TR STUDENT EVALUATION

Gizem Guleli

2022-03-01

Turkey Student Evaluation

Problem

The problem statement is a typical unsupervised learning problem, where with a given dataset we need to find patterns or groupings in the data without any labeled output variable.

About Dataset

In this case, the dataset consists of feedback from students who attended multiple courses at Gazi University, Ankara. Each feedback consists of evaluation questions and various other attributes, such as attendance, difficulty. The series of questions which includes course structure**,** level and quality of delivery, clarity of course objectives, course difficulty, course impact on student's overall college experience and goals, course relevance, and aspects such as willingness and ability, and preferences are answered by the students. There are 28 questions, each answered from 1 (very bad) to 5 (very good).

Attribute Information:

instr: Instructor's identifier; values taken from {1,2,3}

class: Course code (descriptor); values taken from {1-13}

repeat: Number of times the student is taking this course; values taken from {0,1,2,3}

attendance: Code of the level of attendance; values from {0, 1, 2, 3, 4}

difficulty: Level of difficulty of the course as perceived by the student; values taken from {1,2,3,4,5}

Q1: The semester course content, teaching method and evaluation system were provided at the start.
Q2: The course aims and objectives were clearly stated at the beginning of the period.
Q3: The course was worth the amount of credit assigned to it.
Q4: The course was taught according to the syllabus announced on the first day of class.
Q5: The class discussions, homework assignments, applications and studies were satisfactory.
Q6: The textbook and other courses resources were sufficient and up to date.
Q7: The course allowed field work, applications, laboratory, discussion and other studies.

Q8: The quizzes, assignments, projects and exams contributed to helping the learning.
Q9: I greatly enjoyed the class and was eager to actively participate during the lectures.
Q10: My initial expectations about the course were met at the end of the period or year.
Q11: The course was relevant and beneficial to my professional development.
Q12: The course helped me look at life and the world with a new perspective.
Q13: The Instructor's knowledge was relevant and up to date.
Q14: The Instructor came prepared for classes.
Q15: The Instructor taught in accordance with the announced lesson plan.
Q16: The Instructor was committed to the course and was understandable.
Q17: The Instructor arrived on time for classes.
Q18: The Instructor has a smooth and easy to follow delivery/speech.
Q19: The Instructor made effective use of class hours.
Q20: The Instructor explained the course and was eager to be helpful to students.
Q21: The Instructor demonstrated a positive approach to students.
Q22: The Instructor was open and respectful of the views of students about the course.
Q23: The Instructor encouraged participation in the course.
Q24: The Instructor gave relevant homework assignments/projects, and helped/guided students.
Q25: The Instructor responded to questions about the course inside and outside of the course.
Q26: The Instructor's evaluation system (midterm and final questions, projects, assignments, etc.) effectively measured the course objectives.
Q27: The Instructor provided solutions to exams and discussed them with students.
Q28: The Instructor treated all students in a right and objective manner.

Q1-Q28 are all Likert-type, meaning that the values are taken from {1,2,3,4,5}

Analyze Data-set

Preliminary

First, we downloaded and read the required libraries to analyse and visualise the data-set. Then we read the data-set and checked missing values.It appears that the data set contains no missing values and all attributes are numeric. This is a good indication that the data is relatively clean and does not require any preprocessing.
Therefore, it is always a good idea to examine the data carefully and perform exploratory data analysis (EDA) to gain a better understanding of the data, identify potential problems and make informed decisions about pre-processing, modelling and analysing the data.

```
library(cluster)
library(factoextra)
library(flexclust)
library(fpc)
library(ClusterR)
library(rstatix)
library(ggpubr)
library(dplyr)
```

```
library(ggplot2)
library(tidyr)
library(reshape)
library(gridExtra)
library(readr)
library(ggplot2)
library(cowplot)
```

```
##Data Loading
```

```
trstudent <- read_csv("turkiye-student-evaluation_R_Specific.csv")
```

```
## Rows: 5820 Columns: 34
## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## dbl (34): Idnum, instr, class, nb.repeat, attendance, difficulty, Q1, Q2,
Q3...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
#view and Check first and last 6 obs of dataset to be sure that it readed
Clearly
```

```
head(trstudent)
```

```
## # A tibble: 6 x 34
##   Idnum instr class nb.rep~1 atten~2 diffi~3    Q1    Q2    Q3    Q4    Q5
Q6
##   <dbl> <dbl> <dbl>    <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
<dbl>
## 1     1     1     2        1       0       4     3     3     3     3     3
3
## 2     2     1     2        1       1       3     3     3     3     3     3
3
## 3     3     1     2        1       2       4     5     5     5     5     5
5
## 4     4     1     2        1       1       3     3     3     3     3     3
3
## 5     5     1     2        1       0       1     1     1     1     1     1
1
## 6     6     1     2        1       3       3     4     4     4     4     4
4
## # ... with 22 more variables: Q7 <dbl>, Q8 <dbl>, Q9 <dbl>, Q10 <dbl>,
## #   Q11 <dbl>, Q12 <dbl>, Q13 <dbl>, Q14 <dbl>, Q15 <dbl>, Q16 <dbl>,
## #   Q17 <dbl>, Q18 <dbl>, Q19 <dbl>, Q20 <dbl>, Q21 <dbl>, Q22 <dbl>,
## #   Q23 <dbl>, Q24 <dbl>, Q25 <dbl>, Q26 <dbl>, Q27 <dbl>, Q28 <dbl>, and
## #   abbreviated variable names 1: nb.repeat, 2: attendance, 3: difficulty
```

```
tail(trstudent)
```

```
## # A tibble: 6 x 34
##   Idnum instr class nb.rep~1 atten~2 diffi~3    Q1    Q2    Q3    Q4    Q5
Q6
##   <dbl> <dbl> <dbl>    <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
<dbl>
## 1  5815     3    13        1       2       4     1     1     1     1     1
1
## 2  5816     3    13        1       0       1     1     1     1     1     1
1
## 3  5817     3    13        1       3       4     4     4     4     4     4
4
## 4  5818     3    13        1       0       4     5     5     5     5     5
5
## 5  5819     3    13        1       1       2     1     1     1     1     1
1
## 6  5820     3    13        1       1       2     1     1     1     1     1
1
## # ... with 22 more variables: Q7 <dbl>, Q8 <dbl>, Q9 <dbl>, Q10 <dbl>,
## #   Q11 <dbl>, Q12 <dbl>, Q13 <dbl>, Q14 <dbl>, Q15 <dbl>, Q16 <dbl>,
## #   Q17 <dbl>, Q18 <dbl>, Q19 <dbl>, Q20 <dbl>, Q21 <dbl>, Q22 <dbl>,
## #   Q23 <dbl>, Q24 <dbl>, Q25 <dbl>, Q26 <dbl>, Q27 <dbl>, Q28 <dbl>, and
## #   abbreviated variable names 1: nb.repeat, 2: attendance, 3: difficulty
```

```
###Change the names of variables to make more readable

colnames(trstudent)[colnames(trstudent)=="instr"] <- "instructor"
colnames(trstudent)[colnames(trstudent)=="class"] <- "course"
colnames(trstudent)[colnames(trstudent)=="nb.repeat"] <- "repeat"

##Empty value controls

trstudent[!complete.cases(trstudent),]
```

```
## # A tibble: 0 x 34
## # ... with 34 variables: Idnum <dbl>, instructor <dbl>, course <dbl>,
## #   repeat <dbl>, attendance <dbl>, difficulty <dbl>, Q1 <dbl>, Q2 <dbl>,
## #   Q3 <dbl>, Q4 <dbl>, Q5 <dbl>, Q6 <dbl>, Q7 <dbl>, Q8 <dbl>, Q9 <dbl>,
## #   Q10 <dbl>, Q11 <dbl>, Q12 <dbl>, Q13 <dbl>, Q14 <dbl>, Q15 <dbl>,
## #   Q16 <dbl>, Q17 <dbl>, Q18 <dbl>, Q19 <dbl>, Q20 <dbl>, Q21 <dbl>,
## #   Q22 <dbl>, Q23 <dbl>, Q24 <dbl>, Q25 <dbl>, Q26 <dbl>, Q27 <dbl>, Q28
<dbl>
```

```
colSums(is.na(trstudent))
```

```
##      Idnum instructor     course     repeat attendance difficulty
Q1
##          0          0          0          0          0          0
0
##         Q2         Q3         Q4         Q5         Q6         Q7
```

```
Q8
##          0          0          0          0          0          0
0
##         Q9        Q10        Q11        Q12        Q13        Q14
Q15
##          0          0          0          0          0          0
0
##        Q16        Q17        Q18        Q19        Q20        Q21
Q22
##          0          0          0          0          0          0
0
##        Q23        Q24        Q25        Q26        Q27        Q28
##          0          0          0          0          0          0

attach(trstudent)

###Check changes and last version
head(trstudent)

## # A tibble: 6 x 34
##   Idnum instructor course `repeat` atten~1 diffi~2    Q1    Q2    Q3    Q4
Q5
##   <dbl>      <dbl>  <dbl>    <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
<dbl>
## 1     1          1      2        1       0       4     3     3     3     3
3
## 2     2          1      2        1       1       3     3     3     3     3
3
## 3     3          1      2        1       2       4     5     5     5     5
5
## 4     4          1      2        1       1       3     3     3     3     3
3
## 5     5          1      2        1       0       1     1     1     1     1
1
## 6     6          1      2        1       3       3     4     4     4     4
4
## # ... with 23 more variables: Q6 <dbl>, Q7 <dbl>, Q8 <dbl>, Q9 <dbl>,
## #   Q10 <dbl>, Q11 <dbl>, Q12 <dbl>, Q13 <dbl>, Q14 <dbl>, Q15 <dbl>,
## #   Q16 <dbl>, Q17 <dbl>, Q18 <dbl>, Q19 <dbl>, Q20 <dbl>, Q21 <dbl>,
## #   Q22 <dbl>, Q23 <dbl>, Q24 <dbl>, Q25 <dbl>, Q26 <dbl>, Q27 <dbl>,
## #   Q28 <dbl>, and abbreviated variable names 1: attendance, 2: difficulty
```

Exploratory Data Analysis

The Distribution of Instructors graph shows that most of the courses are given by Instructor 3 and distribution is too skewed left .

The Distribution of Courses shows that course 3 and course 13 is the most taken courses out of 13 courses.

The Distribution of Repeating histogram shows that the majority of students (%84) is repeated the course only once while minority (%16) repeat the classes for the second or third time. However, this may somewhat complicate our plan to create an interpretable, acceptable classifier because the distribution is too skewed right.

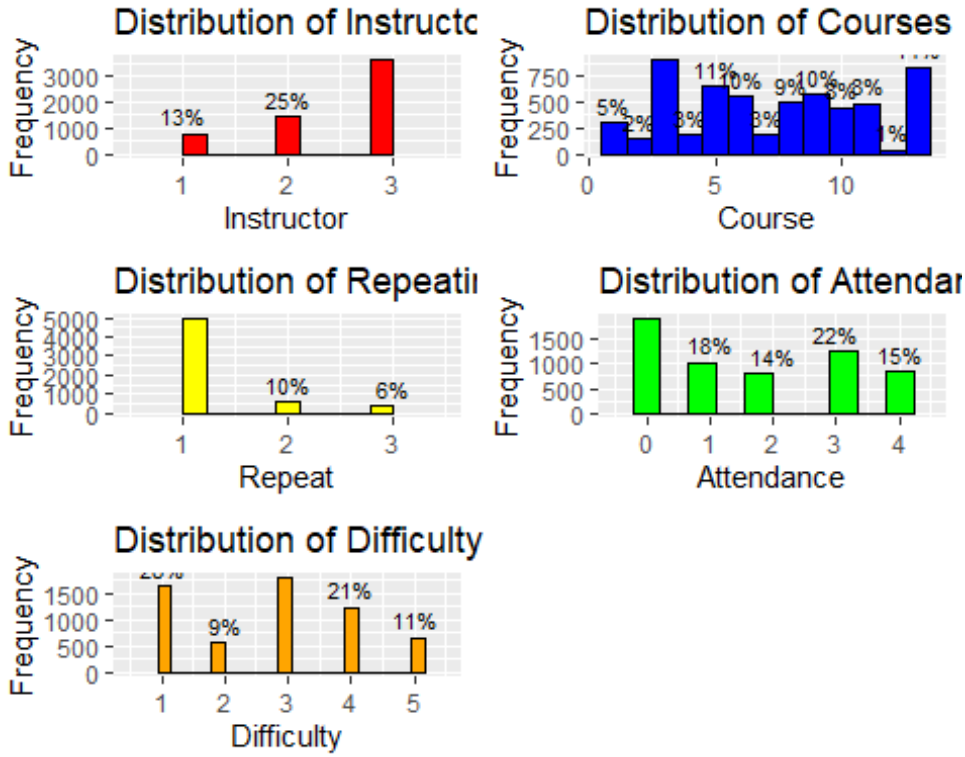
The Distribution of Attendance histogram shows that the majority of students' the attendance level of the course is weak, with a peak at 0 level and 65% of student attendant lesson less then 3 level. This suggests that most students didn't attended class regularly.

