**Cluster Analysis**

To check for the unusual data values we have plotted a boxplot for the entire data

> getwd()

[1] "C:/Users/aaroh/Documents/BA with R"

> setwd("C:/Users/aaroh/Documents/BA with R")

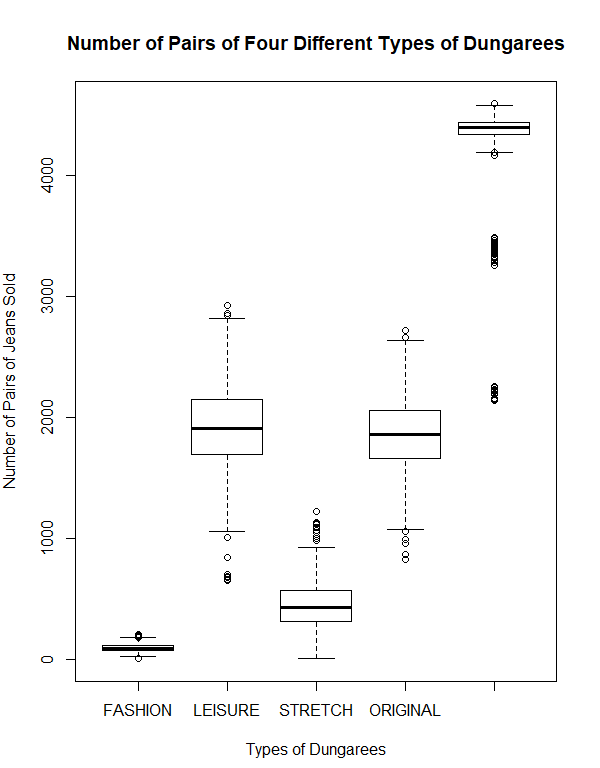
> dungaree.df <- read.csv("dungaree.csv")

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

> dungaree.df <- dungaree.df[,-1]

> boxplot(dungaree.df, main="Number of Pairs of Four Different Types of Dungarees", xlab="Types of Dungarees", ylab="Number of Pairs of Jeans Sold")

Hit <Return> to see next plot:

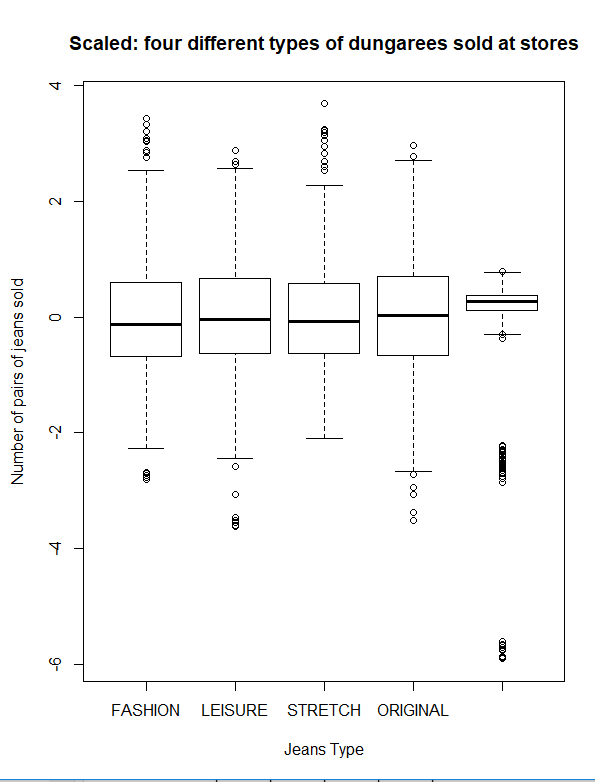


We observe that the variables in the analysis range vary and Leisure has the maximum range. Thus this will show the maximum impact on the results

To avoid this, we can scale the data and again create a boxplot of the scaled values to see the impact.

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

> boxplot(dungaree.df.norm,main="Scaled: four different types of dungarees sold at stores", xlab="Jeans Type", ylab="Number of pairs of jeans sold")



No, there are no missing values. This can be checked in 2 ways.

* Using is.na function on each variable to check for missing values.

> sum(is.na(dungaree.df$FASHION))

[1] 0

> sum(is.na(dungaree.df$LEISURE))

[1] 0

> sum(is.na(dungaree.df$STRETCH))

[1] 0

> sum(is.na(dungaree.df$ORIGINAL))

