DATA SCIENCE AND MACHINE LEARNING (MSC)

DAMA51: Foundations in Computer Science

Academic Year: 2022–2023

#4 Written Assignment	
Submission Deadline	**Wed, 26 April 2023, 11:59 PM**
Student Name:	**Kritikos Antonios**

Remarks

The deadline is definitive.

An indicative solution will be posted online along with the returning of the graded assignments.

The assignment is due via the STUDY submission system. **You are expected to turn in a document (.DOC, .ODT, .PDF) and a compressed (.ZIP, .RAR) file containing all your work:**

- **1 document file (this document) with the answers to all the questions, along with the R code and the results of the execution of the code**
- **1 compressed file with 3 R scripts with the code that answers to each one of the problems to the Topics 3 and 5.**

You should not make any changes in the written assignment file other than providing your own answers. You should also type all of your answers into Word and not attach any handwritten notes as pictures into your work otherwise a 5% reduction of your final grade will be applied. Make sure to name all the files (ZIP file, DOC file and R script files) with **your last name first followed by a dash symbol and the names of each component at the end**. For example, for the student with the last name Aggelou the files should be named as follows: Aggelou-HW4.zip, Aggelou-HW4.doc, Aggelou-Topic3.R, Aggelou-Topic4.R, and Aggelou-Topic5.R. The R script files should automatically run with the **source** command and generate the correct results. Also, please include comments before each command to explain the functionality of the command that follows. In the computations, use three decimal places.

Topic	Points	Grades
1. Online Quiz	40	
2. Article review	5	
3. Prototype-based Clustering	20	
4. Hierarchcal based Clustering using R	20	
5. Itemset Mining and Association Rules using R	20	
TOTAL	**105 (max 100)**	**/100**

Topic 1: Quiz

(40 points) Complete the corresponding online quiz available at:

https://study.eap.gr/mod/quiz/view.php?id=24568

You have one effort and unlimited time to complete the quiz, up to the submission deadline.

Topic 2: Article Review

The article "The planning and care of data" (https://dl.acm.org/doi/10.1145/3532633) makes a point about what drives complexity in software systems compared to data-oriented projects. What is this comparison? Why does the author believe that modern start-ups are less inclined to treat data engineering properly?

Note: You should write up your answer to a maximum of 100 words. Any text in excess of 100 words will not be taken into consideration.

(5 points)

The article goes through a historical reference of how the hardware has evolved the last 70 years and how much importance this holds in the storage of data. Then, it continues on how the abundance of storage tends to make people rely on it and not thinking too much of organizing data to make it easier to go through or always have them ready-to-process by an algorithm, since there can always be more storage to dump data and pull them for use when needed. Finally, it mentions the inevitable cleanup which will happen either by corporate or by legal pressure.

Topic 3: Prototype-based and k-means Clustering

(20 total points) This topic will use the ***seeds*** dataset, which contains data about the physical properties of the internal kernel structure of various wheats. The wheats come from three different varieties.

Read the data using a command like the one below:

```
seeds <- read.csv("seeds_dataset", header = TRUE)
```

Note about reproducibility for the k-means algorithm: Since k-means will pseudo-randomly initialize its state, make sure that exactly before using the k-means algorithm, you call `set.seed(123)`.

All the topics are expected to be answered using R unless explicitly stated otherwise.

(a) (6 points) Perform a cluster analysis with the k-means algorithm. The desired number of clusters is 3. For your analysis use all the features of the dataset except columns `seeID` and `seedType`. Ensure that the dataset is scaled; if not, scale it so that the mean is0 and the standard deviation is 1.

Provide the scaled values of the attribute `perimeter` for each cluster prototype (centroid).

Find the cluster prototype that the data instances of rows 9, 55 and 189 belong to.

