K-means cluster and optimization

Vicente

2024-11-15

# Introduction:

The aim of this R Markdown is to comprehensively show each step of the k-cluster analysis used for the survey of communicaTUM. Before pursuing the final k-means algorithm, we need to choose how many clusters (k) we want. We want the number of clusters to represent the German population the best, while still being plausible. Therefore, we need to do an optimization process in order to see, what cluster number k gives the best results. Lets go to the code

# Preparations:

Install the necessary packages and load the packages. You only need to install the packages once, therefore delete the comment # if you need to install them again.

*#install.packages(factoextra)*  
*#install.packages(cluster)*  
  
**library**(factoextra)

## Warning: package 'factoextra' was built under R version 4.3.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.3

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

**library**(cluster)

Furthermore, we need to read our survey raw data. For that, this R-programme must be in the same folder as the survey data! Important: our survey data contains some columns and rows without information sometimes. The last elements in the following line ensure that these can be ignored and the table read anyways. Those survey elements with no answer will be marked preliminary with “NA”.

df <- **read.table**("2024-12-09 neue Datensatz nach online Vorlesung.txt", header = TRUE, sep = "**\t**", fileEncoding = "Latin1", na.strings = "", fill = TRUE, quote = "")

Before doing the optimization, lets optimize the representation of our raw data further. For this, we will only take the questions we purposedly asked for the k-cluster analysis (interest, participation and delegation). Furthermore, we will remove rows with any missing values (only for the optimization!), and then scale each variable to have a mean of 0 and standard deviation of 1

This method is based on: <https://www.statology.org/k-means-clustering-in-r/>

*# Specify the columns you want to keep (replace these with the indices of the desired columns)*  
selected\_columns <- **c**(10, 11, 12)  
  
*# Subset the data frame to include only the specified columns. Only run once!!*  
df <- df[, selected\_columns, drop = FALSE]  
  
*#remove rows with missing values*  
df <- **na.omit**(df)  
  
*#scale each variable to have a mean of 0 and sd of 1*  
df <- **scale**(df)  
  
*#view first six rows of dataset*  
**head**(df)

## Ich.interessiere.mich.für.Wissenschaft.und.Technologie.  
## 1 -1.2307311  
## 2 0.5212871  
## 3 -1.2307311  
## 4 -1.2307311  
## 5 0.5212871  
## 6 0.5212871  
## In.Bereichen..in.denen.Forschung.und.Entwicklung..F.E..einen.bedeutenden.sozialen.Einfluss.haben..sollte.die.breite.Öffentlichkeit.in.irgendeiner.Form.an.den.Entscheidungsprozessen.über.F.E.beteiligt.werden.  
## 1 0.2541914  
## 2 0.2541914  
## 3 -1.0012930  
## 4 0.2541914  
## 5 0.2541914  
## 6 1.5096758  
## Entscheidungen.über.die.Richtungen.von.F.E.sowie.Wissenschaft.und.Technologie.sollten.von.gut.ausgebildeten.und.erfahrenen.Experten.mit.viel.Vorwissen.getroffen.werden.  
## 1 -0.6523842  
## 2 -0.6523842  
## 3 0.7950933  
## 4 0.7950933  
## 5 0.7950933  
## 6 0.7950933

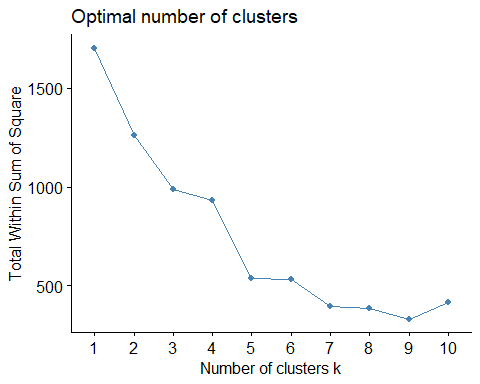
# Optimization

We will use two different plots to decide, what amount of clusters is the most optimal!

