Clustering Analysis in R
Shopping Dataset

Swarnalatha Sethuraman

2024-07-01

Clustering Analysis

Clustering is a type of unsupervised learning algorithm that involves grouping similar data points together based on their characteristics. The goal of clustering is to find similarities within a dataset and group similar data points together while keeping dissimilar data points separate.

About Dataset

Shopping Customer Data is a comprehensive dataset that provides a detailed analysis of a hypothetical shop's ideal customers. By collecting and analyzing customer data, it provides valuable insights that can help a business better understand its customers.

The dataset includes 200 records and 5 columns, providing a wealth of information about the shop's customer base. Each column represents a specific aspect of the customer's profile, including their unique Customer ID, Gender, Age, Annual Income and Spending Score.

By analyzing this data, businesses can gain valuable insights into their customers' preferences, behaviors, and purchasing habits. For example, they can segment customers by age, income, or gender to better understand how these factors impact their purchasing decisions.

#Import the dataset

```
data <- read.csv("datasets/shoppingdata.csv")
```

#View the first 3 rows of the data

```
head(data,n=3)
  CustomerID Gender Age Annual.Income..k.. Spending.Score..1.100.
1          1   Male  19                 15                     39
2          2   Male  21                 15                     81
3          3 Female  20                 16                      6
```

#Structure of data

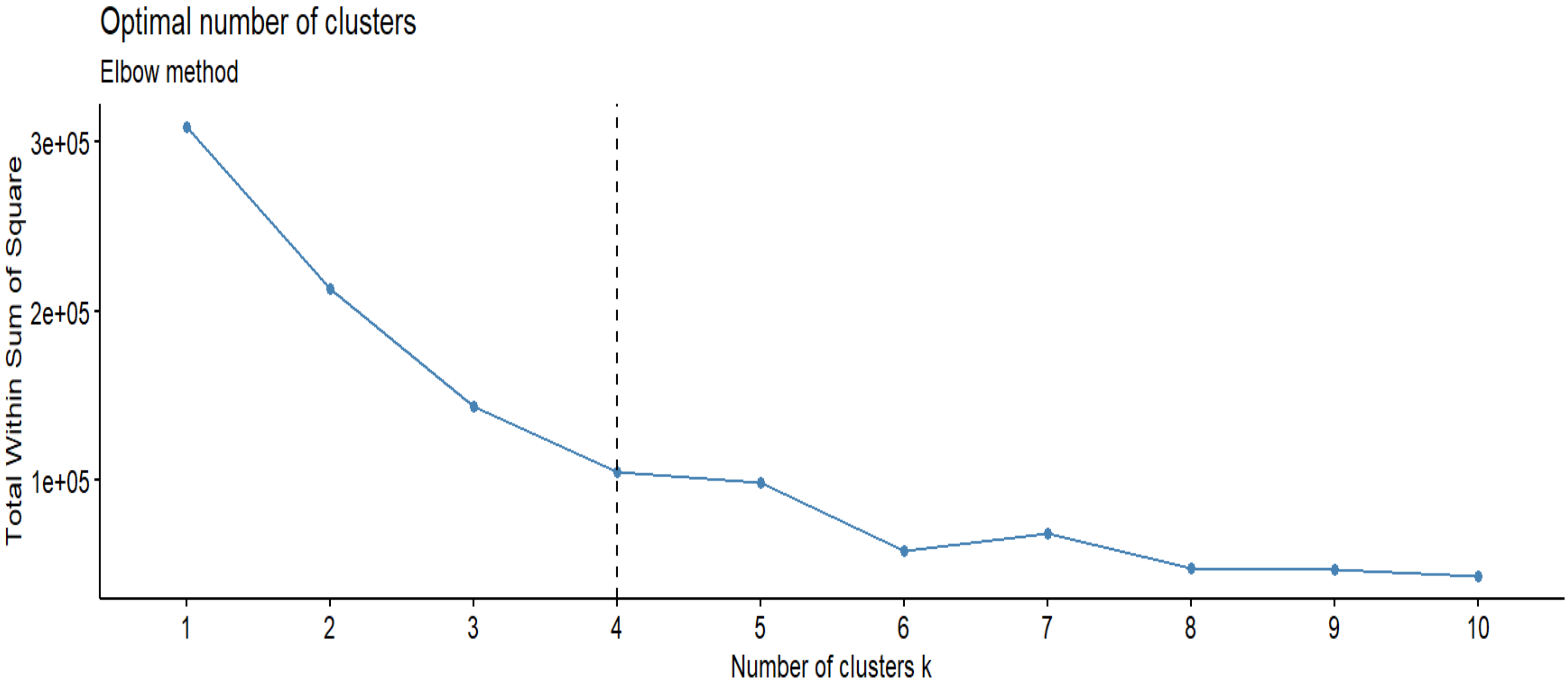
```
str(data)
'data.frame': 200 obs. of  5 variables:
 $ CustomerID            : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Gender                : chr  "Male" "Male" "Female" "Female" ...
 $ Age                   : int  19 21 20 23 31 22 35 23 64 30 ...
 $ Annual.Income..k..    : int  15 15 16 16 17 17 18 18 19 19 ...
 $ Spending.Score..1.100.: int  39 81 6 77 40 76 6 94 3 72 ...
```

#Summary of data

```
summary(data)
  CustomerID        Gender               Age        Annual.Income..k.. Spending.Score..1.100.
 Min.   :  1.00   Length:200         Min.   :18.00   Min.   : 15.00     Min.   : 1.00
 1st Qu.: 50.75   Class :character   1st Qu.:28.75   1st Qu.: 41.50     1st Qu.:34.75
 Median :100.50   Mode  :character   Median :36.00   Median : 61.50     Median :50.00
 Mean   :100.50                      Mean   :38.85   Mean   : 60.56     Mean   :50.20
 3rd Qu.:150.25                      3rd Qu.:49.00   3rd Qu.: 78.00     3rd Qu.:73.00
 Max.   :200.00                      Max.   :70.00   Max.   :137.00     Max.   :99.00
```

