Data Bases & Data Mining

Assignment - 3

Andreas Papagiannis
s2078031
and.papagiannis@gmail.com

Georgios Kyziridis
s2077981
g.kyziridis@umail.leidenuniv.nl

Mohamed Zehni Khairullah
s1902636
mzkhairullah@gmail.com

Athanasios Agrafiotis
s2029413
a.agrafiotis@umail.leidenuniv.nl

October 14, 2018

Abstract

Nowadays more and more scientists try to explore different patterns of item sets in various data. The reason is that the amount of sampling data is increasing and supervised learning is unachievable. Pattern recognition is a way to explore large-scale data in order to gain some interesting information and make some conclusions about the information that data provides. The big advantage of unsupervised learning is that we are able to gain information from data without calculating any statistical-models or mathematical equations which define the data relations but, by contrast, we can explore the data as transactions with patterns searching for the most frequent of them, and define some rules that they follows.

1 Introduction

The majority of research on census data until now has been made with various methods. In recent years, there has been increasing research interest on pattern recognition and unsupervised learning on demographic data. Pattern recognition is an unsupervised-learning method in the field of Data-Mining. It is helpful on describing relations between variables in huge datasets where supervised learning is impossible. Because of the complexity of datasets and the huge amount of observations the procedure of inspection and model training is not capable to produce efficient results.

The following sheet comprises a report on pattern recognition in census dataset case study. The dataset consists of 199523 instances and 42 variables and it is available in the following link :
https://archive.ics.uci.edu/ml/datasets/Census-Income+(KDD)

2 Exploring Data

In this section we will discuss about the data exploration in respect of how variables are distributed, what is initially interesting in the dataset and how we can gain more information about the correlation of variables. Also, we will examine some outliers or missing values.

Data includes forty-two variables mainly factors and some observations are missing so at the first step we consider about missing values. The plot below provides information about the frequency of <NA> values but also about the pattern they follow. At the left we can see the densities of NA and at the right we can see in which block of observations NA occurs. So we can define if the NA values are correlated between the variables.

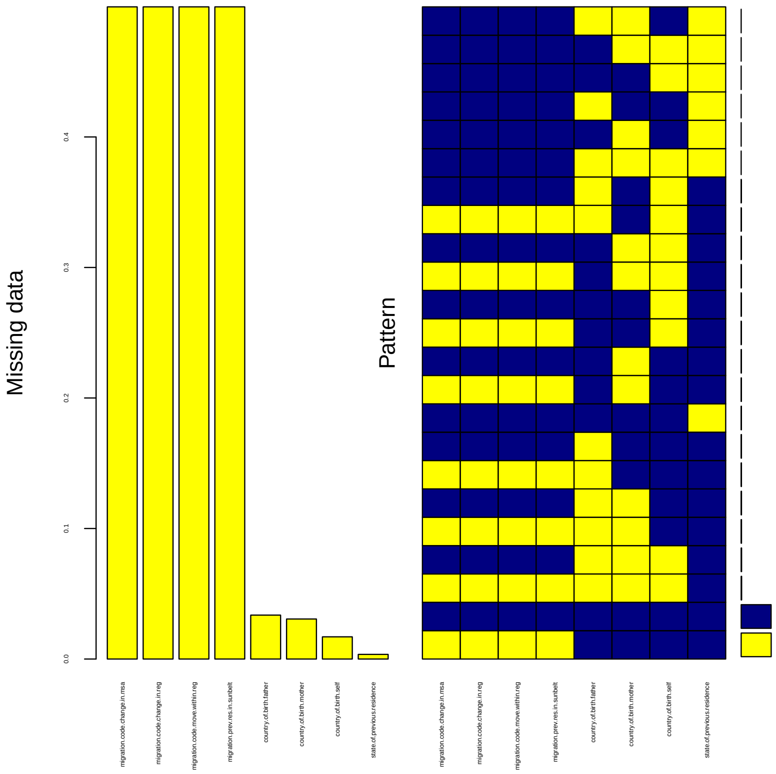


Figure 1: Frequency and Pattern of <NA> values

We can easily observe that the four migration factor variables (migration in.msa/ in.reg/ within.reg/ in.sunbelt) contain equal amount of NA values in the same observations. It is obviously the highest NA occurrence frequency comparatively with the other four. Furthermore, if we cast an eye over the pattern_plot at the right, we can assume that the other four variables which have to do with countries of birth and residence, NA values are almost randomly distributed. According to the previous information we can drive to the conclusion that maybe something wrong happened at the sampling of this dataset on the migration variables. Consider the rest variables, we assume that the proportion of NA values is reasonable, this will be explained in Multiple Imputation section.

