Universitat Politècnica de Catalunya

Data Mining (MD)

Bachelor Degree in Computer Science

Practical Work 1: Speed Dating Dataset

Authors:
Jordi Armengol Estapé, Marc Catrisse Pérez,
Albert Figuera Pérez, Roland Friess,
Enrique González Sequeira, Jacobo Moral Buendía

Professors:
Karina Gibert
Beatriz Sevilla

October 26th
Fall semester, 2018-2019

Contents

1 Motivation and general description of the problem

The motivation of this project is to apply the concepts we have been studying in the Data Mining subject in order to extract knowledge from a dataset based on a survey taken to participants of a series of speed dating. Thus, we will both learn data mining concepts by applying them in real data and hopefully discover what makes a date successful.

In particular, we will apply the following concepts which we have studied in the course (among others):

- Pre-processing.
- Basic statistical and descriptive analysis.
- Factorial analysis.
- Hierarchical clustering.
- Profiling.
- The R programming language.

As far as the dataset is concerned, our intention is to analyze what are the most important features on a date for being successful, taking into account the information available in the survey. For instance, we would like to discover whether the age or the difference of ages have a big impact or otherwise it is more important to have a high income or having a certain career and income.

We think that this dataset can be very interesting to analyze because there are different points that we can study. The main question that we want to answer is: What attributes influence the selection of a romantic partner?

Nevertheless, we have to say that in order to be able to finish this project in time without compromising its quality, we have had to narrow the scope of our data mining process. That is to say, not all the questions in the survey have been taken into account (as detailed in the pre-processing chapter). For instance, there were questions which were actually not related on the dates itself, but on the perceptions that individuals had of themselves. This information could lead to another whole data mining project with the aim of comparing the perceptions which individuals have of themselves with the way other people actually perceive them, but it is clearly out of the scope of this project.

This dataset has proven to be not very *data-science-friendly*, although we were aware of this concern from the beginning of the project. We chose it not because it was easy to work with, but because it was interesting and we were keen on the challenge. We hope that our effort will have been worth it.

2 Data source presentation

Our dataset, which consists of only one CSV file, has been downloaded from *Kaggle*[1], a repository of open datasets and a community of data scientists and machine-learning contests. The process to get the data is, therefore, pretty straightforward. However, a (free) *Kaggle* account is needed in order to be able to start the download. The data was compiled by Columbia Business School professors Ray Fisman and Sheena Iyengar. The original paper can be found at Columbia's website[2]. Data was gathered from participants in experimental speed dating events from 2002-2004. During the events, the attendees would have a four minute "first date" with every other participant of the opposite sex. They were asked to fill questionnaires about different aspects, including demographics, dating habits, lifestyle information, the rating of their partners (six attributes: Attractiveness, Sincerity, Intelligence, Fun, Ambition, and Shared Interests), self-perception across key attributes, whether they had liked their partner, their expectations regarding the dates and other beliefs.

3 Formal description of data structure and metadata

As stated before, our dataset is not simple. The original dataset contained as many as 8378 rows and 190 columns. Also, the meaning of each one of them is not easy to interpret. Fortunately, the researchers who collected the data released a document detailing the meaning of each variable and the way the study was conducted [3]. We encourage the reader to take a look at this document. Nevertheless, we have found some variables and procedures to still not be documented in a way we could properly understand. In the original dataset, about 26% of the data was missing; there were 7 binary variables, 129 numeric variables, 51 categorical variables and 3 id columns. Later on we will detail the final selection.

3.1 Rows

The dataset we chose correspond to information gathered from participants using the following questionnaires:

- The survey filled out by people which is interested in participating in order to register for.
- The scoreboard about their partner that each participant had to fill after each meeting.
- The survey filled out the day after participating in the event.
- The survey filled out 3/4 days after participating in the event.

Each row of our dataset corresponds to a date, which contains filled information by the participant and his or her partner (only heterosexual dates were considered in the original study). However,

[1] Speed Dating Experiment: `https://www.kaggle.com/annavictoria/speed-dating-experiment`

[2] Gender Differences in Mate Selection: Evidence From a Speed Dating Experiment: `https://academiccommons.columbia.edu/doi/10.7916/D8FB585Z`

[3] Speed Dating Data Key: `https://perso.telecom-paristech.fr/eagan/class/igr204/data/SpeedDatingKey.pdf`

as each row tells the date from the point of view of one participant, a single wave data has 2 entries for each date.

3.2 Metadata table

The following table corresponds to the metadatable table of the selected variables (not the original dataset):

Variable	*Modalities*	*Meaning*	*Type*	*Measuring unit*	*Missing code*	*Measuring procedure*	*Range*	*Role*
gender		gender of the subject. Female=0, Male=1	Boolean		*empty cell			Explanatory
order		the number of date that night when met partner	Number		*empty cell		[1,22]	Explanatory
match		there's been a match between the two subjects	Boolean		*empty cell	True if dec = dec_o = true. False otherwise		Response
int_corr		correlation between participant's and partner's ratings of interests	Number		*empty cell		[-1,1]	Explanatory
samerace		participant and the partner were the same race	Boolean		*empty cell			Explanatory
age_o		age of partner	Number	Years	*empty cell		[0-100]	Explanatory
race_o	1: Black; 2: Caucasian; 3: Latino; 4: Asian; 5: Native Americ; 6: Other	race of partner	Qualit.		*empty cell		[1,6]	Explanatory

pf_o_att		partner's stated preference for attractive attribute	Number		*empty cell		[0,100]	Explanatory
pf_o_sin		partner's stated preference for sincere attribute	Number		*empty cell		[0,100]	Explanatory
pf_o_int		partner's stated preference for intelligent attribute	Number		*empty cell		[0,100]	Explanatory
pf_o_fun		partner's stated preference for funattribute	Number		*empty cell		[0,100]	Explanatory
pf_o_amb		partner's stated preference for ambitious attribute	Number		*empty cell		[0,100]	Explanatory
pf_o_sha		partner's stated preference for shared interests attribute	Number		*empty cell		[0,100]	Explanatory
dec_o		decision of partner the night of the event	Boolean		*empty cell			Response
attr_o		rating by partner the night of the event for attractive attribute	Number		*empty cell		[0,10]	Explanatory
sinc_o		rating by partner the night of the event for sincere attribute	Number		*empty cell		[0,10]	Explanatory

intel_o		rating by partner the night of the event for intelligent attribute	Number		*empty cell		[0,10]	Explanatory
fun_o		rating by partner the night of the event for fun attribute	Number		*empty cell		[0,10]	Explanatory
amb_o		rating by partner the night of the event for ambitious attribute	Number		*empty cell		[0,10]	Explanatory
shar_o		rating by partner the night of the event shared interests attribute	Number		*empty cell		[0,10]	Explanatory
age		age of subject	Number	Years	*empty cell		[0-100]	Explanatory

field_cd	1: Law; 2: Math; 3: SocSci Psych; 4: Med Pharma Biotech; 5:Engin; 6: Writ Journ 7: His/Rel Phil; 8:Bis/Econ /Fin; 9:EduA-cad 10: BioSci Chem Phy; 11: SocWork; 12: Und-grad/Undec 13: PolSci 14: Film; 15: Arts; 16: Lang; 17: Arch; 18: Other	field of study	Qualit.		*empty cell		[1,18]	Explanatory
mn_sat		Median SAT score for the undergraduate institution where attended	Number		*empty cell		[400,1600]	Explanatory
tuition		Tuition listed for each response to undergrad in Barron's 25th Edition college profile book	Number	Dollars	*empty cell			Explanatory

race	1: Black; 2: Caucasian; 3: Latino; 4: Asian; 5: Native Americ; 6: Other	race of subject	Qualit.		*empty cell		[1,6]	Explanatory
imprace		Importance to subject that partner is the same race	Number		*empty cell		[1,10]	Explanatory
income		Income/salary of subject	Number	Dollars/year				
goal	1: Seemed like a fun night out. 2: To met new people. 3: To meet new people. 4: Looking for a serious relationship. 5: To say I did it. 6: Other.	Primary goal of participating in the event	Qualit.		*empty cell		[1,6]	Explanatory

date	1: Several times a week; 2: Twice a week. 3: Once a week. 4: Twice a month. 5: Once a month. 6: Several times a year. 7: Almost never	How often does subject go on a date	Qualit.		*empty cell		[1,7]	Explanatory
go_out	1: Several times a week; 2: Twice a week; 3: Once a week; 4: Twice a month; 5: Once a month; 6: Several times a year. 7: Almost never	How often does subject go out	Qualit.		*empty cell		[1,7]	Explanatory
dec		Decision of subject	Boolean		*empty cell			Response
like		How much does subject like partner	Number		*empty cell		[1,10]	Explanatory
imprelig		Importance to subject that partner is the same religion background	Number		*empty cell		[1,10]	Explanatory

diff_age		Absolute difference of between partners age	Number		*empty cell		[1,10]	Explanatory
university		Whether the person attended (or is attending) university	Boolean		*empty cell			Explanatory
round		number of people that met in wave	Number		*empty cell		[6-21]	Explanatory

3.3 Final scope

The original data were collected for a large-scale study on romantic relationships and some factors like the race. As stated before, it had 190 variables, of which 7 were binary, 51 were qualitative and 129 were numeric. In order to cut some of them, we analyzed the documentation of our data in order to find variables that could be removed out without losing the sense of the data. We realized that there were many redundant columns. It have to be taken into account that the scope of our study was the factors influencing the success of dates, and actually the dataset included much more information. So, we removed many variables mainly because of two reasons:

- Because many of them were not actually related to the aim of our study. Please recall that our goal is to answer the question *What attributes influence the selection of a romantic partner?*.
- Because some of them explain an information which is contained in another, more general variable (redundancy). Still, we will keep some redundant variables that we considered useful.

We decided to remove all variables related to the follow-up questions and keep only the main questions, because we were only keen on the dates itself. We removed some of the variables related to the questionnaire filled before of the dates, since we were not interested in the self-perceptions and expectations of individuals (which would require a whole data mining project itself). Interests were removed because they were summarized in one variable, the correlation of interests between partners. Also, a subset of rows were devoted to special waves of dates, that is to say, dates conditioned to certain events and with different questions. We did not consider these rows. We cut out the rows referring to badly answered questionnaires as well. As far as the removed variable *undergra* is concerned, we performed a semantic collapse.

The complete process of rows and variables selection will be detailed in the pre-processing section.

4 Complete data mining process

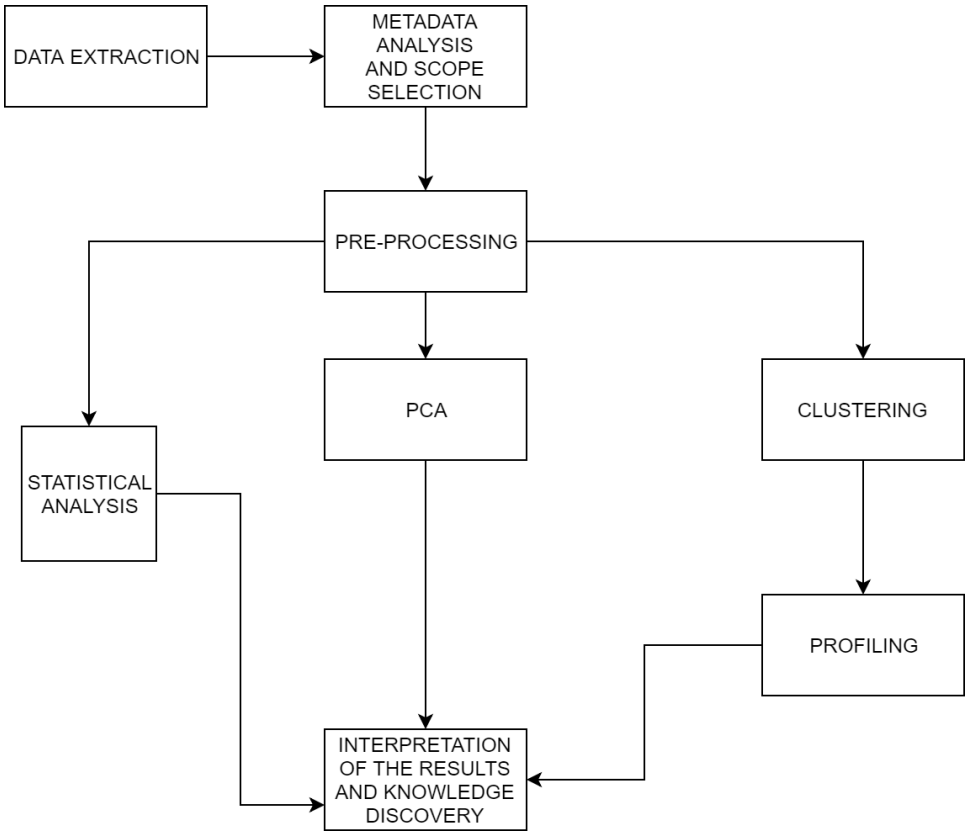

Figure 1: Data mining process chart.

- Data extraction: obtaining the dataset via Kaggle.
- Metadata analysis and scope selection: inspecting the dataset, trying to understand what columns and rows mean and finally constructing the metadata table.
- Pre-processing: selecting columns and rows, cleaning the dataset and imputing missing values.
- Basic statistical analysis: getting basic statistical indicators and performing some early visualizations of the data.
- PCA: decomposing the numerical variables in the principal components (factorial analysis) and projecting the data cloud.
- Clustering: dividing the data into different groups by applying hierarchical clustering techniques.
- Profiling: detailing the common features of the different clusters.
- Interpretation and knowledge discovery: trying to answer our initial question with the results of the different techniques. Performing knowledge extraction based on the previous steps.

5 Preprocessing and data preparation

We tried to follow the steps and methods which we have been learning in class. Particularly, we based our work on the reference survey[4].

5.1 Building the original data matrix

We started by building the original data matrix and introducing the data into the pre-processing tool (R). The dataset is stored in a single CSV file (the default separator will work for us), NAs can be both 'NA' or empty and the header is included.

At this point, we checked the dataset by reading the first rows, reading the summary of the data, counting NAs... Recall that we had 190 variables: 7 binary, 51 categorical, 129 numeric and 3 id columns. These types had to be checked manually by inspecting the dataset, because for instance many qualitative variable were codified and detected as numeric. We had about 26% missing values. We did some early visualization, too (still, in order to check the sanity of the data and get familiar with our dataset).

5.2 Determining the working data matrix

5.2.1 Rows

In order to have coherent and clean data, we discarded the rows belonging to special waves of dates (the ones conditioned to particular events) and the ones with different preference scales (for the preference questions, some waves had a 1-10 scale instead of 0-100 points to distribute). Actually, it can be considered that they had different questions, so it is not feasible to adapt them. We will not take into account these special waves of dates.

5.2.2 Columns

We had many variables. Some of them were introducing noise, some of them were redundant or way too concrete, some of them are out of the scope of our intended analysis. We performed to do a kind of "expert" variable selection. As stated before, we are only interested in the dates itself, not in the self-perceptions, expectations or follow-ups. Also, the variable *int_corr* tells us the correlation of interests between the two partners of a date, so we discarded the other variables containing the information with the exact interests.

So, we are essentially keeping the same variables present in the metadata table.

At this point we still kept the id's because we needed them for some of the following steps (for missing imputation purposes). We kept *undergra*, too. Although it will not be present in the final dataset, we needed it in some intermediate steps. This variable contains the names of the universities attended by each one of the participants (provided they attended university; otherwise the cell is NA).

[4] A survey on pre-processing techniques: Relevant issues in the context of environmental data mining: `https://www-eio.upc.edu/~karina/datamining/refmaterial/metainfoPrepro/AIC710def.pdf`

5.3 Declaring qualitative variables and relabeling

R detected as numeric some variables which were actually qualitative, because they were codified as numbers, so we had to manually declare them as categorical. Also, for data visualization, we relabeled the categories. For instance, instead of *gender* being 0 or 1, the final dataset had F or M (as in female or male). The exact procedures and affected variables can be consulted in the R code.

5.4 Declaring numeric variables

Strangely enough, R detected as qualitative variables some columns which were actually numeric: *mn_sat*, *tuition* and *income*, so we had to declare them as numeric.

5.5 Outlier detection and visualization

We did a loop which did plots of all variables to manually inspect them in order to find potential candidates for being outliers. We make R print the percentage of missing values for each variable, too.

This part was very hard because we had to go variable by variable trying to figure out what to do in each case. The whole process is detected in the R codes (appendix).

For instance, some ages did not fit in the boxplot, but they were real ages.

5.6 Error detection and treatment

In *imprace*, 0 was not a valid value in the scale, so we replaced all occurences of 0 by NA.

We deleted the whole *met* because their values did not follow at all the specifications of the document of the original study (it was supposed to be binary but actual values were between 0 and 7), so we had no way of treating or interpreting it.

5.7 Missing imputation

We had two structural missings: *tuition* and *mn sat*, because they only made sense for people who had attended university. So we used *undergra* in order to revalue as 0 all NA's of people who hadn't attended university. We created a new binary variable using both *undergra* and *tuition*: *university*, which tracked whether participants had attended university. This had to be done at this point.

We created new labels for missings for qualitative variable: *unknown*, which would replace NAs.

We run the Little Test but in our case it was not very informative.

For the remaining missing numeric values, we considered both KNN and MIMMI, but we chose KNN. The problem was that all numeric variables had at least one missing, so we could not run KNN. By manually inspecting the missings, we observed that 3 particular individuals were

responsible for many missings. By removing it and manually curating a very few rows (using expert knowledge and common-sense), we could start applying the KNN iteratively (starting with the variables that had the less missings).

5.8 New variables

As stated before, we will have the variable *university*. Also, we created a new numeric variable, *diff age*, the absolute difference of ages.

5.9 Re-scaling

The KNN method is not aware that the preferences must add up to 100 for each row, so we had to re-scale these columns. There was only one row for each all preferences added up to 0, so we removed this row.

6 Statistical descriptive analysis

For a better understanding of the data we performed a basic statistical analysis with some variables we found relevant. Some variables have been discarded because they do not provide useful information.

6.1 Numerical variables

The following table presents a summary of basic statistical indicators for the numerical variables:

Variable name	Min	1st Qu	Median	Mean	3rd Qu	Max	Sd
int_corr	-0.7000	-0.0200	0.2100	0.1948	0.4300	0.9000	0.3076232
age	21.00	24.00	26.00	26.09	28.00	39.00	3.353241
age_o	21.00	24.00	26.00	26.1	28.00	39.00	3.358056
diff_age	0.00	1.00	3.00	3.592	5.00	17.00	2.866799
order	0.000	4.000	8.000	8.875	13.000	21.000	5.399698
round	6.00	16.00	18.00	16.84	19.00	21.00	4.10614
pf_o_amb	0.000	5.000	10.000	9.486	15.000	36.000	6.135659
pf_o_att	0.00	15.00	20.00	23.72	30.00	100.00	13.73557
pf_o_sin	0.00	11.75	20.00	17.38	20.00	40.00	7.633928
pf_o_int	0.00	18.00	20.00	21.21	25.00	50.00	8.014568
pf_o_fun	0.00	12.00	18.00	17.18	20.00	45.00	6.783696
pf_o_sha	0.00	5.00	10.00	11.02	15.00	30.00	6.667148
attr_o	0.00	14.00	16.00	15.58	18.00	40.00	3.928395
amb_o	0.00	15.00	17.00	17.57	19.00	50.00	3.920065
sinc_o	0.0	17.0	18.0	18.3	20.0	39.0	3.594274
func_o	0.00	15.00	17.00	16.05	18.00	53.00	3.440163
intel_o	0.00	17.00	18.00	19.01	20.00	50.00	3.337487
shar_o	0.00	11.00	14.00	13.72	17.00	40.00	4.280074
mn_sat	0.0	0.0	1100.0	727.3	1365.0	1490.0	661.1259
like	1.000	5.000	6.000	6.135	7.000	10.000	1.829171
income	8607	31432	43664	45892	55080	109031	17809.83
imprelig	1.000	1.000	3.000	3.805	6.000	10.000	2.8707
imprace	1.000	1.000	3.000	3.589	6.000	10.000	2.807687
tuition	0	0	12900	12384	26019	34300	12093.88

Table 2: Qualitative variables raw information

Now we will analyze the qualitative variables which have useful information in order to achieve our goal of extracting knowledge from the data:

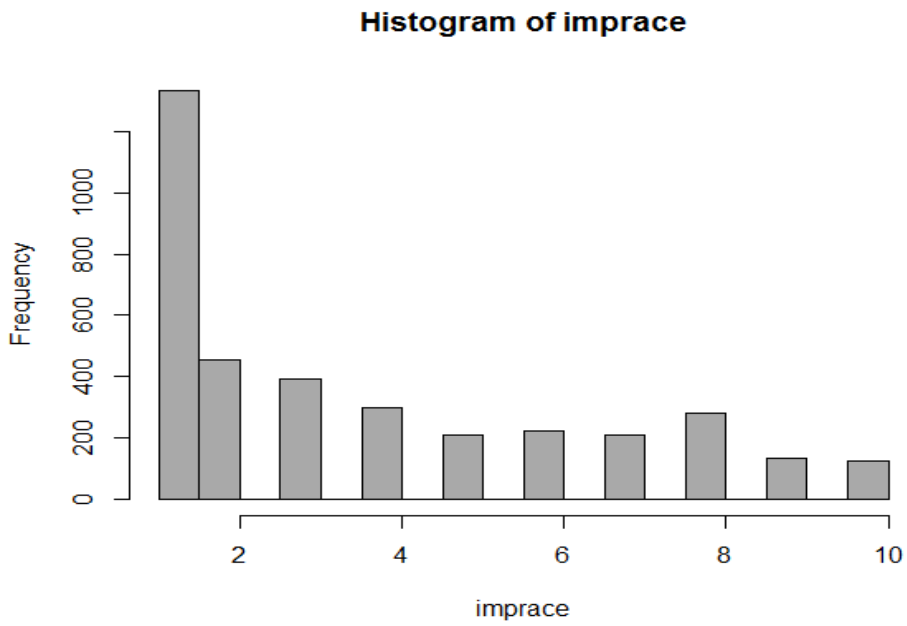

Figure 2: Frequency of the importance of the race on dates.

IMPORTANCE OF RACE

In Figure 1 we can see the frequency of the mark that people give to the importance of the race of the partner on a date. The mark is an integer between 1 and 10. And as we can see, most of the people give very little importance to race. The most voted mark by far is a 1.

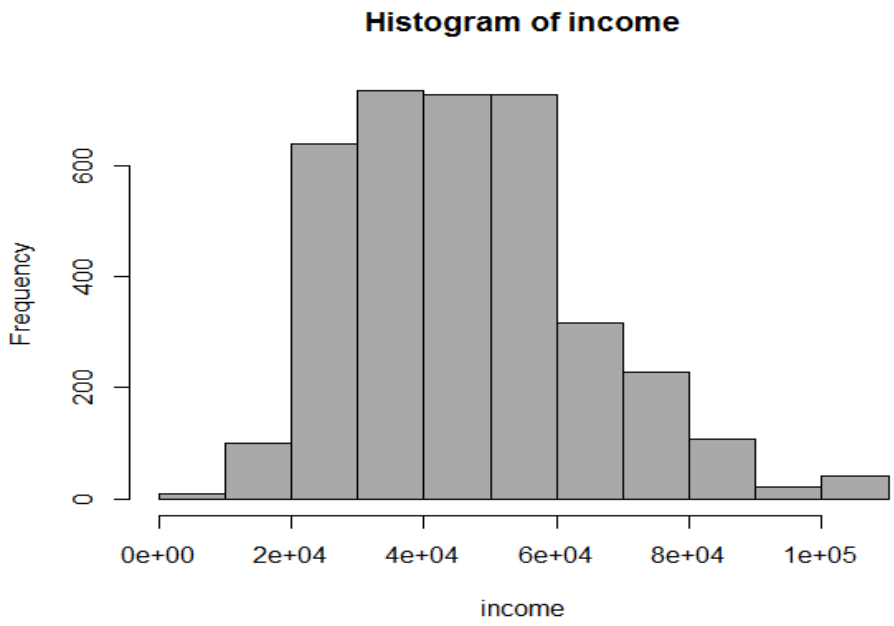

Figure 3: Frequency of the income variable.

