Segmentation group assignment

2023-06-30

Load library and read file

```r
library(dplyr)
customers = read.csv('/Users/mandy/Desktop/r/HW1/Wholesale customers data.csv',sep = ',')
```

Load library and read csv file

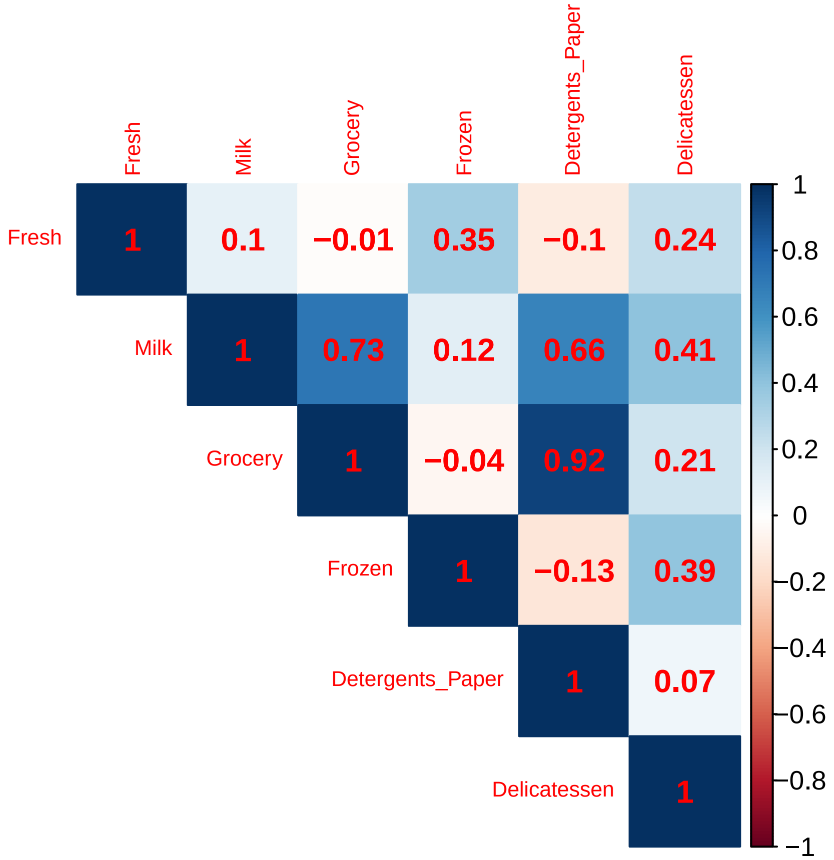
```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
options(repos = "https://cran.rstudio.com")
chooseCRANmirror(ind = 77)
features = customers[, c("Fresh", "Milk", "Grocery", "Frozen", "Detergents_Paper", "Delicatessen")]
cor_matrix <- cor(features)
cor_matrix
```

```
##                        Fresh      Milk     Grocery      Frozen Detergents_Paper
## Fresh             1.00000000 0.1005098 -0.01185387  0.34588146       -0.1019529
## Milk              0.10050977 1.0000000  0.72833512  0.12399376        0.6618157
## Grocery          -0.01185387 0.7283351  1.00000000 -0.04019274        0.9246407
## Frozen            0.34588146 0.1239938 -0.04019274  1.00000000       -0.1315249
## Detergents_Paper -0.10195294 0.6618157  0.92464069 -0.13152491        1.0000000
## Delicatessen      0.24468997 0.4063683  0.20549651  0.39094747        0.0692913
##                  Delicatessen
## Fresh               0.2446900
## Milk                0.4063683
## Grocery             0.2054965
## Frozen              0.3909475
## Detergents_Paper    0.0692913
## Delicatessen        1.0000000
```

```r
corrplot(cor_matrix, method = "color", type = "upper", tl.cex = 0.7,addCoef.col = "Red")
```



Normalize features

```
normalize = function(x){
  return ((x - min(x))/(max(x) - min(x)))}
features_normalized = features %>%
  mutate(Fresh_n = normalize(Fresh),
         Milk_n = normalize(Milk),
         Grocery_n = normalize(Grocery),
         Frozen_n = normalize(Frozen),
         Detergents_Paper_n = normalize(Detergents_Paper),
         Delicatessen_n = normalize(Delicatessen))
features_normalized = features_normalized[, c("Fresh_n", "Milk_n", "Grocery_n",
                                              "Frozen_n", "Detergents_Paper_n",
                                              "Delicatessen_n")]
```

```
library(ggplot2)
head(features_normalized)
```

