K means 실습 1

데이터 파악

1. 데이터 다운로드

```
# https://archive.ics.uci.edu/ml/datasets/Wholesale+customers
```

UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Wholesale customers Data Set

Download: Data Folder, Data Set Description

Abstract: The data set refers to clients of a wholesale distributor. It includes the annual spending in monetary units (m.u.) on diverse product categories

Data Set Characteristics:	Multivariate	**Number of Instances:**	440	**Area:**	Business
Attribute Characteristics:	Integer	**Number of Attributes:**	8	**Date Donated**	2014-03-31
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	335183

C:\\디렉토리명\\파일 이름 ⇒ 주소는 각자 해당 CSV 파일이 위치한 파일 경로를 입력

2. 데이터 read

```
df <- read.csv('Wholesale customers data.csv', stringsAsFactors = F, header = T)
```

3. 데이터 확인

```
library(dplyr)
head(df)
```

```
##   Channel Region Fresh Milk Grocery Frozen Detergents_Paper Delicassen
## 1       2      3 12669 9656    7561    214             2674       1338
## 2       2      3  7057 9810    9568   1762             3293       1776
## 3       2      3  6353 8808    7684   2405             3516       7844
## 4       1      3 13265 1196    4221   6404              507       1788
## 5       2      3 22615 5410    7198   3915             1777       5185
## 6       2      3  9413 8259    5126    666             1795       1451
```

```
df$Channel <- df$Channel %>% as.factor() # 범주형 데이터 팩터로 변경
df$Region <- df$Region %>% as.factor() # 범주형 데이터 팩터로 변경
```

→ dplyr 패키지에서 제공하는 연산자 (좌측에다 우측 함수를 적용)

4. 결측치 확인

```
colSums(is.na(df))
```

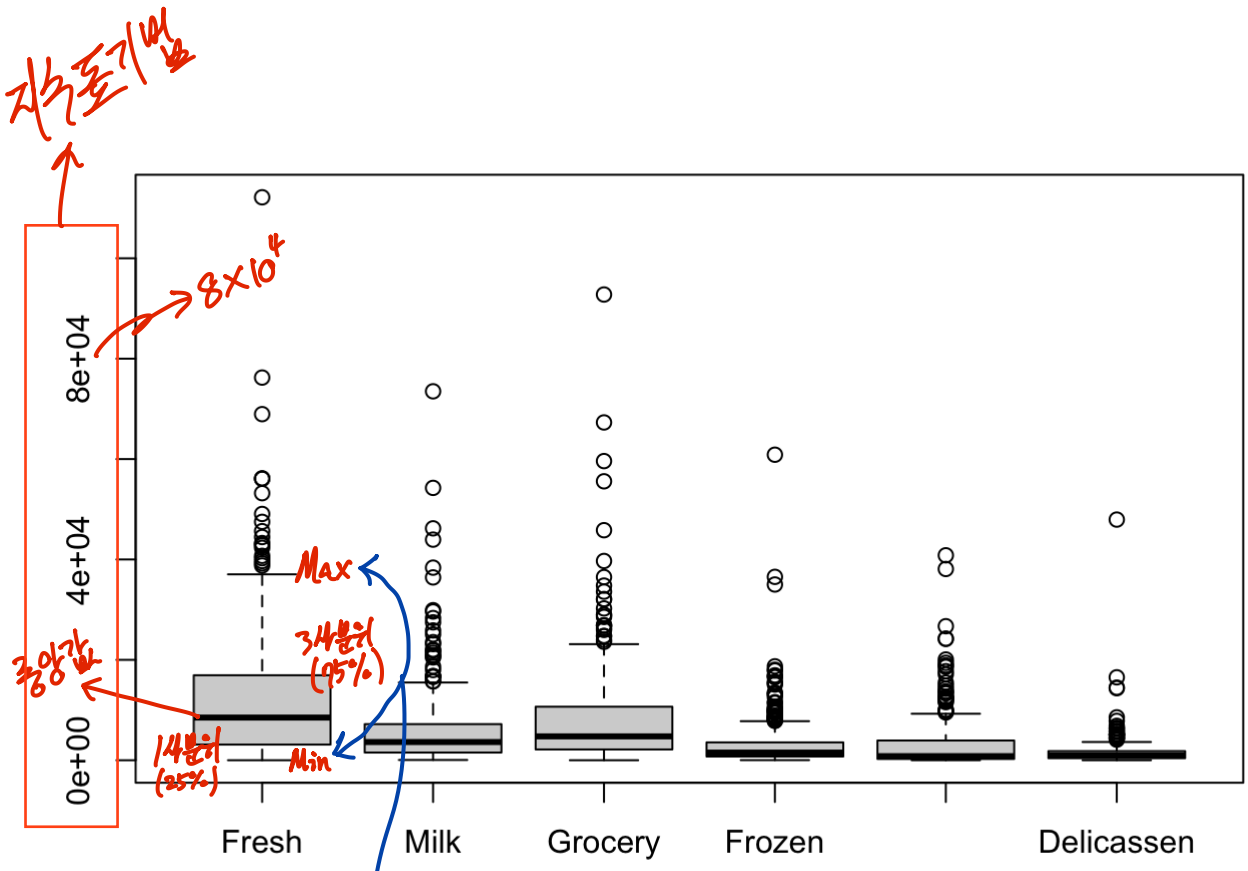
```
##          Channel           Region            Fresh             Milk
##                0                0                0                0
##          Grocery           Frozen Detergents_Paper       Delicassen
##                0                0                0                0
```

