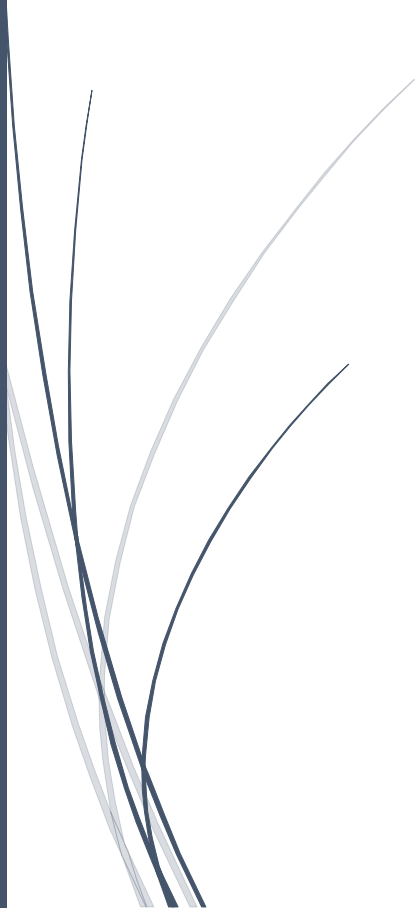
The Big D's

# Assignment 3

Rick Banerjee, Anna Maria, John See,



Student Survey Response Dataset

## Introduction

As an educational institute it is important that the courses being offered are well received by students, otherwise it does not serve a function. By utilizing the scores of the survey, it can be determined as to the validity of such a course and whether it should be discontinued or not. By using data mining techniques, the student responses can be analyzed and accordingly can be interpreted to determine their satisfaction with the overall course. For the client in question our objective is to develop a model which will assist them in finding the key variables which will have a measurable and meaningful impact on their decision to continue with the program or not.

## Analytics Problem

Our goal is to group the students based on the similarity of their answers on the survey. The main purpose of the application of clustering in this project is to explore the factors affecting the level of satisfaction of students with the performance of Teaching Professor, as well as courses.

## Data Source

The dataset is collected from University of California Irvine (UCI) Machine Learning Repository at http://archive.ics.uci.edu/ml/datasets/turkiye+student+evaluation and contains a total of 5,820 evaluation scores provided by students from Gazi University in Ankara, Turkey. There is a total of 28 course specific questions and additional 5 attributes.

The attributes information given below are divided into two sections: First one contains general information and the second holds evaluation questions:

| General Attributes | Description | Data Type |
|---|---|---|
| instr | Instructor's identifier; values taken from {1,2,3} | Numeric |
| class | Course code (descriptor); values taken from {1-13} | Numeric |
| repeat | Number of times the student is taking this course; values taken from {0,1,2,3,...} | Numeric |
| attendance | Code of the level of attendance; values from {0, 1, 2, 3, 4} | Numeric |

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

| Evaluation Attributes | Description | Data Type |
|---|---|---|
| Q1 | The semester course content, teaching method and evaluation system were provided at the start. | Numeric |
| | | |
| Q2 | The course aims and objectives were clearly stated at the beginning of the period. | Numeric |
| Q3 | The course was worth the amount of credit assigned to it. | Numeric |
| Q4 | The course was taught according to the syllabus announced on the first day of class. | Numeric |
| Q5 | The class discussions, homework assignments, applications and studies were satisfactory. | Numeric |
| Q6 | The textbook and other courses resources were sufficient and up to date. | Numeric |
| Q7 | The course allowed field work, applications, laboratory, discussion and other studies. | Numeric |
| Q8 | The quizzes, assignments, projects and exams contributed to helping the learning. | Numeric |
| Q9 | I greatly enjoyed the class and was eager to actively participate during the lectures. | Numeric |
| Q10 | My initial expectations about the course were met at the end of the period or year. | Numeric |
| Q11 | The course was relevant and beneficial to my professional development. | Numeric |
| Q12 | The course helped me look at life and the world with a new perspective. | Numeric |
| Q13 | The Instructor's knowledge was relevant and up to date. | Numeric |
| Q14 | The Instructor came prepared for classes. | Numeric |

| | | |
|---|---|---|
| Q15 | The Instructor taught in accordance with the announced lesson plan. | Numeric |
| Q16 | The Instructor was committed to the course and was understandable. | Numeric |
| Q17 | The Instructor arrived on time for classes. | Numeric |
| Q18 | The Instructor has a smooth and easy to follow delivery/speech. | Numeric |
| Q19 | The Instructor made effective use of class hours. | Numeric |
| Q20 | The Instructor explained the course and was eager to be helpful to students. | Numeric |
| Q21 | The Instructor demonstrated a positive approach to students. | Numeric |
| Q22 | The Instructor was open and respectful of the views of students about the course. | Numeric |
| Q23 | The Instructor encouraged participation in the course. | Numeric |
| Q24 | The Instructor gave relevant homework assignments/projects, and helped/guided students. | Numeric |
| Q25 | The Instructor responded to questions about the course inside and outside of the course. | Numeric |
| Q26 | The Instructor's evaluation system (midterm and final questions, projects, assignments, etc.) effectively measured the course objectives. | Numeric |
| Q27 | The Instructor provided solutions to exams and discussed them with students. | Numeric |
| Q28 | The Instructor treated all students in a right and objective manner. | Numeric |

## Data Exploration

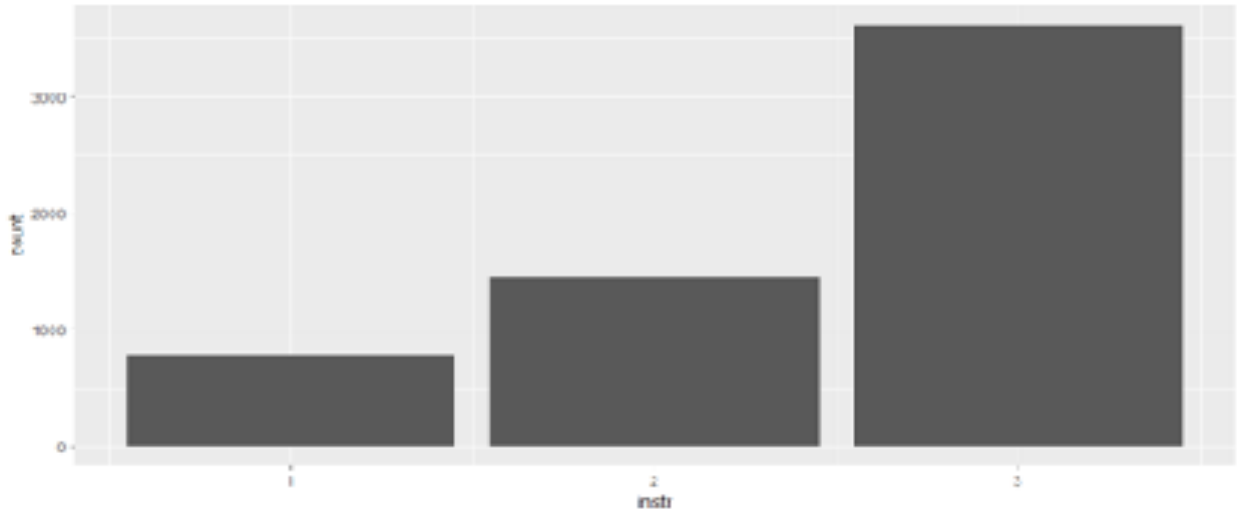
The data set exploration in R as well as attribute information given above provided valuable information regarding the data set.

**Data Description**

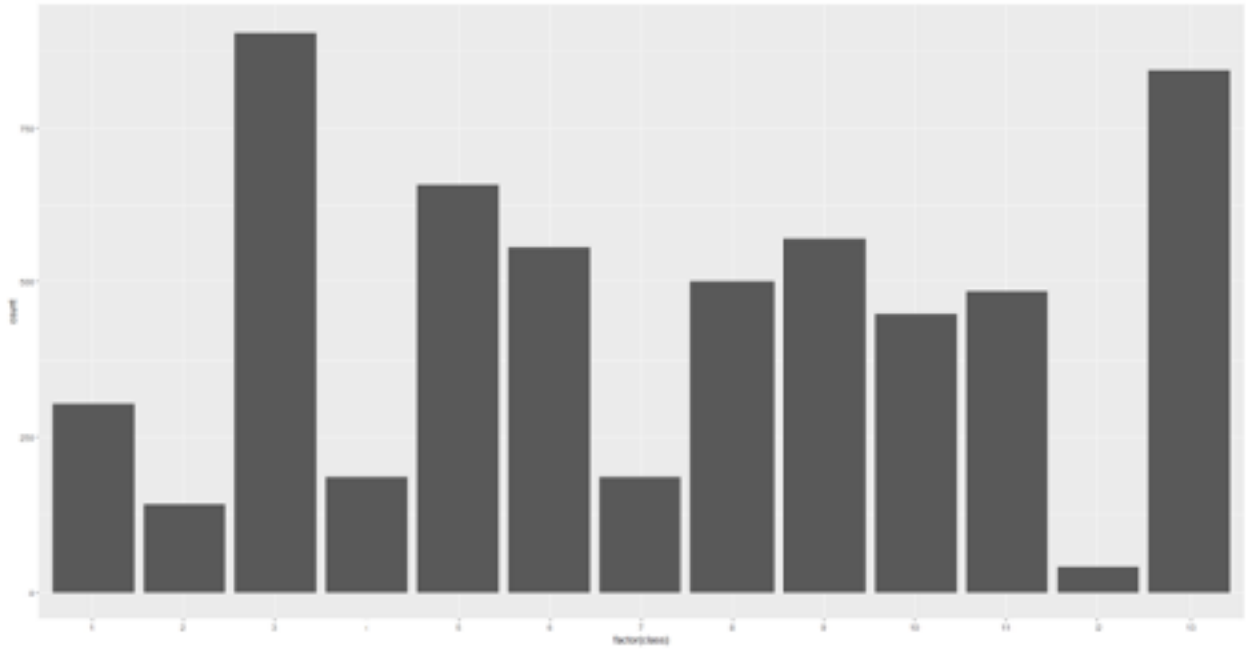
1) Q1-Q28 are all Likert-type in which responses are scored as {1, 2, 3, 4, 5}.Q1-Q12 are course based questions, Q13-Q28 are instructor based questions.
2) The class label attribute is suggested to be result and takes values also as {1, 2, 3, 4, 5} whereas result values that are greater than 3 are considered to be very good, result values that are equal to 3 are considered as good, and result values that are less than 3 are considered to be bad.
3) Moreover, the level of attendance values is taken as {0, 1, 2, 3, 4} whereas values less 2 are considered to be weak, values equal to 2 are considered to be medium, and values greater than 2 are considered to be good.
4) Furthermore, level of difficulty of the course values are taken as {1, 2, 3, 4, 5}, whereas values less than 3 are considered to be low, values equal to 3 are medium, and values greater than 3 noted as high.
5) We will start by working on the entire dataset, however there are 3 distinct parts:
    1. The first 5 columns indicate the instructor who taught the course, course being surveyed, difficulty, and attendance. These are key attributes to further separate the responses from the questionnaires to address clusters of students' responses by either specific course, which instructor, or the difficulty level of a given course (this can also be grouped).
    2. Questions related to the specific course itself. If we would like to focus on responses that relate only to the course we would extract columns Q1:Q12 → This will give an indication of how different clusters of students perceived the course. This subset can also be used to assess overall satisfaction for a specific course (for example: by setting attribute to 'course = 2').
    3. Questions related to the specific instructor. If we would like to focus on responses that relate only to the instructor we would extract columns Q13:Q28 and cluster students based on that → and we'll have an idea how distinct groups of students perceived their instructor. This subset can also be used to assess overall satisfaction for a specific instructor (for example: by setting attribute to 'instructor = 1').
6) Columns 1:5 are descriptive of the course, instructor, difficulty and number of times the course was taken. As these were non-questionnaire data, these columns were not included as part of the dataset used for clustering algorithms. As mentioned earlier, these attributes will help find clusters which can be attributed to a specific course, difficulty level, or instructor.