The difficulty_hist histogram shows that the difficulty level of the course was more evenly distributed, with peaks at 3 on the scale. This suggests that some students found the course relatively easy, while others found it more challenging.

```
# Create histograms of the Instructor, Class, Repeat, Attendance and
Difficulty  variables

ins_hist <- ggplot(trstudent, aes(x = `instructor`)) +
  geom_histogram(color = "black", fill = "red", bins = 10) +
  stat_bin(aes(label = paste0(round((..count../nrow(trstudent))*100), "%")),
geom = "text", vjust = -0.5, color = "black", size = 3, bins = 3) +
  labs(x = "Instructor", y = "Frequency", title = "Distribution of
Instructors")


course_hist <- ggplot(trstudent, aes(x = `course`)) +
  geom_histogram(color = "black", fill = "blue", bins = 13) +
  stat_bin(aes(label = paste0(round((..count../nrow(trstudent))*100), "%")),
geom = "text", vjust = -0.5, color = "black", size = 3, bins = 13) +
  labs(x = "Course", y = "Frequency", title = "Distribution of Courses")


rep_hist <- ggplot(trstudent, aes(x = `repeat`)) +
  geom_histogram(color = "black", fill = "yellow", bins = 10) +
  stat_bin(aes(label = paste0(round((..count../nrow(trstudent))*100), "%")),
geom = "text", vjust = -0.5, color = "black", size = 3, bins = 3) +
  labs(x = "Repeat", y = "Frequency", title = "Distribution of Repeating")


att_hist <- ggplot(trstudent, aes(x = attendance)) +
  geom_histogram(color = "black", fill = "green", bins = 10) +
  stat_bin(aes(label = paste0(round((..count../nrow(trstudent))*100), "%")),
geom = "text", vjust = -0.5, color = "black", size = 3, bins = 5) +
  labs(x = "Attendance", y = "Frequency", title = "Distribution of
Attendance")

dif_hist <- ggplot(trstudent, aes(x = difficulty)) +
  geom_histogram(color = "black", fill = "orange", bins = 20) +
  stat_bin(aes(label = paste0(round((..count../nrow(trstudent))*100), "%")),
geom = "text", vjust = -0.5, color = "black", size = 3, bins = 5) +
```

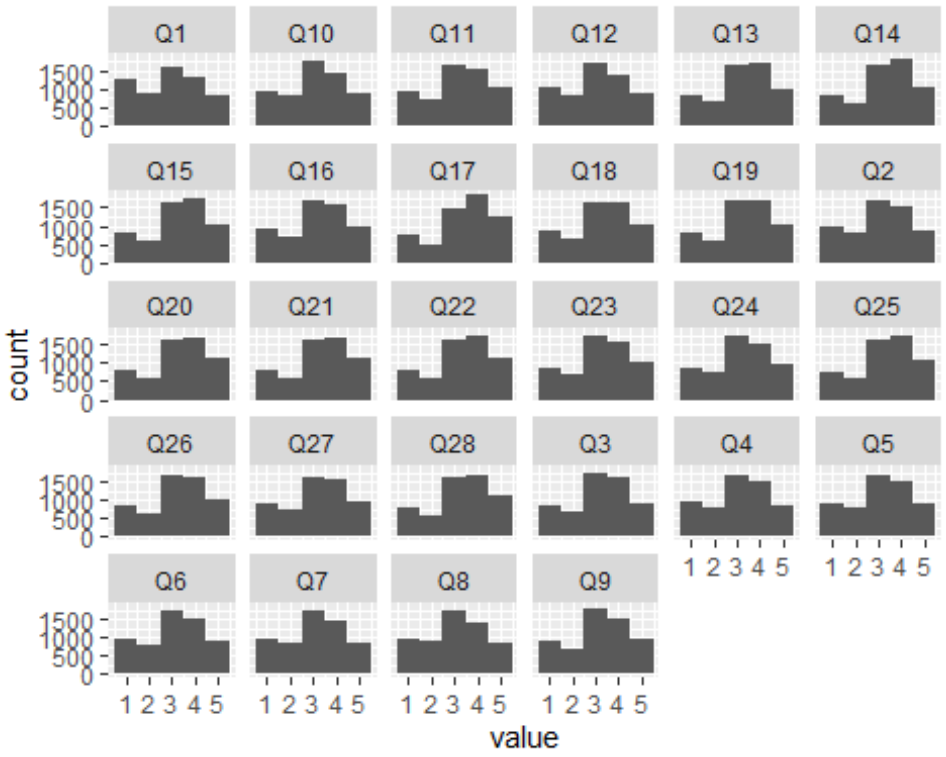
```
  labs(x = "Difficulty", y = "Frequency", title = "Distribution of
Difficulty")


grid.arrange(ins_hist, course_hist, rep_hist, att_hist,dif_hist)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2
3.4.0.
## i Please use `after_stat(count)` instead.
```

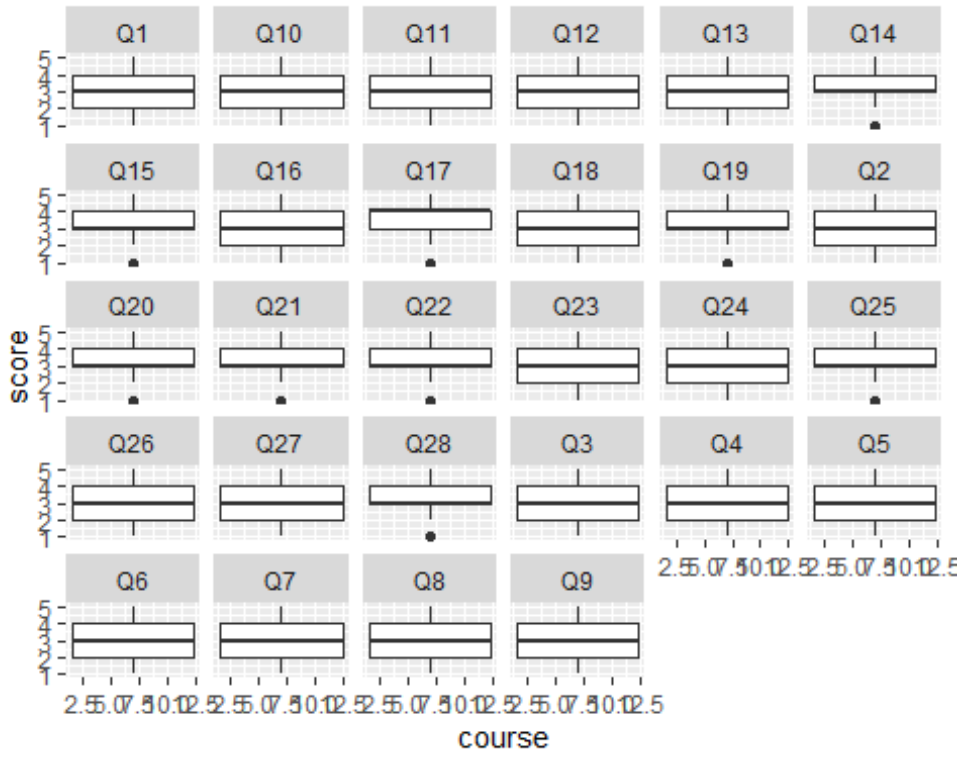


When we check the distribution of evaluation question, most of them seems similar and kind of hard to read histogram graphs and need to look detailiy. From boxplot its more clear to see that some questions (#14,15,17,19,20,21,21,25,28) are higly rated even there is a some outliers which means few students gave less rate while other questions seems more normally distributed.

```
# Plot a histogram of the scores for each question
trstudent %>%
  select(starts_with("Q")) %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_histogram(bins = 5) +
  facet_wrap(~key, nrow = 5)
```




```
# Plot a boxplot of the scores for each question by course
trstudent %>%
  select(starts_with("Q"), course) %>%
  gather(key = "question", value = "score", starts_with("Q")) %>%
  ggplot(aes(course, score)) +
  geom_boxplot() +
  facet_wrap(~question, nrow = 5)

## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```



Dimension Reduction

PCA

Here we also used to PCA analysis to identify the key features that differentiate the groups of students. By computing the principal components of the survey data, we can identify the survey questions that have the highest impact on the clustering results.

We are not able to imagine 28 Dimension and thanks to PCA, we can reduce the columns from 28D to 2D. Therefore, we would able to plot the clustering results based on the first two principal components and visually inspect how well the clusters are separated in this 2D space. We also analyse the loadings of each survey question on the principal components to see which questions are most important in differentiating the clusters.

There are many ways to compute the principal components, but I used here the prcomp() function, which uses single value decomposition. We are standardizing datasets with scale() function and subset data which includes only evaluation questions to be able to focus clustering the question. As it can be seen below (summary of pca1) we started to get the ability to explain %82 of the variance in the first component and being able to catch 86% with second components out of 28.

```
## Subset the dataset to include only the evaluation questions
subset_data <- trstudent[, 7:34]

# Scale the data to normalize the variables
scaled_subset <- scale(subset_data)
```

```r
# Perform PCA analysis
pca1<-prcomp(scaled_subset, center=FALSE, scale.=FALSE) # stats::
pca1$rotation
```

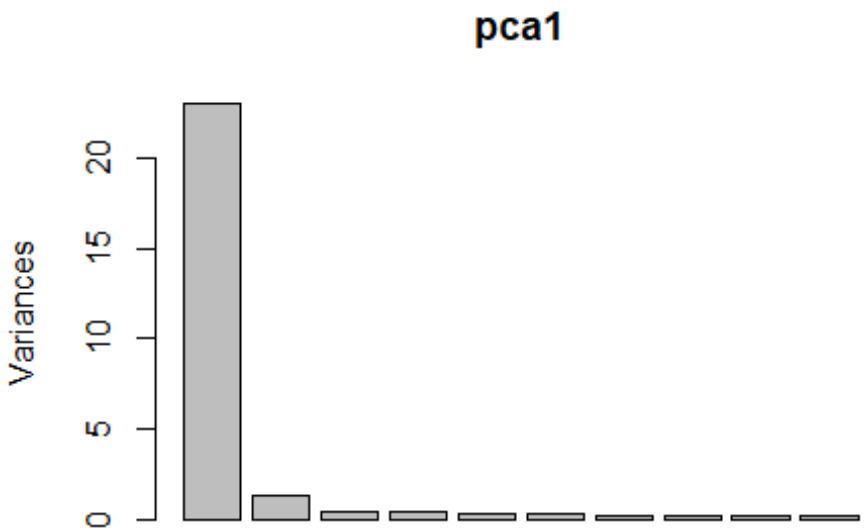
```
##            PC1         PC2          PC3           PC4         PC5
PC6
## Q1  -0.1697760  0.33713170  0.471473561 -0.0002795052  0.16789340 -
0.391976233
## Q2  -0.1855459  0.23299469  0.320261243  0.1337424999  0.09993407 -
0.124545901
## Q3  -0.1855657  0.12218837  0.146745386  0.3375728017  0.12387154
0.250839075
## Q4  -0.1828628  0.24638813  0.350488200  0.0887976940  0.04793488
0.008472256
## Q5  -0.1897697  0.21209935  0.069979403 -0.0419317017 -0.19426406
0.229990136
## Q6  -0.1863937  0.20590369 -0.040595856  0.0164418067 -0.22103196
0.435444498
## Q7  -0.1873440  0.24852272 -0.108392162 -0.1129792977 -0.15397498
0.293105123
## Q8  -0.1856411  0.25359638 -0.163426840 -0.1650507236 -0.08455212
0.163958733
## Q9  -0.1834801  0.13550861 -0.318487322  0.2416205977  0.15387529 -
0.029681295
## Q10 -0.1924670  0.19507424 -0.213190367 -0.0491348479 -0.02801110 -
0.002263101
## Q11 -0.1839239  0.11363151 -0.434748466  0.2813327918  0.24821857 -
0.163980695
## Q12 -0.1818928  0.21147301 -0.368113415 -0.0392384262  0.13393583 -
0.387521373
## Q13 -0.1943247 -0.10514696 -0.005489179  0.0613283743 -0.33290466 -
0.161818110
## Q14 -0.1946421 -0.15731412  0.012123529  0.1382651366 -0.28888050 -
0.082375406
## Q15 -0.1940270 -0.15680234  0.039666620  0.1363431909 -0.27413330 -
0.064754849
## Q16 -0.1946208 -0.04495300 -0.020024137 -0.1770699380 -0.33348456 -
0.213256116
## Q17 -0.1824796 -0.26392128  0.033671292  0.3908610157 -0.01244443
0.089144854
## Q18 -0.1932407 -0.12622174  0.005914380 -0.0636790696 -0.28031343 -
0.216281648
## Q19 -0.1941508 -0.15255750 -0.002866990  0.0071990823 -0.06683799 -
0.125770835
## Q20 -0.1933655 -0.19503577  0.037688686  0.0490506431  0.03965443 -
0.016093191
## Q21 -0.1923313 -0.21999740  0.028166298  0.0660028922  0.16654673
0.009639091
## Q22 -0.1923365 -0.22370310  0.031765736  0.0504349937  0.17913762
```