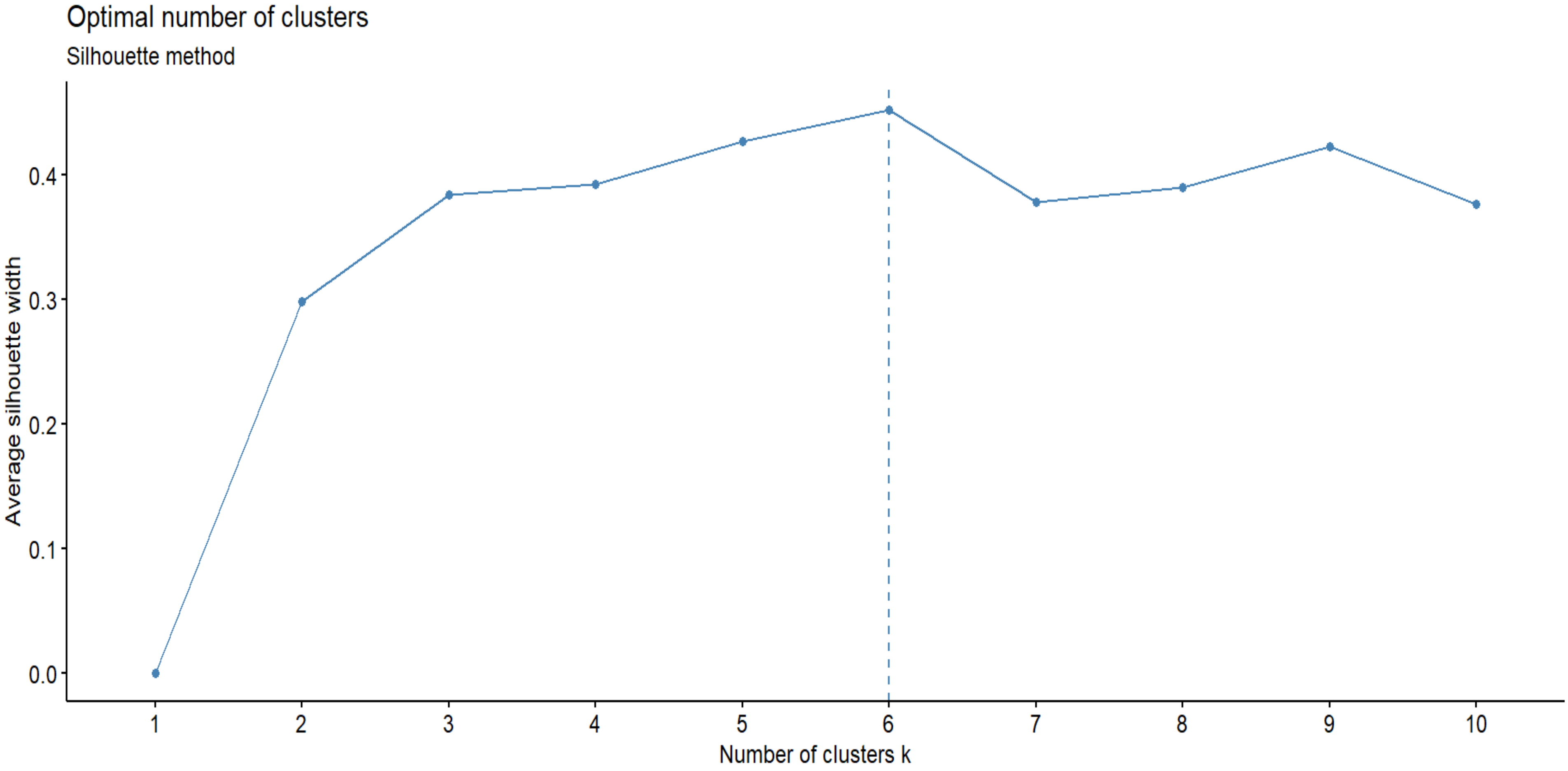
```
#Finding missing values in data
sum(is.na(data))
0

#Creating new column of gender column with numeric values
data$Sex <- NA
data$Sex <- ifelse(data$Gender=="Female",1,0)
head(data,3)
CustomerID Gender Age Annual.Income..k.. Spending.Score..1.100. Sex
1         1     Male  19                15                     39   0
2         2     Male  21                15                     81   0
3         3   Female  20                16                      6   1

#Kmeans works with numeriacal data hence subsetting the dataset without ID and Gender
df <- subset(data,select=c(3,4,5,6))
head(df,n=3)
Age Annual.Income..k.. Spending.Score..1.100. Sex
1  19                15                     39   0
2  21                15                     81   0
3  20                16                      6   1
```

```
#Plot for optimal number of clusters
library(factoextra)
# Elbow method
fviz_nbclust(df, kmeans, method = “wss”) + geom_vline(xintercept = 4, linetype = 2)
```



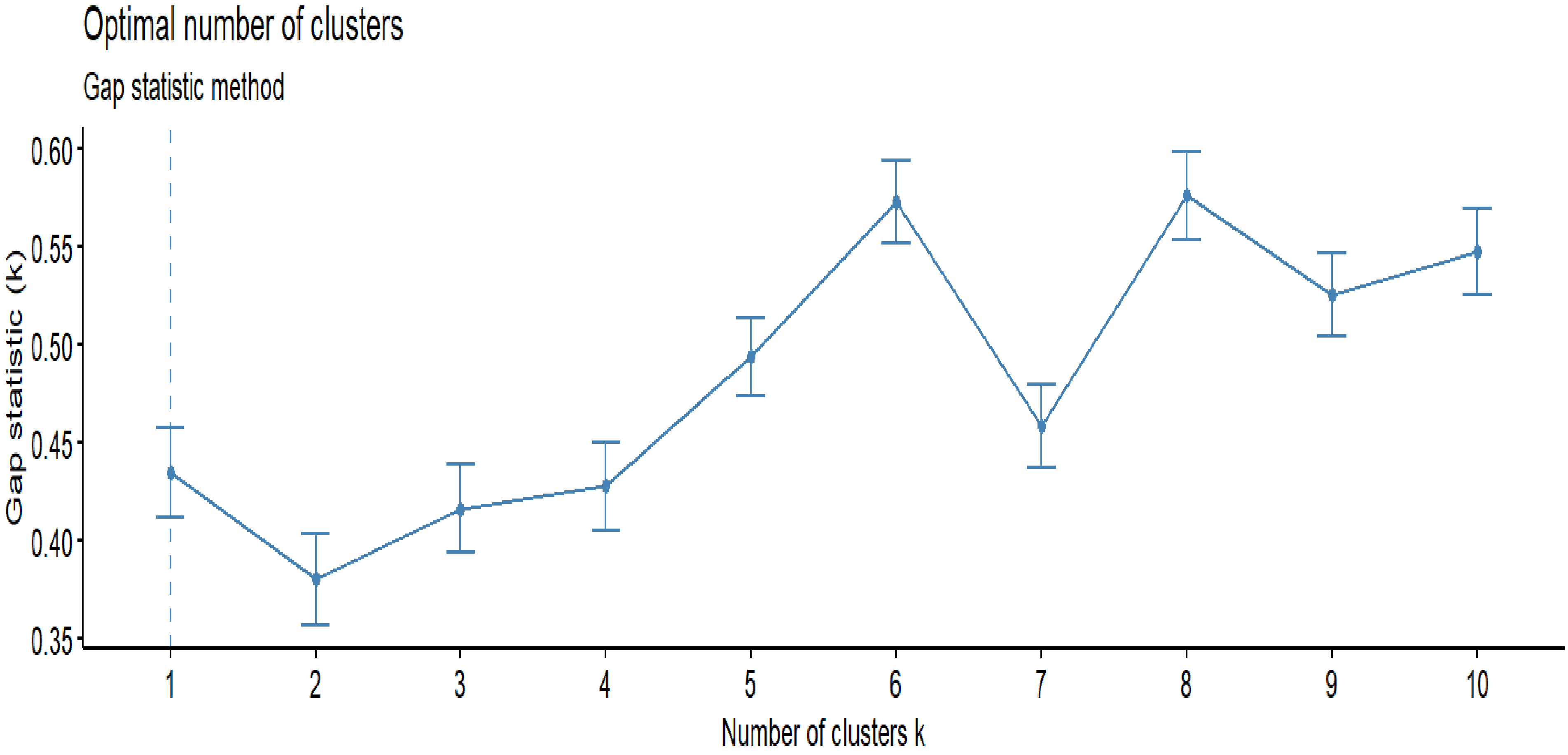
Silhouette method

```
fviz_nbclust(df, kmeans, method = "silhouette")+  labs(subtitle = "Silhouette method")
```



