Group 4 - Homework 1 R Code and Procedure

Setup

Reading the Wholesale Customer Data into the variable ‘wholesale’:

```
library(dplyr)
```

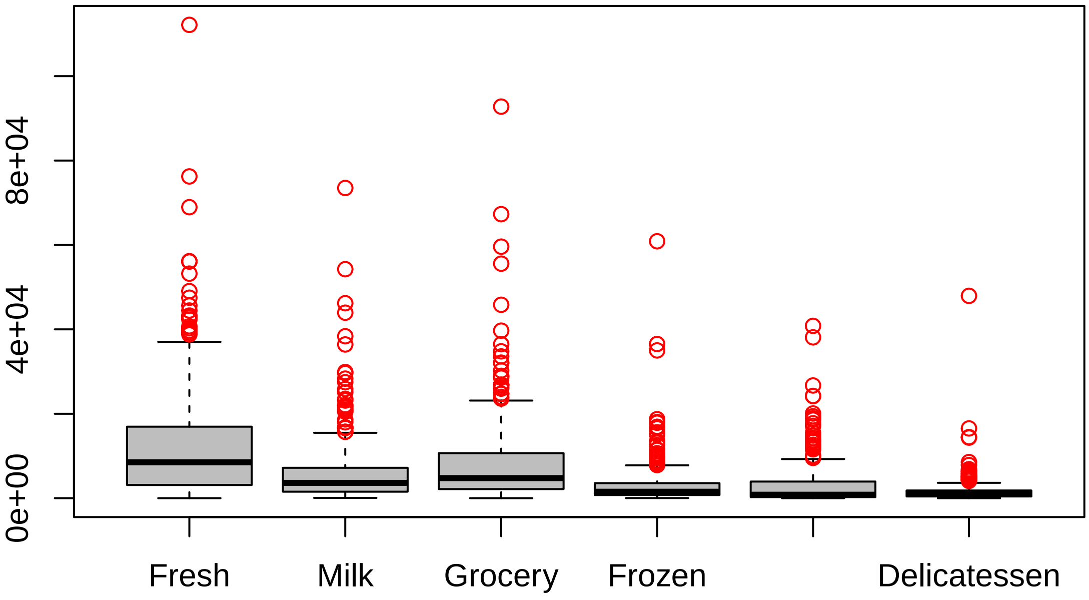
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
wholesale = read.csv("Wholesale_customers_data.csv")
```

Initial understanding of the data

Viewing the table and boxplots for numerical columns:

```
boxplot(wholesale[3:8], col="grey", outcol="red", las =0.5)
```



Outlier Removal

After viewing and visualizing the data points, we could clearly see that there were a few outliers that would create noise in the clustering solutions. Observing that almost all the outliers were spending substantially higher than the other data points, we decided to omit them out and possibly consider them as a separate cluster later.

At this point, we decided to find outliers within each category and count the number of outlier values each client had.

Creating a function to detect outliers in each column, and adding a column that counts total number of outliers for each row:

```
outlier <- function(x) {
  return (ifelse(x < (quantile(x, 0.25) - 1.5 * IQR(x)) |
                           x > (quantile(x, 0.75) + 1.5 * IQR(x)), 1,0))
}
wholesale <- wholesale %>%
  mutate(Outliers = outlier(Fresh) + outlier(Milk) + outlier(Grocery) + outlier(Frozen)
         + outlier(Detergents_Paper) + outlier(Delicatessen))
```

To decide on the criteria to remove clients with outliers, we ran code lines 42 to 94 multiple times with different Outlier counts to get high silhouette coefficients. After repeating this process for the same number of clusters, we found that removing clients 2 or more outliers gave the best clustering solution with good silhoutte coefficients.

Removing clients with 2 or more outliers:

```r
#storing outliers in 'wholesale_out'
wholesale_out = wholesale[wholesale$Outliers>1,]
wholesale = wholesale[wholesale$Outliers<2,]
```

Clustering

Starting the clustering process with normalization:

```r
normalize <- function(x) {
  return ((x - min(x))/(max(x) - min(x)))
}
wholesaleDataNormalized <- wholesale %>% mutate_at(c(3:8), normalize)
```

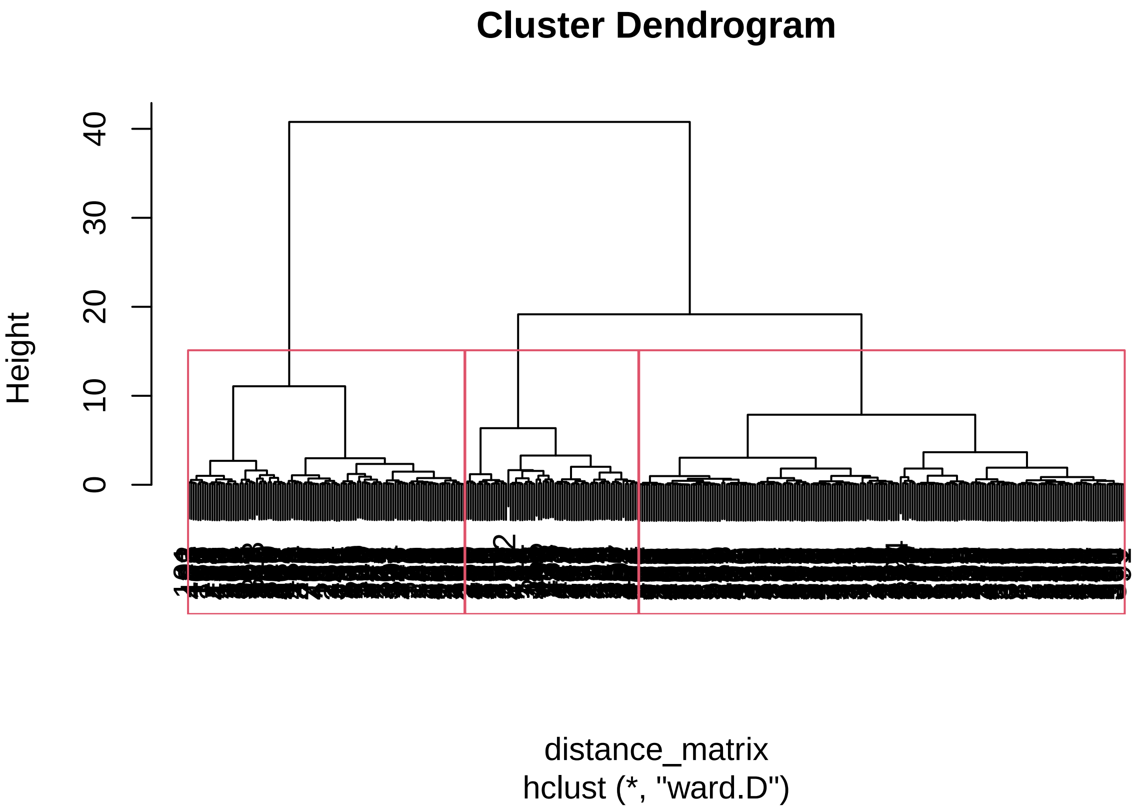
Calulating the distance matrix with Euclidean method:

```r
library(stats)
distance_matrix = dist(wholesaleDataNormalized[,3:8], method = "euclidean")
```

Hierarchical clustering

Trying Hierarchical clustering, primarily to create a dendogram to deduce ideal number of clusters to consider (repeated this step for multiple k values):

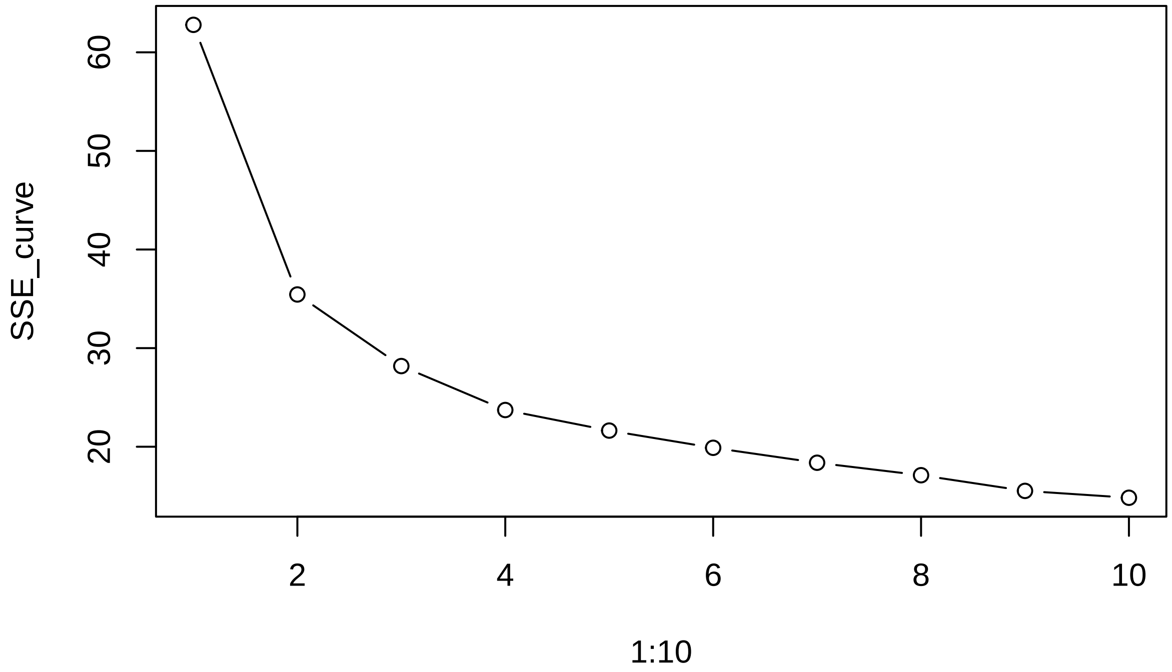
```r
hierarchical = hclust(distance_matrix, method = "ward.D")
plot(hierarchical)
rect.hclust(hierarchical, k = 3)
```



A k value of 2,3, or 4 seem to be ideal.

Plotting an SSE curve to detect an elbow point:

```
SSE_curve <- c()
for (n in 1:10) {
  kcluster1 = kmeans(wholesaleDataNormalized[,3:8], n)
  sse = kcluster1$tot.withinss
  SSE_curve[n] = sse}
# plot SSE against number of clusters
plot(1:10, SSE_curve, type = "b")
```



This again reinforces value of k to be 2,3, or 4.

K-Means Clustering

We started performing K-Means clustering. Also repeating this clustering for k = 2,3, and 4, while checking if centers are meaningful and if silhouette coefficients are high.

```
kcluster = kmeans(wholesaleDataNormalized[,3:8], centers = 3)
kcluster$centers
```

```
##       Fresh      Milk   Grocery     Frozen Detergents_Paper Delicatessen
## 1 0.1526372 0.1197507 0.1337180 0.08224870       0.04465210   0.06174272
## 2 0.1013300 0.4129266 0.5759462 0.04041199       0.34441373   0.10997347
## 3 0.5300413 0.2210388 0.2451525 0.10776450       0.05959962   0.12773055
```

```
wholesale = wholesale %>% mutate(Cluster = kcluster$cluster)
wholesaleDataNormalized = wholesaleDataNormalized %>% mutate(Cluster = kcluster$cluster)

library(cluster)
sc = silhouette(wholesaleDataNormalized$Cluster, dist = distance_matrix)
summary(sc)
```

