ThesisSurveyCluster

04 June, 2021

Barcelona School of Management

Clustering & Correspondence analysis

Steps: 1. Decide which columns to use, decide coding to get homogeneous for clustering

2. Clustering

3. Correspondence analysis, bring in demographics.

4. Individual cluster analysis - sub-sample

Loading datasets & Packages

[Reference to Methodology applied.]()

Transforming variables

Primarily several multi-option variables are removed from the data set. Next dummies are created for the categorical variables.

If the option is not selected, the answer corresponds with 0.
If the option has been selected, the answer corresponds with 1.

```
data <- separate(
  data,
  Criteria_Type_Coffee,
  into = c("Criteria_A", "Criteria_B"),
  sep = "([,])",
  remove = TRUE,
  convert = FALSE,
  extra = "drop",
  fill = "right",
)

data <- separate(
  data,
  Subscription_Not_Likely,
  into = c("Subscription_A", "Subscription_B"),
  sep = "([,])",
  remove = TRUE,
  convert = FALSE,
  extra = "drop",
  fill = "right",
)

data <- separate(
  data,
  "Supermarket_Negative_ Reasons",
  into = c("Supermarket_NO_A", "Supermarket_NO_B"),
  sep = "([,])",
  remove = TRUE,
  convert = FALSE,
  extra = "drop",
```

```r
  fill = "right",       )                data <- separate(         data,
  "Supermarket_Positive_ Reasons",
  into = c("Supermarket_YES_A", "Supermarket_YES_B"),         sep = "([,])",
  remove = TRUE,          convert = FALSE,         extra = "drop",
  fill = "right",       )
```

One-hot Encoding

The variables selected for clustering are: 1. Purchase Location 2. Frequency Specialty coffee consumption 3. Amount brand change 4. Amount consumed per week 5. Money spend on coffee 6. Likeliness to set up and app 7. Likeliness to set up and subscription

```r
# "AmountWeek", "MoneyCoffee"
```

Separate the second, responses to 0.1. Do not categorize on demographics together with the preferences. Everything to do with coffee has to be in cluster analysis.

Which variables to use

```r
#BrandChange, PurchaseLocation, App_Likely, Subscription_Likely,
```

Clustering including categorical variables

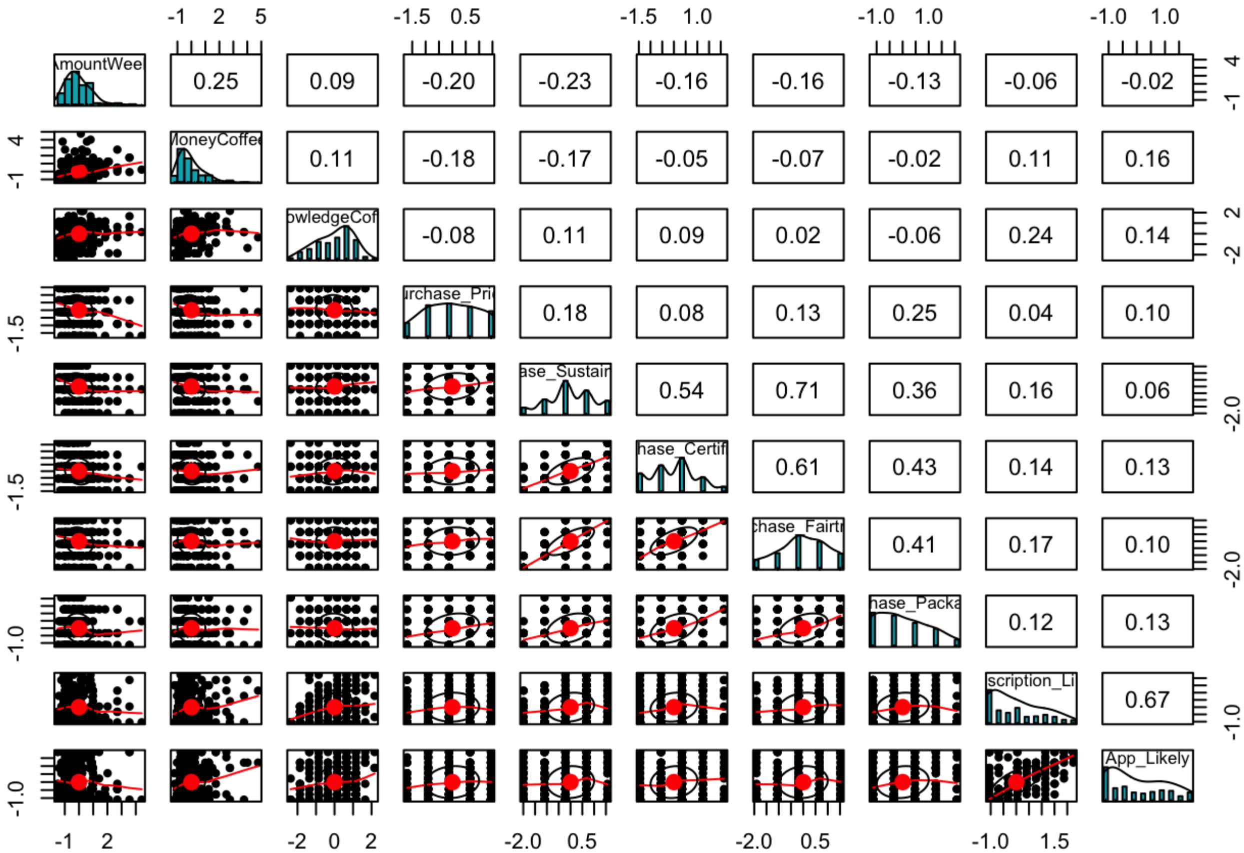
The next step is to remove this missing variables (NA), standardize the numerical variables and combine to make the new data table to be used for the cluster analysis.

```r
# Prepare Data
Orgdata <- na.omit(dataf[,c(11:23)]) # listwise deletion of missing
stdata <- scale(dataf[,c(0:10)]) # standardize variables
NewData <- cbind(Orgdata, stdata)
```

```r
names(NewData)
```

```
 [1] "BrandChange_Every time"
 [2] "BrandChange_Never"
 [3] "BrandChange_Sometimes"
 [4] "BrandChange_Very often"
 [5] "PurchaseLocation_E-commerce"
 [6] "PurchaseLocation_Online subscription"
 [7] "PurchaseLocation_Specialty stores or cafés"
 [8] "PurchaseLocation_The supermarket"
 [9] "Frequency_Specialty_Always"
[10] "Frequency_Specialty_I do (did) not know what this is"
[11] "Frequency_Specialty_Never"
[12] "Frequency_Specialty_Only in cafes"
[13] "Frequency_Specialty_Sometimes"
[14] "AmountWeek"
[15] "MoneyCoffee"
[16] "KnowledgeCoffee"
[17] "Purchase_Price"
[18] "Purchase_Sustainability"
[19] "Purchase_Certificate"
[20] "Purchase_Fairtrade"
[21] "Purchase_Packaging"
[22] "Subscription_Likely"
[23] "App_Likely"
```

Correlation Matrix



Performing clusters

```
set.seed(1234)

# K-Means Cluster Analysis
fit <- kmeans(na.omit(NewData), centers = 4, nstart = 50) #4 cluster solution
fit$betweenss/fit$totss
```

```
[1] 0.2881619
```

```
fit2 <- kmeans(na.omit(NewData), centers = 3, nstart = 50) #3 cluster solution
fit2$betweenss/fit2$totss
```

```
[1] 0.2388196
```

```
fit3 <- kmeans(na.omit(NewData), centers = 2, nstart = 50) #2 cluster solution
fit3$betweenss/fit3$totss
```

```
[1] 0.1660102
```

```
fit4 <- kmeans(na.omit(NewData), centers =1, nstart = 50) #1 cluster solution
fit4$betweenss/fit4$totss
```

```
[1] 0.00000000000002368555
```

% between-group variance:

Finding optimal number of clusters

Although the execution of the algorithm always improves the Ward criterion at each step and converges, that does not mean that the final solution is the best one (i.e., it does not necessarily maximize between necessarily maximize between

-group variance equivalent to group variance, equivalent to minimizing within-group variance). It depends on the starting seeds.

