MVA PROJECT:

16/01/2024

Project by: Andreja Andrejic, David Candela, Nayara Costa, Julie Oppedal, Alexandros Tremopoulos

Index

1. INTRODUCTION

DATA
(Motivation of the work and general description of the problem to be analyzed)

The Bank Marketing dataset compiles call information from a Portuguese banking institution with the objective of predicting whether a client will subscribe to a term deposit. This classification task relies on various categorical features, such as the client's education level, occupation, and marital status. Additionally, numerical features like the client's age and average account balance are considered in the analysis.

METADATA

How to download: The files can be downloaded directly through this link or going through the Bank Marketing link and pressing the 'Download' button. This will download a zip folder called 'bank+marketing'. When the contents are extracted, there is a zip folder called 'bank'. Inside this folder, there is the file 'bank.csv' that we are using. 'bank.csv' contains 10% of the examples and 17 inputs, randomly selected from 3 (older version of this dataset with less inputs).

Description: The Bank Marketing dataset compiles call information from a Portuguese banking institution with the objective of predicting whether a client will subscribe to a term deposit. This classification task relies on various categorical features, such as the client's education level, occupation, and marital status. Additionally, numerical features like the client's age and average account balance are considered in the analysis. For more information visit the Bank Marketing web pag

WORKING PLAN

Task Division

Task	Andreja	David	Nayara	Julie	Alexandros
Report	*X*	*X*	*X*	*X*	*X*
Data cleaning				*X*	**X**
Descriptive analysis			**X**	*X*	
PCA	*X*	**X**	*X*		
ACM and MFA	*X*	*X*		**X**	
Associations rules			*X*		**X**
Clustering	**X**				*X*
Profiling		*X*		**X**	
Discriminant analysis	**X**		*X*		
Decision trees		**X**			*X*
Presentation preparation	*X*	*X*	**X**	*X*	*X*

Table 1: Task division

Risk Management

A team member leaves the group.

How to prevent: All the tasks need to have more than one member assigned to them.

How to manage: Re-assign the tasks to compensate for workload.

- A team member can't keep up with his/her task.

How to prevent: More than one member in a task.

How to manage: Extra members get involved and re-assign.

- Previous task not completed fully.

How to prevent: After each task, communicate and discuss the outcomes

How to manage: Update the previous task and check with the group

- Inconsistent data through tasks

How to prevent: Common database

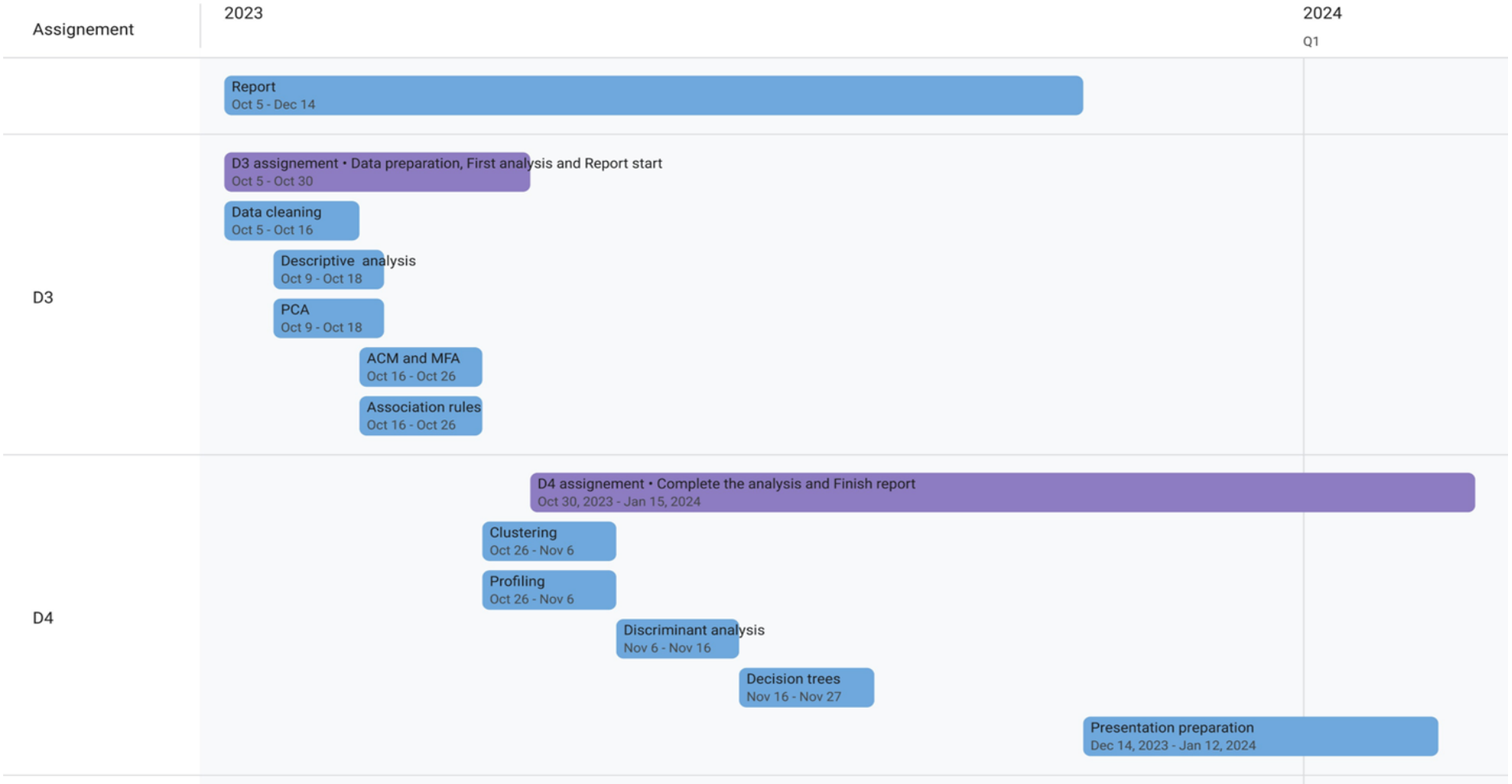
How to manage: Server or drive backups

- • A team member has problems with oral presentations

How to prevent: Ensure communication within the group to detect the team members that will need assistance

How to manage: Divide the oral presentation accordingly

Gantt Diagram


Assignement
2023
2024
Q1
Report
Oct 5 - Dec 14
D3
D3 assignement • Data preparation, First analysis and Report start
Oct 5 - Oct 30
Data cleaning
Oct 5 - Oct 16
Descriptive analysis
Oct 9 - Oct 18
PCA
Oct 9 - Oct 18
ACM and MFA
Oct 16 - Oct 26
Association rules
Oct 16 - Oct 26
D4
D4 assignement • Complete the analysis and Finish report
Oct 30, 2023 - Jan 15, 2024
Clustering
Oct 26 - Nov 6
Profiling
Oct 26 - Nov 6
Discriminant analysis
Nov 6 - Nov 16
Decision trees
Nov 16 - Nov 27
Presentation preparation
Dec 14, 2023 - Jan 12, 2024

2. PREPROCESSING

General pre-processing:

Ten out of seventeen variables contain characters, so they are converted to factors to be easily handled. Four of the variables are binary with ‘yes’ or ‘no’ as options. They are also converted to factors.

MISSING VALUES

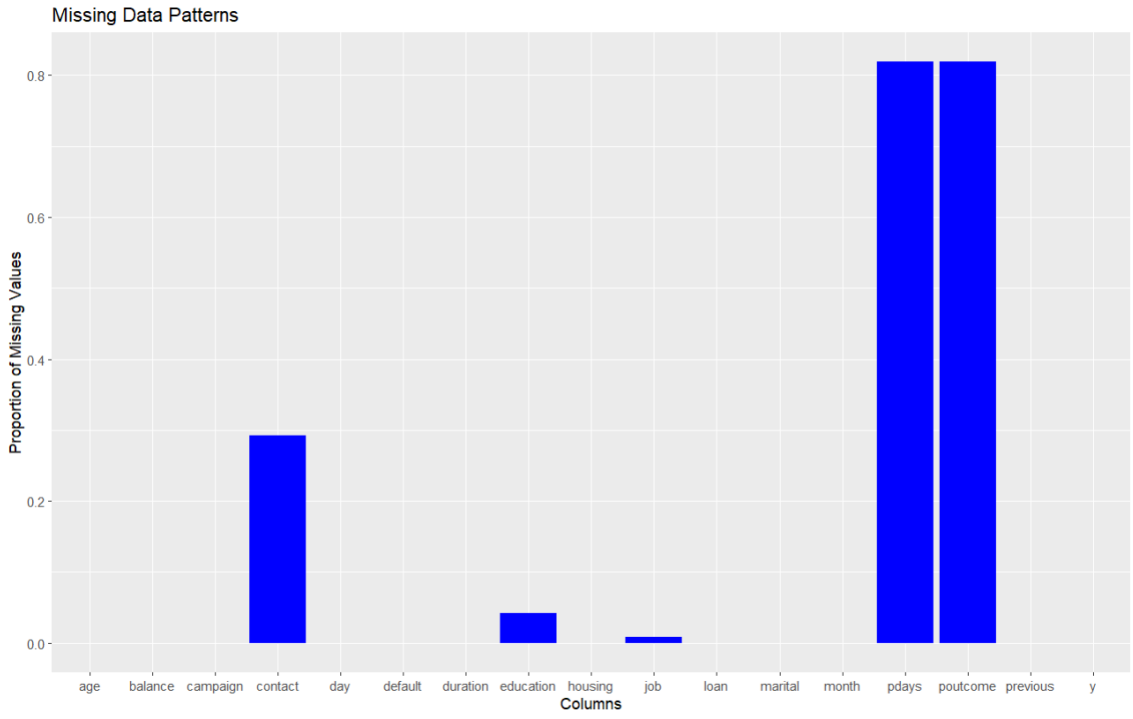
There are five features with missing values. Four of them are categorical and one is numeric. Note that in the dataset missing values were replaced with ‘unknown’ or -1 in categorical and numeric variables respectively. So, in order to explore missing values replacement takes place again and ‘unknown’ and -1 values are converted to NAs.

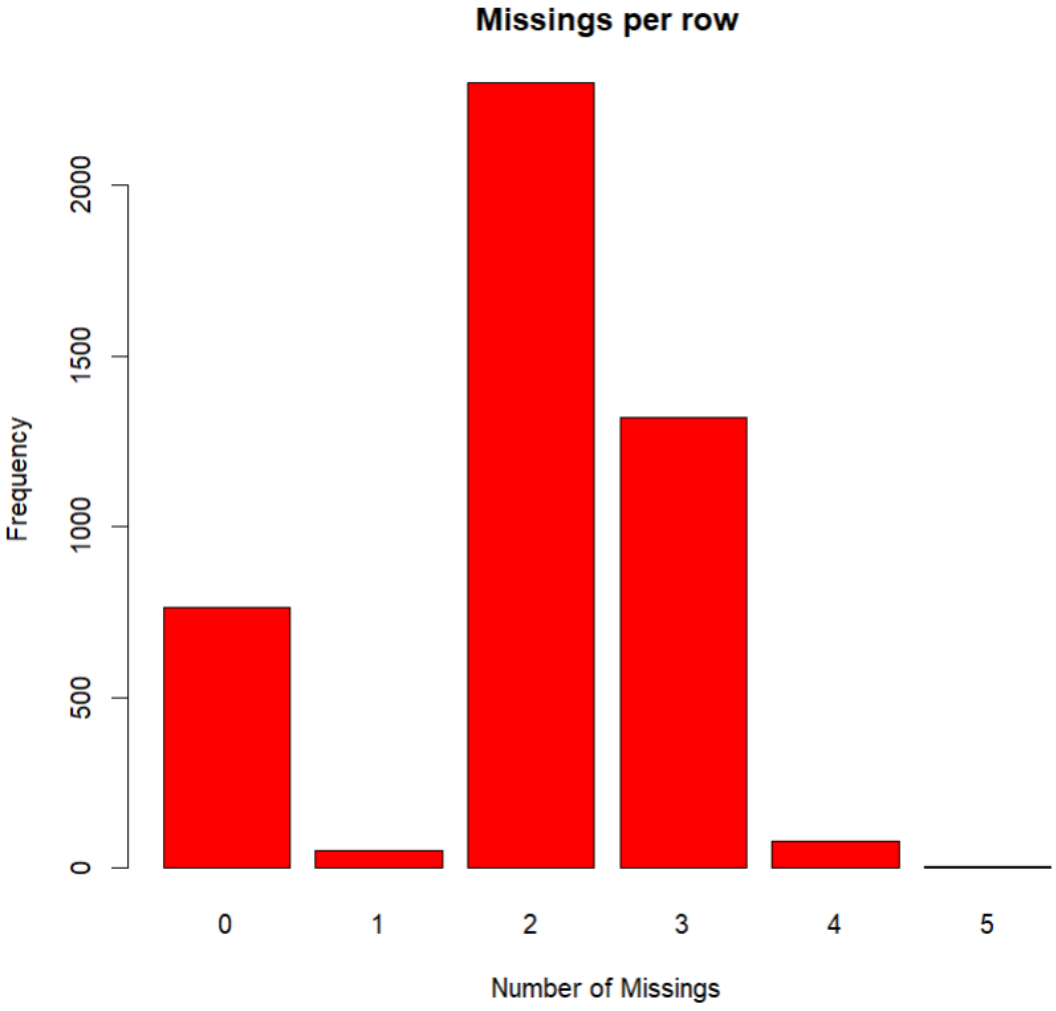
The percentage of the missing values in all cells is ~11.65%. Below in the figure the amount and the percentages of variables which miss values are displayed. Here the numeric variable is ‘pdays’ and the other four are categorical.

