집분석 방법으로 구한 결과 값과 같거나 거의 비슷한 것을 볼 수 있다. 또한 [그림 4.5]의 덴드로그램과 [표 4.5]의 결과도 같음을 확인할 수 있다.

즉 두 개의 군집으로 나눈다고 했을 때 (6) (1, 9, 4, 14, 2, 10, 7, 3, 12, 15, 8, 5, 11, 13) 로 나누어지는 것을 그림을 통해 확인할 수 있다.

# 제5장 맺음말

본 논문에서는 비유사성 행렬에 의해 집단으로 군집화하는 군집분석의 알고리즘을 정리하고 그 알고리즘을 Excel에서의 Visual Basic으로 구현하여 보았다. 구현된 프로그램은 실제 자료를 가지고 이용방법에 대해서 좀 더 자세히 살펴보기 위하여 분석하고 그 결과를 제시하였다.

같은 자료에 대해 이를 기존의 통계패키지인 SAS와 본 프로그램을 사용하여 분석과정을 비교하여 보고 검토하였다. 분석결과는 거의 비슷하게 나왔으나 SAS는 프로그램화 하는 작업이 필요하고 이는 통계지식이 충분하지 못한 사용자들에게는 어려울 수도 있다. 이에 비해 본 논문에서 구현된 VBA프로그램은 버튼 클릭으로 모든 과정을 수행할 수 있다는 점과 출력형태가 간결하게 제시되는 장점이 있다. 또한 Excel 프로그램만 있으면 언제 어디에서든지 활용될 수 있으므로 접근성이 뛰어나다.

본 논문에서는 VBA의 기본적인 기능만을 사용하였으나 부록에 실은 프로그램을 추가 변형하거나 수정하여 VBA의 좀 더 많은 기능들을 사용한다면 SAS나 다른 통계 패키지 못지않은 프로그램이 될 것이다.

# 참고문헌


[1] 허명회 (2002), 사회과학을 위한 다변량자료분석 , 자유아카데미

[2] 김기영, 전명식 (1990), SAS군집분석 , 자유아카데미

[3] 김재희 (2005), SAS를 이용한 다변량 통계분석 , 교우사

[4] 허명회, 양경숙 (2001), SPSS다변량자료분석 , 고려정보산업

[5] 신봉섭 (2004), OA 및 통계학을 위한 엑셀 비주얼베이직 VBA 실무 , 도서출판 그린

[6] 권현욱 (2004), 엑셀러 권현욱의 VBA로 엑셀에 날개달기 , 디지털북스

[7] 이재현, 정규장외 (2004), 엑셀 VBA 고수 따라하기 , PCBOOK

[8] 오양환 (2004), 엑셀 VBA 프로그래밍 (고급) , 기전연구사

[9] Haruka Seto [공]저, 차현희 번역 (2002), (10일 만에 배우는) 엑셀 2002 VBA / VB Tech Lab. , 영진닷컴

[10] 박재영 (2003), 박재영의 엑셀 VBA , 베스트북

[11] 강이중 (2001), JAVA를 이용한 군집분석의 툴 구현 , The Journal of Korean Data Analysis Society, 2001, Vol 3, No. 4, pp.429-440

[12] 田中 豊・垂水共之・脇本和昌 (1984), パソコン統計解析ハンドブツク, 共立出版株式會社

[13] Lance, G.N. & Williams, W.T. (1967), "Mixed-data classificatory programs", Comput. J., 9, 373-380

[14] Rencher, A. C. (1998), Multivariate Statistical Inference and Applications

# ABSTRACT

## Implementation of Cluster Method with Excel VBA

**Yoon, Min Jeong**
**Department of Statistics**
**The Graduate School**
**Sungshin Women's University**

Cluster analysis is one of the multivariate analysis methods, which is used to classify observation subjects with variety of characteristic into a homogeneous group, using similarity or dissimilarity.