• SOLUTION: Repeat the algorithm repeatedly from different starting seeds and choose the best solution. Often there are several identical best solutions from different starting points.

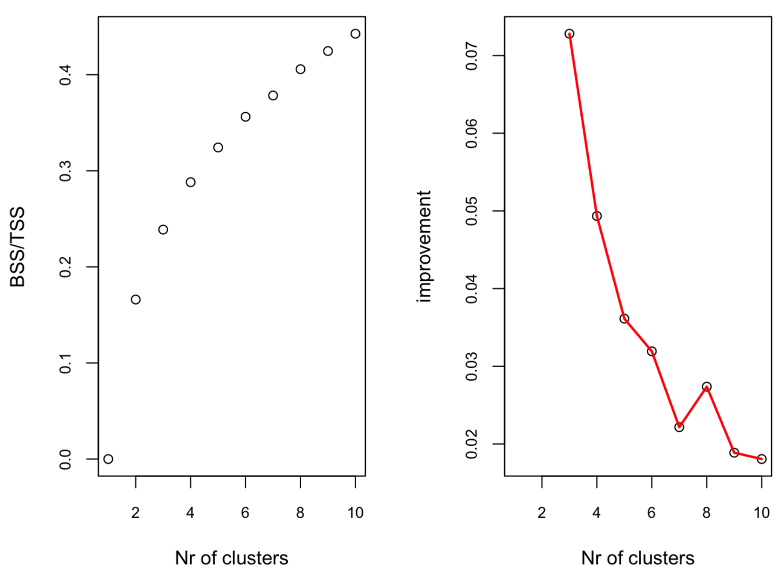
• The algorithm applies to a pre-defined number of clusters. How many clusters should be retained?

• SOLUTION: Repeat the whole process on different numbers of clusters, compare the solutions and decide what appears to be an empirically justified number of clusters, combined on domain knowledge domain knowledge.

```
## looping on k-means algorithm to decide how many clusters
set.seed(1234)
lifestyle.BW <- rep(0, 10)
for(nc in 2:10) {
  lifestyle.km <- kmeans(NewData, centers=nc, nstart=20, iter.max=200)
  lifestyle.BW[nc] <- lifestyle.km$betweenss/lifestyle.km$totss
}
lifestyle.BW
```

```
 [1] 0.0000000 0.1660102 0.2388196 0.2881619 0.3242922 0.3562143 0.3783748
 [8] 0.4057495 0.4246262 0.4426855
```

```
## plot the proportion of between-cluster variance
par(mar=c(4.2,4,1,2), cex.axis=0.8, mfrow=c(1,2))
plot(lifestyle.BW, xlab="Nr of clusters", ylab="BSS/TSS")

## plot the increments in between-cluster variance
lifestyle.BWinc <- lifestyle.BW[2:10]-lifestyle.BW[1:9]
plot(1:10, c(NA,NA, lifestyle.BWinc[2:9]), xlab="Nr of clusters", ylab="improvement")
lines(3:10, lifestyle.BWinc[2:9], col="red", lwd=2)
```



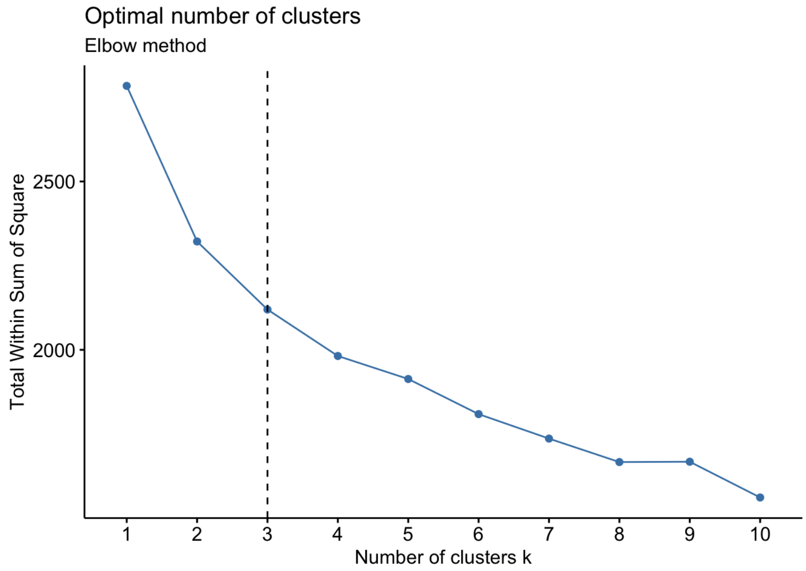
Elbow method

Based on the graph below, I have decided to use 4 numbers of cluster.

Distance

```
NewData <- na.omit(NewData)
distance <- get_dist(NewData)
graph <- fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high =
        "#FC4E07"))
```

```
library("NbClust")

# Elbow method
fviz_nbclust(NewData, kmeans, method = "wss") +
    geom_vline(xintercept = 3, linetype = 2)+
  labs(subtitle = "Elbow method")
```

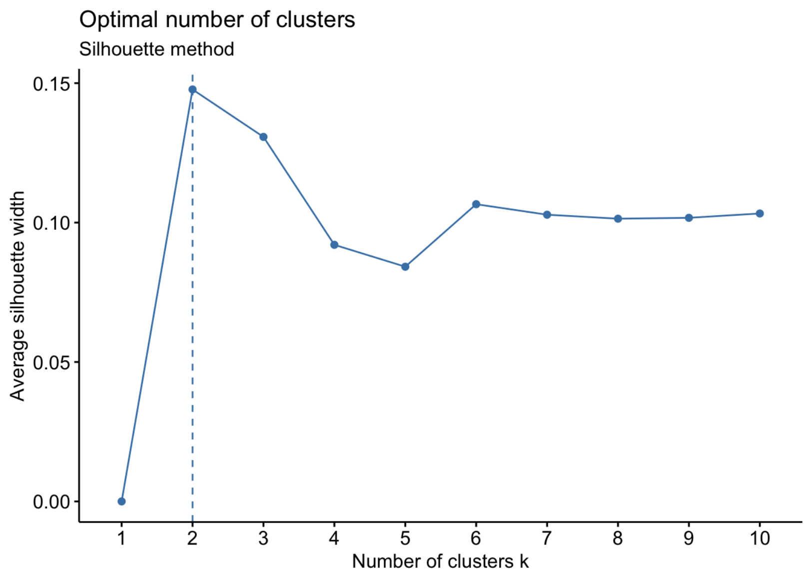


Average silhoutte method

The average silhouette approach we'll be described comprehensively in the chapter cluster validation statistics. Briefly, it measures the quality of a clustering. That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering.

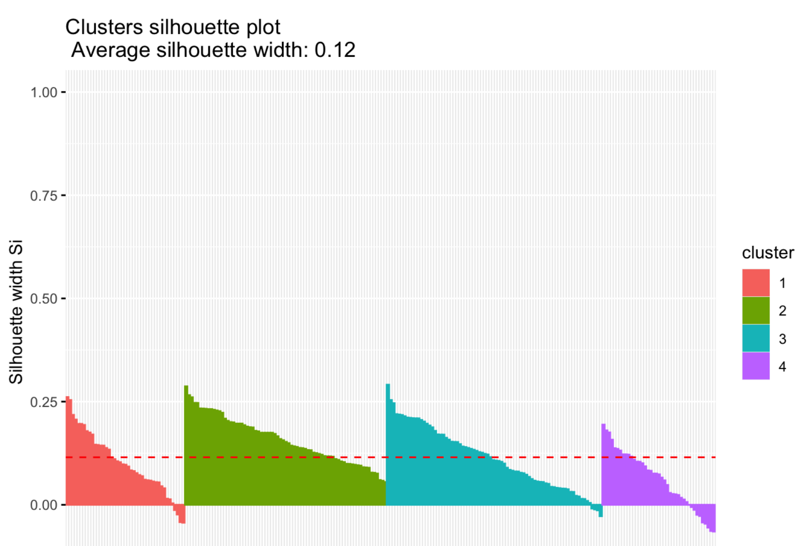
Average silhouette method computes the average silhouette of observations for different values of k. The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k (Kaufman and Rousseeuw 1990).