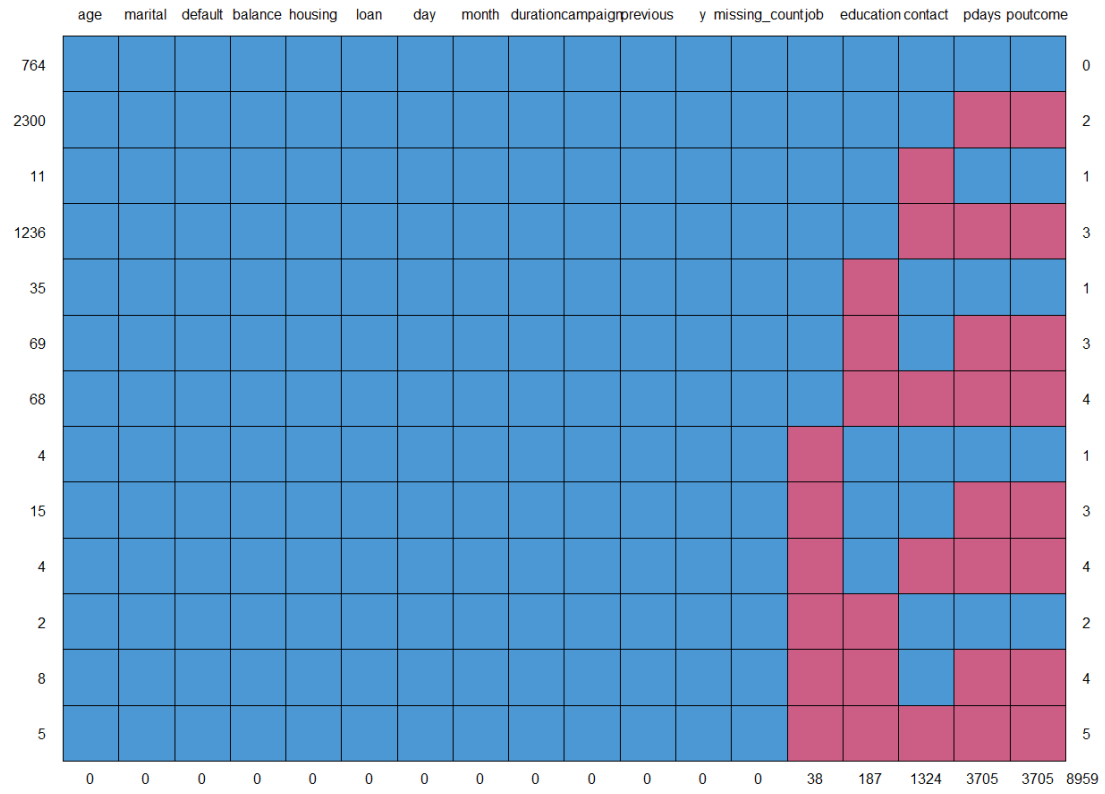
Variable <chr>	Missing_Count <dbl>	Percentage <dbl>
pdays	3705	81.950896
poutcome	3705	81.950896
contact	1324	29.285556
education	187	4.136253
job	38	0.840522

The next step is to check if the values are missing completely at random (MCAR) or not. Thus, little’s mcar test is performed. However this can be performed only in numeric variables and there is only one such which misses values. So, mcar test is only performed to the numeric features (11 in total) and the result is a p-value equal to 0, which indicates that missing values of ‘pdays’ variable are not completely at random and should be explored more in depth.

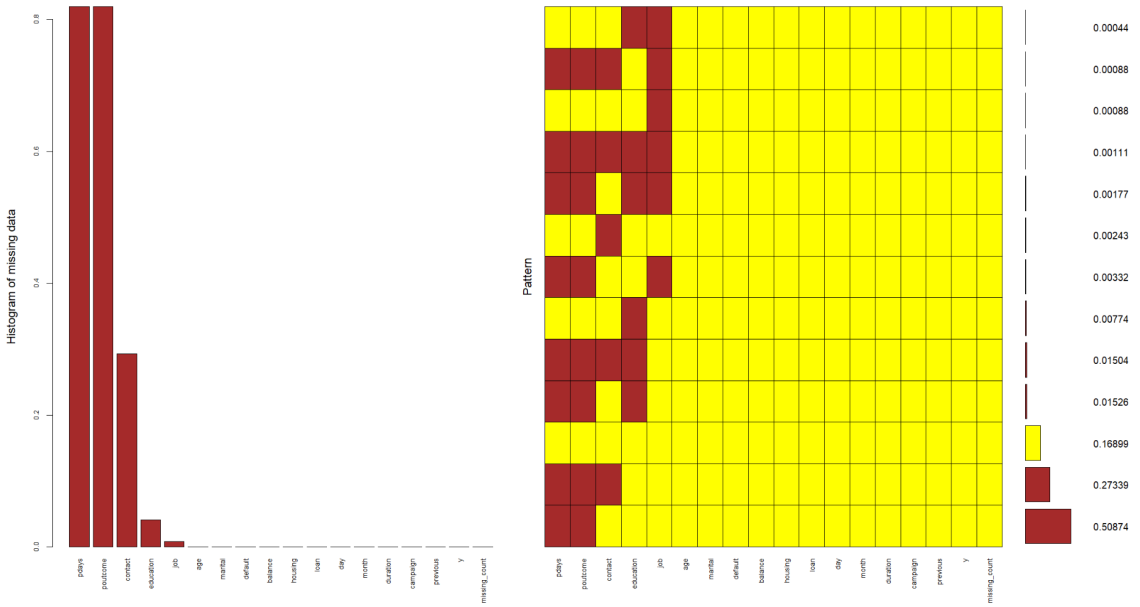
There are some plots that could help explore patterns in the missing values. Initially there is a plot with the frequency of missing values in each feature. Then in the next figure the frequency of missing values in rows can be observed. There are more than 2000 rows with at least two missing values.


Missing Data Patterns
Proportion of Missing Values
0.0
0.2
0.4
0.6
0.8
age
balance
campaign
contact
day
default
duration
education
housing
job
loan
marital
month
pdays
poutcome
previous
y
Columns


Missings per row
Frequency
0
500
1000
1500
2000
0
1
2
3
4
5
Number of Missings



In the figure just above there is the plot of the ‘md.pattern() function. There is a header (first row) that contains all the variable names, and the rows represent missing data. The left column indicates the number of times each pattern exists. The right column represents the number of missing values in each pattern and the bottom row shows the number of missing values for each feature. The next plot shows again the percentages of missing values in each feature and the pattern table with the difference that instead of a number that indicates how many times a pattern exists; it shows the percentage of each pattern.



So, from these two plots we come to some conclusions. First of all ‘pdays’ and ‘poutcome’ go together i.e. they only miss values in the exact same rows. A reminder that ‘pdays’ are the time since the client was last contacted from a previous campaign while ‘poutcome’ is the outcome of the previous marketing campaign. From that, we conclude that these values are not missing out of an intention, so they are not MNAR (Missing Not At Random).

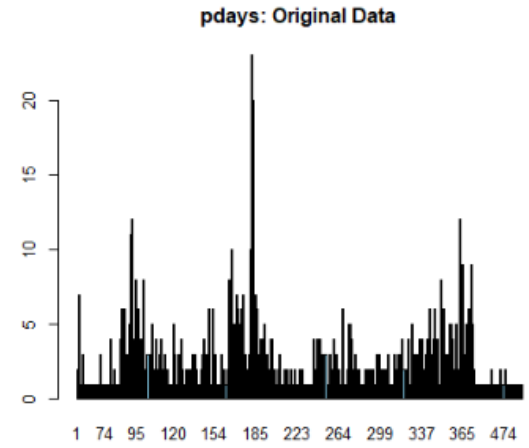
It is also observed that ‘contact’ variable is missing 99,17% of the time (1313 out of 1324 missing values) along with ‘pdays’ and ‘poutcome’. Eleven times it is the only value missing. Contact is the communication method with the client, and it makes sense when days from the previous campaign, outcome of the campaign and the contact are missing altogether. We conclude that those three features have a pattern of missing values in the same rows (same clients). Thus, we consider ‘contact’ not being MNAR as well. The ‘education’ variable has ~80.2% of values of the time while ‘pdays’ and ‘poutcome’ miss as well and 60.07% when these two along with ‘contact’ miss values.

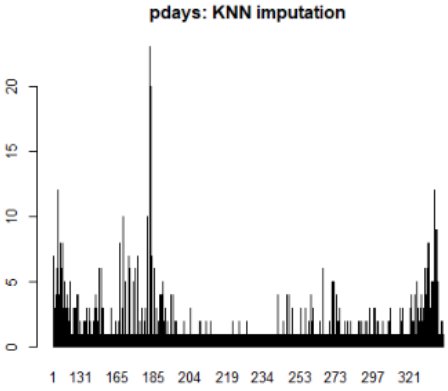
In addition, 84,2% of the times ‘job’ values are NAs, ‘pdays’ and ‘poutcome’ are NAs as well. Although there may seen patterns of ‘education’ and ‘job’ missing along with ‘pdays’ and ‘poutcome’, we will consider them as MNAR, because firstly ‘pdays’ and ‘poutcome’ are missing almost 82% of the time, which is a lot and may not be related, but most importantly due to chances that someone did not give his job and education status to the bank out of intention and personal reason. Consequently, we replace NAs in ‘job’ and ‘education’ again with ‘unknown’.

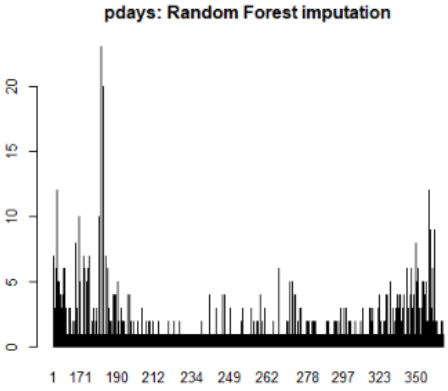
Imputation of missing values

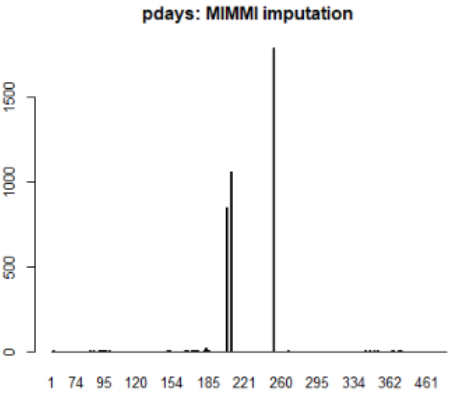
Four different imputation methods were used. Those are KNN[k=5, k=67 (67 is the length of dataframe squared], MIMMI(k=25, k=5) , Random Forest and MICE. KNN is only tested on ‘pdays’, which is the only numerical feature that misses values and KNN is supposed to only handle numeric variables. The other 3 methods were used, for all 3 variables (‘pdays’, ‘poutcome’ and ‘contact’). After inputting them with all the methods, we have to decide which one performs/imputes better. Thus, there are some plots and some absolute numbers comparison, for before and after imputation.

