Cluster Analysis of Categorical Data

Xiaoyi Ma and Robert Long

4/14/2021

Contents

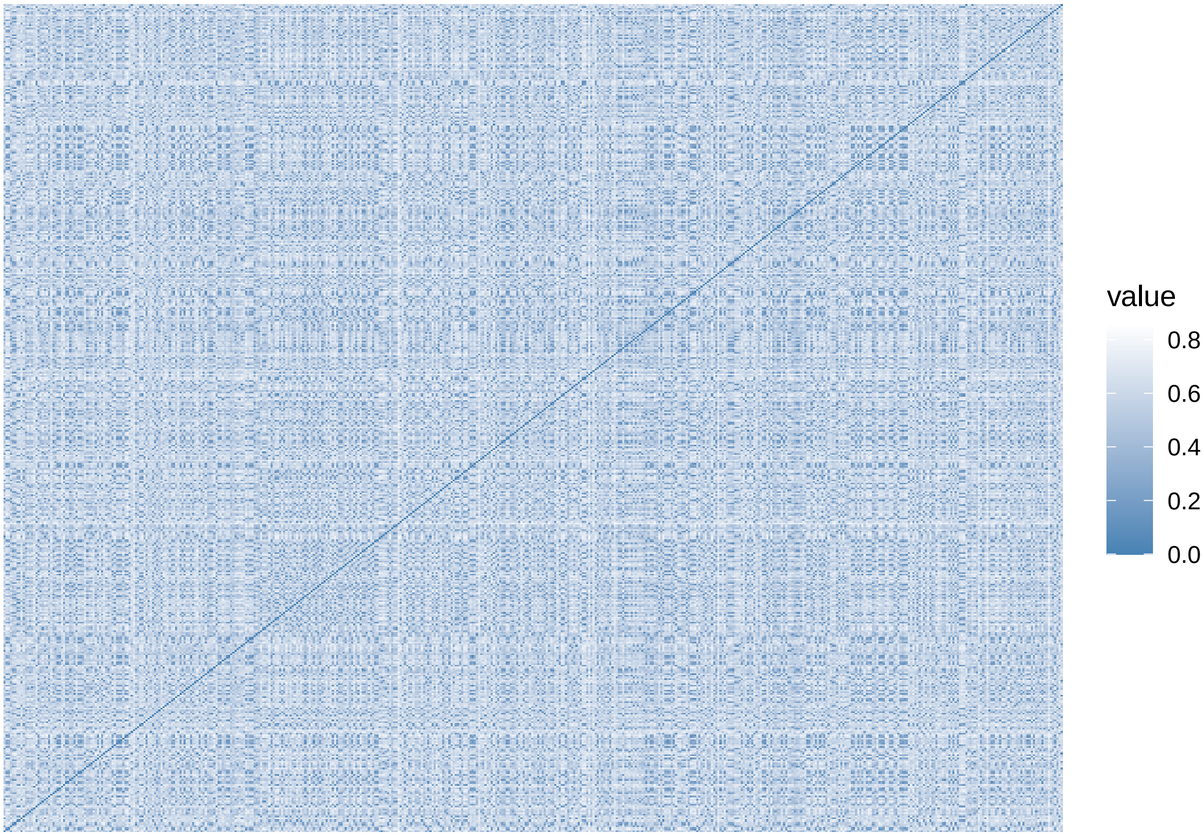
```r
data <- read.csv("mushrooms.csv")

data[sapply(data, is.character)] <- lapply(data[sapply(data, is.character)], as.factor)

str(data)
```

```
## 'data.frame':    8124 obs. of  23 variables:
##  $ class                   : Factor w/ 2 levels "e","p": 2 1 1 2 1 1 1 1 2 1 ...
##  $ cap.shape               : Factor w/ 6 levels "b","c","f","k",..: 6 6 1 6 6 6 1 1 6 1 ...
##  $ cap.surface             : Factor w/ 4 levels "f","g","s","y": 3 3 3 4 3 4 3 4 4 3 ...
##  $ cap.color               : Factor w/ 10 levels "b","c","e","g",..: 5 10 9 9 4 10 9 9 9 10 ...
##  $ bruises                 : Factor w/ 2 levels "f","t": 2 2 2 2 1 2 2 2 2 2 ...
##  $ odor                    : Factor w/ 9 levels "a","c","f","l",..: 7 1 4 7 6 1 1 4 7 1 ...
##  $ gill.attachment         : Factor w/ 2 levels "a","f": 2 2 2 2 2 2 2 2 2 2 ...
##  $ gill.spacing            : Factor w/ 2 levels "c","w": 1 1 1 1 2 1 1 1 1 1 ...
##  $ gill.size               : Factor w/ 2 levels "b","n": 2 1 1 2 1 1 1 1 2 1 ...
##  $ gill.color              : Factor w/ 12 levels "b","e","g","h",..: 5 5 6 6 5 6 3 6 8 3 ...
##  $ stalk.shape             : Factor w/ 2 levels "e","t": 1 1 1 1 2 1 1 1 1 1 ...
##  $ stalk.root              : Factor w/ 5 levels "?","b","c","e",..: 4 3 3 4 4 3 3 3 4 3 ...
##  $ stalk.surface.above.ring: Factor w/ 4 levels "f","k","s","y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ stalk.surface.below.ring: Factor w/ 4 levels "f","k","s","y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ stalk.color.above.ring  : Factor w/ 9 levels "b","c","e","g",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ stalk.color.below.ring  : Factor w/ 9 levels "b","c","e","g",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ veil.type               : Factor w/ 1 level "p": 1 1 1 1 1 1 1 1 1 1 ...
##  $ veil.color              : Factor w/ 4 levels "n","o","w","y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ ring.number             : Factor w/ 3 levels "n","o","t": 2 2 2 2 2 2 2 2 2 2 ...
##  $ ring.type               : Factor w/ 5 levels "e","f","l","n",..: 5 5 5 5 1 5 5 5 5 5 ...
##  $ spore.print.color       : Factor w/ 9 levels "b","h","k","n",..: 3 4 4 3 4 3 3 4 3 3 ...
##  $ population              : Factor w/ 6 levels "a","c","n","s",..: 4 3 3 4 1 3 3 4 5 4 ...
##  $ habitat                 : Factor w/ 7 levels "d","g","l","m",..: 6 2 4 6 2 2 4 4 2 4 ...
```

```
x.data <- subset(data, select=-c(veil.type, class))
```