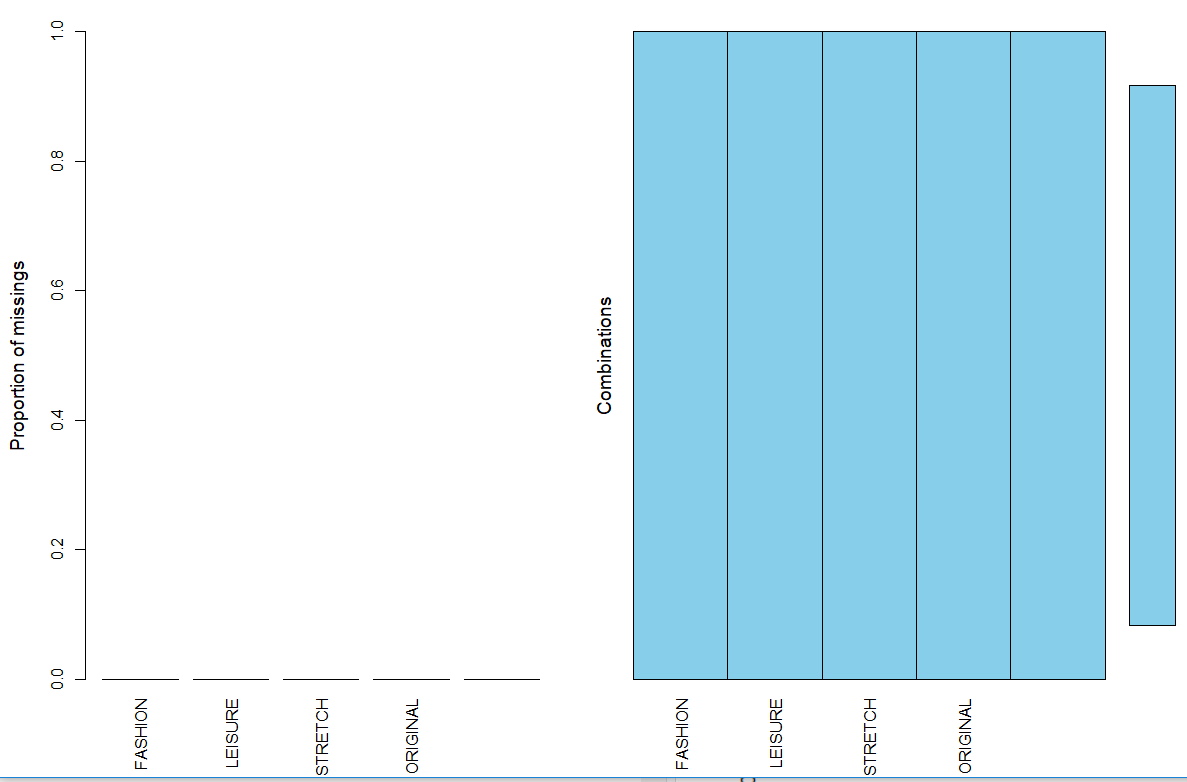
[1] 0

* Using aggr() function from VIM package which plots the number of missing values for each variable alone and each combination of variables.

> library("VIM")

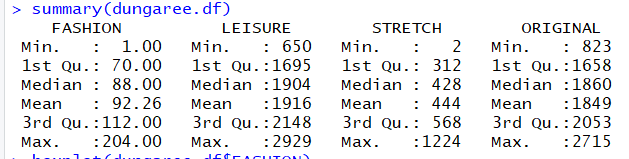
> aggr(dungaree.df)

Hit <Return> to see next plot:



* 1. Check the data for outliers.

The boxplot also displays that we have outliers. The outliers are the values depicted as dots in the boxplot which are outside the interquartile range depicted by the lower and upper hinge for each variable. To check for outliers, we check the summary of the data frame and plot boxplot



> boxplot(dungaree.df$FASHION)

> boxplot(dungaree.df$FASHION)

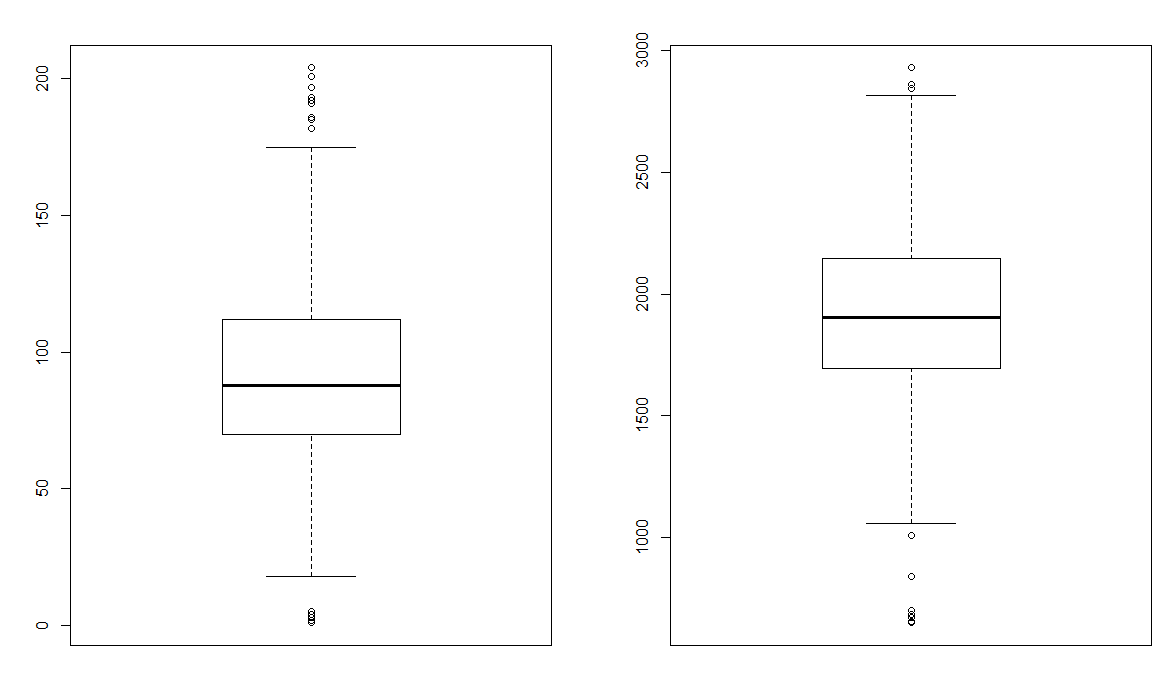
Hit <Return> to see next plot:

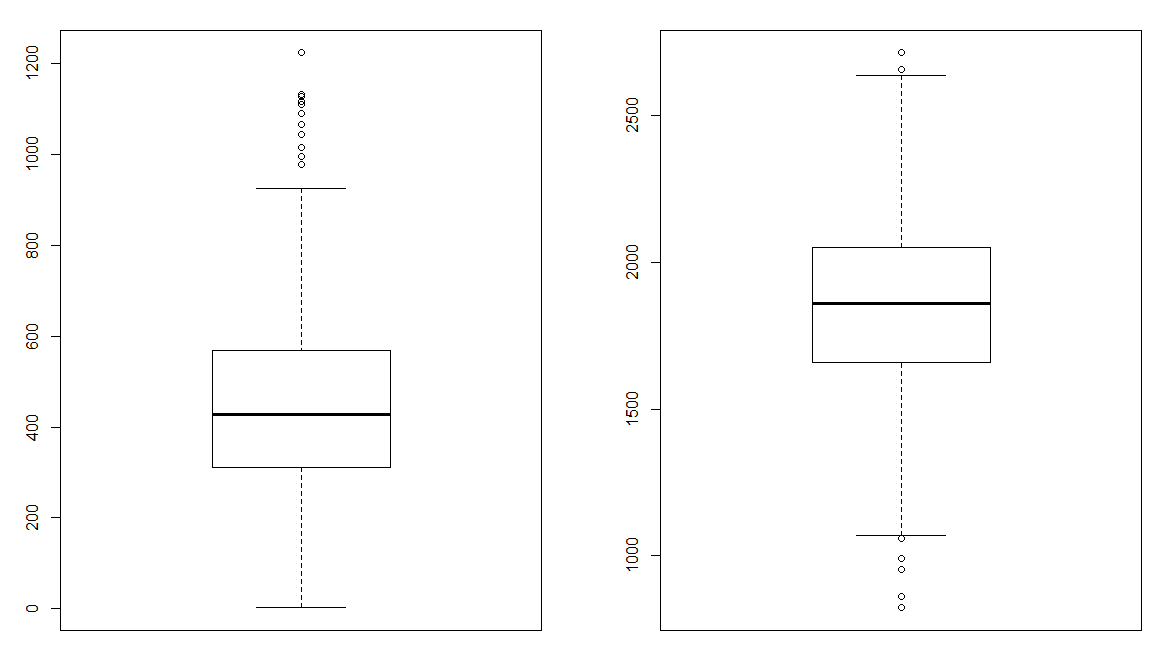
> boxplot(dungaree.df$LEISURE)

> boxplot(dungaree.df$STRETCH)

Hit <Return> to see next plot:

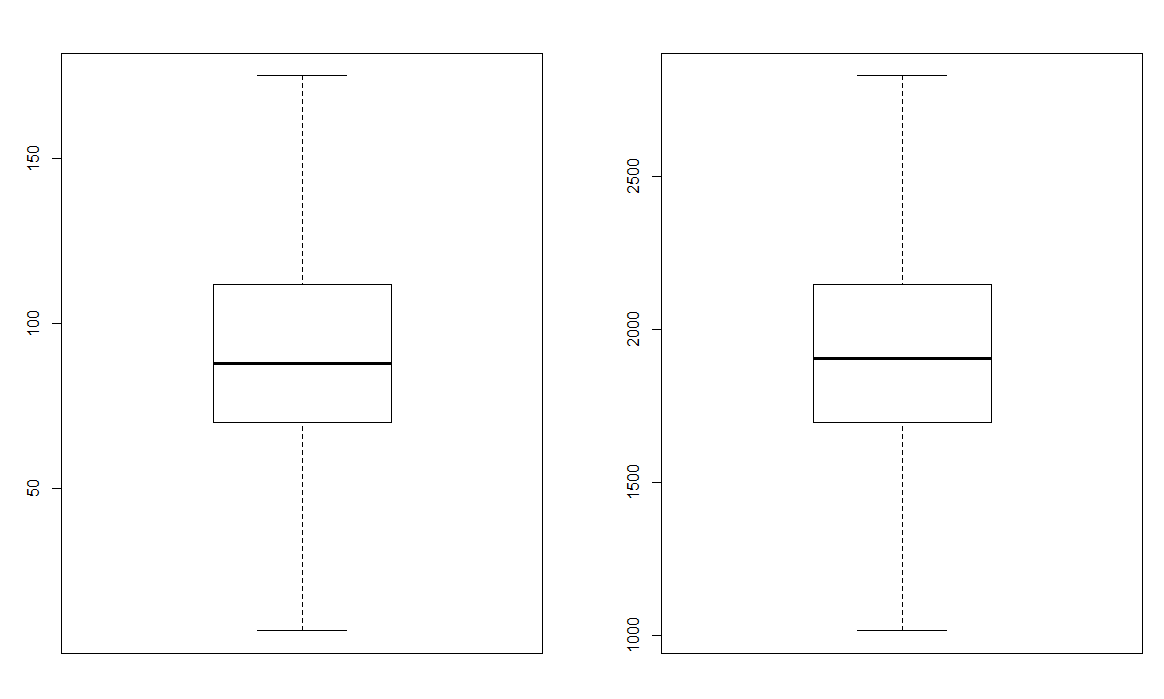
> boxplot(dungaree.df$ORIGINAL)