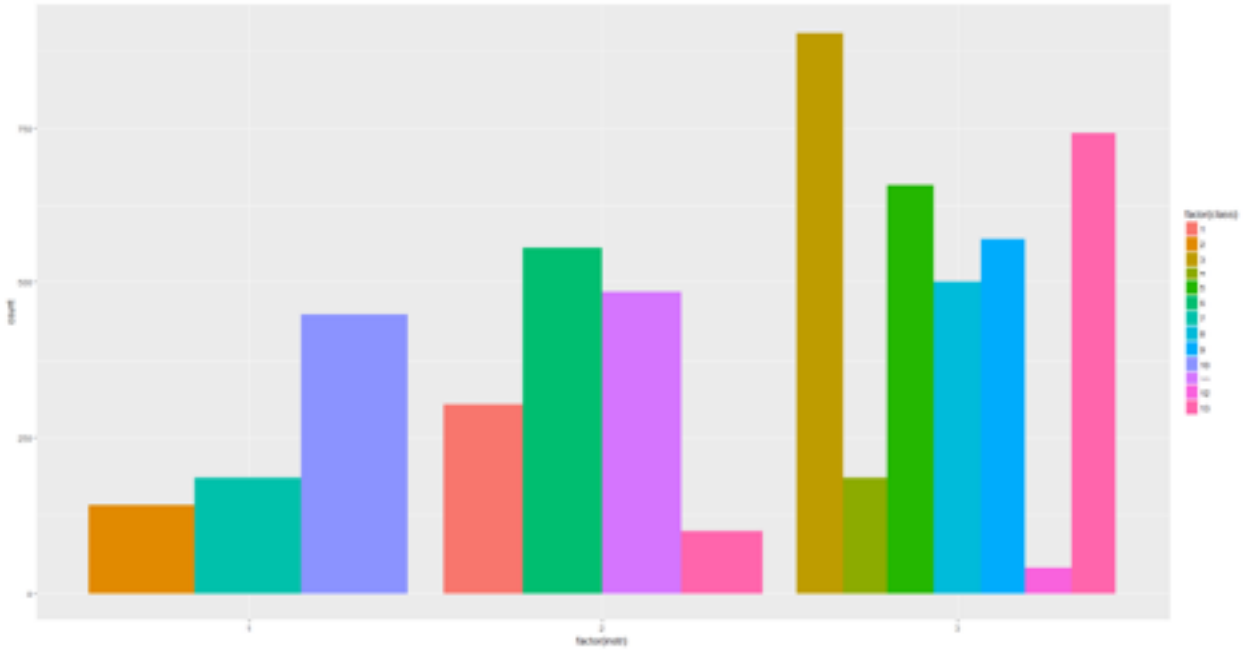
**Data Understanding**

To understand the data, we can compare the number of responses by those 3 Instructors. So as per the below graph most student taught by the third Instructor provided more responses, but this don't necessarily mean he taught most of the classes.




The number of responses of some classes is clearly much lower compared to the rest. Doing a little bit more analysis, we could see that the 3rd instructor taught 7 classes: class no 3, 4, 5, 8, 9, 12, and 13.

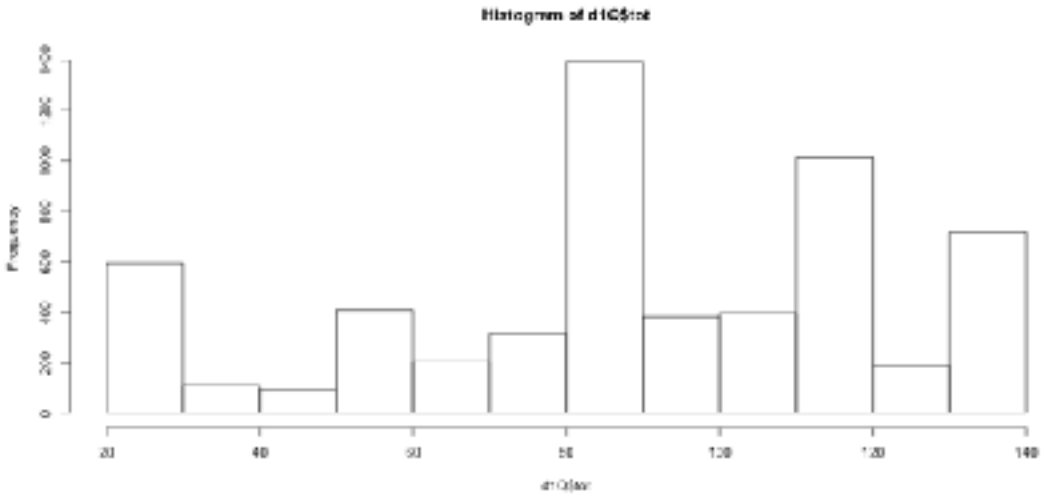




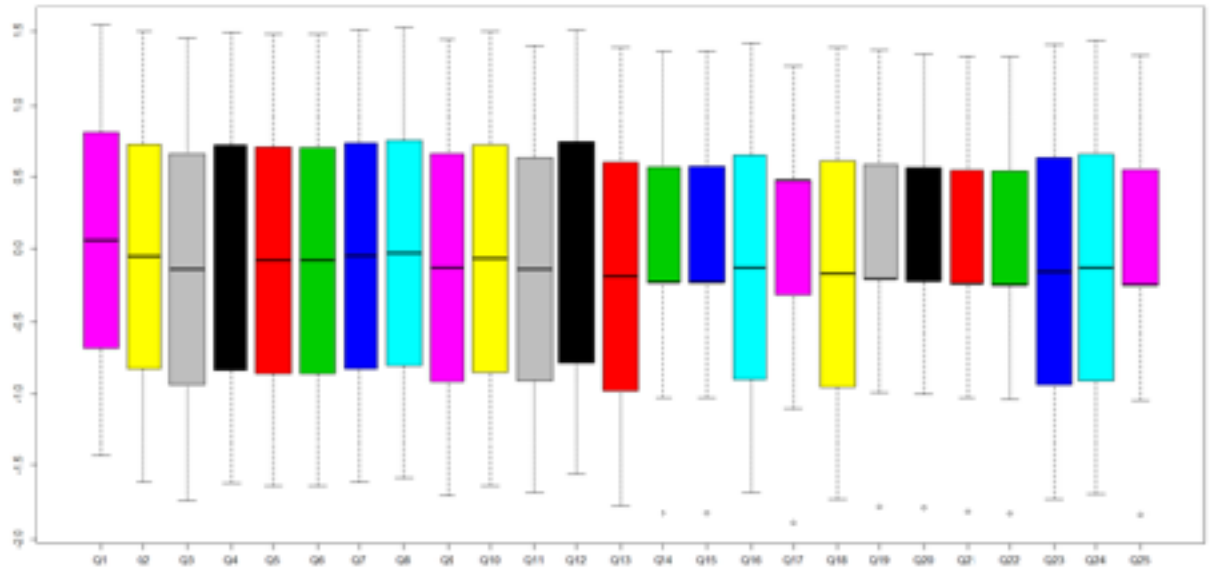
Sum of Student Questionnaire Score provides a basic overall view of how students perceived both the instructors and courses. We can infer that if a student is satisfied with the course, they will likely be satisfied with the instructor and vice-versa. For this reason an assumption can be made that a higher total score across the 28-question survey means a higher level of satisfaction.

```
> d1Q<- d1[,6:33]
> d1Q$tot<-rowSums (d1Q, na.rm = FALSE, dims = 1)
> summary(d1Q$tot)

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  28.00   72.00   86.00   89.21  112.00  140.00
```

Interpreting the histogram below it can be seen that over 75% of the students surveyed fall in the group of being neutral to ablove satisfied, with neutral to satisfied be greater than 70 total score. The median score is 89, which indicates that greater than 50% can be considered as closer to satisfied (in both the course and instructor assuming course satisfaction and instructor satisfaction is correlated - This would require further correlation analysis which will be done tie permitting).