Cluster analysis can be done by SAS, SPSS and other existing statistical packages. In this research, Excel VBA was used to create a cluster analysis program, allowing statistical analysis in the widely used Excel program.

To help understand the direction and the application of the program, the operation process of cluster analysis with real data and its results were shown and compared with the SAS analysis results.

## 부록 : '계층적 방법' 프로그램

---

'계층적 방법' is a program to perform Cluster Method.
The paper related to this program is
"Implementation of Cluster Method with Excel VBA" - Yoon, Min Jeong

---

**[모듈1]**

```
Option Explicit
Private Const SHEET_DATA = 1
Private Const SHEET_RESULT = 2
Private Const SHEET_GRAPH = 3

Private DATA_POSITION  As Long, Private iru As Integer
Private ide As Integer, Private c_out As String
Private i_out As String, Private dist() As Variant
Private nc As Integer, Private nv As Integer
Private s() As Double, Private n() As Double
Private mat() As Double, Private xmin() As Double
Private iy() As Double, Private xx() As Double
Private y1() As Double, Private y3() As Double
Private dists() As Double, Private x() As Double
Private ap As Double, aq As Double, be As Double, ga As Double
Private iho As Integer, Private ri() As Double
Private nPrintRow As Long, Private nPrintRow2 As Long
Private ks0 As Double, Private so As Double
Private amin As Double, Private ip As Integer
Private jp As Integer, Private m As Double
Private i As Integer, Private j As Integer
Private k As Integer, Private k0 As Double
Private l0 As Double, Private k1 As Double
Private l1 As Double, Private p0 As Double
Private q0 As Double, Private p1 As Double
Private q1 As Double, Private iclear As Double
Private px As Double, Private py As Double
Private str As String, Private k3 As Double
```

```
Private k4 As Double, Private l3 As Double
Private l4 As Double, Private k2 As Double
Private l2 As Double, Private p2 As Double
Private q2 As Double

Public fOutput As fOutput, Private sOutputStr As String
Private sLineStr As String, Private preStartPos As Integer
Private stepnum As Integer, Private result_sheet As Sheets

Public Sub init()
    Sheets(SHEET_RESULT).Cells.ClearContents
    Sheets(SHEET_GRAPH).Cells.ClearContents
    Sheets(SHEET_RESULT).Cells.ClearFormats
    initQuestion
    Sheets(SHEET_DATA).Select
    stepnum = 0

End Sub

Private Function getInputValue(ByVal startPos As Integer, ByVal endPos As Integer) As
String
    Dim fIn As New fInput
    Dim sStr As String
    Dim iInputCount As Integer
start:
'    For iInputCount = startPos To endPos
'        If iInputCount = startPos Then
'            sStr = Sheets(SHEET_VALUE).Range("A" & iInputCount).Value & vbCrLf
'        Else
'                    sStr = sStr & "        " & Sheets(SHEET_VALUE).Range("B" &
iInputCount).Value & vbCrLf
'        End If
'    Next
'
    fIn.lbDisplay.Caption = q(startPos)
    If preStartPos = startPos Then
        fIn.lbErr.Caption = "입력이 잘못되었습니다. 다시 입력하세요"
    End If
    fIn.Show vbModal
```

```
    getInputValue = fIn.txtInput
    Unload fIn

    If getInputValue = "" Then
        If MsgBox("그만두시겠습니까?", vbOKCancel) = vbOK Then
            End
        Else
            Set fIn = New fInput
