# Assignment-4

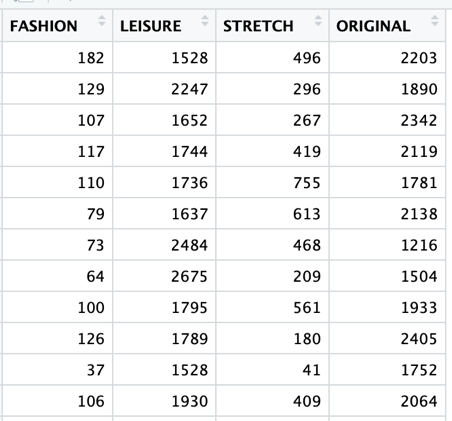
1.

(a) Import the data to R and remove the column(s) that you are not going to use. Copy the R code

used below. (0.25)

dungaree.df <- read.csv("dungaree.csv")

dungaree.df <- select(dungaree.df,-c(1,6)) #Removing the StoreID, and SalesSlot

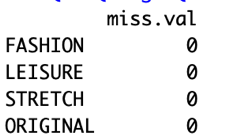


(b) Examine the input variables: Are there any unusual data values? Are there missing values that

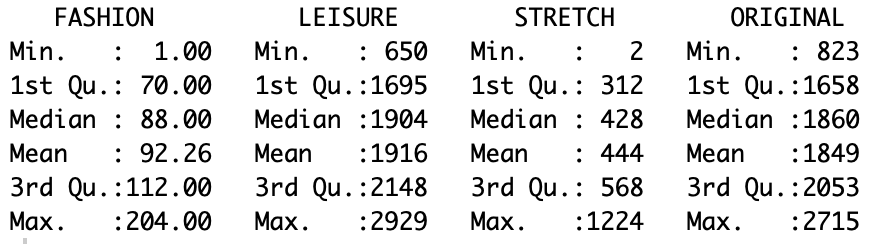
should be replaced? (0.5)

data.frame(miss.val=sapply(dungaree.df, function(x)

(sum(length(which(is.na(x)))))))



summary(dungaree.df)



boxplot(dungaree.df$FASHION)

boxplot(dungaree.df$LEISURE)

boxplot(dungaree.df$STRETCH)

boxplot(dungaree.df$ORIGINAL)

There are no missing values in the dataset and the range of FASHION and LEISURE are two wide range and variables with outliers.

(c) Normalize the data. Copy the R code used below. What would happen if you did not

standardize/normalize your inputs? (0.5 + 0.5 )

dungaree.df.norm <- sapply(dungaree.df, scale)

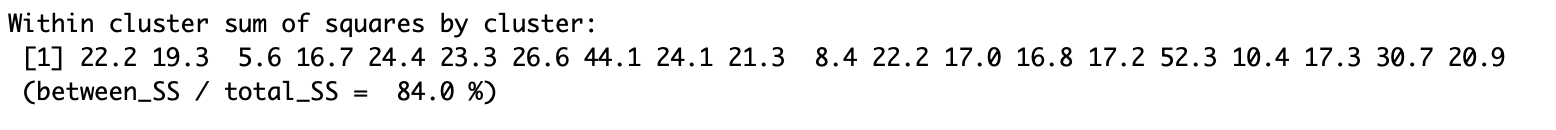
The Euclidian distance that are sensitive to variations within the magnitude or scales from the attributes. In actual applications, due to the variations in selection of the attribute's value, one attribute might overpower another one. Normalization prevents outweighing features having a large number over features with smaller numbers. The aim would be to equalize the dimensions or magnitude and also the variability of those features

(d) Run k-means clustering using a seed = 42, and choose k = 20. Copy the R code used below. (0.5)

set.seed(42)

km <- kmeans(dungaree.df.norm, 20)

(e) Based on the results, does k=20 clusters seem appropriate? Why or why not? (0.5)



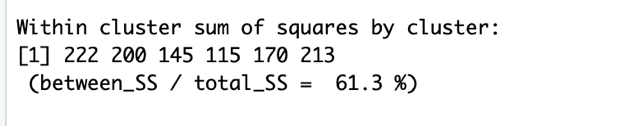
We observe the within cluster sum of squares by cluster is 84.0%. Our objective is to maximize between\_ss/total\_ss.  
However, if you choose the number of clusters to be the same as the number of observations - then total\_ss is exactly equal to between\_ss and the desired ratio will be 1 (or 100%). So, in order to get high percentage, we can just increase the number of clusters - but then we miss out the point of clustering. A way to deal with it is to use the elbow method to choose a reasonable number of clusters.

So, K=20 does not seem appropriate.