The following plots provide information about some factor and continues variables respectively.

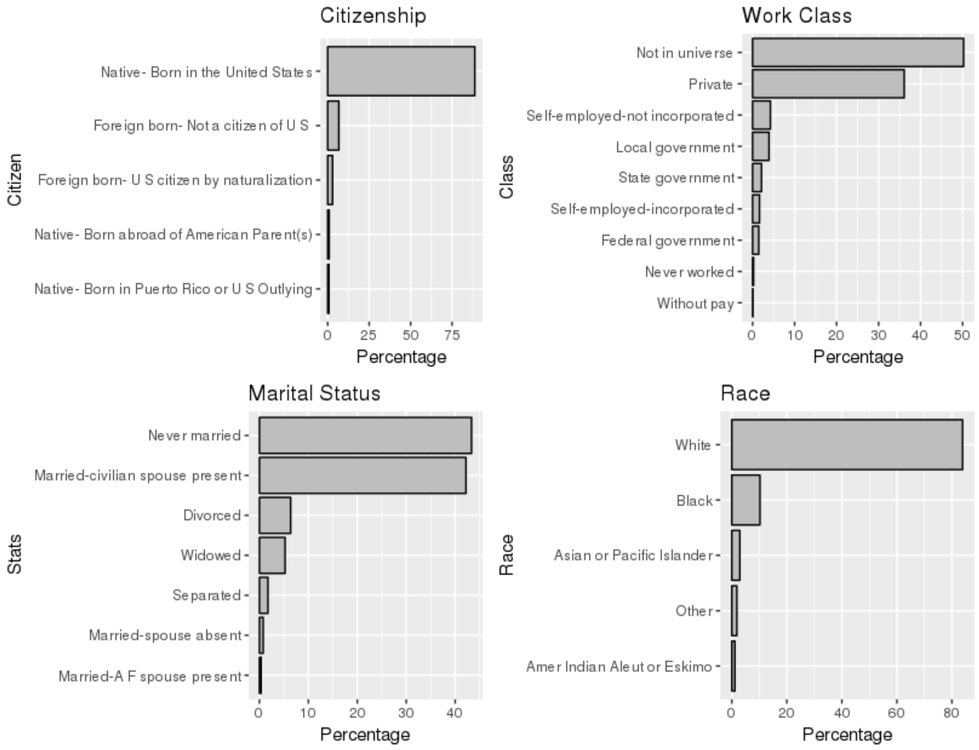


Figure 2: Barplots for some factor variables

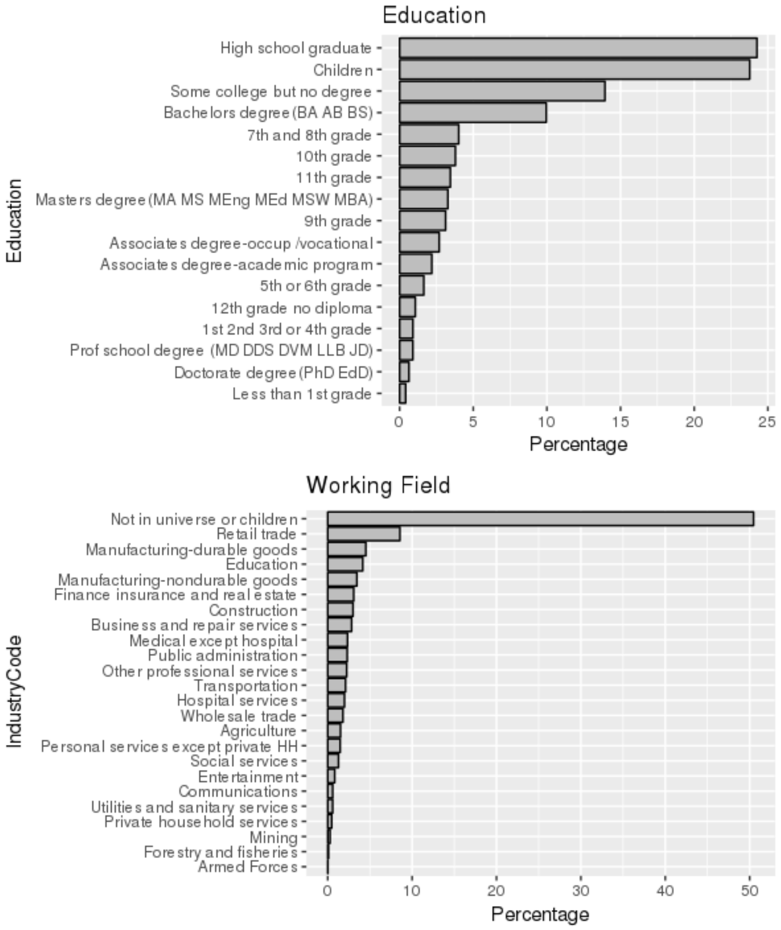


Figure 3: Barplot for Education and Working field

It can be easily observed that some of the variables include many different levels e.g. 'Education' which can be easily reconstructed for better results. We tried to group levels where it was efficient and when there were levels which overlapping others.

The following histograms visualize the how the continues variables are distributed.

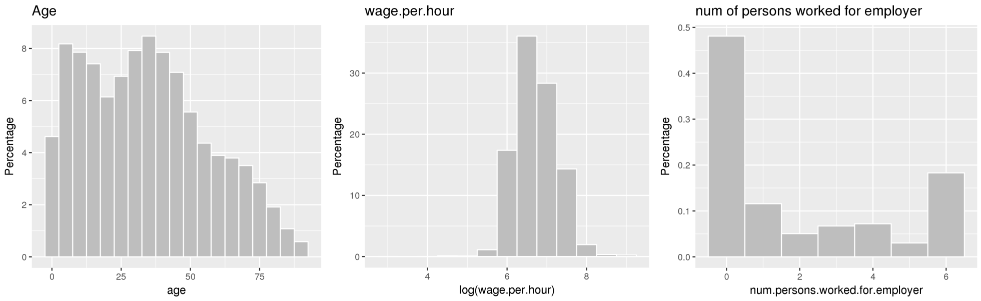


Figure 4: Histograms for some of continuous variables

3 Multiple Imputation

After a lot of examination on the specific dataset and according to the previous information we came to a decision to impute the missing values. We decided to implement multiple imputation in order to fulfill the NA values with the predicted ones. The fact that influenced us to move on this procedure is that the NA values followed specific pattern in variables with the biggest (and equal) amount of NAs. We assumed that the lower frequency of NA occurrence on the rest variables is reasonable due to the big amount of the observations, also the NA_pattern of those variables was not specific.

The idea of NA imputation is to build and train specific models to predict each observation with NA value using as prior information the whole dataset or the pattern. For this procedure, in order to test different methods and models for imputation, we construct a training model consist of all observations without NA values from the initial dataset. Then, we tried to simulate the NA different patterns for each variable and in the end we produced a training-simulation dataset which was almost the same as the initial, just with less instances.

On this dataset we implemented various methods and models of imputation using specific R.libraries. Then we tested the imputed results using Normalized Mean Squared Error(NRMSE) which represents error derived from imputing continuous values and Proportion of Falsely Classified(PFC) which represents error derived from imputing categorical values. The table below provides information about the tries with better imputation accuracy.

Table 1: My caption

-	NRMSE	PFC
Predictive Mean Matching	NaN	5.7
Random Forest	NaN	0.05
Multinomial Logistic Regression	NaN	0.13
Linear discriminant analysis	NaN	0.78

The imputation model selection had been done by researching efficient statistical models based on factor analysis for factor variables such us logistic regression or decision trees because only categorical variables contained NA values. We also used some general methods like predictive mean matching. The best result can be observed in Random Forest counted by 0.05 error. The interesting fact is that Random Forest gave the optimal error instead of logistic regression which commonly generates robust and optimal predictions in categorical data cases.

