Cluster Analysis Tutorial

R 공개강좌

서울대학교 통계연구소

오늘 다룰 내용

1. K-means clustering
2. Partitioning Around Medoids (PAM) clustering
3. Hierarchical clustering

Data Description

```
data_univ <- read.csv("data/University.csv",header=T)
head(data_univ)
```

```
##   University   SAT Top10 Accept SFRatio Expenses Grad
## 1    Harvard 14.00    91     14      11   39.525   97
## 2  Princeton 13.75    91     14       8   30.220   95
## 3       Yale 13.75    95     19      11   43.514   96
## 4   Stanford 13.60    90     20      12   36.450   93
## 5        MIT 13.80    94     30      10   34.870   91
## 6       Duke 13.15    90     30      12   31.585   95
```

- University : University name
- SAT : average SAT score of new freshmen
- Top10 : percentage of new freshmen in top 10
- Accept : percentage of applicants accepted
- SFRatio : student-faculty ratio
- Expenses : estimated annual expenses
- Grad : graduation rate

표준화

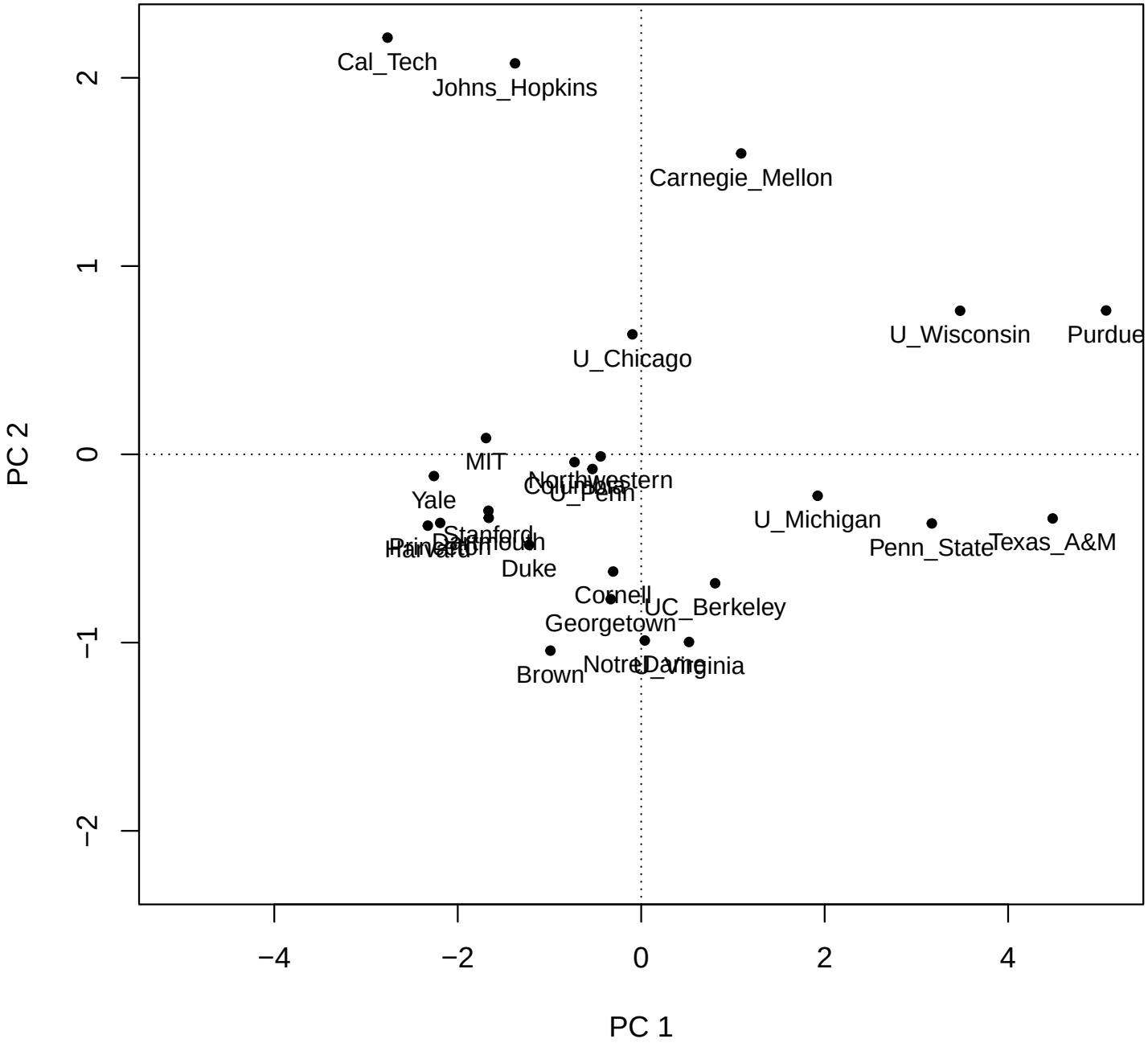
```
#분석에 사용되는 변수들의 범위에 차이가 있다면,
#가장 큰 범위를 갖는 변수가 결과에 가장 큰 영향을 미치게 되기에
#이를 방지하려면 표준화를 통해 변수들의 범위에 차이를 줄여줄 필요
data_univ_new <- scale(data_univ[,-1])
rownames(data_univ_new) <- data_univ[,1]
head(data_univ_new)
```

```
##                  SAT     Top10     Accept    SFRatio  Expenses       Grad
## Harvard    1.2325607 0.7471478 -1.2774171 -0.4228798 0.8413933 1.1349362
## Princeton  1.0018478 0.7471478 -1.2774171 -1.1604608 0.1963274 0.9141315
## Yale       1.0018478 0.9529737 -1.0239613 -0.4228798 1.1179293 1.0245339
## Stanford   0.8634201 0.6956914 -0.9732702 -0.1770194 0.6282200 0.6933268
## MIT        1.0479904 0.9015172 -0.4663586 -0.6687401 0.5186870 0.4725221
## Duke       0.4481368 0.6956914 -0.4663586 -0.1770194 0.2909556 0.9141315
```

