The University of Texas at Dallas

Naveen Jindal School of Management

Dungaree Data Analysis

The DUNGAREE data set shows the number of pairs of four different types of dungarees sold at stores over a specific time 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.

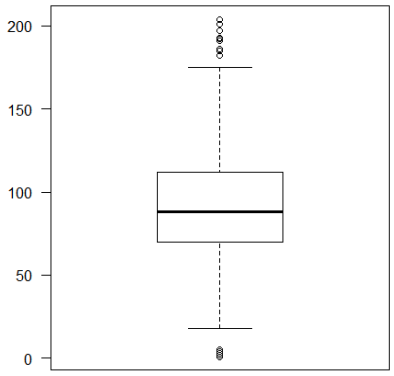
Name	Model Role	Data Type	Description
STOREID	Ident	Numeric	Identification number of the store
FASHION	Input	Numeric	Number of pairs of fashion jeans sold at the store
LEISURE	Input	Numeric	Number of pairs of leisure jeans sold at the store
STRETCH	Input	Numeric	Number of pairs of stretch jeans sold 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)

1.> a.) No unusual data values found in the data sheet.

b.) There are no missing values.

c.) Checked the data for outliers. There are outliers present for all the four types of jeans sold. Used boxplot to find the outliers. Stored the outliers for all the variables in different tables. Joined all the outlier tables using rbind to form a single outlier data frame. Subtracted the outlier table from the existing data.

```
> b<- boxplot(dungaree_data$FASHION)
> b$out
 [1] 182   1   2   1 197   4   5 185 191   5 201   2   1   3 204 186
[17] 193 192
> FASHION_OUT<-dungaree_data[dungaree_data$FASHION %in% b$out,]
> FASHION_OUT
```



```
> #Collect outliers for each variable in a single dataframe
> Outliers<-rbind(ORIGINAL_OUT,STRETCH_OUT,LEISURE_OUT,FASHION_OUT)
>
> #Substract Outliers from the dataframe dungaree_data
> dungaree_data<-dungaree_data[ !(dungaree_data$STOREID %in% Outliers$STOREID), ]
> dim(dungaree_data)
[1] 653   6
```

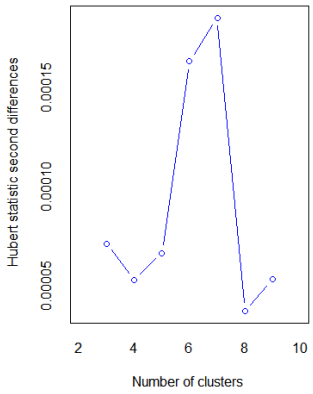
After removing the outliers gave the row names as store ids and removed the store ID column and the last column i.e sales total. Store ID is the identifier column and therefore it is insignificant for clustering. Sales total is a derived column from the existing columns and should therefore be removed. After this I scaled the remaining 4 columns using z score to bring all of them to a common scale.