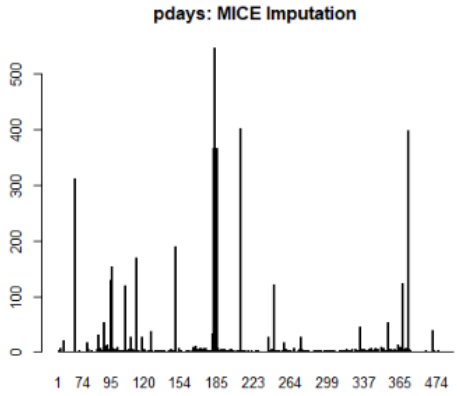
- Initially for ‘pdays’ the summary for each method are compared to the original data. We deduce that MICE is the better solution, because quartiles and mean are the closest to the original data among the 4 methods.:


pdays: Original Data
0
5
10
15
20
1 74 95 120 154 185 223 264 299 337 365 474


pdays: KNN imputation
0
5
10
15
20
1 131 165 185 204 219 234 253 273 297 321

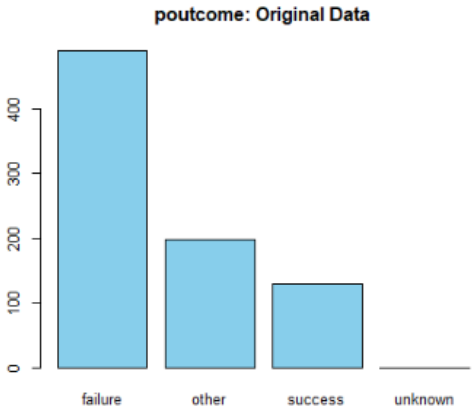

pdays: Random Forest imputation
0
5
10
15
20
1 171 190 212 234 249 262 278 297 323 350

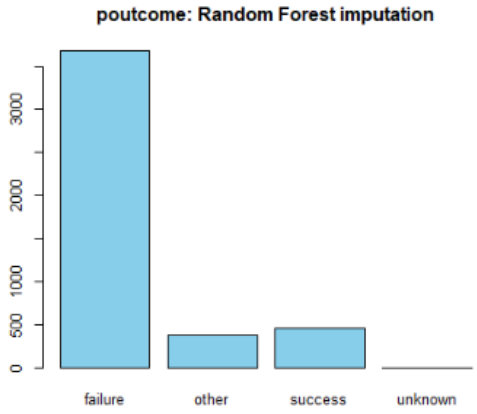

pdays: MIMMI imputation
0
500
1000
1500
1 74 95 120 154 185 221 260 295 334 362 461

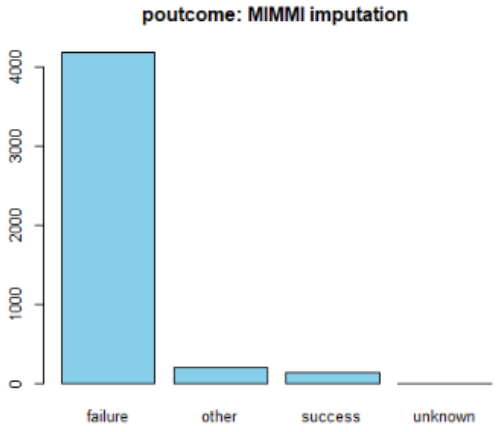

pdays: MICE Imputation
0
100
200
300
400
500
1 74 95 120 154 185 223 264 299 337 365 474

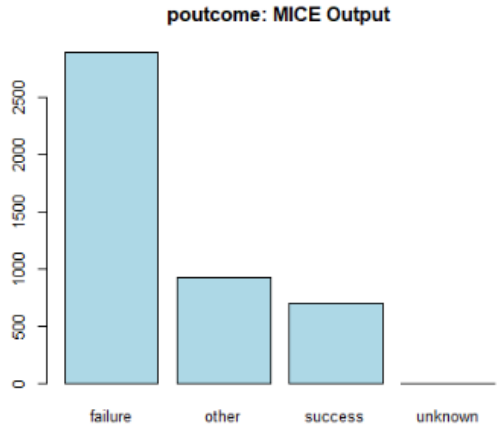
```
> print("pdays")
[1] "pdays"
> print("Original:")
[1] "Original:"
> summary(df$pdays)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
    1.0   136.0   189.0   224.9   330.0   871.0    3705
> print("KNN:")
[1] "KNN:"
> summary(df_imp_knn$pdays)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0   178.1   225.4   229.7   280.8   871.0
> print("Random Forest:")
[1] "Random Forest:"
> summary(df_imp_rf$pdays)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0   199.0   252.9   251.5   299.0   871.0
> print("MIMMI:")
[1] "MIMMI:"
> summary(df_imp_mimmi$imputedData$pdays)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0   198.2   202.9   226.2   253.9   871.0
> print("MICE:")
[1] "MICE:"
> summary(mice.output$pdays)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0   114.0   183.0   198.6   247.0   871.0
```

- Then for ‘poutcome’ we observe from plots that MICE barplots patterns are the nearest to the original data. That can be verified with the absolute numbers and the percentages of each distinct value of this feature. MICE imputed results have by far the closest percentages of values with the original data.



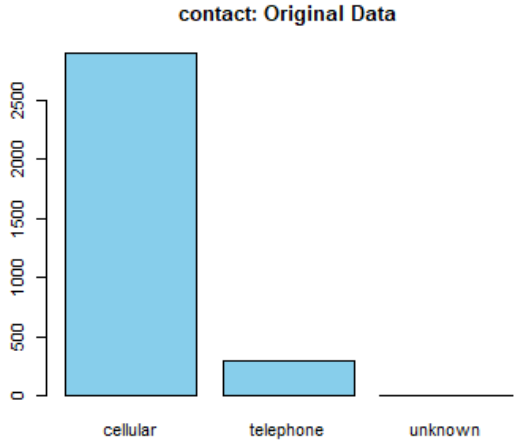


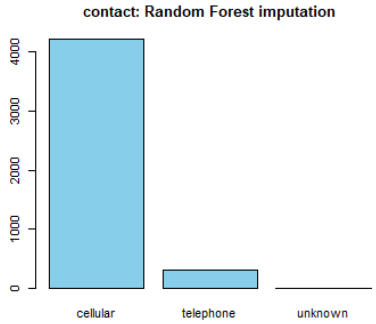


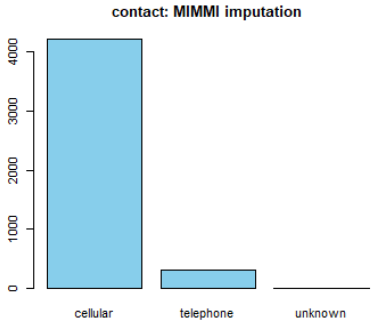


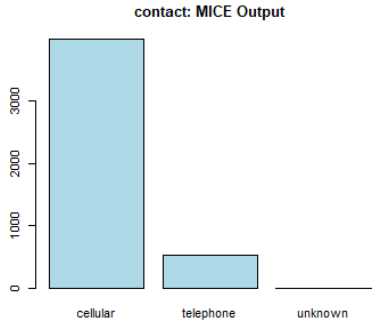

```
> print("poutcome")
[1] "poutcome"
> print("Original:")
[1] "Original:"
> table(df$poutcome)

failure   other success unknown
    490     197     129       0
> prop.table(table(df$poutcome)) * 100

 failure    other  success  unknown
60.04902 24.14216 15.80882  0.00000
> print("Random Forest:")
[1] "Random Forest:"
> table(df_imp_rf$poutcome)

failure   other success unknown
   3687     381     453       0
> prop.table(table(df_imp_rf$poutcome)) * 100

  failure     other   success   unknown
81.552754  8.427339 10.019907  0.000000
> print("MIMMI:")
[1] "MIMMI:"
> table(df_imp_mimmi$imputedData$poutcome)

failure   other success unknown
   4195     197     129       0
> prop.table(table(df_imp_mimmi$imputedData$poutcome)) * 100

  failure     other   success   unknown
92.789206  4.357443  2.853351  0.000000
> print("MICE:")
[1] "MICE:"
> table(mice.output$poutcome)

failure   other success unknown
   2896     924     701       0
> prop.table(table(mice.output$poutcome)) * 100

 failure    other  success  unknown
64.05662 20.43796 15.50542  0.00000
>
```

- Lastly, for ‘contact’ variable the case is similar with ‘poutcome’ in the barplots, but as well in the absolute numbers and the percentages. Here the percentages of MICE output are not by far closer to the original data, but they are clearly closer.









```
> print("Contact")
[1] "Contact"
> print("Original:")
[1] "Original:"
> table(df$contact)

 cellular telephone   unknown
      2896       301         0
> prop.table(table(df$contact)) * 100

 cellular telephone   unknown
90.584923  9.415077  0.000000
> print("Random Forest:")
[1] "Random Forest:"
> table(df_imp_rf$contact)

 cellular telephone   unknown
      4218       303         0
> prop.table(table(df_imp_rf$contact)) * 100

 cellular telephone   unknown
93.297943  6.702057  0.000000
> print("MIMMI:")
[1] "MIMMI:"
> table(df_imp_mimmi$imputedData$contact)

 cellular telephone   unknown
      4220       301         0
> prop.table(table(df_imp_mimmi$imputedData$contact)) * 100

 cellular telephone   unknown
93.342181  6.657819  0.000000
> print("MICE:")
[1] "MICE:"
> table(mice.output$contact)

 cellular telephone   unknown
      3996       525         0
> prop.table(table(mice.output$contact)) * 100

 cellular telephone   unknown
 88.38752  11.61248   0.00000
```

To conclude the imputation of missing values, it seems that MICE performs better than the other three methods, for each of the three variables we wanted to impute. Thus, these are the results we are going to keep in our analysis.

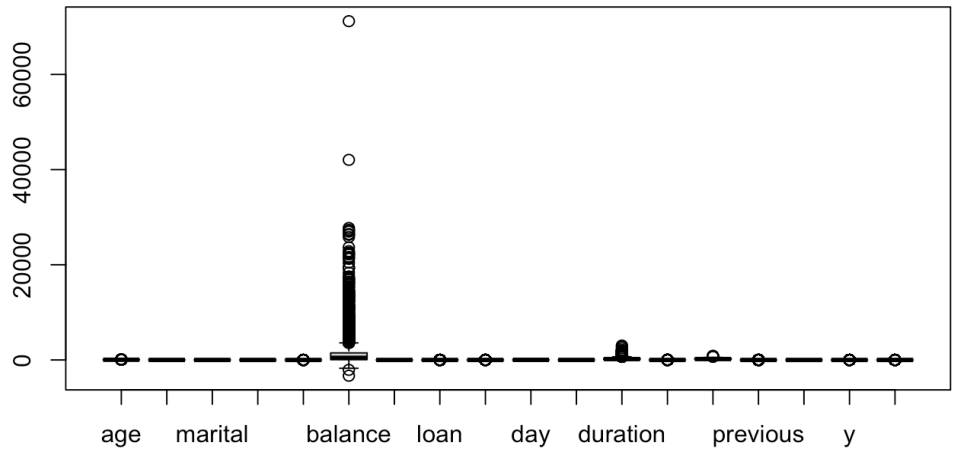
During working on the deliverable and after this imputation took place, we observed something very important. There is one variable called ‘previous’ that indicates the number of times the client was contacted before the campaign. We tested this variable and saw that when it has value 0, ‘pdays’ has value -1 and ‘poutcome’ has value ‘unknown’. That makes perfect sense because if the bank has never contacted the client before, there are no passed days since last spoke with the client and not a previous outcome of the campaign. Thus, the dataset has -1 and ‘unknown’ respectively to indicate this. So, ‘pdays’ and ‘poutcome’ should not be imputed. A reminder that ‘job’ and ‘education’ were not imputed as well because they might be MNAR. That leaves us with only the ‘contact’ variable to be imputed. It is going to be imputed with mice and the results are shown below in the figure. In addition, the ‘pdays’ column is going to be

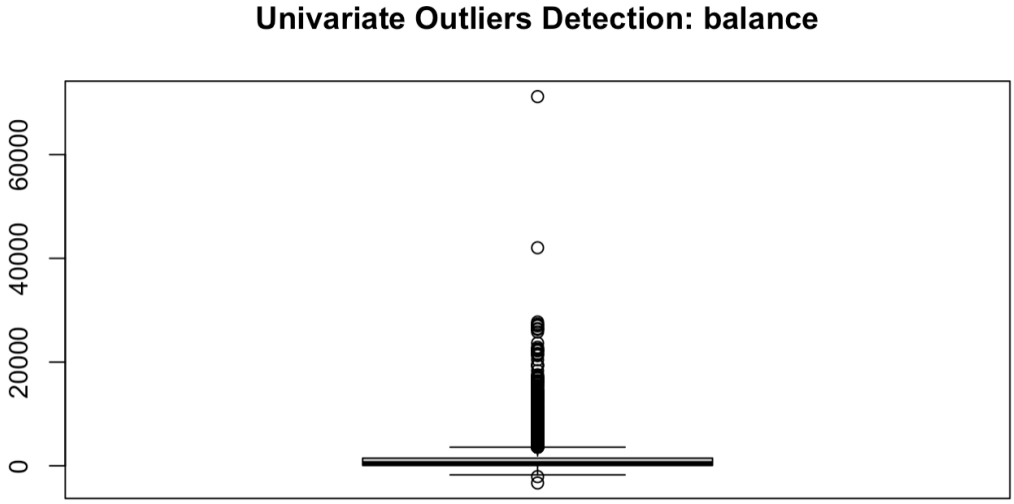
dropped, since it is almost 82% with -1 value, and we cannot use it. Even though our previous imputations will not be used eventually they remain in our report, to show the flow and the process of our dataset until we reach this conclusion.

```
[1] "Original:"

 cellular telephone   unknown
     2896       301         0

 cellular telephone   unknown
90.584923  9.415077  0.000000
[1] "MICE:"

 cellular telephone   unknown
     4100       421         0

 cellular telephone   unknown
90.687901  9.312099  0.000000
```

OUTLIERS

The provided data and charts help us see outliers in the bank's records. Most of these outliers are in the 'balance' section. Other areas like 'age', 'day', 'duration', 'previous', and 'y' have a few of these unusual values too, but not as many as 'balance'. When we look closer at 'balance', there are some people who have way more money than most. We also looked at a small part of the data that showed people with high balances, their age, job, and other details. From this, we can tell things like if they have missed any loan payments. In short, our main observation is that the 'balance' section has some very high values which can affect our analysis, so we should be careful with them.