Biplot I

```
pr_univ <- prcomp(data_univ_new, scale=TRUE)
pc1 <- pr_univ$x[,1]
pc2 <- pr_univ$x[,2]
Mx <- max(abs(pc1))
My <- max(abs(pc2))
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,xlim=c(-Mx,Mx),ylim=c(-My,My))
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),adj=0,pos=1,cex=0.8)
```

Biplot II


Cal_Tech
Johns_Hopkins
Carnegie_Mellon
U_Wisconsin
Purdue
U_Chicago
MIT
Northwestern
Columbia
U_Penn
Yale
U_Michigan
Stanford
Penn_State
Texas_A&M
Harvard
Duke
Cornell
UC_Berkeley
Georgetown
Brown
Notre_Dame
U_Virginia
PC 2
PC 1
2
1
0
−1
−2
−4
−2
0
2
4

Contents

K-means clustering R code I

K-means clustering

```
set.seed(200813)
km2_univ <- kmeans(x=data_univ_new,centers=2,iter.max=1000,nstart=20)
```

1. centers=2는 군집의 개수를 정하는 option이다.

2. 원래 K-means clustering algorithm은 군집이 변하지 않을 때까지 iteration을 반복하는 것이지만, iter.max=1000과 같이 최대 iteration 수를 정해두어 무한 루프가 발생하지 않도록 한다.

3. 종종 초기값으로 쓰인 군집의 중앙값에 따라 군집화의 결과가 다를 수 있다. nstart=20은 20개의 초기값으로 군집화를 수행한 후에 가장 좋은 결과를 보고하라는 뜻이다.

4. 초기 중앙값을 랜덤하게 선택하므로 위의 코드를 수행할 때마다 결과가 다를 수 있다. 이를 방지하기 위해 set.seed 함수를 사용하는 것이 좋다.

