Customer segmentation project in R

1. Introduction

Customers segmentation is one of the applications of unsupervized machine learning algorithms. Its used by companies to identify groups of customers that share similarities in different ways.
In this project, we will analyze a dataset containing data on various customers' annual spendings in monetary units (m.u.) of diverse product categories. Our objective is to best describe the variation in the different types of customers of a wholesale distributor, which will help them know how to best structure their delivery service to meet the needs of each customer.

The dataset for this project is found on the UCI Machine Learning Repository. The dataset has 440 observations and 8 variables.

```
## Libraries
library(plotrix)
library(purrr)
library(cluster)
library(gridExtra)
library(grid)
library(NbClust)
library(factoextra)
library(ggplot2)
```

```
## Reading data
df<-read.csv("Wholesale customers data.csv")
str(df)
```

```
## 'data.frame':    440 obs. of  8 variables:
##  $ Channel         : int  2 2 2 1 2 2 2 2 1 2 ...
##  $ Region          : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Fresh           : int  12669 7057 6353 13265 22615 9413 12126 7579 5963 6006 ...
##  $ Milk            : int  9656 9810 8808 1196 5410 8259 3199 4956 3648 11093 ...
##  $ Grocery         : int  7561 9568 7684 4221 7198 5126 6975 9426 6192 18881 ...
##  $ Frozen          : int  214 1762 2405 6404 3915 666 480 1669 425 1159 ...
##  $ Detergents_Paper: int  2674 3293 3516 507 1777 1795 3140 3321 1716 7425 ...
##  $ Delicassen      : int  1338 1776 7844 1788 5185 1451 545 2566 750 2098 ...
```

```
## Covert Channel and region variables to Factors
df$Channel<-as.factor(df$Channel)
df$Region<-as.factor(df$Region)
```

List of variables in the dataset:

1) FRESH: annual spending (m.u.) on fresh products (Continuous);

2) MILK: annual spending (m.u.) on milk products (Continuous);

3) GROCERY: annual spending (m.u.) on grocery products (Continuous);

4) FROZEN: annual spending (m.u.) on frozen products (Continuous);

5) DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Continuous);

6) DELICATESSEN: annual spending (m.u.) on and delicatessen products (Continuous);

7) CHANNEL: customers' Channel - Horeca (Hotel/Restaurant/Café) or Retail channel (Nominal);

8) REGION: customers' Region - Lisbon, Oporto or Other (Nominal)

2. Data exploration

```
head(df)
```

```
##   Channel Region Fresh Milk Grocery Frozen Detergents_Paper Delicassen
## 1       2      3 12669 9656    7561    214             2674       1338
## 2       2      3  7057 9810    9568   1762             3293       1776
## 3       2      3  6353 8808    7684   2405             3516       7844
## 4       1      3 13265 1196    4221   6404              507       1788
## 5       2      3 22615 5410    7198   3915             1777       5185
## 6       2      3  9413 8259    5126    666             1795       1451
```

```
## check variables statistical summaries and stadard deviation
summary(df)
```

```
##  Channel Region      Fresh             Milk          Grocery
##  1:298   1: 77   Min.   :     3   Min.   :   55   Min.   :    3
##  2:142   2: 47   1st Qu.:  3128   1st Qu.: 1533   1st Qu.: 2153
##          3:316   Median :  8504   Median : 3627   Median : 4756
##                  Mean   : 12000   Mean   : 5796   Mean   : 7951
##                  3rd Qu.: 16934   3rd Qu.: 7190   3rd Qu.:10656
##                  Max.   :112151   Max.   :73498   Max.   :92780
##      Frozen        Detergents_Paper    Delicassen
##  Min.   :   25.0   Min.   :    3.0   Min.   :    3.0
##  1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2
##  Median : 1526.0   Median :  816.5   Median :  965.5
##  Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9
##  3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2
##  Max.   :60869.0   Max.   :40827.0   Max.   :47943.0
```

```
sd(df$Fresh)
```

```
## [1] 12647.33
```

```
sd(df$Milk)
```

```
## [1] 7380.377
```

```r
sd(df$Grocery)
```

```
## [1] 9503.163
```

```r
sd(df$Frozen)
```

```
## [1] 4854.673
```

```r
sd(df$Detergents_Paper)
```

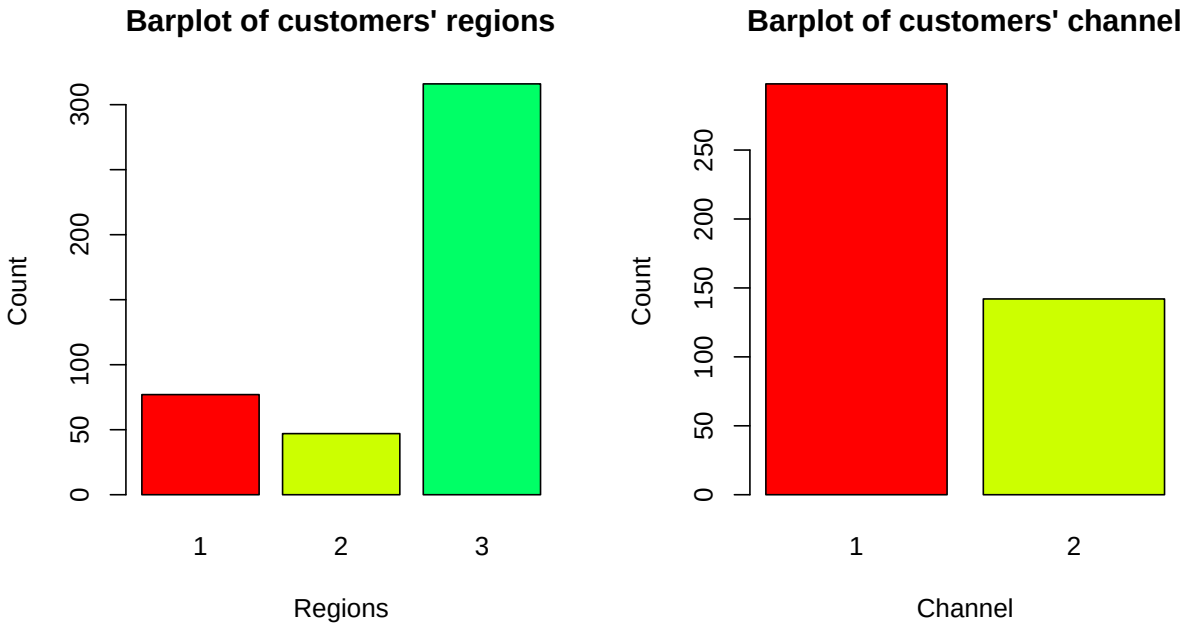
```
## [1] 4767.854
```

```r
sd(df$Delicassen)
```

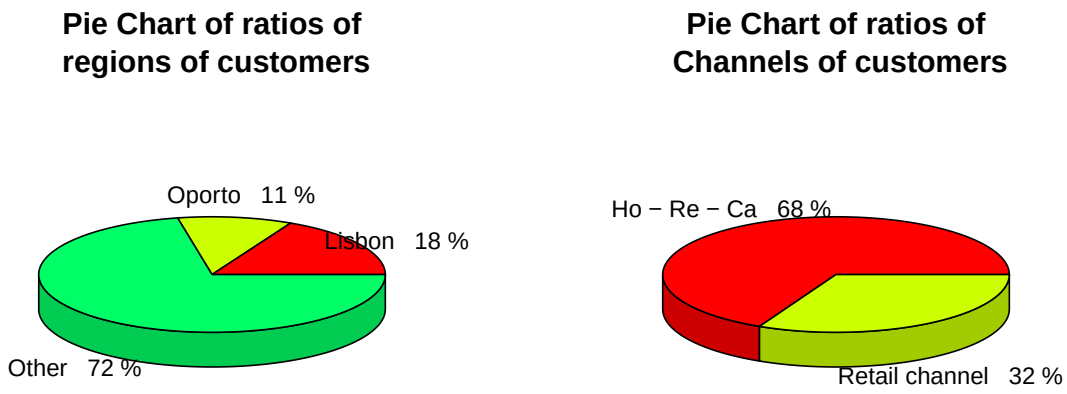
```
## [1] 2820.106
```

Visualizations of Region and Channel variables

```r
par(mfrow = c(1, 2))
barplot(table(df$Region), main = "Barplot of customers' regions",
        ylab = "Count", xlab = "Regions",
        col=rainbow(5))
barplot(table(df$Channel), main = "Barplot of customers' channel",
        ylab = "Count", xlab = "Channel",
        col=rainbow(5))
```