Gap statistic method

```
fviz_nbclust(df, kmeans, method = "gap_stat")+  labs(subtitle = "Gap statistic method")
```



K Means Clustering Analysis

K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

```
#Compute k-means with k = 4
set.seed(12)
km.res <- kmeans(df, 4, nstart = 1)
```

Print the results

```
print(km.res)
K-means clustering with 4 clusters of sizes 40, 38, 53, 69
Cluster means:
       Age Annual.Income..k.. Spending.Score..1.100.       Sex
1 32.87500          86.10000               81.52500 0.5500000
2 40.39474          87.00000               18.63158 0.4736842
3 25.05660          40.73585               62.62264 0.5849057
4 52.05797          46.42029               39.88406 0.5942029
Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 
  3   3   4   3   3   3   4   3   4   3   4   3   4   3   4   3   4   3   4   3   4   3   4   3 
 25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 
  4   3   4   3   4   3   4   3   4   3   4   3   4   3   4   3   4   3   4   3   4   3   4   3 
 49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72 
  3   3   4   3   3   4   4   4   4   4   3   4   4   3   4   4   4   3   4   4   3   3   4   4 
 73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 
  4   4   4   3   4   4   3   4   4   3   4   4   3   4   4   3   3   4   4   3   4   4   4   3 
 97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 
  4   3   4   3   3   4   4   3   4   3   4   4   4   4   4   3   4   3   3   3   4   4   4   4 
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 
  3   4   1   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1 
145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 
  2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1 
169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 
  2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1   2   1 
193 194 195 196 197 198 199 200 
  2   1   2   1   2   1   2   1 
Within cluster sum of squares by cluster:
[1] 14901.85 19003.39 30376.45 41018.29
 (between_SS / total_SS =  65.9 %)
Available components:
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"   
[7] "size"         "iter"         "ifault"
```

#compute the mean of each variables by clusters

```
aggregate(df, by=list(cluster=km.res$cluster), mean)
```

```
cluster      Age Annual.Income..k.. Spending.Score..1.100.      Sex
1       1  32.87500         86.10000            81.52500         0.5500000
2       2 40.39474         87.00000            18.63158         0.4736842
3       3  25.05660         40.73585            62.62264         0.5849057
4     4    52.05797          46.42029              39.88406        0.5942029
```

#add the point classifications to the data

```
dd <- cbind(df, cluster = km.res$cluster)
head(dd,3)
```

```
Age Annual.Income..k.. Spending.Score..1.100. Sex cluster
1  19              15                39                     0     3
2  21              15                81                     0     3
3     20              16                  6                      1      4
```

#Cluster size

```
km.res$size
```

```
40 38 53 69
```

#Cluster means

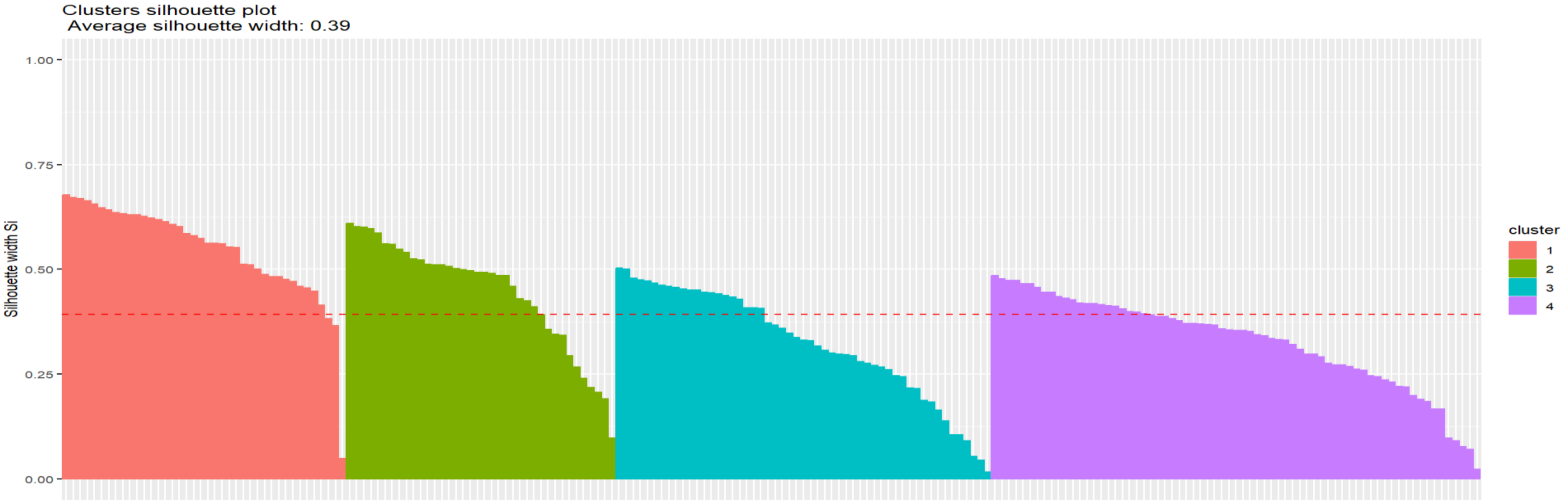
```
km.res$centers
```

```
      Age          Annual.Income..k..       Spending.Score..1.100.       Sex
1 32.87500         86.10000                    81.52500            0.5500000
2 40.39474         87.00000                   18.63158            0.4736842
3 25.05660         40.73585                   62.62264            0.5849057
4 52.05797         46.42029                   39.88406            0.5942029
```