            GoTo start
        End If

    End If

    preStartPos = startPos
End Function
Public Sub showResult()
On Error GoTo Err

    Set fOutput = New fOutput
    fOutput.txtDisplay = sOutputStr
    fOutput.Show vbModeless

Exit Sub
Err:
    MsgBox "분석을 먼저 하세요"
End Sub
Public Sub run()
go_start:

    DATA_POSITION = 5
    sOutputStr = ""
    init
    'fOutput.txtDisplay.Text = ""
    sLineStr = ""

    ReDim ri(100, 100)
    ReDim x(100, 100)

    nPrintRow = 2
```

```
    nPrintRow2 = 1

input_iru:
    iru = getInputValue(2, 4)
    'iru = Sheets(SHEET_VALUE).Cells(2, 6).Value

    If iru = 2 Then
        gosub_10510
        GoTo go_1360
    ElseIf iru = 1 Then
        gosub_10010
    Else
        GoTo input_iru
        Exit Sub
    End If

    ReDim dist(nc, nc)

input_ide:
    ide = getInputValue(6, 10)
    'ide = Sheets(SHEET_VALUE).Cells(6, 6).Value

    If ide = 1 Then
        gosub_12010
    ElseIf ide = 2 Then
        gosub_12210
    ElseIf ide = 3 Then
        gosub_12410
    ElseIf ide = 4 Then
        gosub_12810
    Else
        GoTo input_ide
    End If

go_1360:
    ReDim s(nc, nc)
    ReDim n(nc)
    ReDim mat(nc, nc)
    ReDim xmin(nc, nc)
```

```
    ReDim iy(nc)
    ReDim xx(nc)
    ReDim y1(nc)
    ReDim y3(nc)
    ReDim dists(nc * (nc - 1) / 2)


    gosub_13010
go_1400:
    gosub_20010

input_c_out:
    c_out = getInputValue(21, 21)
    'c_out = Sheets(SHEET_VALUE).Cells(21, 6)
    If c_out = "Y" Or c_out = "y" Then
        i_out = 1
    ElseIf c_out = "N" Or c_out = "n" Then
        i_out = 0
    Else

        GoTo input_c_out
    End If

    gosub_21010
    gosub_23010


input_c_out2:
    c_out = getInputValue(23, 23)
    'c_out = Sheets(SHEET_VALUE).Cells(23, 6)
    If c_out = "Y" Or c_out = "y" Then
        i_out = 1
    Else
        i_out = 0
        GoTo input_c_out2
    End If

    gosub_34010
```

```
    Sheets(SHEET_RESULT).Select
'     c_out = InputBox("Do you want to apply another method ( Y or n ) ", "질문")
'     If c_out = "Y" Or c_out = "y" Then
'          gosub_17510
'          GoTo go_start
'     Else
'          Exit Sub
'     End
End Sub

Private Sub gosub_10010()
  Dim data As Range
    Set data = Application.InputBox(prompt:="자료의 범위를 입력하세요.", Type:=8)
    ActiveWorkbook.Worksheets(SHEET_DATA).Name = "data"

    ActiveSheet.UsedRange.Select

    nc = data.Rows.Count
    nv = data.Columns.Count

    MsgBox "개체수=" & nc & vbCr & "변수=" & nv, vbOKOnly, "자료확인"

    'ReDim x(nc, nv)
    k = 0
    For i = 1 To nc
        For j = 1 To nv
            k = k + 1
            x(i, j) = getData(k, data.Column, data.Row, SHEET_DATA)
        Next j
    Next i
End Sub

Private Sub gosub_10510()
   Dim data As Range
    Set data = Application.InputBox(prompt:="자료의 범위를 입력하세요.", Type:=8)
    ActiveWorkbook.Worksheets(SHEET_DATA).Name = "data"

    ActiveSheet.UsedRange.Select
```

```
    nc = data.Rows.Count
    nv = data.Columns.Count

    'ReDim x(nc, nc)
    k = 0
    For i = 2 To nc
        For j = 1 To i - 1
            k = k + 1
            dist(i, j) = getData(k, data.Column, data.Row, SHEET_DATA)
            dist(j, i) = dist(i, j)
        Next j
    Next i
End Sub

Private Sub gosub_11010()
    ReDim ri(nv, nv), Sum(nv)