Information about the R.packages for imputation and their methods can be found in the following links :
https://www.rdocumentation.org/packages/mice/versions/2.46.0/topics/mice
https://cran.r-project.org/web/packages/missForest/missForest.pdf.

The imputed dataset can be found as a .csv file in the following link:
https://drive.google.com/file/d/1-88BS85JRYrPDvaRz9yBmFIG6ybcU7Wg/view

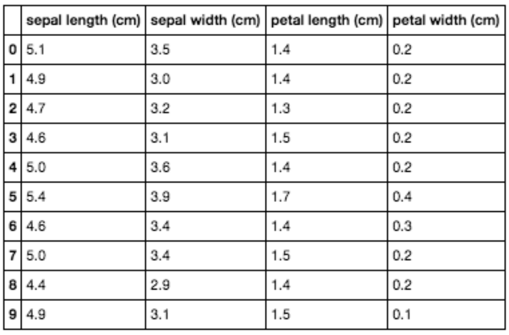
4 Apriori Algorithm

In this section we will discuss the apriori algorithm implementation in our dataset using R in order to find any interesting association rules between the variables.

Data preparation

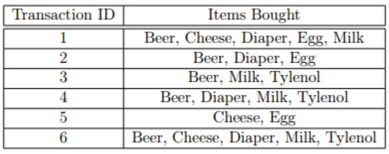
The first thing to do is to transform all of our variables into factor type of variables. For example variable age which was numeric became a factor variable with the following levels: "Minor", "Young", "Middle-aged", "Senior", "Old". Also, at this point we have to mention that we omitted some of the datasets variables because they had no way to be interpreted (like for example "instance.weight"), or because they were subcategories of other variables of our dataset (for example "state.of.previous.residence" is a subcategory of "region.of.previous.residence") and thus would be of no help for our purpose.

In order to mine any rules from the dataset is to transform it from a usual dataset where every variable is represented by a different row to what we will call a transactions dataset. Transaction datasets instead of having a column for each variable contain itemset. An itemset consists of all the variables observed in each observation.



	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1

Figure 5: Traditional dataset



Transaction ID	Items Bought
1	Beer, Cheese, Diaper, Egg, Milk
2	Beer, Diaper, Egg
3	Beer, Milk, Tylenol
4	Beer, Diaper, Milk, Tylenol
5	Cheese, Egg
6	Beer, Cheese, Diaper, Milk, Tylenol

Figure 6: Transaction table

Figures 5 and 6 show the difference between a traditional and a transactions dataset.

After constructing itemsets out of our database, we can have a general overview of our items plotting their frequencies so that we have a general idea of the most frequently occurring items of our dataset in Figure 7.

