Tech gadgets Brand Classification

ABSTRACT

This project aims to analyze and discover the common standard features shared by various tech gadgets from various brands that help the brand stand out in the market.

PROBLEM STATEMENT

Each of the technological gadgets is available in a variety of brands. In this modern era, it is difficult to determine which brand a customer should purchase based on their preferences, so clustering analysis and PCA analysis are used to classify the brand based on the characteristics and standard features shared by each brand.

INTRODUCTION

Smartphone brand data is analyzed and clustered based on feature similarities, and the most important features that contribute the most to the brand are identified using dimensionality reduction using PCA.

METHODOLOGY USED

Clustering Analysis using K- Means Algorithm

Principle Component Analysis

Clustering- data classification based on characteristic similarity

PCA –to identify the variables that have the greatest impact on all brands

SOFTWARE USED

Data Collection - **Typeform**

Data Analysis - **R- Studio**

TECH GADGETS BRAND ANALYSIS

DATASET :

Different brands of tech gadgets with various features are chosen for analysis and collected via typeform.

Sample – 129 (Respondents)

Gadgets names – Smartphone, Mouse Keyboard, Headphone Camera and Laptop

Gadget: Smartphone

Source Code

```
library("factoextra") - extracts and visualizes the result of Multivariate data
library("NbClust") - determine the best number of clusters
library("dplyr") - resolves the data manipulation hurdles
library("cluster") -  perform the cluster analysis with the k-means algorithm
library(ggbiplot) - to visualize the PCA components in 2D
library("rstatix") - helper package of the univariate and multivariate data
library("FactoMineR") - to perform the principle component analysis
library(parameters) - Utilities for processing the parameters of various statistical models
```

Data Preprocessing

```
smartphone <-
  read.table(
    file = "D:/Files/CIT/M.Sc.DCS/4th Semester/17MDC46 - PA Lab/PA Sem IV project/smartphone.csv",
    sep = ",",
    dec = ".",
    header = TRUE,
  )

sm = smartphone[2:11]
#preprocessing the data by replacing the NA values with zero
sm[is.na(sm)] = 0
# sm =  scale(sm)
sm
```

```
> sm
   Brand Price Performance Quality Design Operate.Platform.system Value Get.used.to..Habit Reputation Services
1      7     8          10       0      0                       0     0                  0          0        3
2      0     8           0       0      0                       0     0                  0          0        0
3      7     8           0       9      0                       0     0                  0          0        3
4      7     0          10       9      0                       0     5                  0          0        0
5      0     0          10       0      0                       0     0                  0          0        0
6      7     0           0       0      0                       0     0                  0          0        0
7      0     8           0       0      0                       4     0                  0          0        0
8      0     8           0       0      0                       0     0                  0          0        0
9      0     0           0       9      0                       0     0                  0          0        0
10     0     0          10       0      0                       0     0                  0          0        0
11     0     0           0       9      0                       0     0                  0          0        0
12     7     0          10       9      6                       4     0                  0          0        0
13     7     8          10       9      0                       0     0                  0          1        3
```

The above smartphone data is preprocessed by selecting numeric columns and are scaled to normalize the data to perform the clustering analysis.

Optimal number of clusters and Quality of a *k*-means partition

```
#Elbow method

fviz_nbclust(sm, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2) +
  labs(subtitle = "Elbow method")

#silhouette method

fviz_nbclust(sm, kmeans , method = "silhouette")

#gap statistics method

gap_stat = clusGap(
  sm ,
  FUN = kmeans ,
  nstart = 25 ,
  K.max = 10 ,
  B = 50
```

```
)
fviz_gap_stat(gap_stat)

# Consensus Based Algorithm

n_clust <- n_clusters(sm,
                      package = c("easystats", "NbClust", "mclust"),
                      standardize = TRUE,fast = TRUE,
                      nbclust_method = "kmeans")
n_clust
```

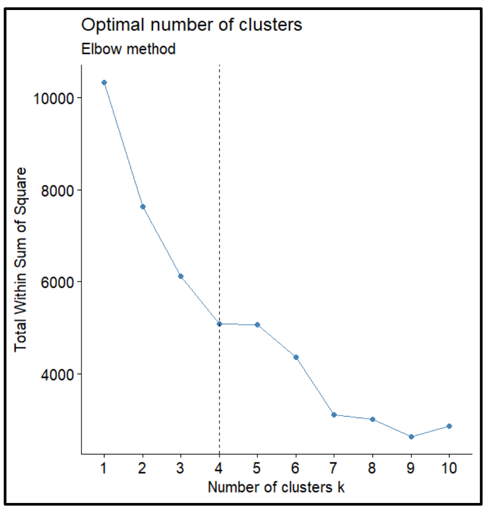
The **Elbow method** looks at the total within-cluster sum of square (WSS) as a function of the number of clusters. Here the optimal number is **4.** The Elbow method is sometimes ambiguous and an alternative is the average silhouette method.

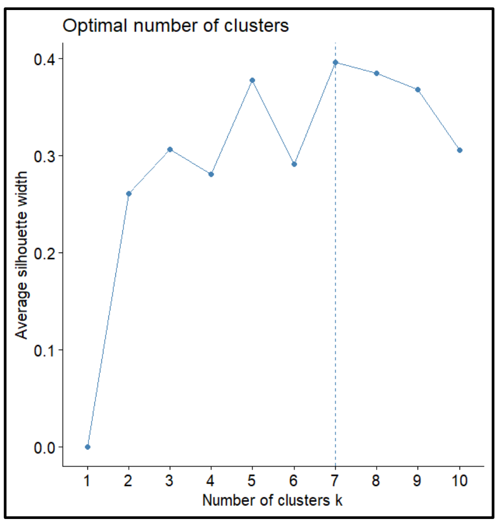
The **Silhouette method** measures the Quality of a clustering and determines how well each point lies within its cluster. The Silhouette method suggests **7** clusters.

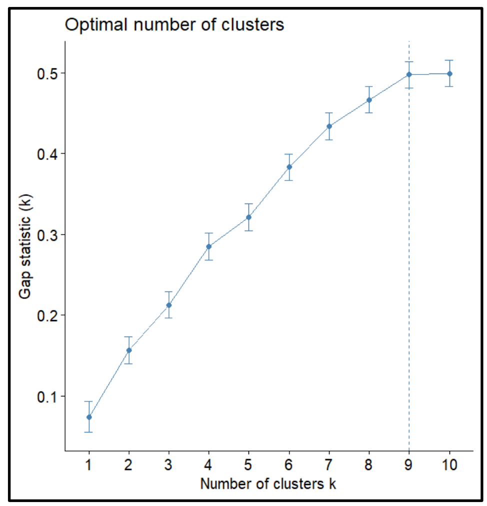
The optimal number of clusters is the one that maximizes the **gap statistic.** This method suggests only **9** clusters.