    For k = 1 To nc
        For i = 1 To nv
            Sum(i) = Sum(i) + x(k, i) - x(1, i)
            For j = i To nv
                ri(i, j) = ri(i, j) + (x(k, i) - x(1, i)) * (x(k, j) - x(1, j))
            Next
        Next
    Next

    For i = 1 To nv
        For j = 1 To nv
            ri(i, j) = (ri(i, j) - Sum(i) * Sum(j) / nv) / (nc - 1)
        Next
    Next
End Sub

Private Sub gosub_12010()

    Dim w As Double
    For i = 1 To nc - 1
        For j = i + 1 To nc
```

```
            w = 0
            For k = 1 To nv
                w = w + (x(i, k) - x(j, k)) ^ 2
            Next k
            dist(i, j) = w
            dist(j, i) = w
        Next
        dist(i, i) = 0
    Next
    dist(nc, nc) = 0
End Sub

Private Sub gosub_12210()
    gosub_11010

    Dim w As Double
    For i = 1 To nc - 1
        For j = i + 1 To nc
            w = 0
            For k = 1 To nv
                w = w + (x(i, k) - x(j, k)) ^ 2 / ri(k, k)
            Next
            dist(i, j) = w
            dist(j, i) = w
        Next
    Next
End Sub

Private Sub gosub_12410()
    gosub_11010

    Dim w As Double
    For i = 1 To nc - 1
        For j = i + 1 To nc
            w = 0
            For k = 1 To nv
                For l = 1 To nv
                    w = w + ri(k, l) * (x(i, k) - x(j, k)) * (x(i, l) - x(j, l))
                Next
```

```
            Next
            dist(i, j) = w
            dist(j, i) = w
        Next j
    Next
End Sub

Private Sub gosub_12810()

    Dim const_k As Double
    Dim w As Double

input_const_k:
    const_k = getInputValue(27, 27)
    'const_k = Sheet3.Cells(27, 6).Value
    If const_k = 0 Then
        GoTo input_const_k
    End If

    For i = 1 To nc - 1
        For j = i + 1 To nc
            w = 0
            For k = 1 To nv
                w = w + Abs(x(i, k) - x(j, k)) ^ const_k
            Next
            dist(i, j) = w ^ (1 / const_k)
            dist(j, i) = dist(i, j)
        Next j
    Next i
End Sub

Private Sub gosub_13010()

    For i = 1 To nc - 1
        For j = i + 1 To nc
            If dist(i, j) = 0 Then
                ks0 = ks0 + 1
            End If
        Next
```

```
    Next
    gosub_17010
    so = 0

    For i = 1 To nc
        mat(i, 1) = i
        n(i) = 1
    Next i
End Sub

Private Sub gosub_17010()

    k = 0
    For i = 1 To nc - 1
        For j = i + 1 To nc
            k = k + 1
            dists(k) = dist(i, j)
        Next
    Next
End Sub

Private Sub gosub_17510()

    Dim k As Double
    k = 0
    For i = 1 To nc - 1
        For j = i + 1 To nc
            k = k + 1
            dist(i, j) = dists(k)
            dist(j, i) = dist(i, j)
        Next
    Next

    so = 0
    For i = 1 To nc
        n(i) = 1
        y1(i) = 0
        y3(i) = 0
    Next
```

```
End Sub

Private Sub gosub_20010()
input_iho:
    iho = getInputValue(12, 19)
    'iho = Sheets(SHEET_VALUE).Cells(12, 6).Value
    If iho < 1 Or iho > 7 Then
        MsgBox "입력이 잘못되었습니다. 다시 입력하세요."
        GoTo input_iho
    End If

    If iho = 7 Then
        be = getInputValue(38, 38)
        'be = Sheets(SHEET_VALUE).Cells(6, 36).Value
    End If
End Sub

Private Sub gosub_21010()

    Dim k As Integer
    Dim i As Integer
    Dim j As Integer