Clustering Validation

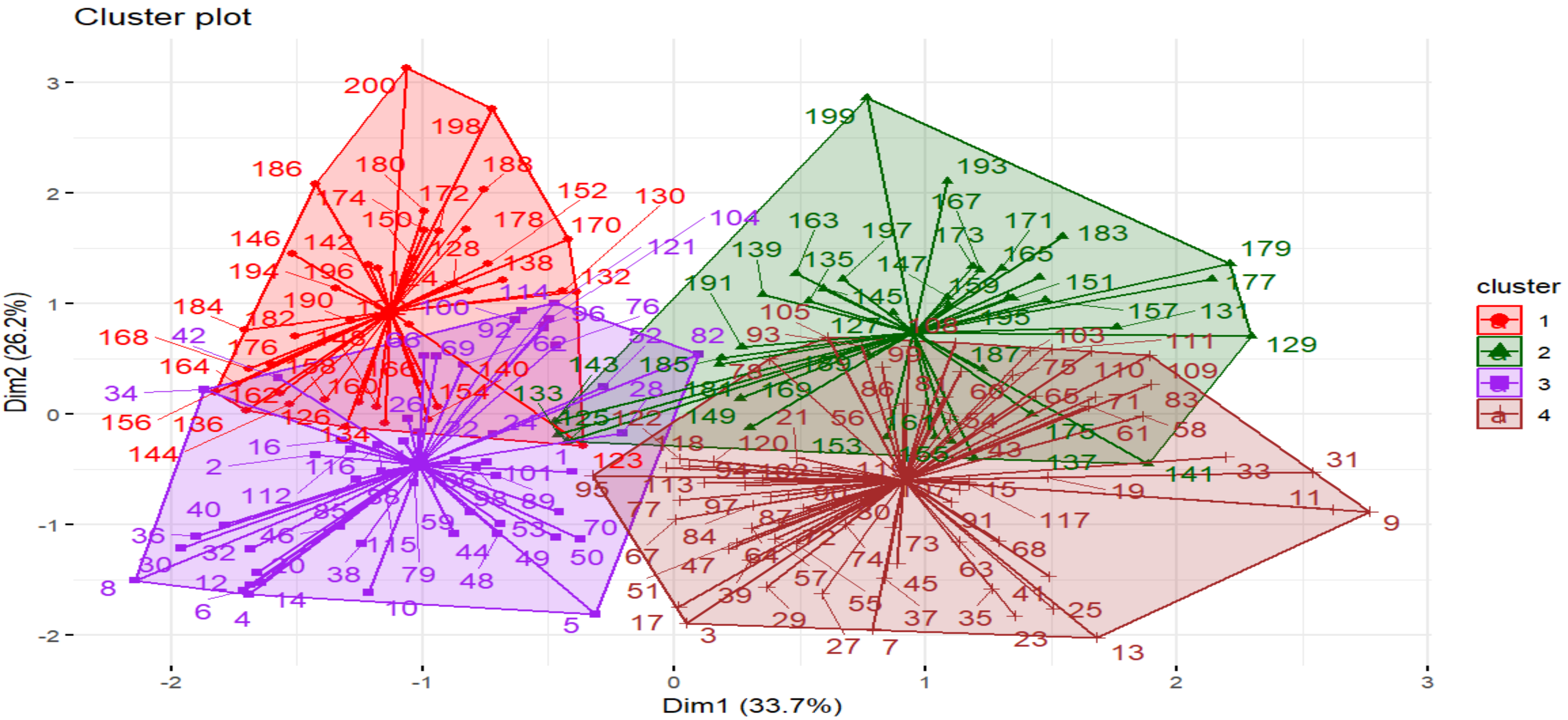
```
library(cluster)
sil <- silhouette(km.res$cluster, dist(df))
fviz_silhouette(sil)
```

	cluster	size	ave.sil.width
1	1	40	0.55
2	2	38	0.44
3	3	53	0.32
4	4	69	0.33



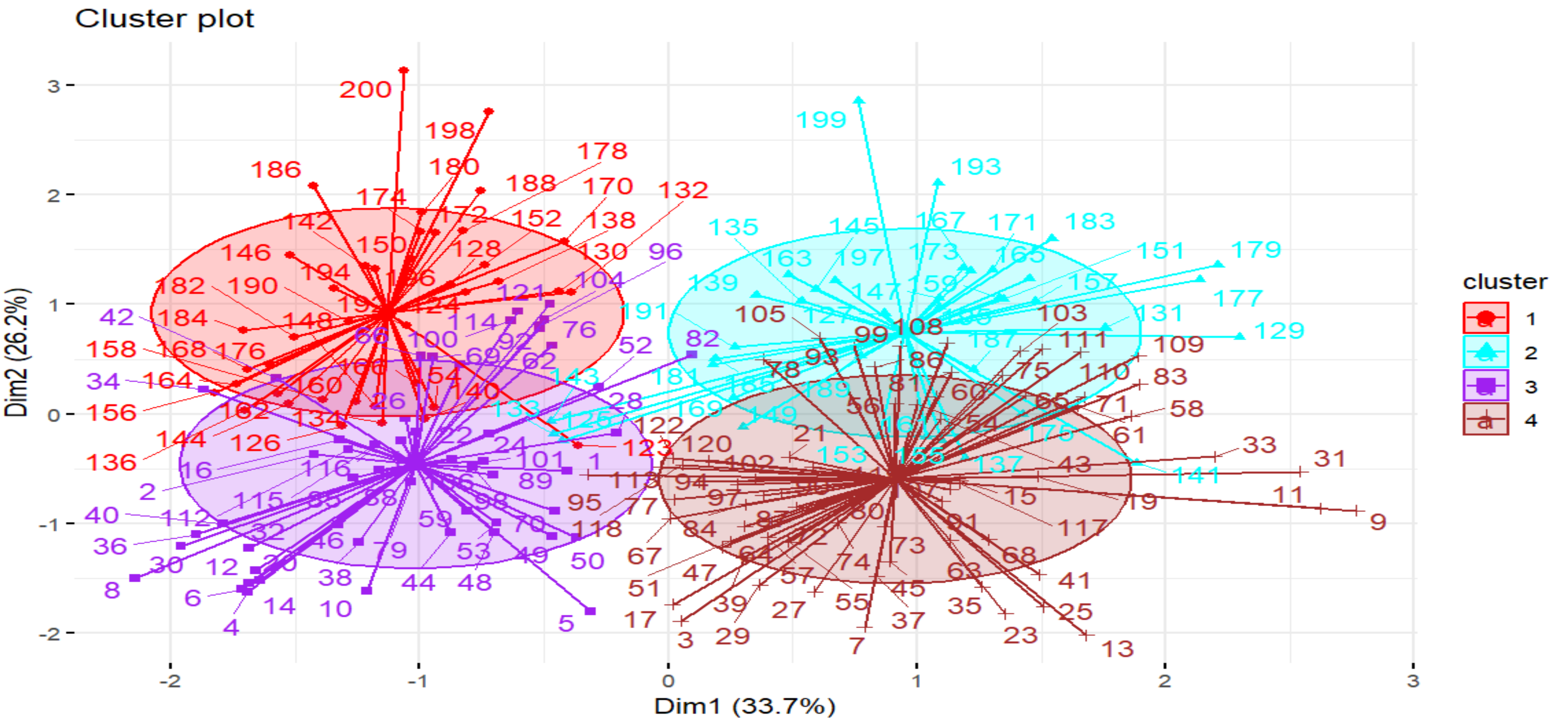
#Visualize cluster with ellipse type convex

fviz_cluster(km.res, data = df, palette = c("red", "darkgreen", "purple", "brown"), ellipse.type = "convex", # Concentration ellipse, other types: confidence,euclid star.plot = TRUE, # Add segments from centroids to items repel = TRUE, # Avoid label overplotting (slow) ggtheme = theme_minimal())