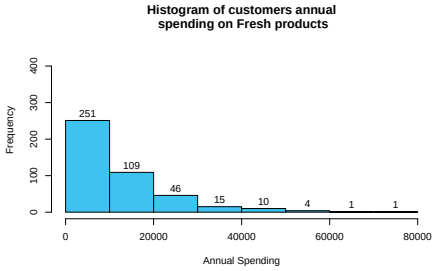
```r
p=round(table(df$Region)/sum(table(df$Region))*100)
labs=paste(c("Lisbon","Oporto", "Other")," ",p,"%",sep=" ")

pc=round(table(df$Channel)/sum(table(df$Channel))*100)
lab=paste(c("Ho - Re - Ca","Retail channel")," ",pc,"%",sep=" ")

par(mfrow = c(1, 2))
pie3D(table(df$Region), labels = labs,labelcex=0.9, col=rainbow(5),
   main="Pie Chart of ratios of\n regions of customers")
pie3D(table(df$Channel), labels = lab,labelcex=0.9, col=rainbow(5),
   main="Pie Chart of ratios of\n Channels of customers")
```

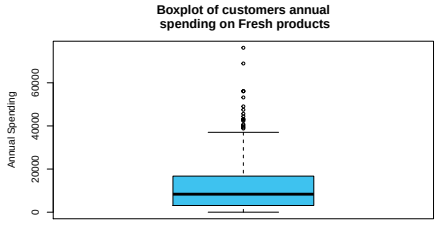


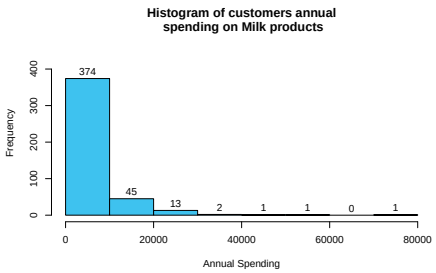
From the graphs above we notice that 18% of customers are from Lisbon and 11% from Oporto. We also conclude that 32% of customers are from a retail channel and the rest from hotels, restaurants or cafés.

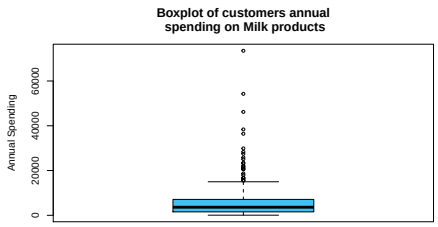
Visualizations of numerical variables

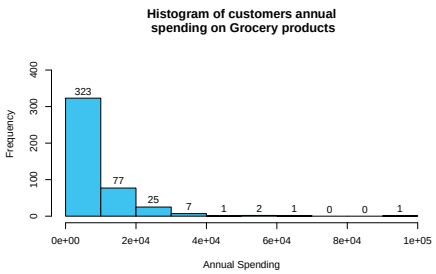
```r
# eliminate outliers
df<-df[which(df$Delicassen<40000),]
df<-df[which(df$Fresh<100000),]
df<-df[which(df$Frozen<50000),]

par(mfrow = c(6, 2))
for(i in 3:8){
  x<-paste0(names(df)[i], " products")
  hist(df[,i], main = paste0("Histogram of customers annual\n spending on ", x),
       xlab = "Annual Spending", labels = TRUE, ylim = c(0,450), col = "#3EC2EF")
  boxplot(df[,i], main = paste0("Boxplot of customers annual\n spending on ", x),
          ylab = "Annual Spending", labels = TRUE, col = "#3EC2EF")}
```

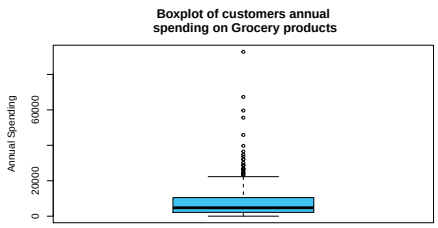

Histogram of customers annual spending on Fresh products
Frequency
Annual Spending
251
109
46
15
10
4
1
1
0
20000
40000
60000
80000

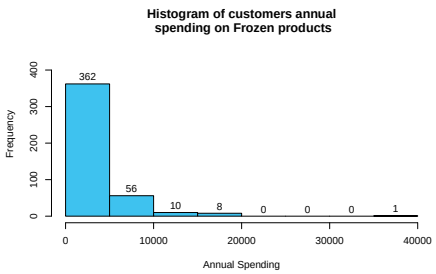

Boxplot of customers annual spending on Fresh products
Annual Spending
0
20000
40000
60000

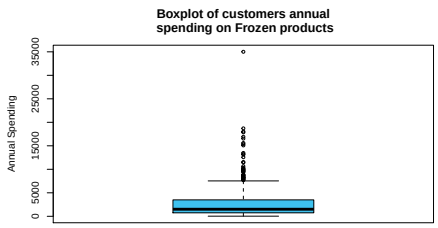

Histogram of customers annual spending on Milk products
Frequency
Annual Spending
374
45
13
2
1
1
0
1
0
20000
40000
60000
80000

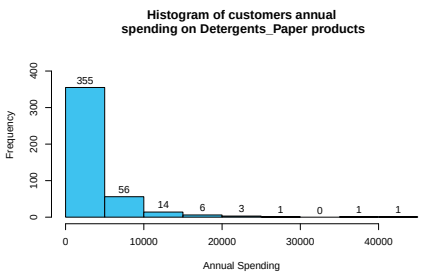

Boxplot of customers annual spending on Milk products
Annual Spending
0
20000
40000
60000

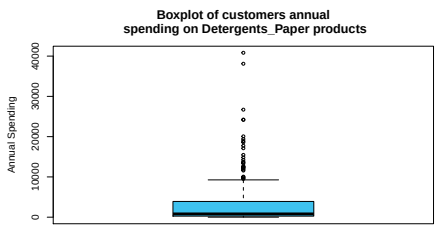

Histogram of customers annual spending on Grocery products
Frequency
Annual Spending
323
77
25
7
1
2
1
0
0
1
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05

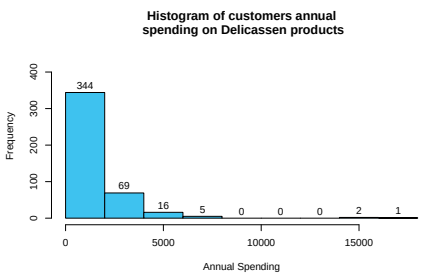

Boxplot of customers annual spending on Grocery products
Annual Spending
0
20000
40000
60000

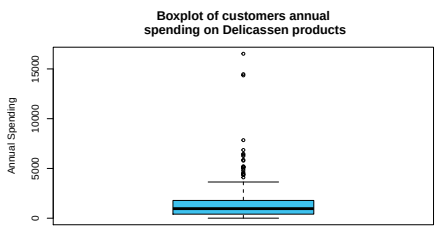

Histogram of customers annual spending on Frozen products
Frequency
Annual Spending
362
56
10
8
0
0
0
1
0
10000
20000
30000
40000


Boxplot of customers annual spending on Frozen products
Annual Spending
0
5000
15000
25000
35000


Histogram of customers annual spending on Detergents_Paper products
Frequency
Annual Spending
355
56
14
6
3
1
0
1
1
0
10000
20000
30000
40000