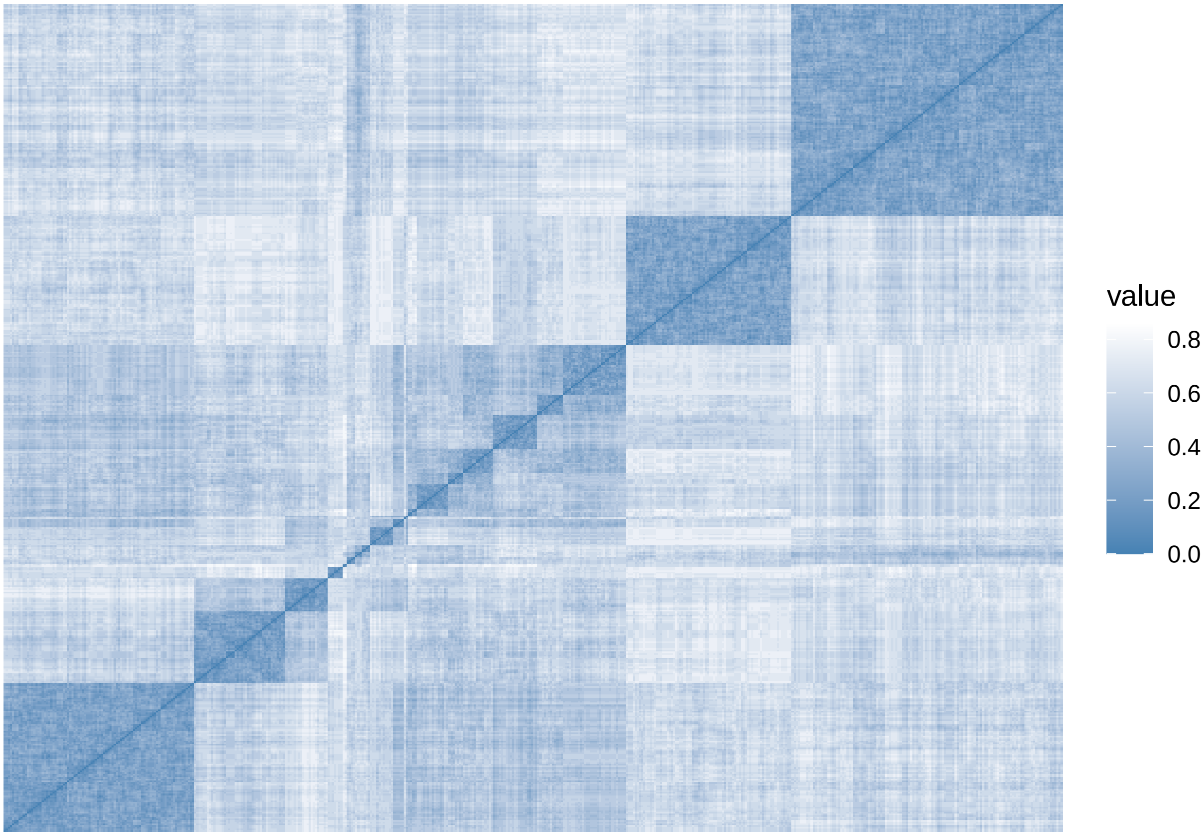
Distance Matrix

```
samps <- sample(8124, 500)

data.dist <- daisy(x.data[samps,], metric="gower")

gradient.color <- list(low = "steelblue",  high = "white")
fviz_dist(data.dist,
   gradient = gradient.color,
   order=F,
   show_labels=F)
```