(Fill all values)

Answer:

```
library(cluster)
library(factoextra)

seeds <- read.csv("C:\\Users\\antonisk\\Desktop\\DAMA\\DAMA51\\DAMA51 - 4th Assignment\\seeds_dataset.csv", header = TRUE)
seeds_scaled <- scale(seeds[, -c(1, 8)])  # Excluding seeID and seedType columns

set.seed(123)
kmeans_result <- kmeans(seeds_scaled, centers = 3)

perimeter_centroids <- kmeans_result$centers[, "perimeter"]
perimeter_centroids[1]
perimeter_centroids[2]
perimeter_centroids[3]
```

```
row_9_cluster <- kmeans_result$cluster[9]
row_9_cluster
row_55_cluster <- kmeans_result$cluster[55]
row_55_cluster
row_189_cluster <- kmeans_result$cluster[189]
row_189_cluster

dist_12 <- sqrt(sum((kmeans_result$centers[1, "perimeter"] - kmeans_result$centers[2, "perimeter"])^2))
dist_12
dist_13 <- sqrt(sum((kmeans_result$centers[1, "perimeter"] - kmeans_result$centers[3, "perimeter"])^2))
dist_13
dist_23 <- sqrt(sum((kmeans_result$centers[2, "perimeter"] - kmeans_result$centers[3, "perimeter"])^2))
dist_23
```

Value of attribute "perimeter" for each cluster prototype:	*Cluster prototype of data instances:*	*Euclidean distance between centroids:*
Perimeter of cluster 1 prototype= -1.004188	Cluster for data row 9 = 3	dist(1,2)= 2.271772
Perimeter of cluster 2 prototype= 1.267583	Cluster for data row 55 = 3	dist(1,3)= 0.8235492
Perimeter of cluster 3 prototype= -0.1806392	Cluster for data row 189 = 1	dist(2,3)= 1.448222

(b) (8 points) Count how many wheats are assigned to each cluster.

For achieving this, first create a new vector to hold all the assignments (i.e., the vector of integers indicating the cluster to which each point is allocated) and, in this vector, rename cluster `1` to cluster `2`, and cluster `2` to cluster `1`. **(2 points)**

Then, using a confusion matrix such as the one below, make a comparison of this vector with the attribute `seedType`. Compare the values of the diagonal elements against the other elements. **(2 points)**

Count how many wheats have been falsely assigned to an incorrect cluster and calculate the accuracy of clustering. **(2 points)**

Then, using **pen and paper**, calculate the precision and recall rates for cluster `1`. **(2 points)**

(Fill all values)

Answer:

```
cluster_assignments <- kmeans_result$cluster

cluster_assignments[cluster_assignments == 1] <- 2
cluster_assignments[cluster_assignments == 2] <- 1

confusion <- table(cluster_assignments, seeds$seedType)

false_assignments <- sum(confusion[1, -1])

accuracy <- sum(diag(confusion)) / sum(confusion)

precision_cluster_1 <- confusion[2, 2] / sum(confusion[2, 2:3])
precision_cluster_1
recall_cluster_1 <- confusion[2, 2] / sum(confusion[2:3, 2])
recall_cluster_1
confusion
```

	cluster		
seedType	1	2	3
1	1		69
2	65		5
3	70		0

Precision rate for cluster 1 = 0.007352941

Recall rate for cluster 1 = 0.01428571

false_assignments: 204

accuracy: 0.02857143

(c) (6 points)

Calculate the average silhouette for the k-means clustering that has been performed (i.e., with k=3) (note that you first need to have the `cluster` package installed). Repeat the calculation for a clustering with 4 clusters (i.e., k=4) and confirm that the average silhouette is lower. **(3 points)**

Using the function `fviz_cluster()` of the `factoextra` R package (you will need to have it installed first), visualize the k-means clusters for k=3 as well as for k=4. Based on the plots, comment whether the clusters are well separated. **(3 points)**

Answer:

Average Silhouette (3 clusters) = 0.3783167

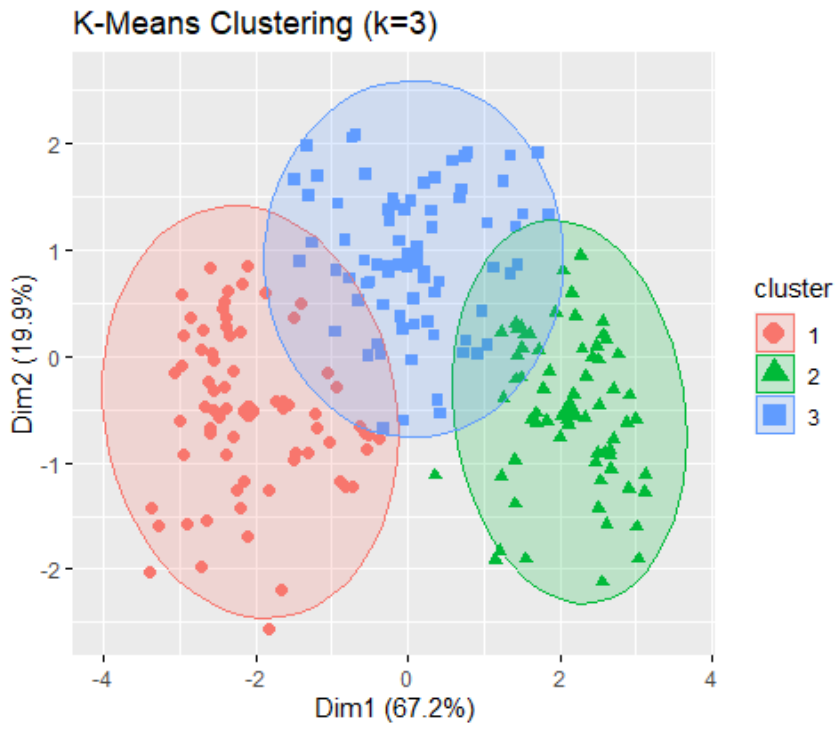
Average Silhouette (4 clusters) = 0.3471662

Comment on whether the clusters are well separated: We can see that for k=3 the clusters are relatively well-separated but for k-4 they are not well-separated