Next analysis is to understand the series of questions answered by the students. There are 28 questions, each answered from 1 (very bad) to 5 (very good). First, we plot a boxplot of Q1 to Q28 to how students have responded for each question. From the graph, we can see that very less students have given completely bad (Rating 1) for Question Q14, Q15, Q17, Q19 - Q22, Q25 which can be used for further analysis by interpreting the answers to the questions. And, positive responses for Q17-22 reflect very well on the behavior, personality, commitment and teaching skills of all instructors (and further filter can be done to focus on each individual instructor.



## Data Preparation

**Import the Dataset**

The dataset provided was “turkiye-student-evaluation_generic.csv”

```
#load data
setwd("csda1010")
d1=read.csv("turkiye-student-evaluation_generic.csv",header = TRUE)
head(d1)
#Explore non survey question variables
plot(d1[,1:5])
```

**Missing Values**

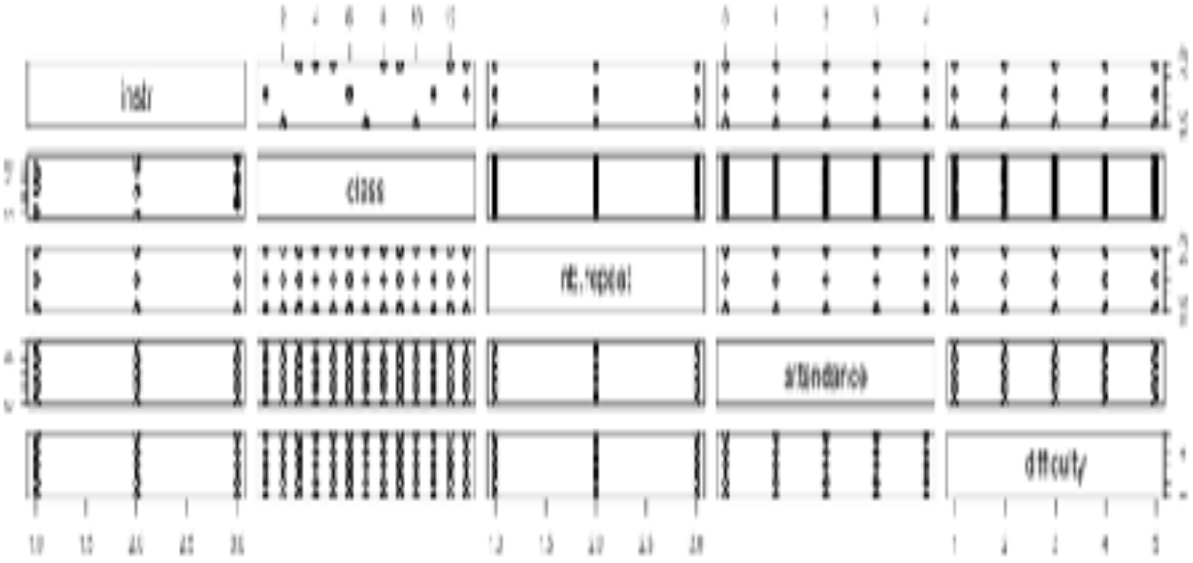
Any missing value in the data must be removed or estimated. Data set has **no missing values**, the following was checked:

```
table(is.na(d1))
str(d1)
dim(d1)
summary(d1)
d1 <- na.omit(d1)
```

**Scaling**

The data must be **standardized (i.e., scaled)** to make variables comparable. Recall that, standardization consists of transforming the variables such that they have mean zero and standard deviation one. The following columns were selected for k-means clustering: d1Q<- d1[, 6:33]

These are the responses to the student questionnaire and the responses are based on a {1,2,3,45} Likert case and therefore do not need to be normalized.

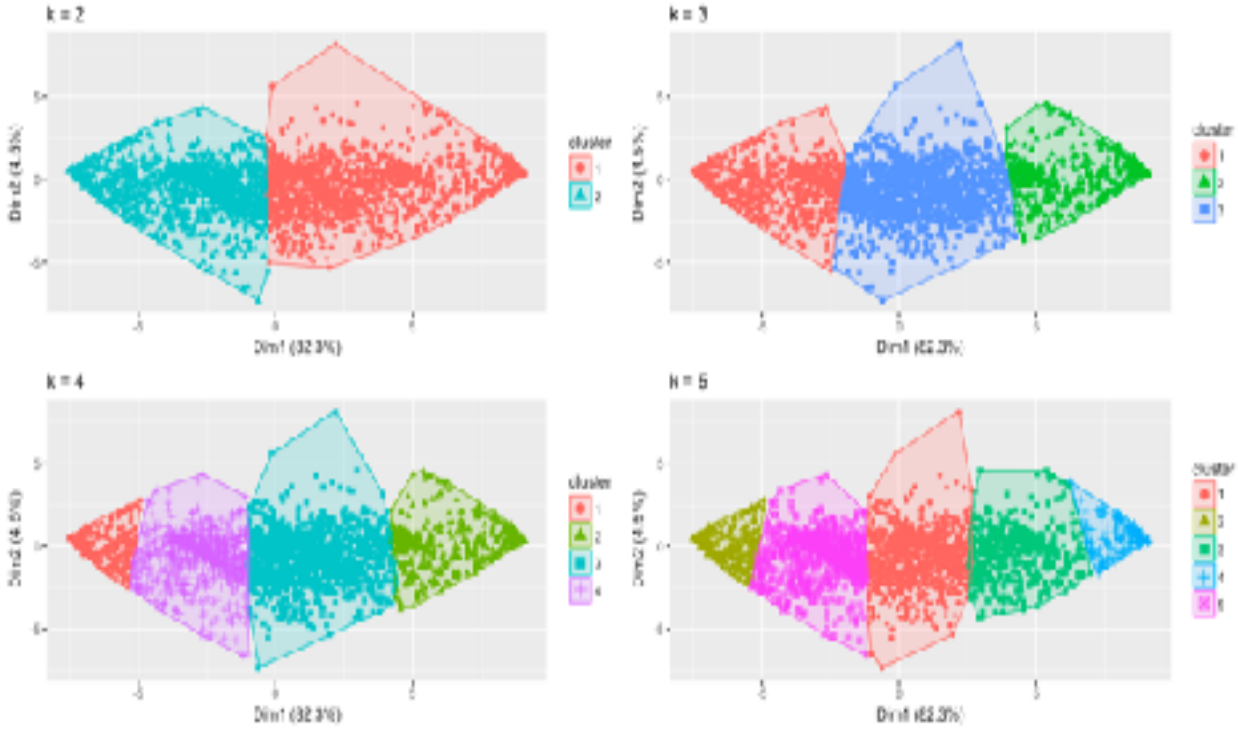


## Modelling

**K-Means Clustering**

Post preparation of the dataset we started with one of the clustering algorithms “K-Means Algorithm” for our analysis. In a nutshell, this algorithm will cluster each observation based on their distance to the nearest *centroid*. *Centroid* is randomly picked from the observations (a sample/row) initially and will be moved on each iteration based on the *mean* (for continuous features) of the observations that were assigned to it. The number of *centroid* will be the number of the clusters.

For the initial analysis we ran with 2,3,4 and 5 centroids, as shown below:

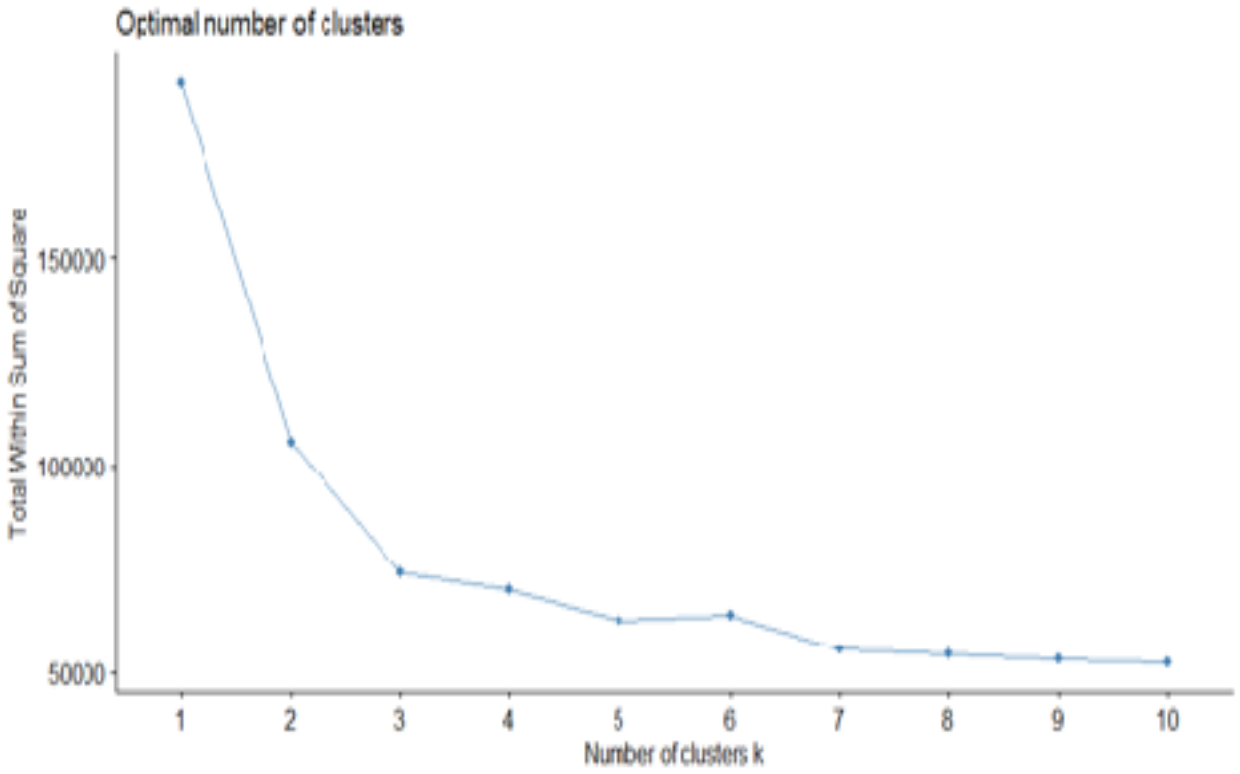



But we cannot arrive at a conclusion of which one is the optimal cluster based on the above plots.