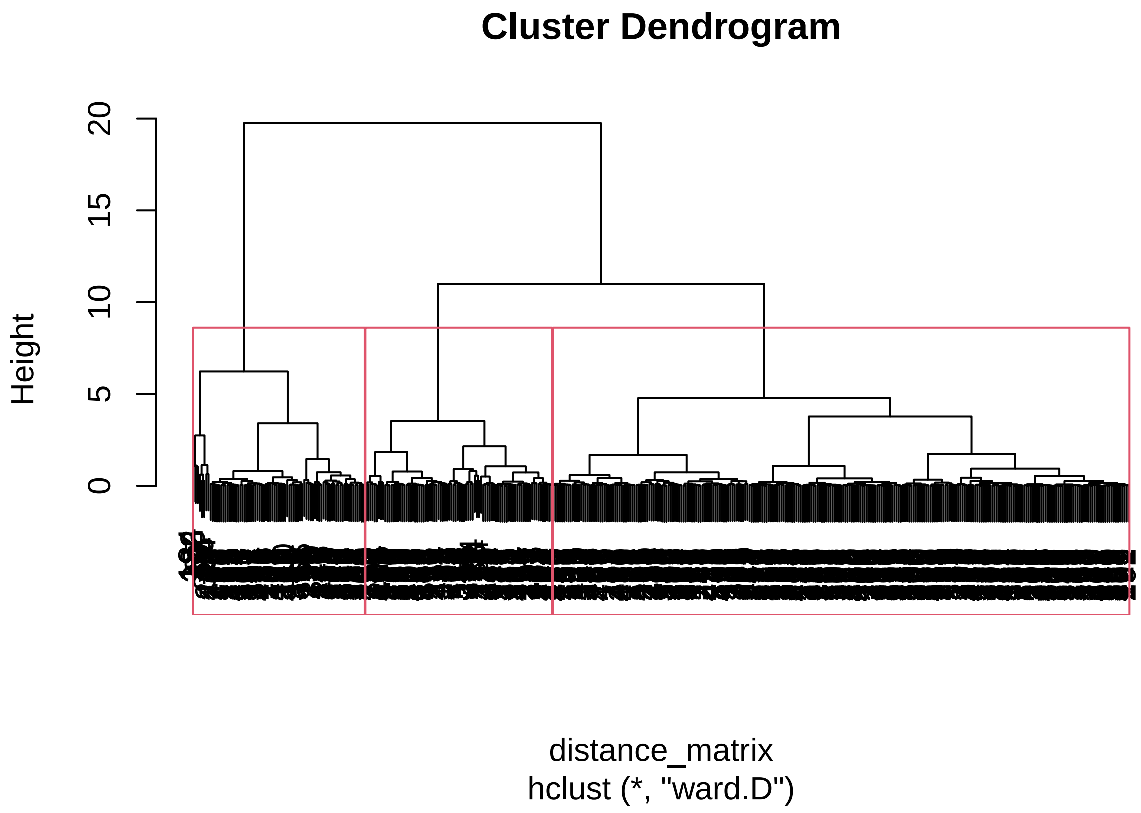
```
##      Fresh_n     Milk_n  Grocery_n    Frozen_n Detergents_Paper_n
## 1 0.11294004 0.13072723 0.08146416 0.003106305         0.06542720
## 2 0.06289903 0.13282409 0.10309667 0.028548419         0.08058985
## 3 0.05662161 0.11918086 0.08278992 0.039116429         0.08605232
## 4 0.11825445 0.01553586 0.04546385 0.104841891         0.01234568
## 5 0.20162642 0.07291369 0.07755155 0.063933995         0.04345483
## 6 0.08390698 0.11170568 0.05521843 0.010535139         0.04389575
##   Delicatessen_n
## 1     0.02784731
## 2     0.03698373
## 3     0.16355861
## 4     0.03723404
## 5     0.10809345
## 6     0.03020442
```

Clustering

```
library(stats)
distance_matrix = dist(features_normalized, method = "euclidean")
hierarchical = hclust(distance_matrix,method = "ward.D")
features_normalized$cluster = cutree(hierarchical, k=3)
```

Dendogram with 3 cut solution

```
plot(hierarchical, labels = features_normalized$Name)
rect.hclust(hierarchical, k = 3)
```