    Dim a As Double

    For k = 1 To nc - 1
        amin = 10000
        For i = 1 To nc - 1
            For j = i + 1 To nc
                If dist(i, j) = 0 And k < ks0 + 1 Then
                    amin = 0
                    ip = i
                    jp = j
                    GoTo go_21090
                End If
                If dist(i, j) > 0 And dist(i, j) < amin Then
                    amin = dist(i, j)
                    ip = i
                    jp = j
                End If
```

```
go_21090:
            Next
        Next

        If amin > so Then
            so = amin
        End If

        xmin(ip, jp) = amin
        a = ip * 256 + CSng(jp)
        If a > 32767 Then
            iy(k) = a - 65536!
        Else
            iy(k) = a
        End If
        For i = 1 To nc
            For j = 1 To nc
                s(i, j) = dist(i, j)
            Next
        Next
        If i_out = 1 Or k = 1 Then
            gosub_25010
        End If

        m = n(ip) + n(jp)
        For i = 1 To nc
            If n(i) = 0 Or i = jp Then
                GoTo go_21260
            End If

            Select Case iho
                Case 1: gosub_22010
                Case 2: gosub_22110
                Case 3: gosub_22210
                Case 4: gosub_22310
                Case 5: gosub_22410
                Case 6: gosub_22510
                Case 7: gosub_22610
```

```
            End Select
            dist(ip, i) = ap * s(ip, i) + aq * s(jp, i) + be * s(ip, jp) + ga * Abs(s(ip, i) -
s(jp, i))
go_21260:
            dist(i, ip) = dist(ip, i)
            dist(jp, i) = -1
            dist(i, jp) = -1
            dist(i, i) = -1
            dist(ip, jp) = -1
            dist(jp, ip) = -1

        Next

        For i = n(ip) + 1 To m
            mat(ip, i) = mat(jp, i - n(ip))
        Next

        n(ip) = m
        n(jp) = 0

    Next

End Sub

Private Sub gosub_22010()
    ap = 0.5
    aq = 0.5
    be = 0
    ga = -0.5
End Sub

Private Sub gosub_22110()
    ap = 0.5
    aq = 0.5
    be = 0
    ga = 0.5
End Sub
Private Sub gosub_22210()
    ap = n(ip) / m
```

```
    aq = n(jp) / m
    be = 0
    ga = 0
End Sub
Private Sub gosub_22310()
    ap = n(ip) / m
    aq = n(jp) / m
    be = -n(ip) * n(jp) / m ^ 2
    ga = 0
End Sub
Private Sub gosub_22410()
    ap = 0.5
    aq = 0.5
    be = -0.25
    ga = 0
End Sub
Private Sub gosub_22510()
    ap = (n(ip) + n(i)) / (m + n(i))
    aq = (n(jp) + n(i)) / (m + n(i))
    be = -n(i) / (m + n(i))
    ga = 0
End Sub
Private Sub gosub_22610()
    ap = (1 - be) / 2
    aq = (1 - be) / 2
    ga = 0
End Sub



Private Sub gosub_23010()
    Dim nce As Integer
    Dim a As Double

    setPrint "  요약 : 군집해가는 과정 "

    nce = nc ' 2
    j = 0
    For i = 1 To nce
        If iy(i) < 0 Then a = -iy(i) Else a = iy(i)
```

```
        ip = a / 256
        jp = a - ip * 256
        setPrint i & "   :   (" & ip & ")  -   (" & jp & ")"
        setPrint Format(xmin(ip, jp), "####.####"), 2
        'setPrint Format(xmin(ip, jp), "####.0000"), 2
        j = nce + i
        If j > nc - 1 Then
            'setPrint ""
            GoTo go_23160
        Else
            'setPrint ""
        End If
        If iy(j) < 0 Then
            a = -iy(j)