Boxplot of customers annual spending on Detergents_Paper products
Annual Spending
0
10000
20000
30000
40000


Histogram of customers annual spending on Delicassen products
Frequency
Annual Spending
344
69
16
5
0
0
0
2
1
0
5000
10000
15000


Boxplot of customers annual spending on Delicassen products
Annual Spending
0
5000
10000
15000

- The minimum annual spending on fresh products is 3, maximum is 76237 and the average is 11667. The histogram is right skewed, we conclude from it that most customers spend less than 20000 of Fresh products.
- The minimum annual spending on Milk products is 55, maximum is 73498 and the average is 5629. The histogram is right skewed, we conclude from it that most customers spend less than 20000 of Milk products.
- The minimum annual spending on Grocery products is 3, maximum is 92780 and the average is 7887. The histogram is right skewed, we conclude from it that most customers spend less than 2000 of Grocery products.
- The minimum annual spending on Frozen products is 25, maximum is 35009 and the average is 2832. The histogram is right skewed, we conclude from it that most customers spend less than 10000 of Frozen products.
- The minimum annual spending on Detergents_Paper products is 3, maximum is 40827 and the average is 2886 The histogram is right skewed, we conclude from it that most customers spend less than 10000 of Detergents_Paper products.
- The minimum annual spending on Delicassen products is 3, maximum is 16523 and the average is 1393 The histogram is right skewed, we conclude from it that most customers spend less than 10000 of Delicassen products.

3. K-means Algorithm

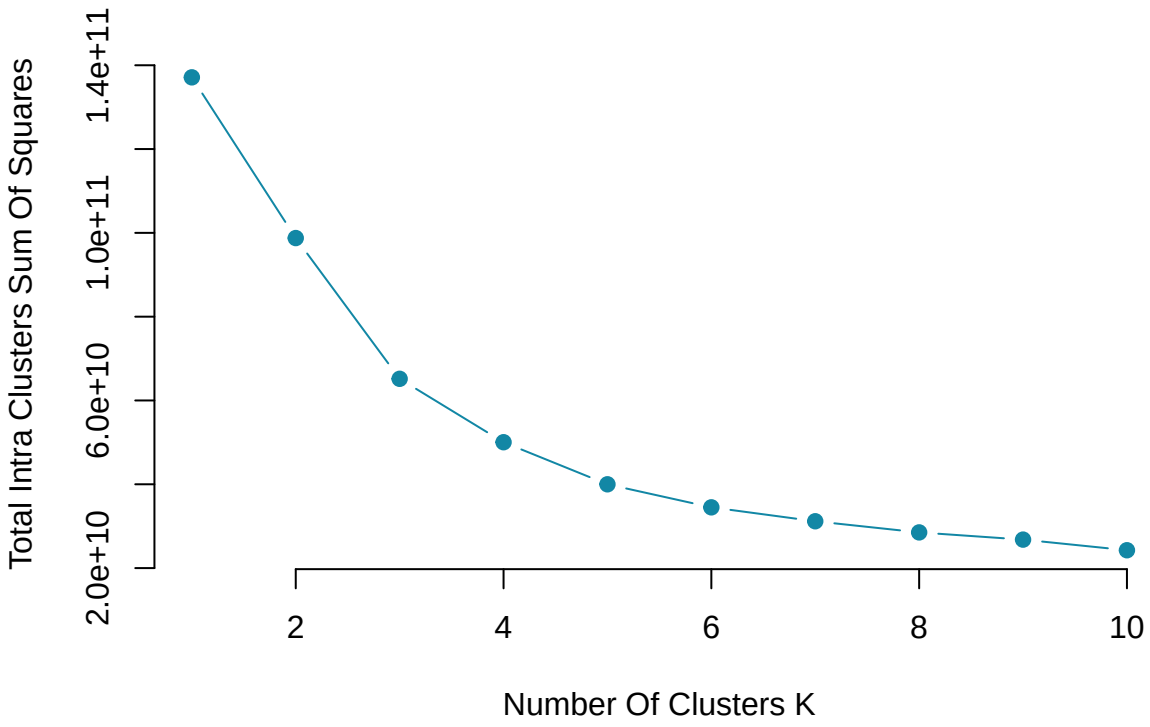
The first step in using the K-means clustering algorithm is to indicate the number of clusters (k) that we wish to produce in the final output. To do so, there three methods we can use: * Elbow method

- Silhouette method
- Gap statistic

Elbow graph method

In the Elbow graph method, we start by calculating the the Clustering Algorithm for values of k from 1 to 10, then we calculate the total ISS (intra-cluster sum of square). Finally, we plot the iss according to the number of clusters which makes the Elbow plot.

```
set.seed(123)
iss <- function(k){
  kmeans(df[,3:8],k,iter.max = 100,nstart = 100,algorithm = "Lloyd")$tot.withinss
}

k.values <- 1:10

iss_values <- map_dbl(k.values,iss)
plot(k.values,iss_values,type = "b",pch=19,frame=FALSE,xlab = "Number Of Clusters K",
     ylab = "Total Intra Clusters Sum Of Squares",col="#1287A5")
```

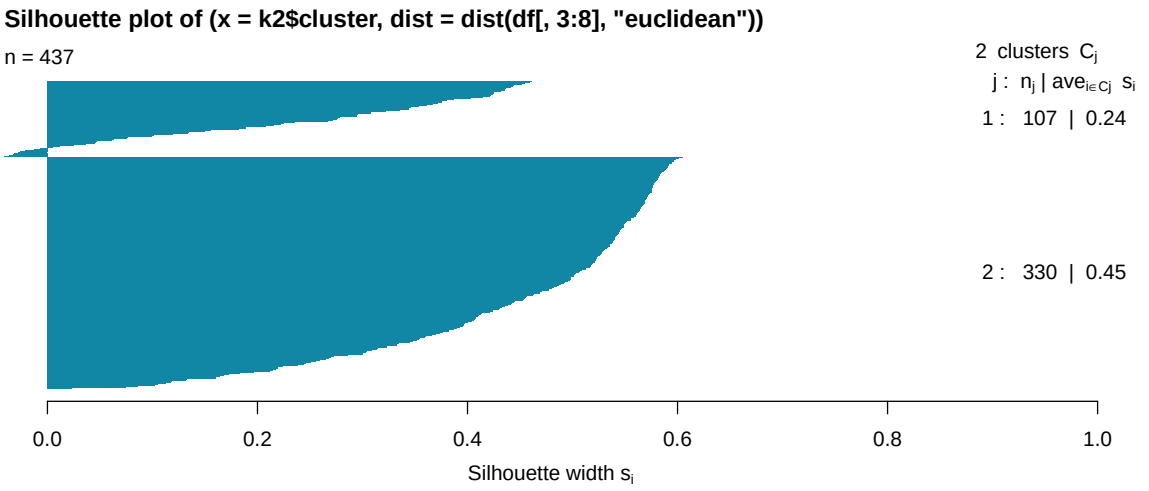


In an Elbow graph, the optimal number of clusters is the one appearing at the bend in the plot, in our case that number is **4**, as shown in the graph above.