Here, the 3 approaches suggest a different number of clusters.

Because no method is clearly better, a fourth alternative is to run many methods and take the number of clusters that is the most agreed upon (i.e., find the consensus).

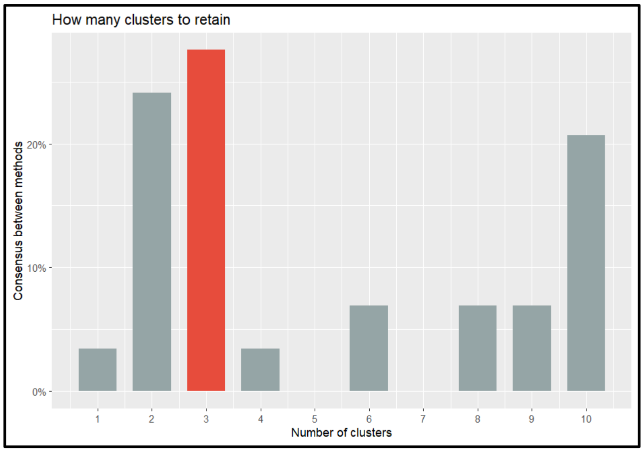







```
>
> # Consensus Based Algorithm
>
> n_clust <- n_clusters(sm,
+                       package = c("easystats", "NbClust", "mclust"),
+                       standardize = TRUE,fast = TRUE,
+                       nbclust_method = "kmeans")
> n_clust
# Method Agreement Procedure:

The choice of 3 clusters is supported by 8 (27.59%) methods out of 29 (Hartigan, Marriot, trcovw, Tracew, Ratkowsky, Ball, PtBiserial, Mixture (EII)).
>
```

Based on all indices, most methods suggest to retain **3 clusters**, followed by a 4-clusters solution.

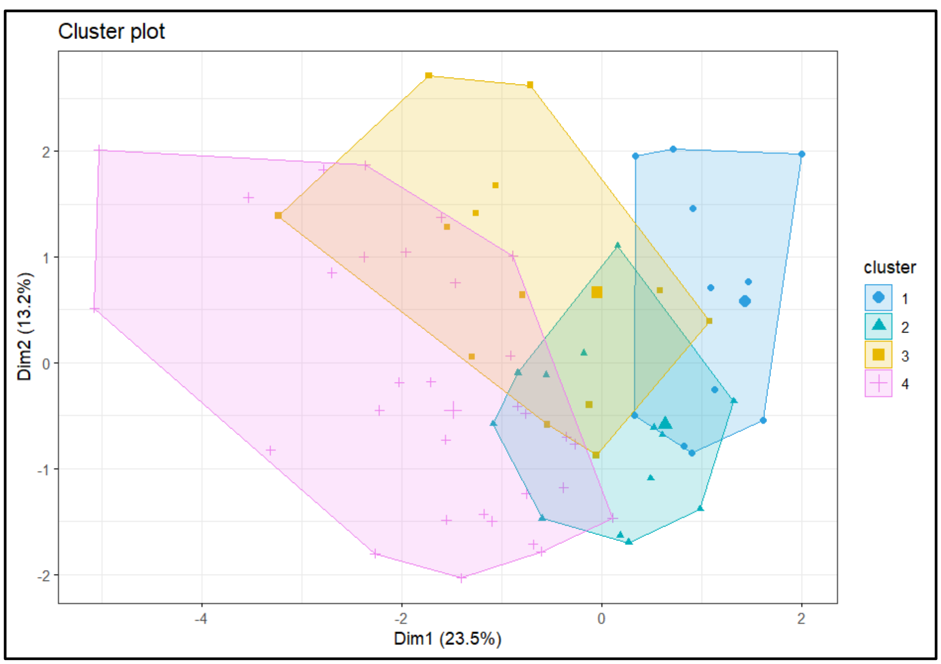


K- Means Clustering

```
# Elbow Method - 4 , silhouette = 7 , gap statistic - 9 , Consense based algorithm -3

smart_result_e = kmeans(sm , 4 , nstart = 25)
smart_result_e

fviz_cluster(
  smart_result_e,
  data = sm,
  palette = c("#2E9FDF", "#00AFBB", "#E7B800", "violet"),
  geom = "point",
  ellipse.type = "convex",
  ggtheme = theme_bw()
)
# silhoutte = 7
smart_result_s = kmeans(sm , 7)
smart_result_s

fviz_cluster(
  smart_result_s,
  data = sm,
  palette = c(
    "#2E9FDF",
    "#00AFBB",
    "#E7B800" ,
    "violet",
    "red",
    "pink",
    "green"
  ),
  geom = "point",
  ellipse.type = "convex",
  ggtheme = theme_bw()
)



# gap statistic - 9
smart_result_g = kmeans(sm , 9)
smart_result_g


fviz_cluster(
  smart_result_g,
  data = sm,
  palette = c(
    "#2E9FDF",
    "#00AFBB",
    "#E7B800" ,
    "violet",
    "red",
```

```
    "pink",
    "green",
    "brown",
    "yellow"
  ),
  geom = "point",
  ellipse.type = "convex",
  ggtheme = theme_bw()
)

# Final Optimal Cluster  #Consense based algorithm  is taken with the optimal clusters of 3

optimal_cluster = kmeans(sm , 3 , nstart = 25)
optimal_cluster



fviz_cluster(
  optimal_cluster,
  data = sm,
  palette = c("#2E9FDF",
              "#00AFBB",
              "#E7B800"),
  geom = "point",
  ellipse.type = "convex",
  ggtheme = theme_bw()
)

#quality of k-means partition
BSS = optimal_cluster$betweenss
TSS = optimal_cluster$totss
# We calculate the quality of the partition
BSS / TSS * 100
```