5. 변수별 기술통계 및 분포 확인

```
summary(df)
```

```
##  Channel Region      Fresh             Milk          Grocery
##  1:298   1: 77   Min.   :     3   Min.   :   55   Min.   :     3
##  2:142   2: 47   1st Qu.:  3128   1st Qu.: 1533   1st Qu.: 2153
##          3:316   Median :  8504   Median : 3627   Median : 4756
##                  Mean   : 12000   Mean   : 5796   Mean   : 7951
##                  3rd Qu.: 16934   3rd Qu.: 7190   3rd Qu.:10656
##                  Max.   :112151   Max.   :73498   Max.   :92780
##      Frozen        Detergents_Paper    Delicassen
##  Min.   :   25.0   Min.   :    3.0   Min.   :    3.0
##  1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2
##  Median : 1526.0   Median :  816.5   Median :  965.5
##  Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9
##  3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2
##  Max.   :60869.0   Max.   :40827.0   Max.   :47943.0
```

```
boxplot(df[,3:ncol(df)])
```

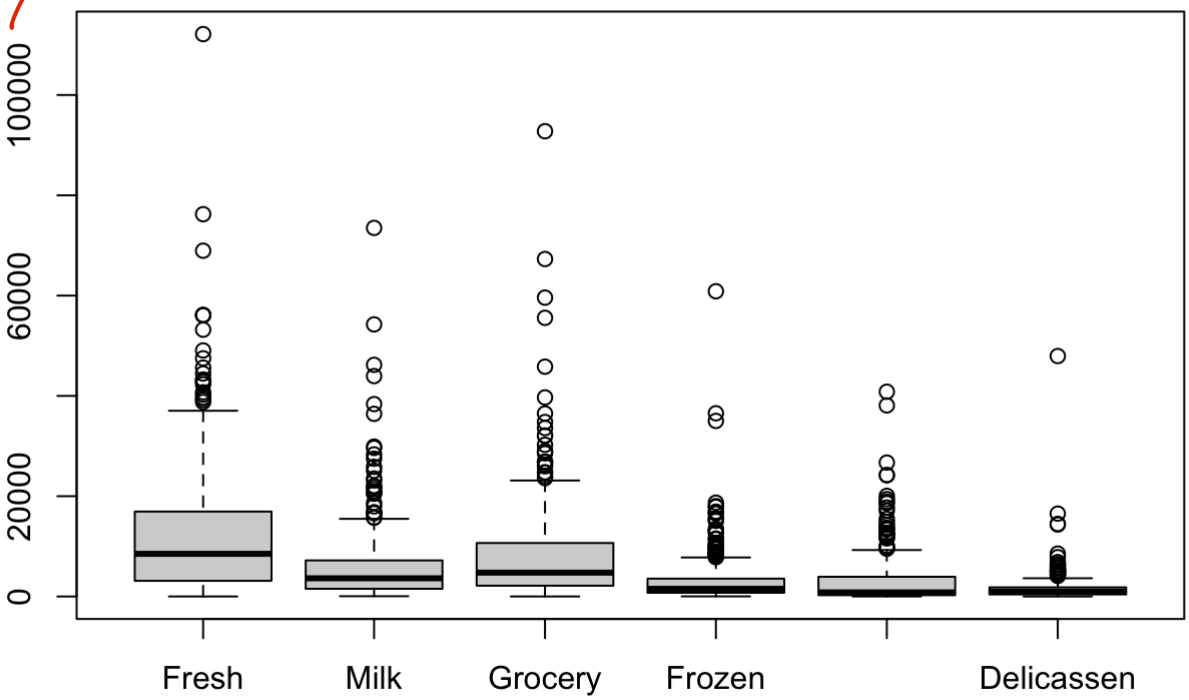


이상치가 아닌 범주의 데이터의 최대, 최솟값

6. 지수표기법 변경

```
options(scipen = 100)
boxplot(df[,3:ncol(df)])
```

아래와 같이 숫자로 변경



7. 이상치 제거

```
temp <- NULL
for (i in 3:ncol(df)) {
  temp <- rbind(temp, df[order(df[,i], decreasing = T),] %>% slice(1:5))