Ordered Distance Matrix

```
gradient.color <- list(low = "steelblue",  high = "white")
fviz_dist(data.dist,
   gradient = gradient.color,
   order=T,
   show_labels=F)
```

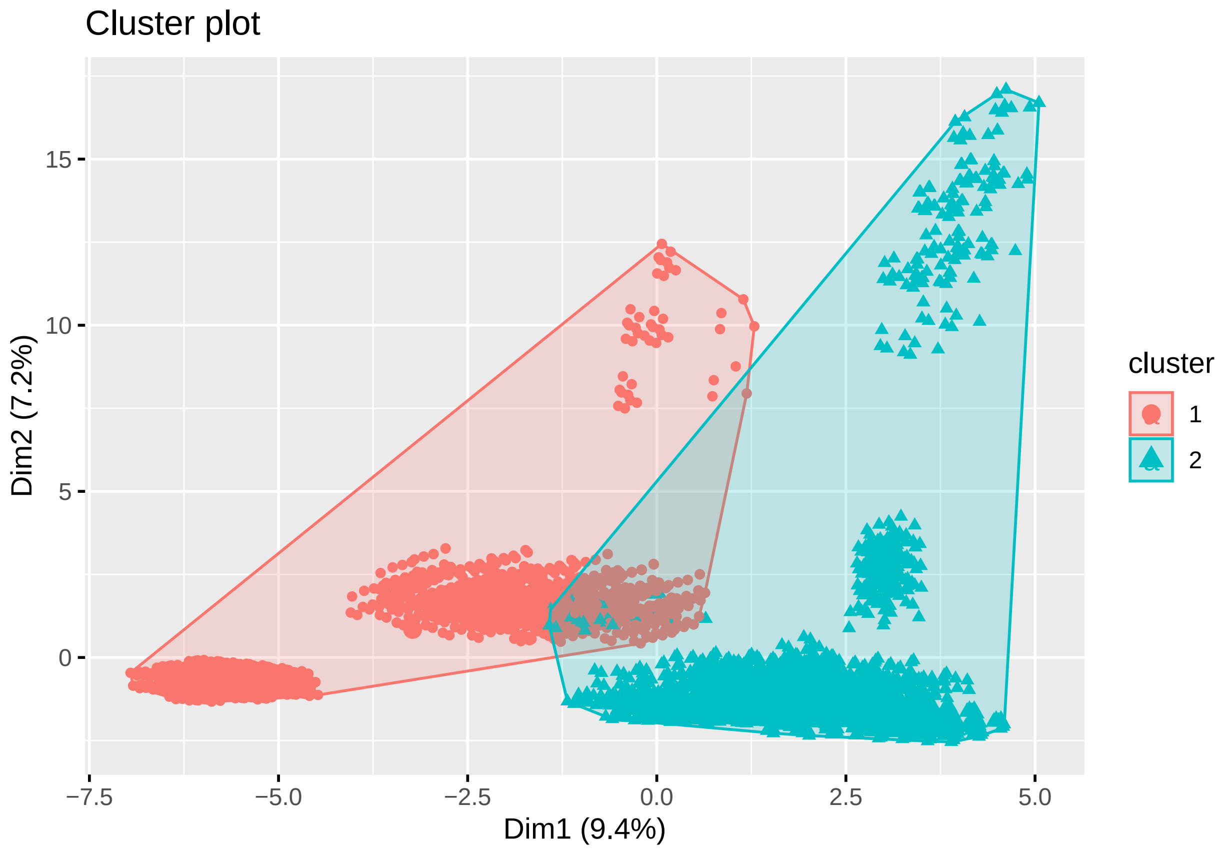


k-means Clustering

```
data.matrix <- model.matrix(~.-1, data=x.data)

fit.kmean = kmeans(data.matrix, 2, iter.max = 15)

result.kmean.mm <- table(data$class, fit.kmean$cluster)
result.kmean.mm
```

```
##
##        1    2
##   e   32 4176
##   p 3098  818
```

```
purity.kmean <- sum(apply(result.kmean.mm, 2, max)) / nrow(x.data)

fviz_cluster(fit.kmean, data.matrix, repel=T)
```