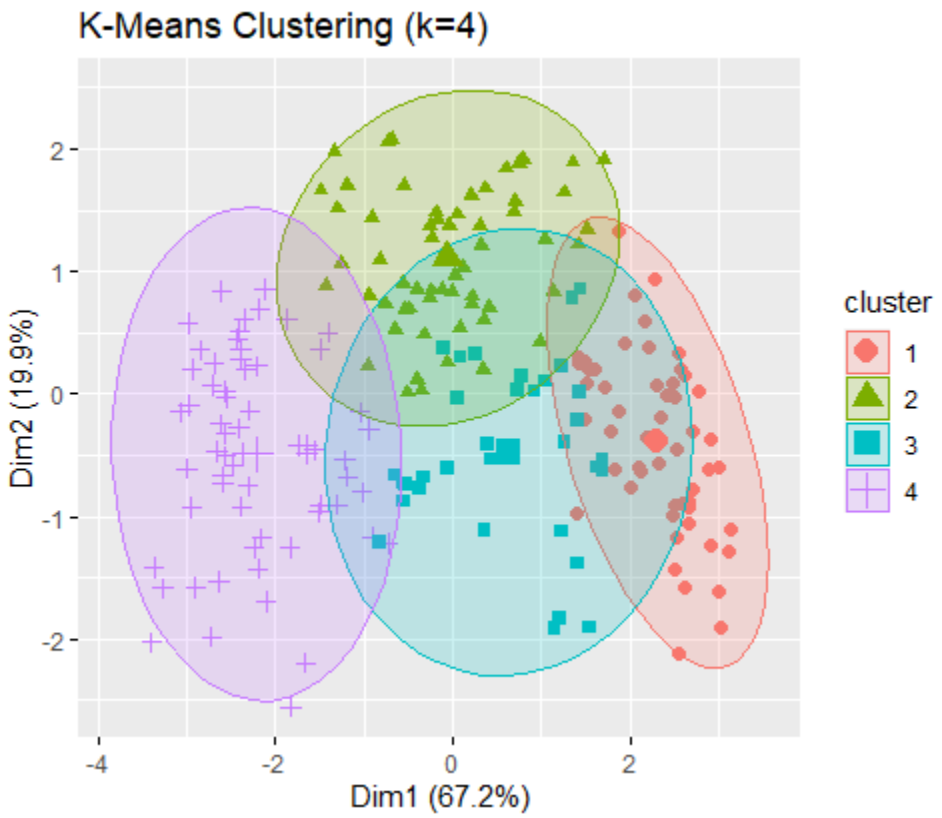
```
kmeans_clusters_3 <- kmeans(seeds_scaled[, -c(1, ncol(seeds_scaled))], centers = 3, nstart = 25)

silhouette_avg_3 <- silhouette(kmeans_clusters_3$cluster, dist(seeds_scaled[, -c(1, ncol(seeds_scaled))]))


kmeans_clusters_4 <- kmeans(seeds_scaled[, -c(1, ncol(seeds_scaled))], centers = 4, nstart = 25)

silhouette_avg_4 <- silhouette(kmeans_clusters_4$cluster, dist(seeds_scaled[, -c(1, ncol(seeds_scaled))]))


avg_silhouette_3 <- mean(silhouette_avg_3[, "sil_width"])
```

```
avg_silhouette_3
avg_silhouette_4 <- mean(silhouette_avg_4[, "sil_width"])
avg_silhouette_4

fviz_cluster(kmeans_clusters_3, data = seeds_scaled[, -c(1, ncol(seeds_scaled))],
             ellipse.type = "norm", geom = "point", pointsize = 2, title = "K-Means Clustering (k=3)")

fviz_cluster(kmeans_clusters_4, data = seeds_scaled[, -c(1, ncol(seeds_scaled))],
             ellipse.type = "norm", geom = "point", pointsize = 2, title = "K-Means Clustering (k=4)")
```

Plot (k=3):



Plot (k=4):


K-Means Clustering (k=4)
Dim2 (19.9%)
Dim1 (67.2%)
2
1
0
-1
-2
-4
-2
0
2
cluster
1
2
3
4

Topic 4: Hierarchical based Clustering using R

(20 points) For this topic, you will work on the ***europe_diet*** dataset which can be found here. This dataset includes records on the kilocalories received daily per person from different food categories in several European countries. For all questions, you are requested to provide your R code and the result of its execution in every answer box. All the topics are expected to be answered using R unless explicitly stated otherwise.

a. (**2 points**) Inspect the dataset, set the row names according to the values of the corresponding country column and then, remove this column.

Answer:

```
europe_diet_og <- read.csv("C:\\Users\\antonisk\\Desktop\\DAMA\\DAMA51\\DAMA51 - 4th Assignment\\europe_diet.csv")

europe_diet <- read.csv("C:\\Users\\antonisk\\Desktop\\DAMA\\DAMA51\\DAMA51 - 4th Assignment\\europe_diet.csv")


rownames(europe_diet) <- europe_diet$X


europe_diet_og

europe_diet$X <- NULL

europe_diet
```

b. **(4 points**) Calculate the dissimilarity distance matrices of the dataset using the Euclidean distance method. Then fill in the following table with the distances of Spain, Belgium and Finland to Greece.