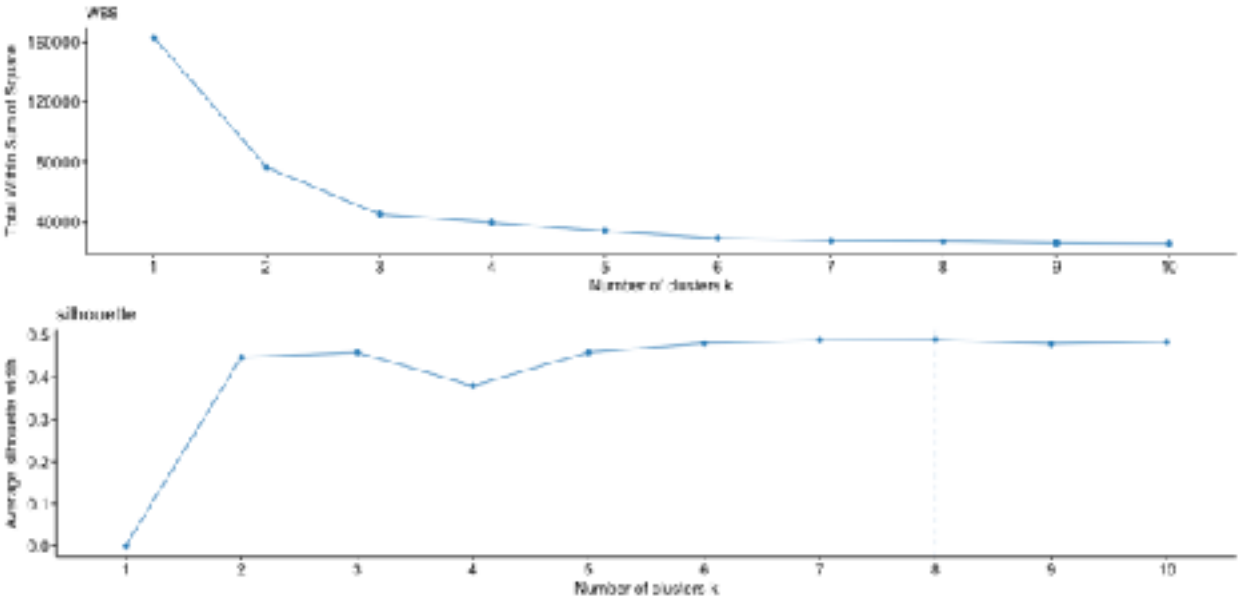
As we have to decide on the optimal number of clusters the following three most popular methods[Elbow method, Silhouette method and Gap statistic] can be used to determine the optimal clusters. For simplicity we will be using the  Elbow method to determine the optimal clusters.

This method picks the best number of clusters (k) by computing the Sum of Squared Error of each cluster (also called distortion). The smaller the value of SSE means the distance between the centroid and the observation in the cluster is closer.  Our goal is to find the cluster where the distortion decreases rapidly. In simple terms the location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

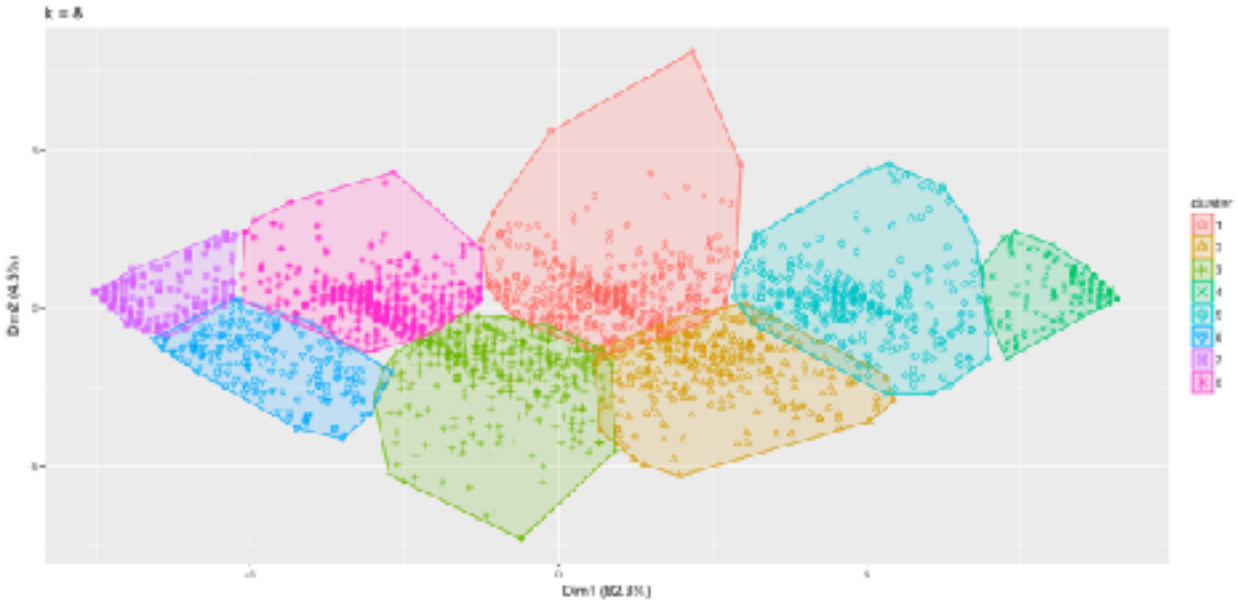
The “elbow” seems to be located at *k* = 3. This mean the distortion will not be decreased significantly if we tell the algorithm to use a larger number of clusters.



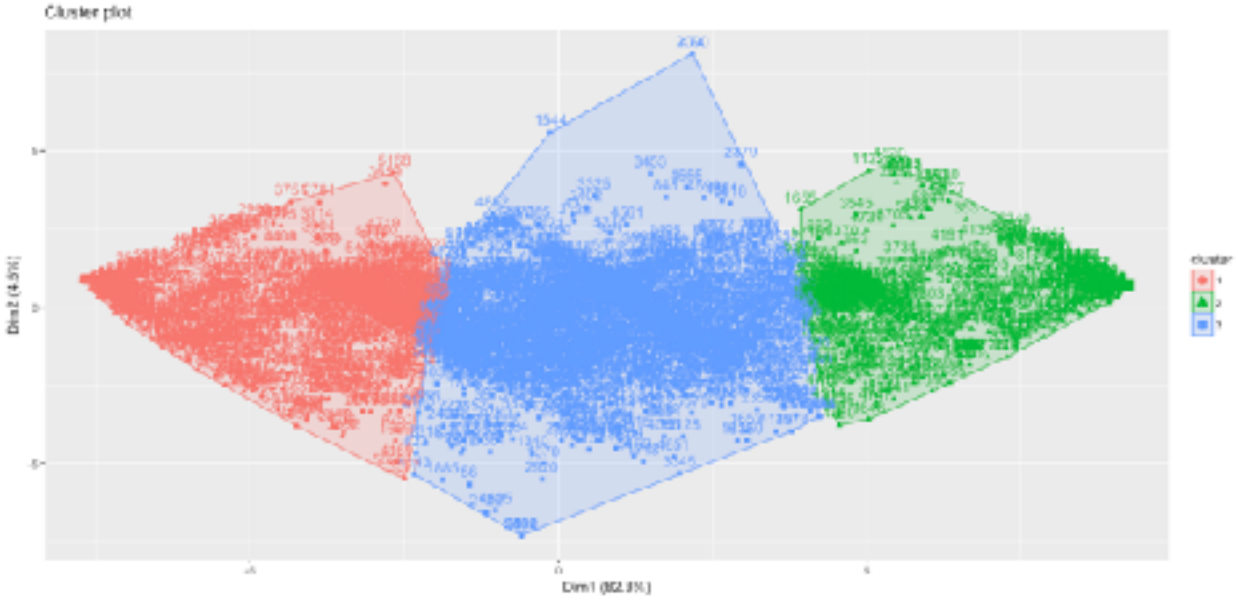

We also tried the silhouette method to compare the results of the clusters:



However, K=8 generates clusters with overlap:



We train the *K-Means* with 3 clusters and visualize the result by scatter plot.



