k-Means Clustering: Dungaree Data

1. Introduction

The DUNGAREE data set shows the number of pairs of four different types of dungarees sold at stores over a specific period. Each row represents an individual store. There are six columns in the data set. One column is the store identification number, and the remaining columns contain the number of pairs of each type of jeans sold.

Name	Model Role	Data Type	Description
STOREID	Ident	Numeric	Identification number of the store
FASHION	Input	Numeric	Number of pairs of fashion jeans sold at the store
LEISURE	Input	Numeric	Number of pairs of leisure jeans sold at the store
STRETCH	Input	Numeric	Number of pairs of stretch jeans sold at the store
ORIGINAL	Input	Numeric	Number of pairs of original jeans sold at the store
SALESTOT	Ignore	Numeric	Total number of pairs of jeans sold (the sum of FASHION, LEISURE, STRETCH, and ORIGINAL)

Data File:

dungaree.csv

2. Data Analysis

a. Check for Unusual Values –

```
> # Dungaree Code
> # read csv file
> dungree <- read.csv("E:/GitHub Projects/K-Means Clustering/dungaree.csv")
> # normalize data
> dungree.norm<- sapply(dungree[,2:6],scale)
> colnames(dungree.norm) <- c('z_fashion','z_leisure','z_stretch','z_original','z_salestot')
> df <- cbind(dungree,dungree.norm)
> head(df)
  STOREID FASHION LEISURE STRETCH ORIGINAL SALESTOT z_fashion z_leisure z_stretch z_original z_salestot
1       1     182    1528     496     2203     4409    2.7513   -1.1071    0.2455     1.2126     0.2919
2       2     129    2247     296     1890     4562    1.1263    0.9423   -0.6993     0.1393     0.7097
3       3     107    1652     267     2342     4368    0.4518   -0.7537   -0.8363     1.6893     0.1799
4       4     117    1744     419     2119     4399    0.7584   -0.4914   -0.1183     0.9246     0.2646
5       5     110    1736     755     1781     4382    0.5438   -0.5142    1.4689    -0.2345     0.2182
6       6      79    1637     613     2138     4467   -0.4066   -0.7964    0.7981     0.9897     0.4503
>
```

Unusual data vales are those values that are 2 standard deviation from the mean. i. e. the variables that have z-score value more than 2 and less than -2 are unusual data values. I have standardized the data to make all the variables of same scale. Therefore, as per the data set of Dungaree there are values present that have z-score less than -2 and greater than 2, which makes them unusual.

$$\text{z-score} = \frac{\text{data value} - \text{mean}}{\text{standard deviation}}$$

b. Check for Missing Values

```
> #check for missing values
> sum(is.na(df$FASHION))
[1] 0
> sum(is.na(df$STRETCH))
[1] 0
> sum(is.na(df$LEISURE))
[1] 0
> sum(is.na(df$ORIGINAL))
[1] 0
> sum(is.na(df$SALESTOT))
[1] 0
> sum(is.na(df))
[1] 0
```

We need to check for the missing values in the data set and impute the missing values using suitable techniques. As per the above code I checked if there is any **NA** value in each column and summed over NA values. The result is that there is sum is zero i.e is there is no NA value and we don't need to impute missing variables.

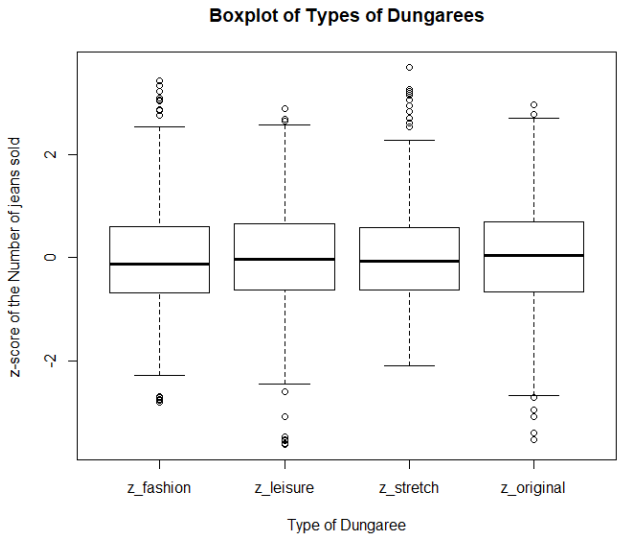
c. Check for Outliers

Outliers plays a vital role in distorting our analysis. So, it is important to remove outliers from the data.