## Number of Clusters vs. Total Within Sum of Squares

The first plot is Number of Clusters vs. the Total Within Sum of Squares. Here, we typically look for an elbow, where the sum of squares begins to level off. This means, that this cluster number is optimal.

**fviz\_nbclust**(df, kmeans, method = "wss")

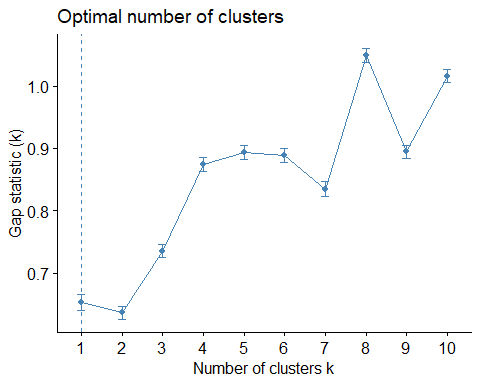


It seems that for this dataset the total sum of squares is lowest at 9 clusters. Alternatively, 5 clusters looks very good.

## Number of Clusters vs. Gap Statistic

The second plot is Number of Clusters vs. Gap Statistic. The gap statistic compares the total intra-cluster variation for different values of k with their expected values for a distribution with no clustering.

*#calculate gap statistic based on number of clusters*  
gap\_stat <- **clusGap**(df,  
 FUN = kmeans,  
 nstart = 25,  
 K.max = 10,  
 B = 50)  
  
*#plot number of clusters vs. gap statistic*  
**fviz\_gap\_stat**(gap\_stat)



In this plot we are looking for the highest possible gap statistic, which would be in this case at k = 8. Nonetheless, if we want to lower our amount of clusters, we should artificially look for local maxima, which in our case would be at k = 5.

FYI: the gap statistic is a measure based on the following publication (cf. Zotero): Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the numbers of clusters in a data set via the gap statistic. J. R. Statist. Soc. B, 63(2): 411-423.

Taking both diagrams into consideration, it would seem that k = 5 is the best cluster number. In comparison to Okamura 2016, k = 3 should NOT be picked as optimal cluster number.

# k-means cluster analysis

Now to the real k-cluster analysis based on the algorithm of Okamura 2016.

*# Load necessary packages*  
**library**(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

**library**(cluster)  
**library**(stats)

You only need to load the packages once. Now we will load our data again and forget about the whole processing from the steps before. We will use new processing steps for our raw data.

*# Read table again and forget about the editting from before.*   
  
df <- **read.table**("2024-12-09 neue Datensatz nach online Vorlesung.txt", header = TRUE, sep = "**\t**", fileEncoding = "Latin1", na.strings = "", fill = TRUE, quote = "")  
  
*# Get the number of rows and delete the extra row*  
n <- **nrow**(df) **-**1  
  
*# List of variables of interest. Use the indices of the columns to select these.*   
dim\_vars <- **c**(10, 11, 12)  
  
*# Rename the selected column names for simplicity*  
**colnames**(df)[**c**(10, 11, 12)] <- **c**("interest", "participation", "delegation")  
  
*#Remove rows with missing values in the selected columns dim\_vars*  
df <- df[**complete.cases**(df[dim\_vars]), ]

The real deal comes now. This is the algorithm for the k-cluster analysis, and the parameters are described below. We will use k = 4 first of all, and then k = 7.

*# Perform k-means clustering with 5 clusters (nstart) and repeating 100 times (nstart). From the 20 times, only the best result will be shown. Okamura does it with 10 times, but I have increased it to 100 because the results seem to converge at that point. Setting the seed number allows for reproducible calculation in all of the runs. The number does not matter, only the fact that it should be the same in all of the runs (I could also write 200 and it wouldnt matter).*  
  
**set.seed**(564)  
kmeans\_result <- **kmeans**(df[dim\_vars], centers = 5, nstart = 200)  
  
*# Add the cluster assignments to the data frame*  
df**$**c3 <- kmeans\_result**$**cluster  
  