`INCOME`

In Figure 2 we can see the frequency of people income. Most people earn between 40.000$ and 60.000$ a year but there is a big difference between the higher and the lower income, the standard deviation is pretty high (17.000$). We can relate this to people who have studies and people who don't.

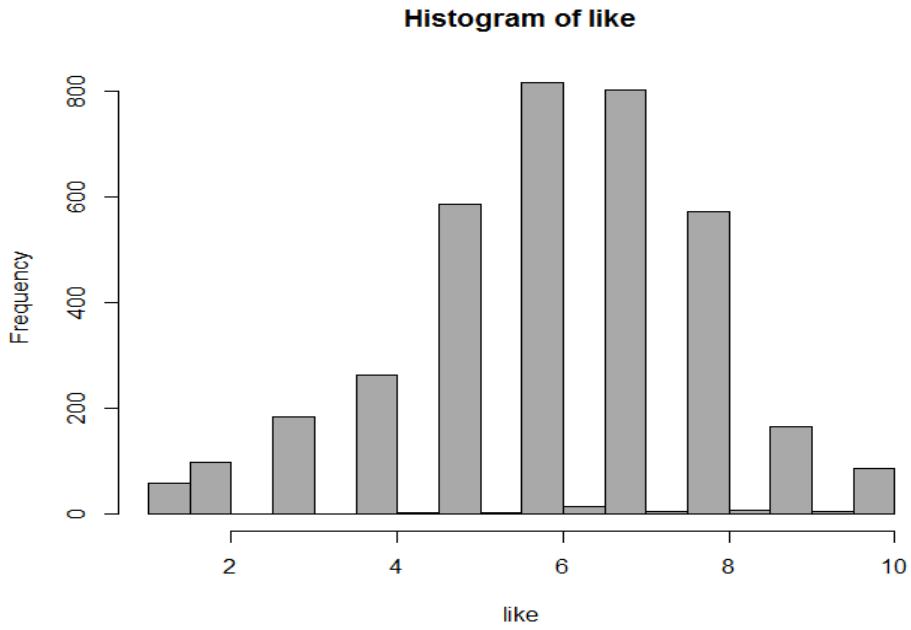

Figure 4: Frequency of the like variable.

LIKE

In Figure 3 we can see the frequency of people voting how much they liked his partner. People tend to vote positively.

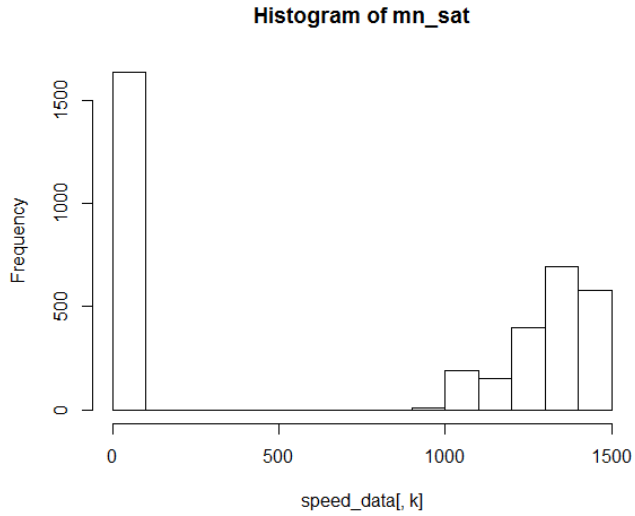

Figure 5: Histogram of mn_sat.

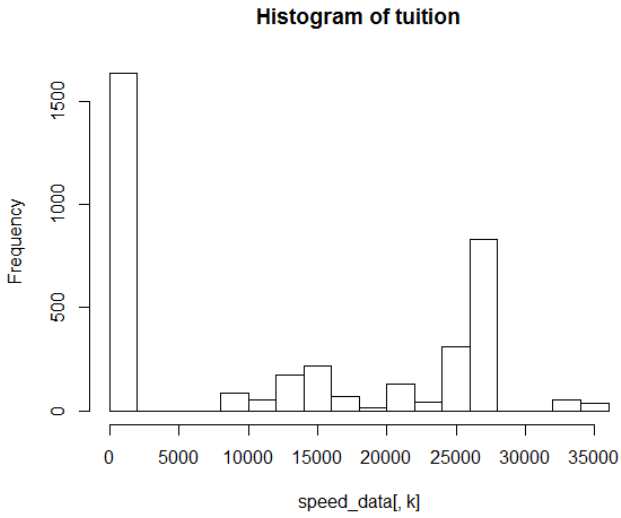

Figure 6: Histogram of tuition.

As we can see on Figures 5 and 6, this variables have been affected by preprocessing due to most of values are zeros. This is because we transformed all missing values to zeros. We assumed that a missing value meant that the person hasn't gone to university so his mn_sat and tuition value are zero. Considering that we can appreciate that mm_sat average is between 1000 and 1500 and that tuition have a peek on 27.500.

6.2 Categorical variables

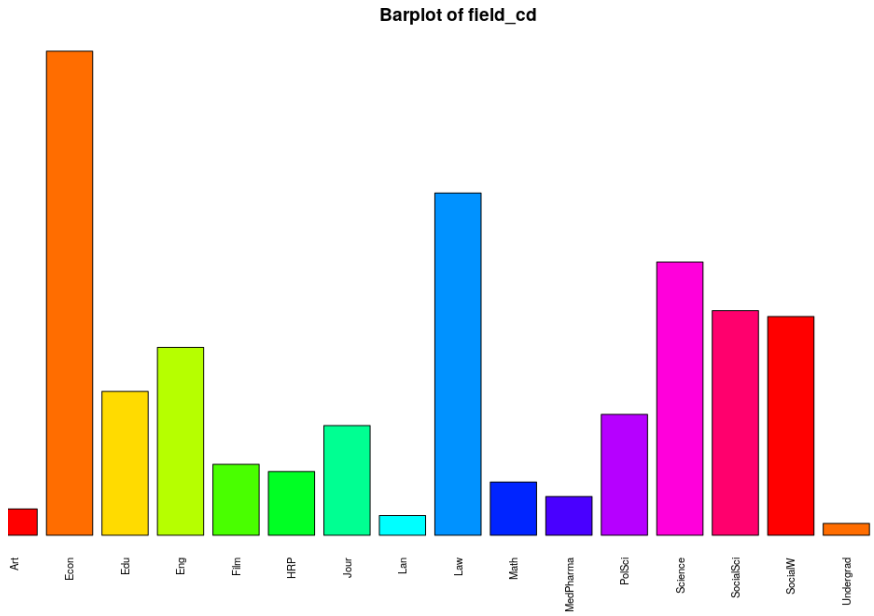

Figure 7: Barplot of people jobs.

FIELD_CD
In Figure 7 we can appreciate the frequency of the occupation of the people who have been dating. With this information we can relate if the occupation of the partner has a big impact on the decision of matching.

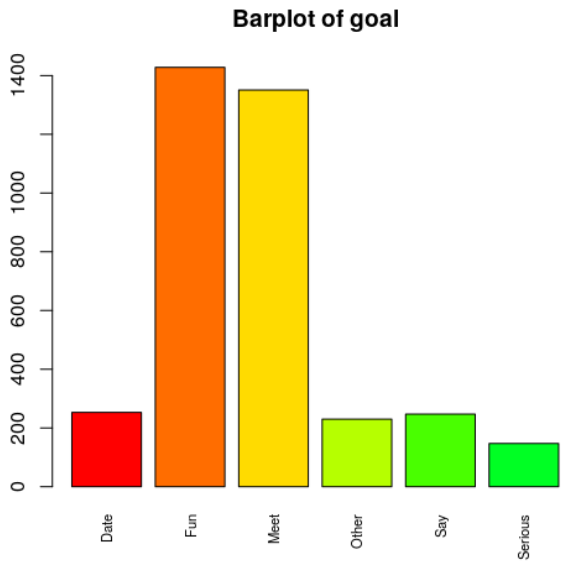

Figure 8: Frequency of the goal of a date.

GOAL In Figure 8 we can see the frequency what are the people looking on a date.

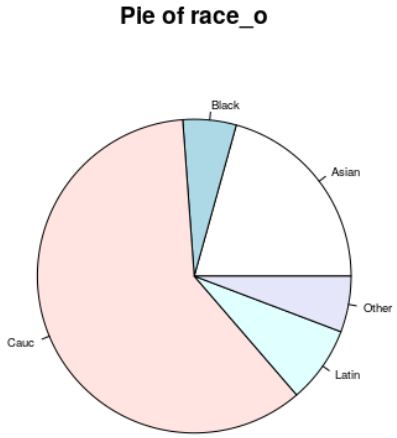

Figure 9: Race information.

`RACE` In figure 9 we can appreciate the frequency of different people races. The majority of the people on this programme are Caucassian. We can relate this variable to the chances of getting a match of every different race.

6.3 Bivariate analysis

In order to answer the main question of this project, we want to relate the variable *match* with the variables that have more information on giving us an answer to out main question.

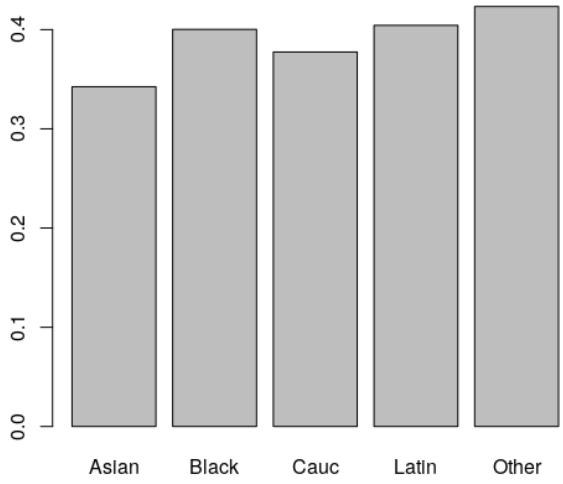

Figure 10: Relation between match and race.

As we can see on Figure 10 the race of the people on a date does not seem to have a big impact on having a match, Asians have a slightly difference of matching but overall it is equal.

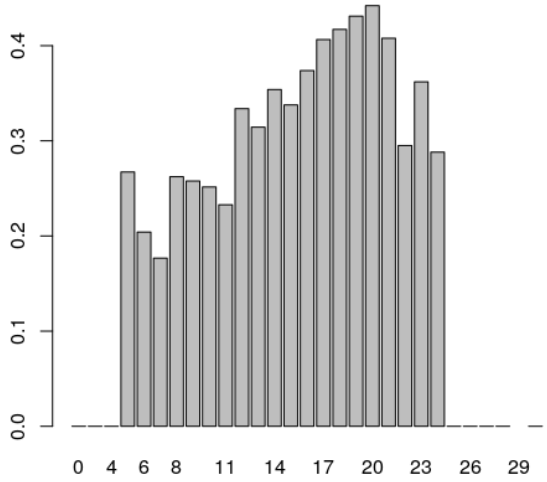

Figure 11: Relation between match and fun_o.

In Figure 11 we can appreciate that people who are more fun on the point of view of their partner have more chances of getting a match that people who have less points on being fun.

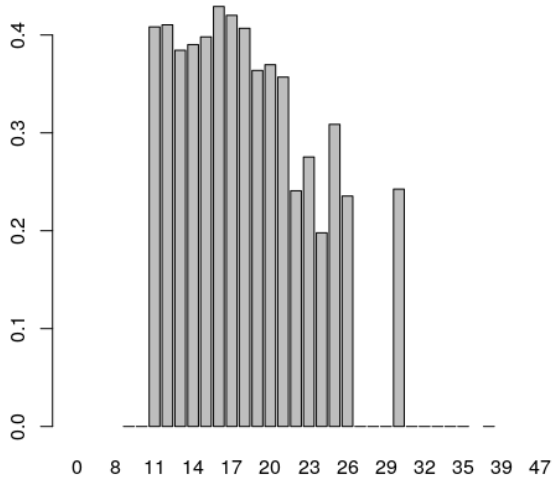

Figure 12: Relation between match and intel_o.

In Figure 12 we can see that if someone sees you intelligent then a match is less probable. People with low points on intelligent tend to have more matches.

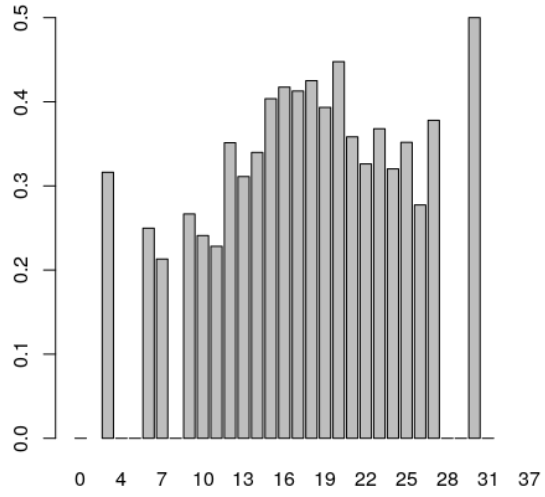

Figure 13: Relation between match and attr_o.

In figure 13 we can observe that the outliers are very clear, people with very few points in attractiveness have very small chances of getting a match and on the other hand people with a lot of points on attractiveness have a lot of more chances of getting a match.

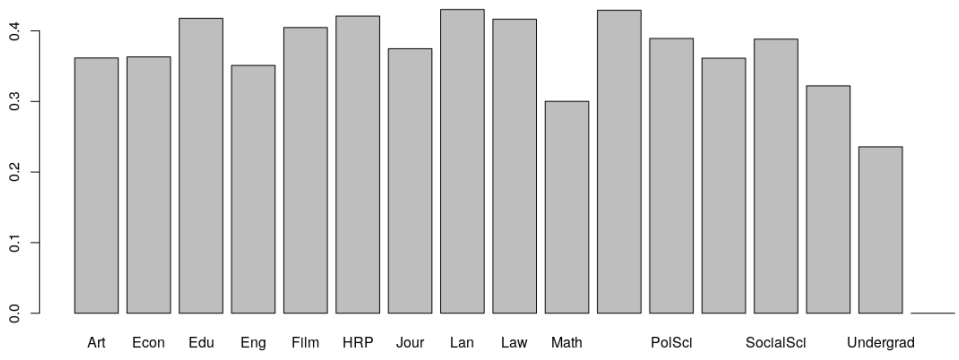

Figure 14: Relation between match and field_cd.

6.4 How is our data?

As we said before our data rows represent a date on the point of view of one individual. We want to answer the question "*What features have impact on having a match on SpeedDating*", looking to this first statistical analysis we can conclude that: race is not important on having a match, being fun and attractive to your partner gives you a lot of chances to get the match.

But if you seem intelligent you have less opportunities to get the match. Also we concluded that people who work on the field of maths have lower chances of getting a match. Men are more likely to give a positive decision while women do the opposite.

7 PCA analysis

Principal Component Analysis (PCA) is a kind of factorial analysis for numerical variables used for finding patterns in high-dimensional space. It consists of different orthogonal transformations. The main idea is to find the principal components which explain the most variance, and then project the data cloud. In our study we did a systematic analysis for each pair of principal components (14 components explained about 80% of the variance), but for briefness we will explain the axis with PCs (x: 1, y:2) and (x: 2, y:3) which are some of the ones which explain the more variance. We started by doing only the study for (x: 1, y:2) but in order to improve our project we added one more pair of principal components.

7.1 Accumulated percentage of inertia

We selected 14 principal components since they are responsible for the 79.92% of the total inertia, as can be seen in the cumulative histogram plot 15.

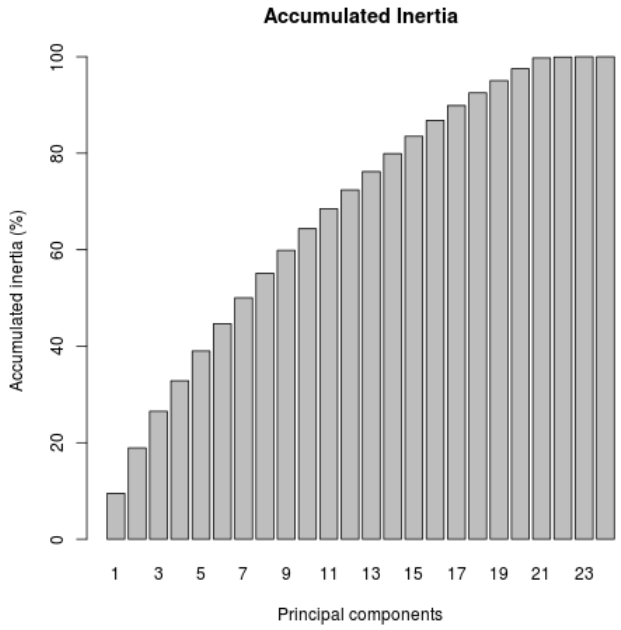

Figure 15: Accumulated percentage of inertia.

7.2 Procedure

Instead of coding an *ad hoc* projection for some axis, what we did instead was systematically and programatically generating the projections for all the combinations of our 14 principal components. The code can be consulted in the appendix. However, for briefness, in this document we will only show the results for the first two iterations of our loop: 1-2 principal components (which explain about 20% of the variance) as axis and 2-3 principal components as axis (which

explain about 15% of the variance). The explanation for what we did for 1-2 follows, but it applies for the 2-3 principal components too.

So, in order to proceed with PCA we will select the 2 principal components which explain most of the variance show in the figure 15, in our case component 1 and component 2 with 10% and 20% of accumulated inertia. As we were told in theory classes, with factorial analysis we want to find the most informative projection planes.

We have to take into account that in PCA the variable which has the highest contribution over the first factorial axis is the longest one among those projection with small angle to axis. Also, if two modalities of qualitative variables project close in the factorial space it means that these modalities are associated.

We plotted a projection of all numerical variables in order to observe the behavior of the variance between our variables. Also we selected which in our opinion were the most important qualitative variable as far as our study was concerned (*match*; remember that our goal was to study what makes a date successful) and plotted all the different levels projected with our numerical two principal components, with the aim of observing how these variables were related with the variance of our numerical values. Then we did the same but with the categorical variables.

7.3 Results

As we can observe in figure 42 we have 2 groups centered in the Y origin, also we can notice that whereas the variance increases values start to scatter from the origin point.

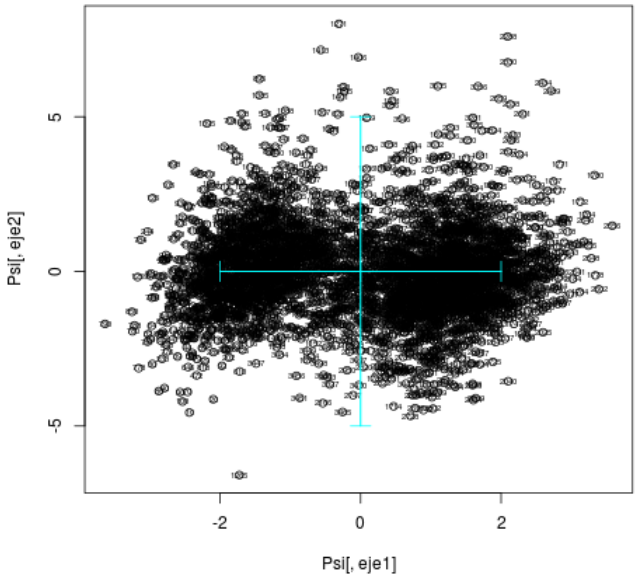

Figure 16: Individuals projection of our first principal components

In figure 17 we have a zoomed projection of all numeric variables. We can observe that most of them are pretty close to 0 variance, but *tuition* and *mn_sat* seem not to follow the other variables,

because they are variables related to the degree of the participant, which is a qualitative variable. However, we can observe that *tuition* and *mn_sat* are correlated as *tuition* might depend on the *mn_sat* mark and people who did not went to the university have a 0 in both variables.

Also we can find that *imprelig* is correlative with *imprace*, which explains that people who gives a lot of importance to the partner's religious background also gives it to race background and vice versa. Another interesting information we can extract from the plot is that income does not seem to be quite related to any other variable (maybe a bit with age) which could tell us that the older people are the more money they earn. Finally we can also determine that people who gives a lot of importance to the attractiveness (*pf_o_attr* does not give too much importance to intelligence ((*pf_o_int*) and vice versa. In other words, these attributes are not related to each other. What is more, this relation is also noticeable with the rate given to all participants (*attr_o*, *intel_o*).

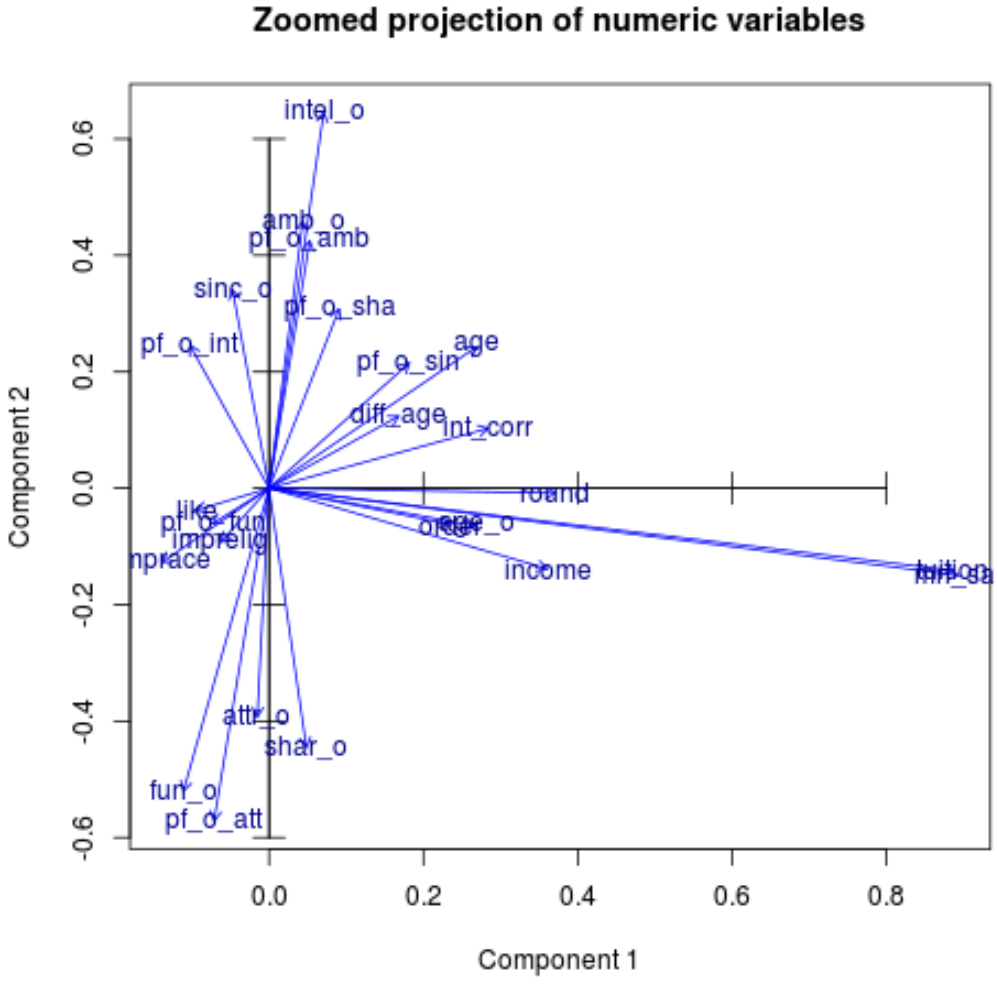

Figure 17: All numerical variables projection.