Use of **Boxplot** to remove the outliers:

- Below is the boxplot before removing the outliers :

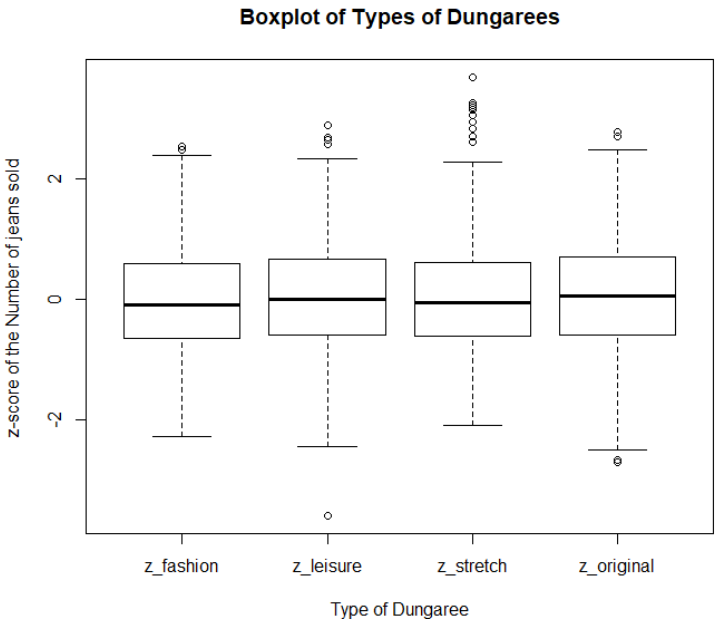
```
> boxplot(df[,7:10], xlab = "Type of Dungaree", ylab = "z-score of the Number of jeans sold ", main = "Boxp
lot of Types of Dungarees")
>
```



- Check for outliers and remove them :

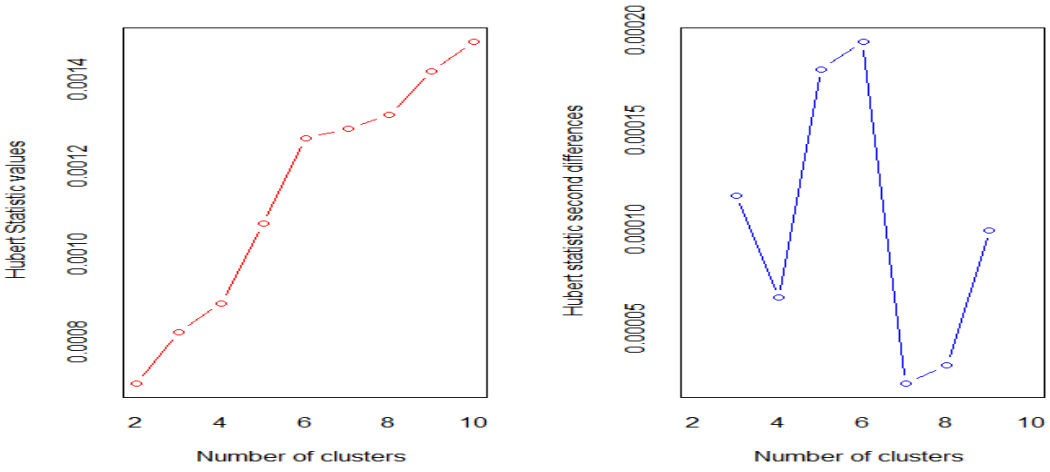
```
> # check for the outliers
> fashion.outlier <- boxplot.stats(df$z_fashion)$out
> leisure.outlier <- boxplot.stats(df$z_leisure)$out
> stretch.outlier <- boxplot.stats(df$z_stretch)$out
> original.outlier <- boxplot.stats(df$z_original)$out
> # create unique vectors of te outlier value
> fashion.outlier.un <- unique(fashion.outlier)
> leisure.outlier.un<- unique(leisure.outlier)
> stretch.outlier.un <- unique(stretch.outlier)
> original.outlier.un <- unique(original.outlier)
> # create function to remove the outliers vairable
> outlier_value <- function(x, factor){
+
+   v <- vector("numeric", length = 0)
+   for (i in 1:length(x)){
+     for (j in 1:length(factor)){
+       if (x[i] == factor[j]){
+         v<- c(v,i)}
+
+     }
+   }
+   return(v)
+ }
> # find the rows containing outlers
> ve1 <- outlier_value(df$z_fashion,fashion.outlier.un)
> ve2 <- outlier_value(df$z_leisure,leisure.outlier.un)
> ve3 <- outlier_value(df$z_stretch,stretch.outlier.un)
> ve4 <- outlier_value(df$z_original,original.outlier.un)
> # remove the rows with outlier values
> df <- df[-ve1,]
> df <- df[-ve2,]
> df <- df[-ve3,]
> df <- df[-ve4,]
> boxplot(df[,7:10], xlab = "Type of Dungaree", ylab = "z-score of the Number of jeans sold ", main = "Boxp
lot of Types of Dungarees")
>
```