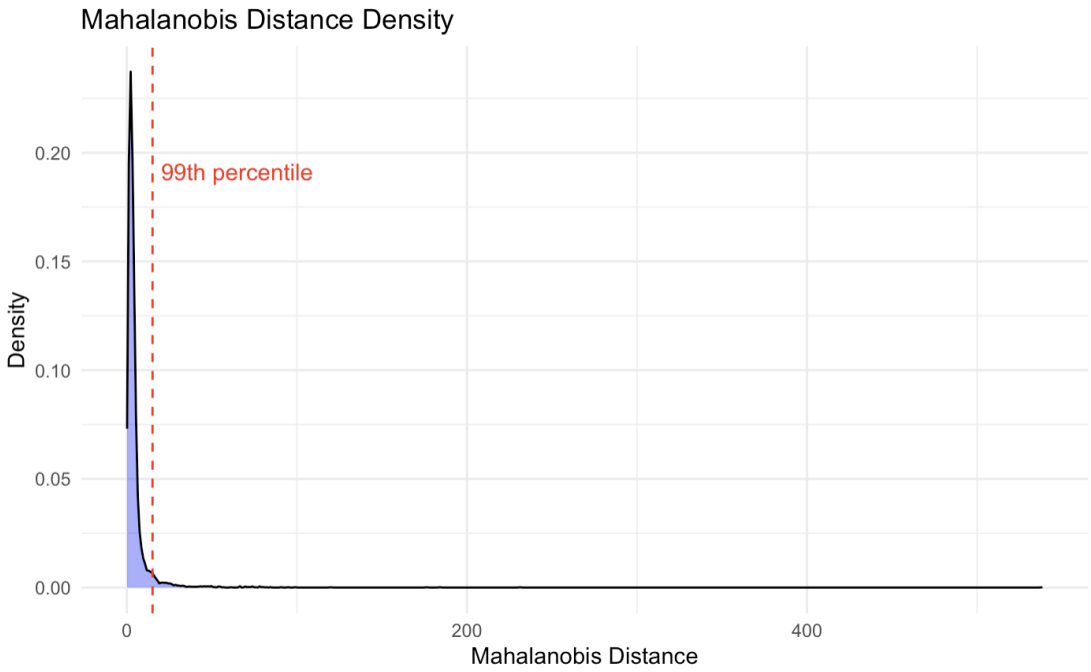




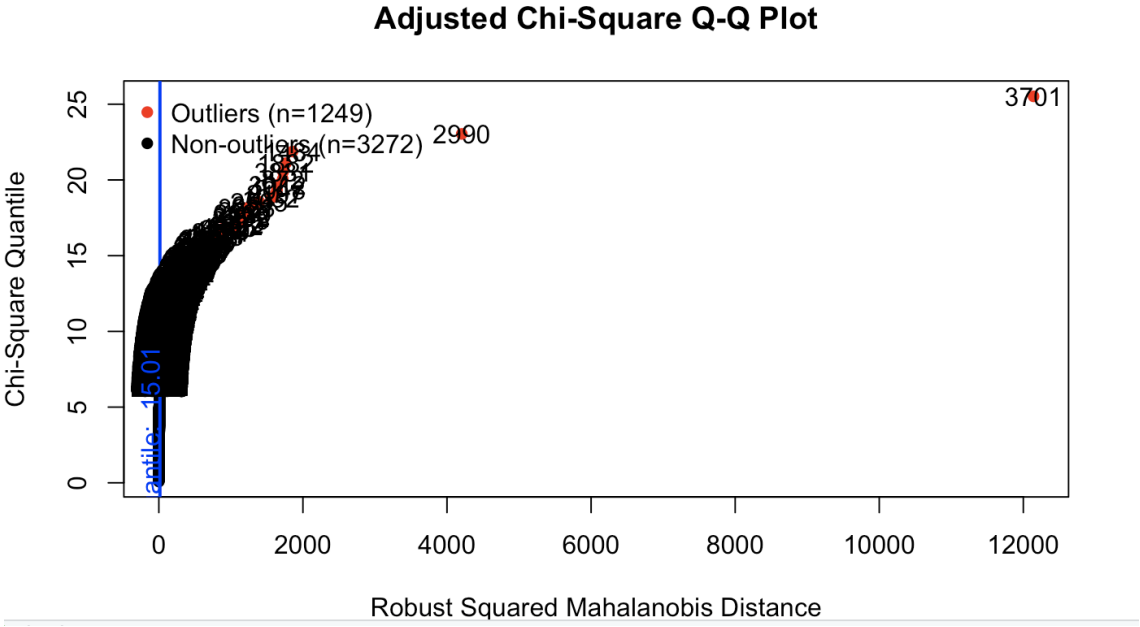
Since the goal is to predict a binary outcome based on the features, the relevance of outliers becomes crucial. Outliers might indicate rare but potentially important scenarios, like high-value clients or clients with unique attributes that sway their decision in favour or against subscribing.

Given the nature of banking datasets, outliers, especially in financial metrics, can represent high-value or unique clients. Removing these outliers might result in losing insights about a crucial segment of the client base. As the dataset's objective is to predict term deposit subscription, understanding the behaviour of high-value clients might be vital. Furthermore, if we consider the wealth distribution in real-world scenarios, it's skewed with a smaller fraction of individuals holding a significant portion of the wealth.

However, we have to use robust methods. While the decision is to keep the outliers, it's essential to ensure that the predictive model is not overly sensitive to these outliers. For a more in-depth insight, consider segmenting the data based on certain thresholds. This way, we can analyze the behaviour of outliers, ensuring that unique patterns aren't weakened in broader analysis.



We visualized the density distribution of the Mahalanobis Distance, a measure used to determine the distance between a point and a distribution. From the plotted graph, it can be observed that the majority of data points cluster closely near the origin, with a sharp decline in density as the Mahalanobis Distance increases. Notably, the 99th percentile is indicated with a red dashed line, highlighting where only 1% of the observed data points exceed this distance. This analysis assists in understanding the spread and relationships of the data points in the given dataset.

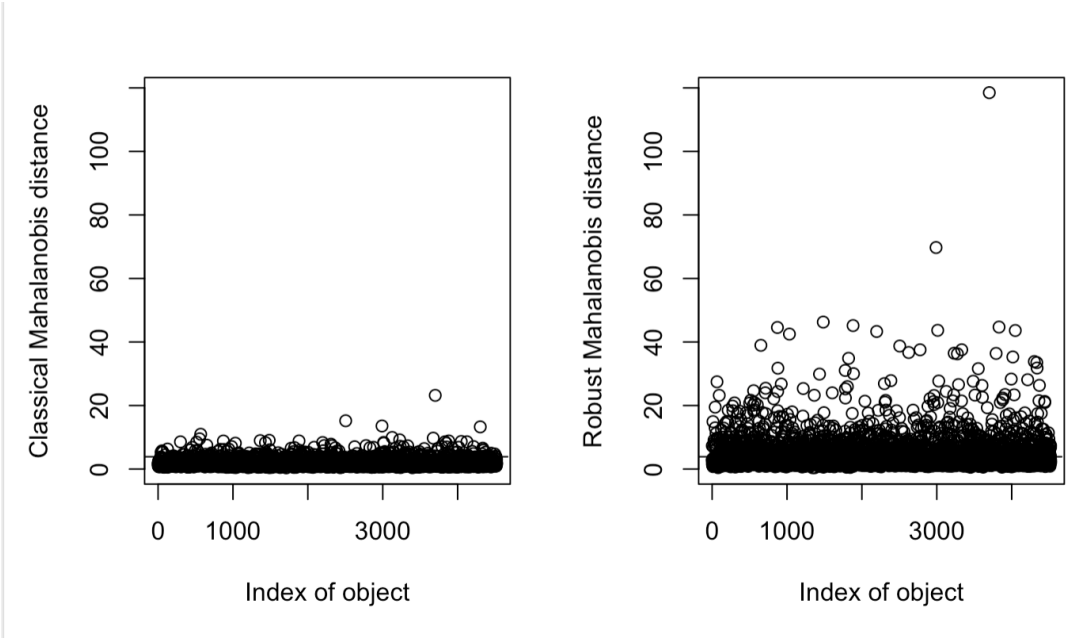


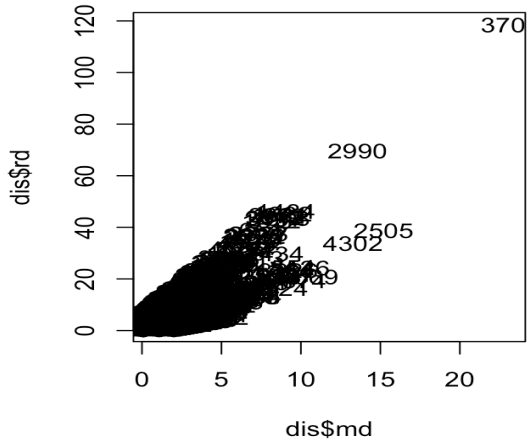
The presented plot showcases an "Adjusted Chi-Square Q-Q Plot," which is a graphical technique used to determine if the data follows a multivariate normal distribution. This plot places the quantiles of the observed distribution of the Robust Squared Mahalanobis Distance against the expected quantiles of the chi-square distribution. If the data is

multivariate normal, most of the data points should lie close to the straight 45-degree reference line.

From the plot, we notice a clear deviation from the reference line, especially for higher quantiles. This deviation is indicative of potential outliers in the data. In this plot, outliers are highlighted in red and marked as "Outliers" with a count of 1,249, whereas the non-outliers are presented in black and have a count of 3,272.

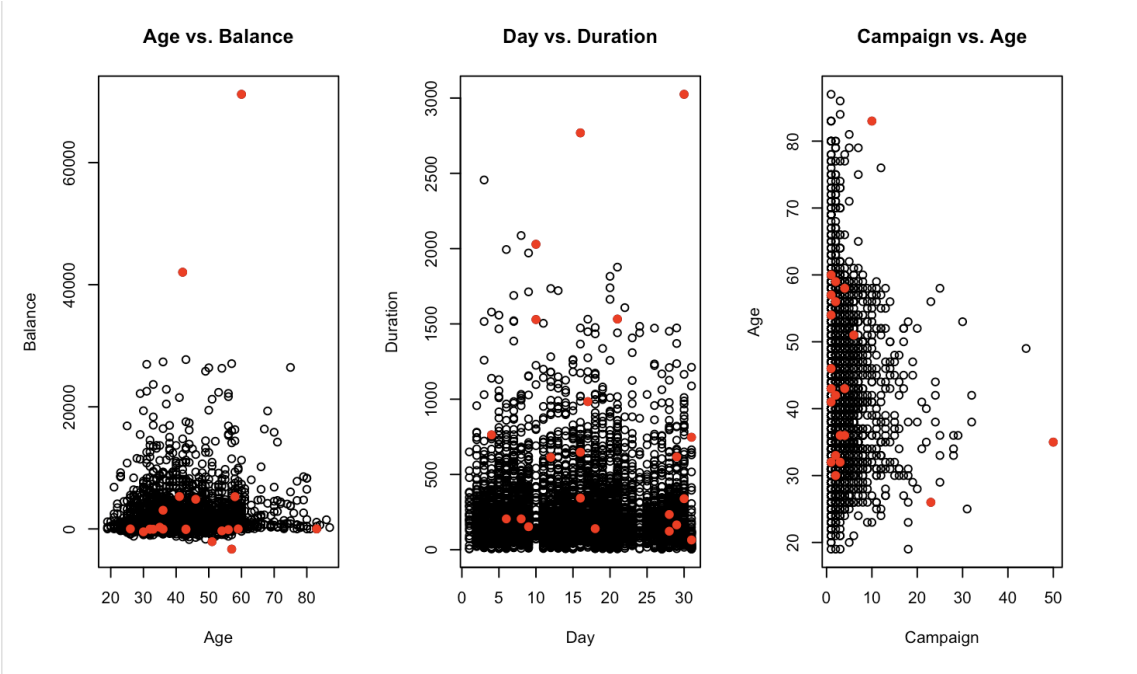
In conclusion, the "Adjusted Chi-Square Q-Q Plot" coupled with the mvn function helps in identifying multivariate outliers in the dataset. In this analysis, 1,249 data points were flagged as potential outliers. This is significant as multivariate outliers can greatly impact statistical analyses and modelling, and identifying them is a crucial step in data pre-processing.