}
```

↳행 기준으로 바인딩 즉, 결합

↳행을 기준으로 자름

```
temp %>% arrange(Fresh) %>% head() # 중복이 있음
```

(복원 추출이기 때문)

Fresh 기준으로 오름차순 정렬

```
##   Channel Region Fresh  Milk Grocery Frozen Detergents_Paper Delicassen
## 1       2      3    85 20959   45828     36            24231       1423
## 2       2      3    85 20959   45828     36            24231       1423
## 3       2      2  8565  4980   67298    131            38102       1215
## 4       2      2  8565  4980   67298    131            38102       1215
## 5       1      3 11314  3090    2062  35009               71       2698
## 6       2      3 16117 46197   92780   1026            40827       2944
```

```
temp <- distinct(temp) # 중복 제거
df.rm.outlier <- anti_join(df,temp) # df에서 temp 제거
```

↳이상치

8. 이상치 제거 후 박스플롯 확인

1행 2열로 그래프 그리기

```
par(mfrow = c(1,2))
boxplot(df[,3:ncol(df)])
boxplot(df.rm.outlier[,3:ncol(df)])
```


```
dev.off()
```

사진 저장

bug 때문 저장하는 코드는 작성 안 했으므로 패스

```
## null device
##           1
```

분석, 결과치 확인 및 해석

설정 방법 다시 복기

1. k 군집 개수 설정 (Elbow method)

```
library(factoextra)
set.seed(1234)
fviz_nbclust(df.rm.outlier[,3:ncol(df.rm.outlier)], kmeans, method = "wss", k.max = 15) +
  theme_minimal() +
  ggtitle("Elbow Method")
```