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



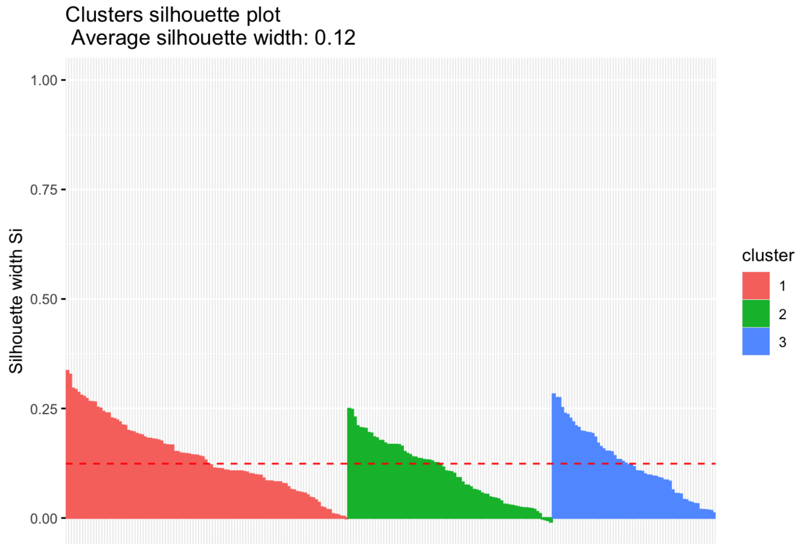

```
sil <- silhouette(fit$cluster, dist(NewData))
fviz_silhouette(sil)
```

```
  cluster size ave.sil.width
1       1   43          0.10
2       2   73          0.16
3       3   78          0.11
4       4   41          0.06
```



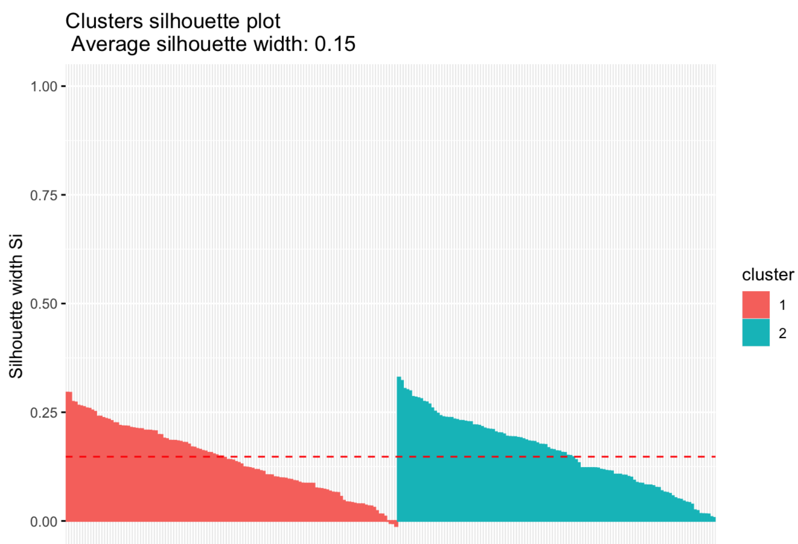

```
sil <- silhouette(fit2$cluster, dist(NewData))
fviz_silhouette(sil)
```

```
  cluster size ave.sil.width
1       1  102          0.14
2       2   74          0.10
3       3   59          0.12
```

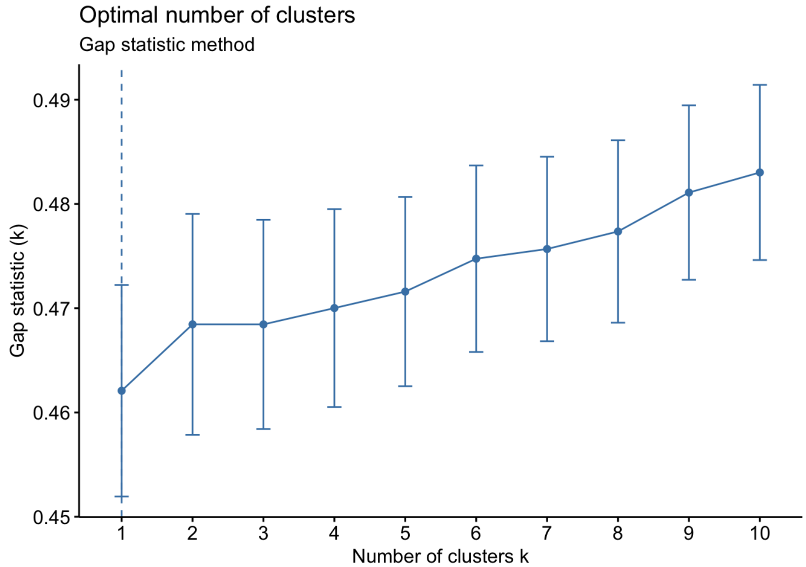



```r
sil <- silhouette(fit3$cluster, dist(NewData))
fviz_silhouette(sil)
```

```
  cluster size ave.sil.width
1       1  120          0.14
2       2  115          0.16
```




```r
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(NewData, kmeans, nstart = 25,  method = "gap_stat", nboot = 50, verbose =
        FALSE)+
  labs(subtitle = "Gap statistic method")
```



http://web.stanford.edu/~hastie/Papers/gap.pdf

http://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/#at_pco=wnm-1.0&at_si=609664423560aa01&at_ab=per-2&at_pos=0&at_tot=1

Adding cluster classification to the original data set

```
# append cluster assignment
mydata <- data.frame(na.omit(dataf), cluster = fit$cluster)
```

Getting cluster means:

```
fitMeans <- aggregate(mydata,
  by=list(cluster = fit2$cluster),
  FUN=mean
  )

fitMeans <- round(fitMeans,1)
my_table(fitMeans)
```

cluster	AmountWeek	MoneyCoffee	KnowledgeCoffee	Purchase_Price	Purchase_Sustainability	Purchase_Certificate

Purchase_Fairtrade	Purchase_Packaging	Subscription_Likely	App_Likely	BrandChange_Every.time	BrandChange_Never

BrandChange_Sometimes	BrandChange_Very.often	PurchaseLocation_E.commerce	PurchaseLocation_Online.subscription

PurchaseLocation_Specialty.stores.or.cafés	PurchaseLocation_The.supermarket	Frequency_Specialty_Always

Frequency_Specialty_I.do..did..not.know.what.this.is	Frequency_Specialty_Never	Frequency_Specialty_Only.in.cafes

Frequency_Specialty_Sometimes	cluster																	
1	22.3	25.7	5.4	2.8	2.4	1.9	2.4	1.7	2.4	3.1	0	0.4	0.6	0.0	0.2	0.0	0.0	0.8

0.0	0.3	0.2	0.2	0.2	2.8																			
2	17.9	31.2	6.6	3.4	3.6	3.0	3.7	2.8	6.9	7.1	0	0.2	0.6	0.1	0.2	0.1	0.2	0.4	0.2	0.1	0.1	0.2	0.4	3.0
3	12.6	17.5	5.1	3.4	4.2	3.5	4.2	3.1	2.6	2.9	0	0.4	0.5	0.2	0.2	0.0	0.1	0.7	0.1	0.3	0.2	0.2	0.2	1.4

Getting cluster medians:

```
fitMeans <- aggregate(mydata,
  by=list(cluster = fit$cluster),
  FUN=median
  )

fitMeans <- round(fitMeans,1)
my_table(fitMeans)
```

cluster	AmountWeek	MoneyCoffee	KnowledgeCoffee	Purchase_Price	Purchase_Sustainability	Purchase_Certificate