```
0.055910897
## Q23 -0.1955702 -0.10053279  0.033839640 -0.2689834369  0.05836168 -
0.037571429
## Q24 -0.1933136 -0.05910570  0.018477290 -0.3652075785  0.05801454 -
0.041502156
## Q25 -0.1920408 -0.20985757  0.046772985 -0.0149839809  0.17382759
0.128001061
## Q26 -0.1918982 -0.11817539  0.002900408 -0.2347670430  0.17537498
0.144482725
## Q27 -0.1875538 -0.06799767 -0.009727468 -0.4203269121  0.27131468
0.038242870
## Q28 -0.1885680 -0.21196677  0.063563225  0.0022150270  0.23690903
0.189954854
##               PC7         PC8          PC9         PC10         PC11
## Q1   0.1060292051 -0.03133953  0.235099607  0.169890785 -0.278541237
## Q2   0.0416113345 -0.04625139 -0.107979199 -0.175881304 -0.182439977
## Q3  -0.2076819502 -0.18618318 -0.692033952 -0.050500844 -0.108218981
## Q4  -0.0946145455  0.09105951  0.172699763 -0.007676282  0.637781544
## Q5  -0.0298107525  0.11196196 -0.090775253 -0.121593333  0.134651450
## Q6  -0.1786592885  0.10681952  0.264056149 -0.145603581 -0.178352692
## Q7   0.0416633915  0.14293493  0.117745478  0.179071941 -0.124067744
## Q8   0.2183156673  0.06881141 -0.033194629  0.433311577 -0.062969053
## Q9   0.6612970804 -0.29216819  0.061870901 -0.255407981  0.176443949
## Q10  0.1468429376 -0.07954183  0.006330617 -0.113788435 -0.109026050
## Q11 -0.3124960335  0.09118560 -0.006941654 -0.018948446  0.002619655
## Q12 -0.3783178148  0.13247768  0.044308242  0.101072308  0.096696071
## Q13 -0.1663624681 -0.16678090  0.125367182 -0.201859351 -0.123556326
## Q14 -0.1102656436 -0.16712379  0.123042801 -0.102517430  0.026172314
## Q15 -0.0934978200 -0.12054271  0.137533026 -0.040827311  0.129762155
## Q16  0.0627479265 -0.13574897 -0.100550528 -0.131242045  0.010856721
## Q17  0.0608795045 -0.07392349  0.077261385  0.597807502  0.157352555
## Q18  0.1184705994 -0.01533150 -0.272258205  0.198128538 -0.015635290
## Q19  0.0889237172  0.17137468 -0.153667587  0.191288234 -0.166645648
## Q20  0.0746133792  0.30535703  0.015670389 -0.029514140 -0.243520828
## Q21  0.0961802781  0.32527554  0.060957273 -0.173924307 -0.149504446
## Q22  0.1074537187  0.29156936  0.066238622 -0.157537373 -0.065052815
## Q23  0.0707318910  0.22880310 -0.156330620 -0.135944199  0.150954792
## Q24  0.0003464241  0.14898839 -0.236552365 -0.036978627  0.311203466
## Q25  0.0304324780 -0.03470870  0.077479961  0.007591976  0.191080252
## Q26 -0.1134244437 -0.30152317 -0.010270550 -0.002932152 -0.020269040
## Q27 -0.1098364380 -0.43740723  0.058422537  0.112954644 -0.114193917
## Q28 -0.1330290653 -0.14638741  0.252318265 -0.032351488 -0.088392358
##              PC12        PC13        PC14          PC15         PC16
## Q1  -0.245404446  0.10961602  0.11501983  0.0448910815  0.122680747
## Q2   0.015684845 -0.35432002  0.14196996  0.0138131596 -0.302563485
## Q3  -0.102277848  0.08349969 -0.15043313 -0.1953456598  0.133138905
## Q4   0.354460522  0.15240607 -0.15290707  0.0432772551  0.121573584
## Q5   0.156671378  0.15717816 -0.09702472  0.2549553606 -0.242846985
## Q6   0.113274733 -0.20666516  0.42216832 -0.2692310900  0.126124670
## Q7  -0.109581273  0.01961407 -0.05591420 -0.0710635350  0.138476555
```

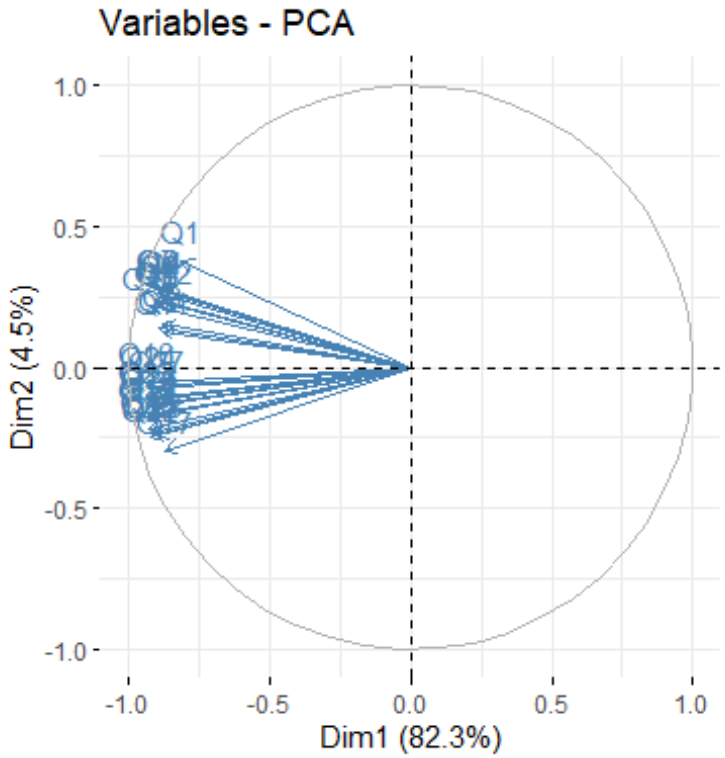
```
## Q8  -0.242011450  0.13646748 -0.27726847  0.1042774421 -0.206985799
## Q9   0.062631169 -0.04788857 -0.01416792 -0.1499423018  0.113231466
## Q10  0.038342395 -0.03516827 -0.07572621  0.1883638970  0.025381050
## Q11 -0.032323653  0.31441360  0.37732181  0.3621283116 -0.226761663
## Q12  0.041398243 -0.36247978 -0.28586877 -0.3270517485  0.150020280
## Q13 -0.188831969  0.16192711 -0.28435187 -0.0501898426  0.057206542
## Q14 -0.192516465  0.16774132 -0.15555153 -0.0126771876  0.077066000
## Q15 -0.137199712  0.06913215  0.16525842 -0.0003262484  0.025744927
## Q16  0.100328105 -0.14198859  0.16549411  0.0440988618 -0.208418088
## Q17 -0.030663106 -0.14739683  0.19250633 -0.0970412040 -0.005332180
## Q18  0.261714428 -0.06518288  0.06682525 -0.0175084245 -0.088203611
## Q19  0.393746059 -0.03657278  0.04123285  0.1093637476  0.160652498
## Q20  0.156991949  0.07008462 -0.19455475  0.0921003378  0.102087828
## Q21  0.062978145  0.19251346 -0.08995346 -0.0699877613  0.100923823
## Q22 -0.065310115  0.06049904  0.04339845 -0.1121083910 -0.001498194
## Q23 -0.182316688 -0.01581299  0.10702185 -0.1388155625  0.015942931
## Q24 -0.305229310 -0.08506121  0.16249657 -0.0303788729 -0.045247397
## Q25 -0.322294583 -0.05940847  0.03689500 -0.0072419820 -0.087587224
## Q26 -0.006389135 -0.25450244  0.02675188  0.5496358100  0.523523883
## Q27  0.295980168  0.43021675  0.12348742 -0.3571658051 -0.088338076
## Q28  0.102176171 -0.31452301 -0.34197292  0.0782094334 -0.487043452
##              PC17         PC18        PC19         PC20         PC21
## Q1  -0.3673303987 -0.139490967 -0.07416886 -0.083466490  0.052228727
## Q2   0.6091669429  0.042163918  0.07855222  0.155359233 -0.062892009
## Q3  -0.1855132714  0.106251236  0.02210160 -0.069852599  0.043042508
## Q4   0.0008949794  0.280786206  0.14493294  0.058351222 -0.025849078
## Q5   0.0233787123 -0.591786894 -0.43214698 -0.119935452  0.070872095
## Q6  -0.1632998963  0.017814018  0.07187873 -0.250995391  0.017750294
## Q7   0.0312076754  0.007095311  0.14170634  0.657800324  0.122519743
## Q8   0.1368708257  0.291619698  0.07751654 -0.355873100  0.069351094
## Q9  -0.0962689386 -0.168234283  0.13191288 -0.039485192  0.109434233
## Q10  0.0191072874  0.188538581 -0.21995856  0.054956375 -0.547369640
## Q11 -0.0779677532  0.038895006  0.15468804  0.099847725  0.046264428
## Q12  0.0713338141 -0.085296892 -0.21162512 -0.104422674  0.101137939
## Q13  0.0463726668 -0.177836745  0.13856679  0.144934114 -0.076522760
## Q14  0.1016399073 -0.066986832  0.06576002  0.005439440 -0.085605618
## Q15  0.0881889662  0.199811064  0.02285499 -0.177954398  0.008459075
## Q16 -0.1077406291  0.291620199 -0.08672601 -0.165141152  0.248740542
## Q17  0.0906724612 -0.172271906 -0.16440340 -0.037393300 -0.305208635
## Q18 -0.2993834964  0.141060187 -0.19883070  0.333316508  0.070743974
## Q19  0.0782530230 -0.267174695  0.37124399 -0.030128435  0.148785536
## Q20  0.1265180191  0.007916723  0.22492195 -0.263228935  0.019113763
## Q21  0.0191979569  0.173437842 -0.22356962  0.002468163  0.029255185
## Q22 -0.0208803493  0.164688926 -0.33036374  0.137333641 -0.081497359
## Q23 -0.0433745274 -0.109555296  0.07492887 -0.086150660 -0.210964399
## Q24 -0.1050258961 -0.157045526  0.29811966  0.024393529 -0.237409410
## Q25  0.1442997083 -0.019800268 -0.13164210  0.082415497  0.572473800
## Q26  0.0529569432  0.043927967 -0.11854189 -0.018383719  0.036933930
## Q27  0.2006464518 -0.044832540 -0.03846151 -0.005120253 -0.045541256
## Q28 -0.4029837033 -0.008594253  0.20325777  0.041463637 -0.079752465
```

```
##             PC22         PC23        PC24        PC25         PC26
PC27
## Q1   0.008105862  0.074861460  0.01850499 -0.02456875 -0.013386223
0.023209455
## Q2  -0.045693420 -0.180591910 -0.03212129 -0.03034618 -0.030891001 -
0.004686654
## Q3  -0.016146207  0.110215596  0.05104402  0.00660499  0.032163939
0.013138514
## Q4   0.078193248 -0.018647937 -0.05452317  0.11402922  0.042179074 -
0.009063972
## Q5  -0.126027950  0.012622575  0.03949373 -0.09705962 -0.008347481
0.002644128
## Q6   0.217454242 -0.171110341 -0.13944760  0.01155003 -0.027370417 -
0.036514817
## Q7  -0.326416626  0.242056065  0.06384012 -0.00636807  0.006528051
0.084331371
## Q8   0.066024197 -0.324650075  0.06750539  0.03648284  0.057639449 -
0.002862304
## Q9  -0.119765510 -0.111078268 -0.02470235 -0.04801433 -0.013674714
0.002286918
## Q10  0.413333919  0.443991863  0.04365678 -0.01782452 -0.070716633 -
0.125356436
## Q11 -0.036910115 -0.073918379 -0.06610502  0.09906692 -0.024442259
0.010487017
## Q12 -0.051288414  0.012685674  0.03936192 -0.07916080 -0.024730700
0.030875879
## Q13  0.087730159 -0.204829175 -0.12620383  0.25870720  0.285950770 -
0.433906063
## Q14  0.166304149 -0.082989715 -0.07022125 -0.06674936 -0.191521045
0.684782640
## Q15 -0.277845589  0.111263852  0.49842292 -0.41321842 -0.232201811 -
0.280582724
## Q16 -0.223579268  0.328744941 -0.11165413  0.24141480  0.387518282
0.214391011
## Q17 -0.124054009  0.092553985 -0.16987898  0.15411376  0.177541773 -
0.026722437
## Q18  0.117593656 -0.342725582 -0.18532014 -0.11988452 -0.372331950 -
0.139847958
## Q19  0.335451921  0.074252352  0.44578247  0.01467325  0.194781071
0.057808310
## Q20 -0.208281758  0.261025061 -0.46968007 -0.07224800 -0.347181280 -
0.016143596
## Q21 -0.108504001 -0.089658714 -0.06592937 -0.16398431  0.320817829 -
0.192546130
## Q22  0.063488494 -0.206508067  0.24049913  0.01729498  0.137509192
0.314032966
## Q23 -0.170830960 -0.030567398  0.22629553  0.60361907 -0.351517849 -
0.067026069
## Q24  0.061922060 -0.007488086 -0.21948731 -0.46027642  0.254644918
0.018101129
```

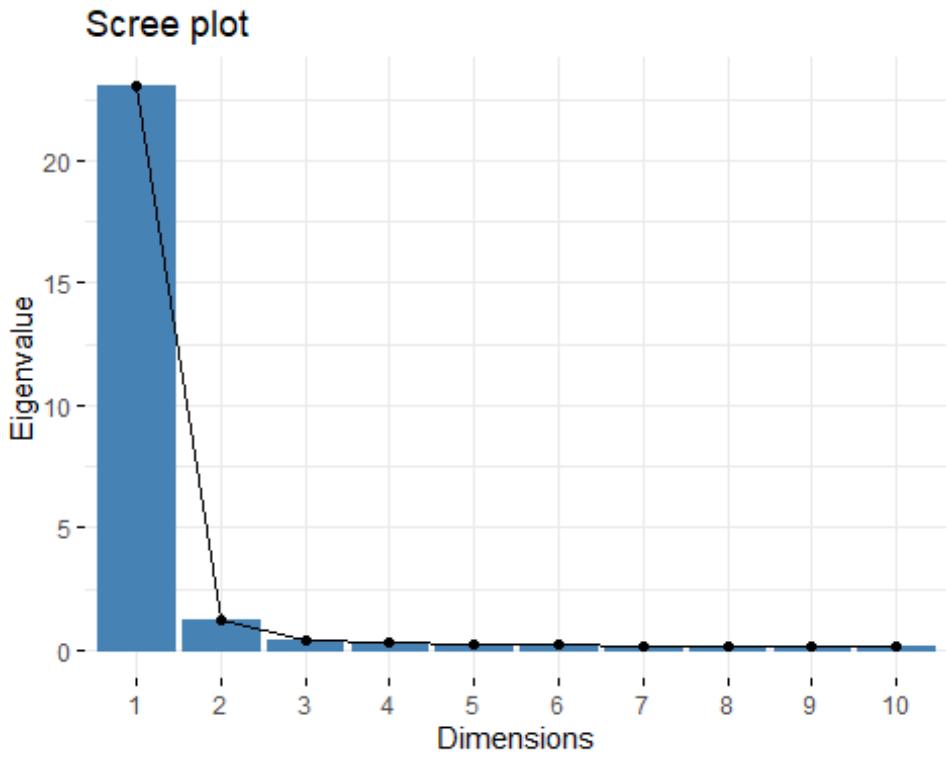
```
## Q25  0.451962126  0.262589653 -0.09455432  0.06403162 -0.149337700 -
0.164132523
## Q26 -0.148621066 -0.224361681 -0.02682280  0.04443805  0.021313355
0.017766990
## Q27 -0.032697826  0.028444103 -0.02744451 -0.03849323 -0.025015580
0.010690501
## Q28 -0.065562964  0.004818553  0.13680609 -0.02948156 -0.026190637
0.017186010
##              PC28
## Q1   2.212927e-05
## Q2  -5.612705e-02
## Q3   2.782601e-02
## Q4   4.173679e-02
## Q5   2.177299e-02
## Q6  -2.363704e-02
## Q7  -2.334679e-02
## Q8   4.394068e-03
## Q9   2.668788e-02
## Q10 -3.282851e-02
## Q11 -6.518241e-03
## Q12  9.059211e-03
## Q13  2.285663e-01
## Q14 -2.998451e-01
## Q15  6.805769e-02
## Q16  2.416302e-02
## Q17 -1.777578e-02
## Q18 -2.707419e-02
## Q19 -3.094000e-02
## Q20  2.809367e-01
## Q21 -6.113201e-01
## Q22  5.807554e-01
## Q23 -2.015831e-01
## Q24  5.971929e-02
## Q25 -1.756117e-02
## Q26  8.011565e-03
## Q27  4.174337e-02
## Q28 -7.223916e-02