        Else
            a = iy(j)
        End If
        ip = a / 256
        jp = a - ip * 256
        setPrint j & "   :   (" & ip & ")  -   (" & jp & ")"
        'setPrint Format(xmin(ip, jp), "####.0000"), 2
        setPrint Format(xmin(ip, jp), "####.####"), 2
go_23160:
    Next i

End Sub

Private Sub gosub_25010()
    Dim lp As Integer
    Dim iss As Integer
    Dim ie As Integer
    Dim pos As Integer

    lp = 8
    stepnum = stepnum + 1
    setPrint "******************** Step : " & stepnum & "  ********************"
    ''setPrint "******************** Step : " & 1 & "  ********************"
    setPrint ""
    setPrint "가장 가까운 대상  (" & ip & ") -  (" & jp & ") = " & amin
```

```
setPrint ""

For iss = 2 To nc Step lp
    ie = iss + lp - 1
    If ie > nc Then ie = nc
    setPrint "n"
    pos = 0
    For i = iss To ie
        pos = pos + 1
        setPrint "'(" & Format(i, "00") & ")", pos
    Next
    str = ""
    For j = 1 To (ie - iss + 1) * 8 + 10
        str = str & "-"
    Next
    setPrint str
    str = ""
    For i = 1 To ie - 1
        setPrint "'" & n(i) & "    (" & Format(i, "00") & ")"
        pos = 0
        For j = iss To ie
            pos = pos + 1
            If i >= j Or s(i, j) = -1 Or (s(i, j) = 0 And k > ks0) Then
                '
            Else
                setPrint Format(s(i, j), "###.###"), pos
                'setPrint Format(s(i, j), "000.000"), pos
            End If
        Next
    Next
    str = "'"
    For j = 1 To (ie - iss + 1) * 8 + 10
        str = str & "="
    Next
    setPrint str
    setPrint ""
    setPrint ""
    str = ""
Next
```

```
End Sub

Private Sub gosub_34010()
    Dim xl As Double
    Dim yl As Double
    Dim ket As Double
    Dim mem As Double
    Dim a As Double
    Dim yh2 As Double
    Dim yy1 As Double
    Dim yy2 As Double
    Dim yy3 As Double

    k3 = 0
    l3 = 0
    k4 = 0
    l4 = 0
    k2 = 0
    l2 = 0
    p2 = 0
    q2 = 0
    px = 0
    py = 0

    k0 = 25
    l0 = 20
    k1 = 375
    l1 = 350
    p0 = 0
    q0 = 0
    p1 = 350
    q1 = 330
    iclear = 3
    gosub_52030

    px = 0
    py = 0
    gosub_52290
```

```
px = 350
gosub_52360

py = 330
gosub_52360

px = 0
gosub_52360

py = 0
gosub_52360

xl = 350 / (nc + 1)
yl = 280 / so

ket = Int(Log(so) / Log(10))

If Int(so / 10 ^ ket) = 1 Then
    ket = ket - 1
End If

For i = 1 To Int(so / 10 ^ ket) + 1
    If i Mod 5 = 0 Then
        mem = 6
    Else
        mem = 4
    End If
    If i Mod 10 = 0 Then
        mem = 8
    End If

    px = 0
    py = i * yl * 10 ^ ket
    gosub_52290
    px = mem
    gosub_52360
Next i
```

```
setPrint2 ""
setPrint2 " 덴드로그램  (" & q(iho + 29) & " 방법)  ", 1
setPrint2 ""
setPrint2 ""
setPrint2 ""
setPrint2 ""
setPrint2 "대상 번호 (왼쪽에서 오른쪽으로)", 7
str = ""
For i = 1 To nc
    str = str & "   (" & mat(1, i) & ")"
    If i Mod 5 = 0 Then
        setPrint2 ""
        setPrint2 str, 7
        str = ""
    Else
        '
    End If
Next
setPrint2 ""
setPrint2 str, 7
str = ""

For i = 1 To nc
    xx(mat(1, i)) = i * xl
Next

For i = 1 To nc - 1
    If iy(i) < 0 Then
        a = -iy(i)
    Else
        a = iy(i)
    End If
    ip = iy(i) / 256
    jp = a - ip * 256
    yh2 = xmin(ip, jp)
    yy1 = y1(ip) * yl