Purchase_Fairtrade	Purchase_Packaging	Subscription_Likely	App_Likely	BrandChange_Every.time	BrandChange_Never

BrandChange_Sometimes	BrandChange_Very.often	PurchaseLocation_E.commerce	PurchaseLocation_Online.subscription

PurchaseLocation_Specialty.stores.or.cafés	PurchaseLocation_The.supermarket	Frequency_Specialty_Always

Frequency_Specialty_I.do..did..not.know.what.this.is	Frequency_Specialty_Never	Frequency_Specialty_Only.in.cafes

Frequency_Specialty_Sometimes	cluster																							
1	13.0	15.0	6	3	5	4	4	3	2	2	0	0	0	0	0	0	0	1	0	0	0	0	0	1
2	14.0	10.0	5	4	3	2	3	2	2	2	0	0	1	0	0	0	0	1	0	0	0	0	0	2
3	14.5	22.5	7	4	4	3	4	3	7	7	0	0	1	0	0	0	0	0	0	0	0	0	0	3
4	28.0	40.0	6	2	2	2	2	1	2	3	0	1	0	0	0	0	0	1	0	0	0	0	0	4

Summary fit k=4

```
print(fit2)
```

```
K-means clustering with 3 clusters of sizes 102, 74, 59

Cluster means:
  BrandChange_Every time BrandChange_Never BrandChange_Sometimes
1             0.00000000         0.3823529             0.5882353
2             0.04054054         0.2162162             0.5945946
3             0.00000000         0.3728814             0.4745763
  BrandChange_Very often PurchaseLocation_E-commerce
1             0.02941176                   0.1666667
2             0.14864865                   0.1891892
3             0.15254237                   0.1525424
  PurchaseLocation_Online subscription
1                           0.02941176
2                           0.13513514
3                           0.01694915
  PurchaseLocation_Specialty stores or cafés PurchaseLocation_The supermarket
1                                 0.04901961                        0.7549020
2                                 0.24324324                        0.4324324
3                                 0.10169492                        0.7288136
  Frequency_Specialty_Always
1                 0.03921569
2                 0.24324324
3                 0.11864407
  Frequency_Specialty_I do (did) not know what this is
1                                           0.31372549
2                                           0.08108108
3                                           0.28813559
  Frequency_Specialty_Never Frequency_Specialty_Only in cafes
1                0.24509804                         0.1960784
2                0.05405405                         0.2162162
3                0.20338983                         0.1864407
  Frequency_Specialty_Sometimes AmountWeek MoneyCoffee KnowledgeCoffee
1                     0.2058824  0.3284711  0.01796938      -0.1483638
2                     0.4054054 -0.0523662  0.29286523       0.4336750
3                     0.2033898 -0.5021856 -0.39838821      -0.2874381
  Purchase_Price Purchase_Sustainability Purchase_Certificate
1     -0.2616285              -0.7166634           -0.6757643
2      0.2048667               0.3284436            0.3067941
3      0.1953555               0.8270311            0.7834779
  Purchase_Fairtrade Purchase_Packaging Subscription_Likely App_Likely
1         -0.7419088         -0.5487640          -0.5276576 -0.3892223
2          0.3815369          0.3061075           1.1008984  0.9035015
3          0.8040842          0.5647792          -0.4685662 -0.4603124

Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
  3   2   1   1   3   1   2   3   3   2   1   1   3   3   3   1   2   1   2   3
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
```

```
 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 
  2   1   1   1   3   1   1   1   1   1   1   3   3   1   3   3   1   2   3   1 
 61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 
  1   3   1   2   2   1   1   3   1   1   1   1   1   1   1   3   1   2   1   1 
 81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 
  3   3   1   3   3   1   1   2   1   1   2   1   3   1   1   1   1   1   1   1 
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 
  1   1   1   1   1   2   2   3   1   1   3   3   1   3   1   1   2   1   1   1 
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 
  1   2   2   1   1   1   2   2   1   3   1   1   3   2   3   3   1   2   2   1 
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 
  3   2   2   2   1   2   2   2   3   1   1   3   3   1   2   1   2   3   3   2 
161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 
  3   3   2   1   2   2   2   1   2   2   2   3   2   1   2   2   1   1   3   1 
181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 
  2   3   2   1   2   1   2   1   2   2   2   3   3   2   2   3   3   2   2   2 
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 
  1   3   2   2   1   1   1   2   2   3   2   1   3   3   1   2   2   1   2   3 
221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 
  3   2   2   3   3   2   2   2   1   2   2   1   2   3   2 

Within cluster sum of squares by cluster:
[1] 938.0178 677.9900 503.0471
 (between_SS / total_SS =  23.9 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"      
```

Validation testing

https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods/

Generally, clustering validation statistics can be categorized into 3 classes (Charrad et al. 2014,Brock et al. (2008), Theodoridis and Koutroumbas (2008)):

1. Internal cluster validation, which uses the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information. It can be also used for estimating the number of clusters and the appropriate clustering algorithm without any external data.

2. External cluster validation, which consists in comparing the results of a cluster analysis to an externally known result, such as externally provided class labels. It measures the extent to which cluster labels match externally supplied class labels. Since we know the “true” cluster number in advance, this approach is mainly used for selecting the right clustering algorithm for a specific data set.