(f) In the next run, specify a maximum of six clusters, and run the k-means clustering algorithm

again. Copy the R code used below. (0.25)

km2 <- kmeans(dungaree.df.norm, 6)



(g) Plot profile plot of centroids for the six clusters generated in (f). Copy the code used and the result

below. (0.5)

plot(c(0), xaxt = 'n', ylab = "", type = "l", ylim = c(min(km2$centers), max(km2$centers)), xlim = c(0, 8))

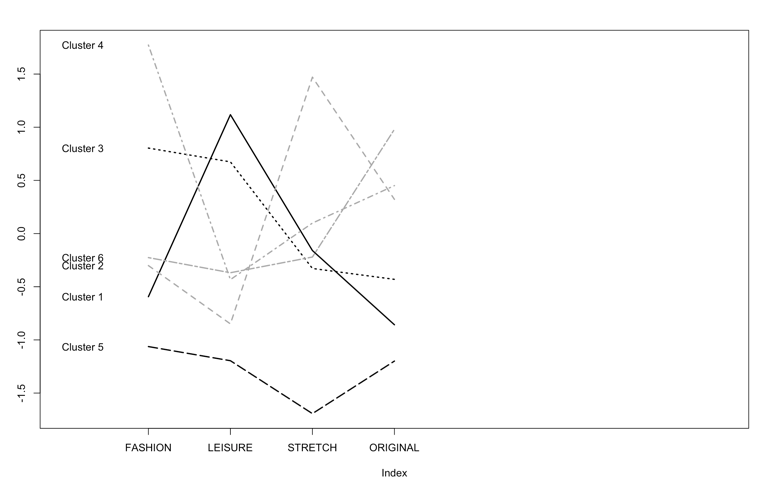
axis(1, at = c(1:4), labels = names(dungaree.df))

for (i in c(1:6))

lines(km2$centers[i,], lty = i, lwd = 2, col = ifelse(i %in% c(1, 3, 5),

"black", "dark grey"))

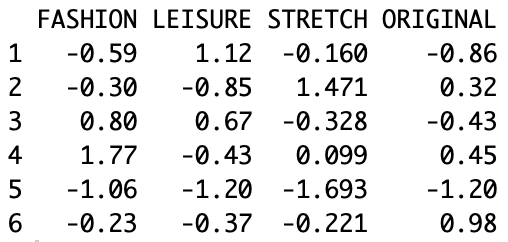
text(x = 0.2, y = km2$centers[, 1], labels = paste("Cluster", c(1:6)))



(h) Using the profile plot of centroids, interpret the characteristics of each cluster as it relates to types

of jeans sold at stores. Describe these clusters, and their similarities and differences in words. (0.5)

km2$centers



Based on the profile plot of centroids clusters 1-3 and 2-6 are very similar.

2.

(a) Import the data to R, set row names to the “Symbol” column, and remove all the columns that you

are not going to use for clustering. Copy the R code used below. (0.5)

Pharmaceuticals.df <- read.csv("Pharmaceuticals.csv")

row.names(Pharmaceuticals.df) <- Pharmaceuticals.df[,1]

Pharmaceuticals.df <- Pharmaceuticals.df[,c(3:11)]

(b) Normalize the data. Copy the R code used below. (0.5)

Pharmaceuticals.df.norm <- sapply(Pharmaceuticals.df, scale)

row.names(Pharmaceuticals.df.norm) <- row.names(Pharmaceuticals.df)

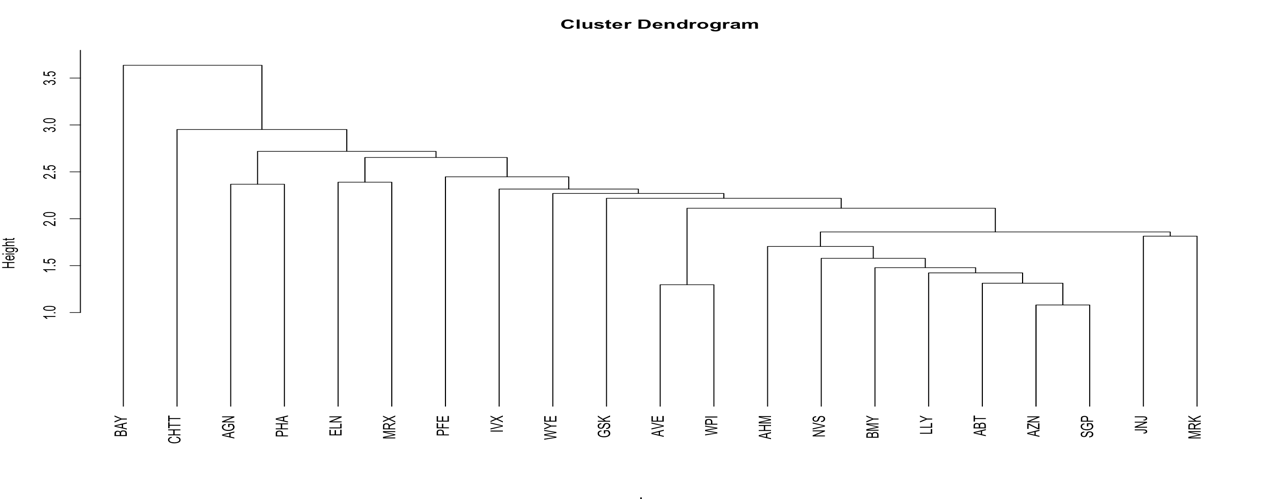
(c) Based on single linkage, run hierarchical clustering to generate Dendrogram. Copy the code used

and the result below. (0.75)

d.norm <- dist(Pharmaceuticals.df.norm, method = "euclidean")

hc1.norm <- hclust(d.norm, method = "single")