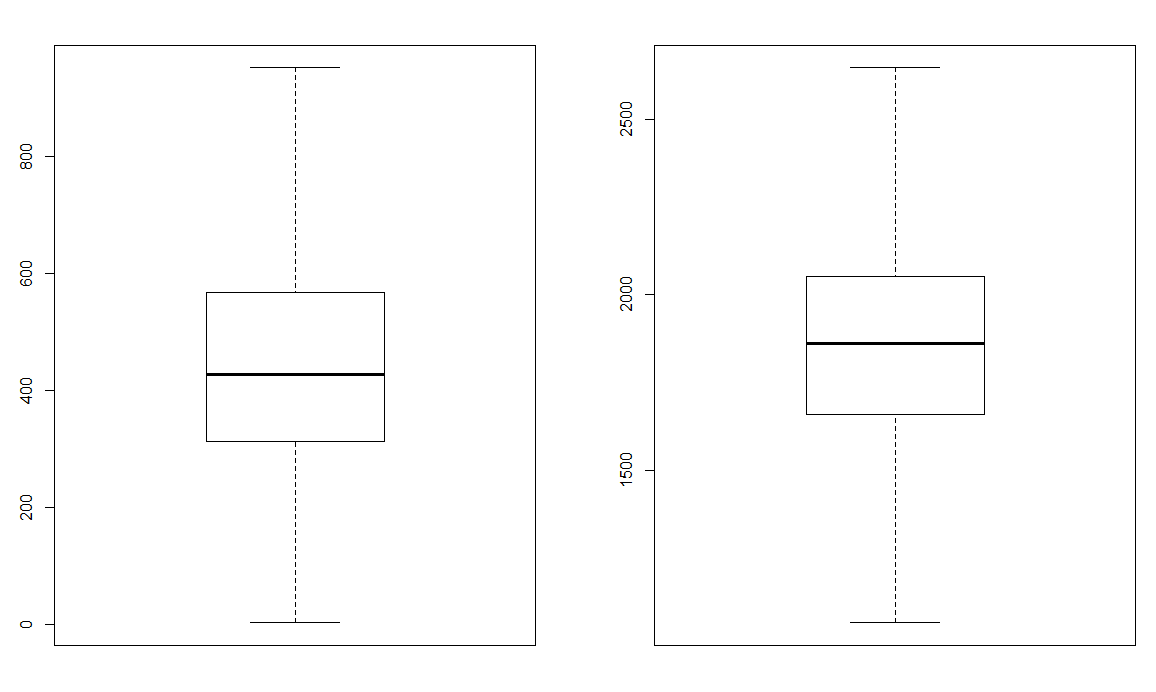




To remove outliers we have used the winsorizing techniques where we eliminate the upper and lower 1% of the data checking if it is greater than third quartile + 1.5\* Interquartile range or first quartile – 1.5\*Interquartile range.

|  |
| --- |
| > dungaree.df$FASHION[dungaree.df$FASHION > 112+1.5\*IQR(dungaree.df$FASHION)] <- 112+1.5\*IQR(dungaree.df$FASHION)  > dungaree.df$LEISURE[dungaree.df$LEISURE > 2148+1.5\*IQR(dungaree.df$LEISURE)] <- 2148+1.5\*IQR(dungaree.df$LEISURE)  > dungaree.df$STRETCH[dungaree.df$STRETCH > 568+1.5\*IQR(dungaree.df$STRETCH)] <- 568+1.5\*IQR(dungaree.df$STRETCH)  > dungaree.df$ORIGINAL[dungaree.df$ORIGINAL > 2053+1.5\*IQR(dungaree.df$ORIGINAL)] <- 2053+1.5\*IQR(dungaree.df$ORIGINAL)  > dungaree.df$FASHION[dungaree.df$FASHION < 70-1.5\*IQR(dungaree.df$FASHION)] <- 70-1.5\*IQR(dungaree.df$FASHION)  > dungaree.df$LEISURE[dungaree.df$LEISURE < 1695-1.5\*IQR(dungaree.df$LEISURE)] <- 1695-1.5\*IQR(dungaree.df$LEISURE)  > dungaree.df$STRETCH[dungaree.df$STRETCH < 312-1.5\*IQR(dungaree.df$STRETCH)] <- 312-1.5\*IQR(dungaree.df$STRETCH)  > dungaree.df$ORIGINAL[dungaree.df$ORIGINAL < 1658-1.5\*IQR(dungaree.df$ORIGINAL)] <- 1658-1.5\*IQR(dungaree.df$ORIGINAL)  > boxplot(dungaree.df$FASHION)  Hit <Return> to see next plot: boxplot(dungaree.df$FASHION)  > boxplot(dungaree.df$LEISURE)  > boxplot(dungaree.df$STRETCH)  Hit <Return> to see next plot:  > boxplot(dungaree.df$ORIGINAL)  > boxplot(dungaree.df$STRETCH)  Hit <Return> to see next plot:  > boxplot(dungaree.df$ORIGINAL) |
|  |
| |  | | --- | |  | |





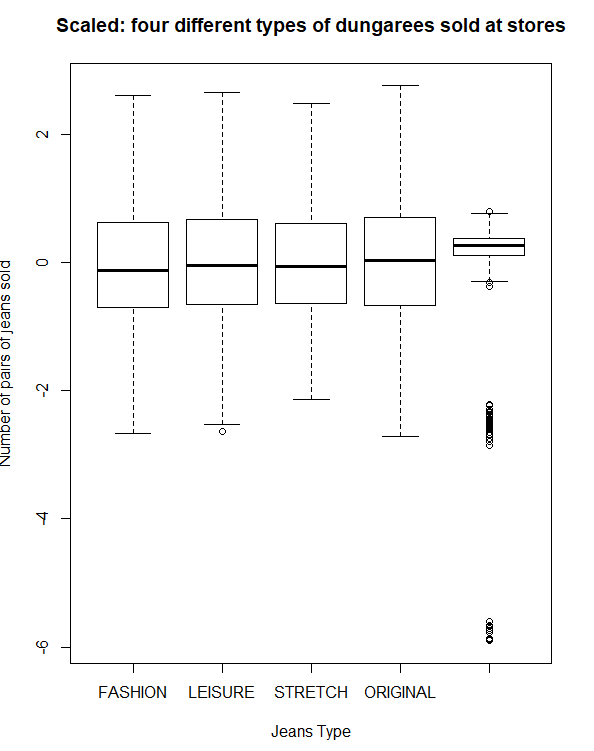
Perform scaling on the data after outliers are removed.

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

Create a boxplot for the scaled data.

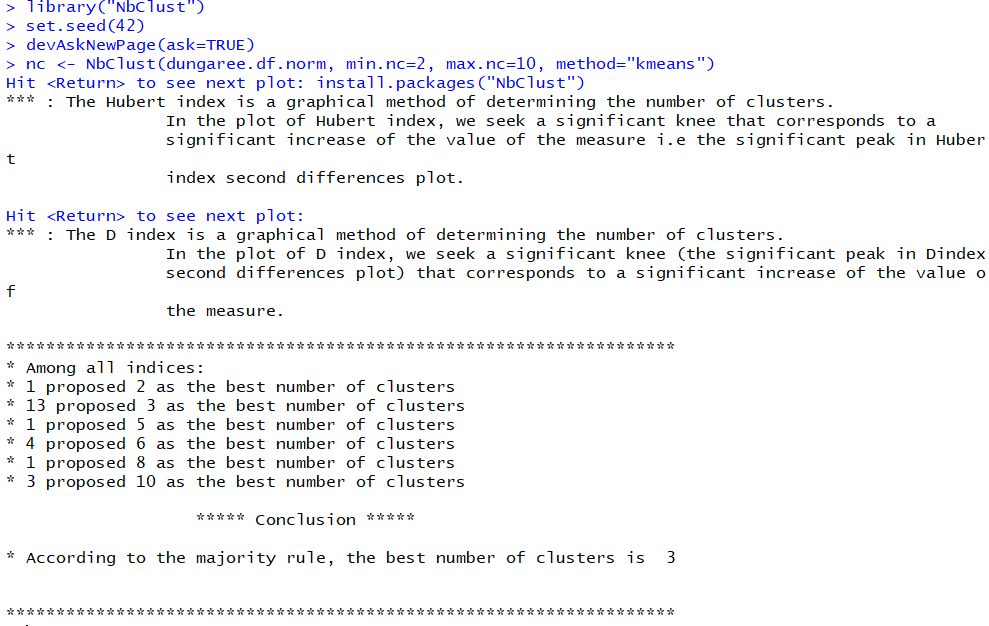
> boxplot(dungaree.df.norm,main="Scaled: four different types of dungarees sold at stores", xlab="Jeans Type", ylab="Number of pairs of jeans sold")