In this analysis, the chemometrics package in R was employed to detect multivariate outliers in the dataset, specifically within the columns "age", "balance", "day", "duration", and "campaign". The primary metric utilized was the Mahalanobis distance, both in its classical and robust forms. This distance measures how far a particular data point is from the center of the distribution, accounting for the data's correlation structure. From the

visualizations, it's evident that data points with high Mahalanobis distances, in both classical and robust interpretations, could be potential outliers. Particularly, in the scatter plot comparing these two distances, points distant from the primary cluster are highlighted as potential outliers. This approach provides a systematic way to identify potential outliers in multivariate datasets, where univariate analyses might miss the intricacies of multivariate outlier patterns.




```
Top Outliers LOF Scores:
[1] 36.140392 16.299055  4.887982

Corresponding Rows for Top Outliers:
```

	age <int>	balance <int>	day <int>	duration <int>	campaign <int>
3701	60	71188	6	205	1
2990	42	42045	8	205	2
4518	57	-3313	9	153	1

3 rows

We employed the Local Outlier Factor (LOF) method to detect potential outliers within our dataset. The LOF scores quantify the degree of abnormality of each data point relative to its neighbors. From the analysis, we identified three top outliers with LOF scores of 36.140392, 16.299055, and 4.887982, respectively. A visualization of the data, with these outliers highlighted, indicates that they deviate significantly from the main cluster of data points in variables such as "Age vs. Balance", "Day vs. Duration", and "Duration vs. Age". Further examination of the specific rows corresponding to these outliers in the dataset revealed the following:

- Row 3701: A 60-year-old individual with a balance of 71,188, contacted on the 6th day for a campaign duration of 205.

- Row 2990: A 42-year-old individual with a balance of 42,045, contacted on the 8th day for a campaign duration of 205.
- Row 4518: A 57-year-old individual with a balance of -3,313, contacted on the 9th day for a campaign duration of 153.

These data points can be considered as anomalies due to their distinct values in the above variables when compared to the general trend in the dataset. As a next step, it may be important to further investigate the reasons behind these deviations and decide whether to retain or exclude these outliers from subsequent analyses, depending on the specific objectives of the study.

In analyzing the provided methods for outlier detection, it becomes evident that multiple approaches are adopted to cater to different data structures. Univariate outlier detection using boxplots serves as a straightforward approach, focusing on each variable individually. This is simple and effective for single-variable anomalies but doesn't account for the multidimensional nature of some datasets.

The Mahalanobis Distance offers a metric to identify multivariate outliers by considering the covariance between variables. Its graphical representation makes it easier to visualize and discern potential outliers. Similarly, the MVN package's multivariate outlier detection considers the multivariate nature of the data and is based on robust statistical measures. The Chemometrics package, too, offers multivariate outlier detection but might be better suited for specific datasets.

The Local Outlier Factor (LOF) algorithm, on the other hand, analyzes the local density deviation of an observation concerning its neighbors. It can differentiate between clusters of varying densities, making it particularly effective for datasets with multiple clusters.

Given the results and the diverse nature of these methods, the best approach depends on the specific characteristics and requirements of the dataset in question. For purely univariate data, the boxplot method is effective and intuitive. For multivariate datasets where relationships between variables matter, the Mahalanobis Distance or MVN package could be ideal. However, if the dataset is expected to have clusters of varying densities, the LOF algorithm might be the most suitable. It is also worth considering a approach employing a combination of methods to ensure comprehensive outlier detection

3. DESCRIPTIVE ANALYSIS

In this chapter, we present a detailed descriptive analysis of a dataset using an R script. The dataset is subjected to various examinations, such as data dimension exploration, summary statistics, and a range of visualizations. This comprehensive analysis aims to provide insights into the trends within the dataset. For this analysis, we first segregated the data between the numerical and categorical variables.

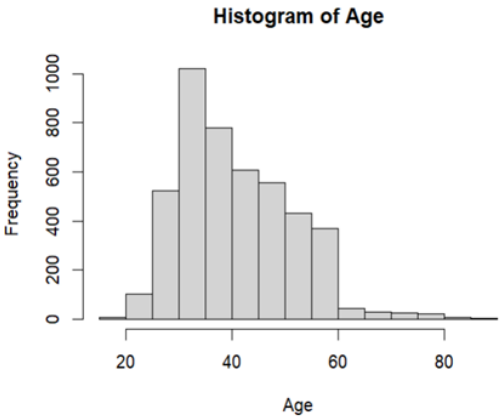
```
> summary(df)
      age                job          marital         education    default       balance      housing
 Min.   :19.00   management :969   divorced: 528   primary  : 678   no :4445   Min.   :-3313   no :1962
 1st Qu.:33.00   blue-collar:946   married :2797   secondary:2306   yes:  76   1st Qu.:   69   yes:2559
 Median :39.00   technician :768   single  :1196   tertiary :1350              Median :  444
 Mean   :41.17   admin.     :478                   unknown  : 187              Mean   : 1423
 3rd Qu.:49.00   services   :417                                               3rd Qu.: 1480
 Max.   :87.00   retired    :230                                               Max.   :71188
                 (Other)    :713
  loan           contact           day              month          duration       campaign          pdays
 no :3830   cellular :2896   Min.   : 1.00   may    :1398   Min.   :   4   Min.   : 1.000   Min.   : -1.00
 yes: 691   telephone: 301   1st Qu.: 9.00   jul    : 706   1st Qu.: 104   1st Qu.: 1.000   1st Qu.: -1.00
            unknown  :1324   Median :16.00   aug    : 633   Median : 185   Median : 2.000   Median : -1.00
                             Mean   :15.92   jun    : 531   Mean   : 264   Mean   : 2.794   Mean   : 39.77
                             3rd Qu.:21.00   nov    : 389   3rd Qu.: 329   3rd Qu.: 3.000   3rd Qu.: -1.00
                             Max.   :31.00   apr    : 293   Max.   :3025   Max.   :50.000   Max.   :871.00
                                             (Other): 571
    previous          poutcome      y
 Min.   : 0.0000   failure: 490   no :4000
 1st Qu.: 0.0000   other  : 197   yes: 521
 Median : 0.0000   success: 129
 Mean   : 0.5426   unknown:3705
 3rd Qu.: 0.0000
 Max.   :25.0000
```

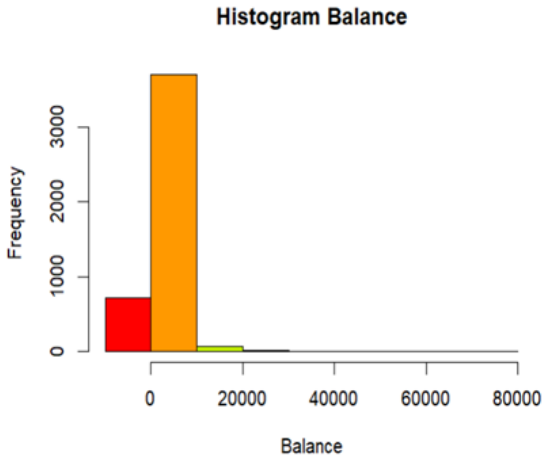
Numerical Columns

Focusing on the numerical variables, we first collected basic statistics to offer a more detailed view of the data. These statistics, including mean, median, and standard deviation, provide a clearer understanding of the data's central characteristics and variation.

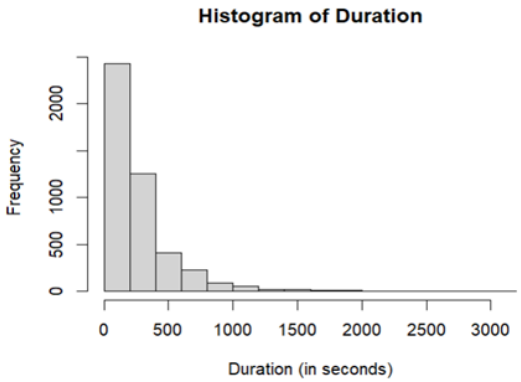
```
> basicStats(data.frame(age,balance,day,duration,campaign,pdays,previous))
                    age        balance          day       duration     campaign         pdays    previous
nobs          4521.000000    4521.000000 4521.000000    4521.000000 4521.000000    4521.000000 4521.000000
NAs              0.000000       0.000000    0.000000       0.000000    0.000000       0.000000    0.000000
Minimum         19.000000   -3313.000000    1.000000       4.000000    1.000000      -1.000000    0.000000
Maximum         87.000000   71188.000000   31.000000    3025.000000   50.000000     871.000000   25.000000
1. Quartile     33.000000      69.000000    9.000000     104.000000    1.000000      -1.000000    0.000000
3. Quartile     49.000000    1480.000000   21.000000     329.000000    3.000000      -1.000000    0.000000
Mean            41.170095    1422.657819   15.915284     263.961292    2.793630      39.766645    0.542579
Median          39.000000     444.000000   16.000000     185.000000    2.000000      -1.000000    0.000000
Sum         186130.000000 6431836.000000 71953.000000 1193369.000000 12630.000000 179785.000000 2453.000000
SE Mean          0.157294      44.760716    0.122663       3.864707    0.046250       1.489047    0.025187
LCL Mean        40.861721    1334.904929   15.674805     256.384577    2.702956      36.847384    0.493199
UCL Mean        41.478469    1510.410709   16.155764     271.538007    2.884303      42.685905    0.591959
Variance       111.856238 9057921.748594   68.024016   67525.469519    9.670897   10024.239560    2.868153
Stdev           10.576211    3009.638142    8.247667     259.856633    3.109807     100.121124    1.693562
Skewness         0.699037       6.592054    0.094564       2.770580    4.740767       2.715269    5.871361
Kurtosis         0.345583      88.250899   -1.040576      12.508007   37.108750       7.942161   51.912098
```

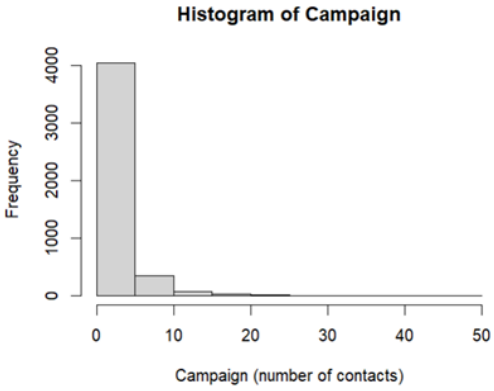
Creating histograms, we visually represent the distributions of the most relevant numerical variables:

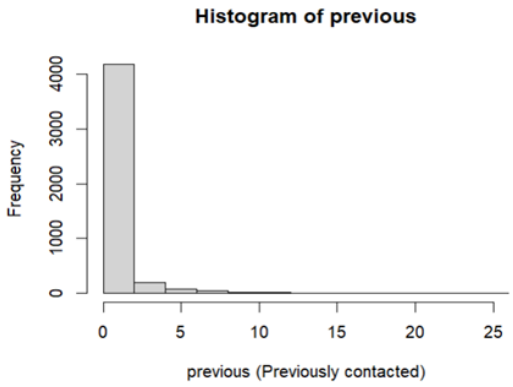


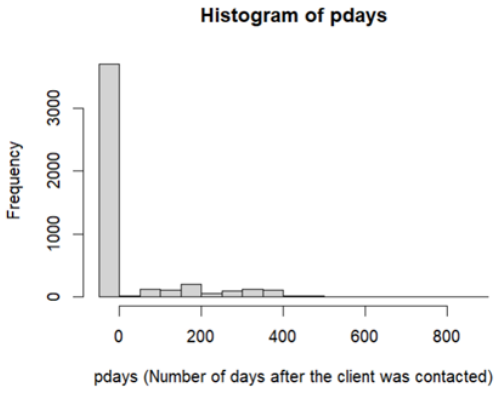


For a better study of the target population, we focused on the age and balance variables in the figures above. With this we can conclude that the target population stands between 30 and 45 years old and has an average yearly balance of 15000 $.