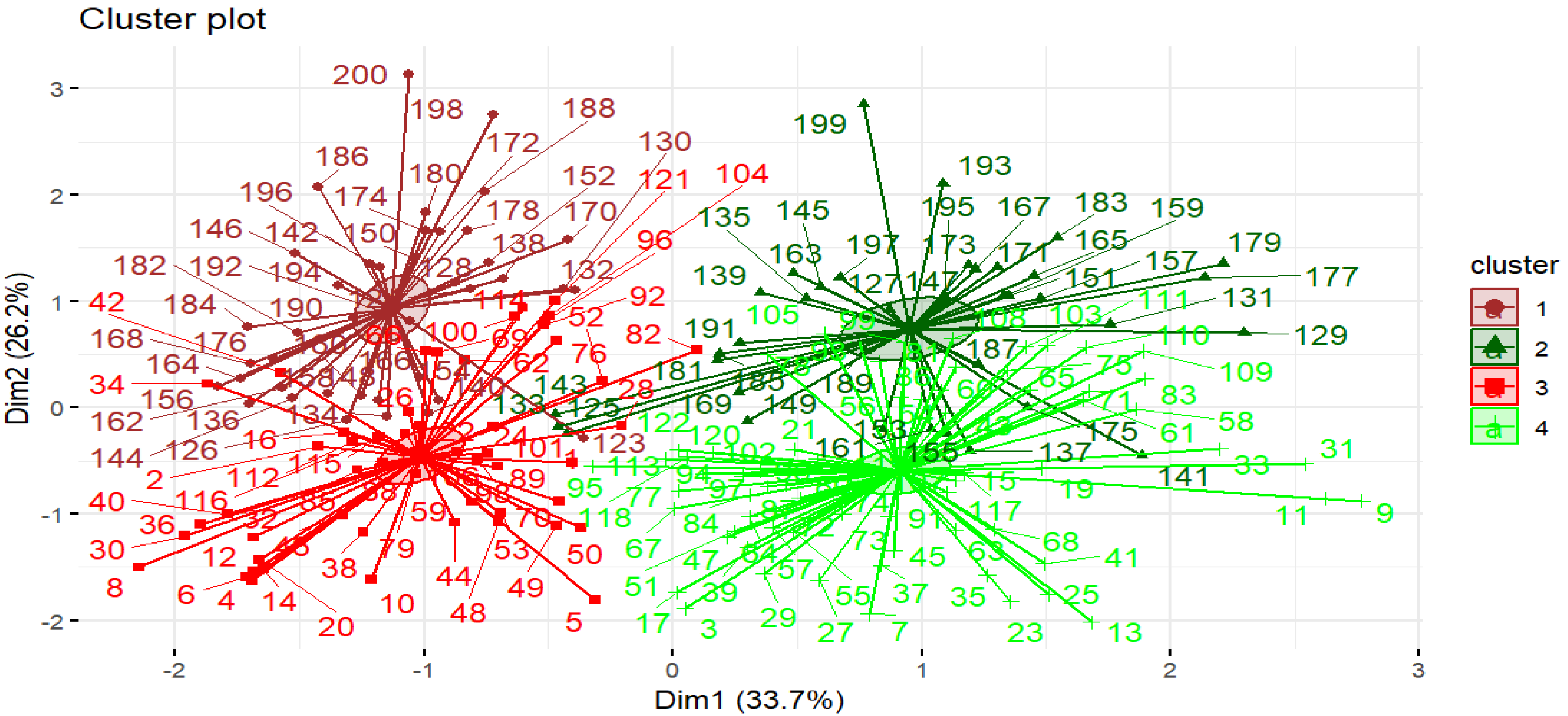
#Visualize cluster with ellipse type euclid

fviz_cluster(km.res, data = df, palette = c("red", "cyan", "purple", "brown"), ellipse.type = "euclid", # Concentration ellipse, other types: confidence, euclid star.plot = TRUE, # Add segments from centroids to items repel = TRUE, # Avoid label overplotting (slow) ggtheme = theme_minimal())



#Visualize cluster with ellipse type confidence

fviz_cluster(km.res, data = df, palette = c("brown", "darkgreen", "yellow", "cyan"), ellipse.type = "confidence", # Concentration ellipse, other types: confidence, euclid star.plot = TRUE, # Add segments from centroids to items repel = TRUE, # Avoid label overplotting (slow) ggtheme = theme_minimal())



#Compute k-means with k = 6

```
set.seed(12)
km.res1 <- kmeans(df, 6, nstart = 1)
```

Print the results

```
print(km.res1)
```

```
K-means clustering with 6 clusters of sizes 22, 38, 21, 35, 39, 45
Cluster means:
       Age Annual.Income..k.. Spending.Score..1.100.       Sex
1 25.27273           25.72727               79.36364 0.5909091
2 27.00000           56.65789               49.13158 0.6578947
3 44.14286           25.14286               19.52381 0.6190476
4 41.68571           88.22857               17.28571 0.4285714
5 32.69231           86.53846               82.12821 0.5384615
6 56.15556           53.37778               49.08889 0.5555556
Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 
  3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1 
 25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48 
  3   1   3   1   3   1   3   1   3   1   3   1   3   1   3   1   6   1   6   2   3   1   6   2 
 49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72 
  2   2   6   2   2   6   6   6   6   6   2   6   6   2   6   6   6   2   6   6   2   2   6   6 
 73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 
  6   6   6   2   6   2   2   6   6   2   6   6   2   6   6   2   2   6   6   2   6   2   2   2 
 97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 
  6   2   6   2   2   6   6   2   6   2   6   6   6   6   6   2   2   2   2   2   6   6   6   6 
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 
  2   2   2   5   2   5   4   5   4   5   4   5   2   5   4   5   4   5   4   5   4   5   2   5 
145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 
  4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5 
169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 
  4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5   4   5 
193 194 195 196 197 198 199 200 
  4   5   4   5   4   5   4   5 
Within cluster sum of squares by cluster:
[1]  4105.136  7751.447  7737.333 16699.429 13982.051  8073.244
 (between_SS / total_SS =  81.1 %)
Available components:
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"   
[7] "size"         "iter"         "ifault"
```

#compute the mean of each variables by clusters

```
aggregate(df, by=list(cluster=km.res1$cluster), mean)
```

```
cluster      Age Annual.Income..k.. Spending.Score..1.100.       Sex
1       1 25.27273          25.72727           79.36364        0.5909091
2       2 27.00000          56.65789           49.13158        0.6578947
3       3 44.14286          25.14286           19.52381        0.6190476
4       4 41.68571          88.22857           17.28571        0.4285714
5       5 32.69231          86.53846           82.12821        0.5384615
6       6 56.15556          53.37778           49.08889        0.5555556
```

#add the point classifications to the data

```
dd2 <- cbind(df, cluster = km.res1$cluster)
head(dd2,3)
```

```
Age Annual.Income..k.. Spending.Score..1.100. Sex cluster
1  19              15                39                    0      3
2  21              15                81                    0      1
3     20              16                 6                    1      3
```