Boxplot after removing the outliers:

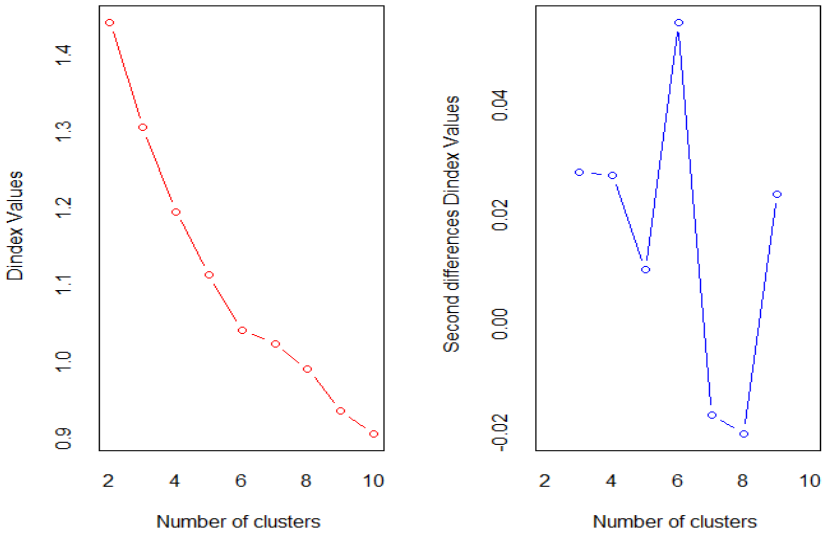


d. Application of NbClust algorithm:

```
> library(NbClust)
> devAskNewPage(ask=TRUE)
> nc <- NbClust(df[,7:10], min.nc=2, max.nc=10, method="kmeans")
Hit <Return> to see next plot:
*** : The Hubert index is a graphical method of determining the number of clusters.
                In the plot of Hubert index, we seek a significant knee that corresponds to a
                significant increase of the value of the measure i.e the significant peak in Hubert
                index second differences plot.

Hit <Return> to see next plot:
```



```
Hit <Return> to see next plot:
*** : The D index is a graphical method of determining the number of clusters.
                In the plot of D index, we seek a significant knee (the significant peak in Dindex
                second differences plot) that corresponds to a significant increase of the value of
                the measure.

*******************************************************************
* Among all indices:
* 5 proposed 2 as the best number of clusters
* 2 proposed 3 as the best number of clusters
* 2 proposed 4 as the best number of clusters
* 5 proposed 5 as the best number of clusters
* 5 proposed 6 as the best number of clusters
* 2 proposed 9 as the best number of clusters
* 2 proposed 10 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  2


*******************************************************************
```

