Flow 2

Jeon Jong-June

Monday, April 06, 2015

함수의 작성

- 출력 데이터가 객체하나인 경우

```
afun = function(x,y)
  {
  v = x^2 - abs(x+y)
  v
  }
afun(1,2)
```

```
## [1] -2
```

- 출력데이터가 여러가지인 경우

```
afun = function(x,y)
  {
  v1 = x^2 - abs(x+y)
  v2 = y^2 - abs(x^2+y)
  return( list(v1 = v1, v2 = v2))
  }
afun(1,2)
```

```
## $v1
## [1] -2
##
## $v2
## [1] 1
```

- 예제: 행렬의 열의 평균을 구하는 함수를 작성해보자

```
s_colMean = function(x)
  {
  v = rep(0, ncol(x))
  for ( i in 1:ncol(x))
    {
    v[i] = mean( x[i,] )
    }
  v
  }
x = matrix(runif(100), 20, 5)
s_colMean(x)
```

```
## [1] 0.4937734 0.5438174 0.5596592 0.5438205 0.5116000
```

- 예제: 행렬의 행의 평균을 구하는 함수를 작성해보자

데이터의 관리

1. 다음 x 행렬의 행은 하나의 데이터다. 데이터를 임의의 두개의 그룹으로 나누고자 한다.

```
sid = sample(1:2, 100, replace = T)
```

Note: sample takes a sample of the specified size from the elements of x using either with or without replacement.

2. 나누어진 그룹별로 평균을 구하고자 한다. 그룹평균을 어떻게 관리할것인가?

참고: 데이터가 $\mathbb{R}^p$의원소일때 평균은 각 원소들의 평균으로 정의한다.

```
gmean = matrix(0, 2, ncol(x) )
for ( i in 1:2)
  {
  x.sub = x[ which(sid = i) ,]
  gmean[i,] = colMeans(x.sub)
  }
```

3. 각 데이터와 그룹평균과 의 거리를 구하고자 한다. 벡터와 벡터의 거리는 유클리디안 거리로 정의한다.

```
idist = matrix(0, nrow(x), 2)
for ( i in 1:nrow(x))
  {
     for (j in 1:2)
       {
       idist[i, j] = sqrt( sum( (x[i,]-gmean[j,])^2) )
       }
  }
```

4. 각 데이터가 어떤 그룹평균과 가까운지 표시하고자 한다.

```
i.min = rep(0, nrow(x))

for ( i in 1:nrow(x))
  {
  i.min[i] = which.min(idist[i,])
  }
```

i.min은 각 데이터가 어떤 데이터와 가까운지 표시해주는 벡터다. 이를 다시 그룹으로 정의하고 다시 1번부터 실행한다.

(K-mean clustering 을 조사해오자)