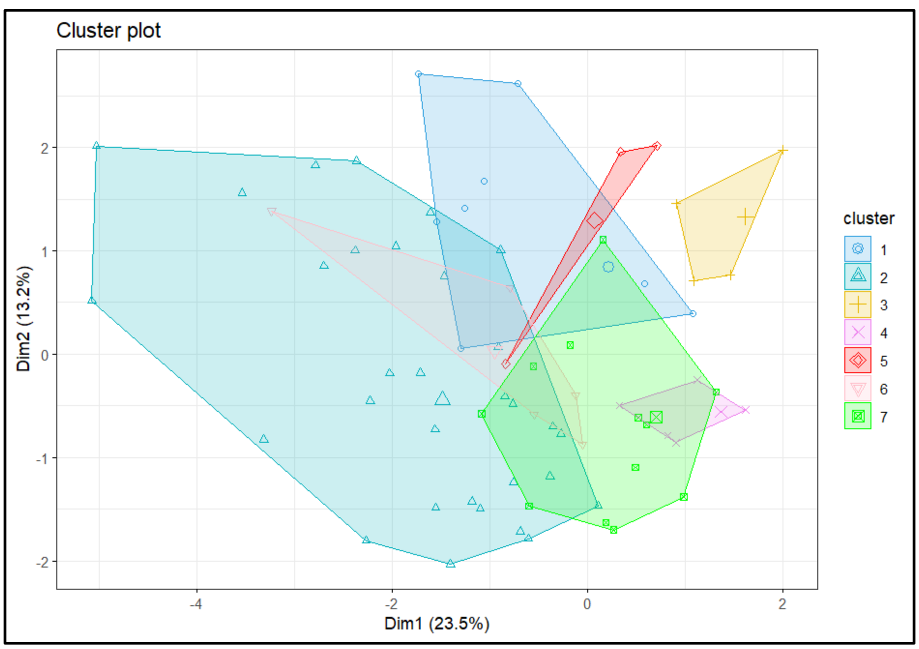
Elbow Method



```
K-means clustering with 4 clusters of sizes 31, 25, 22, 40

Cluster means:
     Brand    Price Performance Quality    Design Operate.Platform.system
1 2.032258 3.354839           0       0 0.7741935               0.1290323
2 1.680000 3.200000          10       0 0.9600000               0.8000000
3 1.590909 1.818182           0       9 1.6363636               0.3636364
4 3.675000 5.000000          10       9 3.1500000               1.3000000
     Value Get.used.to..Habit Reputation   Services
1 0.483871         0.58064516  0.0000000 0.09677419
2 0.400000         0.16000000  0.0400000 0.12000000
3 1.136364         0.09090909  0.1363636 0.81818182
4 1.750000         0.25000000  0.2000000 0.75000000

Clustering vector:
  [1] 2 1 3 4 2 1 1 1 3 2 3 4 4 2 1 2 2 4 2 3 3 3 1 3 1 2 2 4 4 2 4 2 1 4
 [35] 4 1 1 1 1 1 2 3 4 2 1 1 1 1 4 4 4 2 4 4 4 3 4 1 3 3 3 4 2 4 3 4 3 3
 [69] 2 4 4 1 4 2 4 4 3 4 2 1 1 1 2 1 4 4 2 3 4 1 3 4 3 4 3 4 1 2 2 2 1 4
[103] 2 2 1 4 1 4 3 4 1 3 1 4 4 4 1 4

Within cluster sum of squares by cluster:
[1] 1038.9677  855.3600  765.0455 1907.1750
 (between_SS / total_SS =  55.8 %)
```

The WSS value of cluster 4 in Elbow method is high 1907.1750 which means that there is dissimilarity in the clusters members

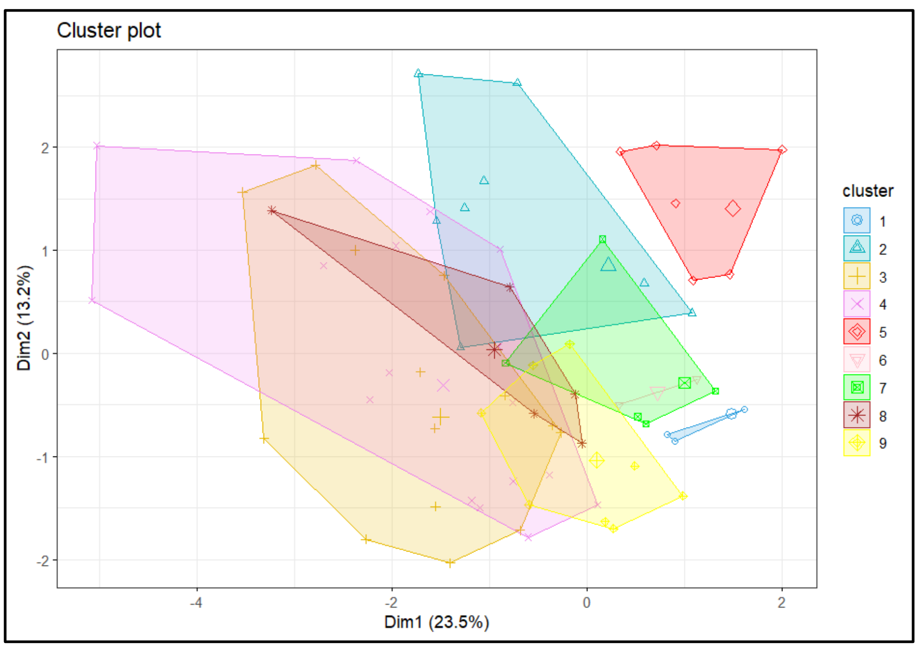
Silhouette Method



```
K-means clustering with 7 clusters of sizes 17, 40, 16, 13, 3, 5, 24

Cluster means:
     Brand    Price Performance Quality    Design Operate.Platform.system
1 1.235294 0.000000    0.000000       9 1.0588235               0.4705882
2 3.675000 5.000000   10.000000       9 3.1500000               1.3000000
3 2.625000 0.000000    0.000000       0 0.0000000               0.0000000
4 1.076923 8.000000    0.000000       0 0.9230769               0.3076923
5 4.666667 0.000000    3.333333       0 6.0000000               0.0000000
6 2.800000 8.000000    0.000000       9 3.6000000               0.0000000
7 1.458333 3.333333   10.000000       0 0.7500000               0.8333333
      Value Get.used.to..Habit Reputation  Services
1 0.8823529          0.1176471 0.11764706 0.7058824
2 1.7500000          0.2500000 0.20000000 0.7500000
3 0.6250000          0.8750000 0.00000000 0.1875000
4 0.0000000          0.0000000 0.00000000 0.0000000
5 3.3333333          1.3333333 0.00000000 0.0000000
6 2.0000000          0.0000000 0.20000000 1.2000000
7 0.2083333          0.1666667 0.04166667 0.1250000

Clustering vector:
  [1] 7 4 6 2 7 3 4 4 1 7 1 2 2 7 3 7 7 2 7 1 1 1 3 1 3 7 7 2 2 7 2 7 4 2
 [35] 2 5 4 3 4 3 5 1 2 7 3 3 3 3 2 2 2 7 2 2 2 1 2 3 1 1 6 2 7 2 1 2 1 6
 [69] 7 2 2 4 2 7 2 2 1 2 7 3 3 3 7 3 2 2 7 1 2 4 6 2 1 2 1 2 4 7 7 7 5 2
[103] 7 7 4 2 4 2 6 2 4 1 3 2 2 2 4 2

Within cluster sum of squares by cluster:
[1]  333.0588 1907.1750  251.6875  158.6154  118.6667  143.6000  766.0000
 (between_SS / total_SS =  64.4 %)
```