2.> Run NbClust to check the best number of clusters based on the indexes. The best number of clusters comes to 2. However, based on the Hubert's statistic second differences plot the knee appears at 7.

```
Among all indices:
6 proposed 2 as the best number of clusters
2 proposed 3 as the best number of clusters
4 proposed 4 as the best number of clusters
4 proposed 6 as the best number of clusters
3 proposed 7 as the best number of clusters
1 proposed 8 as the best number of clusters
3 proposed 10 as the best number of clusters

                ***** Conclusion *****

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



Now, running k-means for k=10 and seed=10. Check the number of observations in each cluster and their within cluster sum of squared distances from centroids and their centroid values.

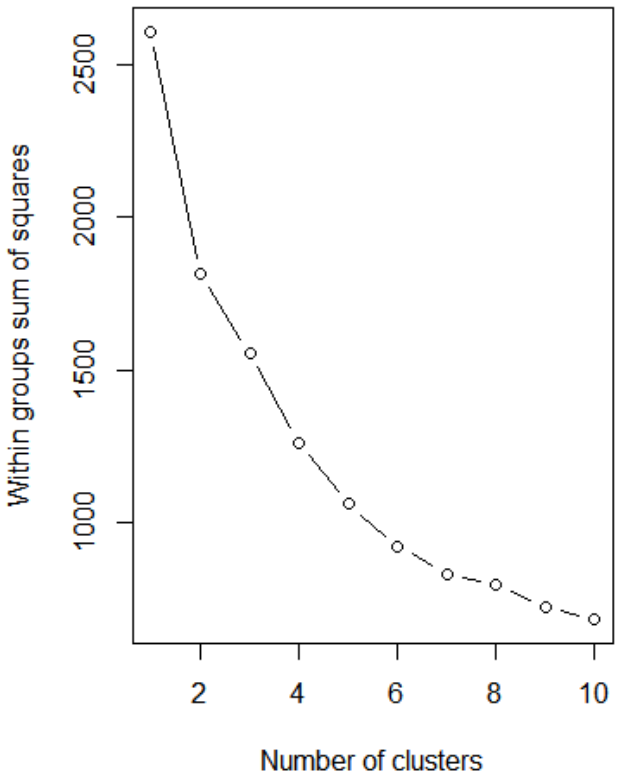
```
> fit.km <- kmeans(dungaree_data.norm, 10, nstart=10)
> fit.km$size
 [1] 74 67 91 59 34 84 63 82 44 55
> fit.km$withinss
 [1] 84.65851 77.69952 66.33314 67.35446 44.65680 57.52986 68.30819 58.34788
 [9] 75.48999 59.90789
> fit.km$centers
       FASHION     LEISURE    STRETCH   ORIGINAL
1  -0.33822847 -1.29154492  1.4623550  0.7422540
2  -0.42132553  0.05196443  1.1425978 -0.6243294
3  -0.42551360  0.83673440 -0.1987417 -0.5443499
4   0.02042267 -1.08424596 -0.2453337  1.6757907
```

3.> Here, k=10 is not that appropriate. If we look at the centers of the clusters using fit.km$centers. Clusters 3, 6 and 8 are difficult to characterize. The reason being the values of the four variables

in these cluster centers is neither very high nor very low. This is due to the high number of clusters which lead to some clusters which do not have a proper meaning.