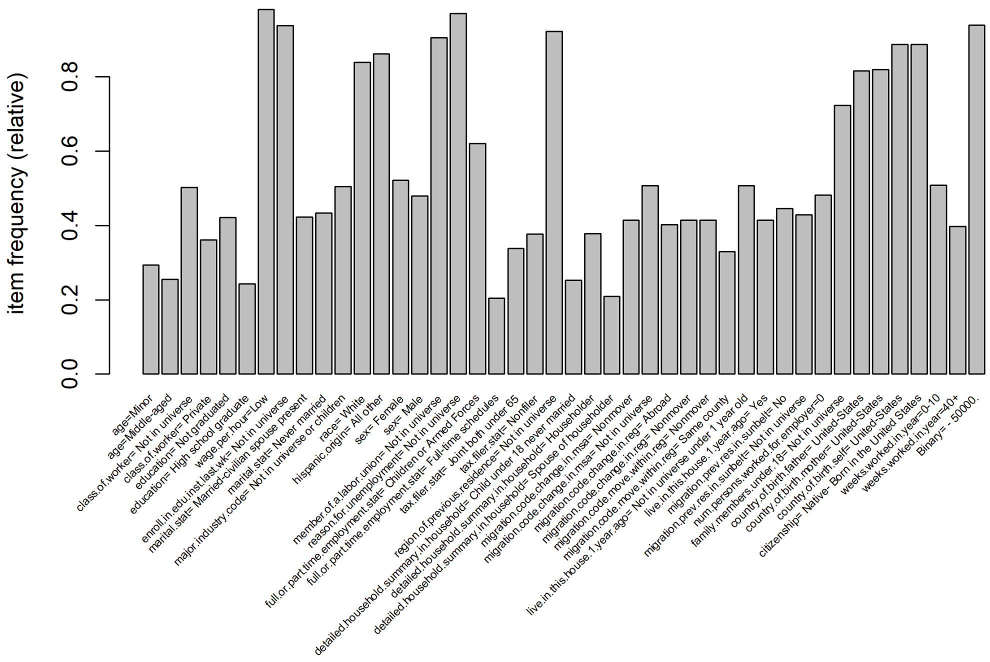


Figure 7: Item frequencies of items with support that is greater than 20%

Apriori algorithm implementation

After having a general overview of our data so far, it is time to implement the Apriori algorithm to them. In order to export rules that will be interpretable we have the ability to tune parameters such minimum support and minimum confidence for each rule and of course we can choose the maximum and minimum length of the rules we are about to export. In our case we chose support to be equal to 0.001, confidence equal to 0.5 for reasons we will discuss later and the length of our rules to be between two and four. Having less than two rules would only show us the frequencies of each item which is not what we are looking for and having more than four rules would make them considerably more difficult to interpret.

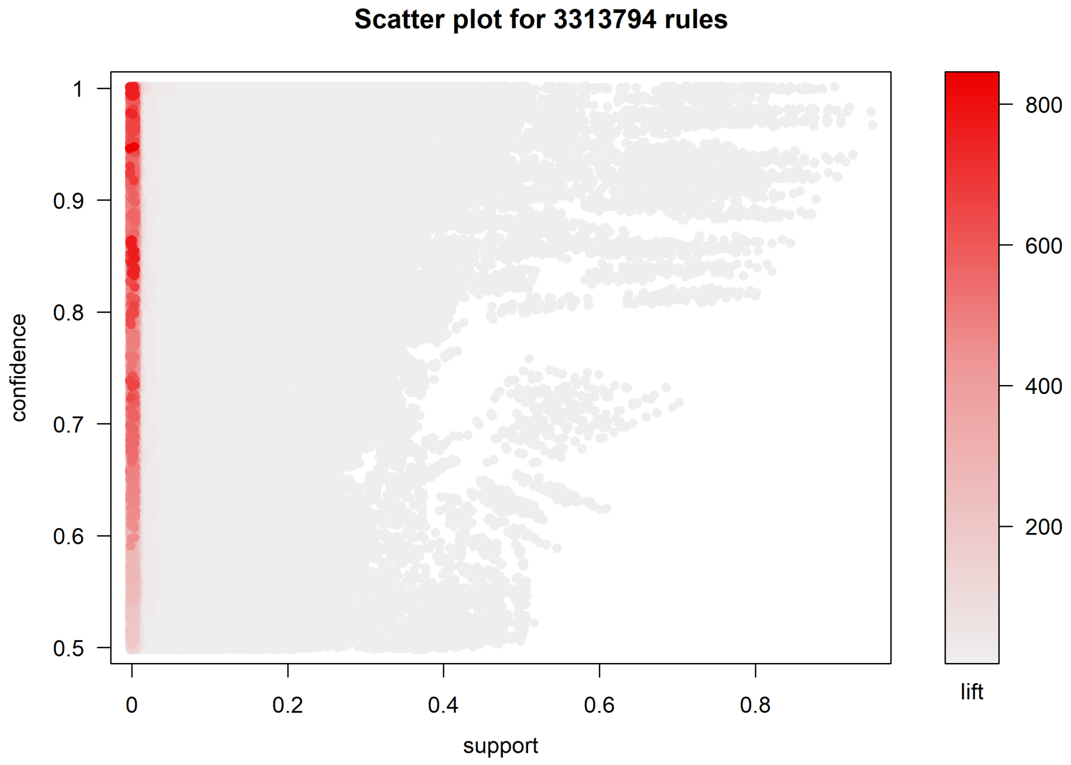


Figure 8: Scatterplot of all the rules constructed by the Apriori algorithm

In Figure 8 we visualize the support, confidence and lift values of each rule. As we can see the highest lift scoring rules are the ones that have a small support value and a big confidence value. That is the reason we chose a minimum support value of 0.001 and a minimum confidence value of 0.5. If our minimum support value was bigger we would lose a considerable amount of high scoring lift values, which is not the case if we chose a bigger value of minimum confidence.