Once we had checked how the numerical variables relate among each other, we plotted all qualitative variables projected together. In this plot we cannot perceive a clear relation between

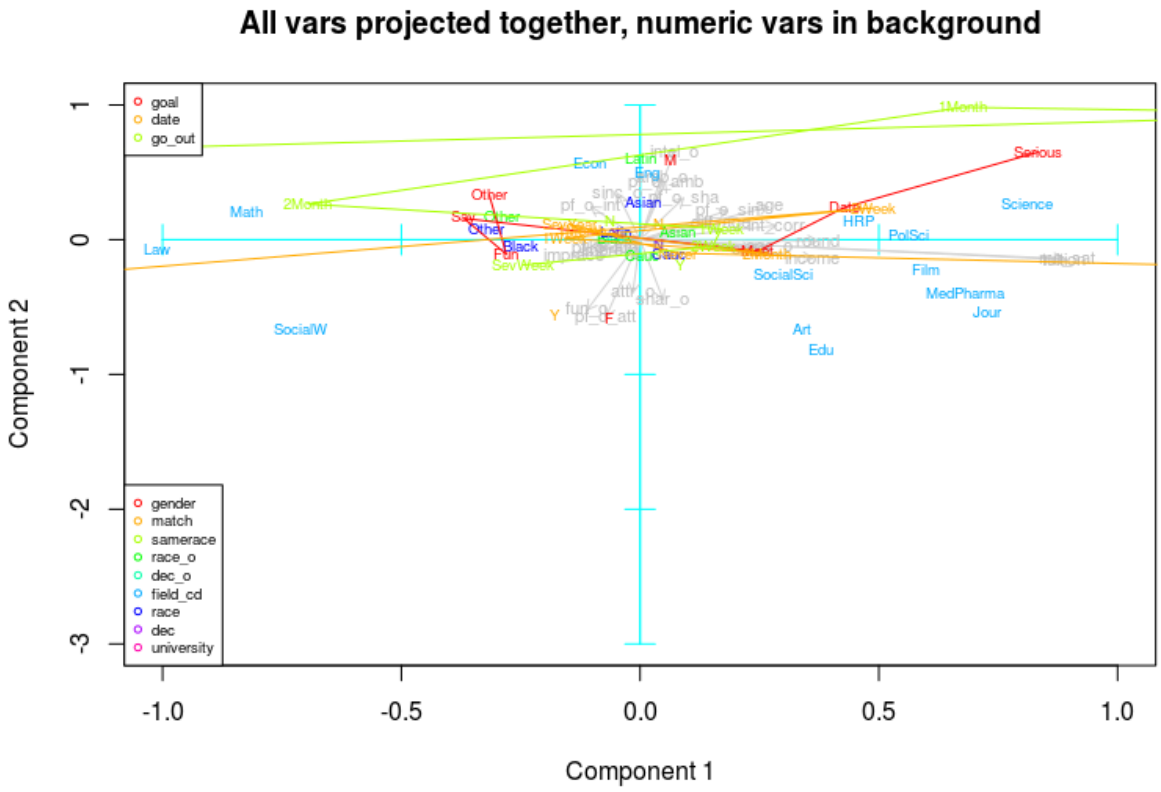

Figure 18: All qualitative variables projection.

match and another qualitative, perhaps we can affirm that there is relation between the fact of being a female and having a match.

In figure 18 we projected the numerical to the previous plot, which gives us more information about the relation between categorical and numerical values. The qualitative variables of the legend in the upper left are the ordinal ones, the other ones are explained in the legend located in the bottom left.

In response to the initial question "Which attributes influence the selection of a romantic partner?" we can observe a clear relation between having a match (orange Y) and fun, having shared interests/hobbies and attractiveness. In contrast it seems that intelligence (*intel_o*) is not related with having a match. These results are pretty coherent with the ones observed in the bivariate analysis.

In addition, we projected both cdgs of levels of the selected qualitative variable (*match*) without representing the individual anymore:

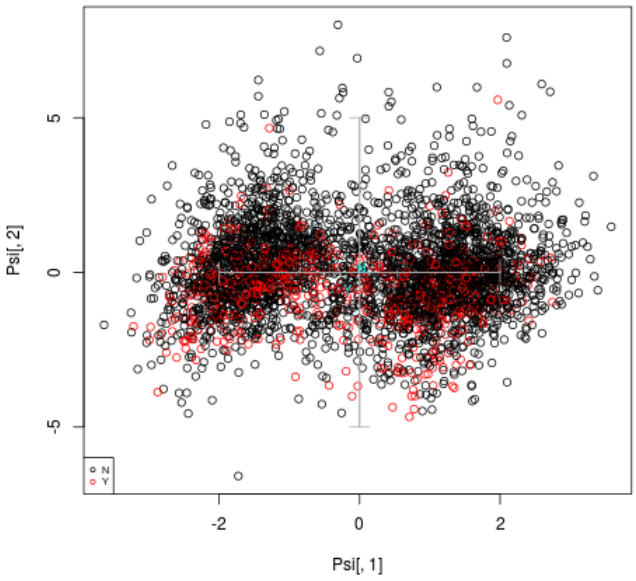

Figure 19: Match

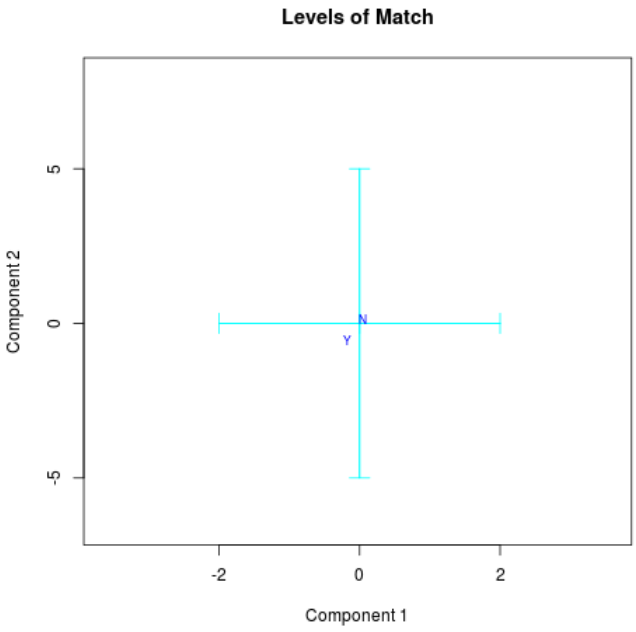

Figure 20: Match

7.4 Extra results: principal components 2 and 3

As we can observe in figure 21, we have a single group centered in the (X,Y) origin. Also it is noticeable the fact that there is a group in the third sector which differs from the other points, as whereas the variance increases values start to scatter from the origin point.

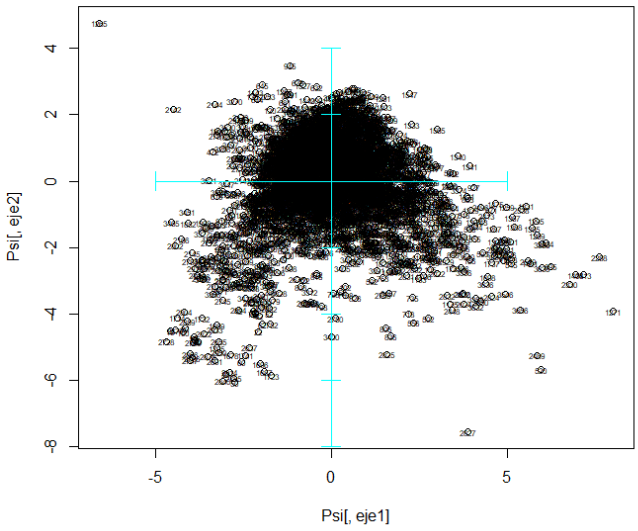

Figure 21: Accumulated percentage of inertia.

In our new zoomed projection of numeric variables we notice groups of variables that relate to each other. Notice how attributes variables(fun_o, shar_o...) are distributed among 2 groups (shar, fun, attr) and (amb, intel, sinc), and each group does not relate to the other. Finally we can also extract information about preferences, and how all preferences are related to each other.

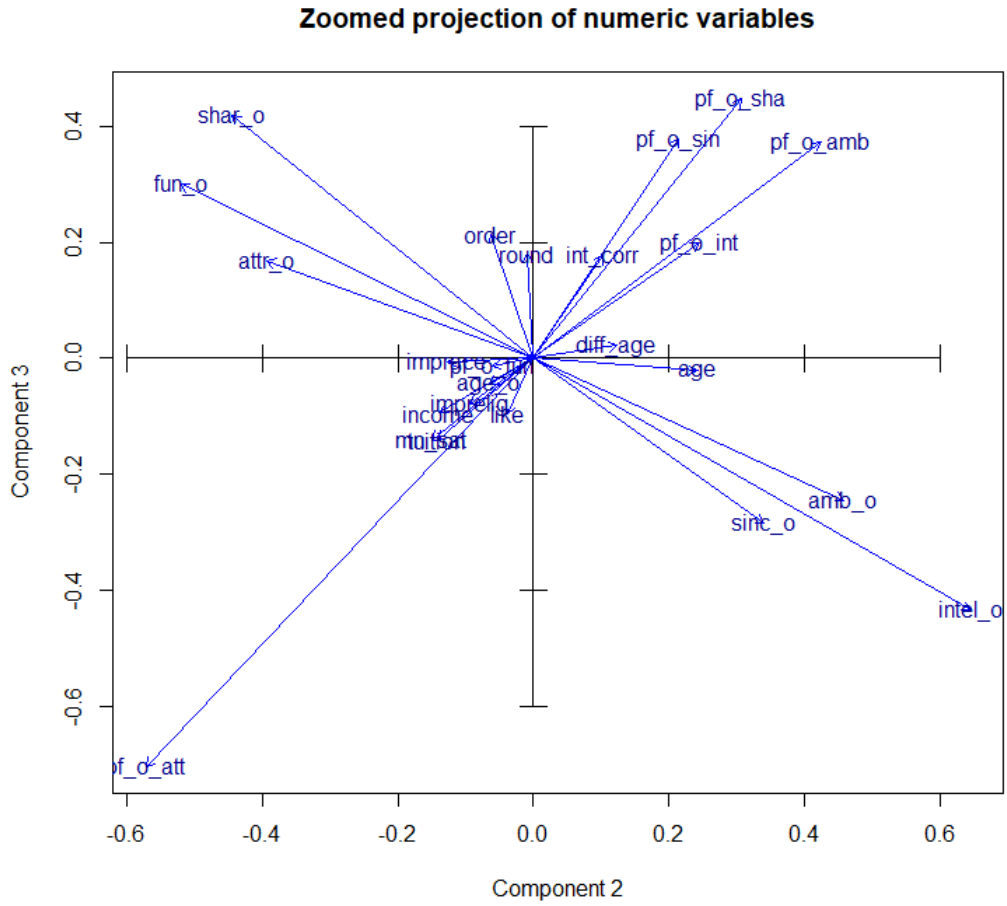

Figure 22: Zoomed projection of numeric variables

The all variables projected plot 23 is quite different from the first one, but most of the relations noticed in the previous one still hold. However we can notice a relation between some field_cd attributes, like Language, Social or medPharma which are quite related to having a match. This plot confirms the previous relations studied in the 1-2 iteration between having a match, being fun, and having shared interest/hobbies and attractiveness.

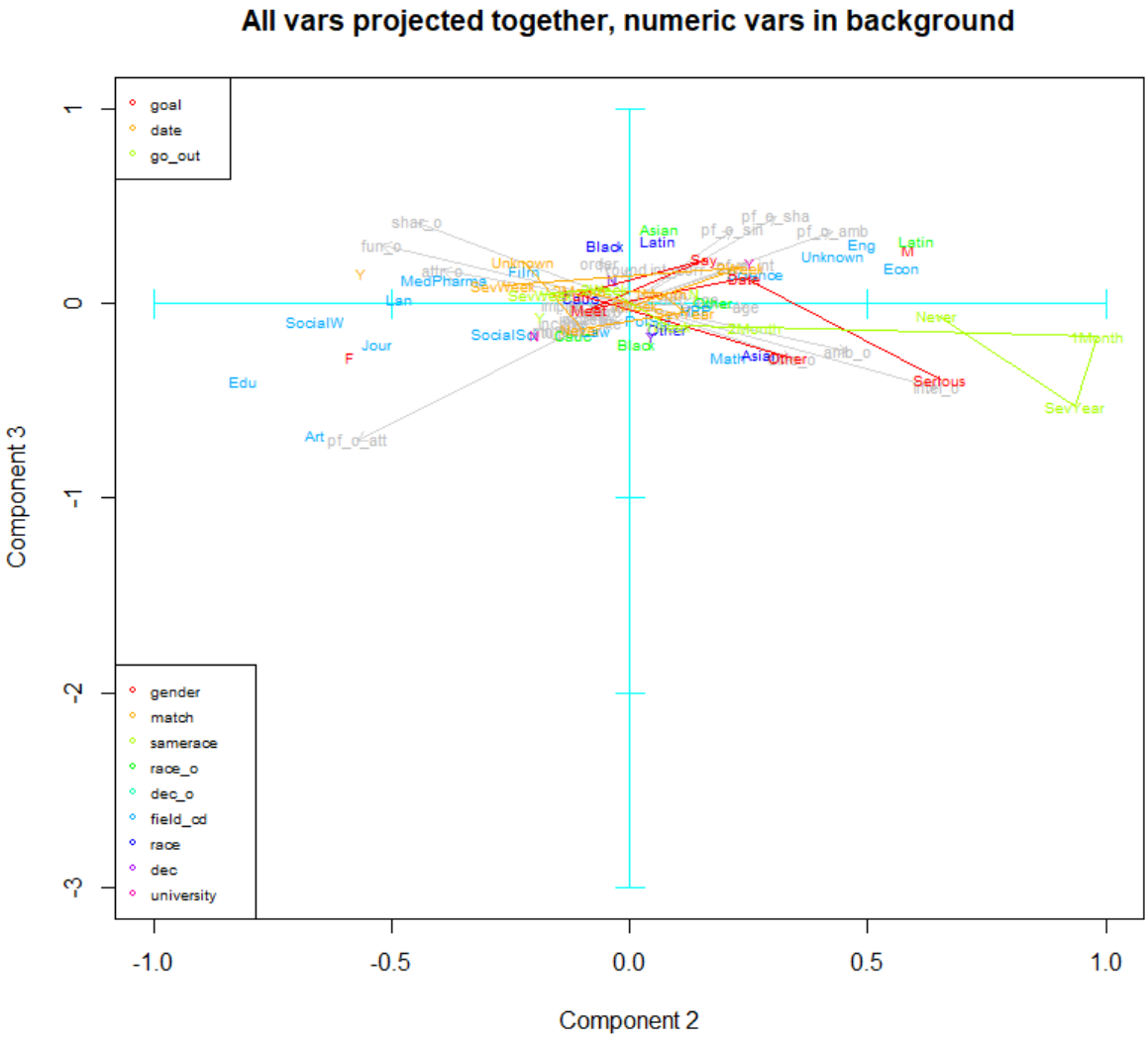

Figure 23: All vars projected together, numeric vars in background

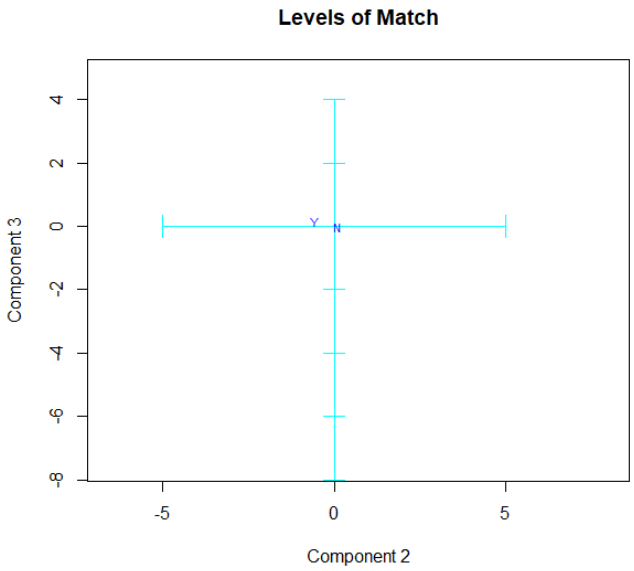

Figure 24: levels of match

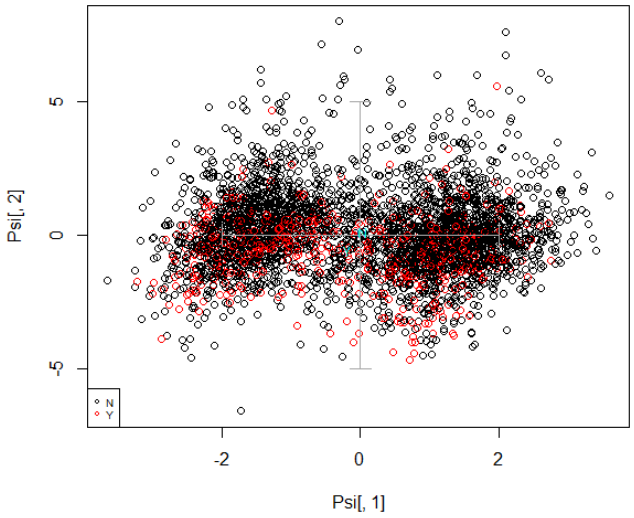

Figure 25: Match

8 Hirerarchical Clustering

Cluster analysis is «the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields »[5].

8.1 Data used

Our original dataset had 8387 records, where each one had 190 variables. More than 100 variables are numerical (some repeated or for different waves), 6 variables are binary and 19 variables are qualitative. The dataset has a 26.44% of missing data.

After pre-processing our dataset has 3656 rows, where each one had 36 variables. 20 variables are numerical, 6 variables are categorical and 5 variables are binary.

8.2 Clustering method used

As requested in the instructions for this project, we will perform a hierarchical clustering.

To solve the problem of having numerical and categorical variables at the same time we will be using the Gower distance, which generates an N^2 end distance matrix.

After that, for the hierarchical clustering analysis of our dataset, we have used Ward's minimum variance method, which is based on the general agglomerative hierarchical clustering procedure. Basically, this method can be understood as follows. We choose the two groups to merge by looking at the value of an objective function which we want to optimize. In the case of Ward's minimum variance method, the objective function is the error sum of squares. This method takes into account the variance instead of using distance metrics and it generates a tree. We will cut the tree in such a way that we obtain the desired number of clusters. We will use R packages that implement this method [6].

[5] Clustering: `http://shodhganga.inflibnet.ac.in/bitstream/10603/36013/11/chapter7.pdf`

[6] Agglomerative Hierarchical Clustering: `https://onlinecourses.science.psu.edu/stat505/node/143/`

8.3 Discuss about number of clusters

There are many different methods in order to determine what is the optimal number of segment. The decision might be subjective and it depends on the method used. One of them relays on visual assessment by visually inspect the dendogram in order to decide if the tree suggest a particular number of clusters, but this method is also subjective [7]`http://www.sthda.com/english/wiki/print.php?id=239`.

Nevertheless, there is another method using some existing algorithms that somehow can help help us to take the right choice in a more objective way.

These algorithms include using a R function from *factoextra* package: *fviz_nbclust()*. We used two different methods, which are: elbow method and silhouette.

We can see from the two figures below the optimal clusters number according to these algorithms: 3.

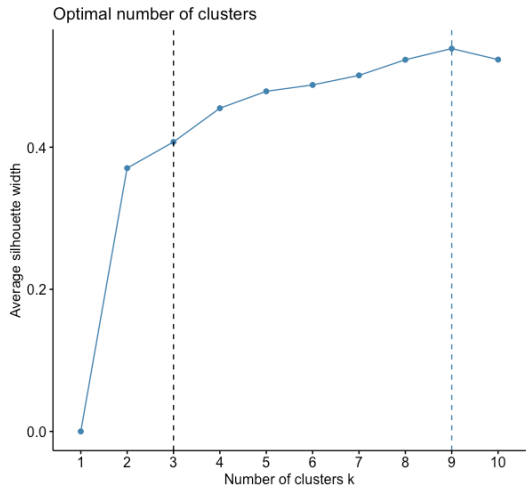

Figure 26: Optimal number of clusters using Silhouette method.

[7] Determining the optimal number of clusters: 3 must known methods - Unsupervised Machine Learning

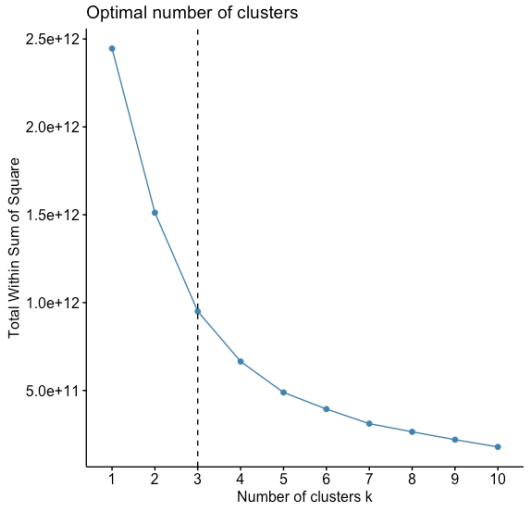

Figure 27: Optimal number of clusters using Elbow method.

8.4 Clusters size

We can see in the table below the amount of data in each cluster.

Cluster	Number of rows
1	1600
2	1213
3	843

Table 3: Size of each cluster

8.5 Resulting dendogram

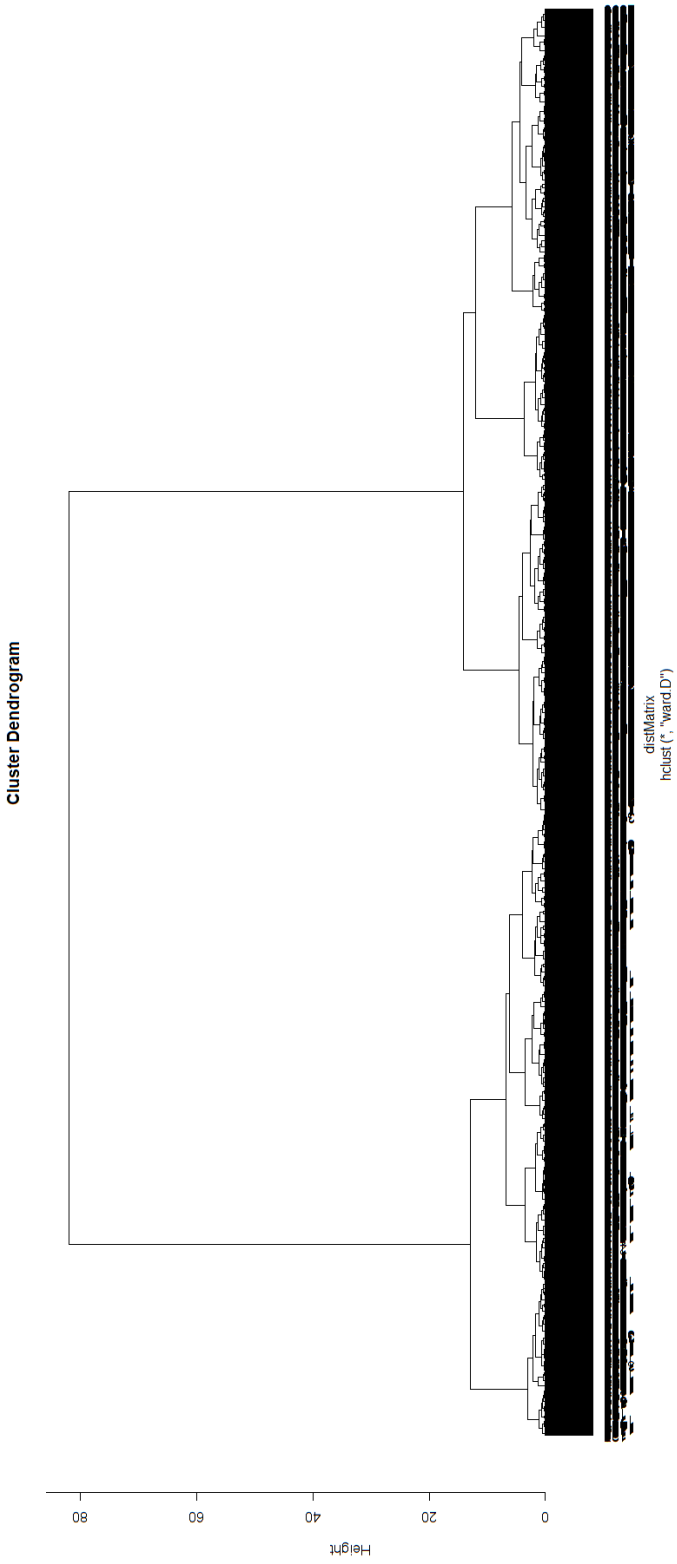

Figure 28: Dataset clusters dendogram.