#Cluster size

```
km.res1$size
```

```
22 38 21 35 39 45
```

#Cluster means

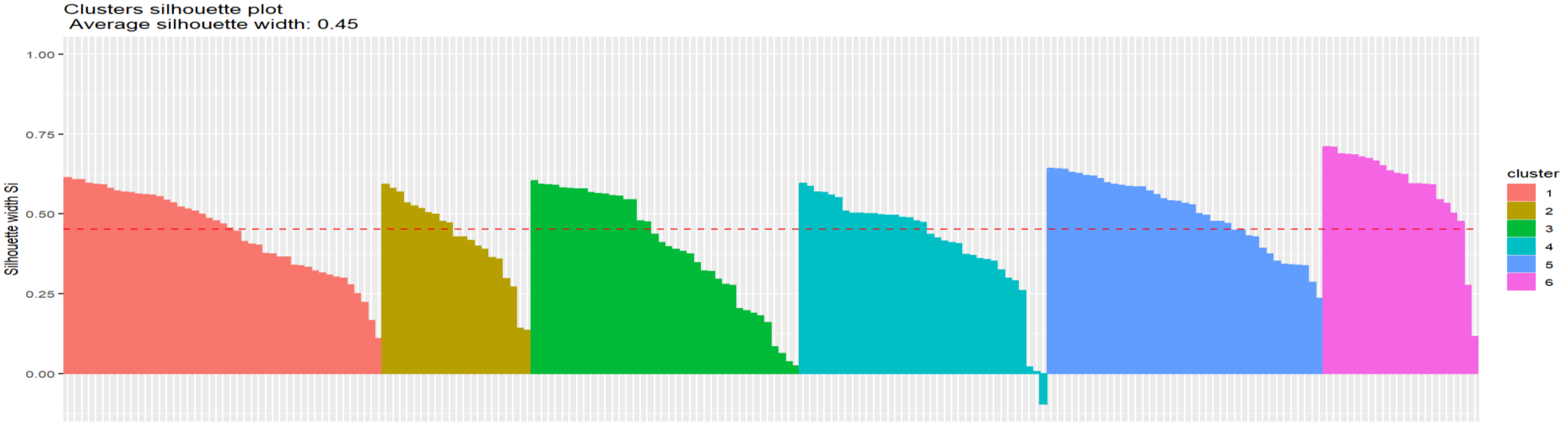
```
km.res1$centers
```

```
      Age Annual.Income..k.. Spending.Score..1.100.       Sex
1 25.27273          25.72727           79.36364        0.5909091
2 27.00000          56.65789           49.13158        0.6578947
3 44.14286          25.14286           19.52381        0.6190476
4 41.68571          88.22857           17.28571        0.4285714
5 32.69231          86.53846           82.12821        0.5384615
6 56.15556          53.37778           49.08889        0.5555556
```

Clustering Validation

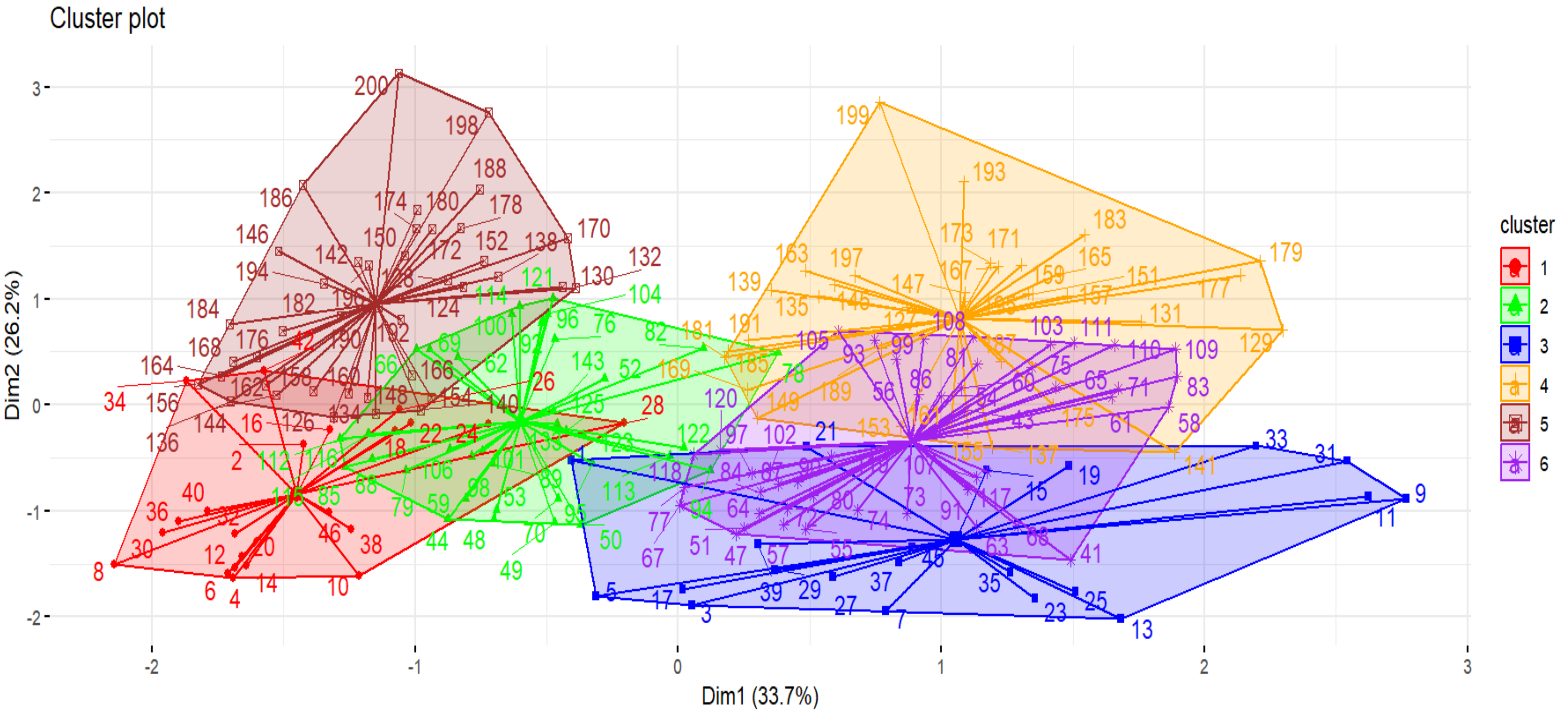
```
library(cluster)
sil2 <- silhouette(km.res1$cluster, dist(df))
fviz_silhouette(sil2)
```

```
 cluster size ave.sil.width
1       1   45          0.44
2       2   21          0.42
3       3   38          0.39
4       4   35          0.41
5       5   39          0.50
6       6   22          0.58
```



#Visualize cluster with ellipse type convex

fviz_cluster(km.res, data = df, palette = c(“red”, “darkgreen”, “purple”, “brown”), ellipse.type = “convex”, # Concentration ellipse, other types: confidence,euclid star.plot = TRUE, # Add segments from centroids to items repel = TRUE, # Avoid label overplotting (slow) ggtheme = theme_minimal())