3. Relative cluster validation, which evaluates the clustering structure by varying different parameter values for the same algorithm (e.g.,: varying the number of clusters k). It’s generally used for determining the optimal number of clusters.

```r
## perform nonparametric ANOVA between-clusters to get chi-squares and p-values
var_names <- colnames(mydata[,-ncol(mydata)])

anova_tests <- list()
for (var_name in var_names) {
  anova_tests[[var_name]] <- kruskal.test(get(var_name) ~ cluster, data = mydata)
}

anova_tests
```

```
$AmountWeek
```

```
Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 43.447, df = 3, p-value = 0.000000001978


$MoneyCoffee

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 64.498, df = 3, p-value =
0.00000000000006424


$KnowledgeCoffee

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 17.858, df = 3, p-value = 0.0004706


$Purchase_Price

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 43.257, df = 3, p-value = 0.00000000217


$Purchase_Sustainability

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 127.92, df = 3, p-value <
0.00000000000000022


$Purchase_Certificate

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 99.679, df = 3, p-value <
0.00000000000000022


$Purchase_Fairtrade

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 125.13, df = 3, p-value <
0.00000000000000022


$Purchase_Packaging
```

```
data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 50.469, df = 3, p-value = 0.00000000006347


$Subscription_Likely

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 116.81, df = 3, p-value <
0.00000000000000022


$App_Likely

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 91.134, df = 3, p-value <
0.00000000000000022


$BrandChange_Every.time

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 6.0905, df = 3, p-value = 0.1073


$BrandChange_Never

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 11.988, df = 3, p-value = 0.007424


$BrandChange_Sometimes

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 7.8944, df = 3, p-value = 0.04825


$BrandChange_Very.often

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 10.612, df = 3, p-value = 0.01402


$PurchaseLocation_E.commerce

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 9.1007, df = 3, p-value = 0.02798
```

```
$PurchaseLocation_Online.subscription

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 14.898, df = 3, p-value = 0.001906


$PurchaseLocation_Specialty.stores.or.cafés

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 12.025, df = 3, p-value = 0.007297


$PurchaseLocation_The.supermarket

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 27.26, df = 3, p-value = 0.000005192


$Frequency_Specialty_Always

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 19.513, df = 3, p-value = 0.0002141


$Frequency_Specialty_I.do..did..not.know.what.this.is

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 11.879, df = 3, p-value = 0.007809


$Frequency_Specialty_Never

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 6.9876, df = 3, p-value = 0.0723


$Frequency_Specialty_Only.in.cafes

    Kruskal-Wallis rank sum test

data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 0.52388, df = 3, p-value = 0.9136


$Frequency_Specialty_Sometimes

    Kruskal-Wallis rank sum test
```

```
data:  get(var_name) by cluster
Kruskal-Wallis chi-squared = 2.7258, df = 3, p-value = 0.4359
```

```r
var_names <- colnames(mydata[,-ncol(mydata)])

anova_tests <- list()
for (var_name in var_names) {
  anova_tests[[var_name]] <- oneway.test(get(var_name) ~ cluster, data = mydata)
}

anova_tests
```

```
$AmountWeek

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 13.636, num df = 3.00, denom df = 106.71, p-value = 0.0000001353


$MoneyCoffee

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 21.883, num df = 3.00, denom df = 106.19, p-value =
0.00000000004136


$KnowledgeCoffee

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 7.3066, num df = 3.00, denom df = 109.19, p-value = 0.0001645


$Purchase_Price

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 22.371, num df = 3.00, denom df = 110.73, p-value =
0.00000000002134


$Purchase_Sustainability

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 87.626, num df = 3.00, denom df = 109.79, p-value <
0.00000000000000022


$Purchase_Certificate

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 51.637, num df = 3.00, denom df = 108.02, p-value <
```

```
0.00000000000000022


$Purchase_Fairtrade

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 91.727, num df = 3.00, denom df = 114.33, p-value <
0.00000000000000022


$Purchase_Packaging

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 26.102, num df = 3.00, denom df = 110.81, p-value =
0.0000000000007564


$Subscription_Likely

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 77.496, num df = 3.00, denom df = 109.96, p-value <
0.00000000000000022


$App_Likely

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 57.351, num df = 3.00, denom df = 107.36, p-value <
0.00000000000000022


$BrandChange_Every.time

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = NaN, num df = 3, denom df = NaN, p-value = NA


$BrandChange_Never

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 4.1307, num df = 3.0, denom df = 106.7, p-value = 0.008184


$BrandChange_Sometimes

	One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 2.6139, num df = 3.00, denom df = 109.52, p-value = 0.05485
```

```
$BrandChange_Very.often

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 3.7943, num df = 3.00, denom df = 102.05, p-value = 0.0126


$PurchaseLocation_E.commerce

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 2.5646, num df = 3.00, denom df = 104.76, p-value = 0.05861


$PurchaseLocation_Online.subscription

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = NaN, num df = 3, denom df = NaN, p-value = NA


$PurchaseLocation_Specialty.stores.or.cafés

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 4.1503, num df = 3.00, denom df = 104.64, p-value = 0.008028


$PurchaseLocation_The.supermarket

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 11.779, num df = 3.00, denom df = 105.59, p-value = 0.000001021


$Frequency_Specialty_Always

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 8.2549, num df = 3.000, denom df = 92.329, p-value = 0.00006354


$Frequency_Specialty_I.do..did..not.know.what.this.is

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 5.1188, num df = 3.00, denom df = 103.19, p-value = 0.002426


$Frequency_Specialty_Never

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
```

```
F = 2.2844, num df = 3.00, denom df = 107.48, p-value = 0.08304


$Frequency_Specialty_Only.in.cafes

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 0.17192, num df = 3.00, denom df = 109.97, p-value = 0.9152


$Frequency_Specialty_Sometimes

    One-way analysis of means (not assuming equal variances)

data:  get(var_name) and cluster
F = 0.86045, num df = 3.0, denom df = 110.7, p-value = 0.464
```

Within sum of squares:

```
print(fit$withinss)
```

```
[1] 374.0707 529.9048 680.1541 397.5611
```

```
print(fit2$withinss)
```

```
[1] 938.0178 677.9900 503.0471
```

```
print(fit3$withinss)
```

```
[1] 1202.231 1119.518
```

Between sum of squares

```
print(fit$betweenss)
```

```
[1] 802.2156
```

```
print(fit2$betweenss)
```

```
[1] 664.8514
```

```
print(fit3$betweenss)
```

```
[1] 462.1568
```

```
print(fit4$betweenss)
```

```
[1] 0.00000000006593837
```

Total sum of squares

```
print(fit$totss)
```

```
[1] 2783.906
```

```
print(fit2$totss)
```

```
[1] 2783.906
```

```
print(fit3$totss)
```

```
[1] 2783.906
```

```r
cluster <- c(1:4)
t.test(fit$withinss, cluster)
```

```
	Welch Two Sample t-test

data:  fit$withinss and cluster
t = 6.993, df = 3.0005, p-value = 0.006
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 268.6193 717.2261
sample estimates:
mean of x mean of y 
 495.4227    2.5000 
```

```r
t.test(fit2$withinss, cluster)
```

```
	Welch Two Sample t-test

data:  fit2$withinss and cluster
t = 5.57, df = 2.0001, p-value = 0.03075
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  160.1738 1247.5295
sample estimates:
mean of x mean of y 
 706.3516    2.5000 
```

```r
t.test(fit3$withinss, cluster)
```

```
	Welch Two Sample t-test

data:  fit3$withinss and cluster
t = 28.006, df = 1.0005, p-value = 0.02269
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  633.431 1683.319
sample estimates:
mean of x mean of y 
 1160.875     2.500 
```

```r
library("fpc")
cluster.stats(d = dist(NewData), fit2$cluster)
```

```
$n
[1] 235

$cluster.number
[1] 3

$cluster.size
[1] 102  74  59

$min.cluster.size
[1] 59

$noisen
[1] 0
```

```
$diameter
[1] 7.895409 7.880855 7.003931

$average.distance
[1] 4.196440 4.197662 4.088266

$median.distance
[1] 4.164674 4.101028 4.108588

$separation
[1] 1.588872 1.588872 1.670990

$average.toother
[1] 5.122106 5.037340 4.958412

$separation.matrix
         [,1]     [,2]     [,3]
[1,] 0.000000 1.588872 1.901169
[2,] 1.588872 0.000000 1.670990
[3,] 1.901169 1.670990 0.000000

$ave.between.matrix
         [,1]     [,2]     [,3]
[1,] 0.000000 5.167806 5.064787
[2,] 5.167806 0.000000 4.811788
[3,] 5.064787 4.811788 0.000000

$average.between
[1] 5.046551

$average.within
[1] 4.169666

$n.between
[1] 17932

$n.within
[1] 9563

$max.diameter
[1] 7.895409

$min.separation
[1] 1.588872

$within.cluster.ss
[1] 2119.055

$clus.avg.silwidths
        1         2         3
0.1390846 0.1029587 0.1243138

$avg.silwidth
[1] 0.1240003

$g2
NULL

$g3
NULL
```

```
$pearsongamma
[1] 0.3650153

$dunn
[1] 0.20124

$dunn2
[1] 1.146302

$entropy
[1] 1.073106

$wb.ratio
[1] 0.8262408

$ch
[1] 36.39489

$cwidegap
[1] 3.774041 4.223510 3.422740

$widestgap
[1] 4.22351

$sindex
[1] 1.896749

$corrected.rand
NULL

$vi
NULL
```

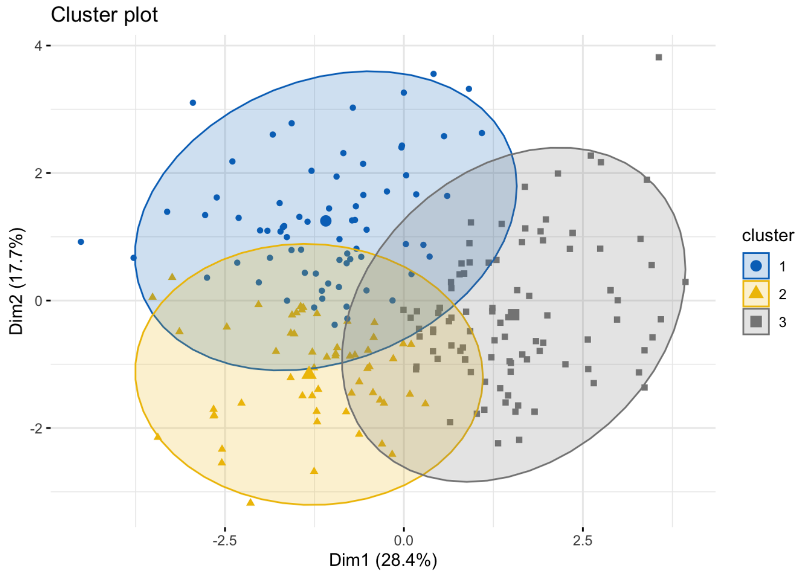
Creating new data sets for each cluster group

```
data <- read_excel("Main doc survey.xlsx")
clustereddata <- cbind(data, Cluster = fit2$cluster)

cluster1 <- subset(clustereddata, clustereddata$Cluster=='1')
cluster2 <- subset(clustereddata, clustereddata$Cluster=='2')
cluster3 <- subset(clustereddata, clustereddata$Cluster=='3')
cluster4 <- subset(clustereddata, clustereddata$Cluster=='4')
```