Silhouette method

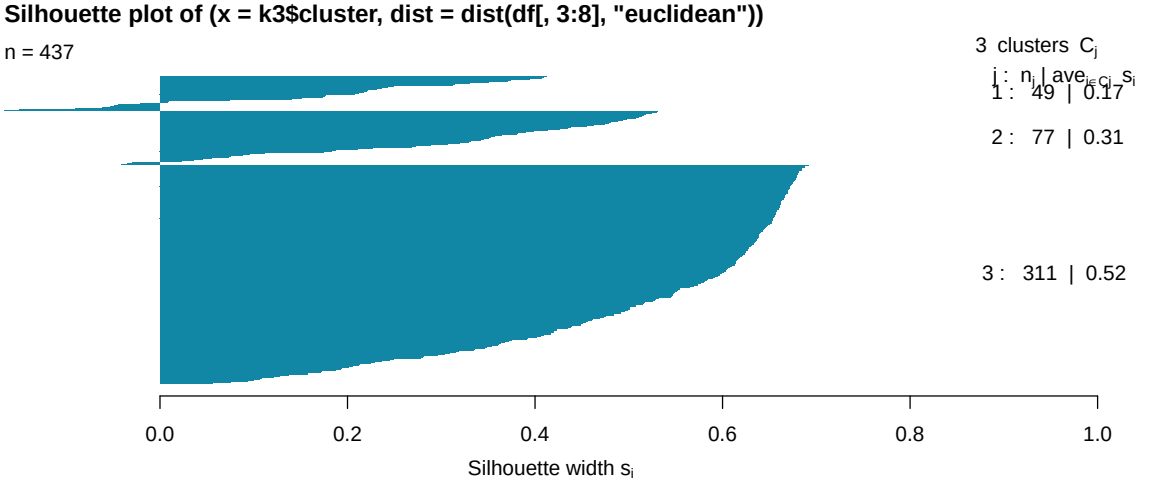
The average silhouette method helps determin the optimal number of clusters k by maximizing the average silhouette width. It does so by calculating the mean of silhouette observations for different k values.

```
k2 <- kmeans(df[,3:8],2,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s2 <- plot(silhouette(k2$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```



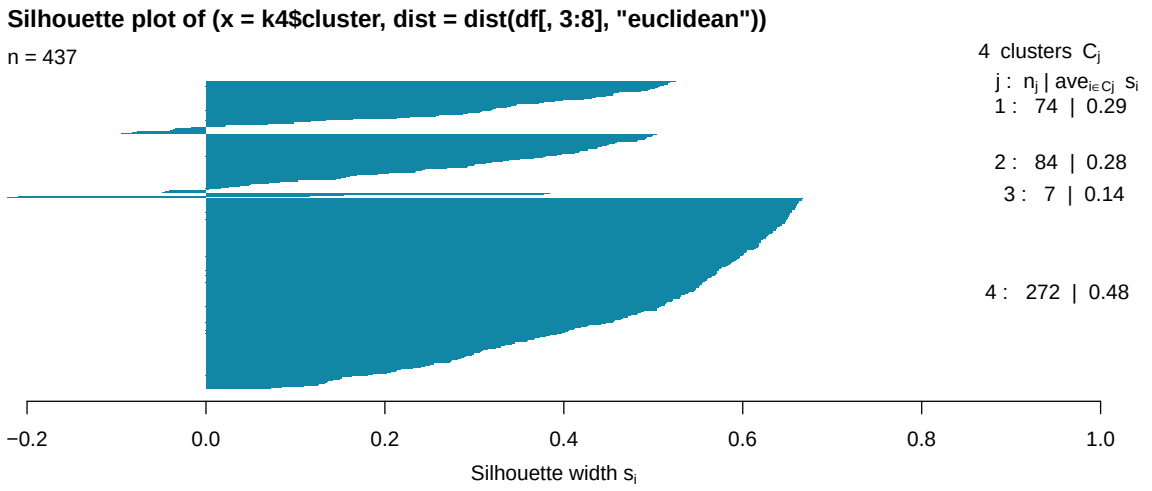
Average silhouette width : 0.4

```
k3 <- kmeans(df[,3:8],3,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s3 <- plot(silhouette(k3$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```



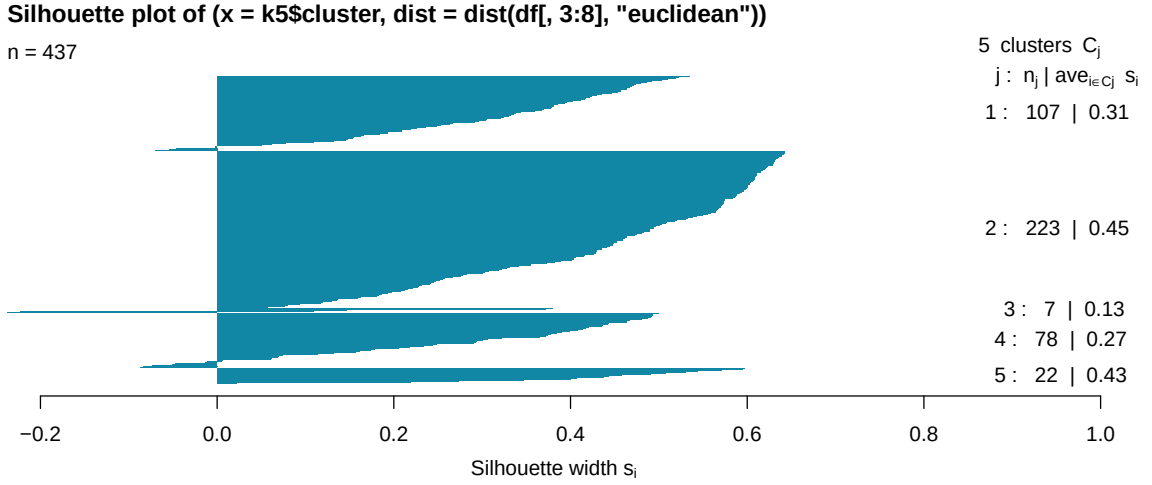
Average silhouette width : 0.45

```
k4 <- kmeans(df[,3:8],4,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s4 <- plot(silhouette(k4$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```



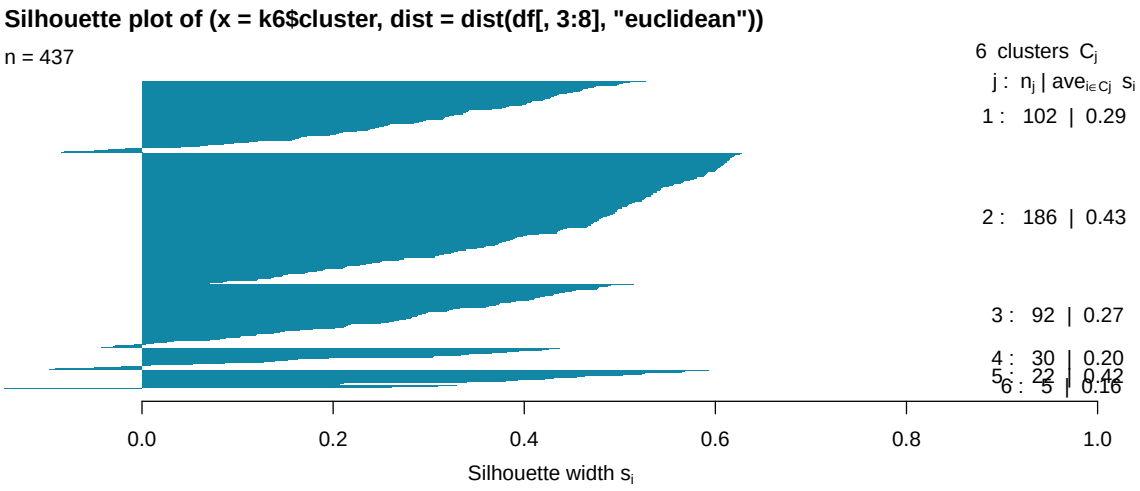
Average silhouette width : 0.4

```
k5 <- kmeans(df[,3:8],5,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s5 <- plot(silhouette(k5$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```