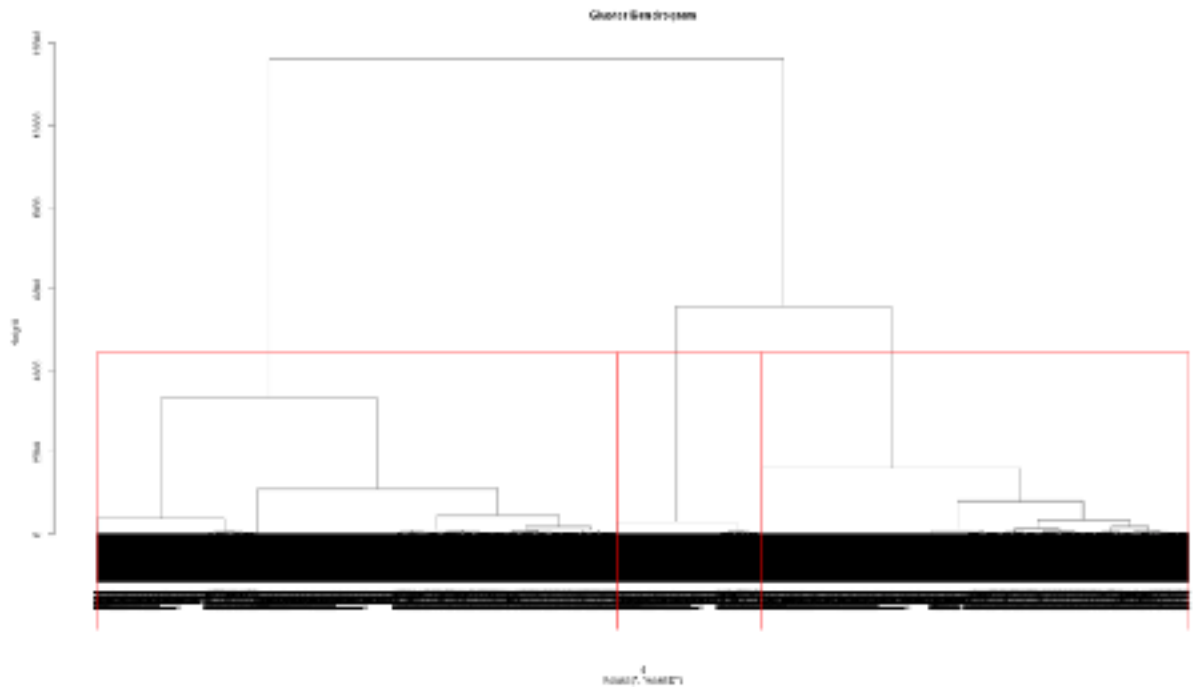

Looking at a sample of questions (Q1:2, Q26:Q28), the kmeans["centers"], we can see that the 2nd cluster represents the neutral response (questions around 3 on the Likert scale), with a center value around 0. Cluster 3 falls on the positive side and represents responses in the 4:5 range, and Cluster 1 falls negative representing closer to the 1:2 response range. So we have 2358 students who have given negative ratings overall , 2222 students with positive ratings and 1240 students with neutral response

```
Q1 Q2          Q26        Q27          Q28
C1 -1.14   -1.3001124 1.4106873 -1.3535883-1.42347651
C2 -0.2231965 -0.1912916-0.1186967 -0.1365980 -0.06758346
C3  0.8698672   0.9230285  0.9076448   0.8948961  0.86073397

Within cluster sum of squares by cluster:
[1]  9037.411 20874.671 15330.344  (between_SS / total_SS = 72.2 %)

Available components:
[1] "cluster"      "centers" "totss"        "withinss" "tot.withinss" "betweenss"
[7] "size"         "iter" "ifault"

Clustering vector:
  [1] 2 2 3 2 1 3 3 3 3 3 2 2 3 1 1 3 1 3 1 3 1 1 2 2 3 1 3 1 3 3 2 2 3 3 1 2 1 3 1 2 3 3 2 3 3 2 3 3

 Q1         Q2 Q3         Q4 Q5 Q6         Q7 Q8 Q9 Q10   Q11Q12
C1:
C2: -1.14 -1.30 -1.33 -1.28  -1.34 -1.32 -1.30 -1.29  -1.31-1.35 -1.32 -1.26
C3: 0.86   0.92 0.88 0.90  0.94   0.91   0.93   0.93   0.89 0.96 0.88   0.90
Within cluster sum of squares by cluster:
[1]  9037.411 20874.671 15330.344  (between_SS / total_SS = 72.2 %)
```

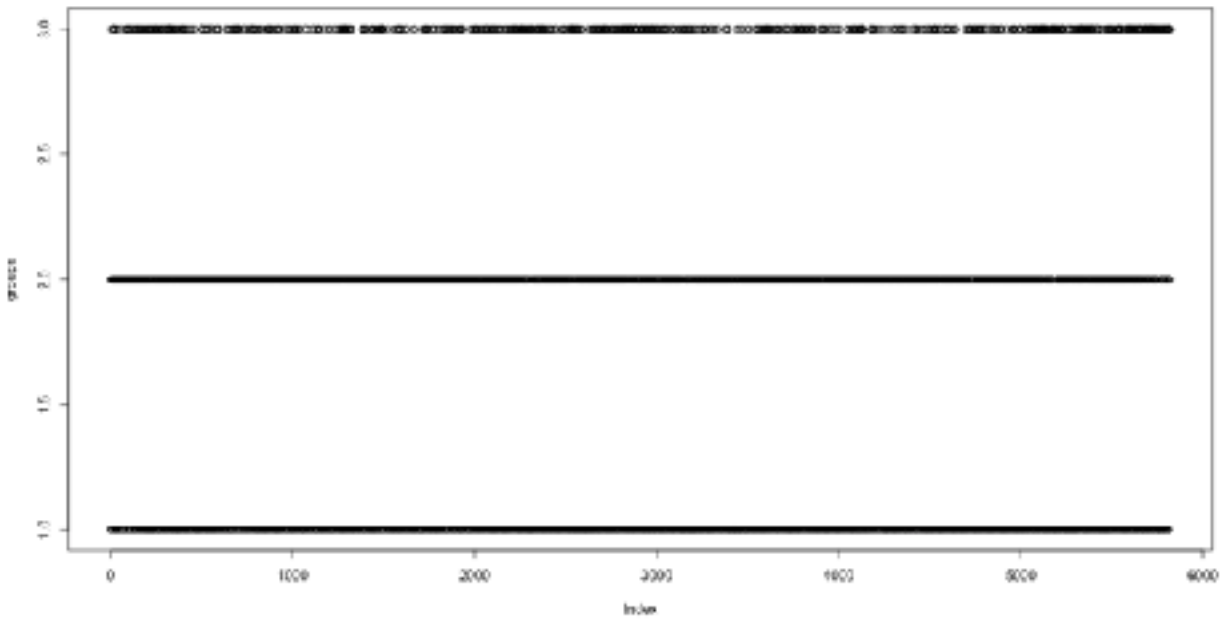
**Hierarchical Clustering**

Next, we will explore with Hierarchical clustering which involves creating clusters that have a predetermined ordering from top to bottom. This algorithm will assign each observation as individual cluster and merge those clusters based on their distance (similarity) pair by pair, iteratively.

Call: hclust(d = d, method = "ward.D") Cluster method   : ward.D  Distance  : euclidean



> str(groups)
 int [1:5820] 1 1 2 1 3 2 2 2 2 2 ...
> groups
  [1] 1 1 2 1 3 2 2 2 2 2 1 1 2 3 3 2 3 2 3 2 1 3 1 1 2 1 2 1 2 2 1 1 2 2 1 1 1 2 3 1 2 2 1 2 2 1 2 2



PCA using princomp() algorithm (Kmeans does PCA automatically):

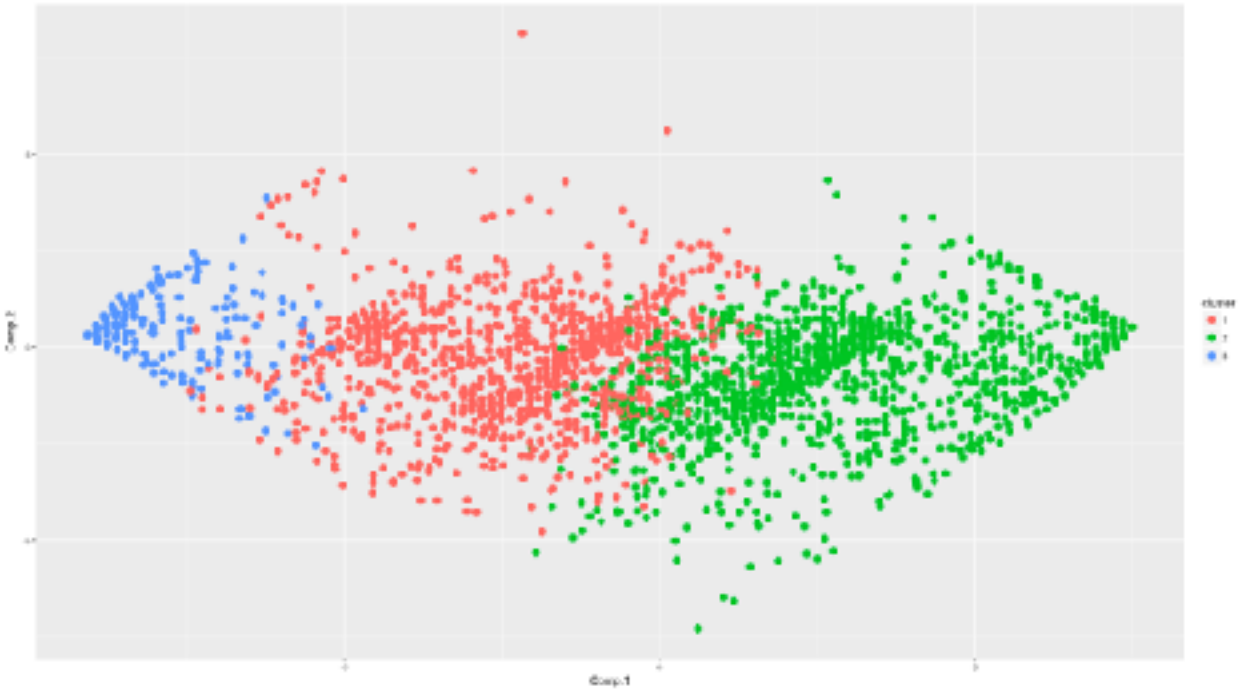
Call: princomp(x = d1Qs)