3:8

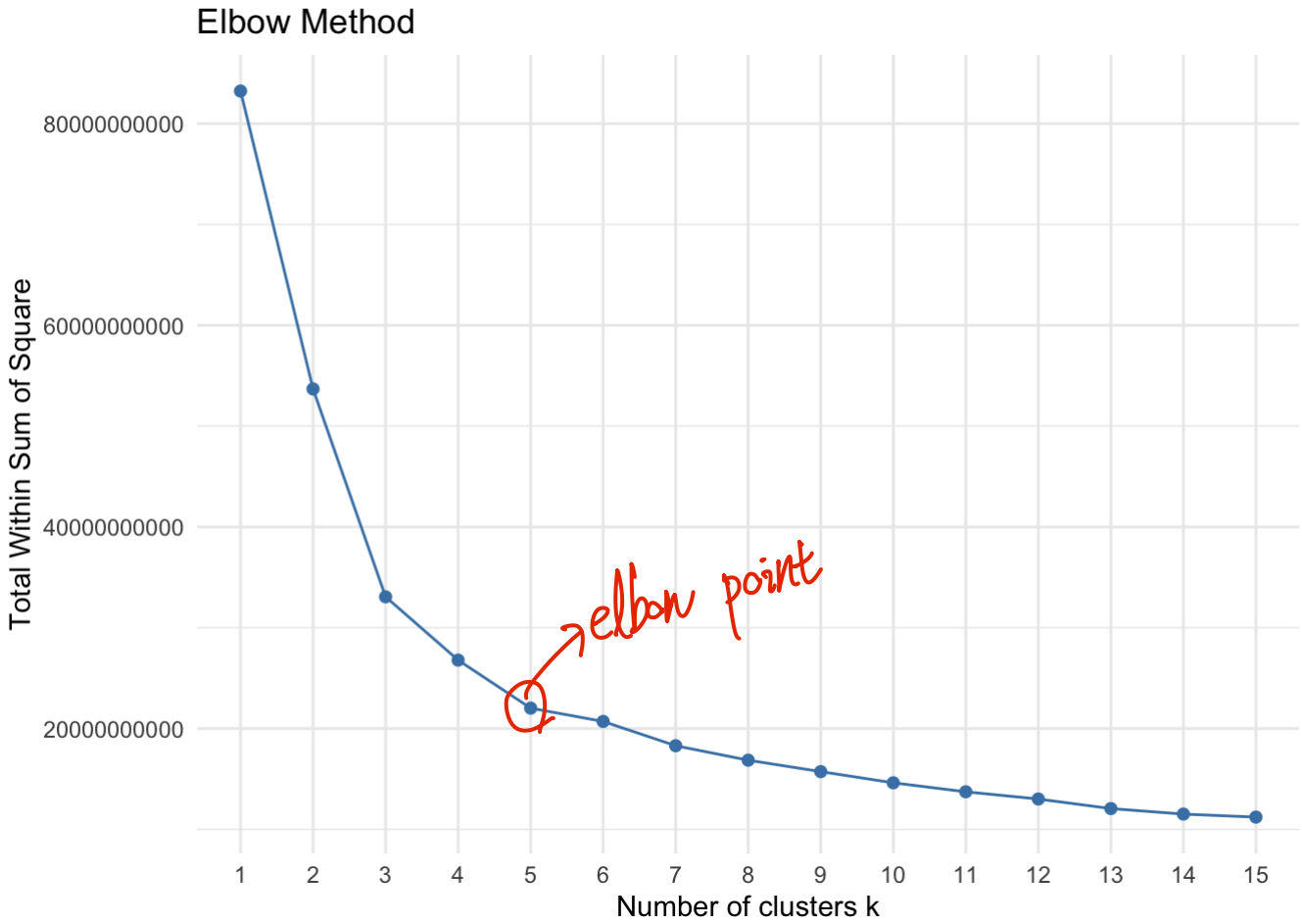
데이터

within sum of square

15개 군집까지 그래프로 나타내는 의미

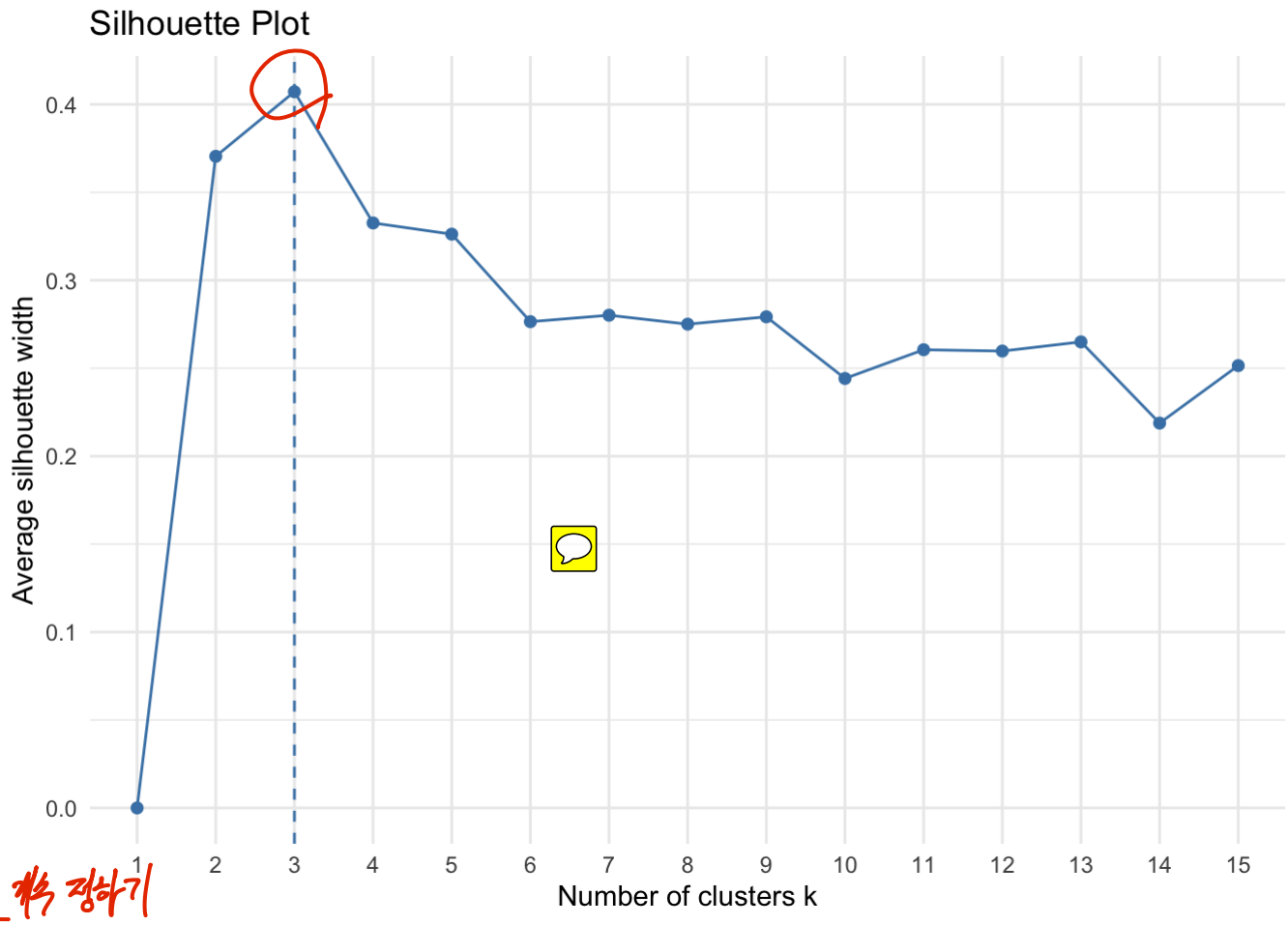
그래프의 테마

그래프의 제목



2. k 군집 개수 설정 (Silhouette method)

```
fviz_nbclust(df.rm.outlier[,3:ncol(df.rm.outlier)], kmeans, method = "silhouette", k.
max = 15) +
  theme_minimal() +
  ggtitle("Silhouette Plot")
```



k 개수 정하기

3. k means 모델 생성

```
df.kmeans <- kmeans(df.rm.outlier[,3:ncol(df.rm.outlier)], centers = 5, iter.max =100
0 )
df.kmeans
```