*# List the relevant variables from the n-th row to the last row (as in Stata)*  
*# Use tail to select last n rows*  
selected\_rows <- **tail**(df, n = **nrow**(df) **-** (n **-** 1))  
**print**(selected\_rows[dim\_vars])

## [1] interest participation delegation   
## <0 rows> (or 0-length row.names)

*# Drop the selected rows from the data frame*  
df <- df[1**:**(n **-** 1), ]  
  
*# Create a new column 'cnum' and assign cluster numbers based on conditions*  
df**$**cnum <- 0  
df**$**cnum[df**$**c3 **==** 2] <- 1 *# Cluster 1*  
df**$**cnum[df**$**c3 **==** 1] <- 2 *# Cluster 2*  
df**$**cnum[df**$**c3 **==** 3] <- 3 *# Cluster 3*  
df**$**cnum[df**$**c3 **==** 4] <- 4 *# Cluster 4*  
df**$**cnum[df**$**c3 **==** 5] <- 5 *# Residual*  
df**$**cnum[**is.na**(df**$**c3)] <- 5 *# Residual for missing values*  
  
*# Calculate and display summary statistics by 'cnum'*  
summary\_table <- df **%>%**  
 **group\_by**(cnum) **%>%**  
 **summarise**(  
 count = **n**(),  
 **across**(**all\_of**(dim\_vars), **list**(mean = mean, sd = sd), .names = "{col}\_{fn}")  
 )  
  
*# Print summary to the console*  
**print**(summary\_table)

## # A tibble: 5 × 8  
## cnum count interest\_mean interest\_sd participation\_mean participation\_sd  
## <dbl> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 1 25 1.84 0.374 2.92 0.640  
## 2 2 72 3.78 0.419 1.88 0.333  
## 3 3 170 3.72 0.451 3.19 0.397  
## 4 4 105 3.87 0.342 1.77 0.422  
## 5 5 196 3.81 0.392 3.33 0.470  
## # ℹ 2 more variables: delegation\_mean <dbl>, delegation\_sd <dbl>

*# Display summary table in Word document using kable*  
knitr**::kable**(summary\_table, caption = "Summary Statistics by Cluster")

*Summary Statistics by Cluster*

| cnum | count | interest\_mean | interest\_sd | participation\_mean | participation\_sd | delegation\_mean | delegation\_sd |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 25 | 1.840000 | 0.3741657 | 2.920000 | 0.6403124 | 3.240000 | 0.7788881 |
| 2 | 72 | 3.777778 | 0.4186572 | 1.875000 | 0.3330398 | 2.777778 | 0.5366097 |
| 3 | 170 | 3.717647 | 0.4514740 | 3.194118 | 0.3966883 | 2.794118 | 0.4728994 |
| 4 | 105 | 3.866667 | 0.3415650 | 1.771429 | 0.4219265 | 4.000000 | 0.0000000 |
| 5 | 196 | 3.811224 | 0.3923323 | 3.326531 | 0.4701448 | 4.000000 | 0.0000000 |

*# Export the summary statistics to a .txt file*  
**write.table**(summary\_table, file = "2024-12-09 cluster\_summary\_after\_online\_lecture.txt", sep = "**\t**", row.names = FALSE, col.names = TRUE, quote = FALSE)

Now lastly, we will assign each person of our raw data their corresponding cluster number. These values will be imprinted into a CSV and TXT format.

*# Exclude the column named "c3" cause it doesn`t mean anything. Only include the column cnum which represents the cluster number of each person.*   
df\_excluded <- **subset**(df, select = **-**c3)  
  
*# Write the dataframe to a CSV file*  
**write.csv**(df\_excluded, "2024-12-09 cluster analysis after online lecture.csv", row.names = FALSE)  
  
*# Write the dataframe to a TXT file with tab-separated values*  
**write.table**(df\_excluded, "2024-12-09 cluster analysis after online lectur.txt", sep = "**\t**", row.names = FALSE, col.names = TRUE)

In order to open the data as a nice Excel table, please take the txt document and open it in Excel. You will see that Excel already recognizes it and organizes the information in column and rows properly.