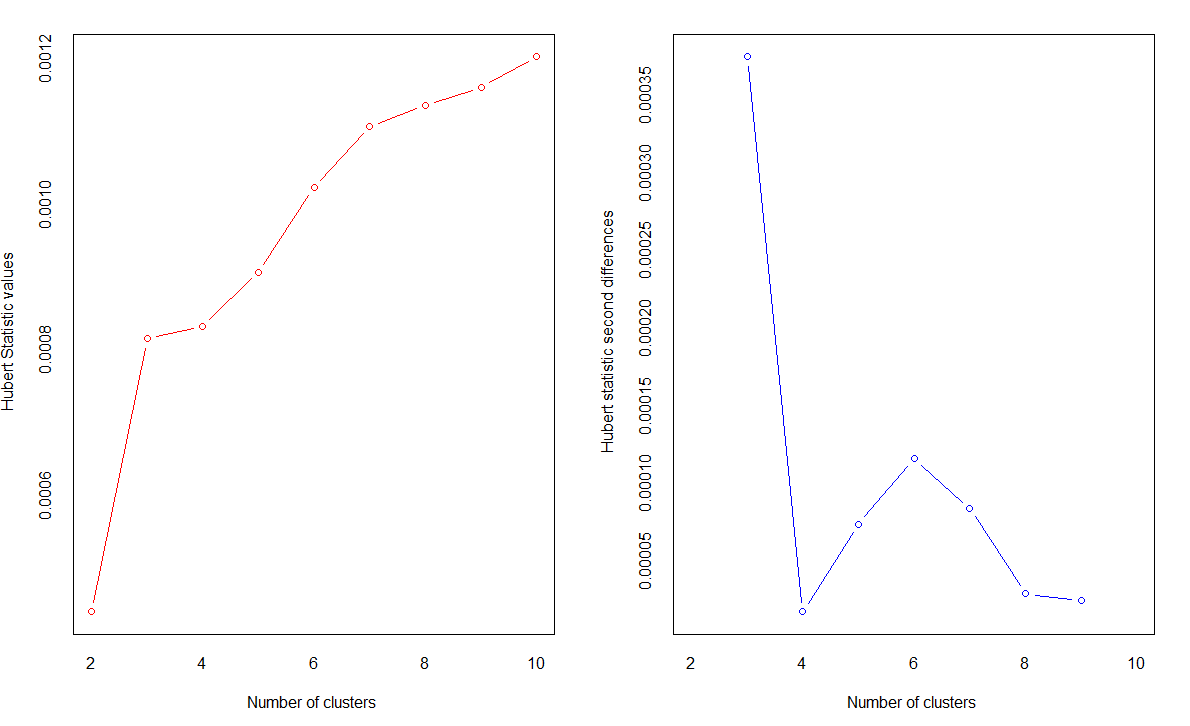
Hit <Return> to see next plot:

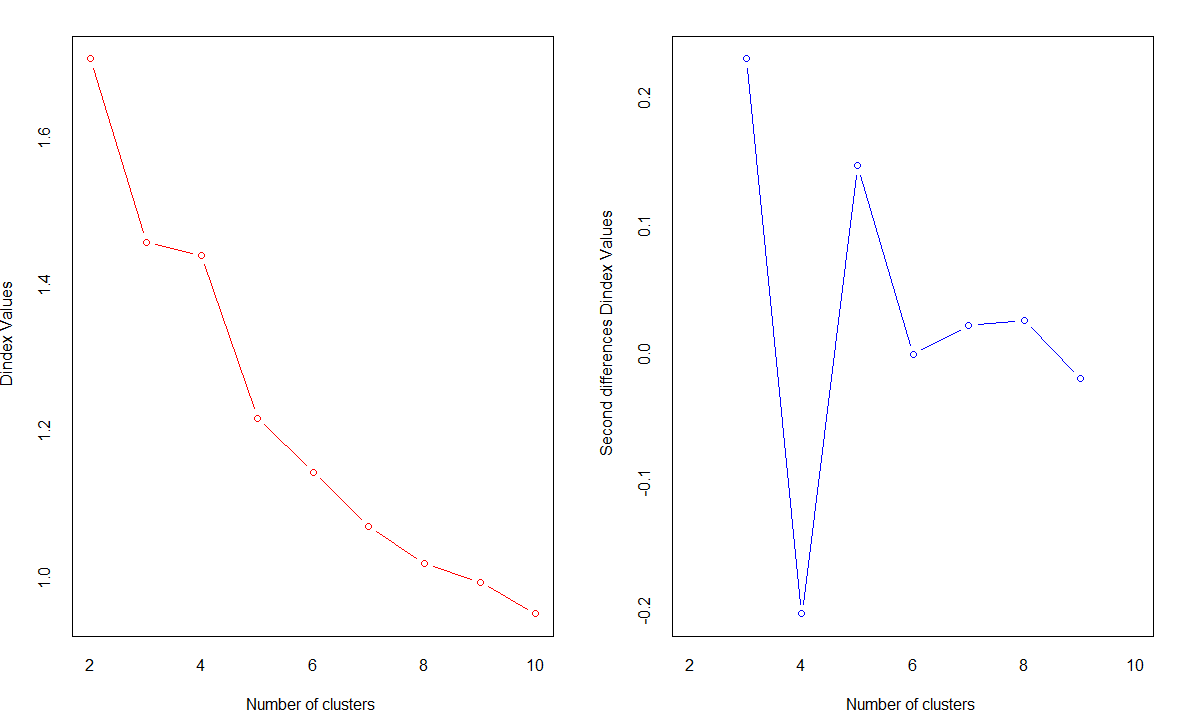


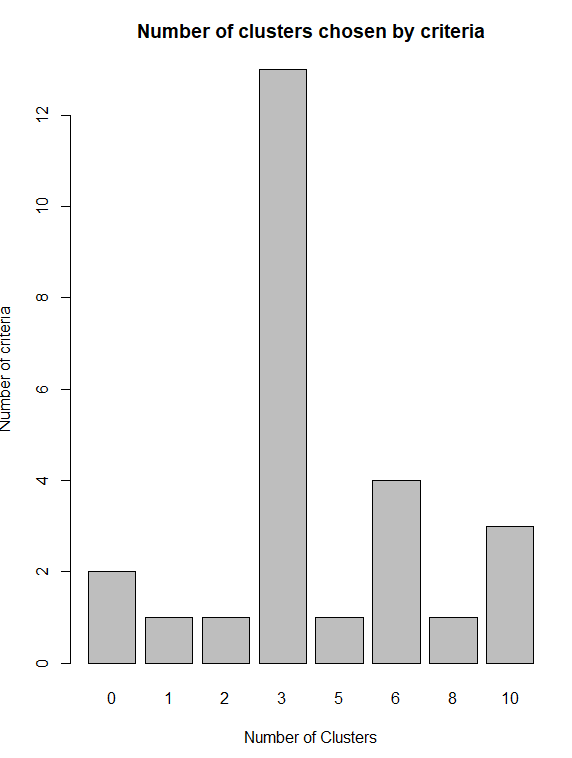
1. Run k-means clustering using a seed = 42, and choose k = 10.

After preforming standardization, K means clustering with seed=42 and k=10 was performed. K means requires us to determine the number of clusters using NbClust() function from NbClust package.

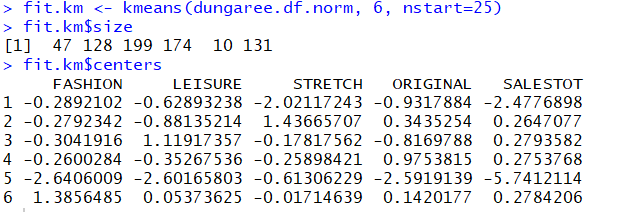








Among the indices, 13 proposed that 3 is the best number of cluster and 4 proposed that 6 is the best number of clusters. We will select 6 as the number of clusters because we have 689 observations and 3 will be too cluttered up not being able to clearly show the properties of a cluster. Also, if we choose 6 as the number of clusters, the model would be more efficient than choosing 3 as the number of clusters. Thus, we choose 6 as the number of clusters.



K=10 clusters doesn’t seem appropriate because according to the above analysis, 3 is the best number of clusters and 6 is the second-best number of clusters. Also, keeping 10 as the clusters is not an efficient strategy because it would be not cost effective since it is the maximum number of clusters.

> wssplot <- function(data, nc=10, seed=42){

+ wss <- (nrow(data)-1)\*sum(apply(data,2,var))

+ for (i in 2:nc){

+ set.seed(42)

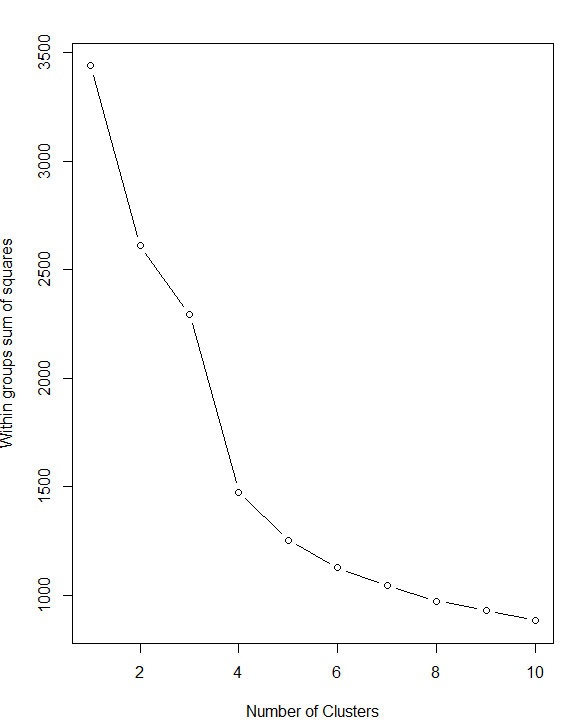
+ wss[i] <- sum(kmeans(data, centers=i)$withinss)}

+ plot(1:nc, wss, type="b", xlab="Number of Clusters",

+ ylab="Within groups sum of squares")}

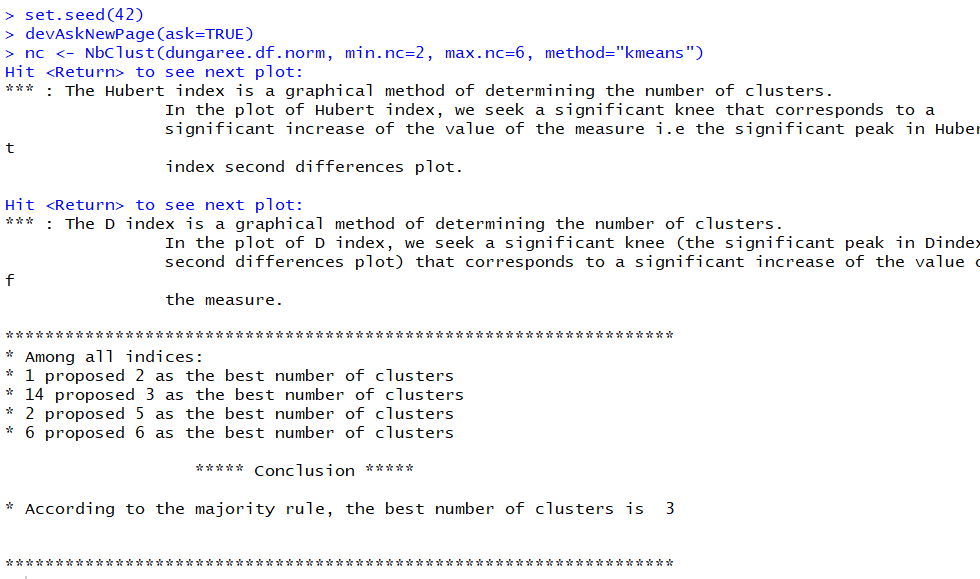
> wssplot(dungaree.df.norm,nc=10,seed=42)