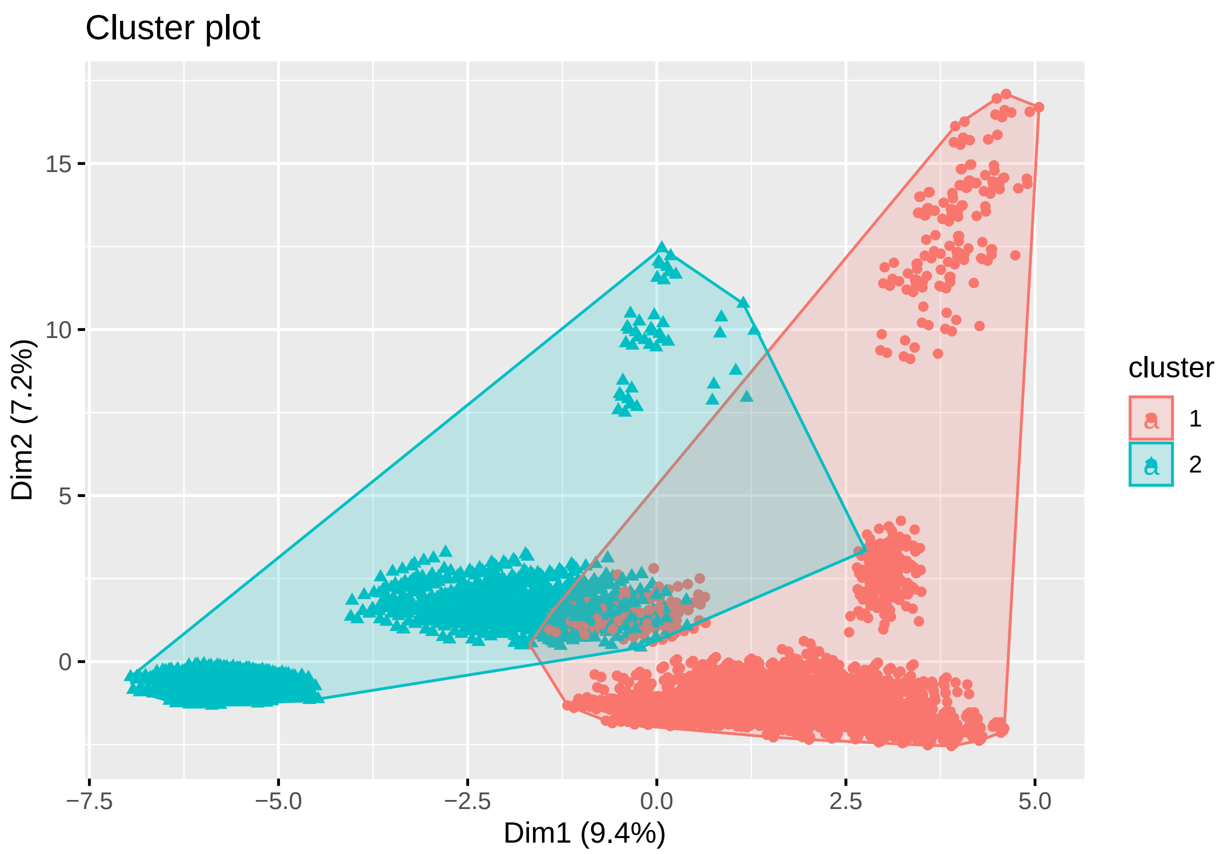
Purity: 0.8953717

PAM

```
fit.pam <- pam(data.matrix, 2)

result.pam.mm <- table(data$class, fit.pam$clustering)
result.pam.mm
```

```