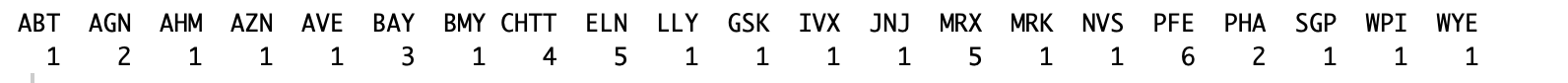
plot(hc1.norm, hang = -1)



(d) If we are interested in 6 clusters based on Dendrogram in (c), what are the members of each

cluster? Copy the code used and the result below. (0.5)

members <- cutree(hc1.norm, k = 6)

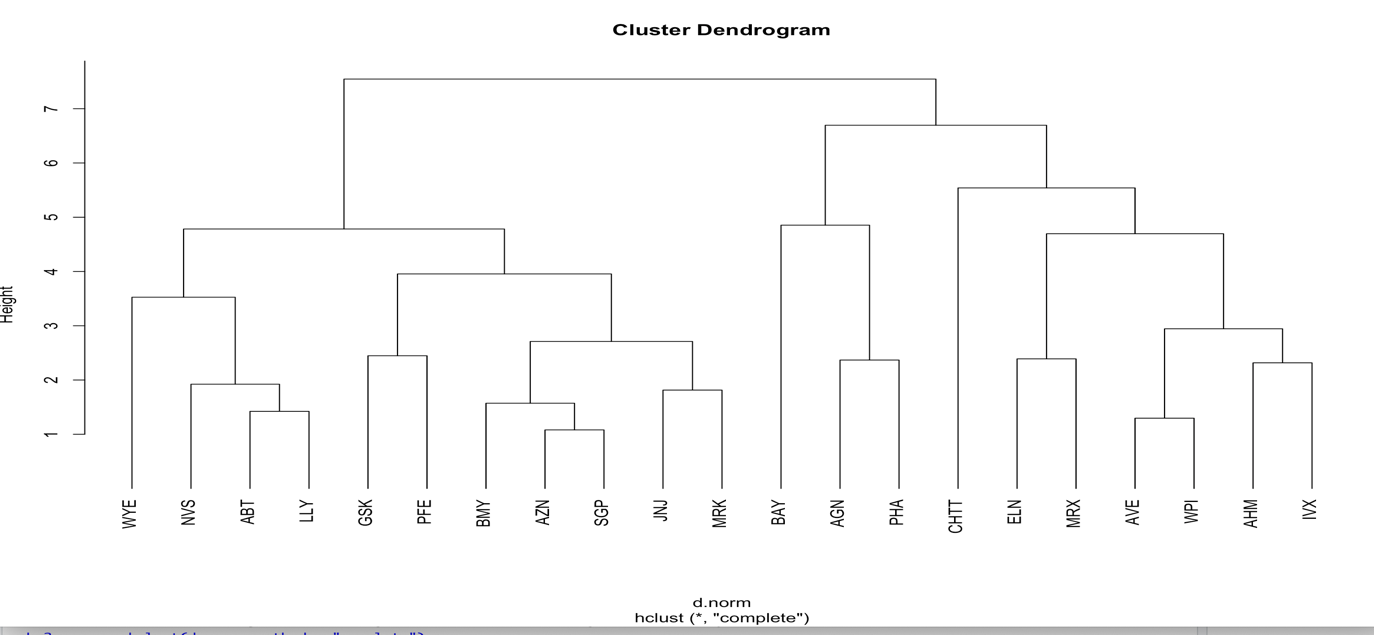


(e) Based on complete linkage, run hierarchical clustering to generate Dendrogram. Copy the code

used and the result below. (0.75)

hc2.norm <- hclust(d.norm, method = "complete")

plot(hc2.norm, hang = -1)



(f) If we are interested in 6 clusters based on Dendrogram in (e), what are the members of each

cluster? Copy the code used and the result below. (0.5)

memberc <- cutree(hc2.norm, k = 6)



(g) Do (d) and (f) lead to the same six clusters? Explain why. (0.5)

No, the distance the computation in single and complete linkage are different. Hence, they will not lead to the same six clusters.