```
> fit.km$centers
       FASHION     LEISURE    STRETCH   ORIGINAL
1  -0.33822847 -1.29154492  1.4623550  0.7422540
2  -0.42132553  0.05196443  1.1425978 -0.6243294
3  -0.42551360  0.83673440 -0.1987417 -0.5443499
4   0.02042267 -1.08424596 -0.2453337  1.6757907
5   1.88180797 -0.79272748  0.8413207  0.3396998
6  -0.79076026 -0.09104249 -0.1153740  0.5423324
7   1.42780073  0.86220359 -0.3309741 -0.6469864
8   0.64226508 -0.07004172 -0.3643991  0.5537892
9  -0.41504523 -0.82348607 -2.0832401 -0.9459665
10 -0.56615061  1.85779406 -0.5223087 -1.5012053
```

4.> Run the wssplot to check if k=10 clusters is appropriate. Wssplot plot gives the plot for within group sum of squares vs the number of clusters selected. Within group sum of squares is inversely proportional to similarity within the clusters. So, the lesser the within group sum of squares the more is the similarity among the observations in a cluster. Here, k=10 is not that appropriate. The decrease in the within group sum of squares in wssplot is negligible after number of clusters=7. Even though the within group sum of squares is low for k=10 but having 10 clusters for this dataset is not meaningful.

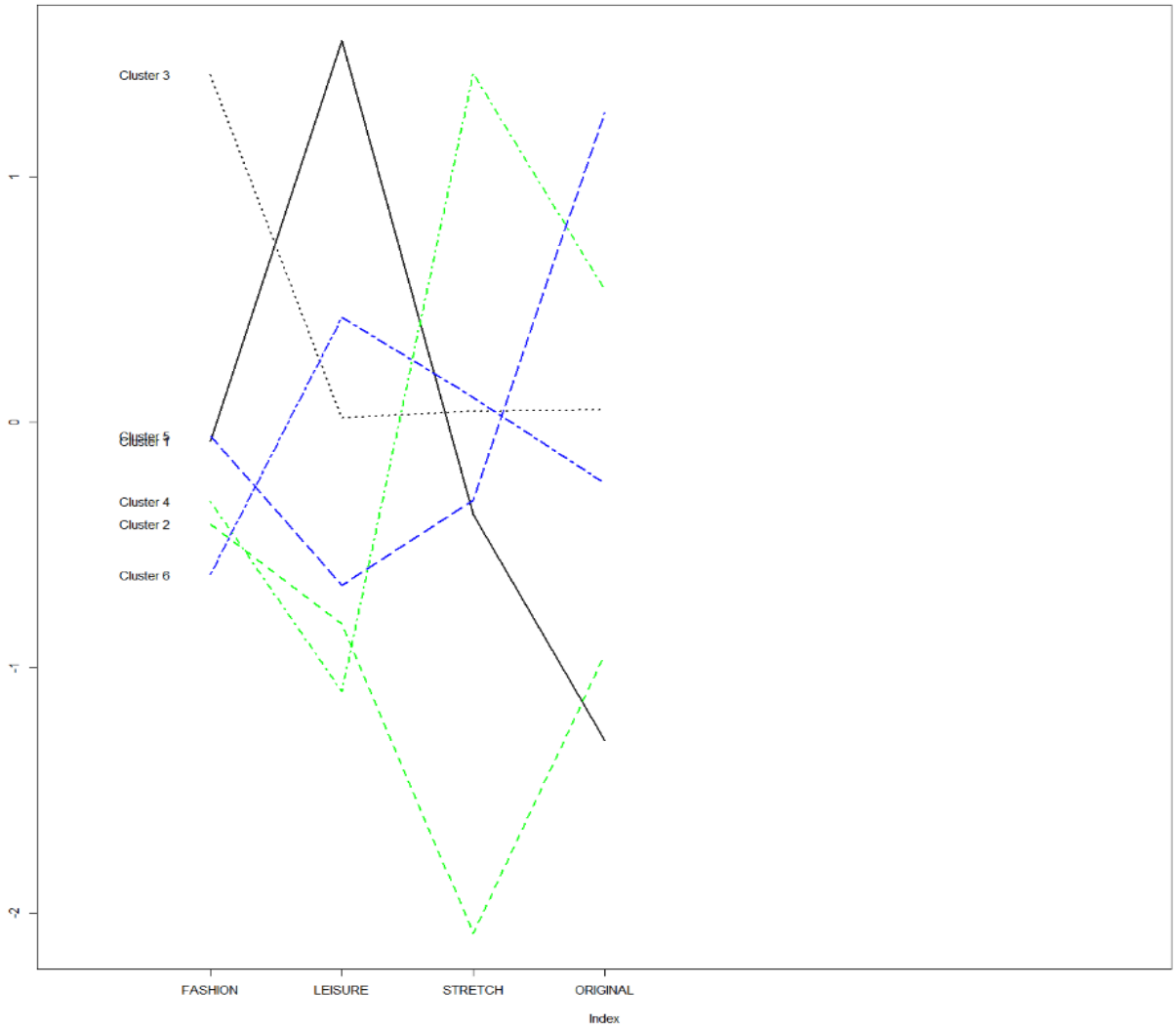


5.> After this, I run k-means for k=6 i.e. for number of clusters=6 and check the cluster sizes, the within cluster sum of squares and their centers.

```
> fit.km <- kmeans(dungaree_data.norm, 6, nstart=10)
> fit.km$size
[1] 103  44 118 104 122 162
> fit.km$withinss
[1] 168.55983  75.48999 193.12181 156.87747 150.90399 177.83319
> fit.km$centers
      FASHION     LEISURE     STRETCH    ORIGINAL
1 -0.07706356  1.55476195 -0.37855021 -1.29858623
2 -0.41504523 -0.82348607 -2.08324012 -0.94596651
3  1.41736658  0.01743575  0.04349308  0.04976645
4 -0.32326659 -1.09927414  1.42303164  0.54021091
```

6.> To interpret the above out I created a Profile plot visualization of clusters centroids.