plot(pca1)
```



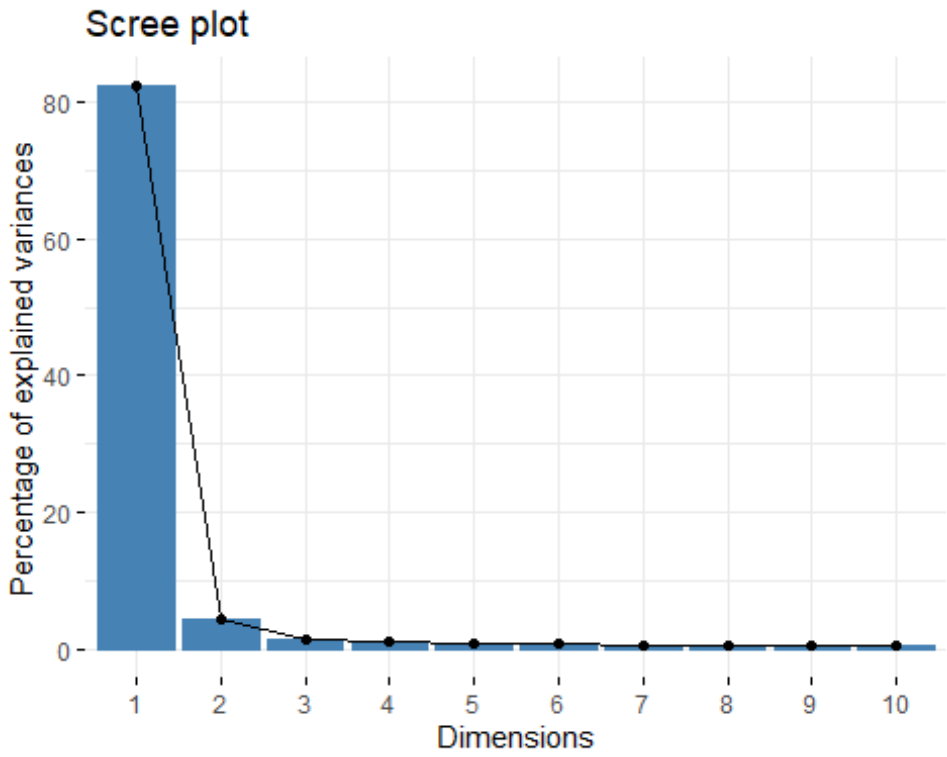

```
#visulation of PCA results
fviz_pca_var(pca1, col.var="steelblue")
```



```r
# visusalisation of quality
fviz_eig(pca1, choice='eigenvalue')
```



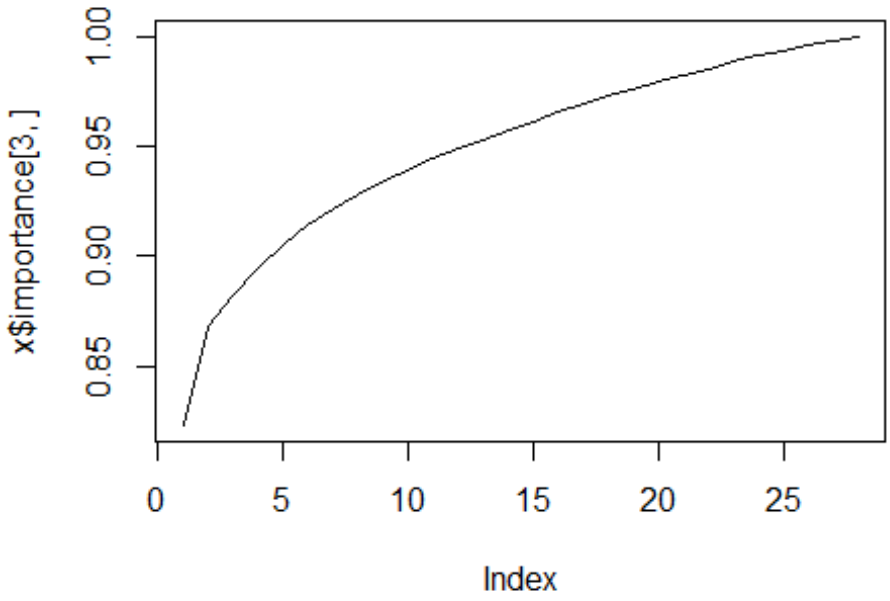

```r
fviz_eig(pca1)
```




```
# table of eigenvalues
eig.val<-get_eigenvalue(pca1)
eig.val
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1  23.04090298       82.2889392                     82.28894
## Dim.2   1.25291747        4.4747053                     86.76364
## Dim.3   0.39493764        1.4104916                     88.17414
## Dim.4   0.36086149        1.2887910                     89.46293
## Dim.5   0.28988071        1.0352883                     90.49822
## Dim.6   0.25623311        0.9151182                     91.41333
## Dim.7   0.20415776        0.7291349                     92.14247
## Dim.8   0.18326004        0.6545001                     92.79697
## Dim.9   0.17247115        0.6159684                     93.41294
## Dim.10  0.14267452        0.5095519                     93.92249
## Dim.11  0.13814680        0.4933814                     94.41587
## Dim.12  0.13693953        0.4890697                     94.90494
## Dim.13  0.11906254        0.4252234                     95.33016
## Dim.14  0.11637583        0.4156280                     95.74579
## Dim.15  0.11420619        0.4078792                     96.15367
## Dim.16  0.10969318        0.3917614                     96.54543
## Dim.17  0.10575177        0.3776849                     96.92312
## Dim.18  0.10059034        0.3592512                     97.28237
## Dim.19  0.09527980        0.3402850                     97.62265
## Dim.20  0.09276426        0.3313009                     97.95395
## Dim.21  0.08460563        0.3021630                     98.25612
## Dim.22  0.08430559        0.3010914                     98.55721
```

```
## Dim.23  0.08045549        0.2873410                    98.84455
## Dim.24  0.07736759        0.2763128                    99.12086
## Dim.25  0.07077776        0.2527777                    99.37364
## Dim.26  0.06772164        0.2418630                    99.61550
## Dim.27  0.05579613        0.1992719                    99.81477
## Dim.28  0.05186304        0.1852251                   100.00000

x<-summary(pca1)
plot(x$importance[3,],type="l")
```

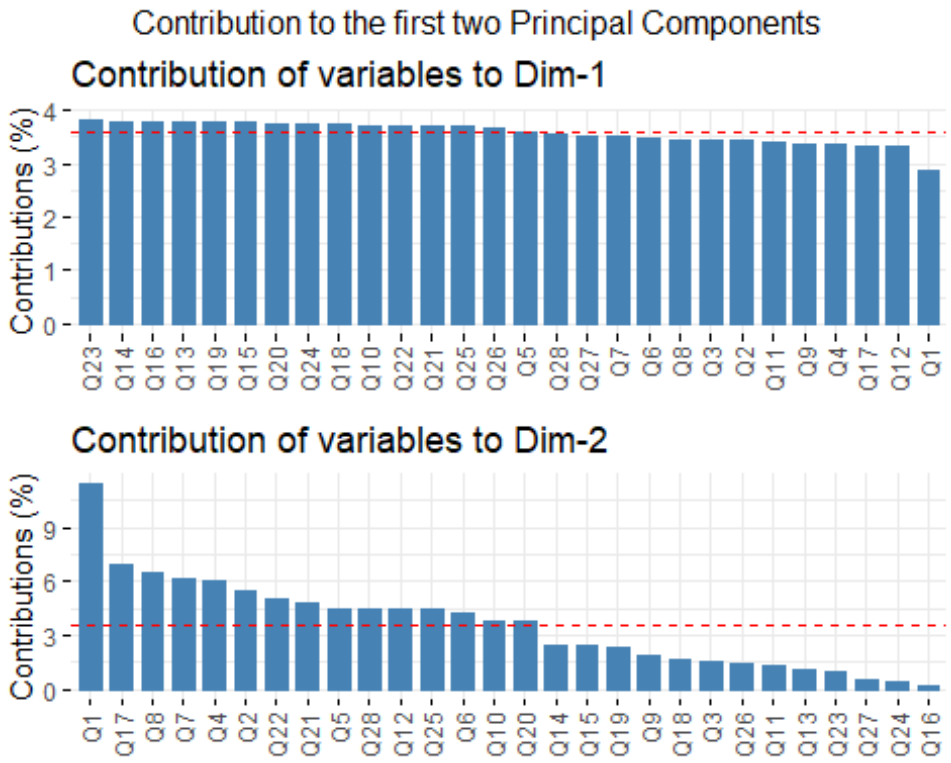



```
# displaying the most significant questions that constitute PC1
loading_scores_PC_1<-pca1$rotation[,1]
fac_scores_PC_1<-abs(loading_scores_PC_1)
fac_scores_PC_1_ranked<-names(sort(fac_scores_PC_1, decreasing=T))
pca1$rotation[fac_scores_PC_1_ranked, 1]

##        Q23        Q14        Q16        Q13        Q19        Q15
Q20
## -0.1955702 -0.1946421 -0.1946208 -0.1943247 -0.1941508 -0.1940270 -
0.1933655
##        Q24        Q18        Q10        Q22        Q21        Q25
Q26
## -0.1933136 -0.1932407 -0.1924670 -0.1923365 -0.1923313 -0.1920408 -
0.1918982
##         Q5        Q28        Q27         Q7         Q6         Q8
Q3
## -0.1897697 -0.1885680 -0.1875538 -0.1873440 -0.1863937 -0.1856411 -
```

```
0.1855657
##         Q2        Q11         Q9         Q4        Q17        Q12
Q1
## -0.1855459 -0.1839239 -0.1834801 -0.1828628 -0.1824796 -0.1818928 -
0.1697760

# individual results with factoextra::
ind<-get_pca_ind(pca1)
print(ind)

## Principal Component Analysis Results for individuals
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the individuals"
## 2 "$cos2"    "Cos2 for the individuals"
## 3 "$contrib" "contributions of the individuals"