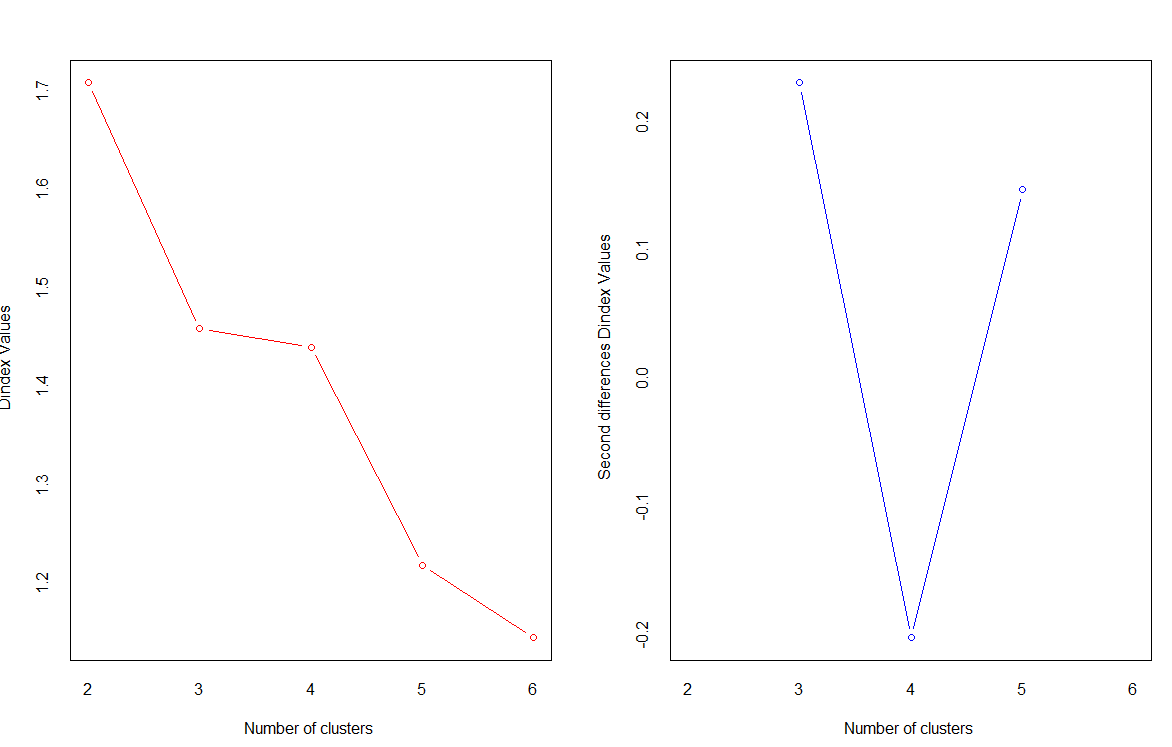
Hit <Return> to see next plot:

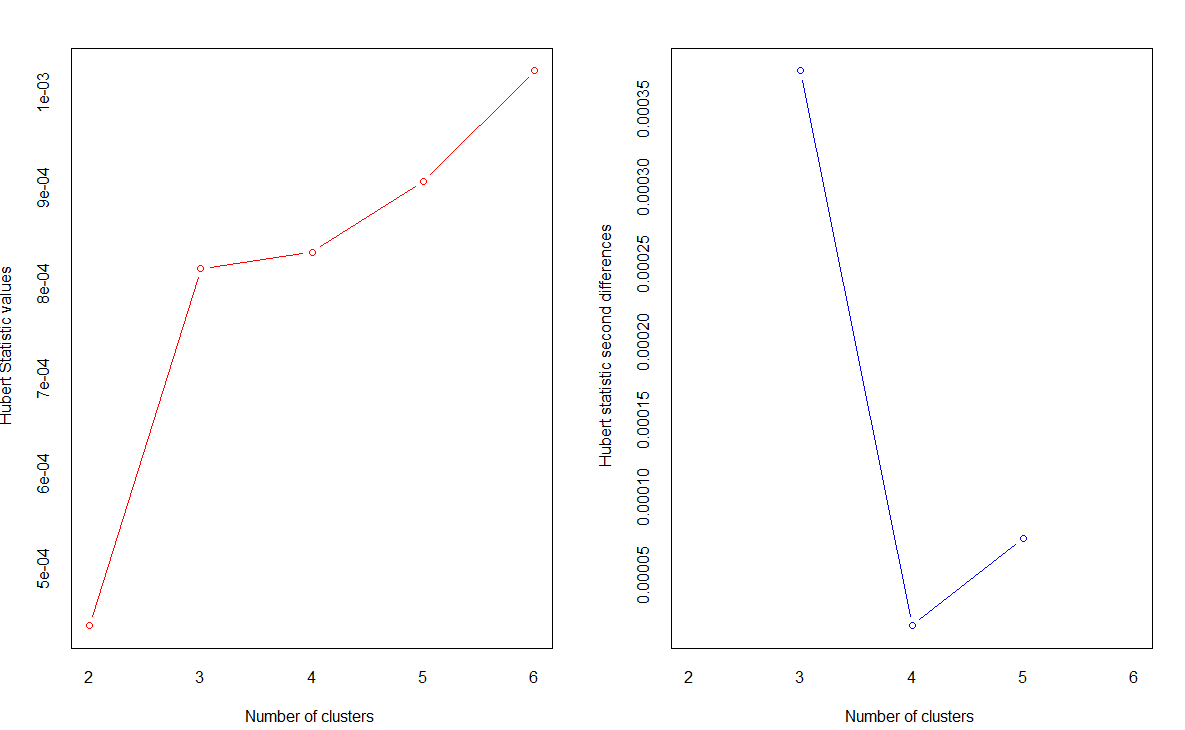


According to the plot, there is a distinct drop when moving from three to six clusters. After six and three clusters, this decrease drops off, suggesting that a three-cluster solution or a six-cluster solution can be a good fit to the data.