```
> plot(c(0), xaxt = 'n', ylab = "", type = "l", ylim = c(min(fit.km$centers), max(
fit.km$centers)), xlim = c(0, 8))
Hit <Return> to see next plot: # label x-axes
> axis(1, at = c(1:4), labels = names(dungaree_data))
> # plot centroids
>
> for (i in c(1:6))
+   lines(fit.km$centers[i,], lty = i, lwd = 2,
+         col = ifelse(i %in% c(1, 3),"black",
+                      (ifelse(i %in% c(5, 6),"blue",
+                              "green"))))
>
> # name clusters
> text(x = 0.5, y = fit.km$centers[, 1], labels = paste("Cluster", c(1:6)))
```



Here, based on the characteristics or the profiles of each cluster centroids we can name the clusters, find the similarities between the observations in a cluster and the differences between any two clusters.

From the above visualization:

Cluster 3 can be called as the **'young generation'** based on the observation that this is the cluster consisting of stores from which maximum fashion clothes are bought.

Cluster 5 can be called as '**Previous generation'** based on the observation that they are the ones selling the maximum original clothes (considering original clothing as Vintage clothing based on google)

Cluster 1 can be called as the **'Beachsiders'** based on the observation that these stores have maximum sales of leisure clothes which helps the buyer relax. Also, the least sales here is of original clothes (vintage style clothing).

Cluster 6 can be called as the **'old generation'** based on the observation that these stores have the least fashion sales and an above average sales of leisure clothes. Considering older people prefer a bit more of leisure clothes over original and stretch and do not prefer fashion clothes.
Cluster 4 can be called as '**Too many women'** based on the observation that it has the minimum sales of leisure and maximum sales of stretch clothes. Stretch clothes generally come for woman. Very few stretch clothes are for men.
Cluster 2 can be called as **'Too many men'** based on the observation that sales of stretch is the least and the sales of all the other types of jeans is average and not too high (considering men do less of shopping)

7.> Next testing the cluster validity. Evaluate the cluster validity by finding the ratio of the sum of squared distances within clusters when k=6 to the sum of squared distances when the entire dataset is considered as a single cluster i.e. k=1. The value comes to 0.3538. The possible values for this ratio is between 0 and 1. The more this value is close to zero the better are the inter cluster relations and the more is the differentiation between clusters. The differentiation here will be (1 - 0.3538)= 0.6462.

```
> fit.km1 <- kmeans(dungaree_data.norm, 1, nstart=10)
> fit.km1$withinss
[1] 2608
>
> #Evaluate the cluster validity. Find the ratio of sum of squared
> #distances within clusters where k=6 and sum of squared distances
> #within cluster where k=1. The ratio should be closer to 0 and farther from 1.
>
> a<-sum(fit.km$withinss)/fit.km1$withinss
> a
[1] 0.3538291
```

Now doing the same for k=10 we get 0.2524. The differentiation here will be (1-0.2524)= 0.7476

```
> fit.km <- kmeans(dungaree_data.norm, 10, nstart=10)
> fit.km$withinss
 [1] 67.51319 62.84010 74.03133 64.30762 61.62144 71.06331 65.57574 75.48999
 [9] 53.22364 62.64434
> #Evaluate the cluster validity. Find the ratio of sum of squared
> #distances within clusters where k=6 and sum of squared distances
> #within cluster where k=1. The ratio should be closer to 0 and farther from 1.
>
> a<-sum(fit.km$withinss)/fit.km1$withinss
> a
[1] 0.2524198
```

Here, although the ratio for 10 clusters is more close to 0 rather than the ratio for 6 clusters, still having 6 clusters is better than having 10 clusters as it is difficult to figure out the characteristics of clusters and some clusters have no meaning.

So the clustering that provides greater differentiation between clusters is the one with 10 clusters but the clustering that provides better interpretability is the one with 6 clusters.