K-means clustering R code II

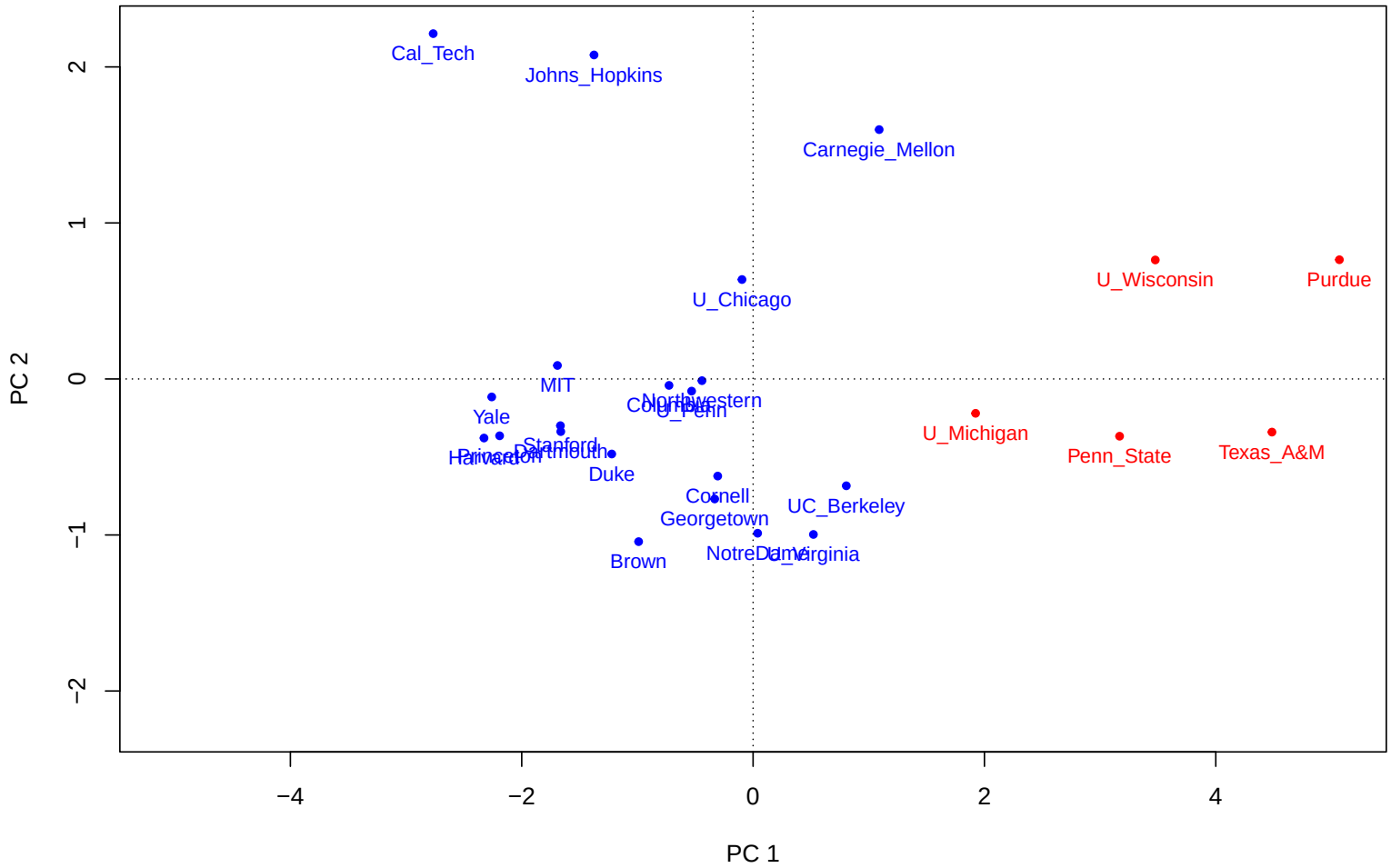
Visualization

```
col_vec <- c("red","blue")
km2_univ$cluster # 각 군집의 labeling
```

```
##          Harvard         Princeton             Yale          Stanford
##                2                 2                 2                 2
##              MIT             Duke          Cal_Tech         Dartmouth
##                2                 2                 2                 2
##            Brown     Johns_Hopkins         U_Chicago           U_Penn
##                2                 2                 2                 2
##           Cornell     Northwestern          Columbia        NotreDame
##                2                 2                 2                 2
##        U_Virginia        Georgetown  Carnegie_Mellon        U_Michigan
##                2                 2                 2                 1
##      UC_Berkeley       U_Wisconsin       Penn_State           Purdue
##                2                 1                 1                 1
##        Texas_A&M
##                1
```

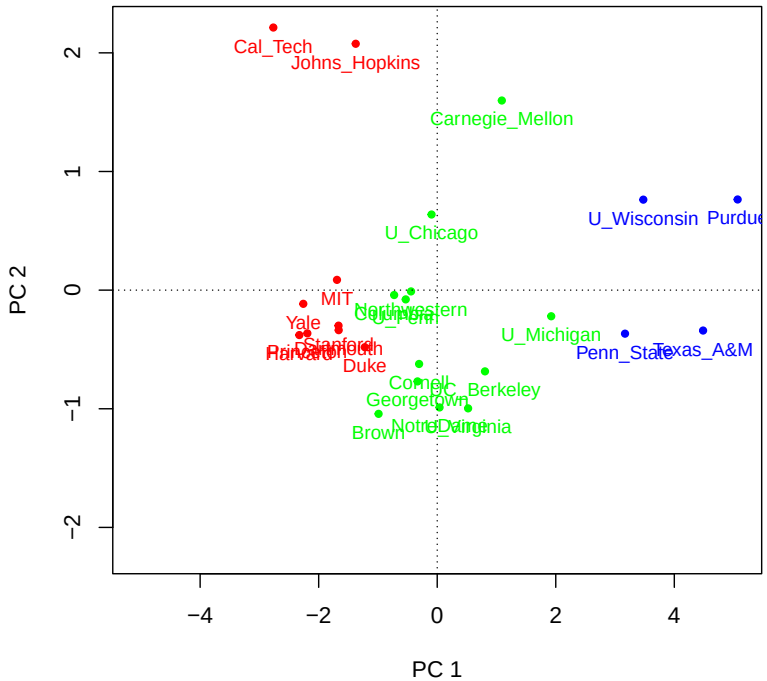
K-means clustering R code II.v2

```
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),col=col_vec[km2_univ$cluster])
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),
     adj=0,pos=1,cex=0.8,col=col_vec[km2_univ$cluster])
```



K-means clustering R code III

```
set.seed(200813)
km3_univ <- kmeans(x=data_univ_new,centers=3,iter.max=1000,nstart=20)
col_vec <- c("red","blue","green")
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),col=col_vec[km3_univ$cluster])
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),
     adj=0,pos=1,cex=0.8,col=col_vec[km3_univ$cluster])
```



Calinski-Harabasz Index

- Calinski-Harabasz Index는 군집화가 얼마나 잘 수행되었는지를 나타내는 지수로서, 다음과 같이 정의된다.