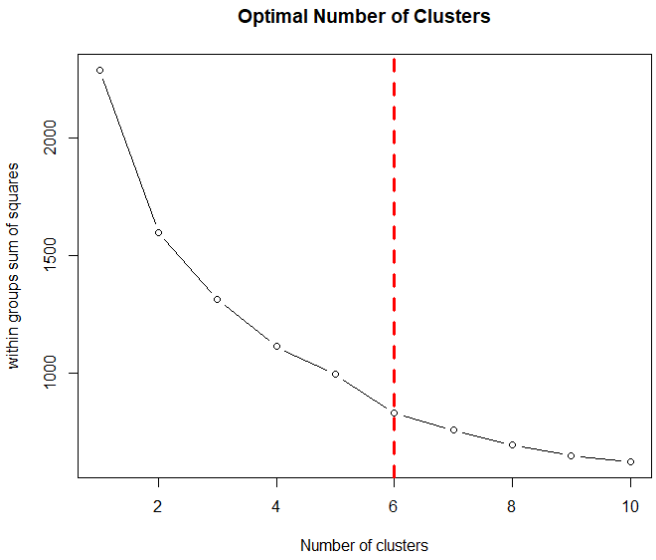


Result of Hubert Index & Dindex Values – as per Hubert Statistic and Dindex Value the significant knee is at cluster 6, as there is a significant increase in the value. I.e.at cluster 6 there is a significant peak in the Hubert Second difference plot & Second difference Dindex Values.

Therefore, optimum number of clusters is 6 that can divide the data into 6 different groups. Each group will have large inter Cluster distance and small intra cluster distance.

e. Application of k-Means algorithm

```
> #function to calculate the withing sum of sqaures for a range of number of clusters
> wssplot <- function(data, nc=10, seed=1234) {
+   wss <- (nrow(df)-1)*sum(apply(df[,7:10], 2, var))
+   for (i in 2:10) {
+     set.seed(1234)
+     wss[i] <- sum(kmeans(data, centers=i)$withinss)
+   }
+   plot(1:10, wss, type="b",main = "Optimal Number of Clusters" , xlab="Number of clusters", ylab="within
 groups sum of squares")
+ }
> wssplot(df[,7:10])
> abline(v=6, col="red", lty=2, lwd=3)
>
```



K-means algorithm divides the data into different groups. I have used Elbow method that says that as we increase the number of clusters the within sum of squares distance of clusters decreases and the elbow is the point of optimum number of clusters where after increasing the clusters the variance explained by the clusters decreases and doesn't add much information on increasing clusters. Hence 6 is the optimum number of clusters.

f. Run k-means for 6 clusters:

```
> fit.km <- kmeans(df[,7:10], 6, nstart=25)
> fit.km
K-means clustering with 6 clusters of sizes 98, 100, 41, 112, 157, 135

Cluster means:
    z_fashion  z_leisure   z_stretch  z_original
1 -0.06061345  1.5076324 -0.43560889 -1.18941809
2 -0.23248203 -1.0045020  1.54721121  0.44102149
3 -0.38045541 -0.7322634 -1.92868943 -0.89961818
4  1.26705385  0.1334049 -0.04329743  0.08122828
5 -0.55387205  0.4613341  0.08080612 -0.20768622
6 -0.02486021 -0.5490026 -0.23285208  1.17608624
```

Interpretation of Clusters:

The variable which has higher mean for the particular cluster will be more dominating in cluster. Therefore, below are the observations :

1. In cluster 1 Leisure jeans were sold the most and Original jeans were sold the least.
2. In cluster 2 Stretch jeans were sold the most and Leisure jeans were sold the least.
3. In cluster 3 Fashion jeans were sold the most and Stretch jeans were sold the least.
4. In cluster 4 Fashion jeans were sold the most and Stretch jeans were sold the least.
5. In cluster 5 Leisure jeans were sold the most and Fashion jeans were sold the least.
6. In cluster 6 Original jeans were sold the most and Leisure jeans were sold the least.