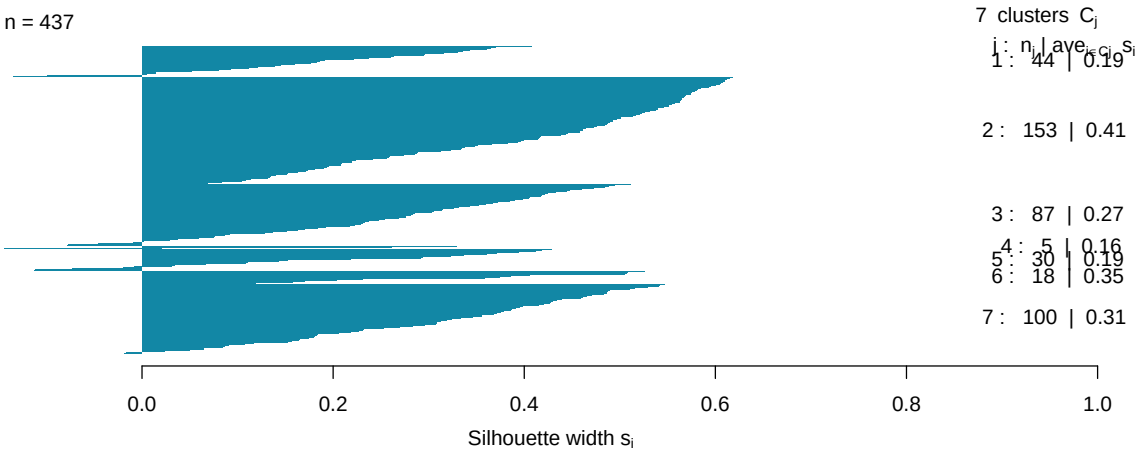
Average silhouette width : 0.38

```
k6 <- kmeans(df[,3:8],6,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s6 <- plot(silhouette(k6$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```



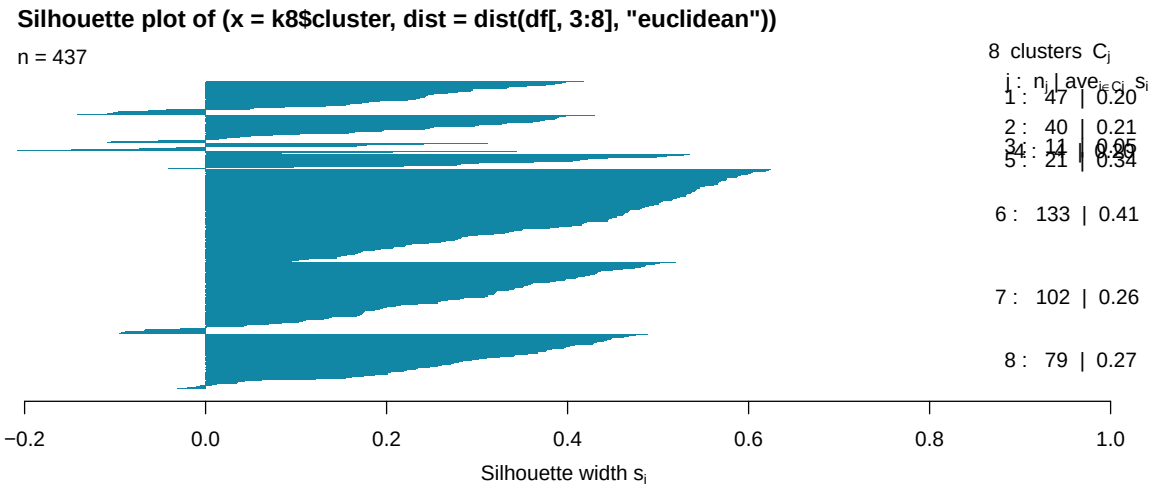
Average silhouette width : 0.35

```
k7 <- kmeans(df[,3:8],7,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s7 <- plot(silhouette(k7$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```

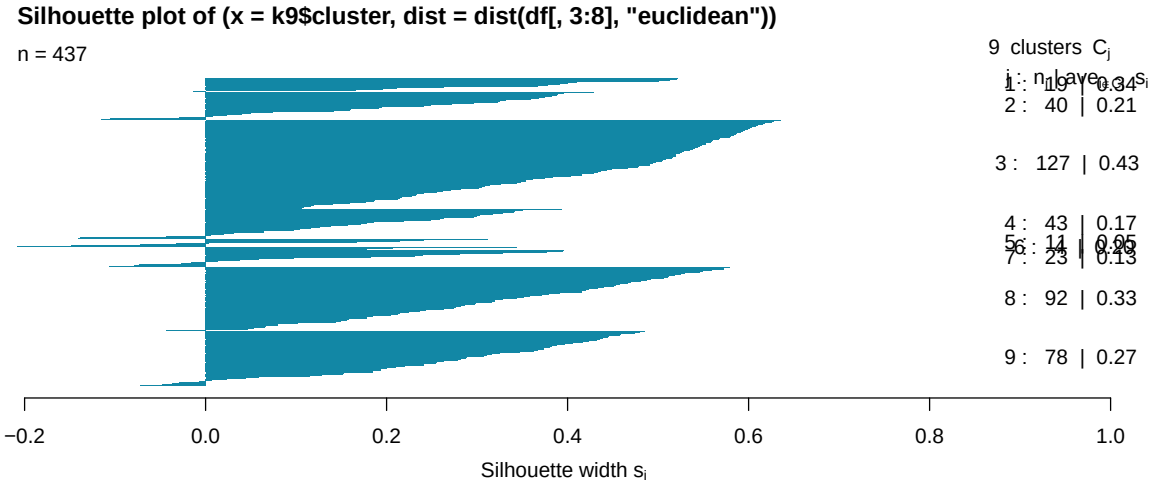


Average silhouette width : 0.31

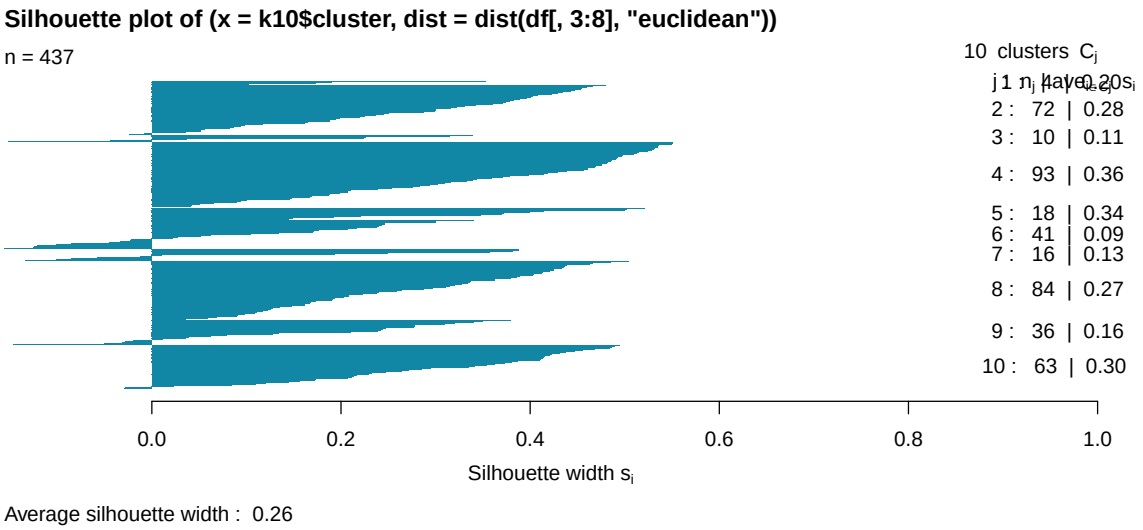
```
k8 <- kmeans(df[,3:8],8,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s8 <- plot(silhouette(k8$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```




```
k9 <- kmeans(df[,3:8],9,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s9 <- plot(silhouette(k9$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```