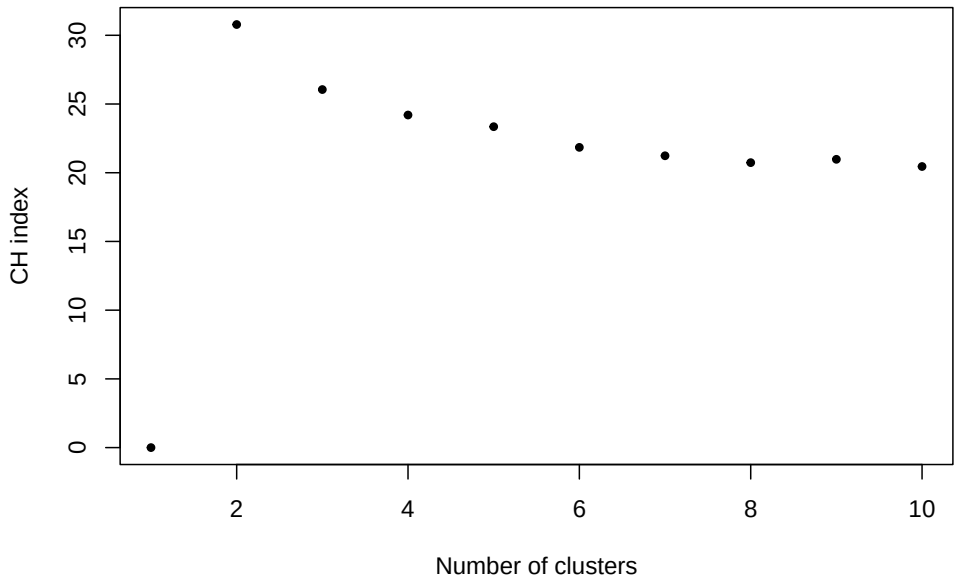
$$\text{CH index} = \frac{(n-K)SS_B}{(K-1)SS_W}$$

- 여기서 n은 관측치의 수, K는 cluster의 수, $SS_B = \sum_{k=1}^{K} |G_k| \, \|\bar{x}_k - \bar{x}\|^2$는 between-cluster variance, $SS_W = \sum_{k=1}^{K} \sum_{i \in G_k} \|x_i - \bar{x}_k\|^2$는 within-cluster variance이다. 여기서 $\bar{x}$ 는 전체 자료의 평균, $\bar{x}_k$ 는 k 군집의 평균, 그리고 $|G_k|$ 는 k 군집에 속한 자료의 갯수, $k = 1, \cdots, K$.
- Calinski-Harabasz Index가 클수록 within-cluster variance에 비해 between-cluster variance가 크다는 것을 나타내며, 따라서 clustering이 잘 되었음을 나타낸다.

K-means clustering R code IV

Calinski-Harabasz Index

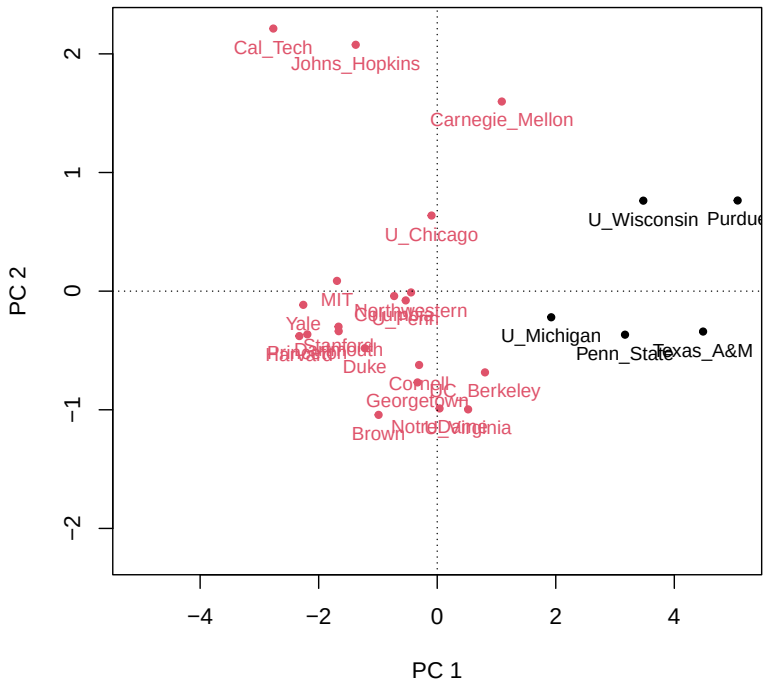
```
# install.packages("fpc")
## 주의 사항
## Do you want to install from sources the package which needs compilation?
## y/n: n
library(fpc)
set.seed(200813)
#
km_ch_univ <- kmeansruns(data=data_univ_new,krange=2:10,criterion="ch",
                         iter.max=1000,runs=20,scaledata=TRUE)
plot(1:10,km_ch_univ$crit,pch=20,xlab="Number of clusters",ylab="CH index")
```



K-means clustering R code V

Visualization

```
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),col=km_ch_univ$cluster)
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),
     adj=0,pos=1,cex=0.8,col=km_ch_univ$cluster)
```



Contents

PAM algorithm

1. Select initial medoids randomly
2. Iterate while the cost decreases:
   (1). In each cluster, make the point that minimizes the sum of distances within the cluster the medoid
   (2). Reassign each point to the cluster defined by the closest medoid determined in the previous step.