The WSS value of cluster 2 in Silhouette method is high 1907.1705 which means that there is dissimilarity in the cluster's members

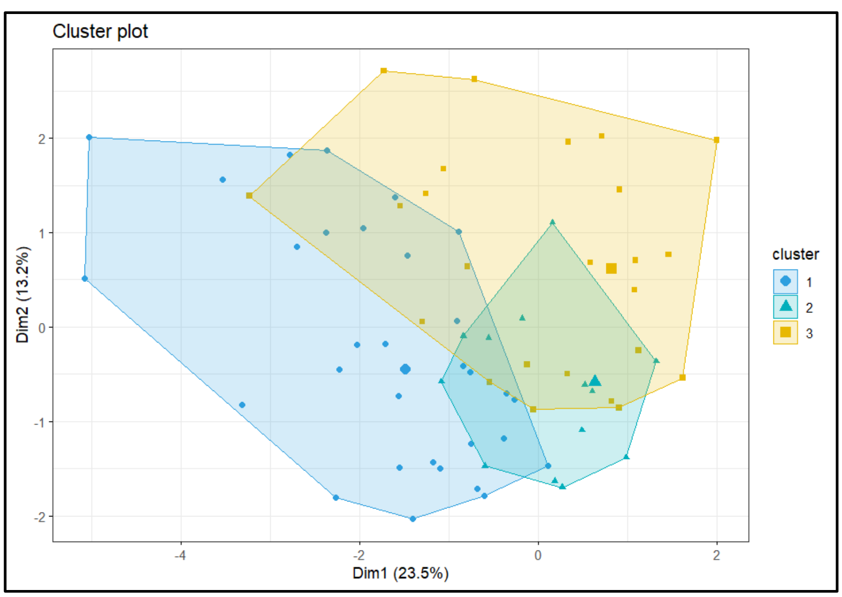
Gap Statistics



```
K-means clustering with 9 clusters of sizes 11, 17, 18, 22, 18, 2, 15, 5, 10

Cluster means:
      Brand    Price Performance Quality    Design
1 0.0000000 8.000000           0       0 0.5454545
2 1.2352941 0.000000           0       9 1.0588235
3 2.3333333 3.555556          10       9 5.3333333
4 4.7727273 6.181818          10       9 1.3636364
5 2.7222222 0.000000           0       0 0.6666667
6 7.0000000 8.000000           0       0 3.0000000
7 0.9333333 0.000000          10       0 0.8000000
8 2.8000000 8.000000           0       9 3.6000000
9 2.8000000 8.000000          10       0 1.2000000
  Operate.Platform.system     Value Get.used.to..Habit Reputation
1               0.3636364 0.0000000          0.0000000  0.0000000
2               0.4705882 0.8823529          0.1176471  0.1176471
3               1.5555556 1.9444444          0.2222222  0.1111111
4               1.0909091 1.5909091          0.2727273  0.2727273
5               0.0000000 0.8333333          1.0000000  0.0000000
6               0.0000000 0.0000000          0.0000000  0.0000000
7               0.5333333 0.3333333          0.1333333  0.0000000
8               0.0000000 2.0000000          0.0000000  0.2000000
9               1.2000000 0.5000000          0.2000000  0.1000000
   Services
1 0.0000000
2 0.7058824
3 0.5000000
4 0.9545455
5 0.1666667
6 0.0000000
7 0.0000000
8 1.2000000
9 0.3000000

Clustering vector:
  [1] 9 1 8 4 7 5 1 1 2 7 2 3 4 7 5 7 9 4 9 2 2 2 5 2 5 9 7 3 3 7 4 9 1 4
 [35] 4 5 6 5 1 5 7 2 3 9 5 5 5 5 4 4 3 7 3 4 4 2 3 5 2 2 8 3 9 3 2 4 2 8
 [69] 7 4 4 1 4 7 4 3 2 4 9 5 5 5 9 5 3 4 7 2 4 1 8 3 2 4 2 3 6 7 9 7 5 3
[103] 7 7 1 3 1 4 8 4 1 2 5 3 3 3 1 4

Within cluster sum of squares by cluster:
[1]  47.27273 333.05882 751.22222 867.04545 362.61111  18.00000 202.13333
[8] 143.60000 243.90000
 (between_SS / total_SS =  71.3 %)
```

The WSS value of cluster 4 in Gap statistic method is high 867.04545 which means that there is dissimilarity in the clusters members

Optimal Clusters



```
K-means clustering with 3 clusters of sizes 40, 25, 53

Cluster means:
     Brand    Price Performance  Quality   Design Operate.Platform.system
1 3.675000 5.000000          10 9.000000 3.150000               1.3000000
2 1.680000 3.200000          10 0.000000 0.960000               0.8000000
3 1.849057 2.716981           0 3.735849 1.132075               0.2264151
     Value Get.used.to..Habit Reputation  Services
1 1.750000          0.2500000 0.20000000 0.7500000
2 0.400000          0.1600000 0.04000000 0.1200000
3 0.754717          0.3773585 0.05660377 0.3962264

Clustering vector:
  [1] 2 3 3 1 2 3 3 3 3 2 3 1 1 2 3 2 2 1 2 3 3 3 3 3 3 2 2 1 1 2 1 2 3 1
 [35] 1 3 3 3 3 3 2 3 1 2 3 3 3 3 1 1 1 2 1 1 1 3 1 3 3 3 3 1 2 1 3 1 3 3
 [69] 2 1 1 3 1 2 1 1 3 1 2 3 3 3 2 3 1 1 2 3 1 3 3 1 3 1 3 1 3 2 2 2 3 1
[103] 2 2 3 1 3 1 3 1 3 3 3 1 1 1 3 1

Within cluster sum of squares by cluster:
[1] 1907.175  855.360 2904.981
 (between_SS / total_SS =  45.1 %)
```