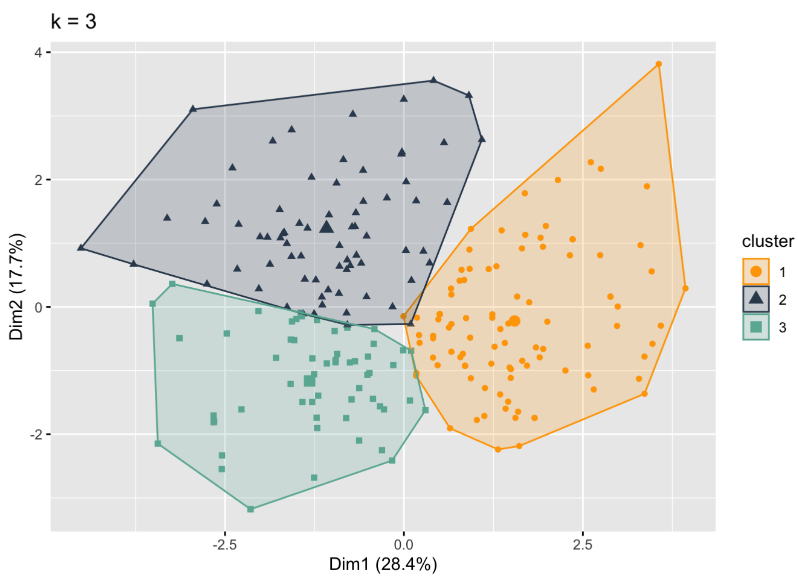
Visualizing clusters

Provides ggplot2-based elegant visualization of partitioning methods including kmeans [stats package]; pam, clara and fanny [cluster package]; dbscan [fpc package]; Mclust [mclust package]; HCPC [FactoMineR]; hkmeans [factoextra]. Observations are represented by points in the plot, using principal components if ncol(data) > 2. An ellipse is drawn around each cluster.

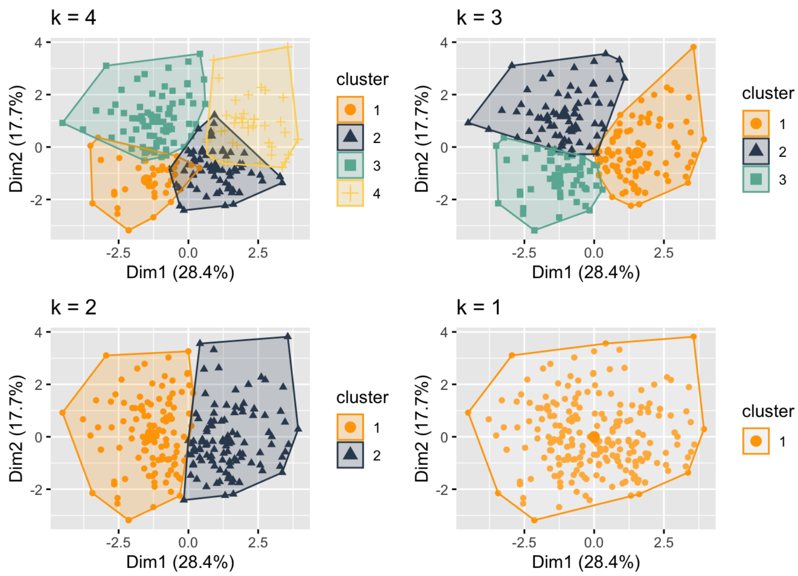
```
# K-means clustering
km.res <- eclust(stdata, "kmeans", k = 3, nstart = 25, graph = FALSE)
# Visualize k-means clusters
fviz_cluster(km.res, geom = "point", ellipse.type = "norm",
             palette = "jco", ggtheme = theme_minimal())
```



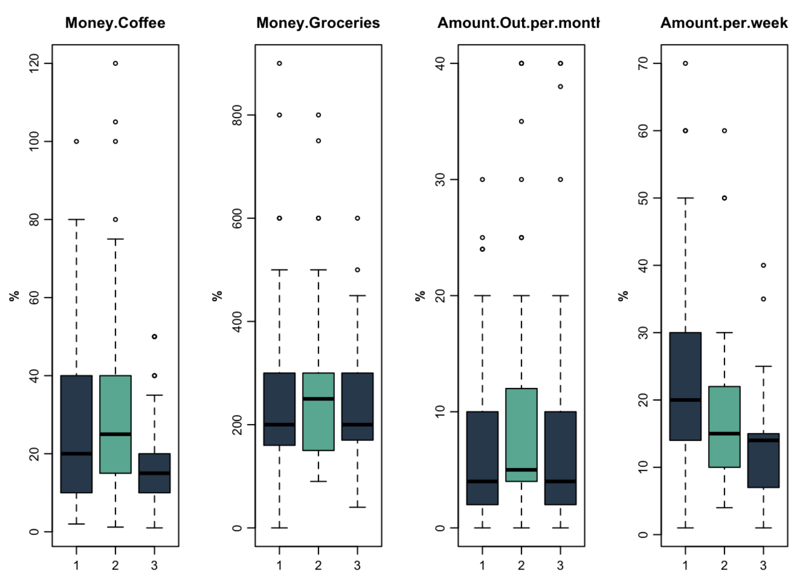

```
fviz_cluster(fit2, geom = "point", data = stdata, outlier.color = "black", palette =
        dani) + ggtitle("k = 3")
```



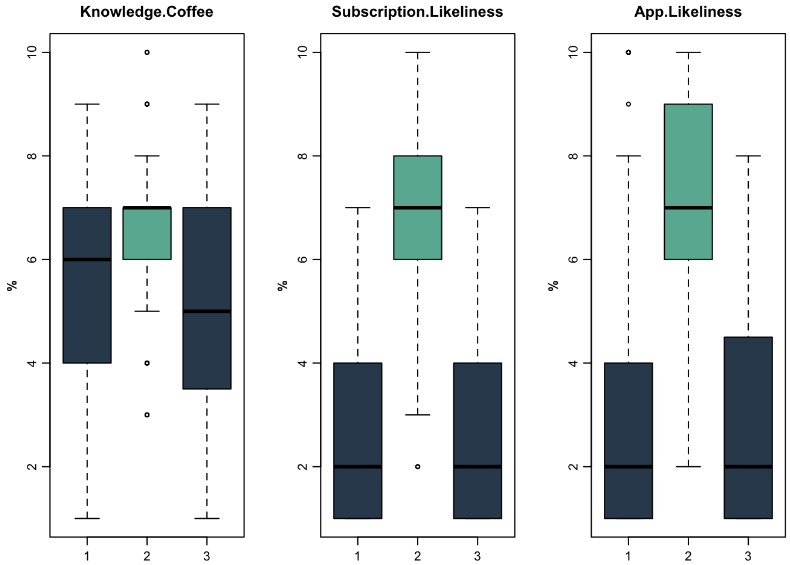

```
p1 <- fviz_cluster(fit, geom = "point", data = stdata, outlier.color = "black",
        palette = dani) + ggtitle("k = 4")

p2 <- fviz_cluster(fit2, geom = "point", data = stdata, outlier.color = "black",
        palette = dani) + ggtitle("k = 3")

p3 <- fviz_cluster(fit3, geom = "point", data = stdata, outlier.color = "black",
        palette = dani) + ggtitle("k = 2")

p4 <- fviz_cluster(fit4, geom = "point", data = stdata, outlier.color = "black",
        palette = dani) + ggtitle("k = 1")

grid.arrange(p1, p2, p3, p4, nrow = 2)
```