    yy2 = yh2 * yl
    yy3 = y3(jp) * yl
    px = xx(ip)
```

```
        py = yy1
        gosub_52290
        py = yy2
        gosub_52360
        px = xx(jp)
        gosub_52360
        py = yy3
        gosub_52360

        xx(ip) = (xx(ip) + xx(jp)) / 2
        y1(ip) = yh2
        y3(ip) = yh2
        y3(jp) = yh2
    Next

    px = xx(ip)
    py = yy2
    gosub_52290

    py = 320
    gosub_52360

End Sub
Private Sub gosub_40010()
    For ip = 1 To n
        ap = 1 / ri(ip, ip)
        For i = 1 To n
            If i = ip Then GoTo go_40110
            For j = 1 To n
                If j = ip Then GoTo go_40100
                ri(i, j) = ri(i, j) - ri(i, ip) * ri(ip, j) * ap
            Next j
        Next i
        For j = 1 To n
            ri(ip, j) = ri(jp, j) * ap
            ri(j, ip) = -ri(j, ip) * ap
        Next j
        ri(ip, ip) = ap
    Next ip
```

```
End Sub


Private Sub gosub_52030()
    If iclear = 1 Or iclear = 2 Or iclear = 3 Then
        'CLS
    End If

    k2 = k1 - k0
    l2 = l1 - l0
    p1 = p1 - p0
    q1 = q1 - q0
    k3 = k0
    l3 = l1
    p2 = p0
    q2 = q0

    Sheets(SHEET_GRAPH).Shapes.AddShape msoShapeRectangle, k0 + 20, l0 + 20, k1 + 20, l1 + 20
    'Sheets(SHEET_RESULT).Shapes.AddShape msoShapeRectangle, k0 + 20, l0 + (14.25 * nPrintRow), k1 + 20, l1 + 20

End Sub

Private Sub gosub_52290()
    gosub_52440
    k3 = k4
    l3 = l4
End Sub

Private Sub gosub_52360()
    gosub_52440
    Sheets(SHEET_GRAPH).Shapes.AddLine k3 + 20, l3 + 20, k4 + 20, l4 + 20
    'Sheets(SHEET_RESULT).Shapes.AddLine k3 + 20, l3 + (14.25 * nPrintRow), k4 + 20, l4 + (14.25 * nPrintRow)
    ' Sheets(SHEET_RESULT).ChartObjects(1).Chart.AddLine k3 + 20, l3 + 20, k4 + 20, l4 + 20
    k3 = k4
    l3 = l4
```

```
End Sub

Private Sub gosub_52440()
    k4 = Int((px - p0) / p1 * k2 + 0.5) + k0
    l4 = l1 - Int((py - q0) / q1 * l2 + 0.5)

    If k4 > k1 Then k4 = k1
    If k4 < k0 Then k4 = k0
    If l4 > l1 Then l4 = l1
    If l4 < l0 Then l4 = l0
    p2 = px
    q2 = py

End Sub

Private Function getData(ByVal i As Integer _
                         , ByVal posX As Long _
                         , ByVal posY As Long _
                         , ByVal PosSheet As Long) As Double
    Dim x As Integer
    Dim y As Integer

    y = Int(i / nv)
    x = i - (y * nv)
    If x = 0 Then
        y = y - 1
        x = nv
    End If
    getData = Sheets(PosSheet).Cells(y + posY, x + posX - 1).Value


End Function

Private Sub setPrint(ByVal x As String, Optional ByVal pos As Integer = 0)


    If pos = 0 Then
        nPrintRow = nPrintRow + 1
```

```
        sOutputStr = sOutputStr & sLineStr & vbCrLf
        sLineStr = ""
    Else
        Dim posCount As Integer
        For posCount = Len(sLineStr) To ((pos + 1) * 10) - Len(x) + 1
            sLineStr = sLineStr & " "
        Next

    End If

    sLineStr = sLineStr & Replace(x, "'", "")

    '여기를 풀면 Excel에 값이 표시된다.
    Sheets(SHEET_RESULT).Cells(nPrintRow, 5 + pos).Value = x

End Sub

Private Sub setPrint2(ByVal x As String, Optional ByVal pos As Integer = 0)
    If pos = 0 Then
        nPrintRow2 = nPrintRow2 + 1
    End If

    Sheets(SHEET_GRAPH).Cells(nPrintRow2, 2 + pos).Value = x
End Sub
```