PAM clustering R code I

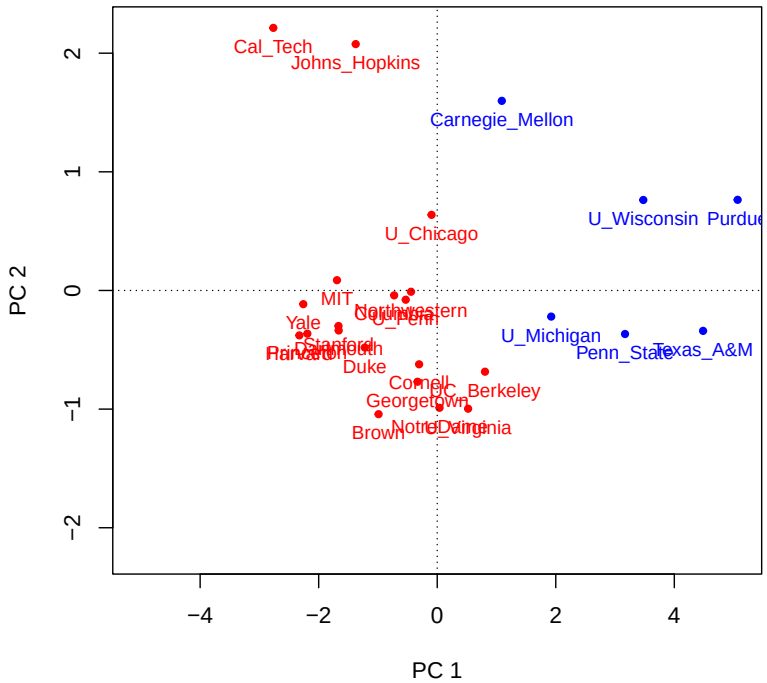
```
# install.packages("cluster")
## 주의 사항
## Do you want to install from sources the package which needs compilation?
## y/n: n
library(cluster)
set.seed(200813)
pam2_univ <- pam(x=data_univ_new,k=2)
```

- k=2는 cluster의 개수를 지정하는 옵션이다.

PAM clustering R code II

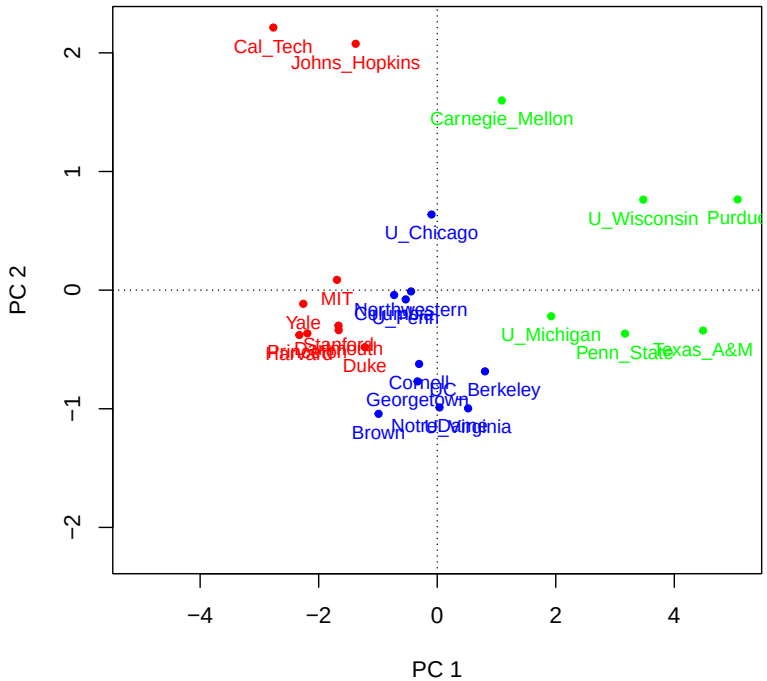
Visualization

```
col_vec <- c("red","blue")
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),col=col_vec[pam2_univ$clustering])
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),
     adj=0,pos=1,cex=0.8,col=col_vec[pam2_univ$clustering])
```