9 Profiling of clusters

To start off with the clusters profiling, we have performed the Pearson's chi-squared test for all categorical and binary variables of our dataset. This way we can know whether our clusters are well established and how dependant is each one of these variables to their actual cluster.

So, after running the test, we have seen a generally low p-value among almost all of these variables. That means we cannot discard our null hypothesis which is that the variables are not dependant on the cluster.

Chi squared test results are shown in the table below.

Variable	**X-squared**	**p-value**
gender	8.9301	0.0115
match	159.6	2.2e-16
samerace	85.201	2.2e-16
race_o	15.997	0.04242
dec_o	501.3	2.2e-16
field_cd	812.14	2.2e-16
race	376	2.2e-16
goal	334.47	2.2e-16
date	440.09	2.2e-16
go_out	517.58	2.2e-16
dec	501.3	2.2e-16
university	3519.5	2.2e-16

Table 4: Chi-squared and p-value for categorical and boolean variables

On the other hand, for numerical variables we have done a simple mean operation for each of the different clusters.

The results of the arithmetic mean for each variable and cluster are shown in the table below.

Variable	**Cluster 1**	**Cluster 2**	**Cluster 3**
round	16.04312	17.31410	17.67141
order	8.506250	8.985161	9.416370
int_corr	0.1560750	0.2065128	0.2514472
age_o	25.67563	26.19456	26.76750
pf_o_att	22.98625	24.20610	24.41044
pf_o_sin	16.92063	17.65622	17.86714
pf_o_int	22.12750	20.58203	20.38078
pf_o_fun	17.17000	17.34378	16.9822
pf_o_amb	9.688125	9.356966	9.288256
pf_o_sha	11.09437	10.87881	11.09964
attr_o	15.38500	15.44847	16.13642
sinc_o	18.40125	18.29349	18.11862
intel_o	19.01250	19.25639	18.66311
fun_o	16.14313	15.90437	16.09371
amb_o	17.61375	17.70816	17.30486
shar_o	13.67812	13.61006	13.95492
age	25.68500	26.26793	26.62040
mn_sat	2.3375	1284.8013	1301.1198
tuition	36.97812	21812.44683	22251.20641
imprace	3.715625	3.034625	4.148280
income	41197.10	47766.52	52105.42
like	6.205938	6.427040	5.580071
imprelig	3.905000	3.574608	3.945433
diff_age	3.595625	3.281946	4.033215

Table 5: Average values for each numerical variable and cluster

We have also checked the match rate for each of the clusters, results of which are shown in the table below.

Cluster	**% of match**
1	17.312500
2	25.886232
3	4.507711

Table 6: Percentage of match for each cluster

After these tables, we can see some of the differences between the three clusters. We are going to outline them, but the reader should take into account that, as stated before, the p-values are low and therefore we cannot affirm that these differences are significant enough, so take the following results with care. The first difference, as shown in table 6, is the rate of match. While the third cluster presents the lowest match rate with roughly 4.5 percent, over 1 out of 4 meetings put classified in the second one end in being a match.

Another noticeable difference we find is about cluster 1. This one includes those meetings which the main subject has the least grade on the exam to enter college (mn_sat) as well as the sum of money charged to enter college (tuition).

Moreover, we can see by joining tables 5 and 6 that subjects who care less about his/her date generally end up having a higher match rate, as we could have guessed. This could specially importamt in a country so much ethnic-diverse as United States is.

Same thing happens age difference between two dating mates (diff_ages), where we can see an inversely proportional relation between age difference and match chance.

Our next step in profiling is the study of the clusters and variables using plots and charts.
First of all, we grouped the data into two groups: those observations who ended up in a match and those which did not. We also divided them into categorical/binary and numerical. We can see the results in the following six figures.

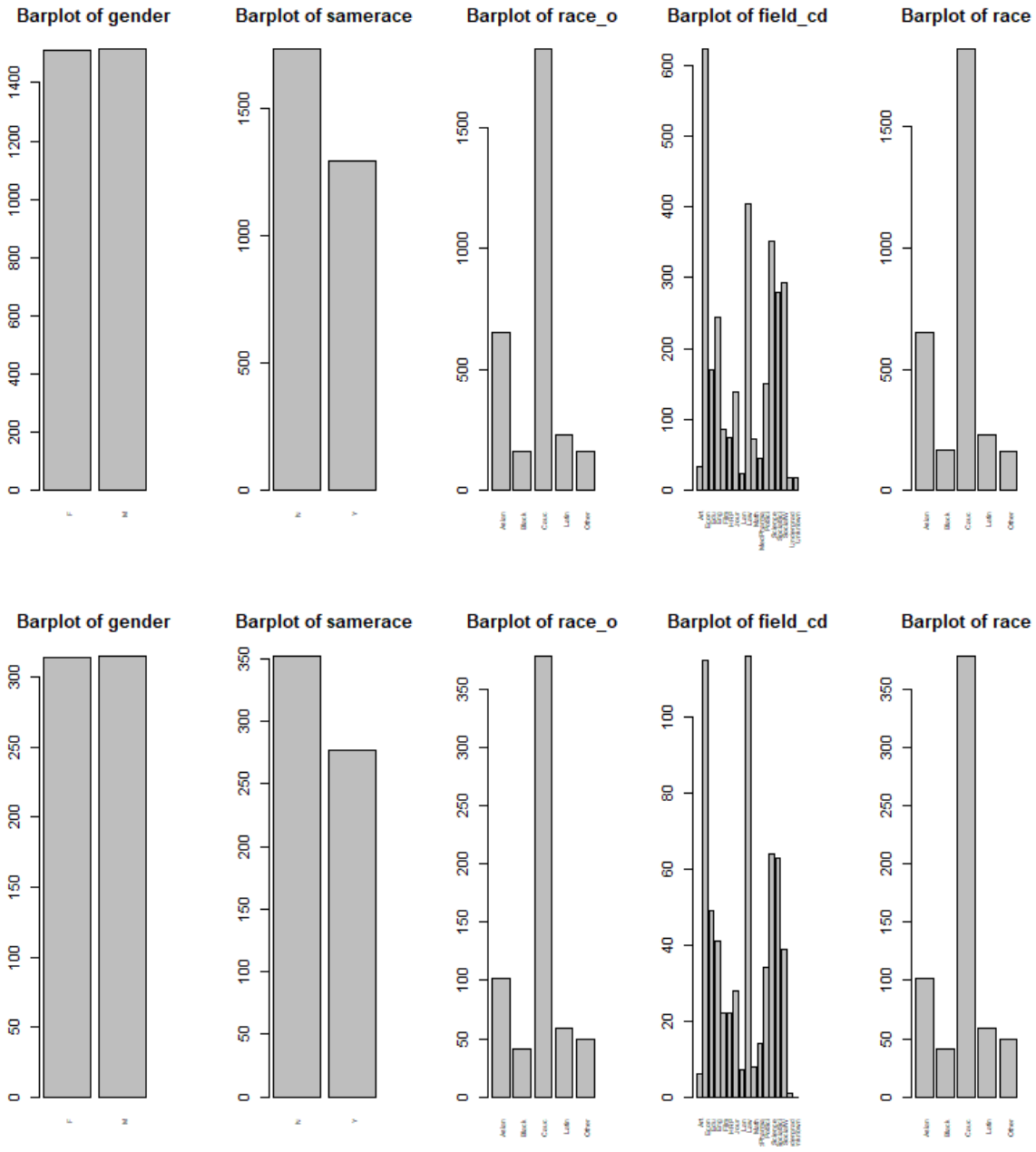

Figure 29: Study of categorical and binary variables depending on variable match. Above, if it is not match; below if it is a match.

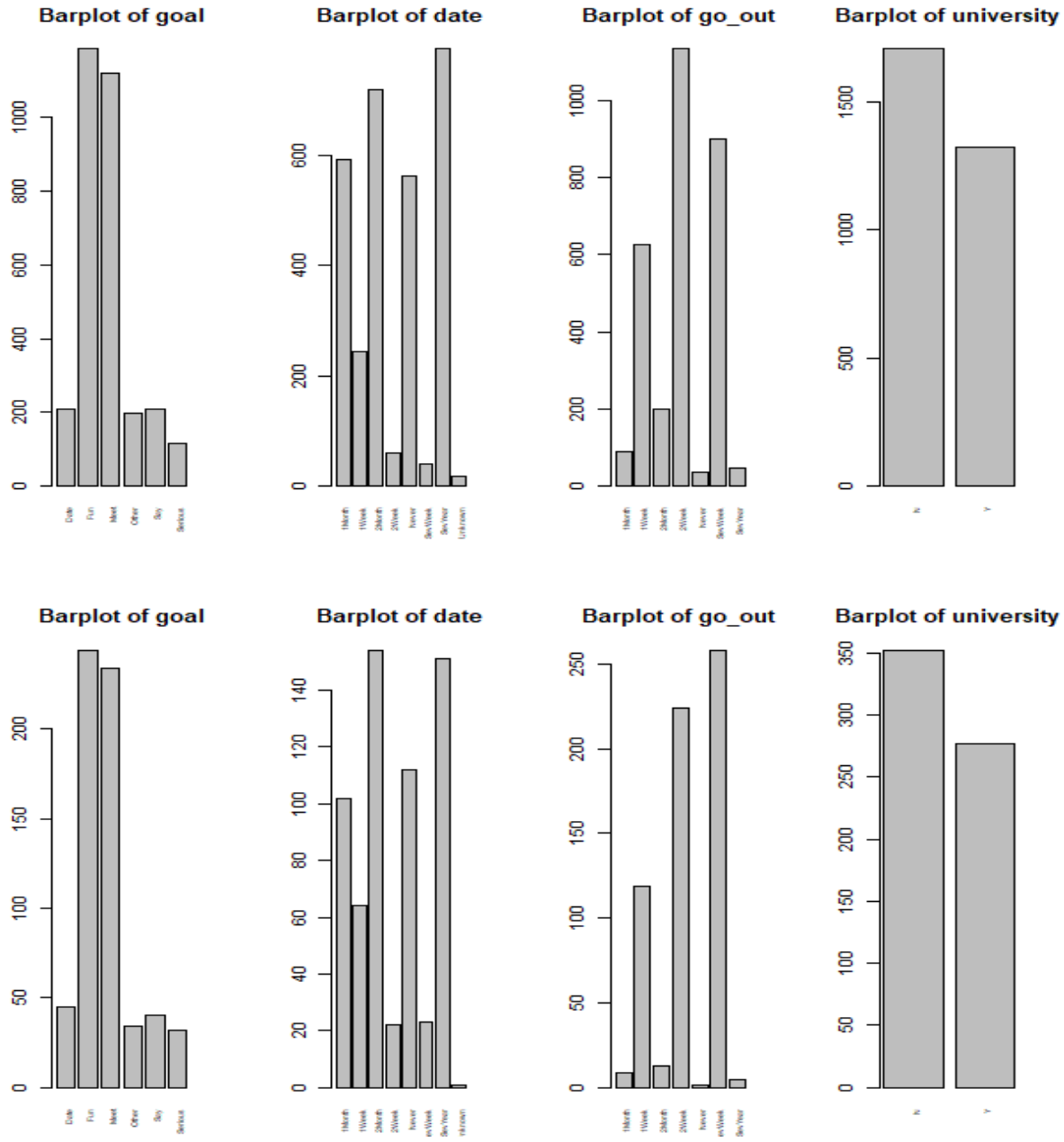

Figure 30: Study of categorical and binary variables depending on variable match. Above, if it is not match; below if it is a match.

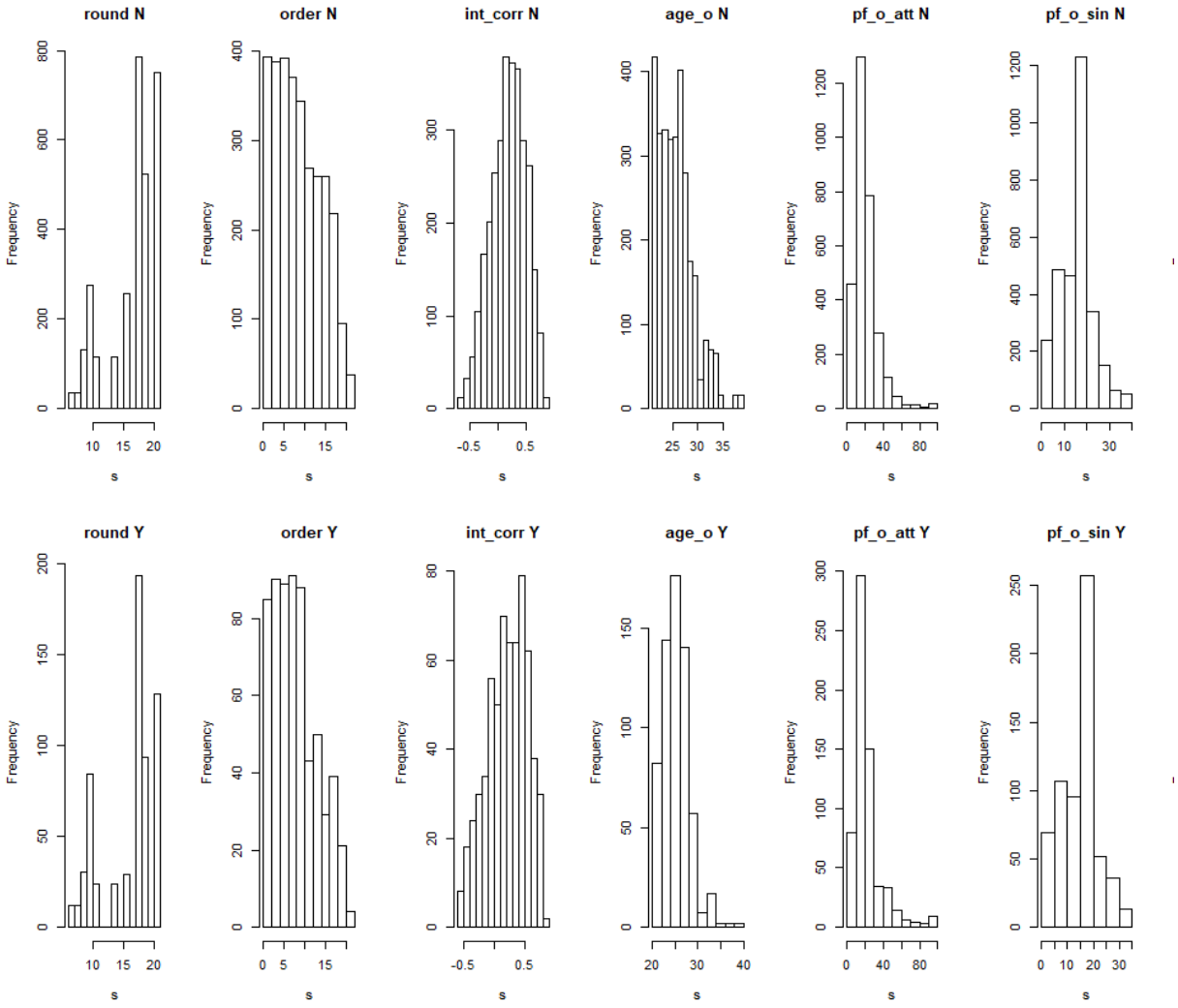

Figure 31: Study of numerical variables depending on variable match. Above, if it is not match; below if it is a match.

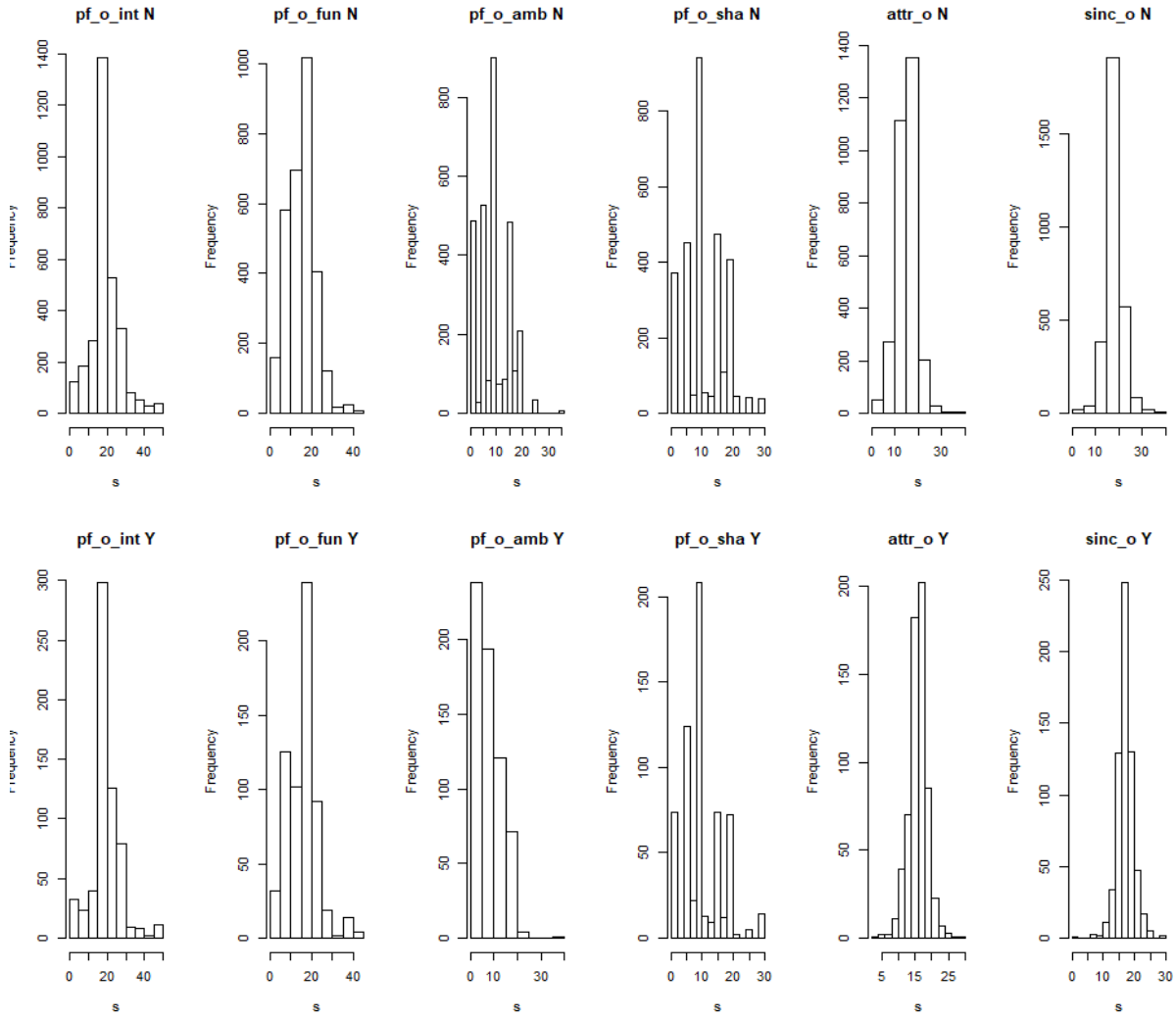

Figure 32: Study of numerical variables depending on variable match. Above, if it is not match; below if it is a match.

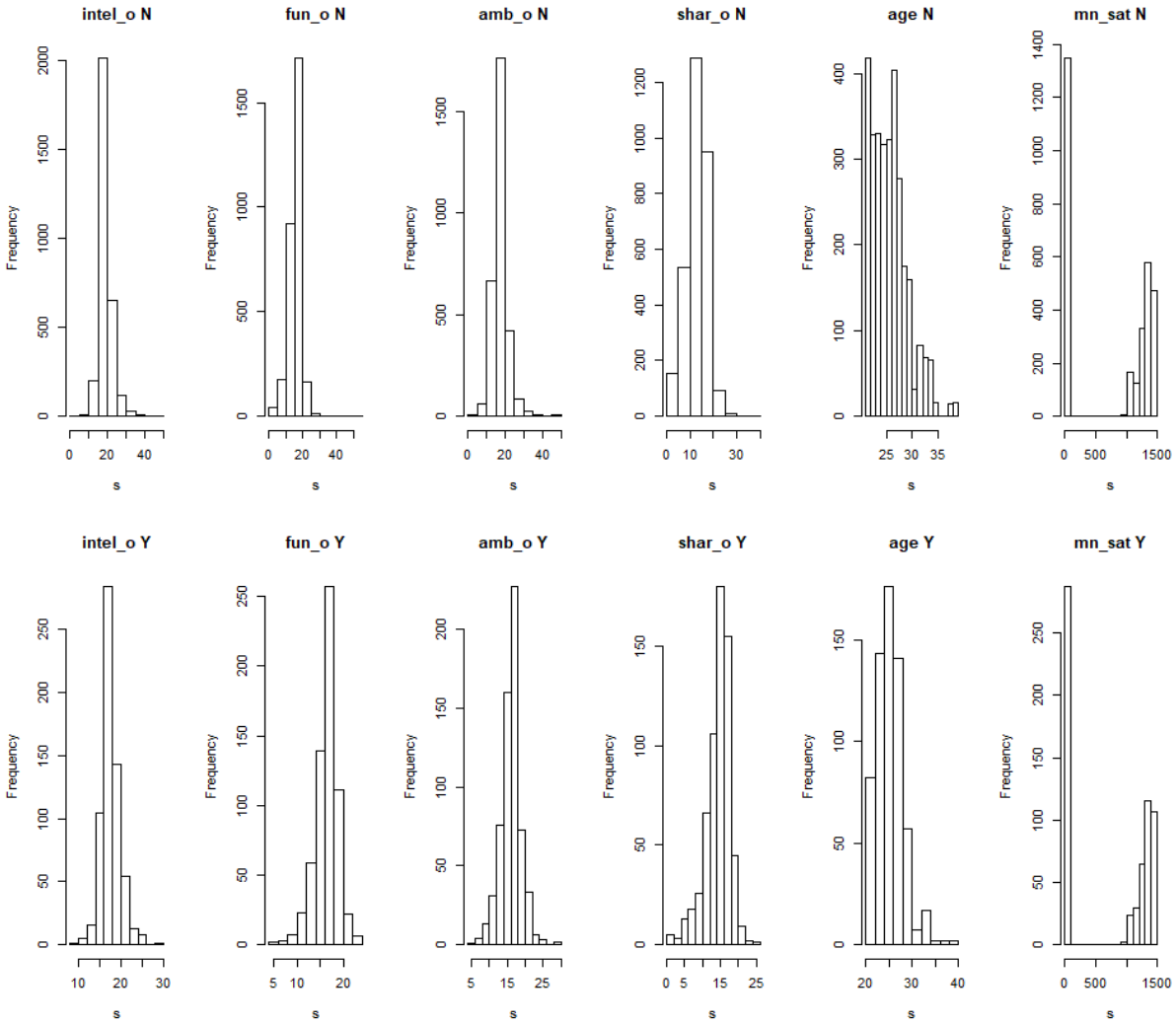

Figure 33: Study of numerical variables depending on variable match. Above, if it is not match; below if it is a match.