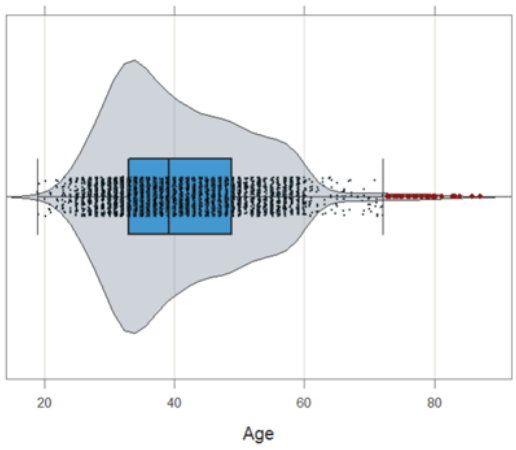


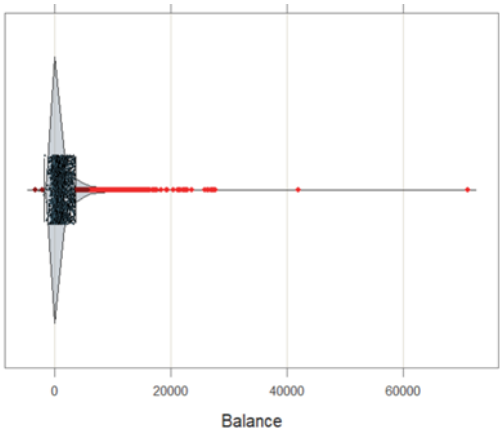
Whereas, with the analysis of the variables related to calls ('campaign,' 'duration,' 'previous,' and 'pdays'), we got the following conclusions:

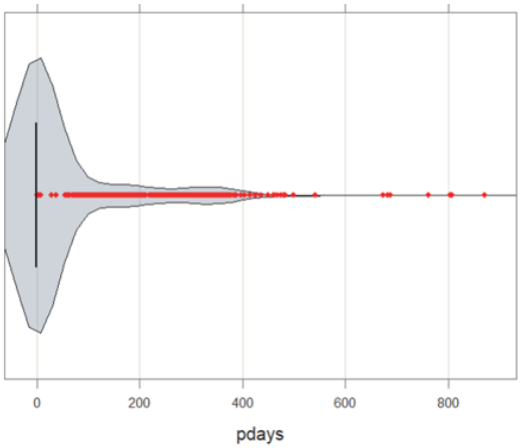
1. **Call Duration**: Most calls were short in duration. Many conversations with customers were brief.

2. **Number of Contacts**: The data suggests that there were relatively few contact attempts with customers.
3. **Previous Contacts:** A significant portion of the target population had not been contacted previously.

For identifying outliers in the dataset, we have used VBS plots, that integrates visualizations of the box plot, density distribution, individual data values and quick detection of outliers (dots in red):

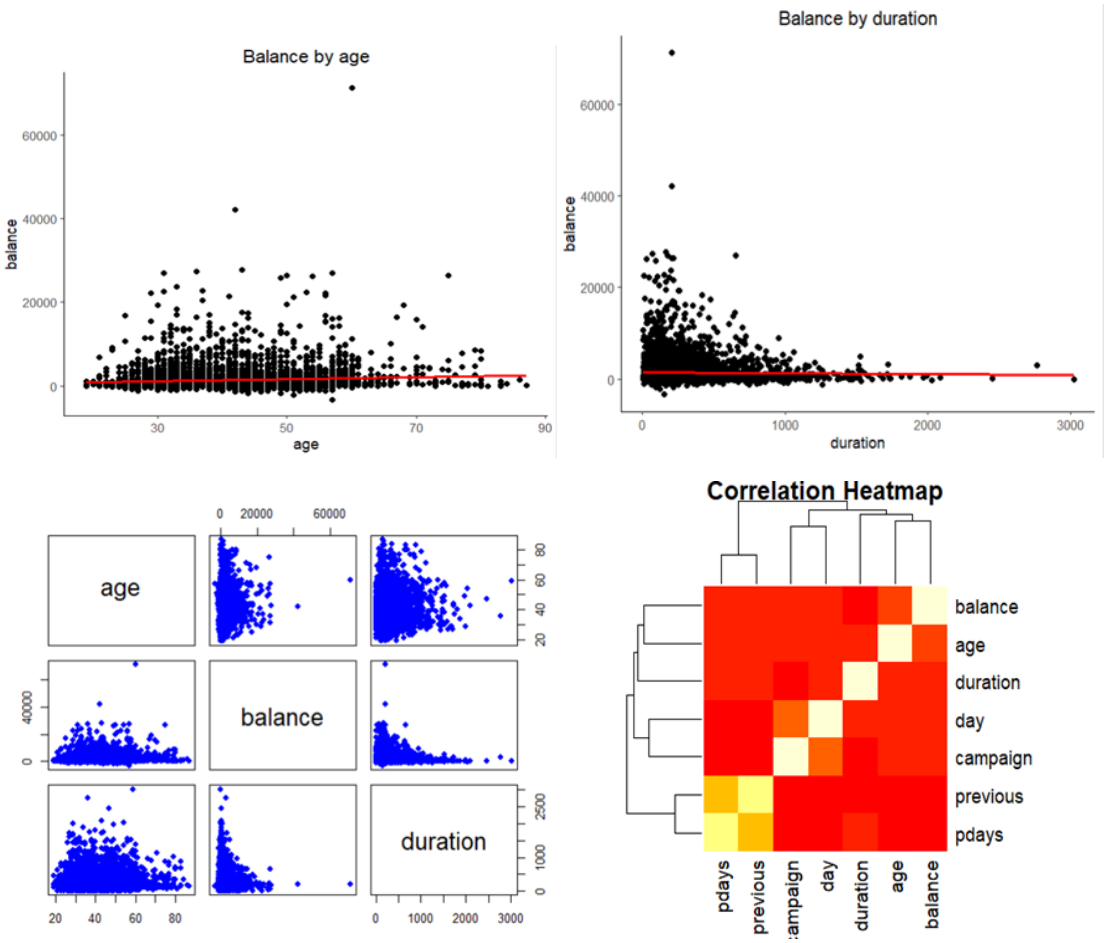






With these plots, it is easily noticeable the number of outliers that the numerical variables have, and the density distribution of the data. Also, we identify that, since most of the target population was not previously contacted, the boxplot plot of pdays is compacted to -1, this, together with the previous preprocessing evaluation, helps us identify that this column is not good for analysis.

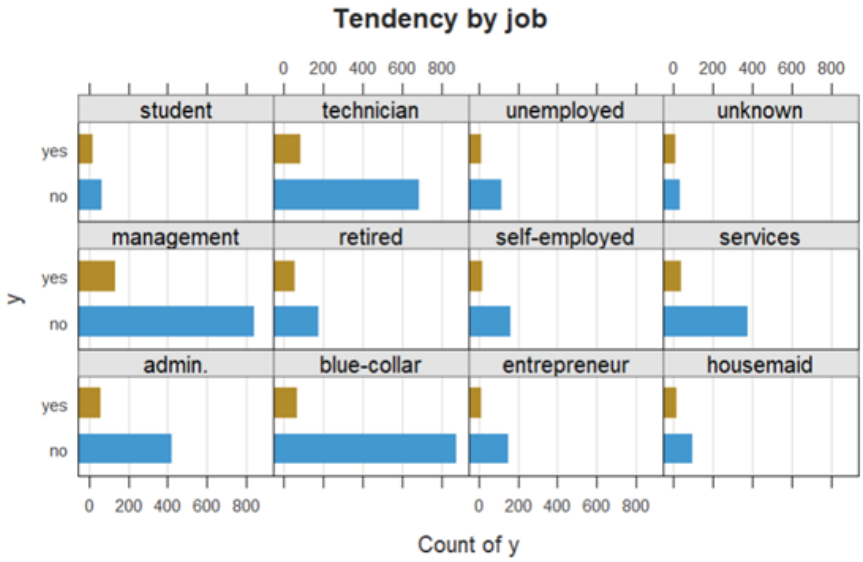
To explore relationships between numerical variables, we've included scatterplots, pairs and heatmap plots, helping us identify trends and correlations between these variables.

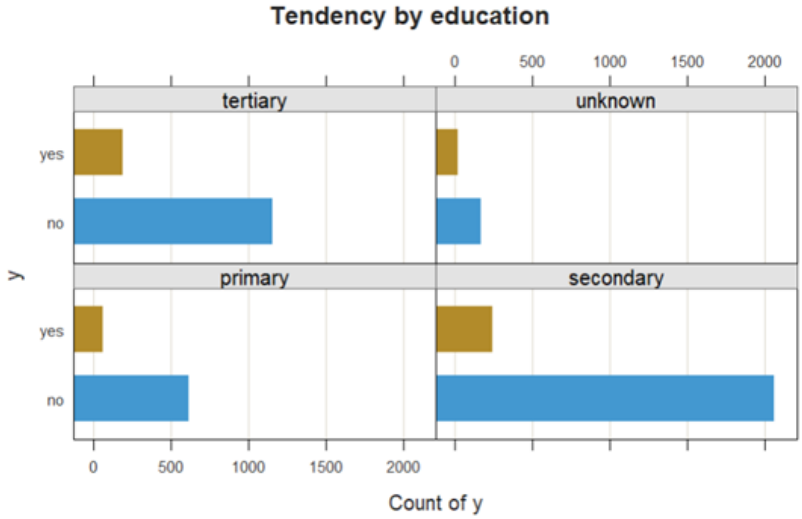


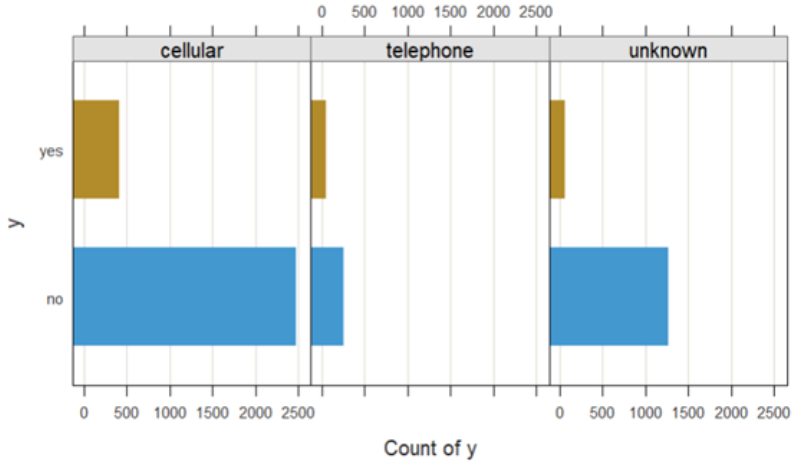
We create pairs and heatmap plots to visualize the correlations between 'age,' 'balance,' and 'duration.' While these plots hint at correlations, it's important to note that the presence of outliers can impact the strength of these relationships.

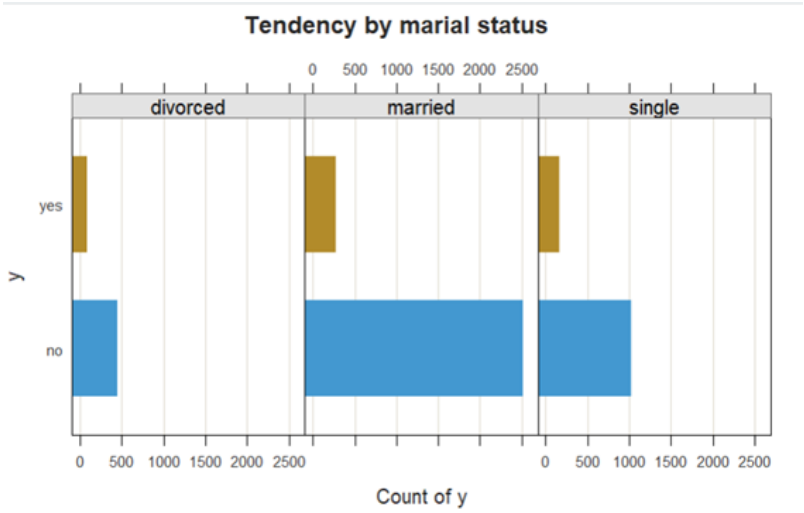
Categorical Columns

The categorical variables allow us to have a deeper analysis over the target population. For this analysis, we counted the occurrences of different categories within the most important columns: job, education, contact, and marital. Using these, we've plotted the count of subscriptions for a term deposit.