Cluster Description:
Cluster 1 - High Leisure Low Original
Cluster 2 - High Stretch Low Leisure
Cluster 3 - High Fashion Low Stretch
Cluster 4 - High Fashion Low Stretch
Cluster 5 - High Leisure Low Fashion
Cluster 6 - High Original Low Leisure

AS per above analysis and cluster description, we can infer that cluster 3 and Cluster 4 are similar, and hence we can merge these two clusters. Resulting in a net of 5 clusters as optimum number of Clusters.

So we have 5 clusters as :

Cluster 1 - High Leisure Low Original
Cluster 2 - High Stretch Low Leisure
Cluster 3 - High Fashion Low Stretch
Cluster 4 - High Leisure Low Fashion
Cluster 5 - High Original Low Leisure

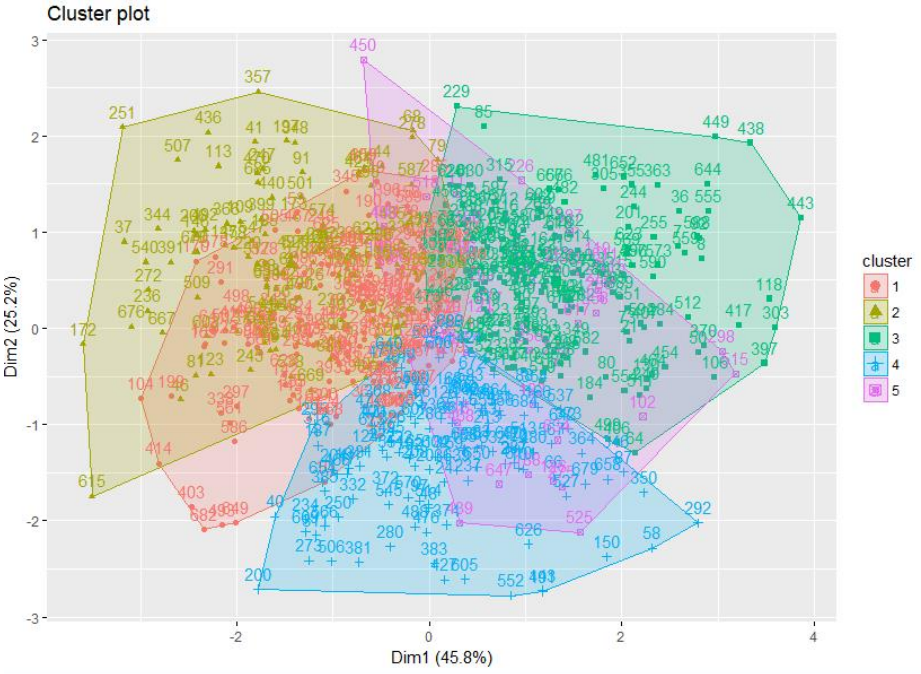
```
> table(fit.km$cluster)

  1   2   3   4   5
171 114 187 128  43
>
```

g. Plot k-means cluster:

Fviz_cluster function of **factoextra** package allows us to plot the k-means clusters. If the data is of higher dimension then the data is converted to 2 dimensions using Principal component Analysis.

Below is a 2-dimensional representation of the dungaree data set clustered in 6 groups

```
library(factoextra)
#with(df[,7:10], pairs(df[,7:10], col=c(1:3)[fit.km$cluster]))
fviz_cluster(fit.km, df[,7:10])
```



3. **Insights**

- There is only 1 cluster (**cluster 3**) of stores having **Fashion** as high sales of jeans.
- Cluster 3 had **highest** number of Stores and Cluster 5 has the **least** number of stores.
- **Leisure Jeans** are **second** to **Fashion Jeans** in sales as Cluster 1 is second to cluster 3 in total number of stores in a cluster.
- **Original Jeans** have the **least** sales across the stores.