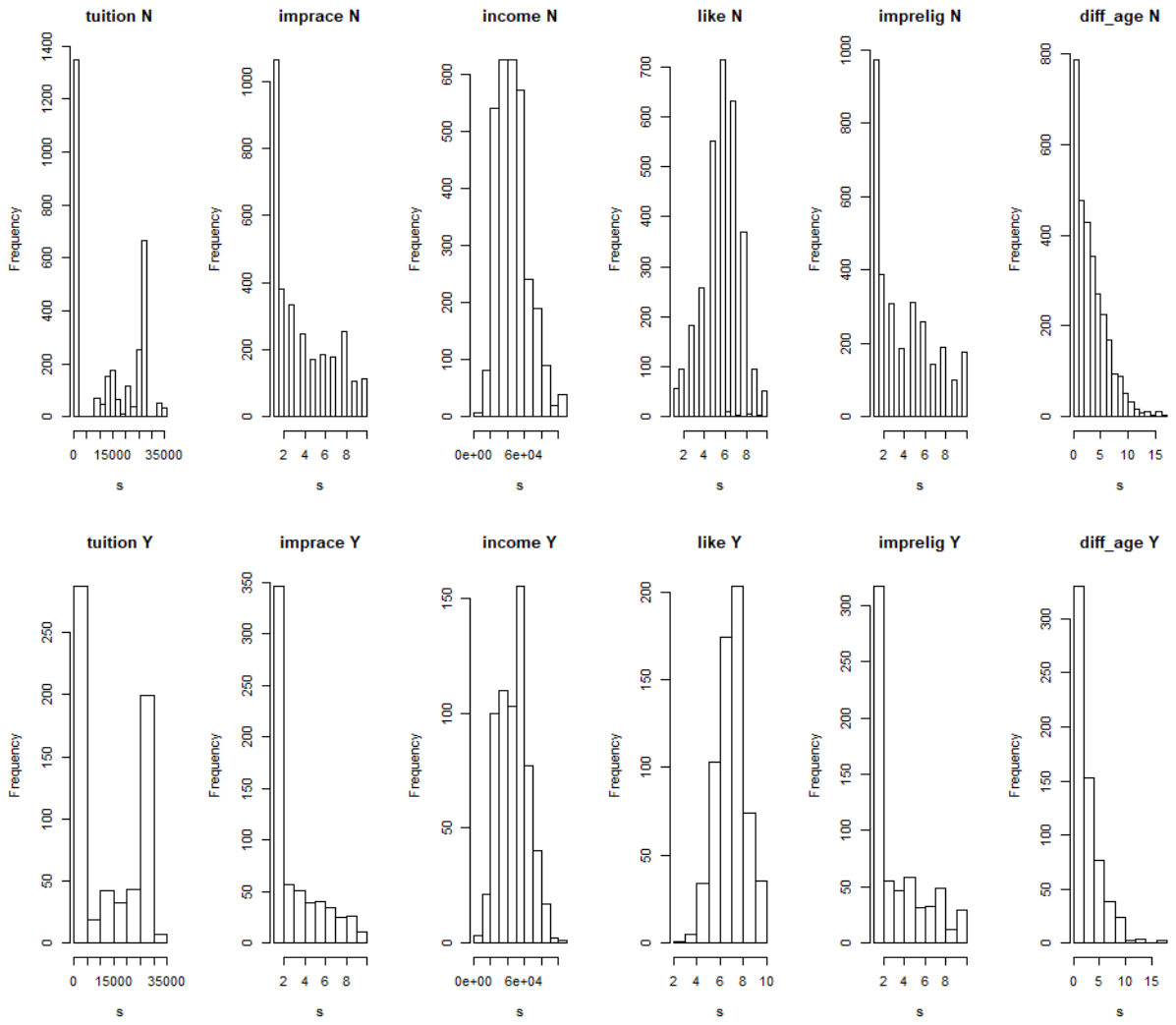

Figure 34: Study of numerical variables depending on variable match. Above, if it is not match; below if it is a match.

We are going to describe and analyze the resulting plots. Again, the reader should take into account that the p-values are low, so the following points might not be significant enough.

We can see in figure 34 that some of the important properties that most influence in whether a date ends up in a match or not are the importance of the religion, the importance of the race and difference of age.

For the importance of the religion, we can observe that the difference in the amount of subjects that do not mind about this at all and the amount of people that mind is larger when a date ends up in a match than on the other way.

It occurs the same way for the importance of the race, favouring those subjects who do not mind at all about his or her partner's race or ethnic group.

For the difference of age, we can observe that a match-ending date has higher chance to happen when the two subjects have a very low or null difference of age.

Income also has a positive direct relation towards a date ending in a match. It seems that subjects who end up matching have a slight higher income.

In figure 30 we can see a very subtle fact. This is that people who go out several times a week (this is the variable which reflects going out most times a week) tend to be more positive towards having a match.

Quite in a similar way it occurs to variable date. Those subjects who go out several times a week (maximum value for the variable) have a higher chance to end the date up in a match.

Regarding to field of study, we see a clear enhancement for the law field. While the other fields remain quite equal from a plot to the other, law field increases the most by far.

We do not see any evidence of whether being the same race affects towards the date ending up in a match or not. Although, we can see a subtle change in the race plot, where Asian subjects are more likely (not by much although) to not ending the date as a match.

Also, those who prefer an ambitious partner (figure 32) seem to have less chances to have a match. As we can observe, while most of subjects who do not end with a match have a more or less high preference about this, those who finally match at the end of the date present a lower rate of preference of ambition.

We have also grouped the data in three groups depending on which cluster is each observation fit in, as we can see in the next eight figures. Note that, just like we did before, we have divided them on whether they are categorical/binary variables or numerical.

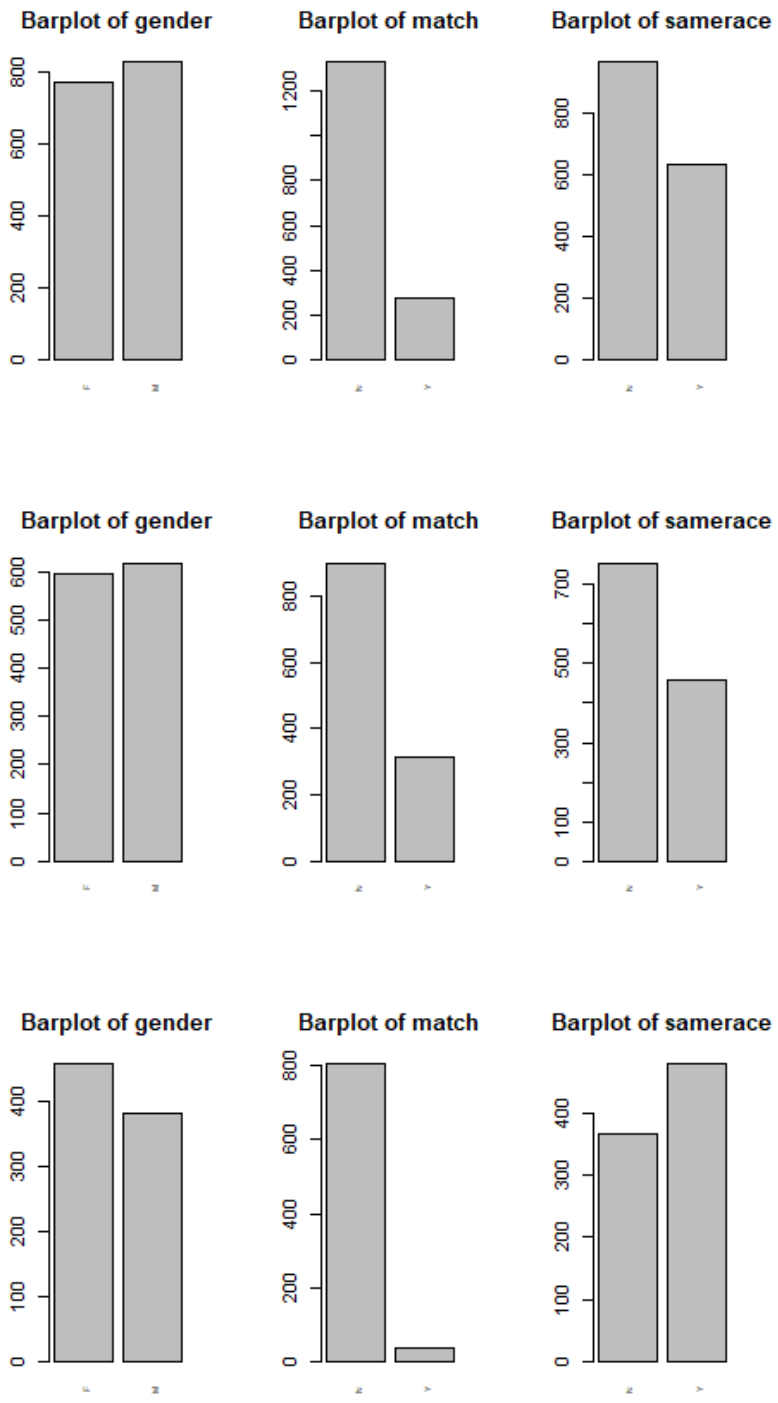

Figure 35: Study of categorical and binary variables depending on the cluster they are fit in. From top to bottom, clusters 1, 2 and 3.

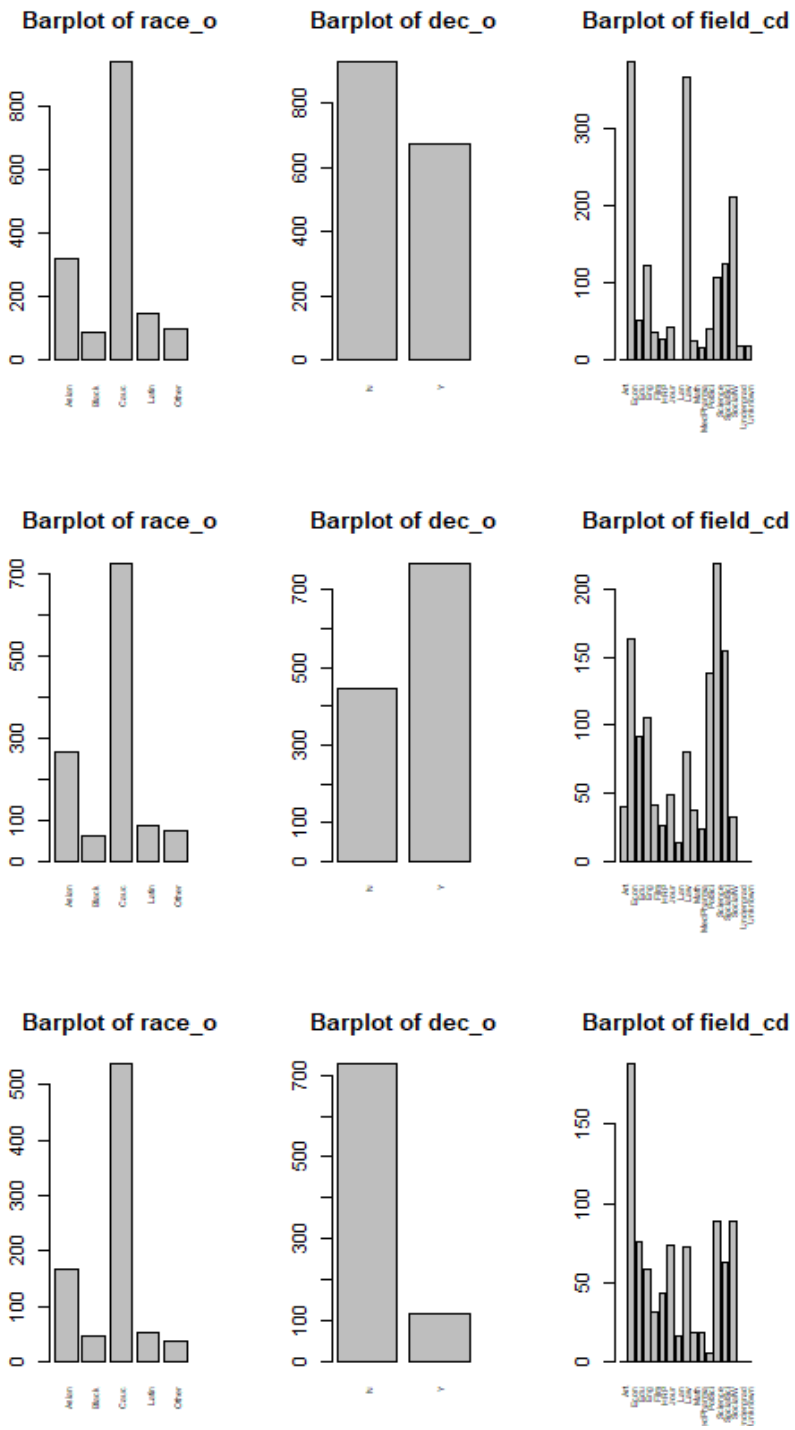

Figure 36: Study of categorical and binary variables depending on the cluster they are fit in. From top to bottom, clusters 1, 2 and 3.

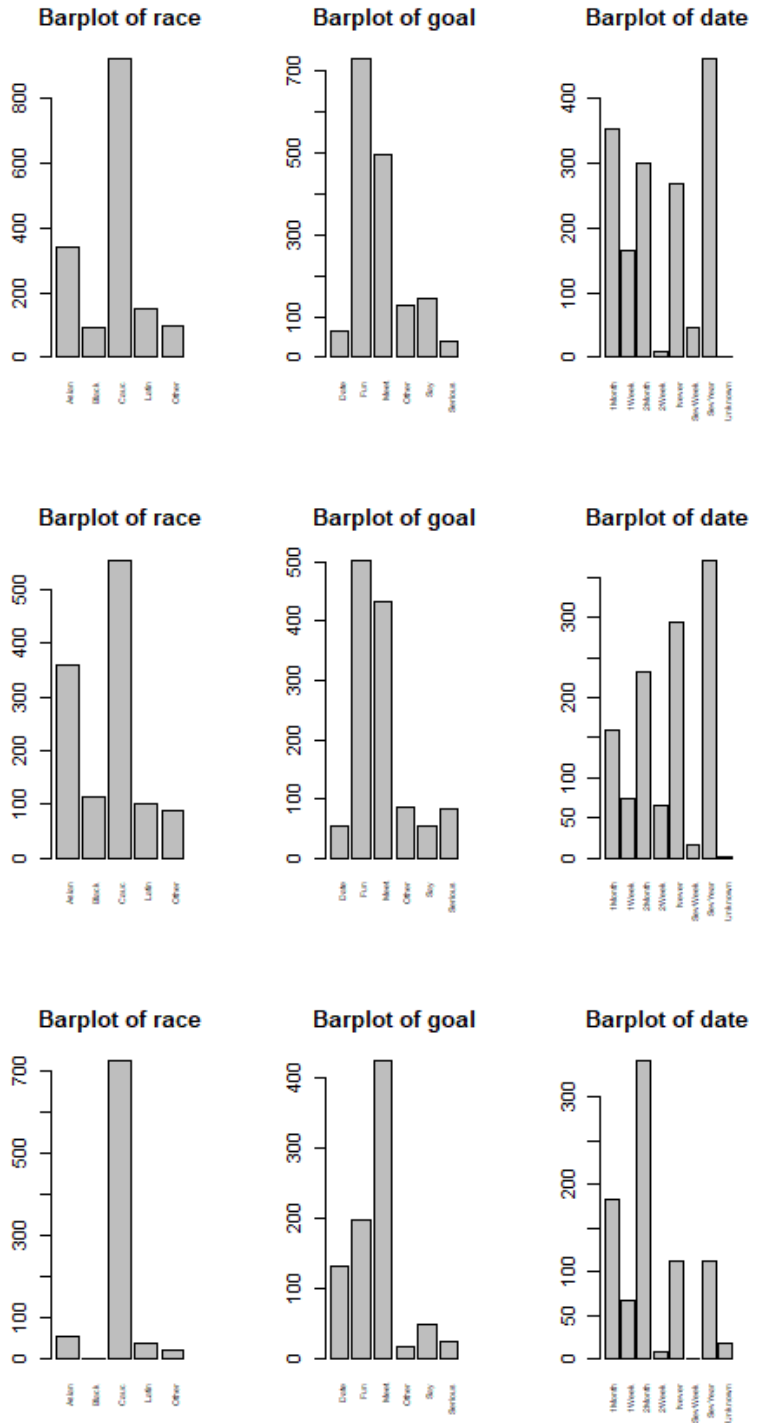

Figure 37: Study of categorical and binary variables depending on the cluster they are fit in. From top to bottom, clusters 1, 2 and 3.

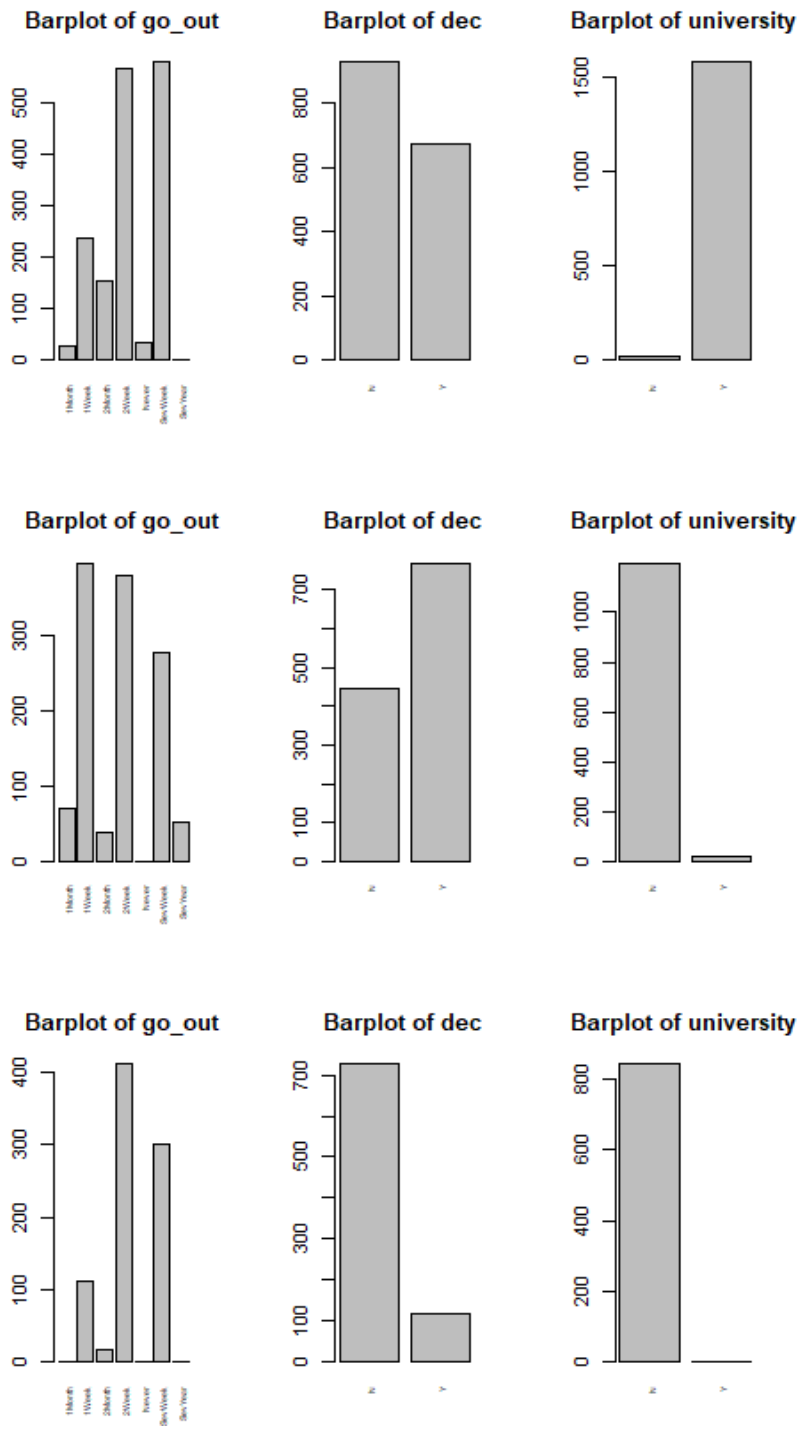

Figure 38: Study of categorical and binary variables depending on the cluster they are fit in. From top to bottom, clusters 1, 2 and 3.

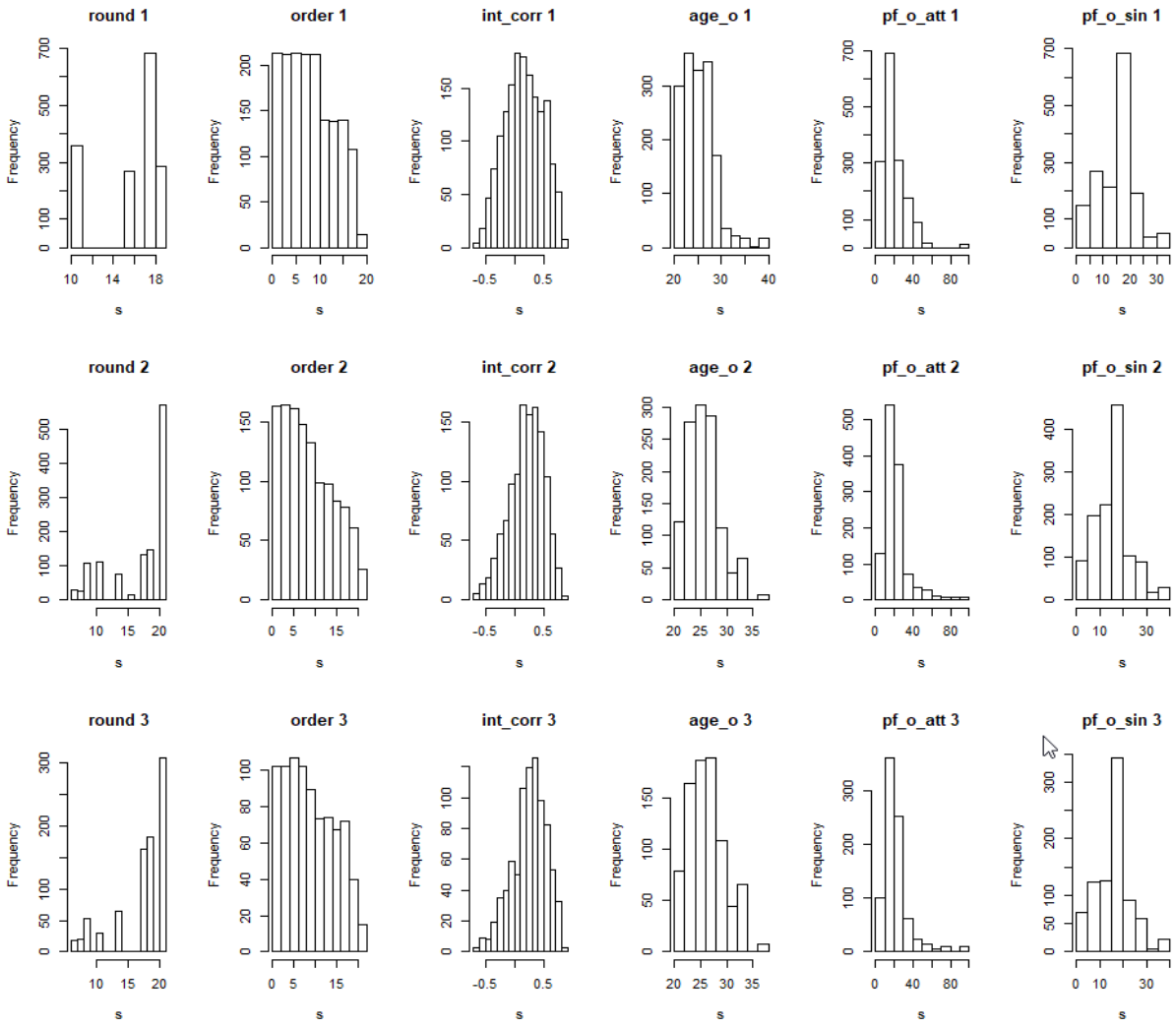

Figure 39: Study of numerical variables depending on the cluster they are fit in. From top to bottom, clusters 1, 2 and 3.