##
##        1    2
##   e 4206    2
##   p  982 2934
```

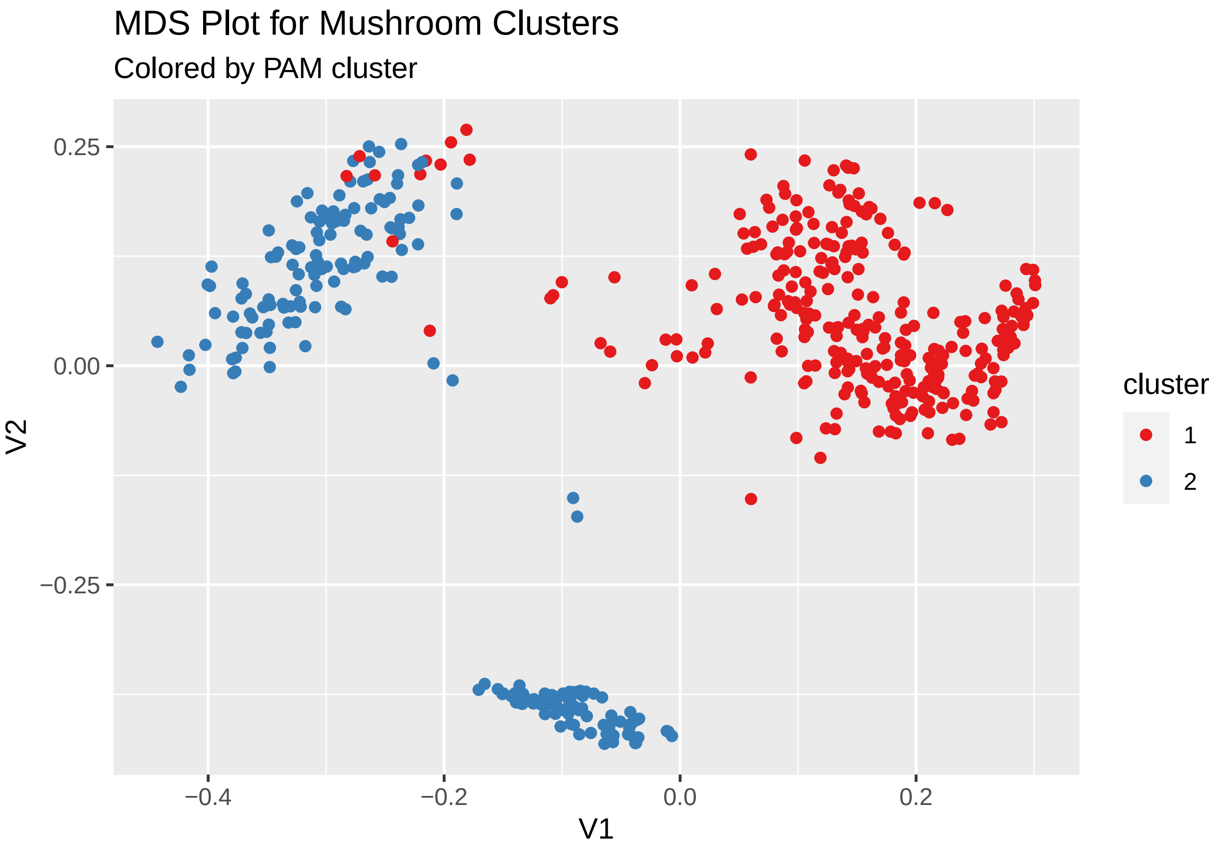
```
purity.pam <- sum(apply(result.pam.mm, 2, max)) / nrow(x.data)

fviz_cluster(fit.pam, repel=T)
```