The WSS value of cluster 3 in Consensus based Algorithm method is high 867.04545 which means that there is dissimilarity in the clusters members

QUALITY TESTING

Centers= 3 (consensus based algorithm)

With scaling

```
> #quality of k-means partition
> BSS = smart_result_e$betweenss
> TSS = smart_result_e$totss
> # We calculate the quality of the partition
> BSS / TSS * 100
[1] 27.13615
>
```

Without scaling

```
> #quality of k-means partition
> BSS = smart_result_e$betweenss
> TSS = smart_result_e$totss
> # We calculate the quality of the partition
> BSS / TSS * 100
[1] 45.12983
>
```

Centers= 4

With scaling

```
> #quality of k-means partition
> BSS = smart_result_e$betweenss
> TSS = smart_result_e$totss
> # We calculate the quality of the partition
> BSS / TSS * 100
[1] 34.4876
>
```

Without scaling

```
> #quality of k-means partition
> BSS = smart_result_e$betweenss
> TSS = smart_result_e$totss
> # We calculate the quality of the partition
> BSS / TSS * 100
[1] 55.78888
>
```

The classification into **four or more** groups allows for a higher explained percentage and a higher quality.

This will always be the case: with more classes, the partition will be finer, and the *BSS* contribution will be higher. On the other hand, the "model" will be more complex, requiring more classes. In the extreme case where $k = n$ (each observation is a singleton class), we have $BSS = TSS$, but the partition has lost all interest.

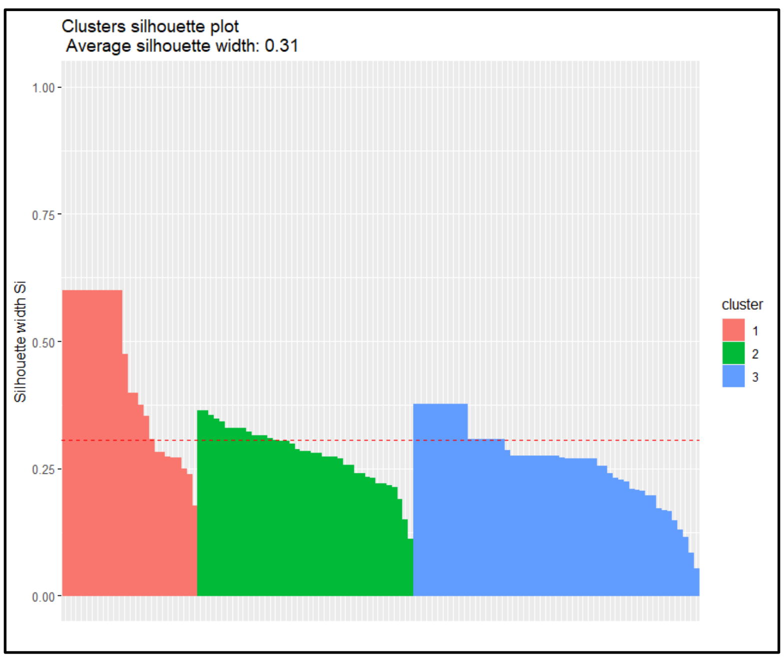
This is the reason we compare partitions via their Quality only for partitions that have the same number of clusters.

So we choose the **3 clusters (without scaling)** based on the optimality using the **consensus based algorithm** The `nstart()` argument in the function also allows to run the algorithm several times with different initial centers, in order to obtain a potentially better partition

Visualizations

To confirm that your number of classes is indeed optimal, there is a way to evaluate the Quality of your clustering via the silhouette plot (which shows the silhouette coefficient on the y axis).

We draw the silhouette plot for 3 clusters, as suggested by the average silhouette method:

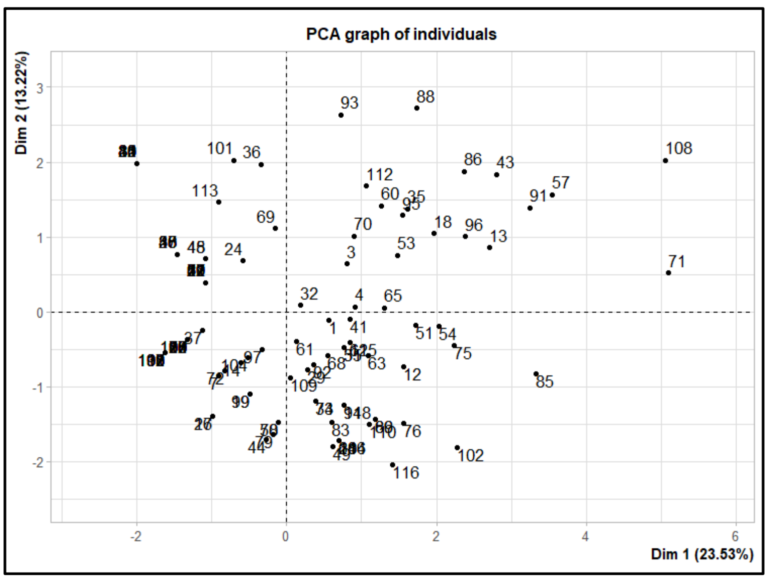


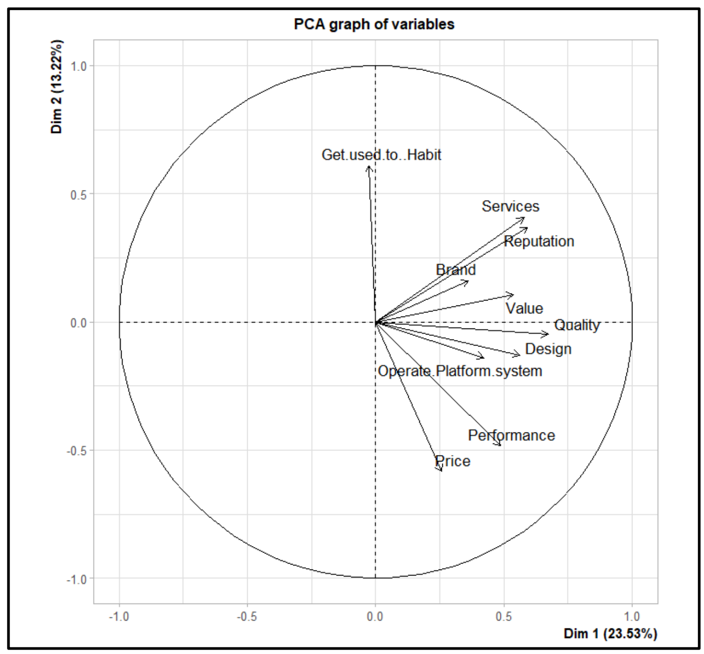
The silhouette plot above and the average silhouette coefficient say that the clustering is good and the clusters are optimal as the value is greater than zero means that the observation is well grouped. The closer the coefficient is to 1, the better the observation is grouped.

Principle Component Analysis

```
res.pca = PCA(sm)
res.pca
```

```
> res.pca = PCA(sm)
> res.pca
**Results for the Principal Component Analysis (PCA)**
The analysis was performed on 118 individuals, described by 10 variables
*The results are available in the following objects:

   name               description
1  "$eig"             "eigenvalues"
2  "$var"             "results for the variables"
3  "$var$coord"       "coord. for the variables"
4  "$var$cor"         "correlations variables - dimensions"
5  "$var$cos2"        "cos2 for the variables"
6  "$var$contrib"     "contributions of the variables"
7  "$ind"             "results for the individuals"
8  "$ind$coord"       "coord. for the individuals"
9  "$ind$cos2"        "cos2 for the individuals"
10 "$ind$contrib"     "contributions of the individuals"
11 "$call"            "summary statistics"
12 "$call$centre"     "mean of the variables"
13 "$call$ecart.type" "standard error of the variables"
14 "$call$row.w"      "weights for the individuals"
15 "$call$col.w"      "weights for the variables"
>
```



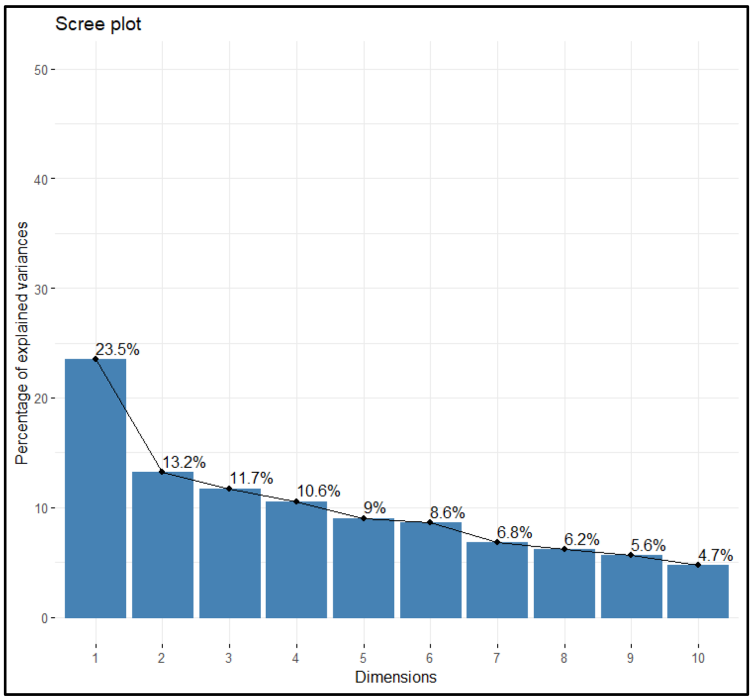


The PCA() function performs PCA analysis for the smartphone data and plots the necessary variables in that contributes the most in the dataset

Eigen Values /Variances

```
#Extract and visualize the eigen values
get_eig(res.pca)
#Visualize the eigen values/variances
fviz_screeplot(res.pca , addlabels = TRUE, ylim = c(0, 50))
```

```
> #Extract and visualize the eigen values
> get_eig(res.pca)
        eigenvalue variance.percent cumulative.variance.percent
Dim.1    2.3529526        23.529526                    23.52953
Dim.2    1.3220895        13.220895                    36.75042
Dim.3    1.1736112        11.736112                    48.48653
Dim.4    1.0556305        10.556305                    59.04284
Dim.5    0.9022899         9.022899                    68.06574
Dim.6    0.8591302         8.591302                    76.65704
Dim.7    0.6796218         6.796218                    83.45326
Dim.8    0.6207263         6.207263                    89.66052
Dim.9    0.5617762         5.617762                    95.27828
Dim.10   0.4721718         4.721718                   100.00000
>
```



The sum of all the eigenvalues gives a total variance of 10.

The proportion of variation explained by each eigenvalue is given in the second column .The PC1 and PC2 has 36% variability over the data

Contribution of the Variables

```
# Extract the results from the variables
var =  get_pca_var(res.pca)
var

#Coordinates of variables
head(var$coord)

#Contribution of variables
head(var$contrib)
```