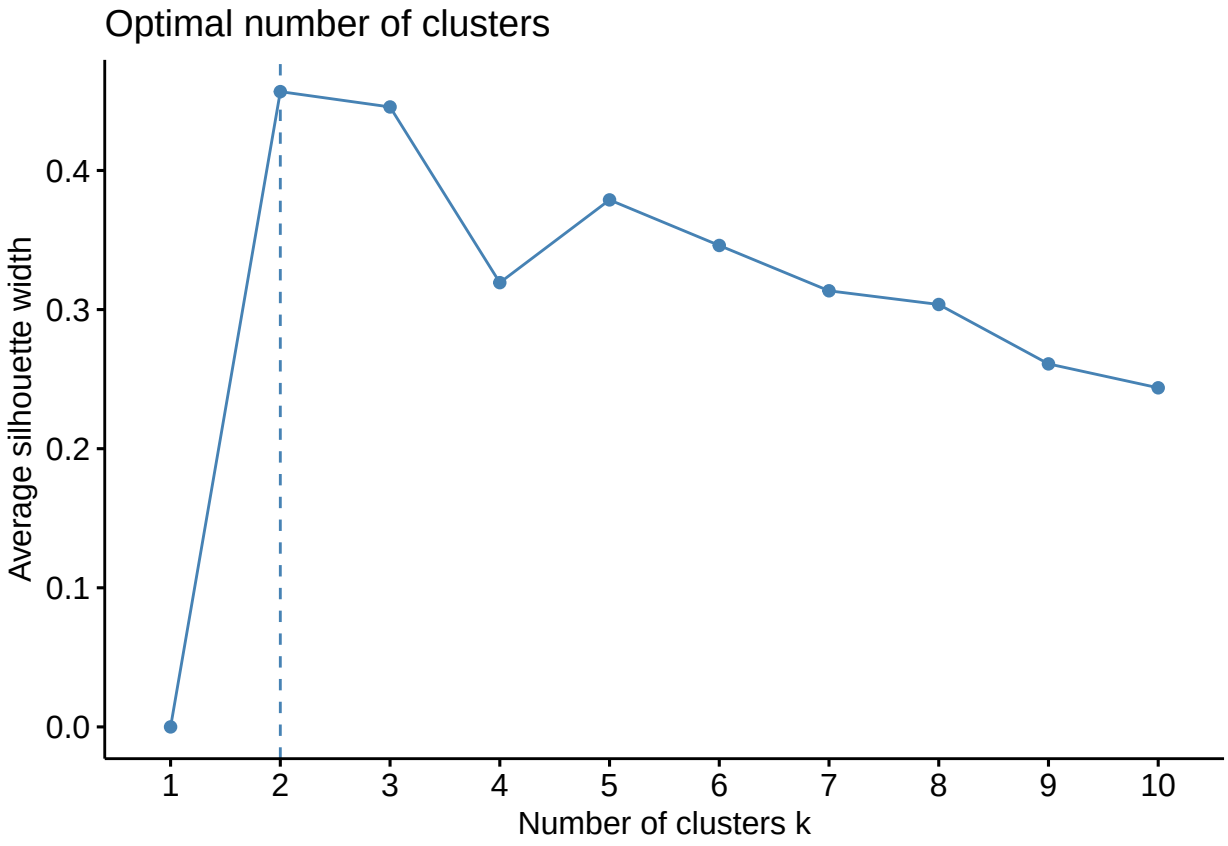



```
k10 <- kmeans(df[,3:8],10,iter.max = 100,nstart = 50,algorithm = "Lloyd")
s10 <- plot(silhouette(k10$cluster,dist(df[,3:8],"euclidean")),col = "#1287A5",border = NA)
```



The maximum average silhouette width we find testing clusters from 2 to 10, is 45, belong to a number of clusters of 3. So according to the silhouette method the optimal number of clusters is 3.

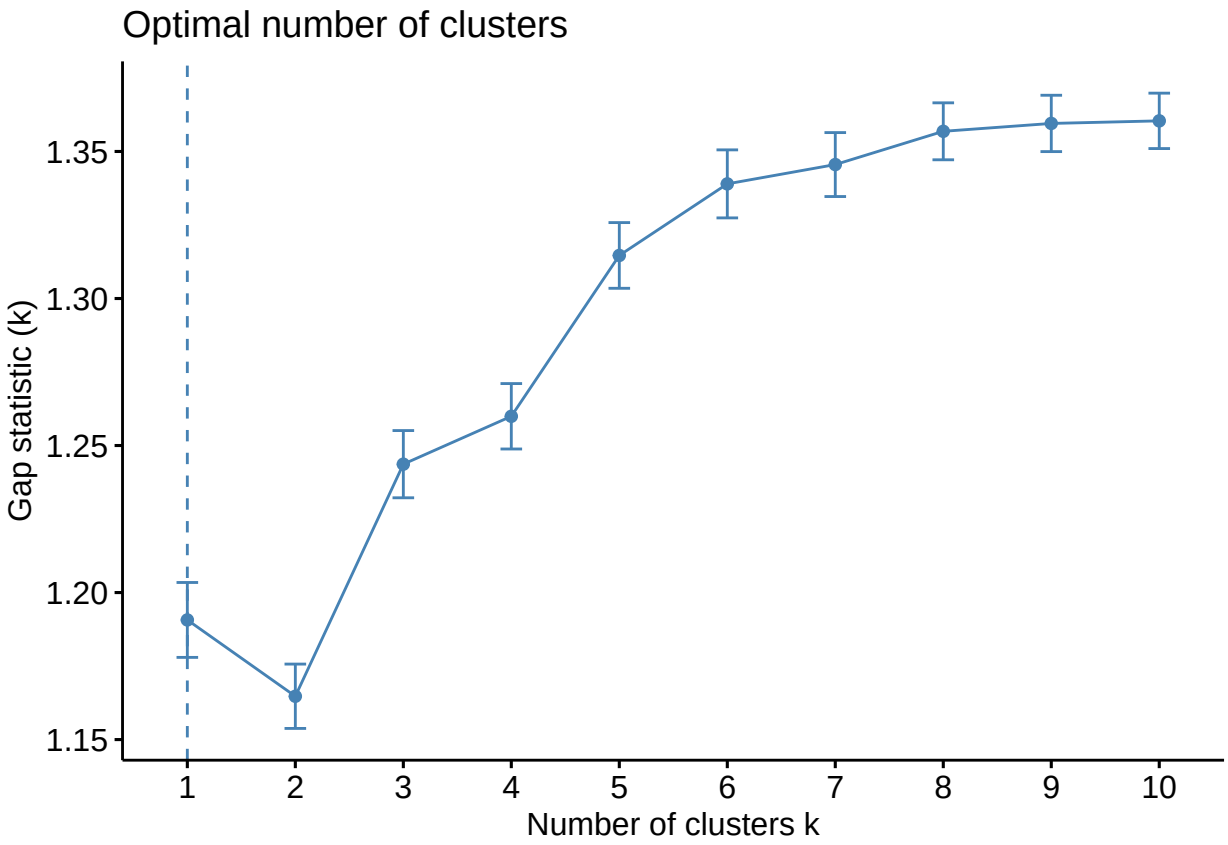
```
fviz_nbclust(df[,3:8],kmeans,method = "silhouette")
```



GAP Statistic

In order to compute the gap statistics method we can utilize the clusGap function for providing gap statistic as well as standard error for a given output.

```
set.seed(125)
stat_gap <- clusGap(df[,3:8],FUN=kmeans,nstart=25,K.max = 10,B=50)
fviz_gap_stat(stat_gap)
```