Answer:

```
greece_row <- europe_diet["Greece", ]

spain_row <- europe_diet["Spain", ]

belgium_row <- europe_diet["Belgium", ]

finland_row <- europe_diet["Finland", ]


distances_to_greece <- sqrt(sum((spain_row - greece_row)^2))

distances_to_belgium <- sqrt(sum((belgium_row - greece_row)^2))

distances_to_finland <- sqrt(sum((finland_row - greece_row)^2))
```

```
distances_table <- data.frame(Euclidean_distance = c(distances_to_greece,
distances_to_belgium, distances_to_finland),
                              row.names = c("Spain", "Belgium",
"Finland"))
distances_table
```

Euclidian distance	Spain	Belgium	Finland
Greece	270.3442	315.0444	493.8279

c. (**6 points**) Now, perform agglomerative hierarchical clustering using the **Euclidian** dissimilarity distance matrix and for both complete **(2 points)** and single **(2 points)** linkage. Provide the dendrograms of both analyses **(2 points)**.

Answer:

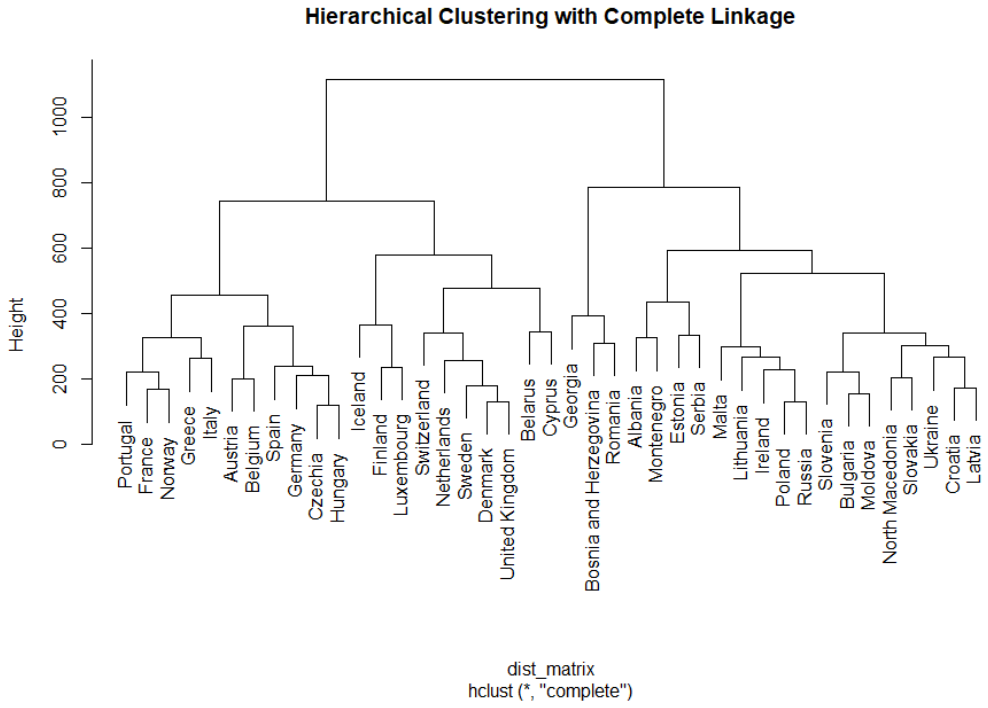
hc_complete <- hclust(dist_matrix, method = "complete")

hc_single <- hclust(dist_matrix, method = "single")