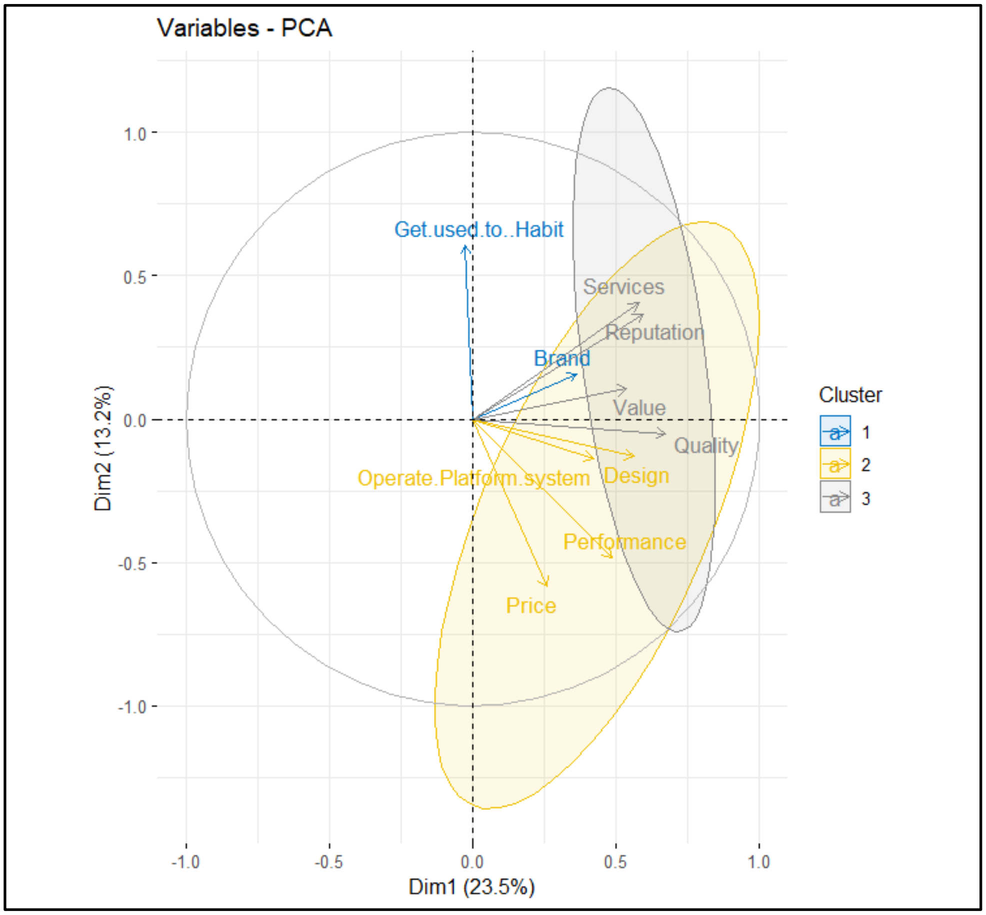
```
sm.pca <- prcomp(sm, center = TRUE, scale. = TRUE)
summary(sm.pca)
```

```
> var
Principal Component Analysis Results for variables
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the variables"
2 "$cor"     "Correlations between variables and dimensions"
3 "$cos2"    "Cos2 for the variables"
4 "$contrib" "contributions of the variables"
>
> #Coordinates of variables
> head(var$coord)
                            Dim.1       Dim.2      Dim.3        Dim.4       Dim.5
Brand                   0.3618670  0.15861147  0.3643670  0.683767802 -0.25349311
Price                   0.2572620 -0.58457364 -0.1490811  0.166685693  0.64438628
Performance             0.4866718 -0.48430004  0.3206401  0.065754001 -0.02844441
Quality                 0.6726847 -0.04976176 -0.1972057 -0.002759058 -0.36173487
Design                  0.5606938 -0.13072623  0.1230075 -0.063400643  0.05810829
Operate.Platform.system 0.4210595 -0.14003461  0.6041283 -0.335211960 -0.05576266
>
> #Contribution of variables
> head(var$contrib)
                            Dim.1      Dim.2     Dim.3        Dim.4       Dim.5
Brand                    5.565252  1.9028665 11.312374 4.428997e+01  7.12174168
Price                    2.812794 25.8474436  1.893742 2.631993e+00 46.01998253
Performance             10.066051 17.7405933  8.760145 4.095741e-01  0.08967013
Quality                 19.231354  0.1872969  3.313711 7.211237e-04 14.50222474
Design                  13.360980  1.2926013  1.289256 3.807811e-01  0.37422267
Operate.Platform.system  7.534834  1.4832349 31.098119 1.064454e+01  0.34462025
>
> sm.pca <- prcomp(sm, center = TRUE, scale. = TRUE)
> summary(sm.pca)
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8     PC9    PC10
Standard deviation     1.5339 1.1498 1.0833 1.0274 0.94989 0.92689 0.82439 0.78786 0.74952 0.68715
Proportion of Variance 0.2353 0.1322 0.1174 0.1056 0.09023 0.08591 0.06796 0.06207 0.05618 0.04722
Cumulative Proportion  0.2353 0.3675 0.4849 0.5904 0.68066 0.76657 0.83453 0.89661 0.95278 1.00000
```

The variables **Performance, Quality, and Design** contribute the most to the PC1 component, while **Price and Performance** contribute the most to the PC2 component. This means that the greater the contribution value, the more the variable contributes to the component.

Validating the variables with clustering

```
# Contribution of the variables with the
res.km <- kmeans(var$coord, centers = 3, nstart = 25)
grp <- as.factor(res.km$cluster)
# Color variables by groups
fviz_pca_var(res.pca, col.var = grp,
             palette = c("#0073C2FF", "#EFC000FF", "#868686FF"),
             legend.title = "Cluster",addEllipses = TRUE ,repel = TRUE)

ggbiplot(
  sm.pca,
  ellipse = TRUE,
  obs.scale = 1,
  var.scale = 1,
  var.axes = T,
  groups = smartphone$Smartphone
) + theme_minimal()
```



In the above graph, the variables **Get. Used to. Habit, Brand** contributes more to the cluster 1

Services, Reputation, Value, and Quality contribute more to the cluster 2

Operating Platform/System, Design, Performance, and Price contribute more to the cluster 3

Because the above-mentioned variables contributed more to a single cluster, this does not imply that it does not contribute to other groups, but their contribution is less than that of its cluster.

Clusters are formed based on the above criteria. When comparing the clusters with the brands, the above characteristic best fits the top three brands, namely **Samsung, Xiaomi, and Apple** and we can say that customers buys smartphones from these brands based on the characteristic depicted by each of the three clusters regarding the variables.