결정한 k값

군집화를 하고
재군집화를 하는 과정
몇번 반복할 것인지

각 군집에 할당된 데이터 개수

```
## K-means clustering with 5 clusters of sizes 179, 42, 72, 110, 18
##
## Cluster means:
##        Fresh      Milk   Grocery   Frozen Detergents_Paper Delicassen
## 1  4267.933  3751.480  4672.950 2211.313        1550.4469   1036.006
## 2 25332.000  5603.548  7160.024 4144.667        1449.2381   2053.333
## 3  5152.250 12536.694 19616.472 1644.014        8794.1389   1696.653
## 4 14527.509  2606.064  3503.873 3202.073         804.8091   1037.882
## 5 40558.056  3113.444  3814.333 2974.833         684.2778   1271.333
##
## Clustering vector:
##   [1] 4 1 1 4 2 1 4 1 1 3 1 4 2 2 2 4 1 1 2 1 4 1 2 2 4 4 4 3 5 2 1 4 2 1 1 2 3
##  [38] 3 2 4 3 3 1 3 3 4 3 1 1 5 3 2 1 3 3 4 1 1 1 3 1 1 2 1 1 4 1 2 1 4 1 3 4 1
##  [75] 1 3 1 4 4 1 2 4 4 3 3 1 1 1 1 4 3 3 1 4 4 1 3 1 3 4 3 4 4 4 4 4 1 4 1 4 1
## [112] 4 4 5 4 2 1 5 1 1 4 4 1 1 1 1 4 1 4 2 5 4 4 3 1 1 1 5 4 1 4 1 1 3 3 4 1 3
## [149] 1 4 4 3 1 3 1 1 1 1 3 3 1 3 1 1 5 4 4 1 4 1 1 1 1 1 3 3 4 4 1 3 1 4 1 4 4
## [186] 3 3 2 1 1 3 1 1 1 3 4 3 1 1 1 3 3 4 3 1 4 1 1 1 1 4 2 1 1 1 4 1 2 1 4 1 1
## [223] 4 1 5 2 2 4 4 1 3 1 4 4 1 1 3 1 2 1 5 4 1 5 1 1 2 1 3 3 3 4 3 4 1 1 1 5 1
## [260] 1 2 4 4 4 1 4 5 2 5 1 4 4 5 1 1 1 3 2 1 4 1 1 1 4 3 1 3 3 1 3 4 1 3 1 2 3
## [297] 4 4 3 1 1 4 3 1 1 4 4 2 1 1 4 1 4 3 2 4 2 4 4 1 1 1 1 1 3 1 1 3 2 1 3 1 3
## [334] 1 3 4 1 