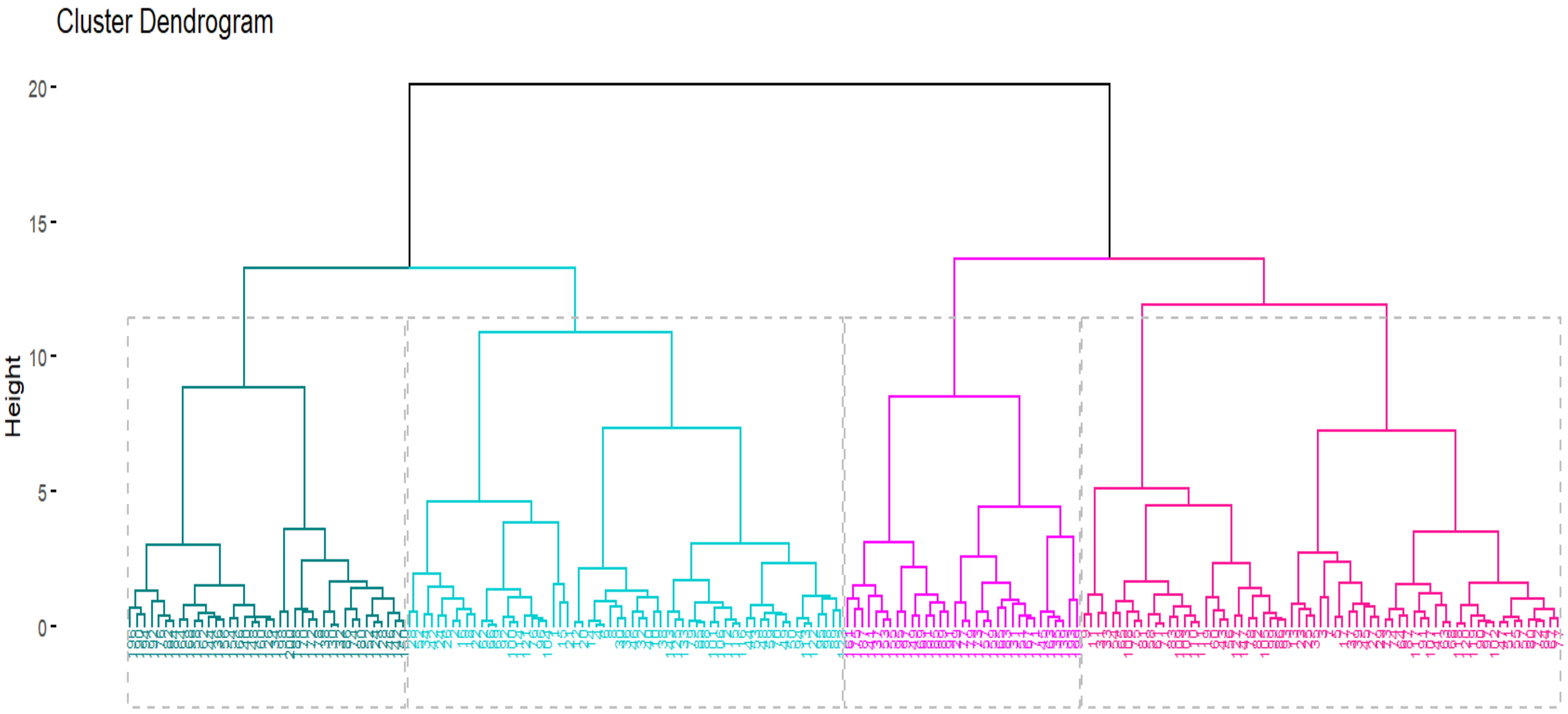
Hierarchical Clustering Analysis

Hierarchical clustering in R Programming Language is an Unsupervised non-linear algorithm in which clusters are created such that they have a hierarchy(or a pre-determined ordering).

```
#Compute with k = 4
hc.res <- hcut(df, k = 4, stand = TRUE)
hc.res
Cluster method   : ward.D2
Distance         : euclidean
Number of objects: 200
```

#Plot for Cluster Dendrogram

```
library("dendextend")
fviz_dend(hc.res, rect = TRUE, cex = 0.5,  k_colors = c("#008080","#00CED1", "#FF00FF", "#FF1493"))
```



#compute the mean of each variables by clusters

```
aggregate(df, by=list(cluster=hc.res$cluster), mean)
cluster      Age Annual.Income..k.. Spending.Score..1.100.       Sex
1       1 26.14754          43.77049             58.96721          0.6229508
2       2 52.71642          46.67164             40.38806          0.5671642
3       3 32.69231          86.53846             82.12821          0.5384615
4       4 41.45455          89.09091             16.18182          0.4545455
```

#add the point classifications to the data

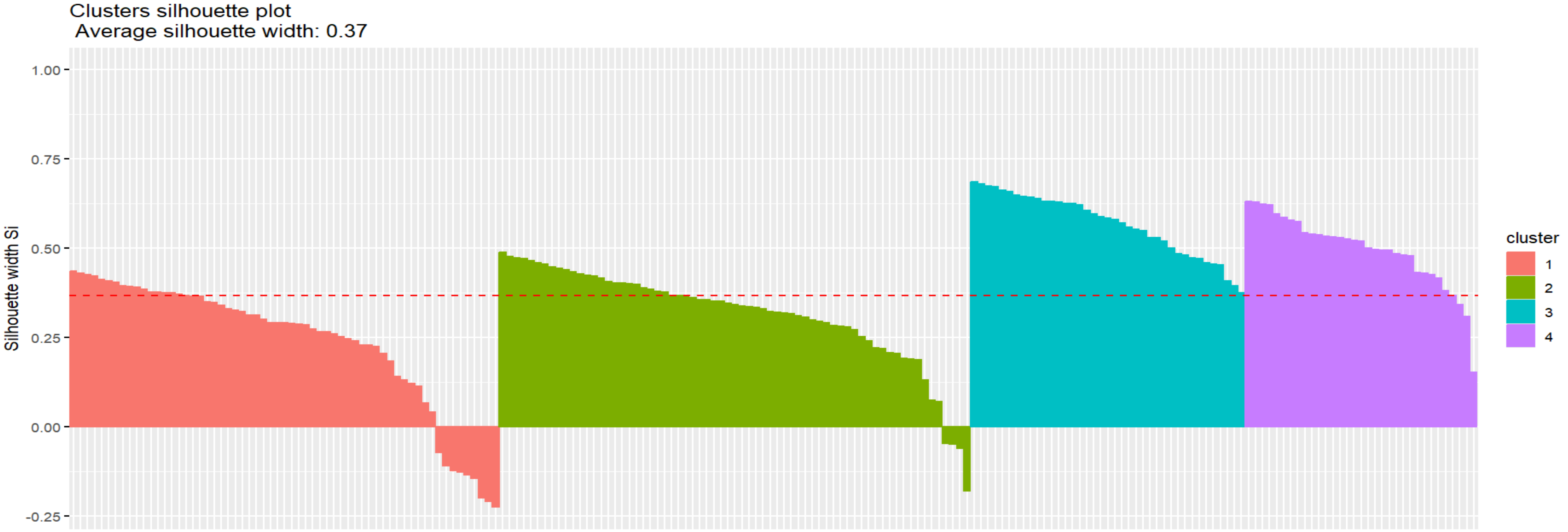
```
dd3 <- cbind(df, cluster = km.res$cluster)
head(dd3,3)
 Age Annual.Income..k.. Spending.Score..1.100.    Sex   cluster
1  19          15                39                    0     1
2  21          15                81                    0     1
3  20          16                 6                    1     2
```