Standard deviations:
Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
4.7996817 1.1192418 0.6283866 0.6006659 0.5383595 0.5061512 0.4517994 0.4280520 0.4152608 0.3776904
Comp.11 Comp.12 Comp.13 Comp.14 Comp.15 Comp.16 Comp.17 Comp.18 Comp.19 Comp.20
0.3716491 0.3700216 0.3450248 0.3411097 0.3379150 0.3311712 0.3251670 0.3171326 0.3086477 0.3045461
Comp.21 Comp.22 Comp.23 Comp.24 Comp.25 Comp.26 Comp.27 Comp.28
0.2908455 0.2903293 0.2836224 0.2781264 0.2660180 0.2602115 0.2361917 0.2277150

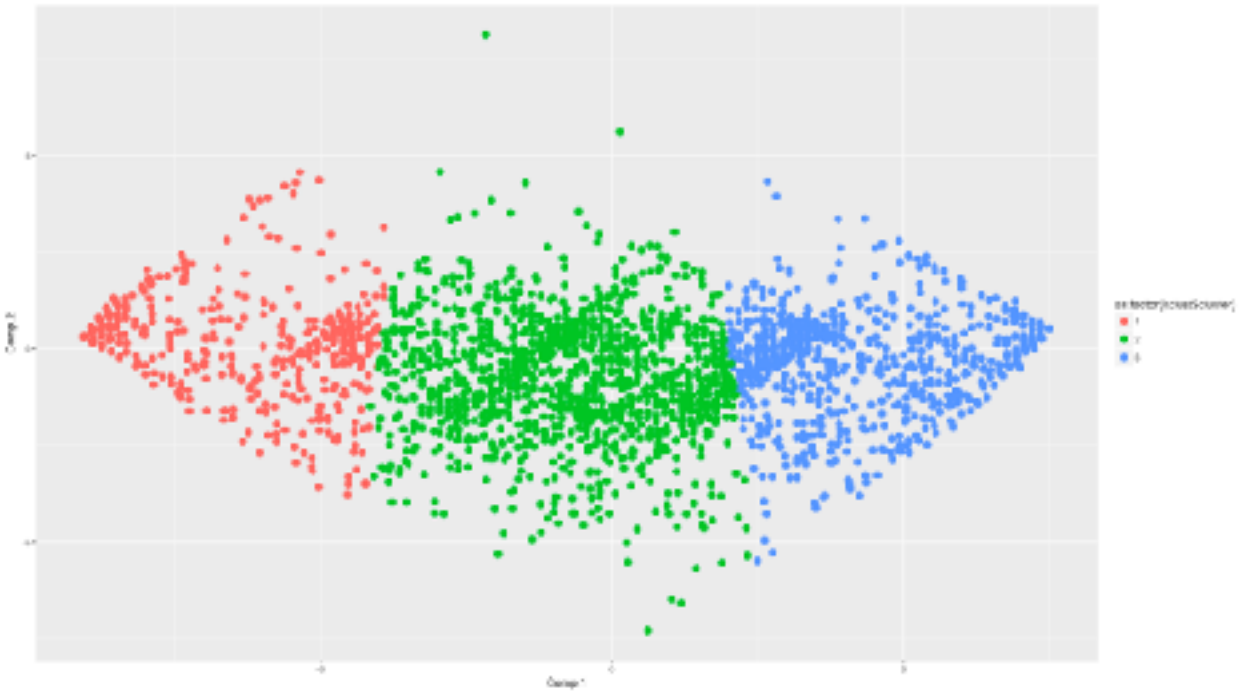
***It is unclear why predict was used in the water_data tutorial, but applying to dataset.***

*pred_pc <- predict(pcmp, newdata=d1Qs)[,1:2]*
comp_dt <- cbind(as.data.table(pred_pc),cluster = as.factor(groups))
ggplot(comp_dt,aes(Comp.1,Comp.2))+ geom_point(aes(color = cluster),size=3)

**Cluster plot using hclust.**



**Cluster plot using k-means.**



It can be seen that using h_clust() results in 3 overlapping clusters, however, using k-means results in 3 non-overlapping clusters.
***→ We do not understand why this is the case and will explore further on the reasoning time permitting***

## Evaluation

It is evident that using a Likert scale {1,2,3,4,5} for survey responses allows for intuitive cluster classifications across multiple questions. Using PCA, the axis are simplified such that values falling on the +ve x dimension demonstrates a more positive response, and similarly a more negative response when falling on the -ve  x dimension. A centred cluster can be considered neutral with the analyst defining the boundaries of neutral (depending on the number of clusters.

In this school survey, neutral was centred around 3, positive was closer to 4 and 5, and unfavourable was assigned for responses closer to 1 and 2. As the questions are divided into two parts, course and instructor evaluation, clusters can be done for each separately and for an overall result.

Using these clusters, it provides:

- A grouping of students who were unsatisfied, neutral or moderately satisfied, and satisfied (very satisfied could be a cluster perhaps if there were more clusters with one of them showing a concentration of results closer to ‘5’).
- Groups of students that can be identified and focused on for course improvements or instructor improvements. By identifying distinct clusters, a further analysis can be done regarding the profile of the students within these clusters:
    - Did the student have the minimum requirements
    - Prior achievements
    - Ability level (related to perceived course difficulty)
    - Instructor incompatibility
    - Other social factors
- Conversely, students who responded favourable can be profiled and further questioned about what was good about the instructor, or reasons why the course was favourable.

Overall, the students responded favourably (given by the distribution of overall score sum), whichat a high level does not raise any red flags. Improvements can be made by focusing on the 1st cluster which responded unfavourably.

Even finer granularity analysis can be done with questione 1:12 and 13:28 to cluster the questions themselves to determine if there are specific questions that identify a specific problem. A survey such as this could be used for both process improvements, or ensuring continued product success, and shifting unfavourable clusters to favourable.

## Deployment

Deployment would consist primarily of communicating our results. We would be presenting the below listed as part of a formal presentation:

- Business Understanding
- Dataset selection, understanding and analysis, preparation and cleansing.
- Model selection, preparation and execution.
- Analysis, interpretation and communication of model results, learnings and insights.

## Monitoring

Regularly communicate with and solicit feedback from our audience.

- Compare model results with future Bigfoot sightings in order to validate findings and reveal new
opportunities for modeling.
- Review any changes to project objectives to maintain relevance of the model.

## Review

Conduct an assessment of what worked well and what didn’t work in order to continuously improve data exploration and modeling processes in order to enhance future work.

- Document and summarize insights and discoveries made throughout the project cycle.
- Conduct interviews with key project participants.

## Code :

```
library(fpc)
library(factoextra)
library(data.table)
library(ggplot2)
library(gridExtra)
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms

#load data
setwd("csda1010")
d1=read.csv("turkiye-student-evaluation_generic.csv",header = TRUE)
head(d1)
#Explore non survey question variables
plot(d1[,1:5])
#ggsave("AllQ.png", plot = plot(d1[,6:33]))
#setDT(d1)
#check missing values
colSums(is.na(d1))
d1Q<- d1[,6:33]
str(d1Q)
head(d1Q)

#scale the variables --> converts data frame o num [1:5820, 1:28] (2-dimensional array - matrix???)
d1Q <-scale(d1Q)
d1Qs <- data.frame(scale(d1Q))
str(d1Qs)
d1Q$tot<-rowSums (d1Q, na.rm = FALSE, dims = 1)
summary(d1Q$tot)
hist(d1Q$tot)

pc <- princomp(d1Qs)
plot(d1Qs) #Error in plot.new() : figure margins too large

# First component dominates greatly. What are the loadings?
summary(pc) # 1 component has > 99% variance
loadings(pc)

#impute missing values with median
## Not required


# Get distance
```

```
dist <- get_dist(d1Qs)
str(dist)
fviz_dist(dist, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))

#KMeans iterations
k2Q <- kmeans(d1Qs, centers = 2, nstart = 25,iter.max = 100)
k3Q <- kmeans(d1Qs, centers = 3, nstart = 25)
k4Q <- kmeans(d1Qs, centers = 4, nstart = 25)
k5Q <- kmeans(d1Qs, centers = 5, nstart = 25)
k8Q <- kmeans(d1Qs, centers = 8, nstart = 25)

# plots to compare
p1Q <- fviz_cluster(k2Q, geom = "point", data = d1Qs) + ggtitle("k = 2")
p2Q <- fviz_cluster(k3Q, geom = "point",  data = d1Qs) + ggtitle("k = 3")
p3Q <- fviz_cluster(k4Q, geom = "point",  data = d1Qs) + ggtitle("k = 4")
p4Q <- fviz_cluster(k5Q, geom = "point",  data = d1Qs) + ggtitle("k = 5")
p8Q <- fviz_cluster(k8Q, geom = "point",  data = d1Qs) + ggtitle("k = 8")

grid.arrange(p1Q, p2Q, p3Q, p4Q, nrow = 2)

set.seed(123)
# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(d1Qs, k, nstart = 10 )$tot.withinss
}

# Compute and plot wss for k = 1 to k = 15
k.values <- 1:15

# extract wss for 2-15 clusters
wss_values <- map_dbl(k.values, wss)

plot(k.values, wss_values,
      type="b", pch = 19, frame = FALSE,
      xlab="Number of clusters K",
      ylab="Total within-clusters sum of squares")

set.seed(123)
fviz_nbclust(d1Qs, kmeans, method = "silhouette")

c1 <- fviz_nbclust(d1Qs, kmeans, method = "wss") + ggtitle("wss")
c2 <- fviz_nbclust(d1Qs, kmeans, method = "silhouette") + ggtitle("silhouette")

grid.arrange(c1,c2)
```

```
#### From water_data tutorial

#Hierarchical Clustering
d <- dist(d1Qs,method = "euclidean") #distance matrix
h_clust <- hclust(d, method = "ward.D") #clustering
plot(h_clust) #dendrogram
rect.hclust(h_clust,k=4)

#extract clusters
groups <- cutree(h_clust,k=3)
str(groups)
groups
plot(groups)

#pca
pcmp <- princomp(d1Qs)
pred_pc <- predict(pcmp, newdata=d1Qs)[,1:2]

comp_dt <- cbind(as.data.table(pred_pc),cluster = as.factor(groups))
ggplot(comp_dt,aes(Comp.1,Comp.2))+
  geom_point(aes(color = cluster),size=3)
kclust <- kmeans(d1Qs,centers = 3,iter.max = 100)

ggplot(comp_dt,aes(Comp.1,Comp.2))+
#ggplot(comp_dt,aes(x=d1$class,d1$instr)) +
  geom_point(aes(color = as.factor(kclust$cluster)),size=3, position = 'jitter')

tunek <- kmeansruns(d1Qs,krange = 1:10,criterion = "ch")
tunek$bestk #3
tunekw <- kmeansruns(d1Qs,krange = 1:10,criterion = "asw")
tunekw$bestk #4

kclust1 <-data.frame(kclust)
```

## APPENDIX

# Pre-Project Work: Learning How Clustering works (in R)

To better understanding of how the clustering algorithm works we created a very simple test questionnaire dataset of 5 Questions and 10 observations (students). We are adding this to the report as it helped us understand at a very basic level how the algorithms were working and to understand if in fact results and analysis from the project dataset ("Turkey school survey.csv") was in fact directionally correct.