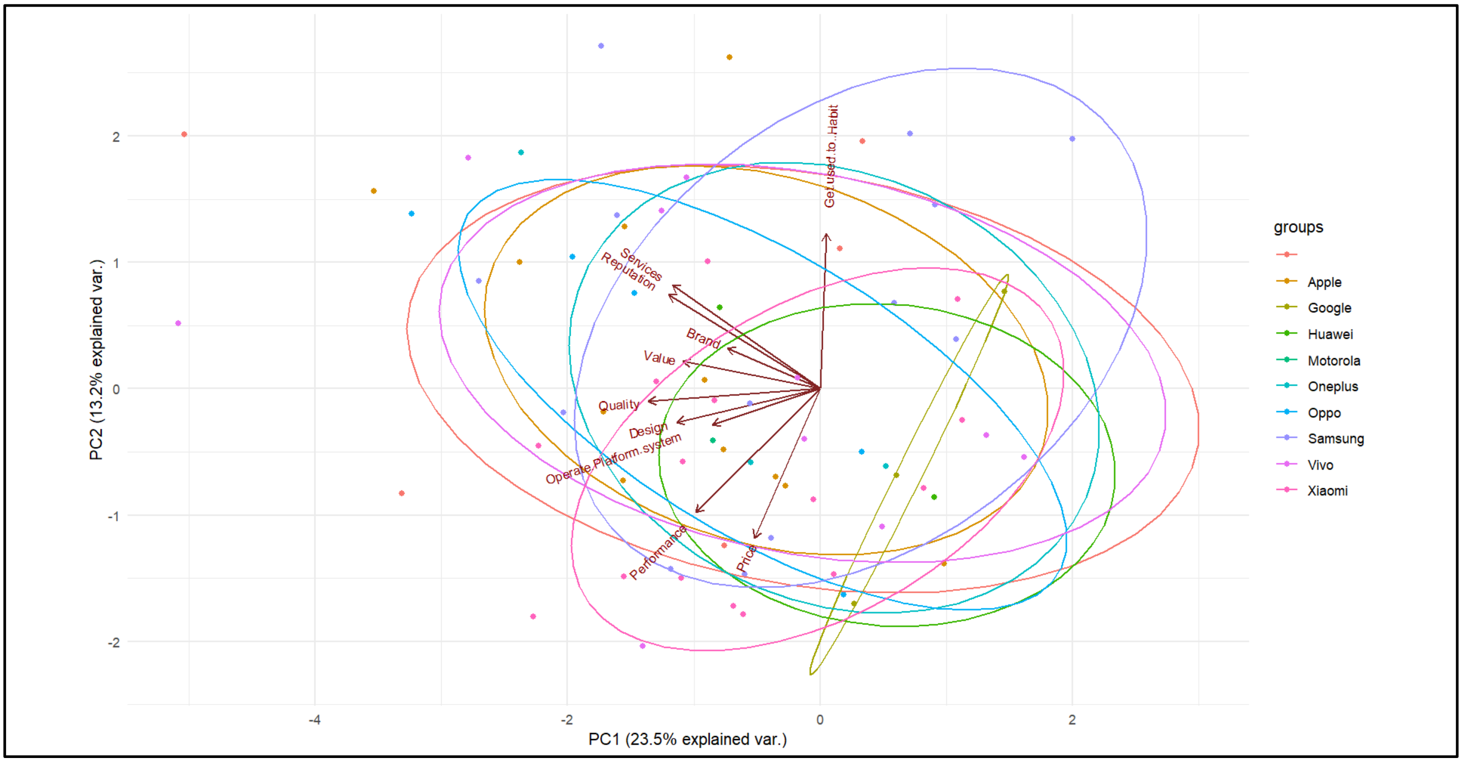
For example :

The person who seeks for best design, price, and performance can buy Apple Smartphones

The person who is Brand-specific will buy Samsung smartphones

The person who strives for the best Quality goes for Xiaomi smartphones

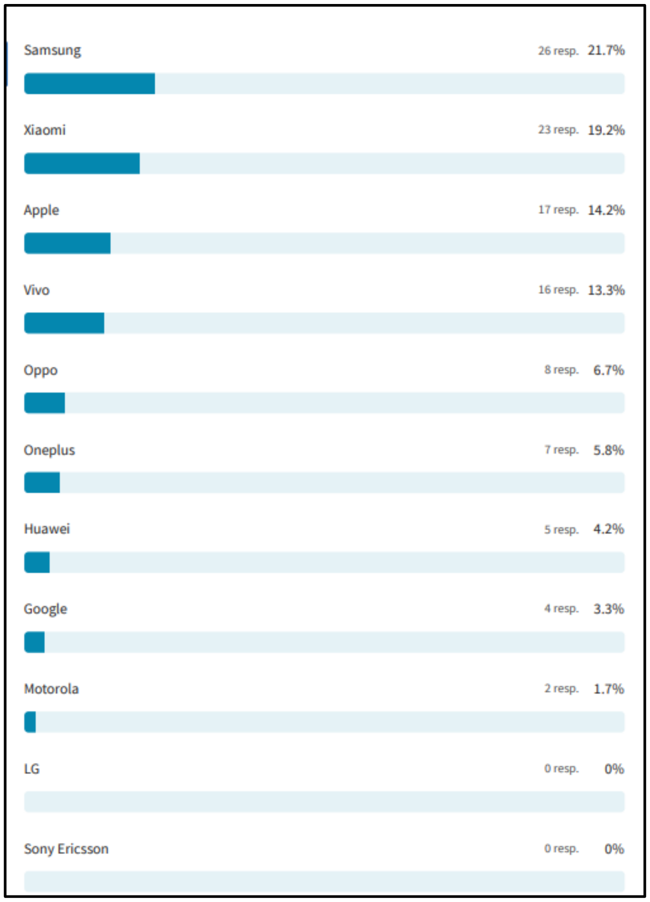
The above graph represents the overall brands and the variables that contributes for each of the brand on a 2 D plane.



CONCLUSION

The above clustering analysis creates clusters based on customers' preferences for brands, and through PCA analysis, the features that contribute the most to each brand of smartphone are identified and validated with the clusters, making it easier for consumers to select brands based on their preferences. For each brand, the output is verified and compared with survey respondents. The techniques described above can also be applied to other types of technology to determine brand categorization based on features.

- **Smartphone - Samsung**
- **Mouse - Logitech**
- **Headphone - Oneplus**
- **Camera – Canon**
- **Keyboard - Apple**
- **Laptop - HP**



Samsung
26 resp. 21.7%
Xiaomi
23 resp. 19.2%
Apple
17 resp. 14.2%
Vivo
16 resp. 13.3%
Oppo
8 resp. 6.7%
Oneplus
7 resp. 5.8%
Huawei
5 resp. 4.2%
Google
4 resp. 3.3%
Motorola
2 resp. 1.7%
LG
0 resp. 0%
Sony Ericsson
0 resp. 0%

REFERENCE

Clustering - https://statsandr.com/blog/clustering-analysis-k-means-and-hierarchical-clustering-by-hand-and-in-r/

https://www.r-bloggers.com/2021/04/cluster-analysis-in-r/

PCA -

http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/112-pca-principal-component-analysis-essentials/