Purity: 0.8788774

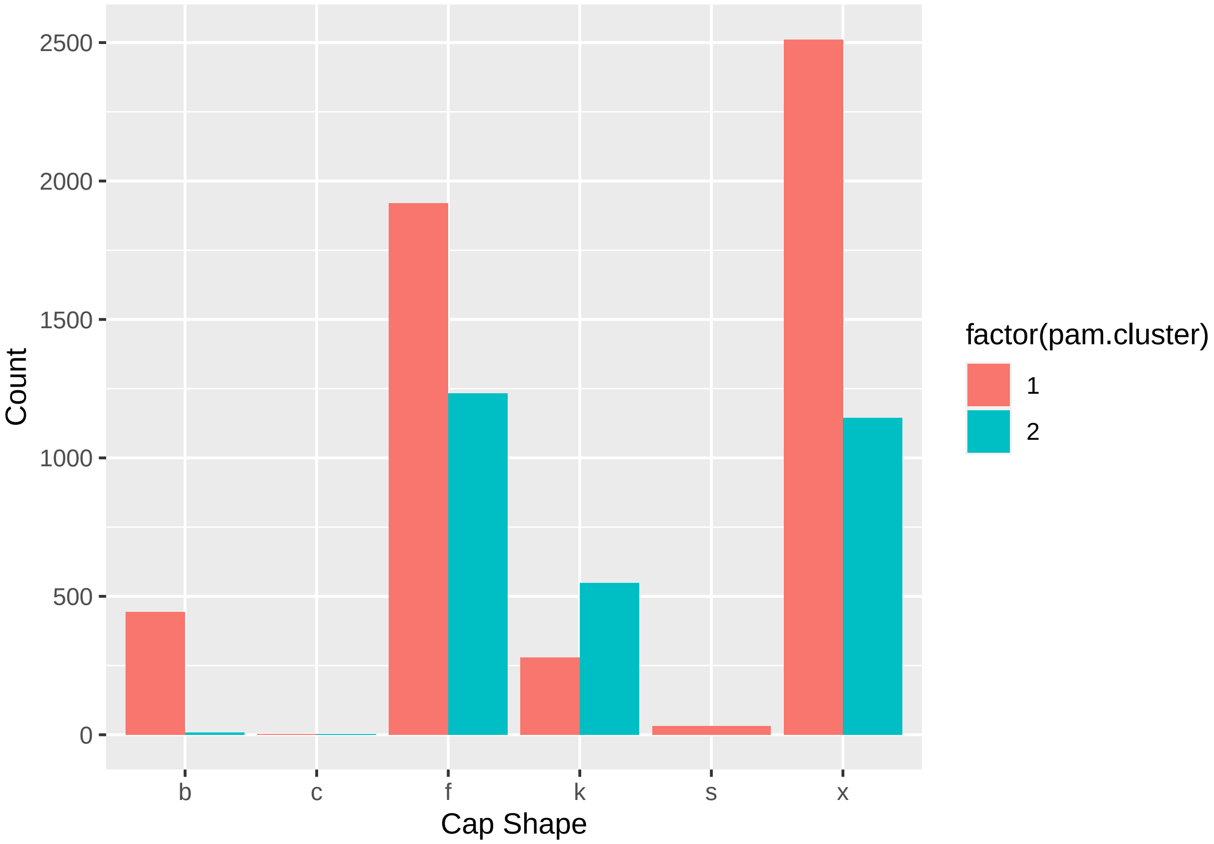
PAM: MDS and Clusters

```
pam.mds <- as.data.frame(cmdscale(data.dist,2))

pam.mds$cluster <- as.factor(fit.pam$clustering[samps])

ggplot(pam.mds,
       aes(x=V1, y=V2, color=cluster)) +
  geom_point() +
  labs(title="MDS Plot for Mushroom Clusters",
       subtitle="Colored by PAM cluster") +
  scale_color_brewer(palette="Set1")
```



PAM Results

```
pam.cluster <- fit.pam$clustering

data.fused <- cbind(data, pam.cluster)

ggplot(data.fused) +
  geom_bar(aes(x=cap.shape, fill=factor(pam.cluster)), position="dodge") +
  xlab("Cap Shape") +
  ylab("Count")
```


Count
2500
2000
1500
1000
500
0
b
c
f
k
s
x
Cap Shape
factor(pam.cluster)
1
2