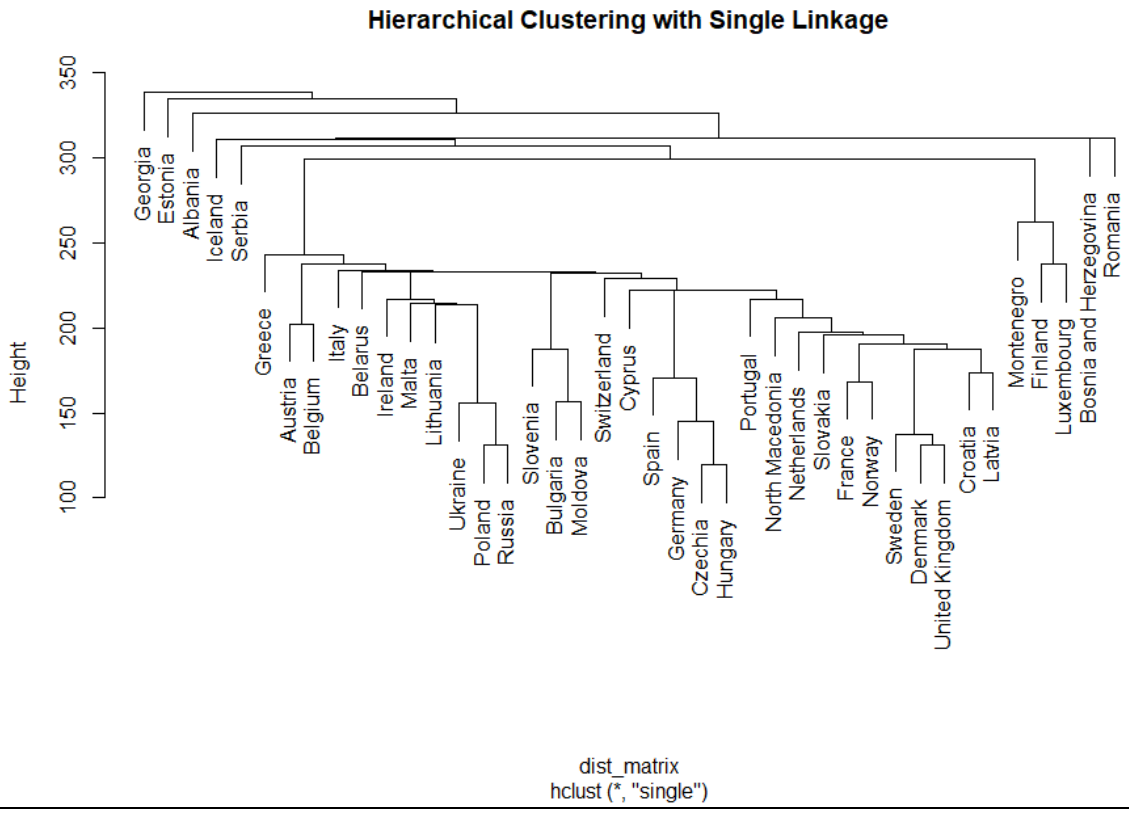
plot(hc_complete, main = "Hierarchical Clustering with Complete Linkage")

plot(hc_single, main = "Hierarchical Clustering with Single Linkage")

Complete linkage:



Single linkage:



d. (**5 points**) Now, based on the complete linkage hierarchical clustering of the previous question, cluster the European countries to 7 different groups **(2 points)**. Using R, identify the countries that have been assigned to the same cluster as i) Switzerland and ii) Norway **(3 points)**.

Answer:

```
clusters_complete <- cutree(hc_complete, k = 7)

switzerland_cluster <- clusters_complete["Switzerland"]
norway_cluster <- clusters_complete["Norway"]

countries_switzerland_cluster <- names(clusters_complete[clusters_complete == switzerland_cluster])
countries_norway_cluster <- names(clusters_complete[clusters_complete == norway_cluster])

cat("Switzerland: ", countries_switzerland_cluster, "\n")
cat("Norway: ", countries_norway_cluster, "\n")
```

OUTPUT:

Switzerland: Belarus Cyprus Denmark Netherlands Sweden Switzerland United Kingdom

Norway: Austria Belgium Czechia France Germany Greece Hungary Italy Norway Portugal Spain

e. (**3 points**) Using R, identify the maximum number of clusters k for which Greece and Cyprus belong to the same cluster.