PAM clustering R code III

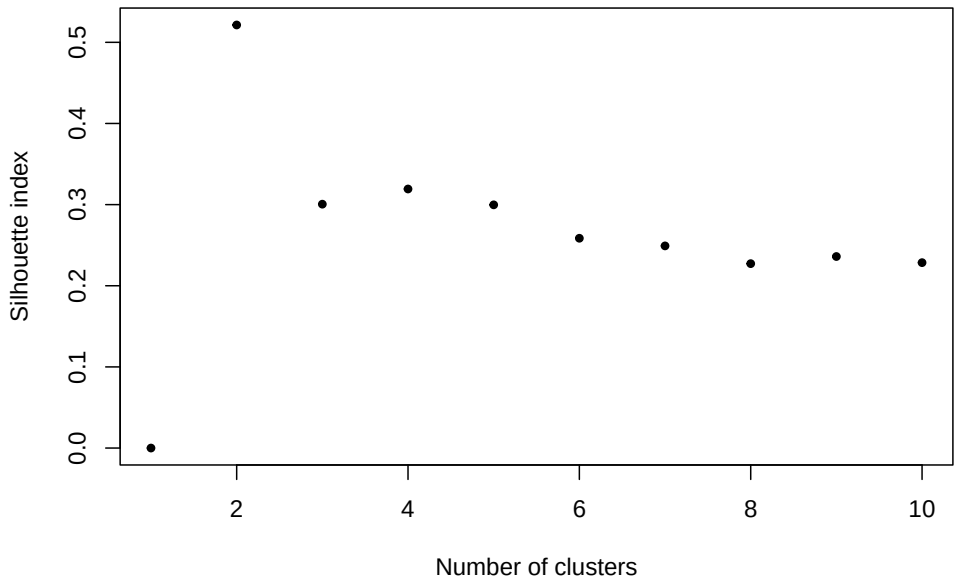
```
set.seed(200813)
pam3_univ <- pam(x=data_univ_new,k=3)
col_vec <- c("red","blue","green")
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),col=col_vec[pam3_univ$clustering])
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),
     adj=0,pos=1,cex=0.8,col=col_vec[pam3_univ$clustering])
```



PAM clustering R code V

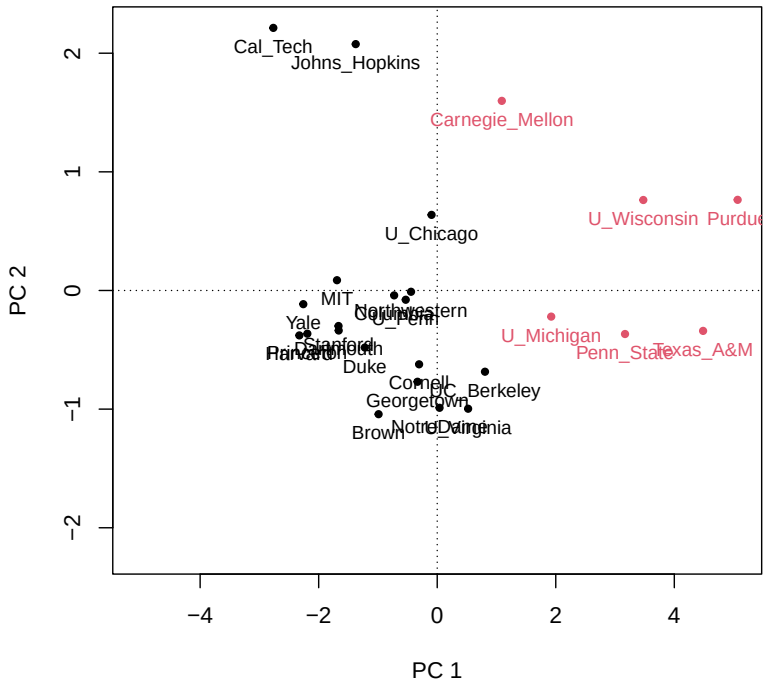
Silhouette index

```
library(fpc)
set.seed(200813)
pamk_univ <- pamk(data=data_univ_new,krange=2:10,
                  criterion="asw",scaling=FALSE) # aws : average silhouette width
plot(1:10,pamk_univ$crit,pch=20,
     xlab="Number of clusters",ylab="Silhouette index")
```



PAM clustering R code VI

```
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),
     col=pamk_univ$pamobject$clustering)
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),
     adj=0,pos=1,cex=0.8,col=pamk_univ$pamobject$clustering)
```



Contents

Hierarchical clustering

Algorithm

1. 한 개의 관측치가 포함된 n개의 군집으로 시작한다.
2. $i = n, n-1, \ldots, 2$에 대해 다음을 반복한다.
   2.1. i개 군집 간의 거리를 계산한다.
   2.2. 거리가 가장 가까운 2개의 군집을 합친다.