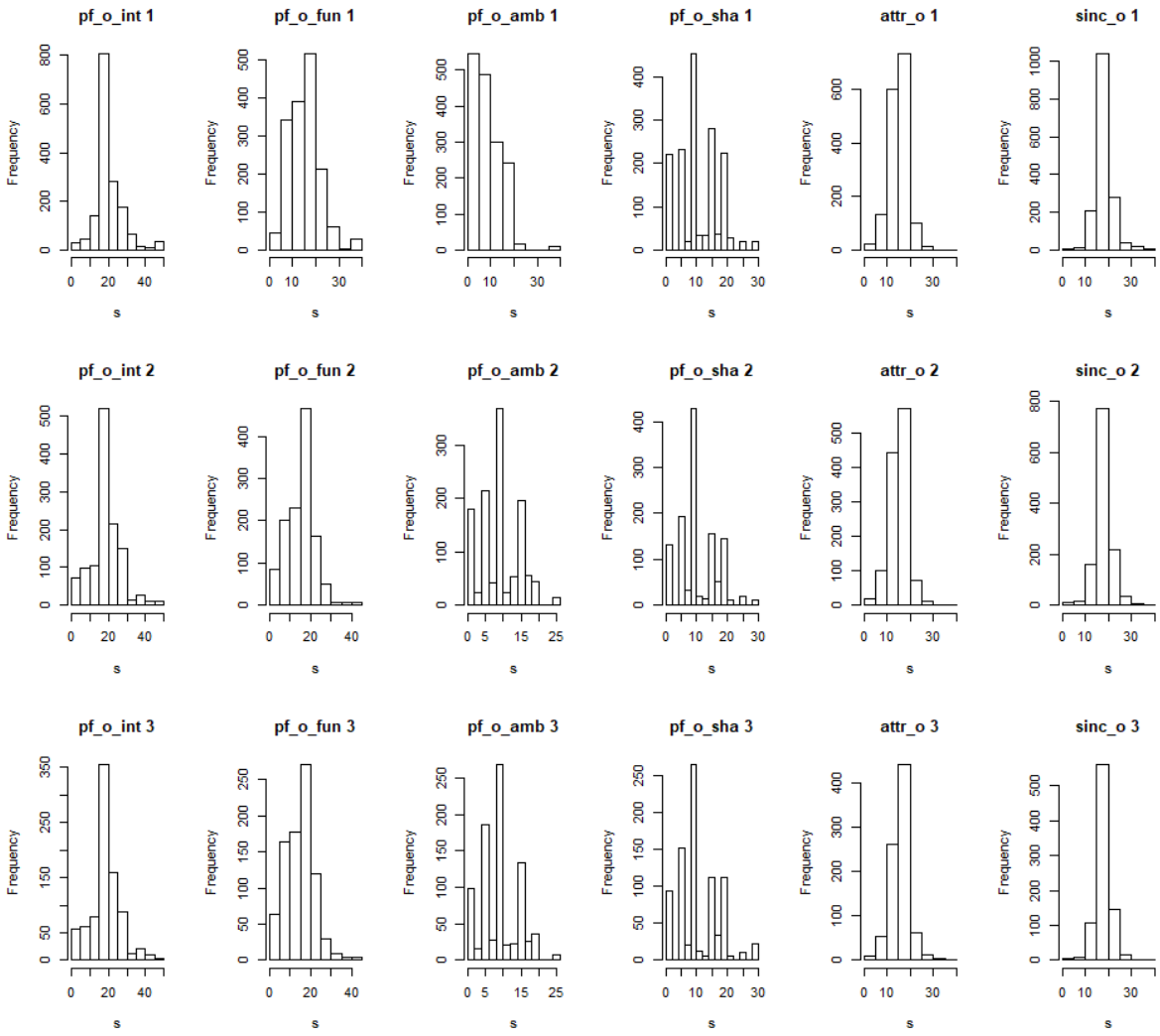

Figure 40: Study of numerical variables depending on the cluster they are fit in. From top to bottom, clusters 1, 2 and 3.

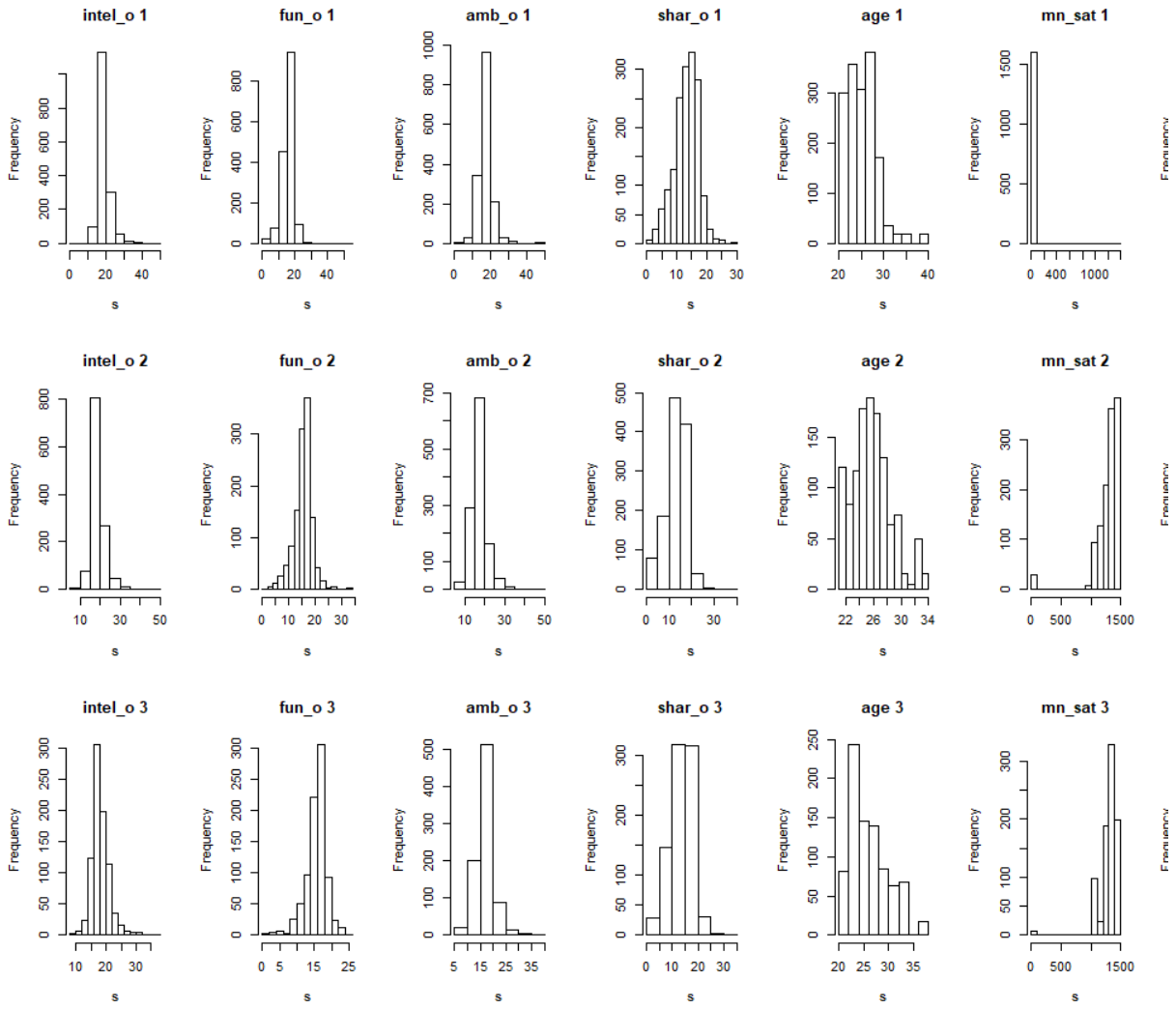

Figure 41: Study of numerical variables depending on the cluster they are fit in. From top to bottom, clusters 1, 2 and 3.

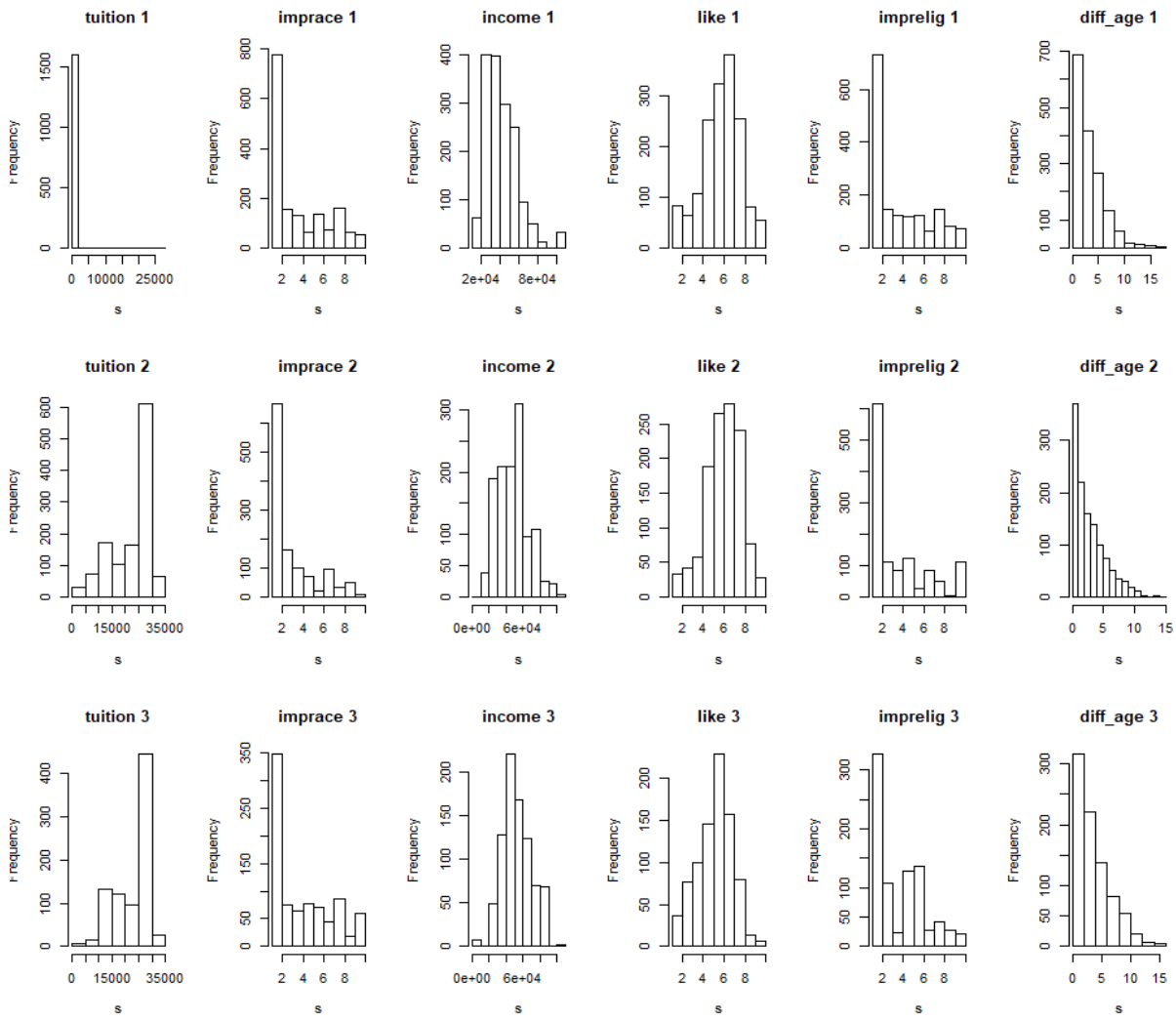

Figure 42: Study of numerical variables depending on the cluster they are fit in. From top to bottom, clusters 1, 2 and 3.

First of all, we can see that cluster 3 presents a very low rate of match observations, while first and second clusters have a higher rate, being the second one the one with the highest of all (figure 35).

We can observe that one of the main criteria to separate clusters has been whether the subject has been to University or not (figure 38). As we see in figure 38, while most subjects that have been to college are placed in cluster 1, the other two clusters mainly have subjects who have not attended University.

Also, we can get to a similar conclusion about importance of religion than we did earlier on. We can observe that cluster 3 is the one that presents a higher rate of subjects who give more importance to the religion of their date (figure 42). If we merged this observation with cluster 3 being the one with the lowest match rate by far, we could see this as a reason.

This last cluster is the one presenting a higher rate of young subjects as well (figure 41).

Moreover, clusters 1 and 2 present a very higher rate of subjects who go frequently on a date compared to cluster 3, whose subjects tend to go on a date in a less frequent manner.

So, in summary, we could say this is what characterize each cluster:

- **Cluster 1:** This cluster's main characteristic is that most of the subjects that it contains have attended University. This cluster's subjects' field of study concentrate between business/finances and law.
- **Cluster 2:** This cluster is the one with the highest match rate. Together with cluster 3, this cluster presents a very poorly number of subjects who have attended University. Moreover, this cluster's subjects' most frequent field of study is science and most frequent race is asian.
- **Cluster 3:** This cluster has a very poor match rate while presenting the highest rate of all three in subjects whose field of study is business or finances. Cluster 3 also presents the highest rate of Caucasian subjects as well as subjects whose main goal of participating is to meet new poeple. Finally, subjects in this cluster tend to rate religion's importance higher.

10 Global discussion

Recall that our goal was to analyze the factors which made a date successful.

Comparing the results obtained with the basic statistical and descriptive analysis and the PCA, we can affirm that they are coherent up to a point. In both of them we arrived to the following results:

- Intelligence is NOT a nice feature to have if your goal is to have a match.
- In contrast, being fun and attractive gives you many changes of getting a match.
- Women are more likely to get a match. Men say yes to many women, but the opposite does not hold.

However, with bivariate analysis we observed additional results than we could not appreciate in the PCA: people with a career related to maths are less likely to get a match, and race is not quite important to have a match. The opposite held, too: in PCA we observed that common interests were related to having a match, and this information could not be extracted from the bivariate analysis. Also, while doing PCA we observed income does not seem to be quite related to any other variable apart from age.

As far as the clustering and profiling are concerned, we have found that the difference of age is a relevant factor to get a match (the greater the difference, the less matches people get). The information about the difference of age is not necessarily contradictory with the results obtained with the bivariate and factorial analysis, but with the latter methods we had not found them. However, for this conclusion and the following ones, again, we have to take into account that the p-values are low.

What we have found to be coherent with the results obtained by both bivariate and PC analysis is that the marks obtained to access the university do not seem to be a relevant factor as far as dating is concerned. However, in profiling we have seen that income is a relevant factor, but the difference might not be significant. Those subjects who go out several times a week have a higher chance to end the date up in a match. We have found other relations stated in the profiling section, too.

Neither of the three methods have found the race to be a relevant factor.

11 Conclusions

In order to sum up this project we can outline the following conclusions:

- We consider that we have successfully applied to real world data the data mining techniques (pre-processing, basis statistical and descriptive analysis, PCA, hierarchical clustering and profiling) which we have been learning during this course.
- However, we have learned the hard way that in practice data mining is much more difficult than in theory.
- That is to say, our dataset (and many other real world datasets, we assume) has proven to be particularly difficult to work with. Many techniques cannot be applied directly and you have to adapt them to your particular case.
- Still, we consider that we have extracted some useful information (knowledge discovery) regarding the factors which make dates successful. We have obtained a similar answer to the initial question by using different techniques applied by independent teams (different tasks were assigned to different members of our group).
- We can affirm that being (or at least being considered) funny and attractive increases the chances of having a match. On the contrary, being perceived as intelligent appears not to be a good idea if the goal is to be liked by the other sex. We have found a weak relation between having common interests and getting a match. Also, women get more matches than men (so matches are more evenly distributed among women).
- The differences between clusters found in the profiling step apparently were not significant enough (the p-values were low). However, as far as we know social sciences tend to find low p-values in general, so it is an issue but we think it is a very common issue in this field.
- As a suggestion to improve this project, with perspective, we believe that perhaps it would have been a better idea to further semantically collapse the dataset in order to have still less variables but more meaningful.
- As far as the team work, working plan and planning are concerned, we all agree that we have achieved our goals and everybody has contributed to the team by doing the assigned tasks. Nevertheless, the final working plan and planning differs from the original one. The main drawback we have had is the fact that pre-processing took so much longer than expected. Also, managing a team of as many as 6 members has proven to be a challenge (many different ideas, many different timetables...). This point will be detailed in the next section.
- As far as the team work, working plan and planning are concerned, we all agree that we have achieved our goals and everybody has contributed to the team by doing the assigned tasks. Nevertheless, the final working plan and planning differs from the original one. The main drawback we have had is the fact that pre-processing took so much longer than expected. Also, managing a team of as many as 6 members has proven to be a challenge (many different ideas, many different timetables...). This point will be detailed in the next section.

- As a future work, we suggest studying other aspects included in this dataset that were out of the scope of this project, basically analyzing the difference between self-perceptions and actual perceptions by other people.

12 Working plan

In the following chapter the initial and final Gantt charts will be shown as well as the risk plan that was developed during the project.

12.1 Gantt chart

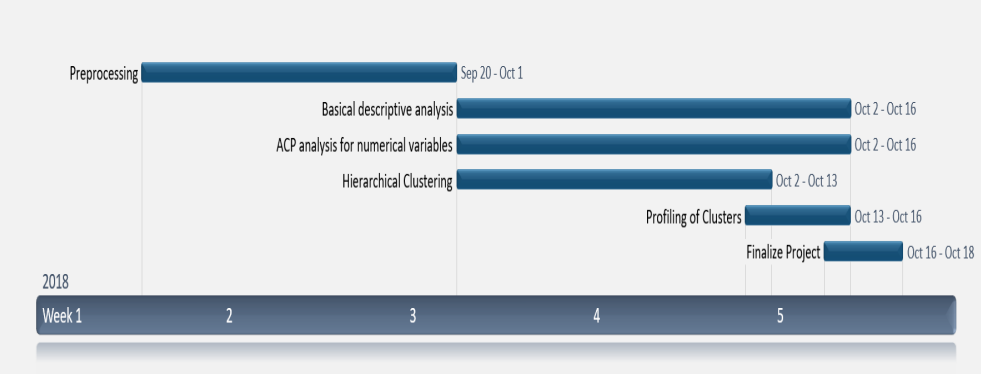

Figure 43: Initial Gantt chart

As shown in figure 43 we tried to finish the work seven days before the actual deadline on the 26th of October. This time was planned to buffer eventualities like persons leaving the course or other not planned things that will increase the time needed to finish. In the beginning the whole team works on the pre-processing, since it is needed for all following steps. After the pre-processing the tasks have been split between the different members of the group. Teams out of two people have been formed and assigned to one of the following tasks:

1. Basic descriptive analysis
2. PCA for numerical variables
3. Hierarchical clustering

The first task for the basic descriptive analysis was assigned to Roland Frieß and Enrique Gonzalez. The PCA of numerical variables was assigned to Jordi Armengol and Marc Catrisse. The hierarchical clustering was scheduled for Albert Figuera and Jacobo Moral. Since the hierarchical clustering is needed to profile the clusters, Albert and Jacobo are also doing the profiling of the clusters. After that the project will be finalized. The finalization includes a feedback-circle in which the groups give feedback to each other. So after getting the feedback from the other group each group implemented the feedback given to them. The finalization also contains including the plots in the final essay and writing it.

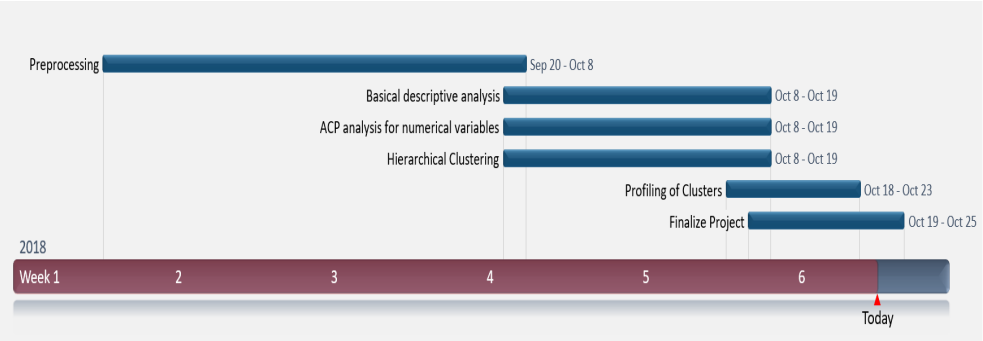

Figure 44: Final Gannt chart

In the final Gannt chart we changed the time for the profiling of the clustering, and added more time to develop our project. We did not think we would need that much time to finalize the project. But since the writing and documentation takes more time than expected, we added more time for the finalization of the project.

12.2 Final tasks assignment grid

Participant	Jordi	Marc	Albert	Roland	Enrique	Jacobo
Motivation of the work and general description	X					
Data Source presentation		X	X			X
Formal description of Data structure and metadata			X			X
Data Mining process performed		X			X	
Preprocessing	X	X		X		
Basic statistical descriptive analysis				X	X	
ACP analysis for numerical variables	X	X				
Clustering			X			X
Profiling			X			X
Conclusions	X					
Working plan				X		
Final Report	X	X	X	X	X	X

12.3 Risks and deviances in scheduling

In order to be prepared for different situations that can endanger the success of the project we developed a risk plan. In the risk plan we identified different risks and found ways to prevent the risk and manage the different scenarios.

Risk	How to prevent	How to manage
A team member leaves the group	Include everyone in decision making and always assign at least two members to a task	Distribute the workload of the leaving member
Data have weak- structure and models don't perform well	Ensure that technical assumptions of models are hold on data	Change to models without hypothesis that do not fit with data
A task turns out to be much more work than another so the assigned members aren't able to perform it in time	Having a good initial distribution of tasks	Encourage people to talk about the size of the task and react by assigning more members
Someone having a problem with solving a task	Making sub-teams of two members and do weekly meetings do discuss	Helping the member to catch up the work and assign other tasks to this person

Table 7: Project risk table

Our main self-critical point about the deviances of the final scheduling with respect to the originally designed one, the main problem is that we underestimated the amount of work that would suppose the pre-processing. It took longer than expected, and this problem affected the rest of the project, because many tasks depended on this one. Also, the pre-processing is not quite parallelizable. Two different sub-teams could be doing PCA and clustering, respectively, but it was not feasible to divide the pre-processing into different sub-teams.

Although this risk had been considered in the risk plan, it could not be prevented, because it hit as at the beginning, Fortunately it could me managed properly, by meeting two members of the team and doing it (so, adding more people to the task).