#Cluster size

```
hc.res$size
61 67 39 33
```

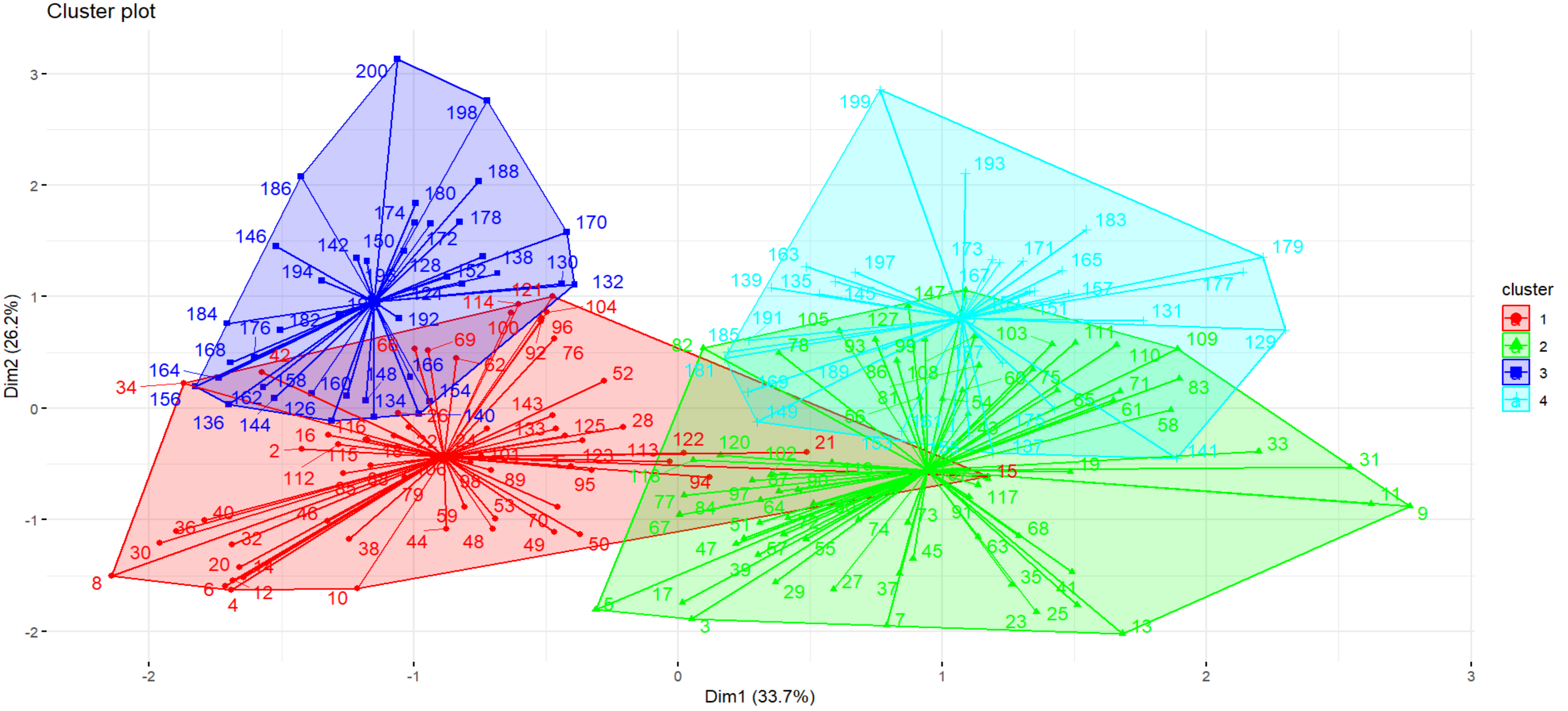
Clustering Validation

```
sil3 <- silhouette(hc.res$cluster, dist(df))
fviz_silhouette(sil3)
 cluster size ave.sil.width
1       1   61          0.23
2       2   67          0.31
3       3   39          0.57
4       4   33          0.49
```



#Visualize cluster with ellipse type convex

fviz_cluster(hc.res, data = df, palette = c(“red”, “darkgreen”, “purple”, “brown”), ellipse.type = “convex”, # Concentration ellipse, other types: confidence,euclid star.plot = TRUE, # Add segments from centroids to items repel = TRUE, # Avoid label overplotting (slow) ggtheme = theme_minimal())



Inference based on the Analysis:

- Optimal number of clusters were obtained from Elbow method, Silhouette method and Gap Statistic method.
- Interpretation of the silhouette width:
  - ✓ Si > 0 means that the observation is well clustered. The closest it is to 1, the best it is clustered.
  - ✓ Si < 0 means that the observation was placed in the wrong cluster.
  - ✓ Si = 0 means that the observation is between two clusters.
- The silhouette plot gives evidence that clustering using four and six groups is good because there's no negative silhouette width and some of the values are bigger than 0.5.

Insights from K Means Clustering:

- Using 4 groups (K = 4) it had 65.9 % of well-grouped data. Using 6 groups (K = 6) that value raised to 81.1 %, which is a good value.
- Using four group of clusters,
  - Females with Annual income more than 80K and Age 30-40 had spending score more than 75.
  - Females with age 20-30 had annual income of 40K had spending score around 60-65.
  - Age of Females more than 50 with annual income above 40K had spending score around 40.
  - Males with age more than 40 with annual income more than 85K had spending score below 20.
- Using six group of clusters,
  - Females with Age 30-40 and Annual income more than 80K had spending score more than 80.
  - Females with age 20-25 with annual income of 20-30K had spending score more than 75.
  - Females with age 25-30 with annual income of above 50K had spending score around 50.
  - Age of Females more than 50 with annual income above 50K had spending score around 50.
  - Females with Age 40-50 and Annual income around 25K had less spending score around 20.
  - Males with age more than 40 with annual income more than 85 K had very less spending score around 15.

Insights from Hierarchical Clustering:

- Females with Age 30-40 and Annual income more than 85K had spending score more than 80.
- Females with age 20-30 had annual income of 40K had spending score around 55.
- Females with Age more than 50 with annual income above 45K had spending score around 40.
- Males with age more than 40 with annual income more than 85K had spending score below 20.

From the data, it is evident that Females with Age of 30-40 with Annual income more than 80K were shopping higher. Females with age of 20-30 were also interested in shopping and spend more. Females with older age showed less interest compared to other females. Males had less interest in shopping although they had higher annual income and spend less.

Hence Females at the age 20-40 showed more interest in shopping and had good spending score than males.