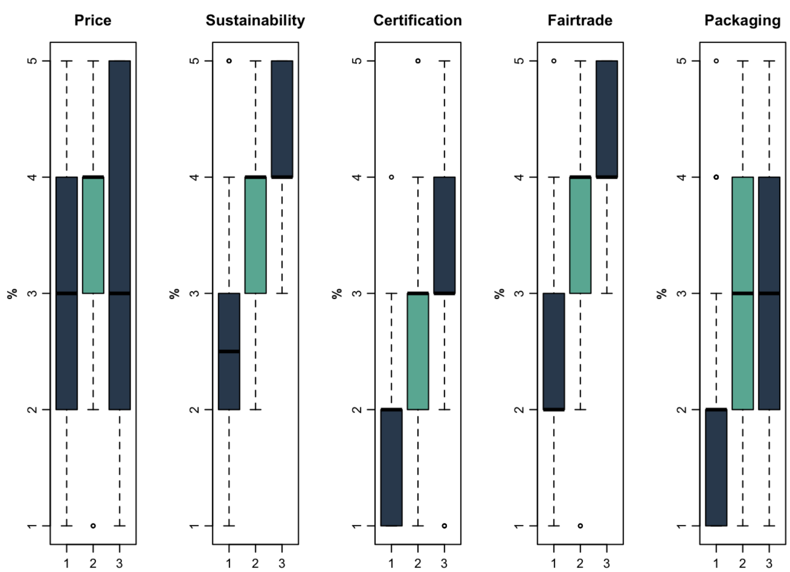

```
Subdata <- data.frame("Money Coffee" = clustereddata$MoneyCoffee, "Money Groceries" =
        clustereddata$MoneyGroceries, "Amount Out per month" =
        clustereddata$AmountOutMonth, 'Amount per week' = clustereddata$AmountWeek)

par(mar=c(2,4,3,1), font.lab=2, mfrow=c(1,4), mgp=c(2,0.7,0))
for(j in 1:4) boxplot(Subdata[,j] ~ clustereddata$Cluster, main=colnames(Subdata)[j],
                      col=c("#34495E", "#69b3a2"), ylab="%")
```




```
Subdata <- data.frame("Knowledge Coffee" = clustereddata$KnowledgeCoffee,
        'Subscription Likeliness' = clustereddata$Subscription_Likely, "App
        Likeliness" = clustereddata$App_Likely)

par(mar=c(2,4,3,1), font.lab=2, mfrow=c(1,3), mgp=c(2,0.7,0))
for(j in 1:3) boxplot(Subdata[,j] ~ clustereddata$Cluster, main=colnames(Subdata)[j],
                      col=c("#34495E", "#69b3a2"), ylab="%")
```




```
Subdata <- data.frame(Price = clustereddata$Purchase_Price, Sustainability =
        clustereddata$Purchase_Sustainability, Certification =
        clustereddata$Purchase_Certificate, Fairtrade =
        clustereddata$Purchase_Fairtrade, Packaging =
        clustereddata$Purchase_Packaging)

par(mar=c(2,4,3,1), font.lab=2, mfrow=c(1,5), mgp=c(2,0.7,0))
for(j in 1:5) boxplot(Subdata[,j] ~ clustereddata$Cluster, main=colnames(Subdata)[j],
                      col=c("#34495E", "#69b3a2"), ylab="%")
```



Robustness check

Same but with other linkage methods etc. If you get a very different pattern, your results are not robust.

The clusters individual results

```
data <- read_excel("Main doc survey.xlsx")
clustereddata <- cbind(data, Cluster = fit2$cluster)

cluster1 <- subset(clustereddata, clustereddata$Cluster=='1')
cluster2 <- subset(clustereddata, clustereddata$Cluster=='2')
cluster3 <- subset(clustereddata, clustereddata$Cluster=='3')
```

```
Results <- as.data.table(aggregate(na.omit(clustereddata[,2:5]),
        by=list(cluster=fit2$cluster), mean), by = round)

Results_Round <- round(Results)
my_table(Results_Round)
```

cluster	AmountWeek	AmountOutMonth	MoneyCoffee	MoneyGroceries
1	22	7	26	242
2	18	10	31	263
3	13	7	17	238

```
Results <- as.data.table(aggregate(na.omit(clustereddata[,c(12,19)]),
        by=list(cluster=fit2$cluster), median), by = round)

Results_Round <- round(Results,1)

my_table(Results_Round)
```

cluster	KnowledgeCoffee	Subscription_Likely
1	6	2
2	7	7
3	5	2

```
Results <- as.data.table(aggregate(na.omit(clustereddata[,13:17]),
        by=list(cluster=clustereddata$Cluster), median), by = round)

Results_Round <- round(Results)
my_table(Results_Round)
```

cluster	Purchase_Price	Purchase_Sustainability	Purchase_Certificate	Purchase_Fairtrade	Purchase_Packaging
1	3	2	2	2	

2					
2	4	4	3	4	3
3	3	4	3	4	3

```
agetable1 <- as.data.table(table(cluster1$AgeCategory))
colnames(agetable1) <- c("Age", "Frequency")

agetable2 <- as.data.table(table(cluster2$AgeCategory) )
colnames(agetable2) <- c("Age", "Frequency")

agetable3 <- as.data.table(table(cluster3$AgeCategory) )
colnames(agetable3) <- c("Age", "Frequency")

my_table(agetable1)
```

Age	Frequency
> 60	9
18-25	24
25-45	40
45-60	29

```
my_table(agetable2)
```

Age	Frequency
> 60	1
18-25	31
25-45	31
45-60	11

```
my_table(agetable3)
```

Age	Frequency
< 18	2
> 60	1
18-25	17
25-45	30
45-60	9

```
table1 <- as.data.table(table(cluster1$Machine))
colnames(table1) <- c("Machine", "Frequency")

table2 <- as.data.table(table(cluster2$Machine) )
colnames(table2) <- c("Machine", "Frequency")

table3 <- as.data.table(table(cluster3$Machine) )
colnames(table3) <- c("Machine", "Frequency")

my_table(table1)
```

Machine	Frequency
CupMachine	41

Espresso machine	27
Filter machine	24
French press	2
Instant coffee	3
Moka pot	5

```
my_table(table2)
```

Machine	Frequency
Aeropress	1
CupMachine	19
Espresso machine	35
Filter machine	7
French press	2
Instant coffee	1
Moka pot	5
V60	4

```
my_table(table3)
```

Machine	Frequency
CupMachine	14
Espresso machine	13
Filter machine	17
French press	5
Instant coffee	1
Moka pot	8
Percolator	1

```
table1 <- as.data.table(table(cluster1$PurchaseLocation))
colnames(table1) <- c("PurchaseLocation", "Frequency")

table2 <- as.data.table(table(cluster2$PurchaseLocation) )
colnames(table2) <- c("PurchaseLocation", "Frequency")

table3 <- as.data.table(table(cluster3$PurchaseLocation) )
colnames(table3) <- c("PurchaseLocation", "Frequency")

my_table(table1)
```

PurchaseLocation	Frequency
E-commerce	17
Online subscription	3
Specialty stores or cafés	5
The supermarket	77

```
my_table(table2)
```

E-commerce	14
Online subscription	10
Specialty stores or cafés	18
The supermarket	32

```
my_table(table3)
```

PurchaseLocation	Frequency
E-commerce	9
Online subscription	1
Specialty stores or cafés	6
The supermarket	43

```
table1 <- as.data.table(table(cluster1$Frequency_Specialty))
colnames(table1) <- c("PurchaseLocation", "Frequency")

table2 <- as.data.table(table(cluster2$Frequency_Specialty) )
colnames(table2) <- c("Frequency_Specialty", "Frequency")

table3 <- as.data.table(table(cluster3$Frequency_Specialty) )
colnames(table3) <- c("Frequency_Specialty", "Frequency")

my_table(table1)
```