We choose 3 as our optimal number of clusters.

```
Number.of.K <- kmeans(df,3,iter.max = 100,nstart = 50,algorithm = "Lloyd")
Number.of.K
```

```
## K-means clustering with 3 clusters of sizes 49, 311, 77
##
## Cluster means:
##    Channel   Region     Fresh      Milk   Grocery   Frozen Detergents_Paper
## 1 1.959184 2.428571  8149.837 18715.857 27756.592 2034.714        12523.020
## 2 1.263666 2.546624  7422.563  3900.363  5368.129 2505.839         1823.119
## 3 1.168831 2.597403 31047.494  4284.494  5416.325 4655.649         1049.091
##   Delicassen
## 1   2282.143
## 2   1130.704
## 3   1887.883
##
```

```
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   2   2   2   2   3   2   2   2   2   1   2   2   3   3   3   2   2   2   2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   3   1   3   2   2   2   1   3   2   2   3   3   2   2   3   2   1   3
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   3   2   2   1   2   1   1   1   2   1   2   2   3   2   3   2   1   2   2   2
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   2   1   2   2   2   1   2   3   2   2   2   2   2   3   2   3   2   1   2   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   2   2   2   3   2   1   1   3   2   3   2   2   1   2   2   2   2   2   2   2
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   2   2   2   3   2   2   2   2   2   1   2   1   3   2   2   2   2   2   3   2
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   2   2   2   2   3   3   3   3   2   3   2   2   2   2   2   2   2   2   2   2
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   2   3   3   2   2   1   2   2   2   3   2   2   2   2   2   1   2   2   2   2
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   2   2   2   1   2   1   2   2   2   2   2   1   2   1   2   2   3   2   2   2
## 181 183 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   3   2   2   2   1   1
## 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222
##   3   2   2   1   2   2   2   1   2   1   2   2   2   2   1   2   2   2   2   2
## 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242
##   2   2   2   2   3   2   2   2   2   2   3   2   2   2   2   2   2   3   3   3
## 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262
##   2   2   2   2   2   2   2   2   2   1   2   3   2   3   2   2   3   3   2   2
## 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282
##   3   2   2   1   1   3   1   2   2   2   2   3   2   2   3   2   2   2   2   2
## 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302
##   3   3   3   3   2   2   2   3   2   2   2   2   3   2   2   2   2   2   2   1
## 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322
##   2   2   1   2   1   2   2   1   2   3   1   2   2   2   2   2   2   1   2   2
## 323 324 325 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343
##   2   2   3   2   2   2   2   2   1   3   1   2   3   2   2   2   2   2   2   2
## 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363
##   1   2   2   2   3   2   1   2   1   2   1   2   2   3   2   2   2   3   2   2
## 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383
##   2   2   2   2   2   3   2   3   3   2   2   2   2   2   3   2   2   3   2   3
## 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403
##   2   1   2   2   3   2   2   2   2   2   3   2   2   2   2   2   2   2   3   3
## 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423
##   3   2   2   3   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   3
## 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440
##   2   2   2   2   3   2   2   2   2   3   2   2   3   3   1   2   2
##
## Within cluster sum of squares by cluster:
## [1] 26105098679 24345397138 14721231297
##  (between_SS / total_SS =  52.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Visualizing the Clustering Results using the First Two Principle Components

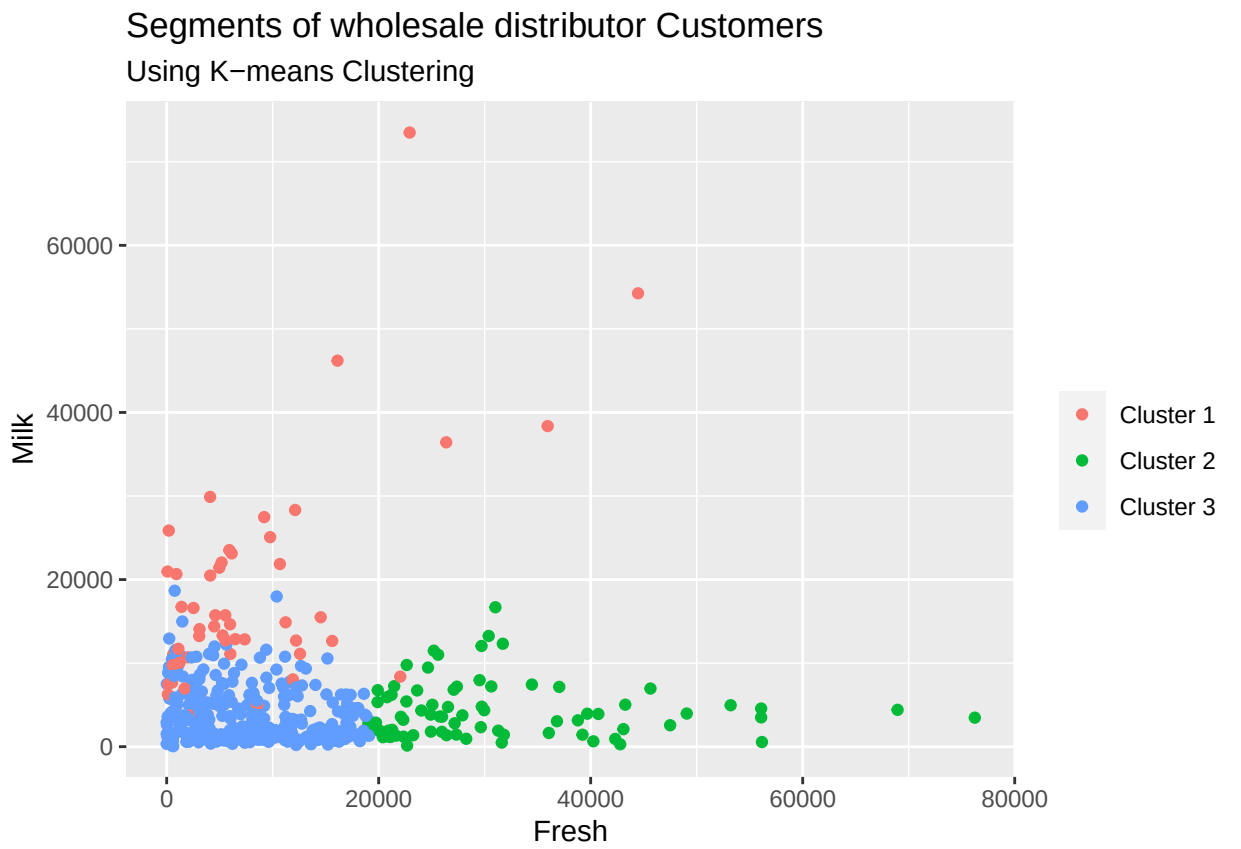
```
pc <- prcomp(df[3:8],scale. = FALSE)
summary(pc)
```

```
## Importance of components:
##                             PC1       PC2       PC3       PC4       PC5
## Standard deviation     12119.007 1.158e+04 4.198e+03 3.348e+03 1.699e+03
## Proportion of Variance     0.467 4.260e-01 5.603e-02 3.565e-02 9.180e-03
## Cumulative Proportion      0.467 8.931e-01 9.491e-01 9.848e-01 9.939e-01
##                               PC6
## Standard deviation     1.381e+03
## Proportion of Variance 6.070e-03
## Cumulative Proportion  1.000e+00
```