---

**[모듈2]**

```
Option Explicit

Sub NewMenu()

    Dim MyMenu As CommandBarControl

    Set MyMenu = CommandBars(1).Controls.Add _
          (Type:=msoControlPopup, _
           Before:=CommandBars(1).FindControl(ID:=30010).Index, _
           temporary:=True)
```

```
    With MyMenu
        .Caption = "군집분석"

        With .Controls.Add(Type:=msoControlButton)
            .Caption = "계층적 방법"
            .FaceId = 150
            .OnAction = "run"
        End With

        With .Controls.Add(Type:=msoControlButton)
            .Caption = "결과 폼 보기"
            .OnAction = "showResult"
        End With

'       With .Controls.Add(Type:=msoControlButton)
'           .Caption = "&음수만 더하기(&M)"
'           .OnAction = "SumMinus"
'       End With
'
'       With .Controls.Add(Type:=msoControlPopup)
'           .Caption = "조건부 서식..."
'           .BeginGroup = True
'
'           With .Controls.Add(Type:=msoControlButton)
'               .Caption = "최소값 표시(&N)"
'               .OnAction = "MinValue"
'           End With
'
'           With .Controls.Add(Type:=msoControlButton)
'               .Caption = "최대값 표시(&X)"
'               .OnAction = "MaxValue"
'           End With
'
'       End With

    End With

End Sub
Sub DeleteMenu()
```

```
    On Error Resume Next
    CommandBars(1).Controls("Special").Delete
End Sub
```

---

**[모듈3]**

```
Public q(40) As String
Public Sub initQuestion()

q(2) = "    데이터" & vbCrLf & _
       " 1. 원자료" & vbCrLf & _
       " 2. 비유사성 자료" & vbCrLf

q(6) = "개체간 비유사도 정의" & vbCrLf & _
       "1. 유클리드 제곱거리 방법" & vbCrLf & _
       "2. 표준화된 유클리드 제곱거리 방법" & vbCrLf & _
       "3. 마하라노비스 거리 방법" & vbCrLf & _
       "4. 민코우스키 거리방법" & vbCrLf
q(12) = "군집방법" & vbCrLf & _
       "1. 최단연결법" & vbCrLf & _
       "2. 최장연결법" & vbCrLf & _
       "3. 군 평균 연결법" & vbCrLf & _
       "4. 중심연결법" & vbCrLf & _
       "5. 중위수연결법" & vbCrLf & _
       "6. 와드법" & vbCrLf & _
       "7. 가변법" & vbCrLf

q(21) = "군집과정을 출력하기 원하는가? (y or n)"

q(23) = "덴드로그램을 출력하기 원하는가? (y or n)"

q(27) = "input k value in eq.(11.16)"

q(29) = "Methods of Cluster Analysis" & vbCrLf & _
"1       nearest neighbor" & vbCrLf & _
"2       furthest neighbor" & vbCrLf & _
"3       Group Average" & vbCrLf & _
"4       centroid" & vbCrLf & _
```

```
"5          Median" & vbCrLf & _
"6          Ward" & vbCrLf & _
"7          flexible" & vbCrLf

q(30) = "최단연결"
q(31) = "최장연결"
q(32) = "군 평균 연결"
q(33) = "중심연결"
q(34) = "중위수연결"
q(35) = "와드"
q(36) = "가변"
q(38) = "베타값은 ?"

End Sub
```