Tendency by job
student
technician
unemployed
unknown
management
retired
self-employed
services
admin.
blue-collar
entrepreneur
housemaid
yes
no
y
0 200 400 600 800
Count of y


Tendency by education
tertiary
unknown
primary
secondary
yes
no
y
0 500 1000 1500 2000
Count of y


Tendency by contact
cellular
telephone
unknown
yes
no
y
0 500 1000 1500 2000 2500
Count of y



In analysing the visualizations, we draw conclusions of the population that is more likely to subscribe to a term deposit:

1. **Marital Status:** Individuals who are not married (single or divorced) are more inclined to subscribe to a term deposit. This suggests that marital status plays a significant role in the decision-making process.
2. **Contact Method:** Subscribers are more likely to have been contacted by telephone. This communication channel seems to be more effective in encouraging term deposit subscriptions.
3. **Education Level:** It appears that individuals with only a primary education are more receptive to subscribe to term deposit.

By understanding the demographics and preferences of potential subscribers, it is possible to optimize and better select which variables to use when creating models.

4. PCA ANALYSIS

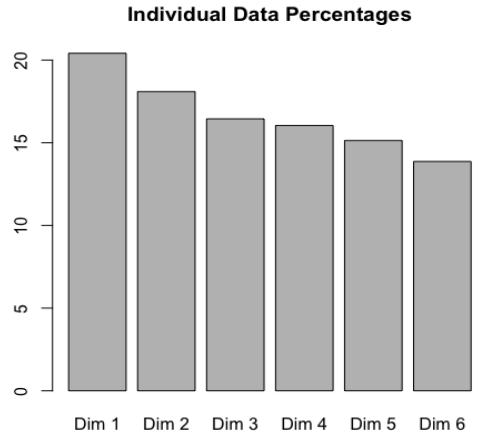
This chapter provides the Principal Component Analysis of our dataset. We used R scripts to execute PCA on the dataset and acquire the best 2D plane to visualize its data. Firstly, we split the dataframe into numerical and categorical parts, since PCA is conducted only on numerical data. We are left with 6 numerical columns as shown:

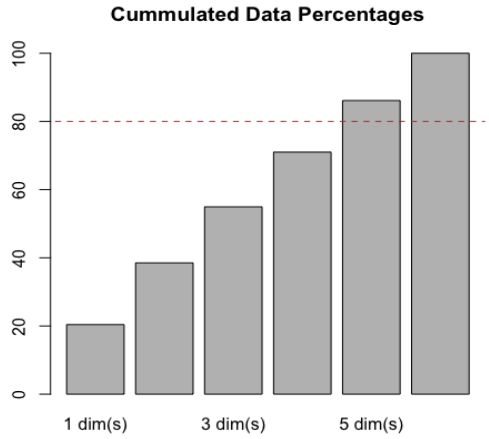
```
> summary(df[numerical])
      age            balance            day            duration       campaign         previous       
 Min.   :19.00   Min.   :-3313   Min.   : 1.00   Min.   :   4   Min.   : 1.000   Min.   : 0.0000  
 1st Qu.:33.00   1st Qu.:   69   1st Qu.: 9.00   1st Qu.: 104   1st Qu.: 1.000   1st Qu.: 0.0000  
 Median :39.00   Median :  444   Median :16.00   Median : 185   Median : 2.000   Median : 0.0000  
 Mean   :41.17   Mean   : 1423   Mean   :15.92   Mean   : 264   Mean   : 2.794   Mean   : 0.5426  
 3rd Qu.:49.00   3rd Qu.: 1480   3rd Qu.:21.00   3rd Qu.: 329   3rd Qu.: 3.000   3rd Qu.: 0.0000  
 Max.   :87.00   Max.   :71188   Max.   :31.00   Max.   :3025   Max.   :50.000   Max.   :25.0000  
```

After executing the PCA, we get the eigenvalues and the rotation matrix for the 6 different dimensions:

```
> print(pca)
Standard deviations (1, .., p=6):
[1] 1.1066900 1.0419491 0.9933714 0.9811107 0.9529797 0.9121644

Rotation (n x k) = (6 x 6):
                PC1         PC2         PC3         PC4         PC5         PC6
age       0.09421272 -0.67118571  0.31800700 -0.1679807  0.63697552  0.07449768
balance   0.11083673 -0.69397914 -0.01840454  0.2233769 -0.67245040 -0.06068480
day      -0.59532770 -0.03303515  0.14779614  0.4384002  0.01836680  0.65583372
duration  0.27964444  0.23451275  0.80040989  0.4279559 -0.05770063 -0.19917613
campaign -0.63296663 -0.10458839 -0.03343592  0.2168777  0.14165457 -0.72124518
previous  0.38154468 -0.02960409 -0.48468049  0.7067590  0.34400570 -0.02799720
```

First of all, the meaning of the eigenvalues (standard deviations) is how much of the variance does each dimension of the transformation explain, and the matrix displays how each of the variables affect the new dimensions. To get a better view of what the deviations mean, we can show their respective values as a percentage, also referred to as data inertia, as well as its cumulative function.

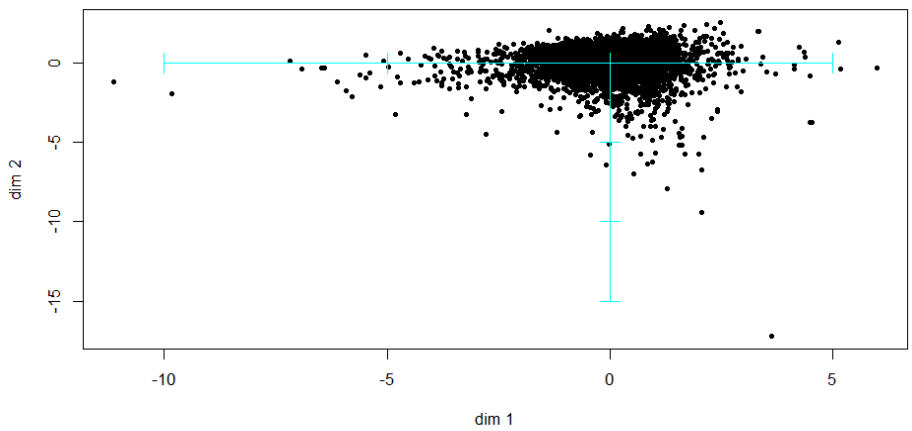


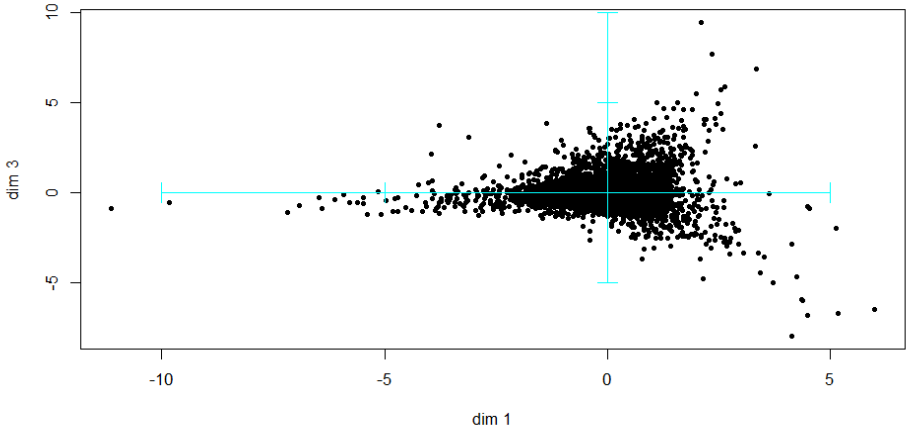


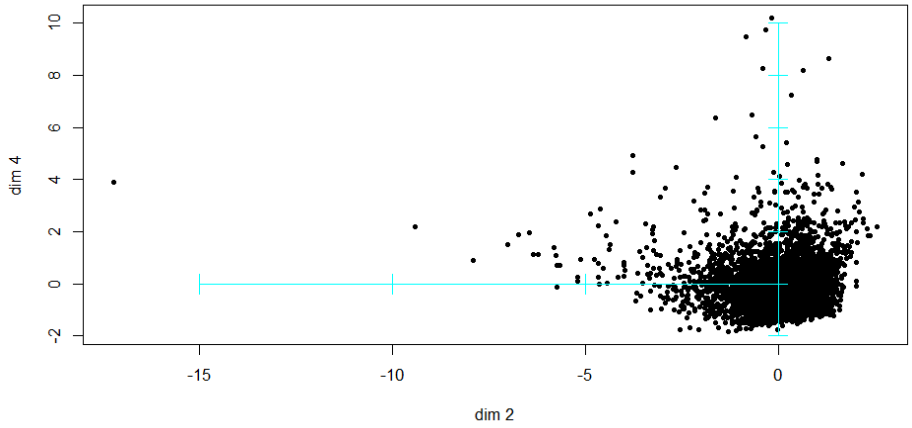
The red dashed line on the cumulative graph represents the 80% threshold of the total data. We use it to find the number of dimensions which represent most of the variance, which in our case is 5, and the only dimensions we'll use for the PCA analysis. We also observe that the PCA can't reduce the dimensions very much as all numerical variables seem not to be greatly correlated.

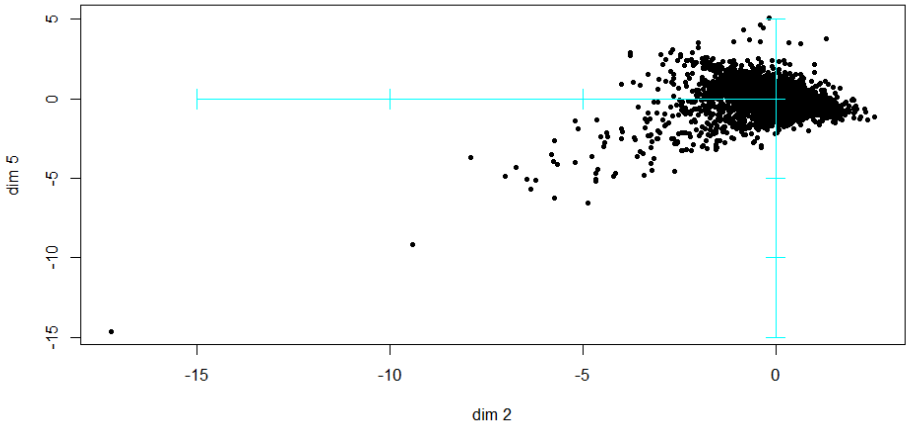
If we plot our data in some of the combinations of these new dimensions, we see some tails appear. That is probably because, even though the covariance is 0, the variables aren't independent, and thus no

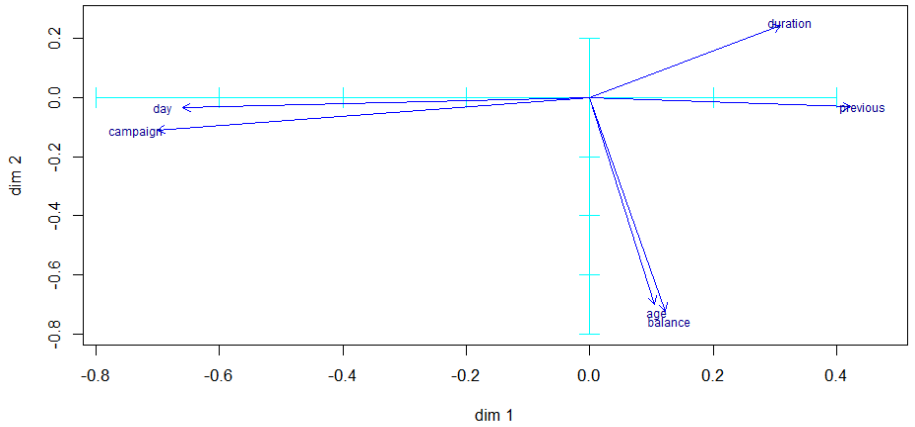
t normally distributed.