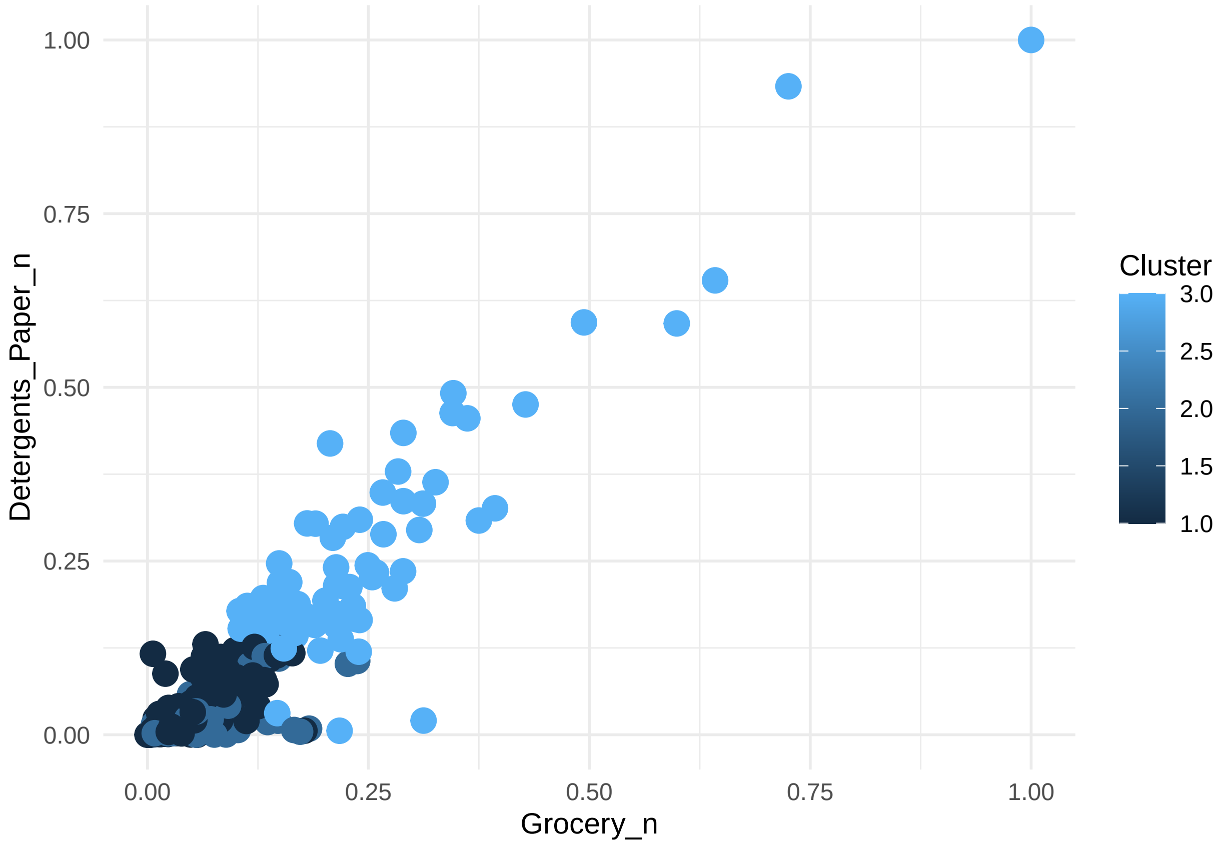
Cluster centroids

```
features_normalized %>% group_by(cluster) %>%
summarise_at(c(1:6), mean)
```

```
## # A tibble: 3 x 7
##   cluster Fresh_n Milk_n Grocery_n Frozen_n Detergents_Paper_n Delicatessen_n
##     <int>   <dbl>  <dbl>     <dbl>    <dbl>              <dbl>          <dbl>
## 1       1  0.0727 0.0439    0.0471   0.0318             0.0309         0.0218
## 2       2  0.243  0.0740    0.0627   0.110              0.0252         0.0480
## 3       3  0.0737 0.197     0.240    0.0464             0.252          0.0473
```

Quick plot

```
ggplot(features_normalized, aes(Grocery_n, Detergents_Paper_n, color = cluster)) + geom_point(size = 4)
```



```
features$cluster <- as.factor(features_normalized$cluster)
features_distribution = features %>% group_by(cluster) %>% count(cluster)
features_distribution
```

```
## # A tibble: 3 x 2
## # Groups:   cluster [3]
##   cluster     n
##   <fct>   <int>
## 1 1         271
## 2 2          88
## 3 3          81
```