13 R Scripts

13.1 Pre-procesing

All R scripts are based on Karina Gibert's reference scripts for data mining.

```
#Install packages
install.packages("BaylorEdPsych")
install.packages("plyr")
install.packages("dplyr")

library(BaylorEdPsych)
install.packages("mvnmle")
library(mvnmle)
installed.packages("cluster")
library(cluster)
installed.packages("dplyr")
library("plyr")
library("dplyr")

# 1 - Building the original data matrix and introducing the data into
# the pre-processing tool
# NAs can be both 'NA' or empty, in this dataset
original_data <-read.csv("Speed.csv", header=TRUE,na.strings=c("","NA"))

# Checking the dataset. Visualization, basic descriptive statistics.

dim(original_data) # size
summary(original_data)
sum(is.na(original_data))/(sum(!is.na(original_data)) + sum(is.na(original_data)))*100
# "raw" type of each variable. Warning: only raw type, integers may codify categorical
# variables, we have had to manually inspect them
sapply(original_data, class)
# Number of NAs per columns
na_count <-sapply(original_data, function(y) sum(length(which(is.na(y)))))
na_count <- data.frame(na_count)
na_count


# Also: read the Kaggle description, read the PDF telling the meaning of each variable



# 2 - Determining the working data matrix

# Rows:we are going to select the non-variations and preference scale 1-100 rounds
```

```
# in order to have coherent and clean data

selected_rows_data <- original_data [!(original_data$wave==5
    | original_data$wave ==6 | original_data$wave ==7 |
    original_data$wave ==8 | original_data$wave ==9 |
    original_data$wave ==12 | original_data$wave ==13 |
    original_data$wave ==14 | original_data$wave ==18 |
    original_data$wave ==19 | original_data$wave ==20 | original_data$wave ==21),]


# Columns:
# We have many, many variables. Some of them are introducing noise,
# some of them are redundant or way to concrete, some of them are out of the scope
# of our intended analysis etc (explained in doc)
# "Expert" Variable selection (justification: redundancy, scope)
# We are kipping only some identification columns, but won't analyze them, obviously
# undergra, iid and pid only for missing imputation purposes

selected_vars<-c("iid","pid","undergra","gender","round","order","match",
    "int_corr","samerace","age_o","race_o","pf_o_att","pf_o_sin",
    "pf_o_int", "pf_o_fun", "pf_o_amb","pf_o_sha",
    "dec_o","attr_o","sinc_o","intel_o",
    "fun_o","amb_o",
    "shar_o","age","field_cd","mn_sat","tuition","race",
    "imprace","income","goal","date","go_out","dec","like","imprelig")
selected_columns_data  <- selected_rows_data[ selected_vars]

#declare qualitative variables


selected_columns_data$gender <- as.factor(selected_columns_data$gender)
selected_columns_data$match <- as.factor(selected_columns_data$match)
selected_columns_data$samerace <- as.factor(selected_columns_data$samerace)
selected_columns_data$field_cd <- as.factor(selected_columns_data$field_cd)
selected_columns_data$race <- as.factor(selected_columns_data$race)
selected_columns_data$race_o <- as.factor(selected_columns_data$race_o)
selected_columns_data$goal <- as.factor(selected_columns_data$goal)
selected_columns_data$date <- as.factor(selected_columns_data$date)
selected_columns_data$go_out <- as.factor(selected_columns_data$go_out)
selected_columns_data$dec <- as.factor(selected_columns_data$dec)

#change labels of categorical values
selected_columns_data$gender <-
    revalue(selected_columns_data$gender,c("0"="F","1"="M"))
selected_columns_data$match <-
    revalue(selected_columns_data$match,c("0"="N","1"="Y"))
selected_columns_data$samerace <-
```

```
    revalue(selected_columns_data$samerace,c("0"="N","1"="Y"))
selected_columns_data$field_cd <- revalue(selected_columns_data$field_cd,
    c("1"="Law","2"="Math",
        "3"="SocialSci", "4"="MedPharma",
        "5"="Eng","6"="Jour", "7"="HRP","8"="Econ",
        "9"="Edu","10"="Science", "11"="SocialW",
        "12" = "Undergrad", "13"="PolSci", "14"="Film",
        "15"="Art", "16"="Lan","17"="Arch", "18"="Other"))

selected_columns_data$race <- revalue(selected_columns_data$race,
    c("1"="Black", "2" = "Cauc","3"="Latin", "4"="Asian", "5"="Nat","6" = "Other"))

selected_columns_data$race_o <- revalue(selected_columns_data$race_o,
    c("1"="Black", "2" = "Cauc", "3"="Latin", "4"="Asian", "5"="Nat","6" = "Other"))

selected_columns_data$goal <- revalue(selected_columns_data$goal,
    c("1"="Fun", "2"="Meet", "3"="Date", "4"="Serious", "5"="Say","6"="Other"))

selected_columns_data$date <- revalue(selected_columns_data$date,
    c("1"="SevWeek", "2"="2Week","3"="1Week","4"="2Month","5"="1Month",
        "6"="SevYear","7"="Never"))

selected_columns_data$go_out <- revalue(selected_columns_data$go_out,
    c("1"="SevWeek", "2"="2Week","3"="1Week","4"="2Month","5"="1Month",
        "6"="SevYear","7"="Never"))

selected_columns_data$dec <- revalue(selected_columns_data$dec,c("0"="N","1"="Y"))


# We realize that tuition, mn_sat and income are detected as categorical by R, and
# simple declaring them as numeric doesn't work, so we have to manually remove commas

selected_columns_data$mn_sat<-as.numeric(gsub(",","",selected_columns_data$mn_sat))
selected_columns_data$tuition<-as.numeric(gsub(",","",selected_columns_data$tuition))
selected_columns_data$income <- as.numeric(gsub(",","",selected_columns_data$income))

# So, now he have the working matrix (selected_columns_data)


# 3 - outlier detection, visualization
# Plots of all variables, manual inspection
n_plot <- c("pid","iid","met")
for(i in 1:length(selected_columns_data[1,])) {
  name <- colnames(selected_columns_data)[i]
  if(!(name %in% n_plot)){
    if (is.factor(selected_columns_data[,i])) {
      plot(selected_columns_data[,i], main=name)
```

```
    }else {
      boxplot(selected_columns_data[,i], main=name)
    }
  }
}

# And: % of NA's for each selected column

na_count_selected <-sapply(selected_columns_data,
    function(y)100*sum(length(which(is.na(y))))/nrow(selected_columns_data))

na_count_selected <- data.frame(na_count_selected)
na_count_selected

# So, inspecting plots, NA's and taking into account the meaning of each variable,
# we try to detect possible outliers (id's ommited)

# Please notice that by using boxplots we are not implying that all variables follow
# a Gaussian distribution, it's for for detecting "potential" outliers
# but we won't necessarily label them as actual outliers

# gender: no outliers, no NAs, all values 0 or 1, OK.
# round: ok.
# order: ok.
# match: same as gender.
# int_corr: 1.91% missing , no outliers in boxplot

# samerace: same as gender.
# age_o: 0.53% missing, 3 values don't fit in the boxplot but are "real" ages
# race_o: 0.53% missing, it seems it has no outliers
# pf_o_att 0.96% missing.
# dec_o: same as gender

# attr_o 1.28% missing
# sinc_o 2.34% missing
# intel_o 2.55% missing
# fun_o 3.21 % missing
# amb_o 8.9% missing
# shar_o 13.48% missing
# No outliers, but, again, fractional values. We are going to take the floor but still
# keep them as categorical.
# age: 0.53% missing, no outliers (same as age_o)
# field_cd: 1.04% missing, no outliers apparently
# mn_sat: 62.5% missing, apparently 2 potential outliers but they are not;
# they are "real" SAT scores
#tuition: 58.0 %missing, no apparent outliers
#race 0.53% missing no outliers apparently
```

```
#imprace 0.96% missing,  There are 8 rows with value 0 it's supposed to be a value between
# 1-10, maybe 0 is equivalent to NA? (outliners in 0)
#income 48.72% missing, no apparent outliers
#goal 0.96% missing no outliners
#date 1.44% missing no outliners
#go_out 0.96% missing no outliners
#dec 0% missing binary with no outliners
#like 1.41% missing, outliners in some intermediate values(ex: 4.5, 5.5...)
# met 3.16% missing wrong data 1 is YES and the rest NO binary? Shall we delete this
# variable?
# imprelig, 0.96 %no outliners



# ERROR DETECTION AND TREATMENT




# imprace -> 0 is not a valid scale value (1-10), we will replace all occurrences
    (8)  by NA
selected_columns_data$imprace[selected_columns_data$imprace == 0] <- NA


# met
# We are going to delete this variable because the values are not coherent with
# the documentation




# MISSING IMPUTATION
data_pending_missing_imputation <- selected_columns_data

# tuition and mn_sat. structural -> 0
data_pending_missing_imputation$tuition[
    is.na(data_pending_missing_imputation$undergra)] <- 0
data_pending_missing_imputation$mn_sat[
    is.na(data_pending_missing_imputation$undergra)] <- 0
data_pending_missing_imputation$university <- data_pending_missing_imputation

data_pending_missing_imputation$university<-as.numeric(
    is.na(data_pending_missing_imputation$undergra))

data_pending_missing_imputation$university<-as.factor(
    data_pending_missing_imputation$university)
```

```
#change labels of categorical values
data_pending_missing_imputation$university<-revalue(
    data_pending_missing_imputation$university, c("0"="N","1"="Y"))

data_pending_missing_imputation<-data_pending_missing_imputation[
    data_pending_missing_imputation$iid != "28" &
    data_pending_missing_imputation$pid != "28" &
    data_pending_missing_imputation$iid != "58" &
    data_pending_missing_imputation$pid != "58" &
    data_pending_missing_imputation$iid != "59" &
    data_pending_missing_imputation$pid != "59", ]



# int_corr: 1.91% missing , no outliers in boxplot.in

rowswithmissingint_corr<-filter(.data = data_pending_missing_imputation,is.na(int_corr))
summary(rowswithmissingint_corr)

# After looking at all the rows with missing Data it can be seen,
# that there are three persons, that are responsible for the missing Values:
# Person 28 is involved in the first 32 rows with missing Values(the probablity is high,
# that this person didn't fill the Questionare right)
# Solving suggestion: Remove all rows for this person(filling the data would need # a lot of
# assumptions about the Person and bias our results)

# The similar problem accurs for Person 58 and 59.
# cutting all the rows where they accur might be the best solution (Discussion)
# These persons also didn't answer many other columns,
# so in my opinion its the best way if we cut them

# age_o: 0.53% missing, 3 values don't fit in the boxplot but are "real" ages
# All persons that didn't talk about their age are 58 and 59
# so if we cut them there won't be any missing values left

# race_o: 0.53% missing, it seems it has no outliers
# Same as age_o Person 58 and 59 are responsible for the missing values
# pf_o_att 0.96% missing.
# For all Values that miss in pf_o_att
# also by cutting Person 28,58,59 all missing Values will be cut out of the data set
# dec_o: same as gender

# attr_o 1.28% missing
# Again some missing Values are related to 58 and 59
# since we want to focus on other values now,
# the rows with pid or iid will be cut out before continuing with the preprocessing
```

```
data_pending_missing_imputation<-as.data.frame(filter(data_pending_missing_imputation,
    data_pending_missing_imputation$pid != 28 & data_pending_missing_imputation$iid != 28 &
    data_pending_missing_imputation$pid != 58 & data_pending_missing_imputation$iid != 58 &
    data_pending_missing_imputation$pid != 59 & data_pending_missing_imputation$iid != 59))


# After that, there are still two types of missing values for the following attributes.
# Rows with single missing Values and rows with all of them missing.
# I would suggest cutting out the rows with all the values missing,
# because imputation could lead to a big bias in our test.
# If there are no values the questionare was probably not answered.

data_pending_missing_imputation <-  as.data.frame(filter(data_pending_missing_imputation,
    !is.na(data_pending_missing_imputation$attr_o)|
    !is.na(data_pending_missing_imputation$sinc_o)|
    !is.na(data_pending_missing_imputation$intel_o)|
    !is.na(data_pending_missing_imputation$fun_o)|
    !is.na(data_pending_missing_imputation$amb_o)|
    !is.na(data_pending_missing_imputation$shar_o)))

# After that there are only 4 rows left with missing attr_o.
# For this rows this is the only missing value. The possibilities are now:

# 1.) Remove these rows (would not suggest because if we will lose the data)
# 2.) Impute values by expert opinion (could be a good shot,
# but we aren't experts and cannot access expert opinions right now)
# 3.) Impute values with the mean of the other 5 values
# 4.) Impute values with the mean of all attr_o values
# 5.) Impute values with the median of all attr_o values related to that person
# I think the most precise Imputation would be the median of the attr_o values
# Filter the data so only the rows with the specific iid are left:

iid10 <- filter(data_pending_missing_imputation,
    data_pending_missing_imputation$iid == 10)

# Median = 8
# set value for this row 8

data_pending_missing_imputation[96,"attr_o"] <- 8

iid22 <- filter(data_pending_missing_imputation,
    data_pending_missing_imputation$iid == 22)

# Median = 7
# set value for this row 7
```

```
data_pending_missing_imputation[224,"attr_o"] <- 7

iid37 <- filter(data_pending_missing_imputation,
    data_pending_missing_imputation$iid == 37)

# Median = 7
# set value for this row 7

data_pending_missing_imputation[447,"attr_o"] <- 7

iid104 <- filter(data_pending_missing_imputation,
    data_pending_missing_imputation$iid == 104)
# Median = 10
# set value for this row 10

data_pending_missing_imputation[1440,"attr_o"] <- 10

#LITTLE TO TEST TO IMPUTE NUMERICAL VALUES

selected_vars<-c("gender","round","order","match","int_corr","samerace","age_o","race_o",
    "pf_o_att","pf_o_sin", "pf_o_int", "pf_o_fun", "pf_o_amb","pf_o_sha",
    "dec_o","attr_o","sinc_o","intel_o","fun_o","amb_o","shar_o","age","field_cd",
    "mn_sat","tuition","race","imprace","income","goal","date","go_out","dec","like",
    "imprelig", "university")

data_pending_missing_imputation  <- data_pending_missing_imputation[selected_vars]

littleTest <- LittleMCAR(data_pending_missing_imputation)
littleTest$amount.missing
#The rows with the most number of NA have 17 NA
#Can we erase this rows with more than 15 NA?
littleTest$data$DataSet80




# Ultimate missing imputation


addUnknown <- function(x){
  if(is.factor(x) && sum(is.na(x) > 0 )) {
    y = factor(x, levels=c(levels(x), "Unknown"))
    y[is.na(y)] <- "Unknown"
    return(y)
  }
  return(x)
}
```

```
data_pending_missing_imputation <- as.data.frame(
    lapply(data_pending_missing_imputation, addUnknown))




vars_according_NAs <- as.data.frame(lapply(data_pending_missing_imputation,
    function(x) if (is.numeric(x)) return(sum(is.na(x))) else return(-1)))

colnames(vars_according_NAs)

fullVariables <- c()
uncompleteVars <- c()

for (var in colnames(vars_according_NAs)){
  if (vars_according_NAs[var] > 0) {
    uncompleteVars <- c(uncompleteVars,vars_according_NAs[var])
  }
  else if (vars_according_NAs[var] == 0) {
    fullVariables <- c(fullVariables,vars_according_NAs[var])
  }

}

fullVariables <- colnames(as.data.frame(fullVariables))

uncompleteVars <- sort(as.data.frame(uncompleteVars))

aux<-data_pending_missing_imputation[,fullVariables]
library(class)







for (k in colnames(as.data.frame(uncompleteVars))){
  aux1 <- aux[!is.na(data_pending_missing_imputation[,k]),]
  dim(aux1)
  aux2 <- aux[is.na(data_pending_missing_imputation[,k]),]
  dim(aux2)

  RefValues<- data_pending_missing_imputation[!is.na
```

```
    (data_pending_missing_imputation[,k]),k]
  #Find nns for aux2
  knn.values = knn(aux1,aux2,RefValues)

  #CARE: neither aux1 nor aux2 can contain NAs


  #CARE: knn.ing is generated as a factor.
  #Be sure to retrieve the correct values

  data_pending_missing_imputation[is.na(data_pending_missing_imputation[,k]),k] =
    as.numeric(as.character(knn.values))
  fullVariables<-c(fullVariables, k)
  aux<-data_pending_missing_imputation[,fullVariables]
}

data_after_imputation <- data_pending_missing_imputation

# NEW VARIABLES
# We are going to create the var "difference of age"
data_after_imputation$diff_age <- abs(data_after_imputation$age -
    data_after_imputation$age_o)


# Correct out of scale knn values
data_after_imputation$like[data_after_imputation$like < 1] <- 1


# 100 scale

data_after_imputation$pf_sum <- rowSums(data_after_imputation[,c("pf_o_att", "pf_o_sin",
    "pf_o_fun", "pf_o_int", "pf_o_amb", "pf_o_sha")])
data_after_imputation$at_o_sum <- rowSums(data_after_imputation[,c("attr_o", "sinc_o",
    "intel_o", "fun_o", "amb_o", "shar_o")])

# No rows with total = 0for pf_sum. Only one for at_o_sum. We are going to delete it.

data_after_imputation<-data_after_imputation[data_after_imputation$pf_sum != 0 &
    data_after_imputation$at_o_sum != 0,]

# Scale: they must add up to 100

data_after_imputation$pf_o_att <-
round(data_after_imputation$pf_o_att/data_after_imputation$pf_sum*100)
data_after_imputation$pf_o_sin <-
round(data_after_imputation$pf_o_sin/data_after_imputation$pf_sum*100)
data_after_imputation$pf_o_fun <-
```

```
round(data_after_imputation$pf_o_fun/data_after_imputation$pf_sum*100)
data_after_imputation$pf_o_int <-
round(data_after_imputation$pf_o_int/data_after_imputation$pf_sum*100)
data_after_imputation$pf_o_amb <-
round(data_after_imputation$pf_o_amb/data_after_imputation$pf_sum*100)
data_after_imputation$pf_o_sha <-
round(data_after_imputation$pf_o_sha/data_after_imputation$pf_sum*100)

data_after_imputation$attr_o <-
round(data_after_imputation$attr_o/data_after_imputation$at_o_sum*100)
data_after_imputation$sinc_o <-
round(data_after_imputation$sinc_o/data_after_imputation$at_o_sum*100)
data_after_imputation$intel_o <-
round(data_after_imputation$intel_o/data_after_imputation$at_o_sum*100)
data_after_imputation$fun_o <-
round(data_after_imputation$fun_o/data_after_imputation$at_o_sum*100)
data_after_imputation$amb_o <-
round(data_after_imputation$amb_o/data_after_imputation$at_o_sum*100)
data_after_imputation$shar_o <-
round(data_after_imputation$shar_o/data_after_imputation$at_o_sum*100)

data_after_imputation$pf_sum <- NULL
data_after_imputation$at_o_sum <- NULL

write.csv(data_after_imputation,file = "SpeedClean.csv",row.names =
    FALSE,col.names = TRUE)
```

13.2 Basic statistical and descriptive analysis

```
# Import the Data of our cleaned dataset without any missing values

speed_data <-read.csv("SpeedClean.csv",header = TRUE)
array_attributes <-

#Univariate Analysis

#gender:
#Counting the values of both available gender:
summary(speed_data$gender)
#   F    M
#1827 1829
```

```
#Match: Counting the "Matches" and "no Matches"
summary(speed_data$match)
# N    Y
# 3027  629
# Percentage of match = 17,2 % Y and 82,8 N


# Int_corr:
summary(speed_data$int_corr)
sd(speed_data$int_corr)
#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#-0.7000 -0.0200  0.2100  0.1948  0.4300  0.9000
# As expected for a correlation there are values between -1 and 1,
# the extreme points arent reached, which means, that there wasn't
# a complete match
# and not a complete missmatch between the participents.
# We can see that the matching seems to be higher than the
# missmatching (0,9 and -0,7).

# samerace:
summary(speed_data$samerace)
# N    Y
# 2085 1571
#age:
summary(speed_data$age)
# age_o:
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 21.00   24.00   26.00   26.09   28.00   39.00
# Standard Deviation:
sd(speed_data$age)
summary(speed_data$age_o)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 21.0    24.0    26.0    26.1    28.0    39.0
# As we can see the values for age and age_o are the same. They should be, by
# checking this we made sure that there wasn't a mistake in the data.
# race_o:
summary(speed_data$race_o)
# Asian Black  Cauc Latin Other
# 756   201  2199   289   211
# There are two big groups of races within the Speed_dating sessions:
# Asian and Cauc
barplot(summary(speed_data$race_o))
# For the following 12 variables we will calculate the basic values(Minimum, 1st
# Quantile, Median, Mean, 3rd Quantile, Maximum and standard Derivation )
# pf_o_amb:
summary(speed_data$pf_o_amb)
```

```
with(speed_data, hist(pf_o_amb, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$pf_o_amb)
# pf_o_att:
summary(speed_data$pf_o_att)
with(speed_data, hist(pf_o_att, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$pf_o_att)

# pf_o_sin:
summary(speed_data$pf_o_sin)
with(speed_data, hist(pf_o_sin, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$pf_o_sin)
# pf_o_int:
summary(speed_data$pf_o_int)
with(speed_data, hist(pf_o_int, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$pf_o_int)
# pf_o_fun:
summary(speed_data$pf_o_fun)
with(speed_data, hist(pf_o_fun, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$pf_o_fun)

# pf_o_sha:
summary(speed_data$pf_o_sha)
with(speed_data, hist(pf_o_sha, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$pf_o_sha)
# dec_o:
summary(speed_data$dec_o)

# standard Deviation:
sd(speed_data$dec_o)
#attr_o:
summary(speed_data$attr_o)
with(speed_data, hist(attr_o, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$attr_o)
```

```
#amb_o:
summary(speed_data$amb_o)
with(speed_data, hist(amb_o, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$amb_o)
# sinc_o:
summary(speed_data$sinc_o)
with(speed_data, hist(sinc_o, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$sinc_o)
#fun_o:
summary(speed_data$fun_o)
with(speed_data, hist(fun_o, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$fun_o)
#intel_o:
summary(speed_data$intel_o)
with(speed_data, hist(intel_o, scale="frequency", breaks="Sturges",
                                                  col="darkgray"))
# standard Deviation:
sd(speed_data$intel_o)

#shar_o:
summary(speed_data$shar_o)
with(speed_data, hist(shar_o, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$shar_o)
 # field_cd:
summary(speed_data$field_cd)
barplot(summary(speed_data$field_cd))
#mn_sat
summary(speed_data$mn_sat)
with(speed_data, hist(mn_sat, scale="frequency", breaks="Sturges",
                      col="darkgray"))
# standard Deviation:
sd(speed_data$mn_sat)

summary(speed_data$like)
with(speed_data, hist(like, scale="frequency", breaks="Sturges",
                      col="darkgray"))
sd(speed_data$like)
#Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#1.000   5.000   6.000   6.135   7.000  10.000
```

```
#This variable represent a mark from 1 to 10 of how much do you liked your partner,
#as we can see the mean and the median are similar and close to 6 so overall there
#are a lightly positive mark.

summary(speed_data$dec)
#  N    Y
#2097 1559
#The 46.5% of people decided to "match" the partner and the remaining 53.5%
#rejected his partner.

summary(speed_data$go_out)
#1Month   1Week  2Month   2Week   Never SevWeek SevYear
#98     745     212    1357      36    1157      51

barplot(summary(speed_data$go_out))
#As we can see on the plot most of the people go out every 2 or several weeks

summary(speed_data$date)

barplot(summary(speed_data$date))
#As we can see on the plot most of the people date every several years or every 2 months.

summary(speed_data$goal)
barplot(summary(speed_data$goal))
#As we can see on the plot the main goals are having fun and meeting new people.

summary(speed_data$income)
with(speed_data, hist(income, scale="frequency", breaks="Sturges",
                      col="darkgray"))
sd(speed_data$income)
#There are a big difference between the higher and the lower income, the standard
#deviation is pretty high (17k). We can relate this to people who have studies
#and people who don't.

summary(speed_data$imprelig)
with(speed_data, hist(imprelig, scale="frequency", breaks="Sturges",
                      col="darkgray"))
sd(speed_data$imprelig)
#In general people give little importance to religion.

summary(speed_data$imprace)
with(speed_data, hist(imprace, scale="frequency", breaks="Sturges",
                      col="darkgray"))
sd(speed_data$imprace)
#In general people give little importance to race.

summary(speed_data$race)
```

```
barplot(summary(speed_data$race))
#Most people have Caucastic race.

summary(speed_data$tuition)
with(speed_data, hist(tuition, scale="frequency", breaks="Sturges",
                      col="darkgray"))
sd(speed_data$tuition)
#There are a high standard deviation (12091) because there are a lot of people
#who haven't gone to college so they didn't paid.

# Bivariate Analysis
# In the following plot will be analized how the different Groups regarding the
# field of study influence the points given in the different categories
with(speed_data, tapply(amb_o, list(field_cd), mean, na.rm=TRUE))

barplot(with(speed_data, tapply(amb_o, list(field_cd), mean, na.rm=TRUE)))
barplot(with(speed_data, tapply(attr_o, list(field_cd), mean, na.rm=TRUE)))
barplot(with(speed_data, tapply(sinc_o, list(field_cd), mean, na.rm=TRUE)))
barplot(with(speed_data, tapply(intel_o, list(field_cd), sd, na.rm=TRUE)))
barplot(with(speed_data, tapply(fun_o, list(field_cd), sd, na.rm=TRUE)))
barplot(with(speed_data, tapply(shar_o, list(field_cd), sd, na.rm=TRUE)))
# Now for the expections that people have:
barplot(with(speed_data, tapply(pf_o_sin, list(field_cd), sd, na.rm=TRUE)))
barplot(with(speed_data, tapply(pf_o_fun, list(field_cd), sd, na.rm=TRUE)))
barplot(with(speed_data, tapply(pf_o_int, list(field_cd), sd, na.rm=TRUE)))
barplot(with(speed_data, tapply(pf_o_amb, list(field_cd), sd, na.rm=TRUE)))
barplot(with(speed_data, tapply(pf_o_sha, list(field_cd), sd, na.rm=TRUE)))
barplot(with(speed_data, tapply(pf_o_att, list(field_cd), sd, na.rm=TRUE)))
# Here can be seen, how much in a particular field they value a particular Attributte
barplot(with(speed_data, tapply(amb_o, list(field_cd), sd, na.rm=TRUE)))

# Decriptive analysis after preprocessing
class(speed_data)
dim(speed_data)
n<-dim(speed_data)[1]
n
K<-dim(speed_data)[2]
K

names(speed_data)

listOfColors<-c("blueviolet","darkviolet","mediumvioletred",
    "palevioletred","violet", "violetred", "violetred4")
listOfColors<-c("orange","blue","green", "white","yellow", "red", "violet")
listOfColors<-palette()
listOfColors<-rainbow(14)
```

```
par(ask=TRUE)

for(k in 1:K){
  if (is.factor(speed_data[,k])){
    frecs<-table(speed_data[,k], useNA="ifany")
    proportions<-frecs/n
    #ojo, decidir si calcular porcentages con o sin missing values
    pie(frecs, cex=0.6, main=paste("Pie of", names(speed_data)[k]))
    barplot(frecs, las=3, cex.names=0.7, main=paste("Barplot of",
        names(speed_data)[k]), col=listOfColors)
    print(frecs)
    print(proportions)
  }else{
    hist(speed_data[,k], main=paste("Histogram of", names(speed_data)[k]))
    boxplot(speed_data[,k], horizontal=TRUE, main=paste("Boxplot of",
        names(speed_data)[k]))
    print(summary(speed_data[,k]))
    print(paste("sd: ", sd(speed_data[,k])))
    print(paste("vc: ", sd(speed_data[,k])/mean(speed_data[,k])))
  }
}
```