In the next run, specify a maximum of six clusters, and run the k-Means clustering algorithm again.







Among the indices, 14 proposed that 3 is the best number of clusters.

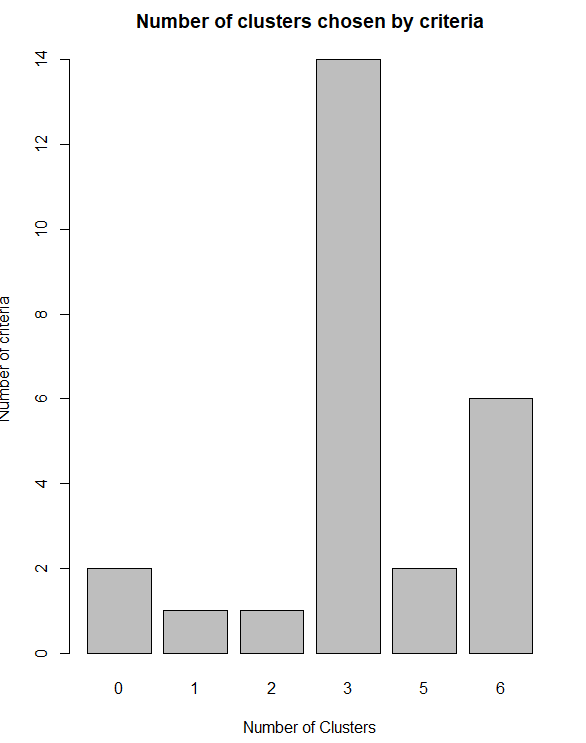
> table (nc$Best.n[1,])

0 1 2 3 5 6

2 1 1 14 2 6

> barplot(table(nc$Best.n[1,]), xlab="Number of Clusters", ylab="Number of criteria", main="Number of clusters chosen by criteria")

Hit <Return> to see next plot:



> # Perform k-means cluster analysis

The following code will perform the k means cluster analysis on 3 clusters.

> fit.km <- kmeans(dungaree.df.norm, 3 , nstart=25)

> fit.km$size

[1] 294 57 338

> fit.km$centers

FASHION LEISURE STRETCH ORIGINAL SALESTOT

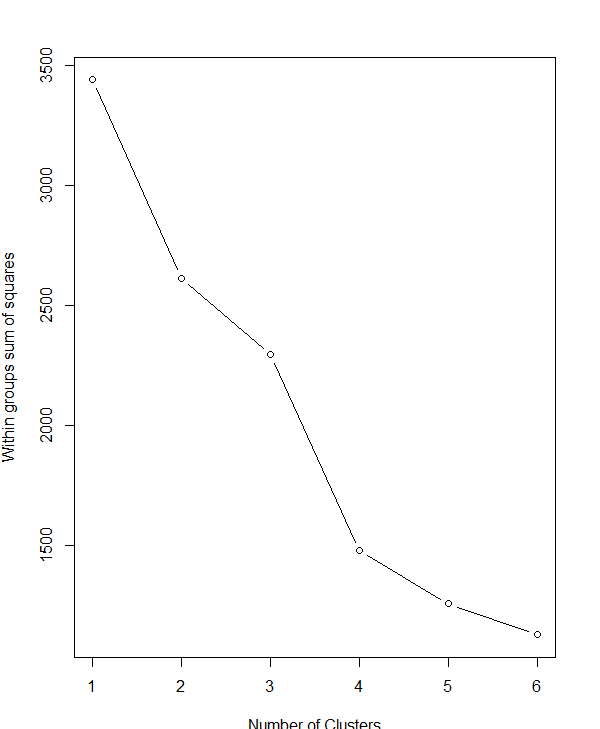
1 -0.07627649 0.9079054 -0.1499153 -0.6093102 0.2812516

2 -0.70173493 -0.9750246 -1.7741356 -1.2230385 -3.0502374

3 0.18468692 -0.6252893 0.4295883 0.7362438 0.2697502

> wssplot(dungaree.df.norm,nc=6,seed=42)

Hit <Return> to see next plot:



According to the plot, there is a distinct drop when moving from one to three clusters. After three clusters, this decrease drops off, suggesting that a three-cluster solution may be a good fit to the data.

Using the output, 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.

To interpret the characteristic of each cluster with the type of jeans, each variable which is jeans needs to have a statistic for each cluster including the mean and median. Based on that we can say that for Fashion type of jeans, centroid 6 is at maximum distance and centroid 5 is at the minimum distance. This means that stores that sell fashion type of jeans will be mainly present in cluster 5. Similarly, stores that sell leisure type of jeans will be mainly present in the cluster 3. Stores selling stretch jeans will most likely be present in the cluster 6 and stores selling original jeans will be present mainly in cluster 3. Also, there are stores which fall in all the clusters and thus sold all the types of jeans in same proportion. There are stores which do not fall in any cluster but one, which suggest that a particular type of jeans is sold in particular store.

Which clustering approach provides greater differentiation between clusters and interpretability? Justify with explanation.

If we perform clustering using 6 as maximum number of clusters and choosing 3 as the best number of clusters, it provides greater differentiation between clusters and interpretability that choosing 3 as the best number of cluster when 10 was chosen as the maximum number of clusters. This is because sales of each type of jeans can be distinguished in more clear manner than choosing just 3 clusters.