Answer:

```
max_k <- 1
for (k in 1:length(europe_diet)) {
  clusters <- cutree(hc_complete, k = k)
  if (clusters["Greece"] == clusters["Cyprus"]) {
    max_k <- k
  } else {
    break
  }
}
max_k
```

Requested maximum number of clusters: 3

Topic 5: Itemset Mining and Association Rules using R

(20 points) For this topic, you will work on the ***application_data*** dataset which can be found here. This dataset includes records on the applications that university students have installed on their smartphones. Each record (transaction) includes the set of applications installed by each student. You are requested to provide your R code and the result of its execution in every answer box. All the topics are expected to be answered using R unless explicitly stated otherwise.

Please read the dataset using the following command:

```
appstrans<-read.transactions("path/application_data.csv", format = "basket", sep=",",rm.duplicates=FALSE)
```

a. (**3 points**) Provide the names of all different applications installed.

Answer:

```
library(arules)

appstrans <- read.transactions("C:\\Users\\antonisk\\Desktop\\DAMA\\DAMA51\\DAMA51 - 4th Assignment\\application_data.csv", format = "basket", sep = ",", rm.duplicates = FALSE)


apps <- unique(itemLabels(appstrans))

apps
```

OUTPUT:

```
"Amazon"      "Amazon Prime" "Discord"     "Facebook"    "Hotstar"     "Instagram"   "Meet"
 "Netflix"     "Pinterest"    "SnapChat"    "Spotify"     "Twitter"     "Whatsapp"    "Wynk"
 "Youtube"     "Zoom"
```

b. (**3 points**) Fill in the table below with the number of students who have installed a specific number of applications

Answer:

```
library(arules)


num_apps <- colSums(as(appstrans, "matrix"))
```

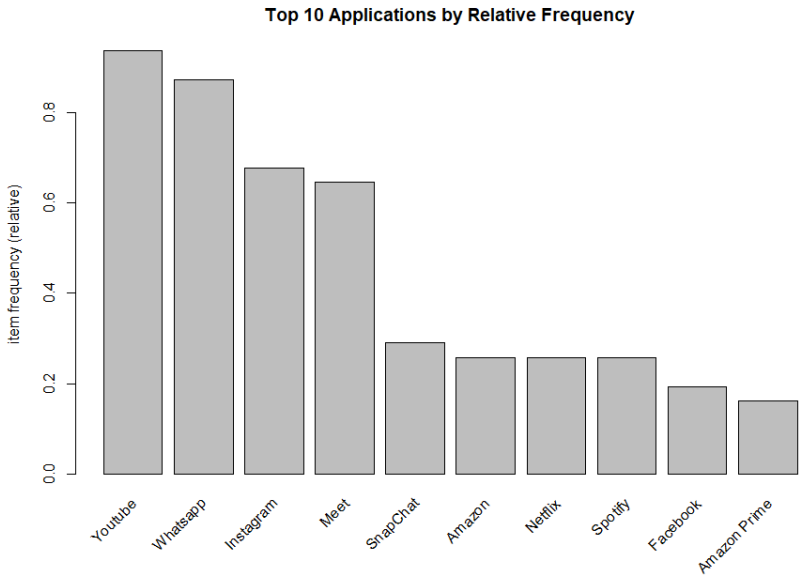
```
result <- data.frame(Number_of_applications_per_student = 1:max(num_apps),
                     Number_of_students = tabulate(num_apps))

print(result)
```

Number of applications/student	Number of students
3	0
5	2
7	0

c. (**2 points**) Now, create an item frequency plot with the top10 applications in terms of relative frequency.