PurchaseLocation	Frequency
Always	4
I do (did) not know what this is	32
Never	25
Only in cafes	20
Sometimes	21

```
my_table(table2)
```

Frequency_Specialty	Frequency
Always	18
I do (did) not know what this is	6
Never	4
Only in cafes	16
Sometimes	30

```
my_table(table3)
```

Frequency_Specialty	Frequency
Always	7
I do (did) not know what this is	17
Never	12
Only in cafes	11
Sometimes	12

Clusters groups

```
data <- read_excel("Main doc survey.xlsx")
clustereddata <- cbind(data, Cluster = fit2$cluster)

cluster1 <- clustereddata[clustereddata$Cluster=='1',]
cluster2 <- clustereddata[clustereddata$Cluster=='2',]
cluster3 <- clustereddata[clustereddata$Cluster=='3',]
```

Cluster 1 medians

```
var_names <- colnames(cluster1[,-ncol(cluster1)])

medians1 <- list()
for (var_name in var_names) {
  medians1[[var_name]] <- median(get(var_name), data=cluster1)
}

medians1
```

```
$Participant
[1] 118

$AmountWeek
[1] 15

$AmountOutMonth
[1] 5

$MoneyCoffee
[1] 20

$MoneyGroceries
[1] 200

$Machine
[1] "Espresso machine"

$BrandChange
[1] "Sometimes"

$PurchaseLocation
[1] "The supermarket"

$`Supermarket_Positive_ Reasons`
[1] "I do not purchase coffee from the supermarket"

$`Supermarket_Negative_ Reasons`
[1] "No reason"

$Criteria_Type_Coffee
[1] "Price, Arabica or Robusta"

$KnowledgeCoffee
[1] 6

$Purchase_Price
[1] 3
```

```
$Purchase_Sustainability
[1] 3

$Purchase_Certificate
[1] 3

$Purchase_Fairtrade
[1] 3

$Purchase_Packaging
[1] 2

$Frequency_Specialty
[1] "Never"

$Subscription_Likely
[1] 3

$Subscription_Not_Likely
[1] "I am happy with my coffee now, The price"

$App_Likely
[1] 4

$Gender
[1] "Female"

$AgeCategory
[1] "25-45"

$Occupation
[1] "Employed (Full time)"

$Education
[1] "Bachelor's degree"

$Home
[1] "Urban (City)"

$Language
[1] "Dutch"
```

Cluster 2 medians

```
var_names <- colnames(cluster2[,-ncol(cluster2)])

medians2 <- list()
for (var_name in var_names) {
  medians2[[var_name]] <- median(get(var_name), data = cluster2)
}

medians2
```

```
$Participant
[1] 118

$AmountWeek
[1] 15

$AmountOutMonth
```

```
$MoneyCoffee
[1] 20

$MoneyGroceries
[1] 200

$Machine
[1] "Espresso machine"

$BrandChange
[1] "Sometimes"

$PurchaseLocation
[1] "The supermarket"

$`Supermarket_Positive_ Reasons`
[1] "I do not purchase coffee from the supermarket"

$`Supermarket_Negative_ Reasons`
[1] "No reason"

$Criteria_Type_Coffee
[1] "Price, Arabica or Robusta"

$KnowledgeCoffee
[1] 6

$Purchase_Price
[1] 3

$Purchase_Sustainability
[1] 3

$Purchase_Certificate
[1] 3

$Purchase_Fairtrade
[1] 3

$Purchase_Packaging
[1] 2

$Frequency_Specialty
[1] "Never"

$Subscription_Likely
[1] 3

$Subscription_Not_Likely
[1] "I am happy with my coffee now, The price"

$App_Likely
[1] 4

$Gender
[1] "Female"

$AgeCategory
[1] "25-45"

$Occupation
```

```
[1] "Employed (Full time)"

$Education
[1] "Bachelor's degree"

$Home
[1] "Urban (City)"

$Language
[1] "Dutch"
```

Cluster 3 medians

```
var_names <- colnames(cluster3[,-ncol(cluster3)])

medians3 <- list()
for (var_name in var_names) {
  medians3[[var_name]] <- median(get(var_name))
}

medians3
```

```
$Participant
[1] 118

$AmountWeek
[1] 15

$AmountOutMonth
[1] 5

$MoneyCoffee
[1] 20

$MoneyGroceries
[1] 200

$Machine
[1] "Espresso machine"

$BrandChange
[1] "Sometimes"

$PurchaseLocation
[1] "The supermarket"

$`Supermarket_Positive_ Reasons`
[1] "I do not purchase coffee from the supermarket"

$`Supermarket_Negative_ Reasons`
[1] "No reason"

$Criteria_Type_Coffee
[1] "Price, Arabica or Robusta"

$KnowledgeCoffee
[1] 6

$Purchase_Price
[1] 3
```

```
$Purchase_Sustainability
[1] 3

$Purchase_Certificate
[1] 3

$Purchase_Fairtrade
[1] 3

$Purchase_Packaging
[1] 2

$Frequency_Specialty
[1] "Never"

$Subscription_Likely
[1] 3

$Subscription_Not_Likely
[1] "I am happy with my coffee now, The price"

$App_Likely
[1] 4

$Gender
[1] "Female"

$AgeCategory
[1] "25-45"

$Occupation
[1] "Employed (Full time)"

$Education
[1] "Bachelor's degree"

$Home
[1] "Urban (City)"

$Language
[1] "Dutch"
```

Appendices

Data set

Field	Description	Scales
AmountWeek	How many cups of coffee do you typically consume weekly?	Ratio, Continous
AmountOutMonth	How frequently do you drink out-of-home per month on average?	Ratio, Continous
MoneyCoffee	How much money on average do you estimate you spend on coffee per month?	Ratio, Continous
MoneyGroceries	How much on average do you spend on general groceries per month?	Ratio, Continous
Machine	How do you brew your coffee at home?	Nominal
Brand change	How often do you switch between coffee brands?	Nominal
Purchase location	Where do you usually purchase your coffee?	Nominal

Supermarket_Positive_Reasons	When you purchase coffee from the supermarket what are your main reasons for doing so?	Nominal
Supermarket_Negative_Reasons	What would be reasons why you would not purchase coffee from the supermarket?	Nominal
Criteria_Type_Coffee	What are your main criteria’s or evaluation points for choosing the type of coffee?	Nominal
KnowledgeCoffee	How would you describe your knowledge level regarding coffee in general?	Ordinal. 0-10, Discrete
Purchase_Price	I believe that the ____ is important to my decision on which coffee to purchase.	Ordinal, likert 0-5
Purchase_Sustainability	I believe that the ____ is important to my decision on which coffee to purchase.	Ordinal, likert 0-5
Purchase_Sustainability	I believe that the ____ is important to my decision on which coffee to purchase.	Ordinal, likert 0-5
Purchase_Fairtrade	I believe that the ____ is important to my decision on which coffee to purchase.	Ordinal, likert 0-5
Purchase_Packaging	I believe that the ____ is important to my decision on which coffee to purchase.	Ordinal, likert 0-5
Frequency_Specialty	How often do you drink specialty coffee?	Ordinal
Subscription_Likely	How likely are you to have an online subscription for (specialty) coffee?	Ordinal 0-10, Discrete
Subscription_Not_Likely	What is the number one reasons why you would be hesitant?	Nominal
App_Likely	How likely are you to value and use an app for your online subscription?	Ordinal, 0-10, Discrete
Gender	What is your gender?	Nominal
AgeCategory	What is your age category?	Ordinal
Occupation	What is your occupational status?	Nominal
Education	What level of education have you completed?	Ordinal
Home	How would you describe the place you currently live in?	Nominal

References

https://towardsdatascience.com/clustering-analysis-in-r-using-k-means-73eca4fb7967

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