4 3 1 1 4 1 1 1 1 1 4 1 2 1 5 4 1 4 1 1 3 5 1 1 2 4 5 1 3 4 1 4 4
## [371] 4 1 1 1 2 4 4 1 4 4 4 1 2 2 2 4 1 2 3 1 1 1 1 1 1 1 1 3 1 3 1 3 4 2 4 4 4
## [408] 3 2 1 1 1 1 4 1 4 2 5 3 4 1
##
## Within cluster sum of squares by cluster:
## [1] 7488224454 2823135964 9143410363 3900150510  861057236
##  (between_SS / total_SS =  70.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

각 군집의 평균값

⇒ 각 데이터들이 몇 번째 군집에 할당이 되었는지 의미

4. 군집별 평균치 시각화

F : stacked bar

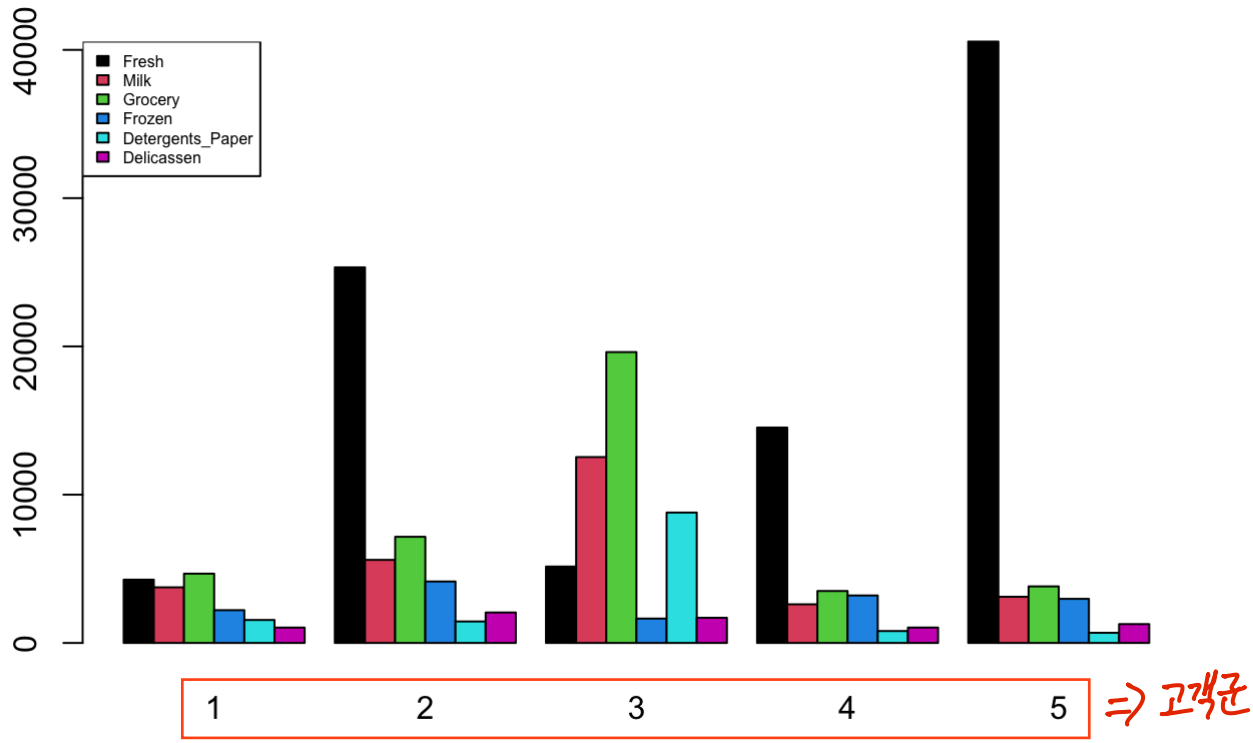
T : juxtaposed bar

```
barplot(t(df.kmeans$centers), beside=TRUE, col = 1:6)
legend("topleft", colnames(df[,3:8]), fill = 1:6, cex = 0.5)
```

범례 생성

색을 6개로 채워라

범례 크기



5. raw data에 cluster 할당 (행별로 클러스터 번호 할당)

```
df.rm.outlier$cluster <- df.kmeans$cluster
head(df.rm.outlier)
```

```
##   Channel Region Fresh Milk Grocery Frozen Detergents_Paper Delicassen cluster
## 1       2      3 12669 9656    7561    214             2674        1338       4
## 2       2      3  7057 9810    9568   1762             3293        1776       1
## 3       2      3  6353 8808    7684   2405             3516        7844       1
## 4       1      3 13265 1196    4221   6404              507        1788       4
## 5       2      3 22615 5410    7198   3915             1777        5185       2
## 6       2      3  9413 8259    5126    666             1795        1451       1
```