Answer:

```
item_freq <- itemFrequency(appstrans)

item_freq_sorted <- sort(item_freq, decreasing = TRUE)

top10_apps <- head(item_freq_sorted, 10)

itemFrequencyPlot(appstrans, topN = 10, type = "relative", main = "Top 10 Applications by Relative Frequency")
```



d. (**6 points**) Run the apriori algorithm for a minimum support threshold of 0.25, a minimum confidence threshold of 0.8 and minimum of 2 items involved in a rule **(2 points)**. Sort the rules generated according to decreasing value of "*support*" and list only the Top4 of them **(2 points)**. Identify the rule length distribution, i.e. the number of rules with 2 items, 3 items, etc . **(2 points)**

Answer:

```
rules <- apriori(appstrans, parameter = list(support = 0.25, confidence = 0.8, minlen = 2))

rules_df <- as(rules, "data.frame")
rules_df <- rules_df[order(rules_df[, "support"]), ]
antecedents <- as.character(rules_df$support)

item_list <- lapply(antecedents, function(x) unlist(strsplit(x, ", ")))

item_list <- as.list(unique(unlist(item_list)))

print(rules_df)
```

Number of rules	Number of items
2	8
2	8
3	8
3	8

e. (**3 points**) Identify all the applications that hold the role of the antecedent in the rules where Instagram is the consequent **.**

Answer:

```
instagram_rules <- subset(rules, rhs(rules) %in% "Instagram")

antecedents <- as(instagram_rules@lhs, "list")

unique_antecedent_apps <- unique(unlist(antecedents))
```

unique_antecedent_apps

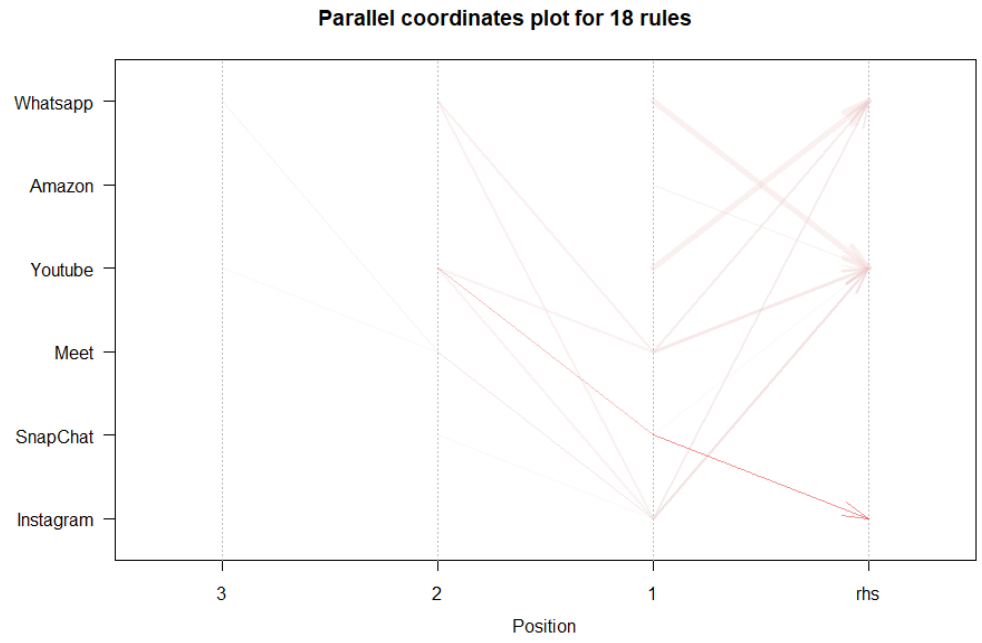
Rule	
Antecedent	**Consequent**
SnapChat	→ Instagram
Youtube	→ Instagram

f. (**3 points**) Install the ***arulesViz*** package and load the ***arulesViz*** library. Create a Parallel Coordinates plot and highlight the arrows representing the rules considered in the previous question .

Answer:

```
library(arulesViz)

plot(rules, method = "paracoord", control = list(alpha = 0.5))

highlight <- subset(rules, rhs(rules) %in% "Instagram")
plot(rules, method = "paracoord", control = list(alpha = 0.5), shading = "confidence", highlight = highlight)
```

Regular



Highlighted