연결법(linkage) : 군집간의 거리를 계산하는 방법

1. complete
$$d(G_1, G_2) = \max_{i \in G_1, j \in G_2} d(x_i, x_j)$$

2. single
$$d(G_1, G_2) = \min_{i \in G_1, j \in G_2} d(x_i, x_j)$$

3. average
$$d(G_1, G_2) = \operatorname{ave}_{i \in G_1, j \in G_2} d(x_i, x_j)$$

4. centroid
$$d(G_1, G_2) = d(\bar{x}_1, \bar{x}_2), \quad \bar{x}_i = \operatorname{ave}_{j \in G_i} x_j$$

Hierarchical clustering R code I

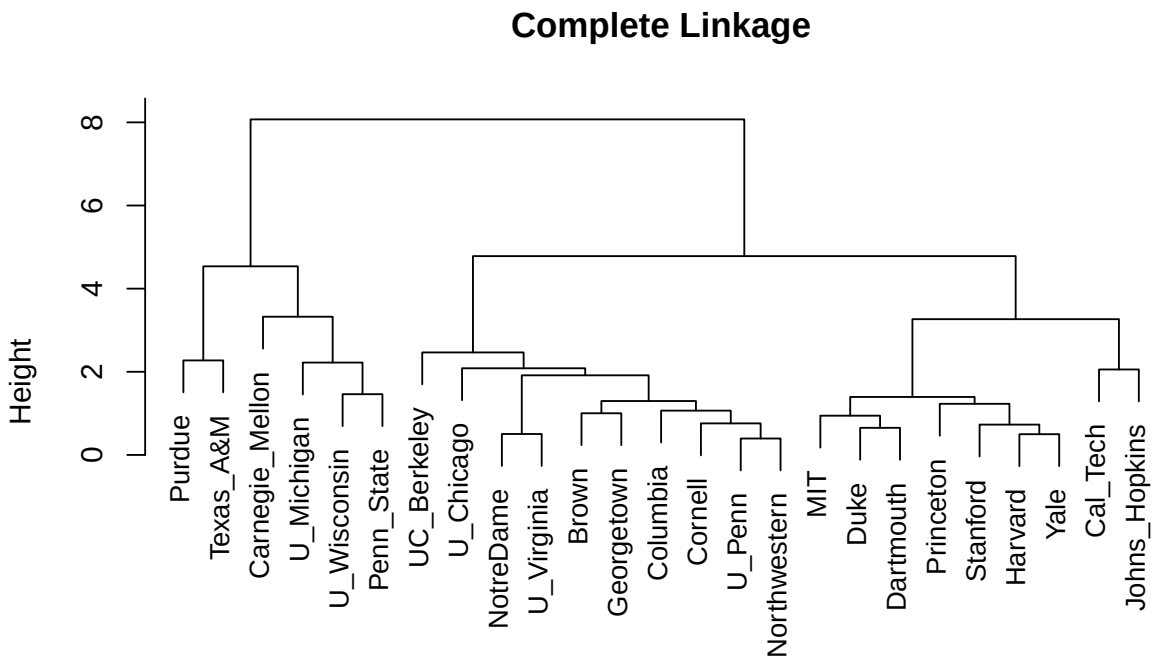
Hierarchical clustering

```
dist_univ <- dist(data_univ_new)
hc_complete_univ <- hclust(dist_univ, method="complete")
hc_single_univ <- hclust(dist_univ, method="single")
```

- `dist` 함수는 자료들 간의 거리를 계산해주는 함수이다.
- hierarchical clustering을 수행하는 함수는 hcluster이며, `method`를 통해 연결 방법을 설정해줄 수 있다.
- dendogram은 `plot` 함수를 이용하여 그릴 수 있다.
- 마지막으로, 군집의 index를 구할 때는 `cutree` 함수를 이용하면 된다.

Hierarchical clustering R code II

```
plot(hc_complete_univ,main="Complete Linkage",
     xlab="",sub="",cex=.9,labels=rownames(data_univ_new))
```



Check the appropriate number of clusters with the cutree function

```
cutree(hc_complete_univ, k=5) # k=  desired number of groups
```