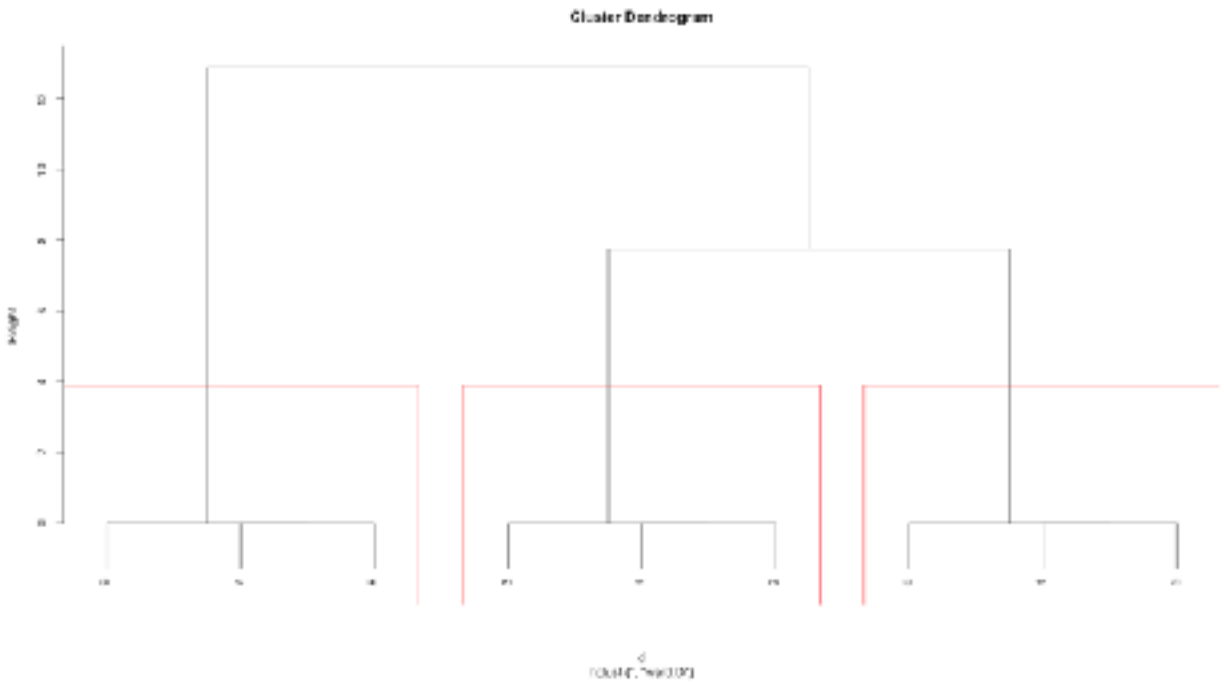
*Please note through this report, we highlight open questions that we have from results we have not fully understood → these questions arose from our own trial and error analysis, learnings from various rblog and kaggle code and analysis sources, use of various R functions for clustering, interpreting charts produced and resulting numeric and statistical values.*
*→ We would like to follow up on and investigate further, time permitting (and of course beyond completion of this module).*

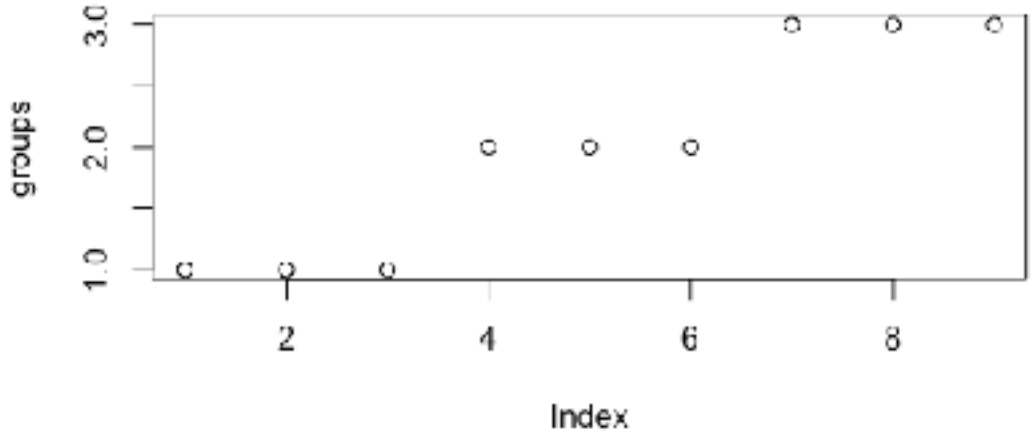
<u>Hierarchical Clustering grouped in 3 boxes:</u>

Based on the following oversimplified dataset:

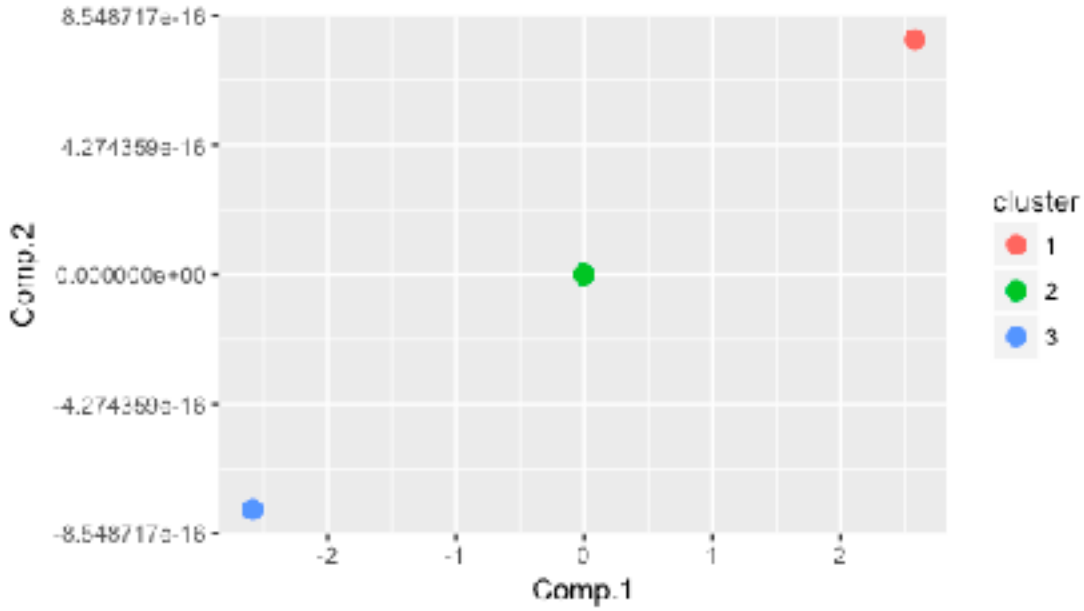
testc

| | Q1 | Q2 | Q3 | Q4 | Q5 |
|---|---|---|---|---|---|
| 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 | 6 | 6 |
| 6 | 6 | 6 | 6 | 6 | 6 |
| 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

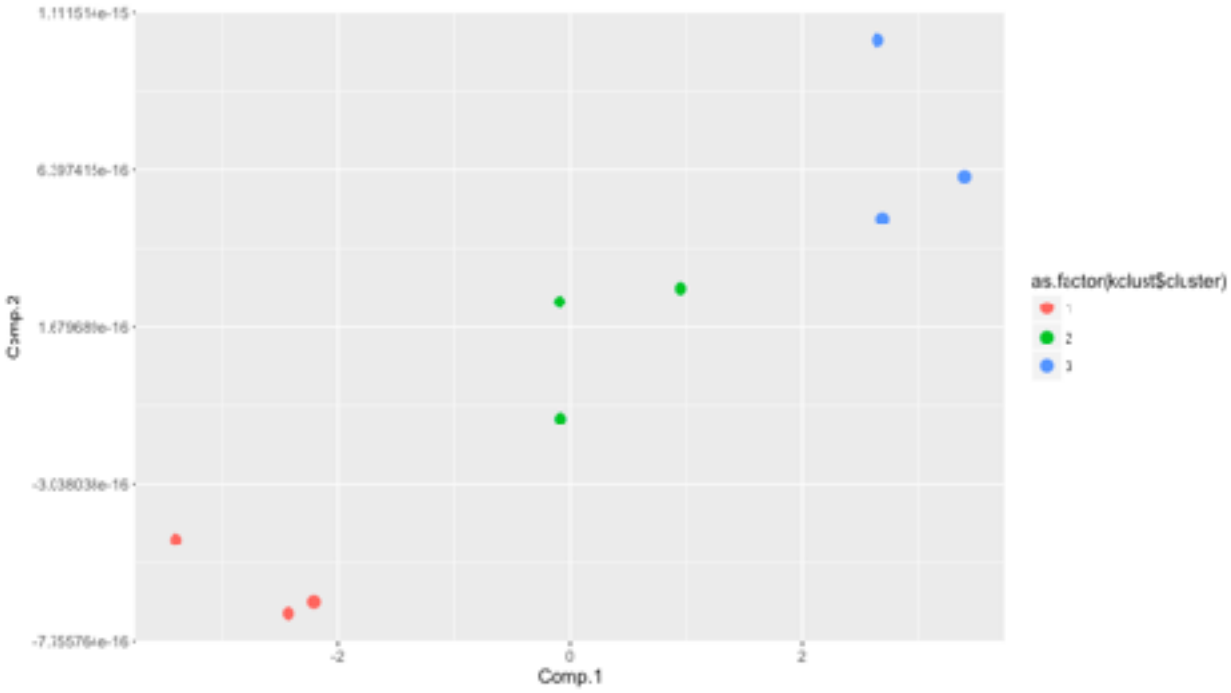



Where groups fall on the PCA component axis:




Clustering of observations (each student that was surveyed:




k-Means which outputted 3 clusters:

Cluster 1 was everyone who answered ‘5’, Cluster 2 answered ‘3’ and cluster 3 answered ‘1’

List of 9

$ **cluster : int [1:9] 1 1 1 2 2 2 3 3 3**
$ centers : num [1:3, 1:5] 1.15 0 -1.15 1.15 0 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:3] "1" "2" "3"
.. ..$ : chr [1:5] "Q1" "Q2" "Q3" "Q4" ...
$ totss : num 40
$ withinss : num [1:3] 0 0 0
$ tot.withinss: num 0
$ betweenss : num 40
$ size : int [1:3] 3 3 3
$ iter : int 1
$ ifault : int 0
- attr(*, "class")= chr "kmeans"

> kt
K-means clustering with 3 clusters of sizes 3, 3, 3:

**Cluster means:**

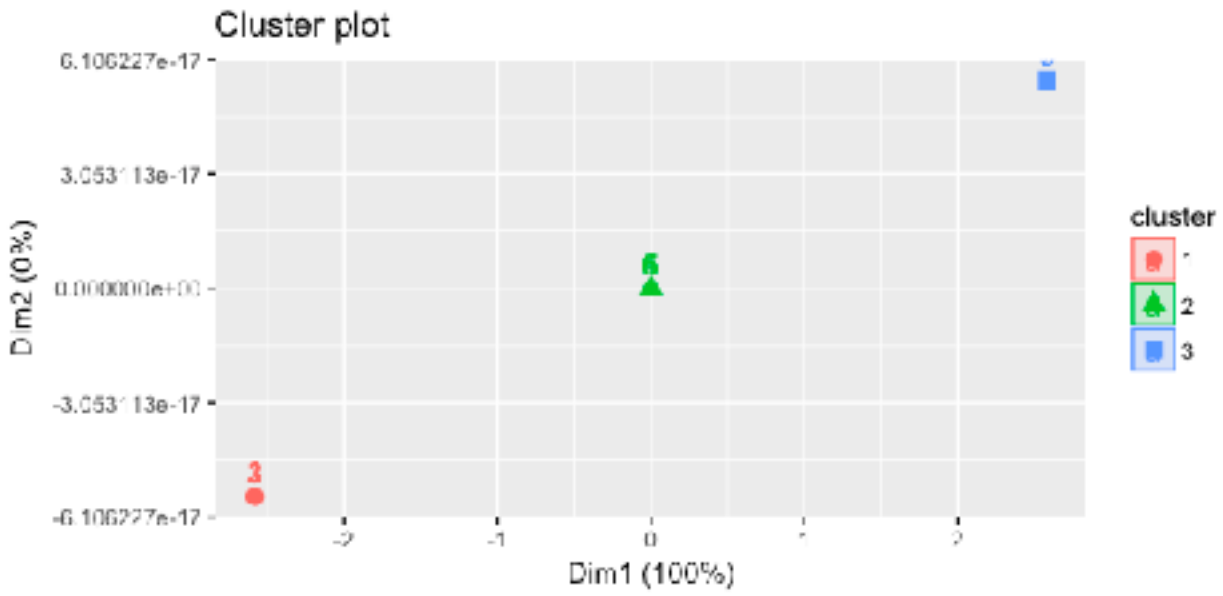
| | Q1 | Q2 | Q3 | Q4 | Q5 |
|---|---|---|---|---|---|
| 1 | 1.154701 | 1.154701 | 1.154701 | 1.154701 | 1.154701 |
| 2 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 3 | -1.154701 | -1.154701 | -1.154701 | -1.154701 | -1.154701 |

**Clustering vector:**
[1] 1 1 1 2 2 2 3 3 3

Within cluster sum of squares by cluster:
[1] 0 0 0 (between_SS / total_SS = 100.0 %)

Available components:
[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss"
[7] "size" "iter" "ifault"
>

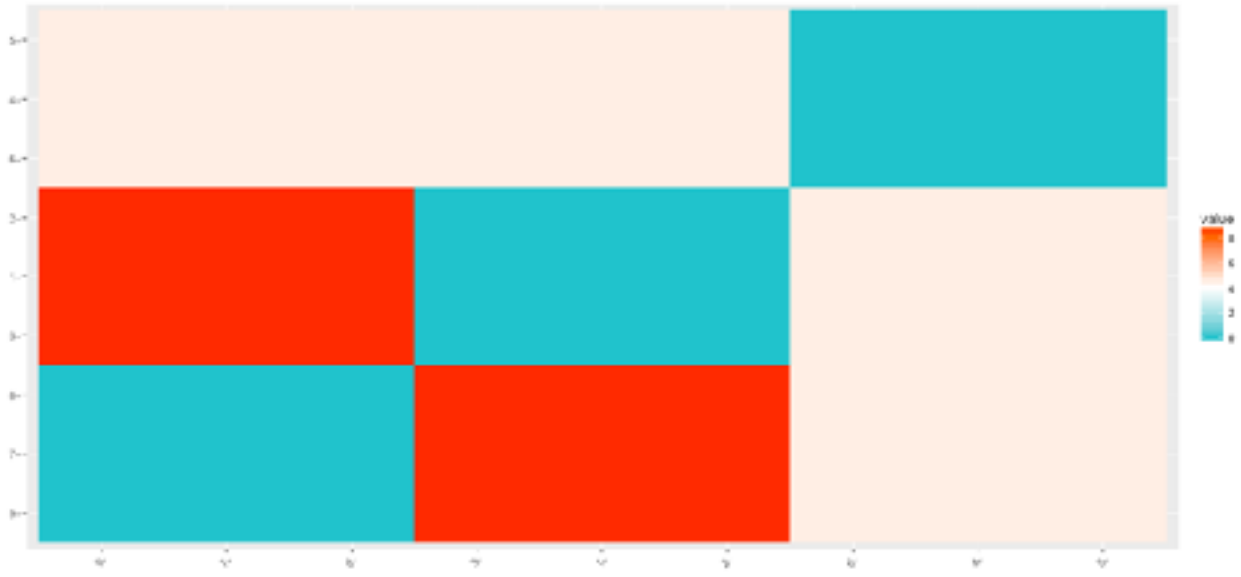
And plotted the clusters:




**Distance Calculations:**

Plotted the distance calculations, ***but would like to better understand the distance matrix that was generated:***

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 2 | 0.000000 | | | | | | | |
| 3 | 0.000000 | 0.000000 | | | | | | |
| 4 | 4.472136 | 4.472136 | 4.472136 | | | | | |
| 5 | 4.472136 | 4.472136 | 4.472136 | 0.000000 | | | | |
| 6 | 4.472136 | 4.472136 | 4.472136 | 0.000000 | 0.000000 | | | |
| 7 | 8.944272 | 8.944272 | 8.944272 | 4.472136 | 4.472136 | 4.472136 | | |
| 8 | 8.944272 | 8.944272 | 8.944272 | 4.472136 | 4.472136 | 4.472136 | 0.000000 | |
| 9 | 8.944272 | 8.944272 | 8.944272 | 4.472136 | 4.472136 | 4.472136 | 0.000000 | 0.000000 |



**Self-Learning Exercise - Test Code**

```
#DATALINES  table creation
t<- read.table(text="
Q1      Q2      Q3      Q4      Q5
5       5       5       5       5
5       5       5       5       5
5       5       5       5       5
3       3       3       3       3
3       3       3       3       3
3       3       3       3       3
1       1       1       1       1
1       1       1       1       1
1       1       1       1       1
", header=TRUE)


#Data Preparation


#Scale the data
t<-scale(t)
plot(t[,1:5])


#Distance Calculaltion
td <- get_dist(t)
str(td)
```

```
fviz_dist(td, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
kt <- kmeans(t, centers = 3, nstart = 25,iter.max = 100)
str(kt)
fviz_cluster(kt, data = t)


#Hierarchical Clustering
d <- dist(t,method = "euclidean") #distance matrix
h_clust <- hclust(d, method = "ward.D") #clustering
plot(h_clust) #dendrogram


rect.hclust(h_clust,k=3)


#extract clusters
groups <- cutree(h_clust,k=3)
str(groups)
groups
plot(groups)


#pca
pcmp <- princomp(t)
pred_pc <- predict(pcmp, newdata=t)[,1:2]


comp_dt <- cbind(as.data.table(pred_pc),cluster = as.factor(groups))
ggplot(comp_dt,aes(Comp.1,Comp.2))+
  geom_point(aes(color = cluster),size=3)


#k-Means
kclust <- kmeans(t,centers = 3,iter.max = 100)


ggplot(comp_dt,aes(Comp.1,Comp.2))+
#ggplot(comp_dt,aes(x=d1$class,d1$instr)) +
  geom_point(aes(color = as.factor(kclust$cluster)),size=3, position = 'jitter')
```