# coordinates of variables
head(ind$coord)

##        Dim.1     Dim.2       Dim.3       Dim.4         Dim.5       Dim.6
## 1  0.7828076 0.4049825  0.02344185 -0.13472381  0.0003747595 -0.06801460
## 2  0.7828076 0.4049825  0.02344185 -0.13472381  0.0003747595 -0.06801460
## 3 -7.5011549 0.5014306  0.05494589 -0.03290546  0.0713849749 -0.05266376
## 4  0.7828076 0.4049825  0.02344185 -0.13472381  0.0003747595 -0.06801460
## 5  9.0667701 0.3085344 -0.00806218 -0.23654216 -0.0706354559 -0.08336544
## 6 -3.3591736 0.4532065  0.03919387 -0.08381463  0.0358798672 -0.06033918
##           Dim.7        Dim.8        Dim.9       Dim.10      Dim.11
## 1 -3.359882e-03  0.001246685 -0.002847188  0.001699091 -0.02821692
## 2 -3.359882e-03  0.001246685 -0.002847188  0.001699091 -0.02821692
## 3  3.293002e-03 -0.037836775  0.024422906  0.065558151 -0.01410519
## 4 -3.359882e-03  0.001246685 -0.002847188  0.001699091 -0.02821692
## 5 -1.001277e-02  0.040330145 -0.030117282 -0.062159968 -0.04232864
## 6 -3.344021e-05 -0.018295045  0.010787859  0.033628621 -0.02116105
##          Dim.12       Dim.13        Dim.14       Dim.15       Dim.16
## 1 -0.0002183858 -0.004922078 -0.0045741304 -0.009989832  0.008853189
## 2 -0.0002183858 -0.004922078 -0.0045741304 -0.009989832  0.008853189
## 3 -0.0221155073 -0.005970327 -0.0002840724 -0.044204686  0.026367512
## 4 -0.0002183858 -0.004922078 -0.0045741304 -0.009989832  0.008853189
## 5  0.0216787356 -0.003873828 -0.0088641884  0.024225023 -0.008661134
## 6 -0.0111669466 -0.005446203 -0.0024291014 -0.027097259  0.017610351
##         Dim.17       Dim.18        Dim.19       Dim.20       Dim.21
## 1 -0.012062937  0.007935729 -3.367341e-05 -0.007515912  0.003459993
## 2 -0.012062937  0.007935729 -3.367341e-05 -0.007515912  0.003459993
## 3 -0.032519693 -0.007265543 -3.259827e-03 -0.016198728  0.013713979
## 4 -0.012062937  0.007935729 -3.367341e-05 -0.007515912  0.003459993
## 5  0.008393819  0.023137002  3.192480e-03  0.001166903 -0.006793994
## 6 -0.022291315  0.000335093 -1.646750e-03 -0.011857320  0.008586986
##         Dim.22        Dim.23        Dim.24       Dim.25       Dim.26
## 1 -0.003431596 -0.0005593455 -0.0029658812 -0.003891139 -0.003379483
## 2 -0.003431596 -0.0005593455 -0.0029658812 -0.003891139 -0.003379483
```

```
## 3 -0.014072912 -0.0137573109 -0.0060386880 -0.012089408  0.006791426
## 4 -0.003431596 -0.0005593455 -0.0029658812 -0.003891139 -0.003379483
## 5  0.007209720  0.0126386200  0.0001069256  0.004307129 -0.013550392
## 6 -0.008752254 -0.0071583282 -0.0045022846 -0.007990273  0.001705972
##          Dim.27        Dim.28
## 1 -0.0012815641  0.0001516178
## 2 -0.0012815641  0.0001516178
## 3 -0.0020560579  0.0011822170
## 4 -0.0012815641  0.0001516178
## 5 -0.0005070703 -0.0008789814
## 6 -0.0016688110  0.0006669174

# contributions of individuals to PC
head(ind$contrib)

##          Dim.1       Dim.2        Dim.3        Dim.4        Dim.5
Dim.6
## 1 0.0004569699 0.002249195 2.390740e-05 8.642217e-04 8.324594e-09
0.0003102030
## 2 0.0004569699 0.002249195 2.390740e-05 8.642217e-04 8.324594e-09
0.0003102030
## 3 0.0419598365 0.003448071 1.313466e-04 5.155518e-05 3.020448e-04
0.0001859797
## 4 0.0004569699 0.002249195 2.390740e-05 8.642217e-04 8.324594e-09
0.0003102030
## 5 0.0613031414 0.001305455 2.827831e-06 2.664119e-03 2.957353e-04
0.0004660300
## 6 0.0084147734 0.002816741 6.683205e-05 3.344846e-04 7.630612e-05
0.0002441409
##          Dim.7        Dim.8        Dim.9       Dim.10       Dim.11
Dim.12
## 1 9.500780e-07 1.457212e-07 8.075936e-07 3.476674e-07 9.902738e-05
5.984077e-09
## 2 9.500780e-07 1.457212e-07 8.075936e-07 3.476674e-07 9.902738e-05
5.984077e-09
## 3 9.126307e-07 1.342263e-04 5.942309e-05 5.175877e-04 2.474539e-05
6.136801e-05
## 4 9.500780e-07 1.457212e-07 8.075936e-07 3.476674e-07 9.902738e-05
5.984077e-09
## 5 8.437607e-06 1.524996e-04 9.036331e-05 4.653205e-04 2.228460e-04
5.896796e-05
## 6 9.411289e-11 3.138168e-05 1.159394e-05 1.361911e-04 5.569431e-05
1.564650e-05
##         Dim.13       Dim.14       Dim.15       Dim.16       Dim.17
Dim.18
## 1 3.496221e-06 3.089095e-06 1.501425e-05 1.227713e-05 2.364261e-05
1.075708e-05
## 2 3.496221e-06 3.089095e-06 1.501425e-05 1.227713e-05 2.364261e-05
1.075708e-05
## 3 5.143967e-06 1.191440e-08 2.939842e-04 1.089020e-04 1.718234e-04
```

```
9.016893e-06
## 4 3.496221e-06 3.089095e-06 1.501425e-05 1.227713e-05 2.364261e-05
1.075708e-05
## 5 2.165622e-06 1.160091e-05 8.829087e-05 1.175024e-05 1.144744e-05
9.143972e-05
## 6 4.280450e-06 8.711752e-07 1.104684e-04 4.857730e-05 8.073479e-05
1.918012e-08
##         Dim.19       Dim.20       Dim.21       Dim.22       Dim.23
Dim.24
## 1 2.044798e-10 1.046308e-05 2.431241e-06 2.400010e-06 6.681618e-08
1.953554e-06
## 2 2.044798e-10 1.046308e-05 2.431241e-06 2.400010e-06 6.681618e-08
1.953554e-06
## 3 1.916307e-06 4.860245e-05 3.819484e-05 4.036348e-05 4.041927e-05
8.098472e-06
## 4 2.044798e-10 1.046308e-05 2.431241e-06 2.400010e-06 6.681618e-08
1.953554e-06
## 5 1.837945e-06 2.522123e-07 9.374067e-06 1.059394e-05 3.411306e-05
2.539111e-09
## 6 4.890255e-07 2.604170e-05 1.497474e-05 1.561207e-05 1.094321e-05
4.501775e-06
##         Dim.25       Dim.26       Dim.27       Dim.28
## 1 3.675646e-06 2.897677e-06 5.057706e-07 7.615868e-09
## 2 3.675646e-06 2.897677e-06 5.057706e-07 7.615868e-09
## 3 3.548054e-05 1.170231e-05 1.301798e-06 4.630346e-07
## 4 3.675646e-06 2.897677e-06 5.057706e-07 7.615868e-09
## 5 4.503561e-06 4.658577e-05 7.917886e-08 2.559637e-07
## 6 1.549899e-05 7.384025e-07 8.576049e-07 1.473544e-07

var<-get_pca_var(pca1)
a<-fviz_contrib(pca1, "var", axes=1, xtickslab.rt=90) # default angle=45°
b<-fviz_contrib(pca1, "var", axes=2, xtickslab.rt=90)
grid.arrange(a,b,top='Contribution to the first two Principal Components')
```



Clustering

K-MEANS

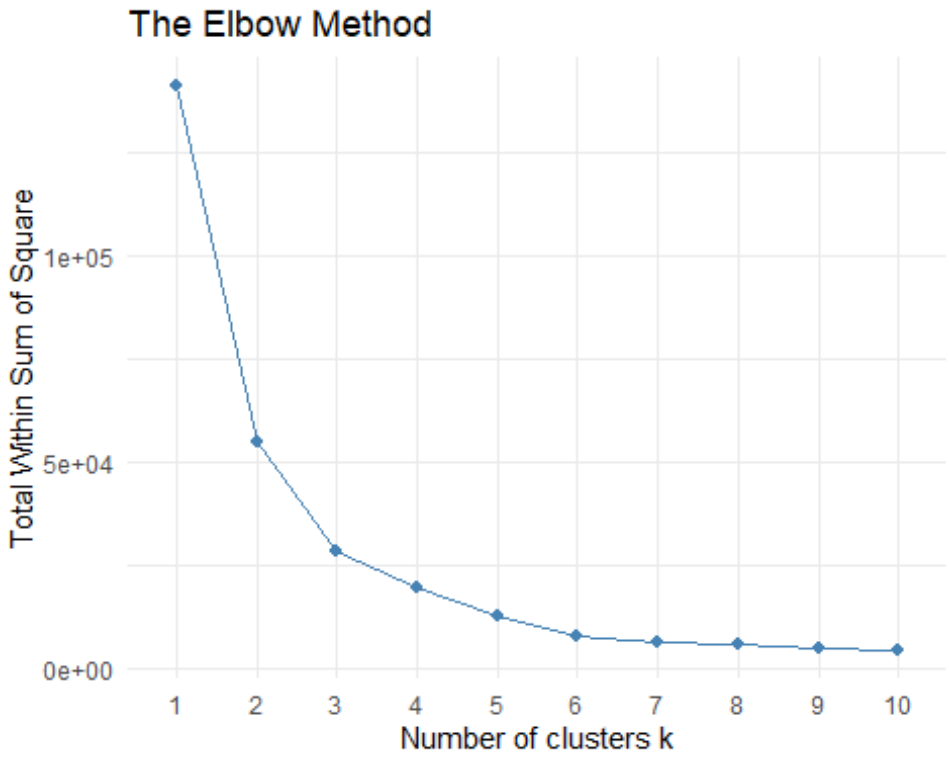
K-means clustering is a popular unsupervised learning algorithm used to identify patterns in the data by grouping similar observations into clusters. After performing PCA analysis, we can use the resulting principal components as the input to the k-means clustering algorithm. By using PCA results as input we can effectively identify the most important features that separate the data into different clusters. This approach can be particularly useful when dealing with high-dimensional data, as it can help to reduce the "curse of dimensionality" and improve the efficiency and interpretability of the clustering results.

Calculating Optimal Number of Clusters

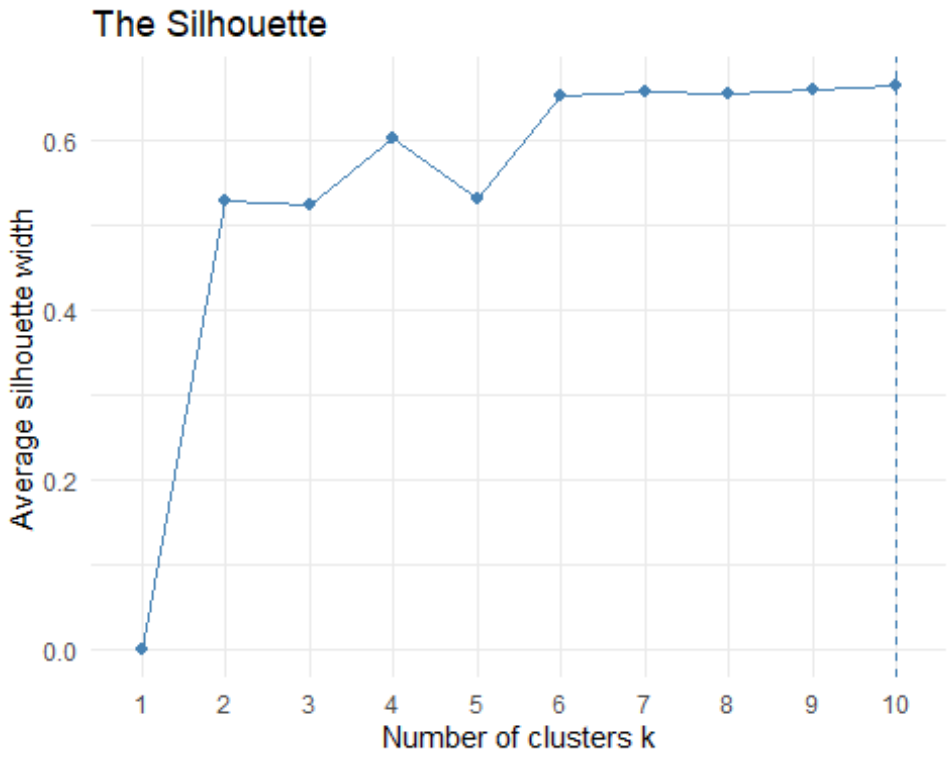
Determining the optimal number of clusters is a crucial step in clustering analysis. There are several methods to determine the optimal number of clusters, and the appropriate method to use may depend on the specific characteristics of your dataset and the clustering algorithm you are using. It is important to note that there is no one "correct" method to determine the optimal number of clusters, and it may be helpful to try multiple methods and compare the results. Additionally, the optimal number of clusters may not always be clear-cut, and it's important to interpret the results with caution and domain knowledge. To determine the optimal number of clusters for k-means clustering, we can use both the elbow method and the silhouette method.Because of different results, I tried to cluster with both way. General idea is the elbow method tends to be more appropriate when the clusters are well separated, while the silhouette method is better when the clusters are

overlapping or irregularly shaped. As we see in graphs there is a overlapping points among clusters but still 3 cluster can be enough.

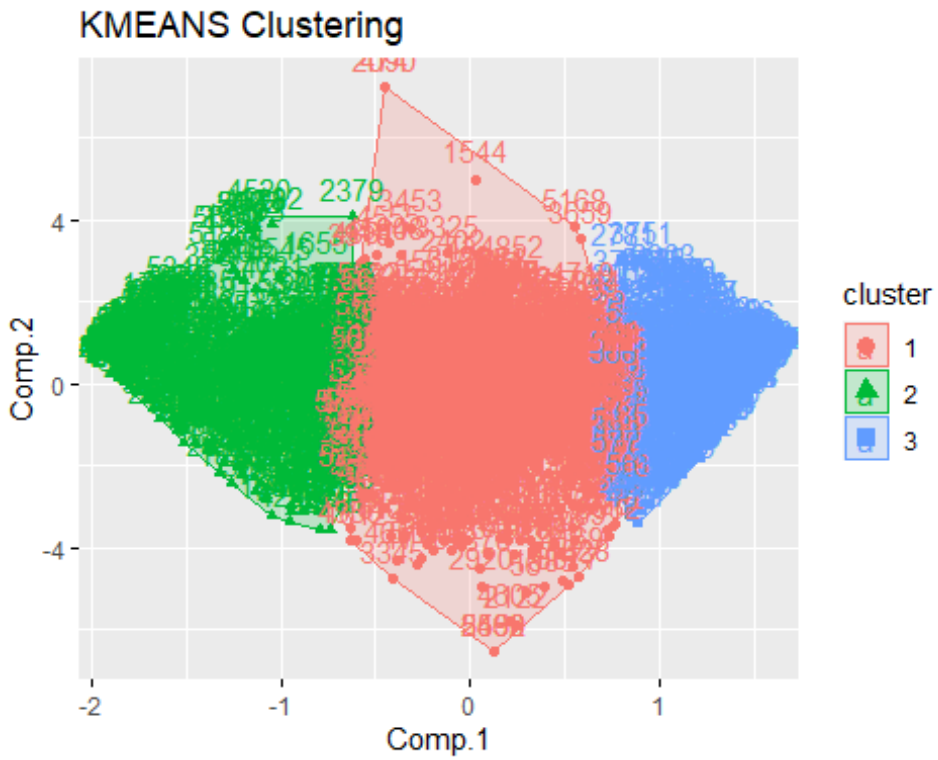
```
###using PCA results for furhther
set.seed(123) # for reproducibility
ss.cs<-center_scale(scaled_subset)
ss.pca<-princomp(ss.cs)$scores[, 1:2]