13.3 PCA

```
dd <-read.csv("SpeedClean.csv", header=TRUE)

# attach(dd)
# names(dd)

nums <- unlist(lapply(dd, is.numeric))
dd_numeric <- dd[ , nums]
path <- "plots/9acp/"


#set a list of numerical variables

# PRINCIPAL COMPONENT ANALYSIS OF dd_numeric

pc1 <- prcomp(dd_numeric, scale=TRUE)
class(pc1)
attributes(pc1)

print(pc1)
```

```
# WHICH PERCENTAGE OF THE TOTAL INERTIA IS REPRESENTED IN SUBSPACES?

pc1$sdev
inerProj<- pc1$sdev^2
inerProj
totalIner<- sum(inerProj)
totalIner
pinerEix<- 100*inerProj/totalIner
pinerEix
barplot(pinerEix)

#Cummulated Inertia in subspaces, from first principal component to the
# 24th dimension subspace
png(paste(path,"a-cummulated-inertia-barplot.png",sep = ""))
barplot(100*cumsum(pc1$sdev[1:dim(dd_numeric)[2]]^2)/dim(dd_numeric)[2],
    main="Accumulated Inertia",xlab = "Principal components",
    ylab = "Accumulated inertia (%)",names.arg = 1:dim(dd_numeric)[2])
dev.off()
percInerAccum<-100*cumsum(pc1$sdev[1:dim(dd_numeric)[2]]^2)/dim(dd_numeric)[2]
percInerAccum

# 14th col = 79.921189 (80%)
# SELECTION OF THE SINGIFICNT DIMENSIONS (keep 80% of total inertia)
nd = 14

# STORAGE OF THE EIGENVALUES, EIGENVECTORS AND PROJECTIONS IN THE nd DIMENSIONS
Psi = pc1$x[,1:nd]

# STORAGE OF LABELS FOR INDIVIDUALS AND VARIABLES
iden = row.names(dd_numeric)
etiq = names(dd_numeric)
ze = rep(0,length(etiq)) # WE WILL NEED THIS VECTOR AFTERWARDS FOR THE GRAPHICS

for (axis_h in 1:(nd-1)) {
  for (axis_v in (axis_h+1):nd)  {
    axis_name <- paste("x-",axis_h,"_","y-",axis_v,sep = "")

    # PLOT OF INDIVIDUALS [APARTAT b 1]
    #select your axis
    eje1<-axis_h
    eje2<-axis_v

    png(paste(path,axis_name,"b-1-individuals.png",sep = ""))

    plot(Psi[,eje1],Psi[,eje2])
    text(Psi[,eje1],Psi[,eje2],labels=iden, cex=0.5)
    axis(side=1, pos= 0, labels = F, col="cyan")
```

```
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")
dev.off()

#library(rgl)
#png(paste(path,axis_name,"b-1-individulals-3d.png",sep = ""))
#plot3d(Psi[,1],Psi[,2],Psi[,3])
#dev.off()

#Projection of variables [APARTAT b 2]

Phi = cor(dd_numeric,Psi)

#select your axis

X<-Phi[,eje1]
Y<-Phi[,eje2]
png(paste(path,axis_name,"b-2-proj-all-nums.png",sep = ""))
plot(Psi[,eje1],Psi[,eje2],type="n",main="Projection of numeric variables",
    xlab=paste("Component",axis_h),ylab=paste("Component",axis_v))
axis(side=1, pos= 0, labels = F)
axis(side=3, pos= 0, labels = F)
axis(side=2, pos= 0, labels = F)
axis(side=4, pos= 0, labels = F)
arrows(ze, ze, X, Y, length = 0.07,col="blue")
text(X,Y,labels=etiq,col="darkblue", cex=0.7)
dev.off()


#zooms
png(paste(path,axis_name,"b-2-zooms-proj-all-nums.png",sep = ""))
plot(Psi[,eje1],Psi[,eje2],type="n",xlim=c(min(X,0),max(X,0)),ylim=c(min(Y,0),
    max(Y,0)),main="Zoomed projection of numeric variables",
    xlab=paste("Component",axis_h),ylab=paste("Component",axis_v))
axis(side=1, pos= 0, labels = F)
axis(side=3, pos= 0, labels = F)
axis(side=2, pos= 0, labels = F)
axis(side=4, pos= 0, labels = F)
arrows(ze, ze, X, Y, length = 0.07,col="blue")
text(X,Y,labels=etiq,col="darkblue", cex=1)

dev.off()



#Now we project both cdgs of levels of a selected qualitative variable without
```

```
#representing the individual anymore

png(paste(path,axis_name,"b-2-match.png",sep = ""))
plot(Psi[,eje1],Psi[,eje2],type="n",main="Levels of Match",xlab=paste("Component",
    axis_h),ylab=paste("Component",axis_v))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#select your qualitative variable: MATCH
varcat<-dd[,"match"]
fdic1 = tapply(Psi[,eje1],varcat,mean)
fdic2 = tapply(Psi[,eje2],varcat,mean)

#points(fdic1,fdic2,pch=16,col="blue", labels=levels(varcat))
text(fdic1,fdic2,labels=levels(varcat),col="blue", cex=0.7)
dev.off()

png(paste(path,axis_name,"b-2-dec.png",sep = ""))
plot(Psi[,eje1],Psi[,eje2],type="n",main="Levels of Dec",
    xlab=paste("Component",axis_h),ylab=paste("Component",axis_v))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#select your qualitative variable: DEC
varcat<-dd[,"dec"]
fdic1 = tapply(Psi[,eje1],varcat,mean)
fdic2 = tapply(Psi[,eje2],varcat,mean)

#points(fdic1,fdic2,pch=16,col="blue", labels=levels(varcat))
text(fdic1,fdic2,labels=levels(varcat),col="blue", cex=0.7)
dev.off()

png(paste(path,axis_name,"b-2-dec_o.png",sep = ""))
plot(Psi[,eje1],Psi[,eje2],type="n",main="Levels of Dec_o",
    xlab=paste("Component",axis_h),ylab=paste("Component",axis_v))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#select your qualitative variable: DEC_O
varcat<-dd[,"dec_o"]
fdic1 = tapply(Psi[,eje1],varcat,mean)
```

```
fdic2 = tapply(Psi[,eje2],varcat,mean)

#points(fdic1,fdic2,pch=16,col="blue", labels=levels(varcat))
text(fdic1,fdic2,labels=levels(varcat),col="blue", cex=0.7)
dev.off()


#all qualitative together
png(paste(path,axis_name,"b-2-all-qual-tog.png",sep = ""))
plot(Psi[,eje1],Psi[,eje2],type="n",main="All qualitative variables projected
    together",xlab=paste("Component",axis_h),ylab=paste("Component",axis_v))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")



#nominal qualitative variables

facts <- unlist(lapply(dd, is.factor))
dcat <- names(dd[ , facts])
dcat <- dcat[dcat != "goal" & dcat !="go_out" & dcat != "date"]
#dcat$goal <- NULL
#dcat$go_out <- NULL
#dcat$date <- NULL
#divide categoricals in several graphs if joint representation saturates

#build a palette with as much colors as qualitative variables

#colors<-c("blue","red","green","orange","darkgreen")
#install.packages("viridis")
#library(viridis)
#viridis_pal(option = "D")(length(dcat)) # n = number of colors seeked

#alternative
colors<-rainbow(length(dcat))

c<-1
for(k in dcat){
  seguentColor<-colors[c]
  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=0.6)
  c<-c+1
}
```

```
legend("bottomleft",dcat,pch=1,col=colors, cex=0.6)

dev.off()

#determine zoom level
#use the scale factor or not depending on the position of centroids
# ES UN FACTOR D'ESCALA PER DIBUIXAR LES FLETXES MES VISIBLES EN EL GRAFIC
#fm = round(max(abs(Psi[,1])))
#fm=40

#scale the projected variables
#X<-fm*U[,eje1]
#Y<-fm*U[,eje2]
#X<-fm*Psi[,eje1]
#Y<-fm*Psi[,eje2]

#X<-fm*Phi[,eje1]
#Y<-fm*Phi[,eje2]



png(paste(path,axis_name,"b-2-num-background.png",sep = ""))
#represent numerical variables in background
plot(Psi[,eje1],Psi[,eje2],type="n",xlim=c(-1,1), ylim=c(-3,1),main="All vars
    projected together, numeric vars in background",xlab=paste("Component",axis_h),
    ylab=paste("Component",axis_v))
#plot(X,Y,type="none",xlim=c(min(X,0),max(X,0)))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#add projections of numerical variables in background
arrows(ze, ze, X, Y, length = 0.07,col="lightgray")
text(X,Y,labels=etiq,col="gray", cex=0.7)

#add centroids
c<-1
for(k in dcat){
  seguentColor<-colors[c]

  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=0.6)
  c<-c+1
```

```
}
legend("bottomleft",dcat,pch=1,col=colors, cex=0.6)


#add ordinal qualitative variables. Ensure ordering is the correct

# dordi<-c(8)
# go_out = 31, date = 30 , goal = 29
dordi <- c(29,30,31)


levels(dd[,dordi[1]])
#reorder modalities(GOAL): when required
dd[,dordi[1]] <- factor(dd[,dordi[1]], ordered=TRUE,
    levels= c("Other","Fun","Say","Meet","Date","Serious"))
levels(dd[,dordi[1]])


#reorder modalities(DATE): when required
dd[,dordi[2]] <- factor(dd[,dordi[2]], ordered=TRUE,  levels=
    c("Unknown","Never", "SevYear","1Month",
    "2Month","1Week","2Week","SevWeek"))
levels(dd[,dordi[2]])


#reorder modalities(GO_OUT): when required
dd[,dordi[3]] <- factor(dd[,dordi[3]], ordered=TRUE,  levels= c("Never",
    "SevYear","1Month", "2Month","1Week","2Week","SevWeek"))
levels(dd[,dordi[3]])



c<-1
col <- c
for(k in dordi){
  #seguentColor<-colors[col]
  seguentColor<-colors[c]
  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  #connect modalities of qualitative variables
  lines(fdic1,fdic2,pch=16,col=seguentColor)
  text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=0.6)
  c<-c+1
  col<-col+1
}
```

```
legend("topleft",names(dd)[dordi],pch=1,col=colors[1:length(dordi)], cex=0.6)

dev.off()

# PROJECTION OF ILLUSTRATIVE qualitative variables on individuals' map
# PROJECCIÓ OF INDIVIDUALS DIFFERENTIATING THE Dictamen
# (we need a numeric Dictamen to color)
# MATCH
varcat=dd[,"match"]
png(paste(path,axis_name,"b-2-match-ill-proj.png",sep = ""))
plot(Psi[,1],Psi[,2],col=varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")
legend("bottomleft",levels(varcat),pch=1,col=c(1,2), cex=0.6)


# Overproject THE CDG OF  LEVELS OF varcat
fdic1 = tapply(Psi[,1],varcat,mean)
fdic2 = tapply(Psi[,2],varcat,mean)

text(fdic1,fdic2,labels=levels(varcat),col="cyan", cex=0.75)
dev.off()
# DEC
varcat=dd[,"dec"]
png(paste(path,axis_name,"b-2-dec-proj.png",sep = ""))
plot(Psi[,1],Psi[,2],col=varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")
legend("bottomleft",levels(varcat),pch=1,col=c(1,2), cex=0.6)


# Overproject THE CDG OF  LEVELS OF varcat
fdic1 = tapply(Psi[,1],varcat,mean)
fdic2 = tapply(Psi[,2],varcat,mean)

text(fdic1,fdic2,labels=levels(varcat),col="cyan", cex=0.75)
dev.off()

# DEC_O
varcat=dd[,"dec_o"]
png(paste(path,axis_name,"b-2-ill-dec-o.png",sep = ""))
plot(Psi[,1],Psi[,2],col=varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
```

```
    axis(side=3, pos= 0, labels = F, col="darkgray")
    axis(side=2, pos= 0, labels = F, col="darkgray")
    axis(side=4, pos= 0, labels = F, col="darkgray")
    legend("bottomleft",levels(varcat),pch=1,col=c(1,2), cex=0.6)


    # Overproject THE CDG OF  LEVELS OF varcat
    fdic1 = tapply(Psi[,1],varcat,mean)
    fdic2 = tapply(Psi[,2],varcat,mean)

    text(fdic1,fdic2,labels=levels(varcat),col="cyan", cex=0.75)
    dev.off()
  }
}
```

13.4 Hierarchical clustering

```
install.packages("ggplot2")
install.packages("cluster")
install.packages("dplyr")
install.packages("factoextra")
install.packages("VIM")

library(ggplot2)
library(cluster)
library(dplyr)
library(factoextra)
library(VIM)

#know your actual working directory
getwd();

#set your working directory
setwd("/Users/yago/Documents/Clase/DataMining-SpeedDating");

#retrieve data obtained from preprocessing
data <-read.csv("SpeedClean.csv", header=TRUE)

#Algorithm to create clusters dealing with mixed numeric and categorical variables
actives<-c(1:ncol(data))
dissimMatrix <- daisy(data[,c(2:ncol(data))], metric = "gower", stand=TRUE)
distMatrix<-dissimMatrix^2
cluster <- hclust(distMatrix,method="ward.D")
#versions noves "ward.D" i abans de plot: par(mar=rep(2,4))
# si se quejara de los margenes del plot
```

```
plot(cluster)

#this would be enough if we only had numerical variables
#cluster <- hclust(dist(as.matrix(data)),method="ward.D2")

#Calculate optimal number of clusters. Possible methods: "silhouette", "wss",
# "gap_stat". "gap_stat not working tho"
fviz_nbclust(data, hcut, method = "silhouette") +
geom_vline(xintercept = 3, linetype = 2)

#Assign the optimal number of clusters obtained from the functions above. It can
# also be any other number.
numClusters <- 3
clusterCut <- cutree(cluster, numClusters)


data$cluster <- clusterCut


clusterCutRect <- rect.hclust(cluster, numClusters, border="red")
clustersTable <- table(clusterCut, data$match)
clustersMatchTable <- table(clusterCut, data$match)

#get percentage of matches (yes/(yes+no)) for each cluster
getMatchChanceForEachCluster <- function(clustersMatchTable) {
  vector <- 1:nrow(clustersMatchTable)
  for (row in 1:nrow(clustersMatchTable)) {
    N <- clustersMatchTable[row, "N"];
    Y <- clustersMatchTable[row, "Y"];
    vector[row] <- (Y/(Y+N))*100
  }
  return (vector)
}
getMatchChanceForEachCluster(clustersMatchTable)

#count how many rows/elements/entries are there in each cluster.
cluster_append <- mutate(data, cluster = clusterCut)
count(cluster_append,cluster)
```

13.5 Profiling

```
getwd();
setwd("/Users/yago/Documents/Clase/DataMining-SpeedDating")
dd <- read.table("SpeedClean.csv",header=T, sep=";", dec='.');
```

```
names(dd)

attach(dd)


actives<-c(1:ncol(data))


#for numerical variables we'll just do the mean for each variable and cluster
numericalMeanOfEachCluster <- aggregate(data[, c(2,3,5,7,9,10,11,12,13,14,16,17,18,
    19,20,21,22,24,25,27,28,33,34,36)], list(data$cluster), mean)
numericalMeanOfEachCluster

active<-c(1,4,6,8,15,23,26,29,30,31,32,35)
Match    <- as.factor(data$match)

#createCPG(dd[,active], Tipo.trabajo)


plotConditionalTable<-function(data, res)
{
  if(ncol(data)==0)
  {
    cat("Number of columns of dataset is 0")
    return()
  }#endif
  if(nrow(data)==0)
  {
    cat("Number of rows of dataset is 0")
    return()
  }#endif
  #proceed only if data frame is non empty

  #transform response variable into a suitable string for printing purposes
  response<-factor(res)

  #create an auxiliary matrix with as much rows as classes to keep the position
  #of figures in the CPG
  nc<-length(levels(response))
  K<-dim(data)[2]
  ncells<-nc*K

  mat<- matrix(data=c(1:ncells),nrow= nc, ncol=K, byrow=FALSE)

  #ojo, que si esta buit el panell peta
  dev.off()
  layout(mat, widths= rep.int(1, K), heights= rep.int(1,nc))
```

```
  for (k in 1:K){
    Vnum<-data[,k]
    for(niv in levels(response)){
      print(niv)
      s<-subset(Vnum, response==niv)
      if(is.numeric(data[,k]))
      {  hist(s, main=paste(names(data)[k], niv))
        #evenctually add other summary statistics, like vc
      }else{
        barplot(table(s), las=3, cex.names=0.5,
            main=paste("Barplot of", names(data)[k]))
      }#endifelse
    }#end for niv
  }#end for k
}#end plot conditional table


#data do not contain the response variable

createCPG<- function(data, response)
{
  if (!is.factor(response))
  {
    cat("The variable ", names(response), " must be a factor" )
  }
  else
  {
    plotConditionalTable(data, response)
  }#end else
}#endcreateCPG



#Fer gran la finestra del R
createCPG(data[,active], as.factor(data$match))

#attach(data)

#fer creixer la finestra de plots
#control - per fer menor el tipus de lletra en R
createCPG(data[,active], as.factor(clusterCut))


levels(Match) <- c("si","no")

#Calcula els valor test de la variable Xnum per totes les modalitats del factor P
```

```
ValorTestXnum <- function(Xnum,P){
  #freq dis of fac
  nk <- as.vector(table(P));
  n <- sum(nk);
  #mitjanes x grups
  xk <- tapply(Xnum,P,mean);
  #valors test
  txk <- (xk-mean(Xnum))/(sd(Xnum)*sqrt((n-nk)/(n*nk)));
  #p-values
  pxk <- pt(txk,n-1,lower.tail=F);
  for(c in 1:length(levels(as.factor(P)))){if (pxk[c]>0.5){pxk[c]<-1-pxk[c]}}
  return (pxk)
}


ValorTestXquali <- function(P,Xquali){
  taula <- table(P,Xquali);
  n <- sum(taula);
  pk <- apply(taula,1,sum)/n;
  pj <- apply(taula,2,sum)/n;
  pf <- taula/(n*pk);
  pjm <- matrix(data=pj,nrow=dim(pf)[1],ncol=dim(pf)[2]);
  dpf <- pf - pjm;
  dvt <- sqrt(((1-pk)/(n*pk))%*%t(pj*(1-pj)));
  zkj <- dpf/dvt;
  pzkj <- pnorm(zkj,lower.tail=F);
  for(c in 1:length(levels(as.factor(P)))){for (s in 1:length(levels(Xquali)))
    {if (pzkj[c,s]> 0.5){pzkj[c,s]<-1- pzkj[c,s]}}}
  return (list(rowpf=pf,vtest=zkj,pval=pzkj))
}


dades<-data
#dades<-df
K<-dim(dades)[2]
par(ask=TRUE)

P<-clusterCut #antigament c2
nc<-length(levels(as.factor(P)))
pvalk <- matrix(data=0,nrow=nc,ncol=K, dimnames=list(levels(P),names(dades)))
nameP<-"Class"
n<-dim(dades)[1]

for(k in 1:K){
  if (is.numeric(dades[,k])){
    print(paste("Anàlisi per classes de la Variable:", names(dades)[k]))
```

```
  boxplot(dades[,k]~P, main=paste("Boxplot of", names(dades)[k], "vs", nameP ),
      horizontal=TRUE)

  barplot(tapply(dades[[k]], P, mean),main=paste("Means of", names(dades)[k],
      "by", nameP ))
  abline(h=mean(dades[[k]]))
  legend(0,mean(dades[[k]]),"global mean",bty="n")
  print("Estadístics per groups:")
  for(s in levels(as.factor(P))) {print(summary(dades[P==s,k]))}
  o<-oneway.test(dades[,k]~P)
  print(paste("p-valueANOVA:", o$p.value))
  kw<-kruskal.test(dades[,k]~P)
  print(paste("p-value Kruskal-Wallis:", kw$p.value))
  pvalk[,k]<-ValorTestXnum(dades[,k], P)
  print("p-values ValorsTest: ")
  print(pvalk[,k])
}else{
  #qualitatives
  print(paste("Variable", names(dades)[k]))
  table<-table(P,dades[,k])
  #   print("Cross-table")
  #   print(table)
  rowperc<-prop.table(table,1)

  colperc<-prop.table(table,2)
  #  print("Distribucions condicionades a files")
  # print(rowperc)

  marg <- table(as.factor(P))/n
  print(append("Categories=",levels(dades[,k])))
  plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
  paleta<-rainbow(length(levels(dades[,k])))
  for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }

  #with legend
  plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
  paleta<-rainbow(length(levels(dades[,k])))
  for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }
  legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)

  #condicionades a classes
  print(append("Categories=",levels(dades[,k])))
  plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
  paleta<-rainbow(length(levels(dades[,k])))
  for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }

  #with legend
```

```
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(dades[,k])))
for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }
legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)

#amb variable en eix d'abcisses
marg <-table(dades[,k])/n
print(append("Categories=",levels(dades[,k])))
plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(as.factor(P))))
for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c]) }

#with legend
plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c])}
legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)

#condicionades a columna
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(as.factor(P))))
for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c]) }

#with legend
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c])}
legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)

table<-table(dades[,k],P)
print("Cross Table:")
print(table)
print("Distribucions condicionades a columnes:")
print(colperc)

#diagrames de barres apilades

paleta<-rainbow(length(levels(dades[,k])))
barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )

barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )
legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)

#diagrames de barres adosades
barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta )

barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta)
legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)
```

```
    print("Test Chi quadrat: ")
    print(chisq.test(dades[,k], as.factor(P)))

    print("valorsTest:")
    print( ValorTestXquali(P,dades[,k]))
  }
}#endfor



for (c in 1:length(levels(as.factor(P)))) {
    if(!is.na(levels(as.factor(P))[c])){print(paste("P.values per
        class:",levels(as.factor(P))[c])); print(sort(pvalk[c,]), digits=3) }}
```