```
## Silhouette of 399 units in 3 clusters from silhouette.default(x = wholesaleDataNormalized$Cluster, di
##  Cluster sizes and average silhouette widths:
##       242        97        60
## 0.4544000 0.3274322 0.1710764
## Individual silhouette widths:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.1334  0.2437  0.4196  0.3809  0.5409  0.6285
```

We found that 3 clusters without outliers was the ideal solution with meaningful centroids (high spenders in horeca, high spenders in retail, and low spenders across both).

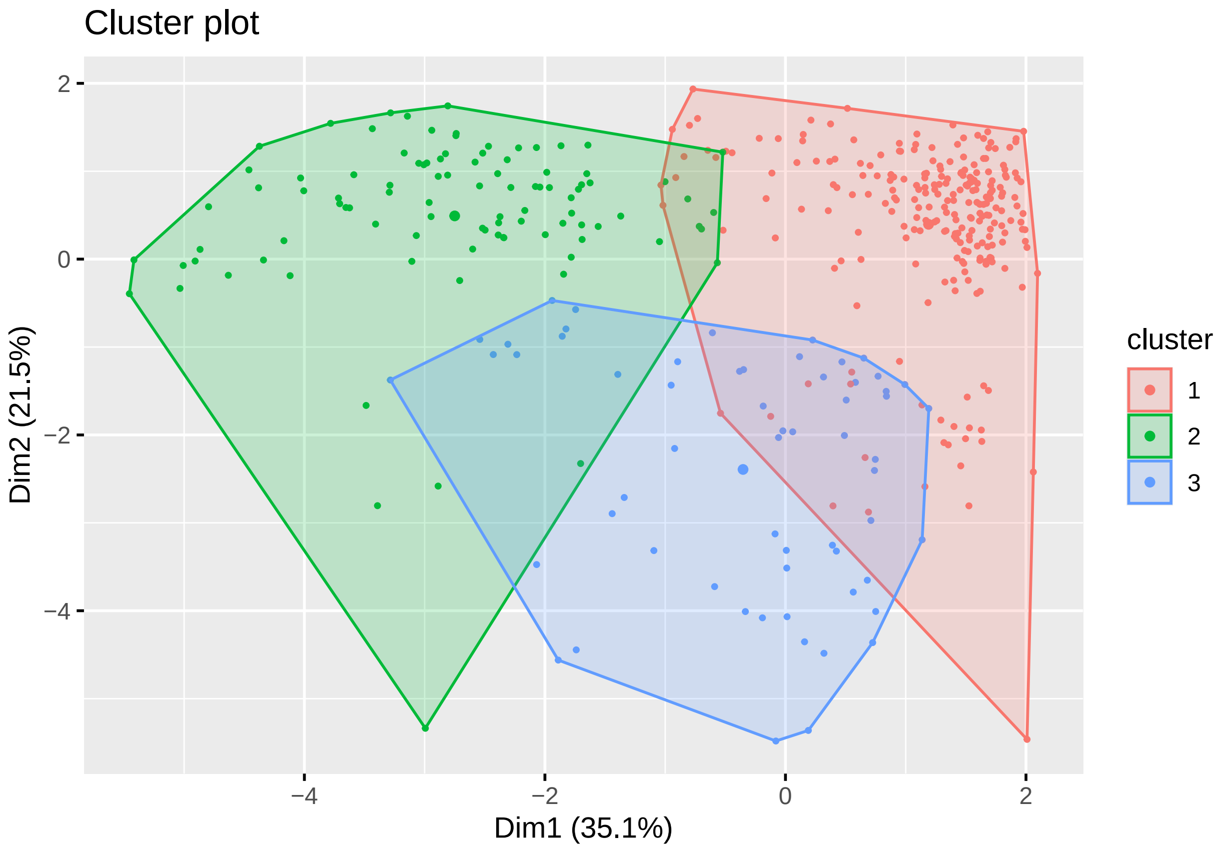
Visualizing these clusters:

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_cluster(kcluster, data = wholesaleDataNormalized, geom = "point",
             pointsize = 1, shape = 20, show.clust.cent = TRUE)
```



```
#clusplot(wholesaleDataNormalized, kcluster$cluster)
```

We decided that the outliers could form an extra cluster that is all highest spenders from huge corporations/establishments.

Adding this 4th cluster to the dataset:

```
wholesale_out = wholesale_out %>% mutate(Cluster = 4)
wholesale <- rbind(wholesale, wholesale_out)

wholesale %>% group_by(Cluster) %>% summarise(count = n())
```

```
## # A tibble: 4 x 2
##   Cluster count
##     <dbl> <int>
## 1       1   242
## 2       2    97
## 3       3    60
## 4       4    41
```

Exporting this dataset to see further relationships and cluster qualities in Excel using write.csv