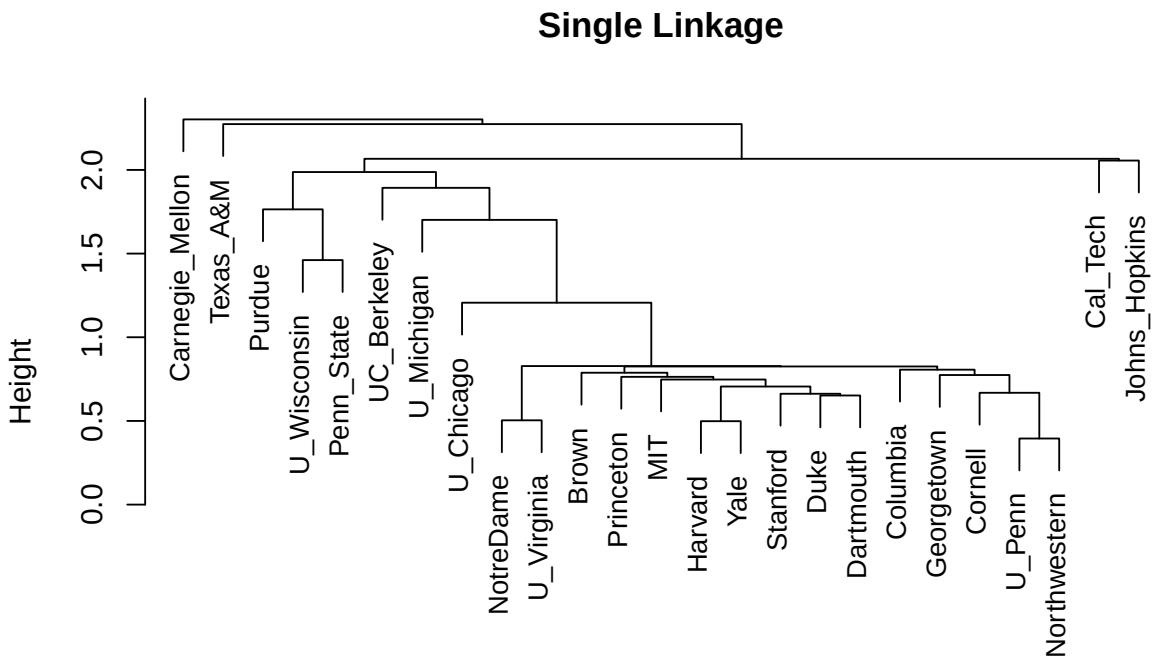
```
##          Harvard         Princeton              Yale          Stanford               MIT
##                1                 1                 1                 1                 1
##             Duke          Cal_Tech         Dartmouth             Brown    Johns_Hopkins
##                1                 1                 1                 2                 1
##        U_Chicago             U_Penn           Cornell     Northwestern          Columbia
##                2                 2                 2                 2                 2
##        NotreDame         U_Virginia        Georgetown  Carnegie_Mellon         U_Michigan
##                2                 2                 2                 3                 4
##      UC_Berkeley       U_Wisconsin        Penn_State           Purdue         Texas_A&M
##                2                 4                 4                 5                 5
```

```
cutree(hc_complete_univ, k=3) # k=  desired number of groups
```

```
##          Harvard         Princeton              Yale          Stanford               MIT
##                1                 1                 1                 1                 1
##             Duke          Cal_Tech         Dartmouth             Brown    Johns_Hopkins
##                1                 1                 1                 2                 1
##        U_Chicago             U_Penn           Cornell     Northwestern          Columbia
##                2                 2                 2                 2                 2
##        NotreDame         U_Virginia        Georgetown  Carnegie_Mellon         U_Michigan
##                2                 2                 2                 3                 3
##      UC_Berkeley       U_Wisconsin        Penn_State           Purdue         Texas_A&M
##                2                 3                 3                 3                 3
```

Hierarchical clustering R code III

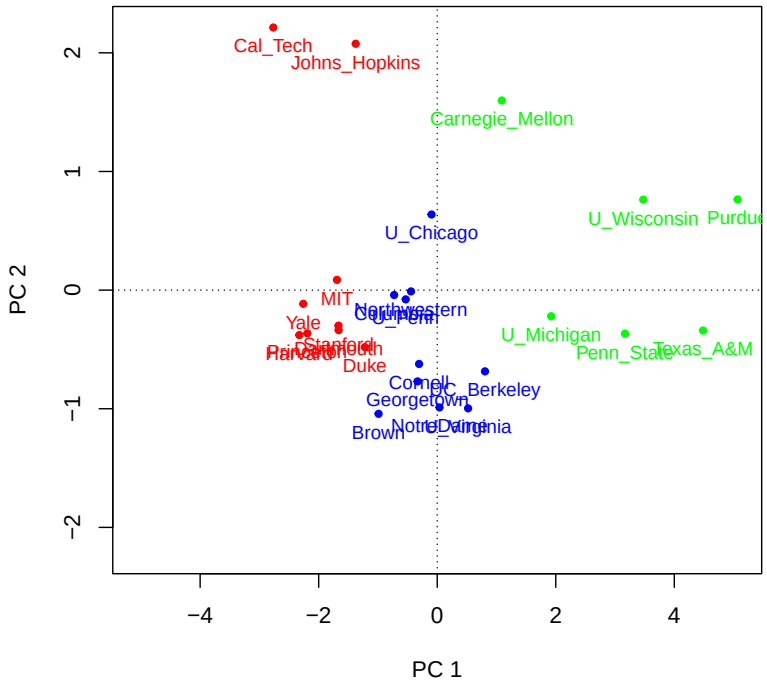
```
plot(hc_single_univ,main="Single Linkage",
     xlab="",sub="",cex=.9,labels=rownames(data_univ_new))
```



Hierarchical clustering R code IV

complete

```
col_vec <- c("red","blue","green")
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),
     col=col_vec[cutree(hc_complete_univ,3)])
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),
     adj=0,pos=1,cex=0.8,
     col=col_vec[cutree(hc_complete_univ,3)])
```



Hierarchical clustering R code V

single

```
col_vec <- c("red","blue","green")
plot(pc1,pc2,xlab="PC 1",ylab="PC 2",pch=20,
     xlim=c(-Mx,Mx),ylim=c(-My,My),
     col=col_vec[cutree(hc_single_univ,3)])
abline(h=0,lty="dotted")
abline(v=0,lty="dotted")
text(x=pc1,y=pc2,labels=rownames(data_univ_new),
     adj=0,pos=1,cex=0.8,
     col=col_vec[cutree(hc_single_univ,3)])
```