```
pc$rotation[,1:2]
```

```
##                         PC1        PC2
## Fresh            -0.51406897 0.84883987
## Milk              0.41710650 0.29435789
## Grocery           0.66562609 0.39715126
## Frozen           -0.08006436 0.07709103
## Detergents_Paper  0.33466550 0.16297091
## Delicassen        0.01690789 0.05091799
```

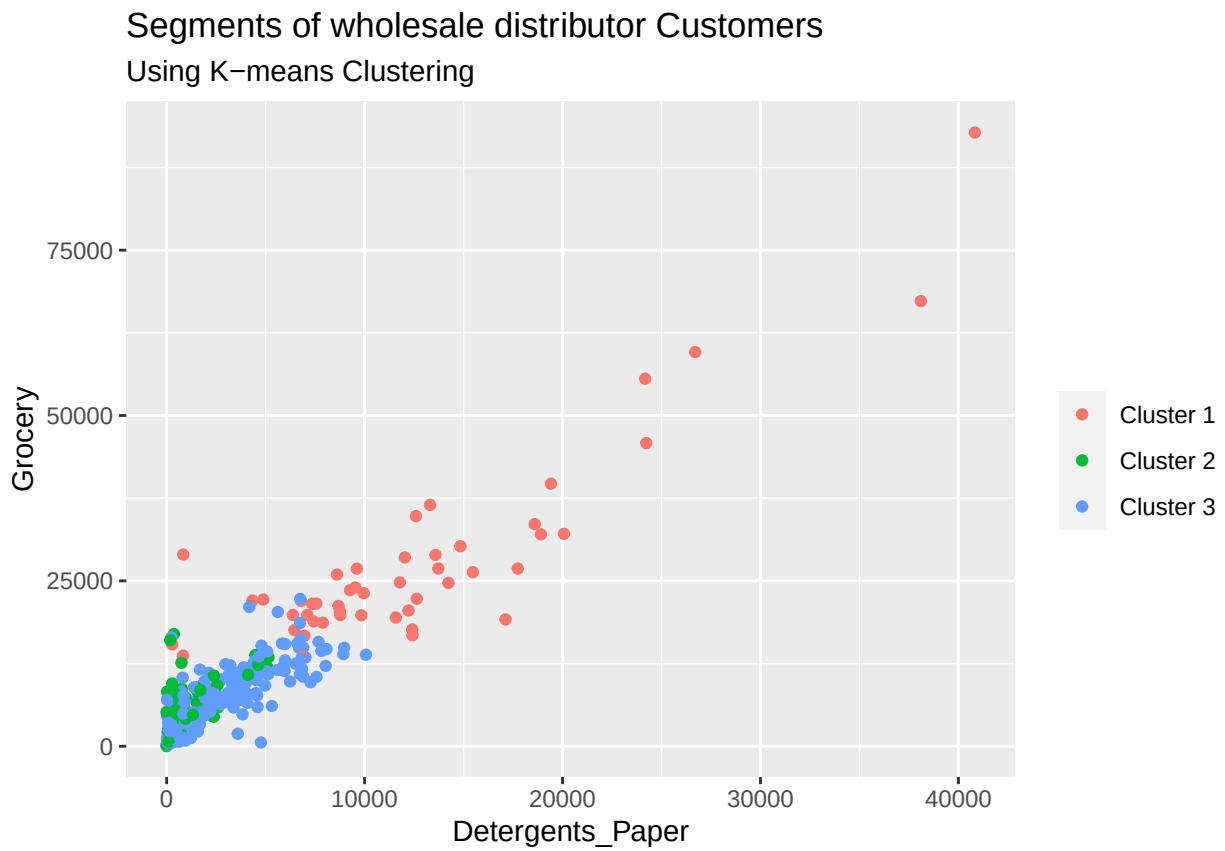
```
set.seed(123)
ggplot(df, aes(x =Fresh, y = Milk)) +
  geom_point(stat = "identity", aes(color = as.factor(k3$cluster))) +
  scale_color_discrete(name=" ",
              breaks=c("1", "2", "3"),
              labels=c("Cluster 1", "Cluster 2", "Cluster 3")) +
  ggtitle("Segments of wholesale distributor Customers",
          subtitle = "Using K-means Clustering")
```



From the above scatterplot, we observe that there is a distribution of 3 clusters as follows:

- Cluster 1 : it represents the customers with the lowest annual spending on Milk and Fresh products.
- Cluster 2 : it represents the customers with the lowest annual spending on Milk and highest annual spending on Fresh products.
- Cluster 3 : it represents the customers with the highest annual spending on Milk and lowest annual spending on Fresh products.

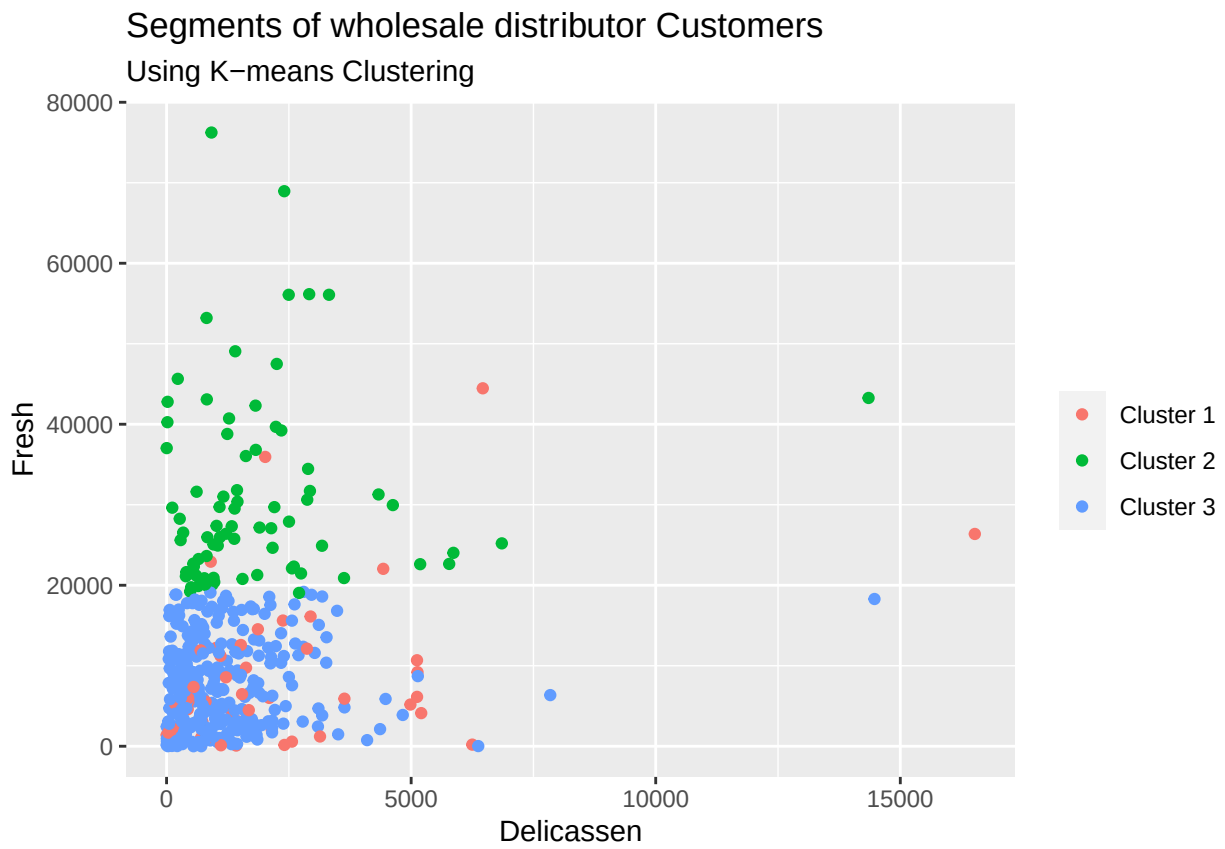
```
set.seed(123)
ggplot(df, aes(x =Detergents_Paper, y = Grocery)) +
  geom_point(stat = "identity", aes(color = as.factor(k3$cluster))) +
  scale_color_discrete(name=" ",
                       breaks=c("1", "2", "3"),
                       labels=c("Cluster 1", "Cluster 2", "Cluster 3")) +
  ggtitle("Segments of wholesale distributor Customers",
        subtitle = "Using K-means Clustering")
```



From the above scatterplot, we observe that there is a linear dependency between *Grocery* and *Detergents_Papers* variables, we notice that there is a distribution of the 3 clusters as follows:

- Clusters 1 & 2: represent customers with the lowest annual spending on Grocery and Detergents_Papers products.
- Cluster 3: represents customers with the highest annual spending on Grocery and Detergents_Papers products.

```
set.seed(123)
ggplot(df, aes(x =Delicassen, y = Fresh)) +
  geom_point(stat = "identity", aes(color = as.factor(k3$cluster))) +
  scale_color_discrete(name=" ",
                       breaks=c("1", "2", "3"),
                       labels=c("Cluster 1", "Cluster 2", "Cluster 3")) +
  ggtitle("Segments of wholesale distributor Customers", subtitle = "Using K-means Clustering")
```



From the above scatterplot, we notice taht there is a distribution of the 3 clusters as follows:

- Clusters 1 & 3: represent customers with the lowest annual spending on Fresh products.
- Cluster 2: represents customers with the highest annual spending on Fresh products.

Conclusion

Thanks to clustering and unsupervized machine learning algorithm K-means clustering, we managed to identify segments of customers according to their spending patterns in the wholstore destributor, which will help them target their customers based on their spending habits.