#Determinin optimal number of cluster

##using the elbow method using wcsse
fviz_nbclust(ss.pca, FUNcluster=kmeans, method = "wss", k.max = 10) +
theme_minimal() + ggtitle("The Elbow Method")
```



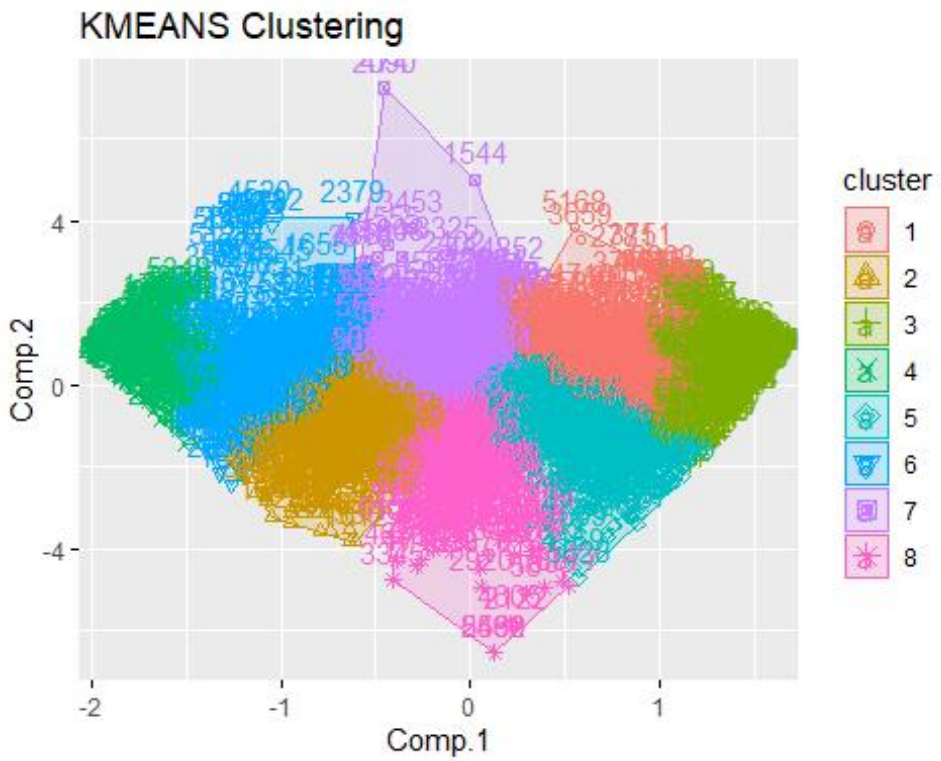

```
###using silhouette and kmeans
fviz_nbclust(ss.pca, kmeans, method="silhouette")+ theme_minimal()+
ggtitle("The Silhouette") # factoextra::
```




```r
# 3 clusters for observations
km<-eclust(ss.pca, k=3)
```



```
km2<-eclust(ss.pca, k=8)
```




```
fviz_silhouette(km)
```

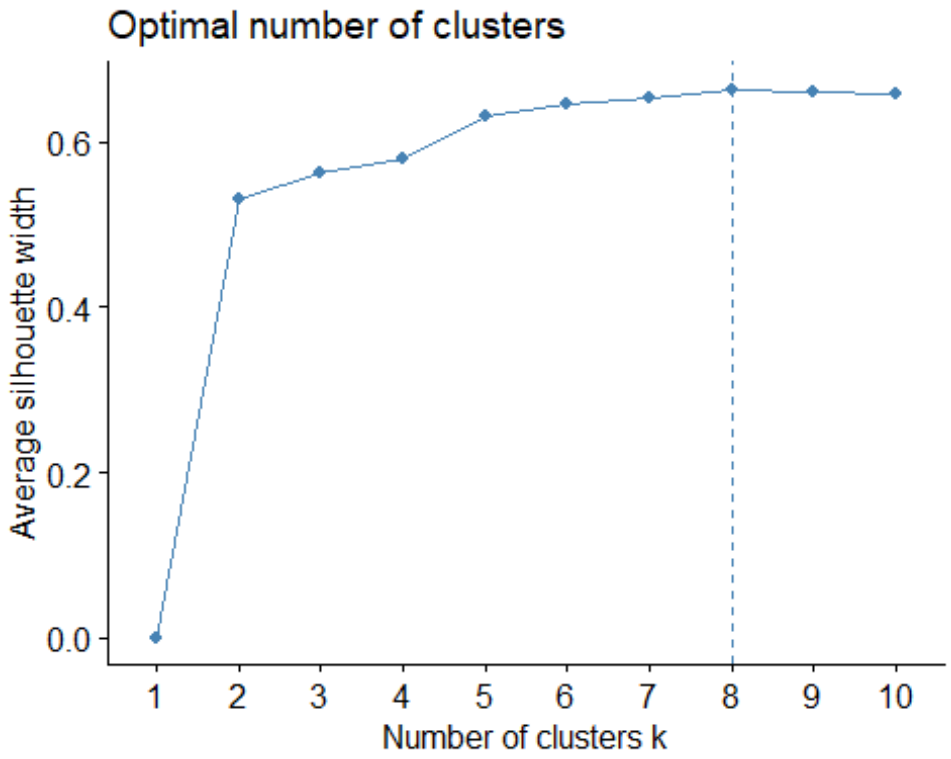
```
##   cluster size ave.sil.width
## 1       1 3454          0.43
## 2       2 1355          0.62
## 3       3 1011          0.71
```


```
fviz_silhouette(km2)

##   cluster size ave.sil.width
## 1       1 1222          0.72
## 2       2  299          0.34
## 3       3  837          0.76
## 4       4  721          0.86
## 5       5  419          0.25
## 6       6  540          0.52
## 7       7 1493          0.72
## 8       8  289          0.28
```

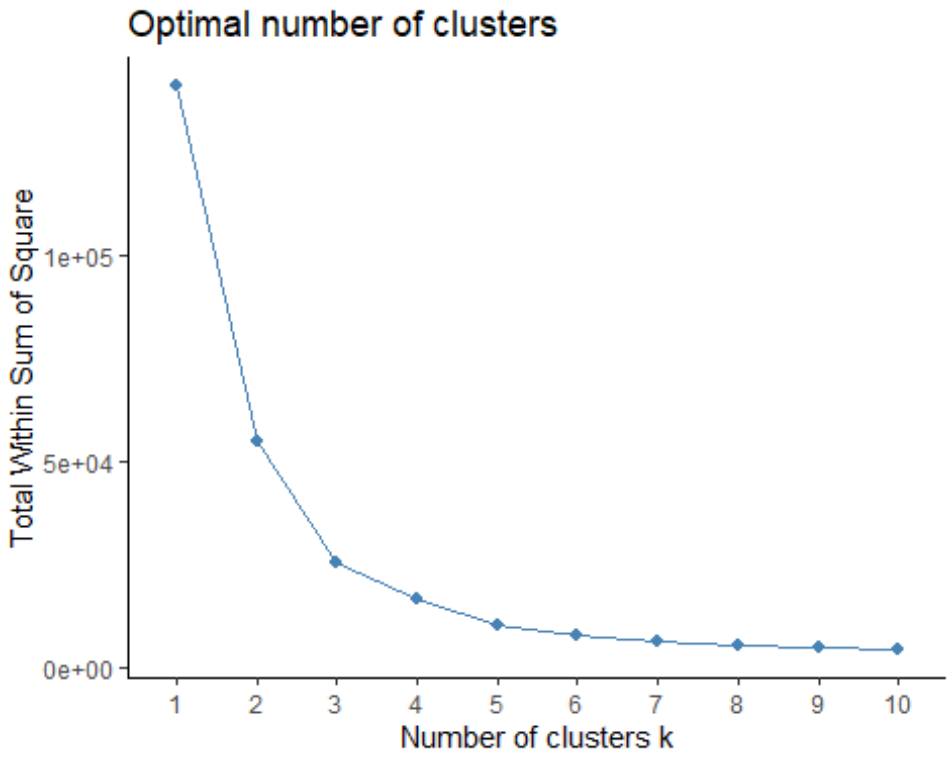

```
# k-means clustering  with PCA result

pcakm3<-KMeans_rcpp(ss.pca, clusters=3, num_init=3, max_iters = 100)
pcakm3
c3<-plot_2d(ss.pca, pcakm3$clusters, pcakm3$centroids)
c3

## KMeans Cluster
##  Call: KMeans_rcpp(data = ss.pca, clusters = 3, num_init = 3, max_iters =
100)
##  Data cols: 2
##  Centroids: 3
##  BSS/SS: 0.8323422
##  SS: 141365.7 = 23701.07 (WSS) + 117664.7 (BSS)

pcakm8<-KMeans_rcpp(ss.pca, clusters=8, num_init=3, max_iters = 100)
pcakm8
c8<-plot_2d(ss.pca, pcakm8$clusters, pcakm8$centroids)
c8

## KMeans Cluster
##  Call: KMeans_rcpp(data = ss.pca, clusters = 8, num_init = 3, max_iters =
100)
##  Data cols: 2
##  Centroids: 8
##  BSS/SS: 0.9615616
##  SS: 141365.7 = 5433.87 (WSS) + 135931.9 (BSS)
```

PAM

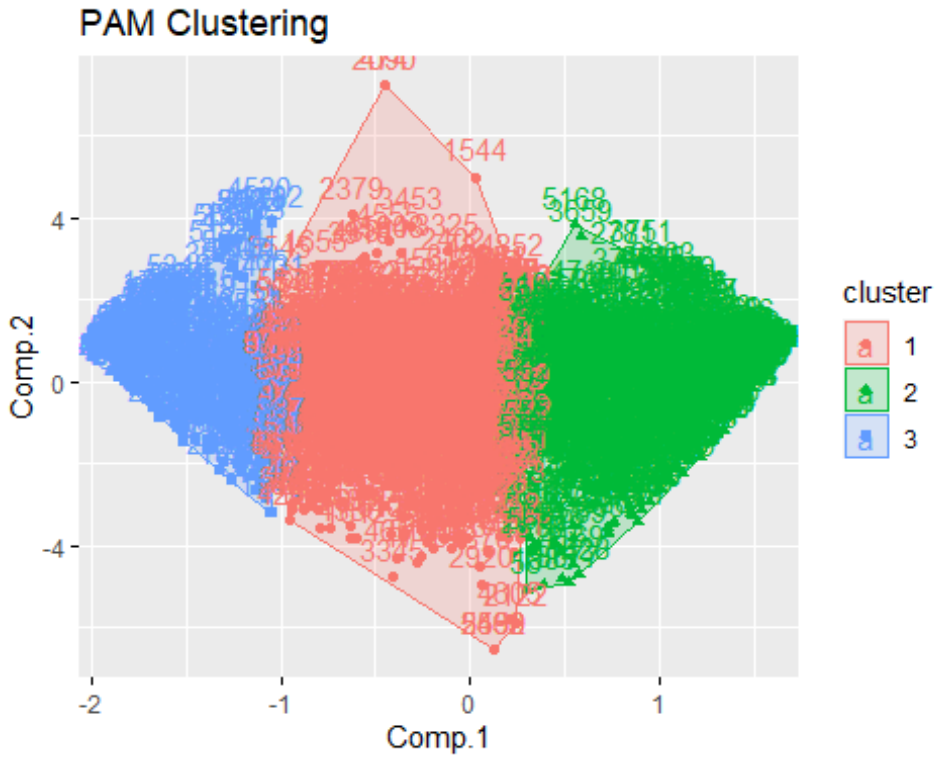
```
#Determinin optimal number of cluster for PAM