Export results

```
write.csv(features, "First three cluster solution v1.csv")
```

further filering for cluster 1

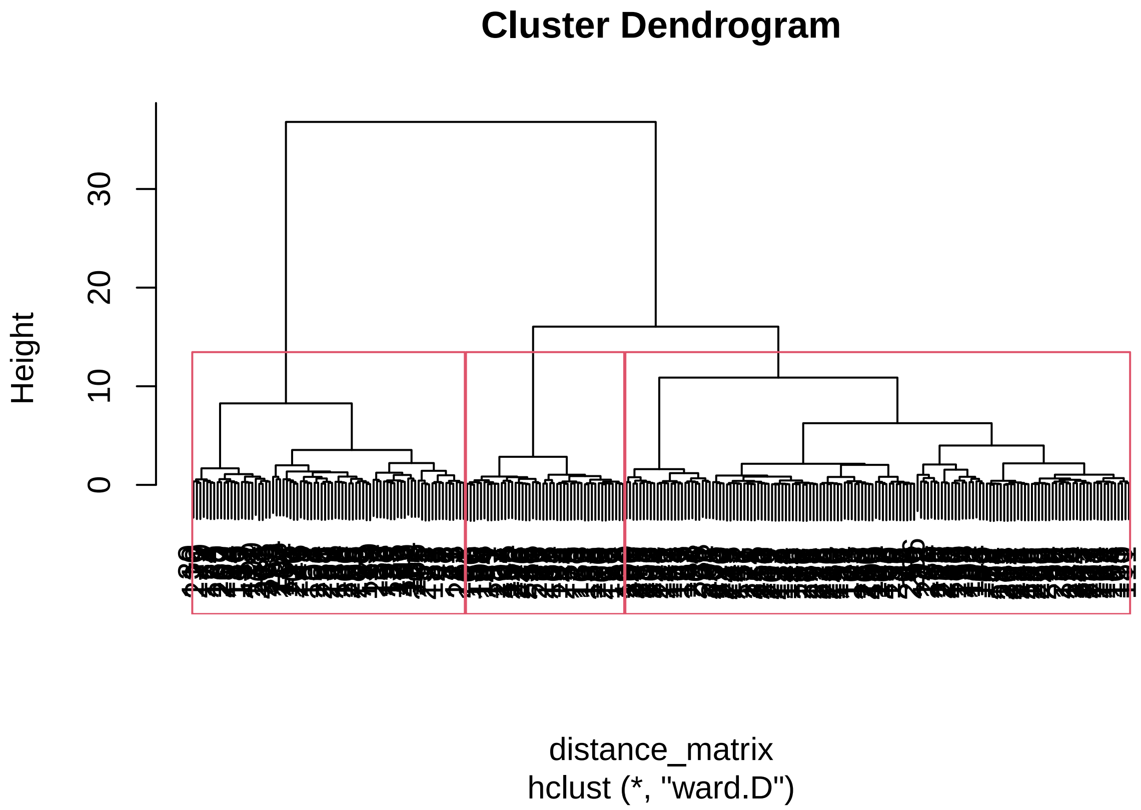
```
data_further = features %>% filter(cluster == 1)
features_normalized_further = data_further %>%
  mutate(Fresh_n = normalize(Fresh),
         Milk_n = normalize(Milk),
         Grocery_n = normalize(Grocery),
         Frozen_n = normalize(Frozen),
         Detergents_Paper_n = normalize(Detergents_Paper),
         Delicatessen_n = normalize(Delicatessen))
features_normalized_further = features_normalized_further[, c("Fresh_n", "Milk_n", "Grocery_n",
                                              "Frozen_n", "Detergents_Paper_n",
                                              "Delicatessen_n")]
head(features_normalized_further)
```

```
##     Fresh_n    Milk_n Grocery_n   Frozen_n Detergents_Paper_n Delicatessen_n
## 1 0.5446805 0.6431969 0.4586165 0.01723782          0.5027292     0.17025890
## 2 0.3033457 0.6535138 0.5804005 0.17702312          0.6192358     0.22611912
## 3 0.2730713 0.5863871 0.4660801 0.24339389          0.6612084     1.00000000
## 4 0.4046616 0.5496081 0.3108617 0.06389348          0.3372859     0.18467032
## 5 0.5213297 0.2106250 0.4230583 0.04469447          0.5904385     0.06912384
## 6 0.3257934 0.3283312 0.5717840 0.16742362          0.6245059     0.32687157
```

Further segmenting the majority cluster (cluster 1)

```
distance_matrix = dist(features_normalized_further, method = "euclidean")
hierarchical = hclust(distance_matrix,method = "ward.D")

plot(hierarchical, labels = features_normalized_further$Name)
rect.hclust(hierarchical, k = 3)
```



```
features_normalized_further$cluster_further = cutree(hierarchical, k=3)
data_further$cluster_further <- as.factor(features_normalized_further$cluster)
features_distribution_further = data_further %>% group_by(cluster_further) %>% count(cluster)
features_distribution_further
```

```
## # A tibble: 3 x 3
## # Groups:   cluster_further [3]
##   cluster_further cluster     n
##   <fct>           <fct>   <int>
## 1 1               1          79
## 2 2               1          46
## 3 3               1         146
```

```
write.csv(data_further, "Second three cluster solution v2.csv")
```