Another striking thing we can observe, is that the number of rules constructed is enormous (approximately 3.3 million). As expected finding interesting in any way rules will be extremely difficult in such a big amount of rules unless we wish to look specifically in number of certain variables which is not our case.

Also, something not depicted in the Figure 8 is that the big majority of them has a length equal to four. Out of all the rules created, only approximately 5000 rules and 200.000 are of length two are of length three respectively. So it is safe to say that the Apriori algorithm seems to have a positive relation between the number of the rules created and their length at least in our case. In other words, Apriori is prune to create rules that are complicated in a sense that their length is not small.
All the rules generated by the apriori algorithm can be found as a .csv file in the following link:
https://drive.google.com/file/d/1uOG7W1KacJ84LixGwzBNwVFNT_YyvhCB/view

Finding interesting rules

The first approach we have tried to find interesting rules is to sort the whole entity of the rules we discovered by their lift value hoping that the highest scoring lift rules will be of some interest. Unfortunately, we discover that the higher a rule's lift value is, the more prone is the rule to be redundant. For example, the highest lift scoring rule out of all is stating that: if there are no underage family members and the person's mother was born in Iran, then that person is more likely to have been born in Iran too. This should not be something that takes us by surprise. Another remark about lift value and our rules is that the great majority of high lift scoring rules explains the country origin of each person based on he's/she's ancestors country of origin. After inspecting the first 30 highest scoring lift rules it would not be a generalization to say that they are of almost no interest as all of them are conclusion we could have come by our own without the help of any computational method such as data mining.

So as the sorting all the rules by their lift value does not give us much information what we tried next in order to find any interesting rules was to look at subsets of rules based on the by right-hand side of each rule's equation. In other words to find the reasons more likely to cause each observation's value for every variable of our dataset. Again in order to find the most important out of all we sort them by their lift value. The difference is that instead of inspecting the highest lift scoring rules out of all the rules generated by the apriori algorithm we choose only a finite number of high lift scoring rules (in our case five rules) that are associated with a certain variable at the right hand of the rule equation for each variable.
These rules can be found as a .csv file in the following link:
https://drive.google.com/file/d/1zpFMop8G1LVYwuyaymh9fFL4dhMkS93c/view?usp=sharing

5 Conclusions

Inspecting our rules in that way still includes some rules that are redundant but we also observe some rules that came of our (personal and thus objective) interest.

Some of them are namely:

- Construction workers are prone to be male (rules #1735817, #1737035 , #431288).
- Mid-payed male workers with a number of persons worked for employer equal to six are prone to be members of a labor union (rule #1096867).
- Black race individuals are prone to have only their mother present in their household (rules #2342364, #2482199, #1552856, #2594225).
- Individuals who work in communications industry code are prone to work more that 40 weeks per year (rules #1498284, #1498285, #1498286).
- Individuals with a professional school degree who work in medical except hospital industry code are prone to earn more that 50.000$ per year (rules #728867, #728864, #728873, #728870).

Note that the lift values vary from rule to rule making some stronger that others. The numbers (#) of each rule are included as a reference in order to easily find each rule on the .csv tables that contain the rules (see above) along with their support, confidence, lift and count values.

Finding any interesting association rules between variables of big datasets can be very tricky. The lift value of each rule is a way to determine if the rule is important or not but is not of much help in finding interesting rules in bag datasets.
In order to find interesting rules, one can look at subsets of the rules generated in many different ways. Specifying rule subsets by variables the way it is done here is only one way to subset all the rules generated by the apriori algorithm.

The code used for this project followed with thorough comments describing each part of it and instructions about how to run it can be found and downloaded as a .zip file in the following link:
https://drive.google.com/file/d/1JhkegGFNZGgOjrsI9kK6_LOZ7bA41wyV/view?usp=sharing

6 References

- Rproject:
  https://www.r-project.org/
- Multiple Imputation for Continuous and Categorical Data :
  http://www.stat.columbia.edu/~gelman/research/published/MI_manuscript_RR.pdf
- Introduction to arules - A computational environment for mining association rules and frequent item sets:
  https://cran.r-project.org/web/packages/arules/vignettes/arules.pdf