fviz_nbclust(ss.pca, FUNcluster=cluster::pam )
```




```
fviz_nbclust(ss.pca, FUNcluster=cluster::pam, method="wss")+ theme_classic()
```

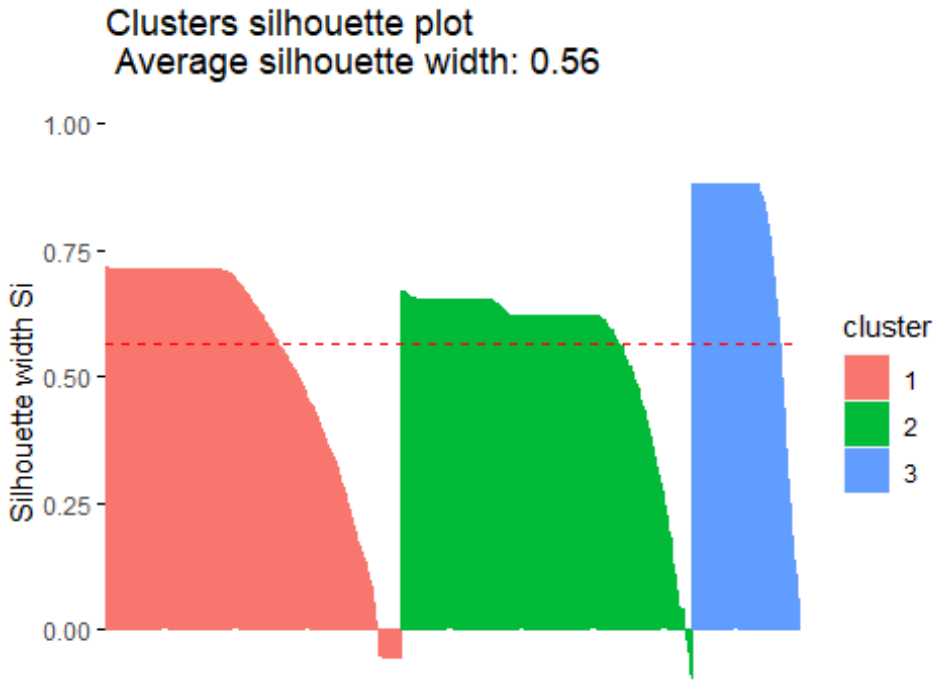



```
##
pam1<-eclust(ss.pca, "pam", k=3) #for 3 cluster
```

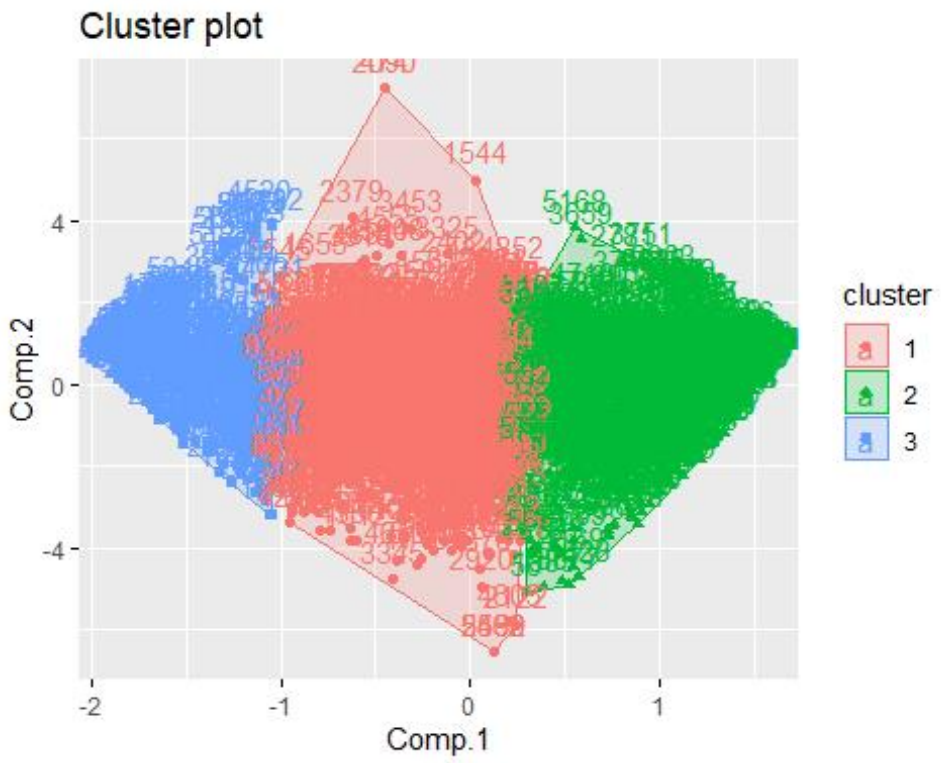


```
fviz_silhouette(pam1)
```

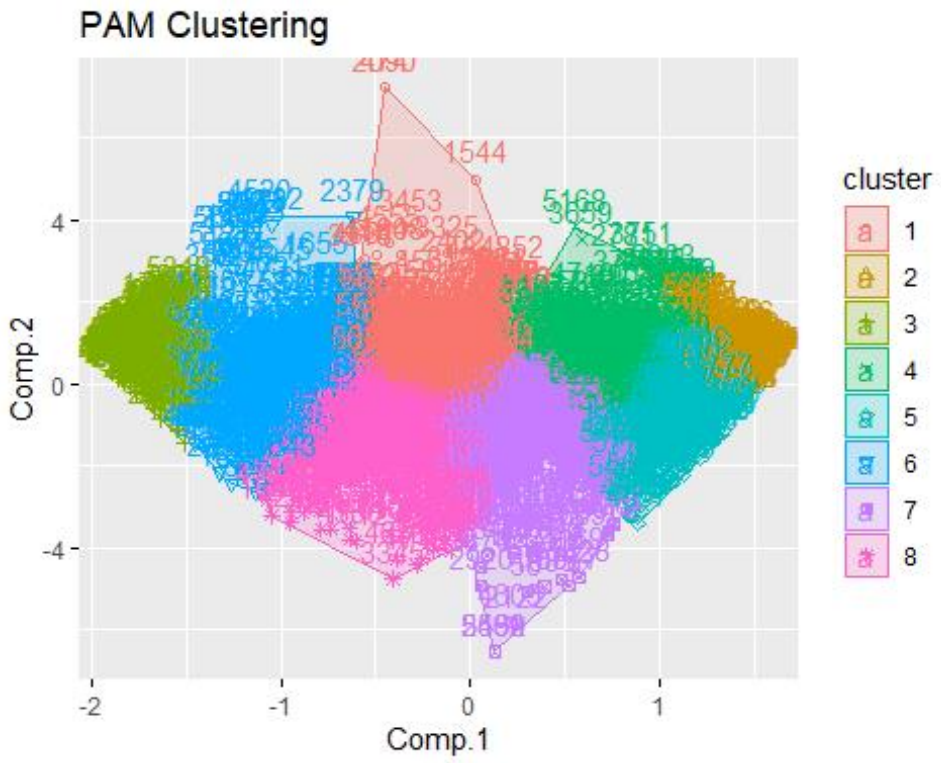
```
##   cluster size ave.sil.width
## 1       1 2481          0.51
## 2       2 2446          0.55
## 3       3  893          0.75
```




```
fviz_cluster(pam1)
```

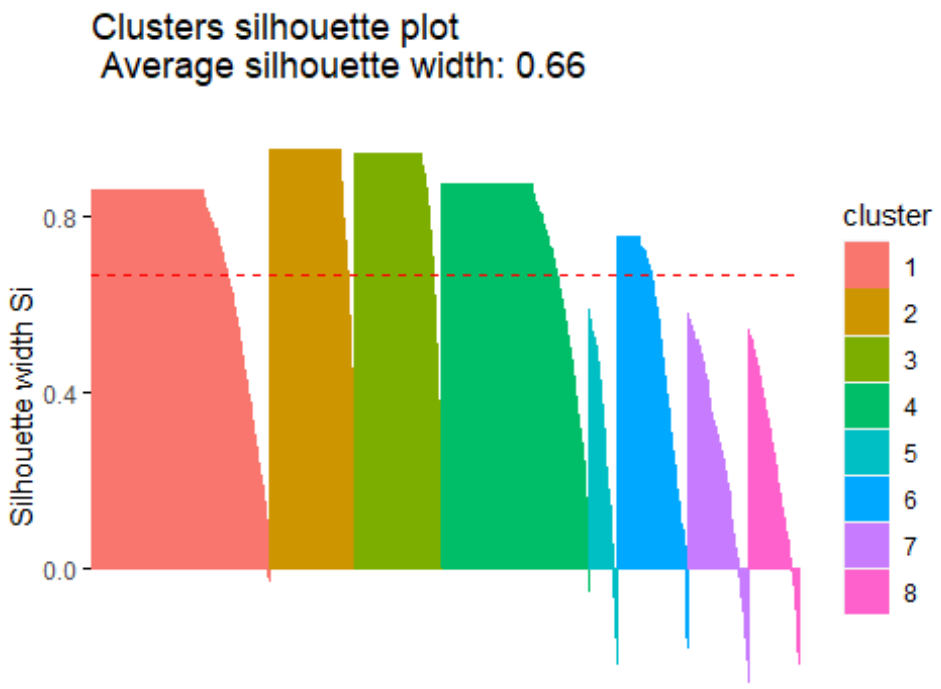



```
pam2<-eclust(ss.pca, "pam", k=8) #for 8
```

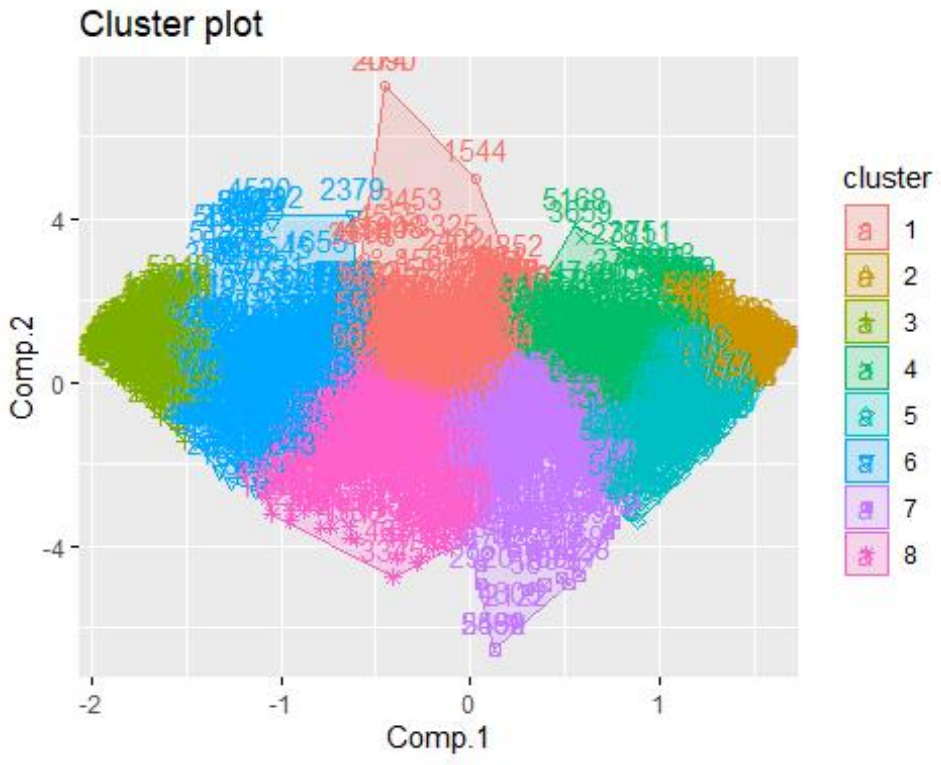



```
fviz_silhouette(pam2)
```

```
##   cluster size ave.sil.width
## 1       1 1456          0.74
## 2       2  708          0.89
## 3       3  715          0.88
## 4       4 1211          0.76
## 5       5  230          0.30
## 6       6  580          0.53
## 7       7  502          0.27
## 8       8  418          0.24
```

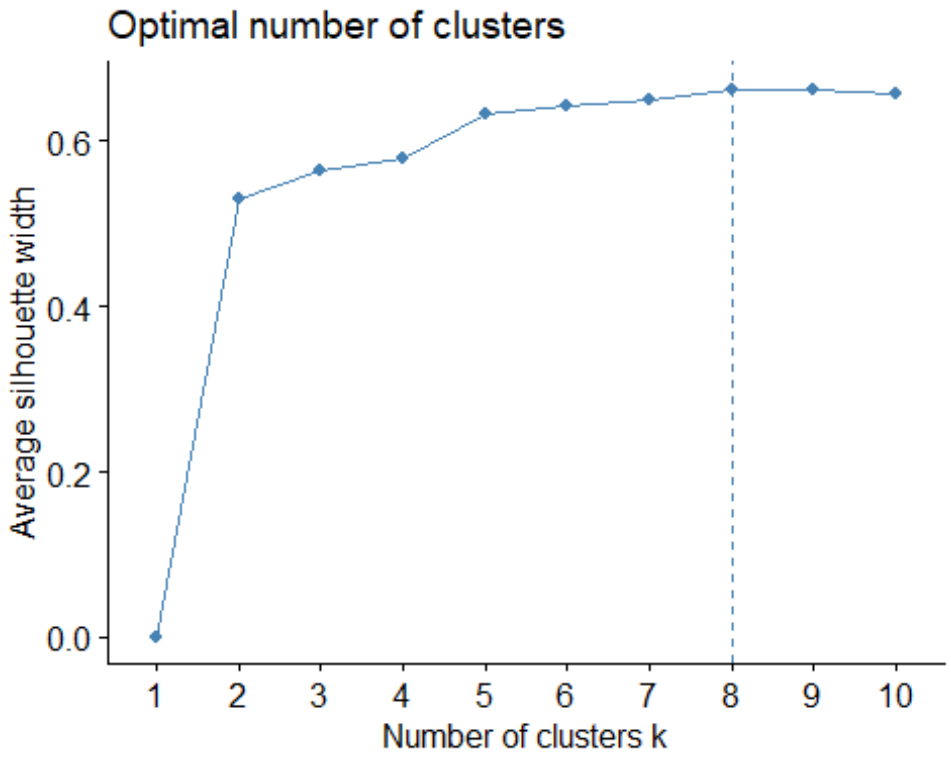



```
fviz_cluster(pam2)
```

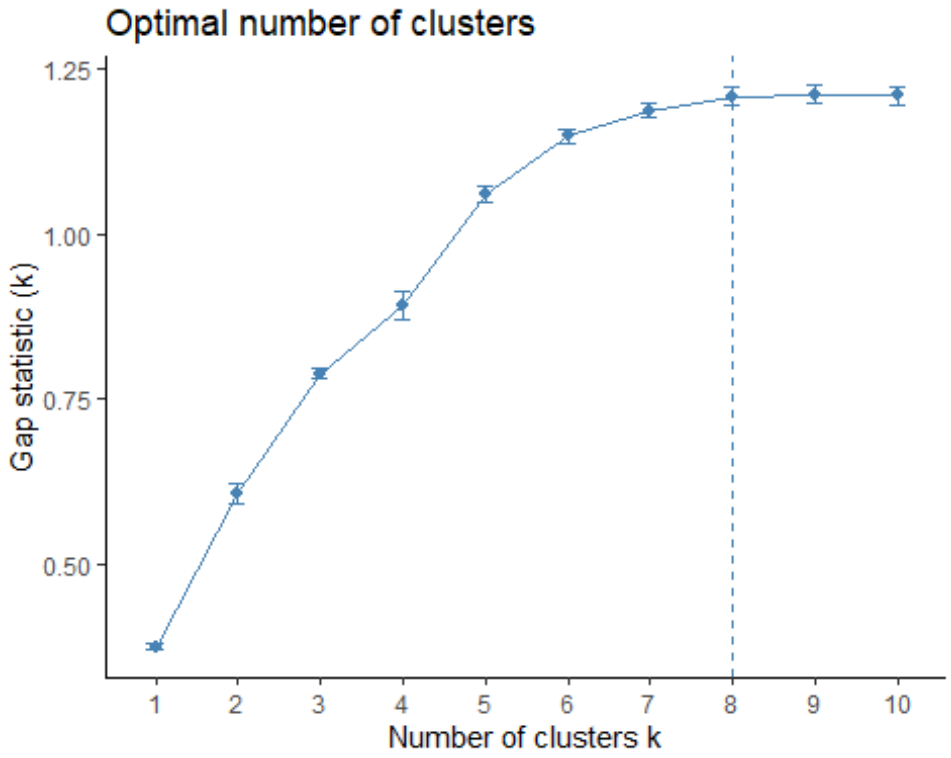


CLARA

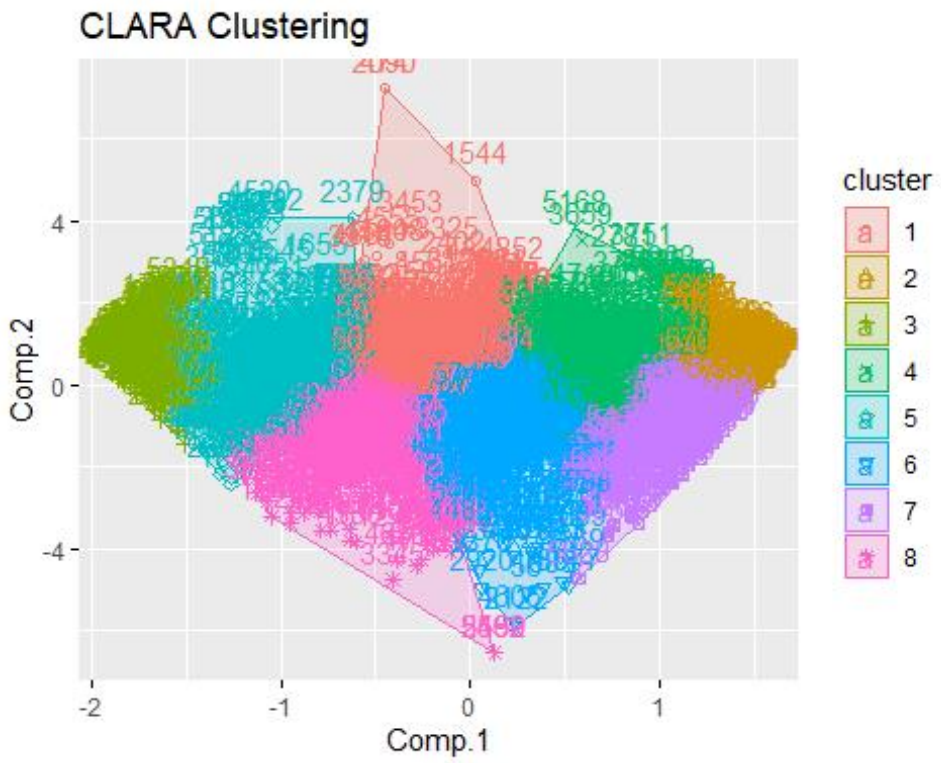
```
#Determinin optimal number of cluster for CLARA

fviz_nbclust(ss.pca, FUNcluster=cluster::clara )
```



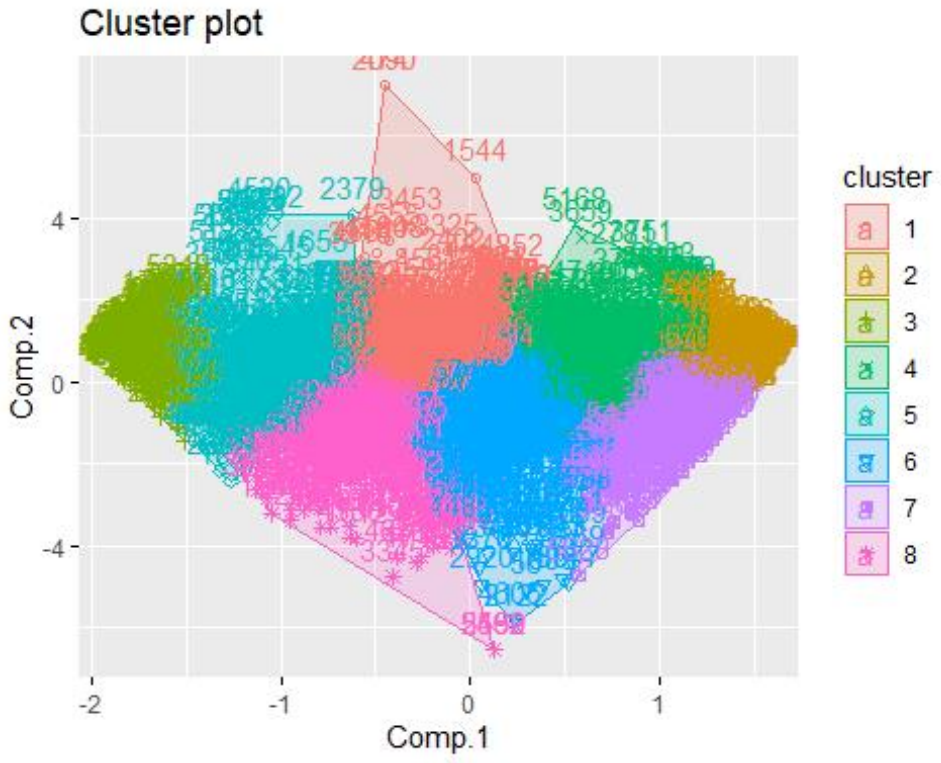

```
fviz_nbclust(ss.pca, FUNcluster=cluster::clara, method="gap_stat")+
theme_classic()
```



```
cl<-eclust(ss.pca, "clara", k=8) # factoextra
```

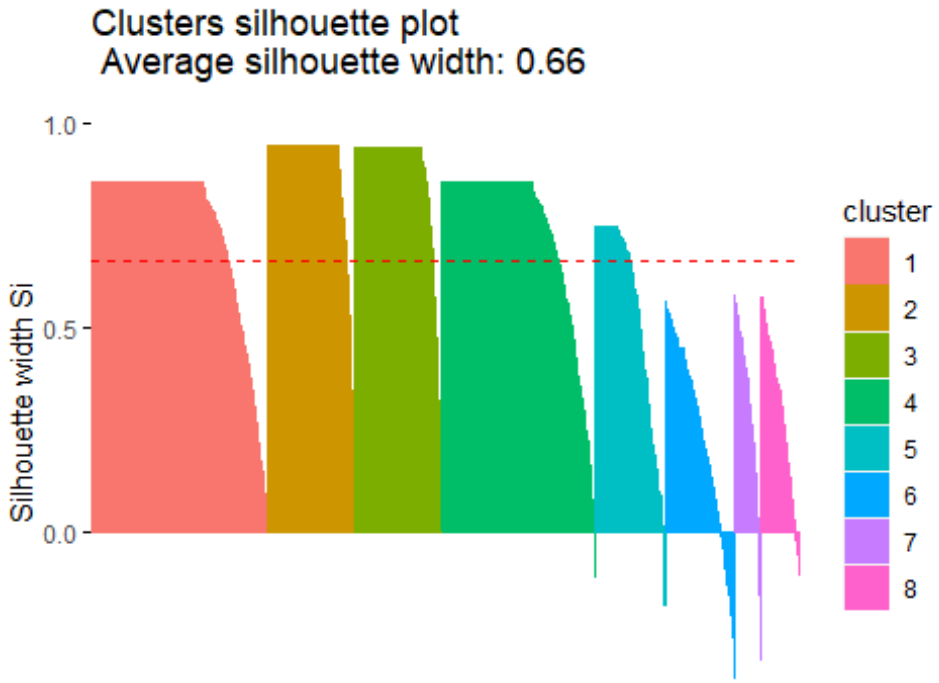



```
fviz_cluster(cl)
```



```
fviz_silhouette(cl)
```

```
##   cluster size ave.sil.width
## 1       1 1439          0.74
## 2       2  720          0.89
## 3       3  715          0.88
## 4       4 1273          0.73
## 5       5  568          0.53
## 6       6  573          0.23
## 7       7  221          0.30
## 8       8  311          0.30
```




```
# Compute silhouette coefficient for k-means clustering
km_silhouette <- silhouette(km2$cluster, dist(ss.pca))
cat("Silhouette coefficient for k-means clustering:",
mean(km_silhouette[,3]), "\n")
```

```
## Silhouette coefficient for k-means clustering: 0.6519274
```

```
# Compute silhouette coefficient for PAM clustering
pam_silhouette <- silhouette(pam2$cluster, dist(ss.pca))
cat("Silhouette coefficient for PAM clustering:", mean(pam_silhouette[,3]),
"\n")
```

```
## Silhouette coefficient for PAM clustering: 0.6629984
```

```
# Compute silhouette coefficient for CLARA clustering
clara_silhouette <- silhouette(cl$cluster,dist(ss.pca))
```

```
cat("Silhouette coefficient for CLARA clustering:",
mean(clara_silhouette[,3]), "\n")

## Silhouette coefficient for CLARA clustering: 0.6622736
```

Conclusion

The dataset contains evaluation questions for courses and instructors by Turkish university students.The dataset was relatively clean, with no missing data or obvious errors.Principal Component Analysis (PCA) was performed to reduce the dimensionality of the evaluation questions, and two principal components were chosen for further analysis. K-means, PAM and CLARA clustering were performed on the PCA results to group evaluations into clusters based on similarities in responses to the evaluation questions. The optimal number of clusters was found as 8.The silhouette coefficients for k-means, PAM, and CLARA clustering are all relatively high and close to each other, indicating that all three clustering algorithms have produced relatively good clustering solutions. However, the fact that the silhouette coefficient for PAM clustering is slightly higher than the other two algorithms may suggest that PAM is a slightly better fit for the data. The PCA and clustering results suggest that there are some underlying patterns in the responses to the evaluation questions, but more detailed analysis would be needed to fully understand the nature of these patterns and their implications.

###References

Gunduz, N., & Fokoue, E. (2015). Pattern Discovery in Students' Evaluations of Professors: A Statistical Data Mining Approach. arXiv preprint arXiv:1501.02263.

Dataset: https://archive.ics.uci.edu/ml/datasets/turkiye+student+evaluation

https://www.r-project.org/

https://scikit-learn.org/stable/modules/clustering.html

###END