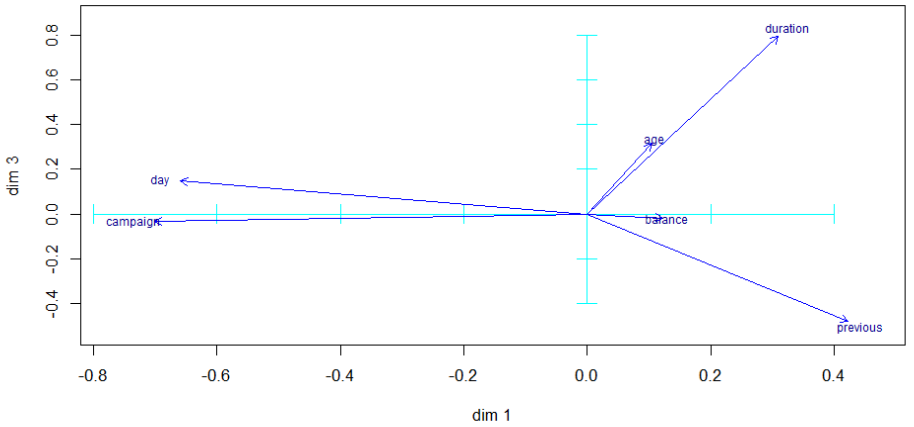


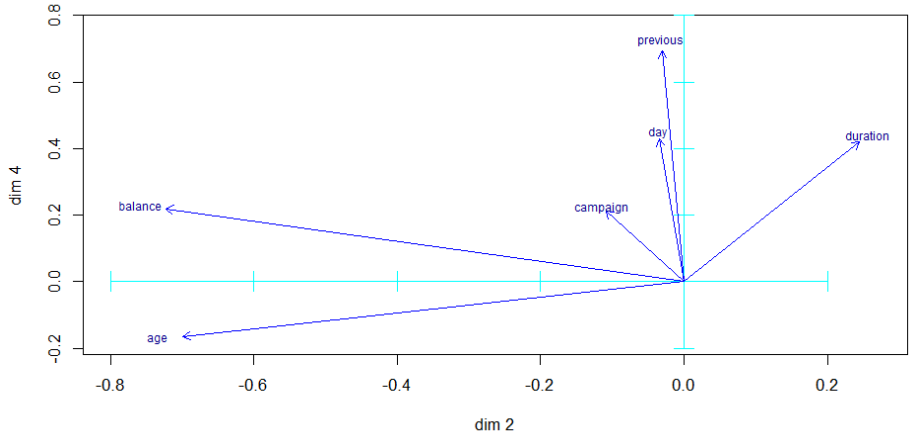


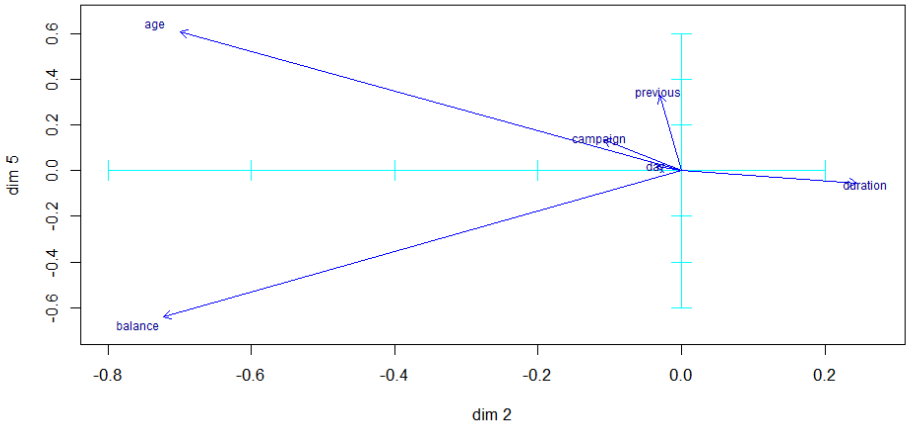





duration
day
previous
campaign
age
balance
dim 2
dim 1
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4


duration
age
day
campaign
balance
previous
dim 3
dim 1
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4


previous
day
duration
balance
campaign
age
dim 4
dim 2
0.8
0.6
0.4
0.2
0.0
-0.2
-0.8
-0.6
-0.4
-0.2
0.0
0.2


age
previous
campaign
day
duration
balance
dim 5
dim 2
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-0.6
-0.4
-0.2
0.0
0.2

Additionally, if we use the original variables in those plots instead of the individuals, we can get a rough feeling for what each of the new dimensions mean. In those plots we see:

- The first dimension seems to represent if the number of calls decreased, and the final call was a bit longer and towards the start of the month.
- The second dimension is slightly about longer calls but mainly about young clients or with less money.
- The third dimension is for older people that hadn't been called much with a focus on long calls.
- The fourth dimension is for long calls at the end of the month, and most importantly, of the clients that had received many calls before.
- The fifth dimension is for older people without too much money.

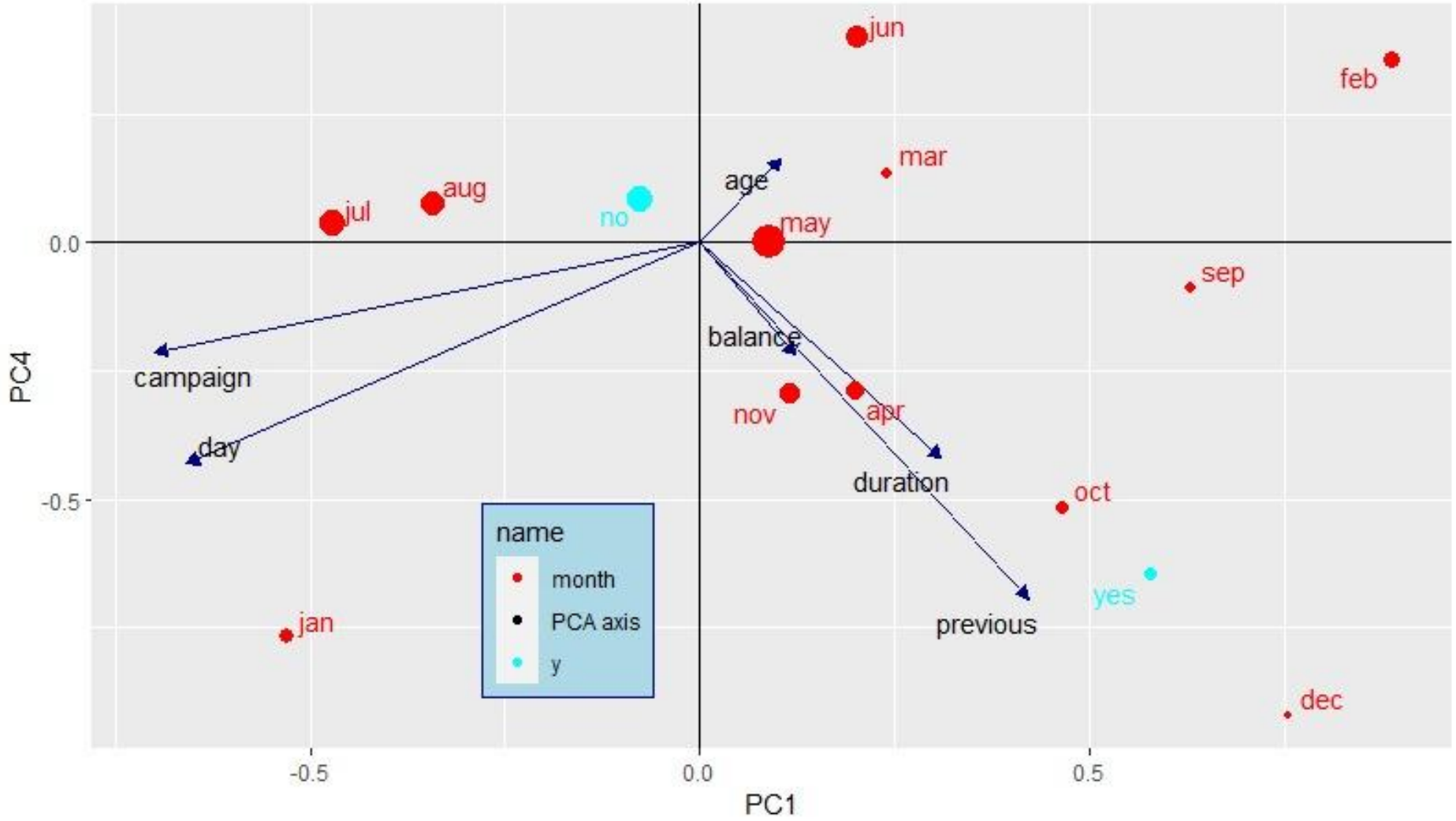
In short, the different dimensions represent the contact frequency, the client profile, the call's length, when and how the call went and the client profile again.

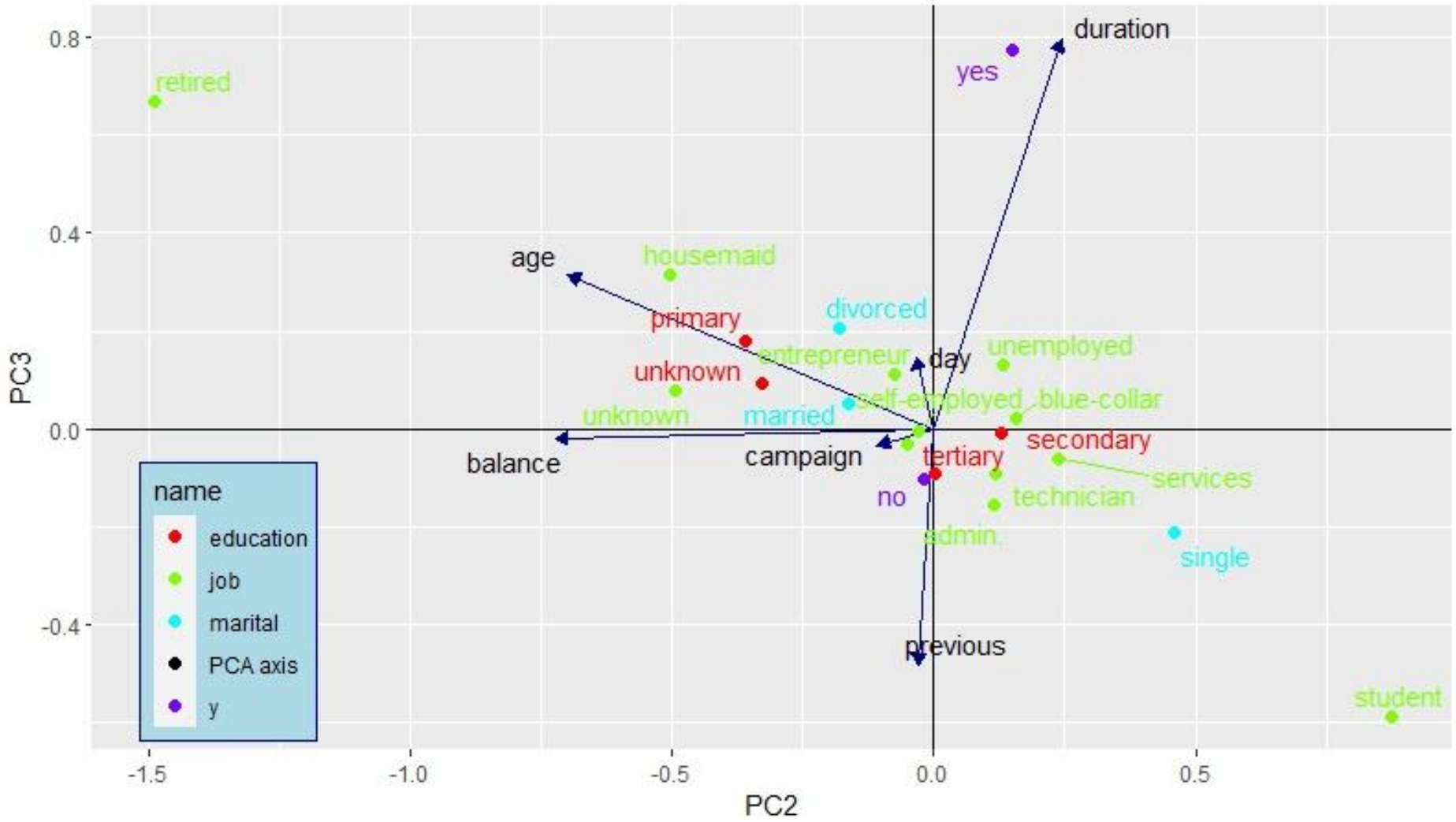
Apart from the numerical variables we also have the categorical ones. We can add their centroids to the plots to check how they relate to the numerical and how useful they may be. After that we can check individually one by one each of the ones with the most relation to the numerical variables (the centroids furthest from the origin).

From those plots we see that the most related are:

- Education with the second and third dimensions
- Job with the second and third dimensions
- Month with the first and fourth dimensions
- Outcome of the previous campaign (poutcome) with the first and fourth dimensions
- Marital status (marital) with the second and third dimensions

We can now plot those categories on their most prominent axis to see the most influential modalities and if the values are clustered. Additionally, plot the y category (our target for the analysis) to check which concept (PCA dimension) represents it the best.


jun
feb
mar
age
aug
jul
no
may
0.0
sep
balance
campaign
day
nov
apr
duration
oct
name
month
PCA axis
y
yes
previous
jan
dec
PC4
-0.5
0.0
0.5
PC1

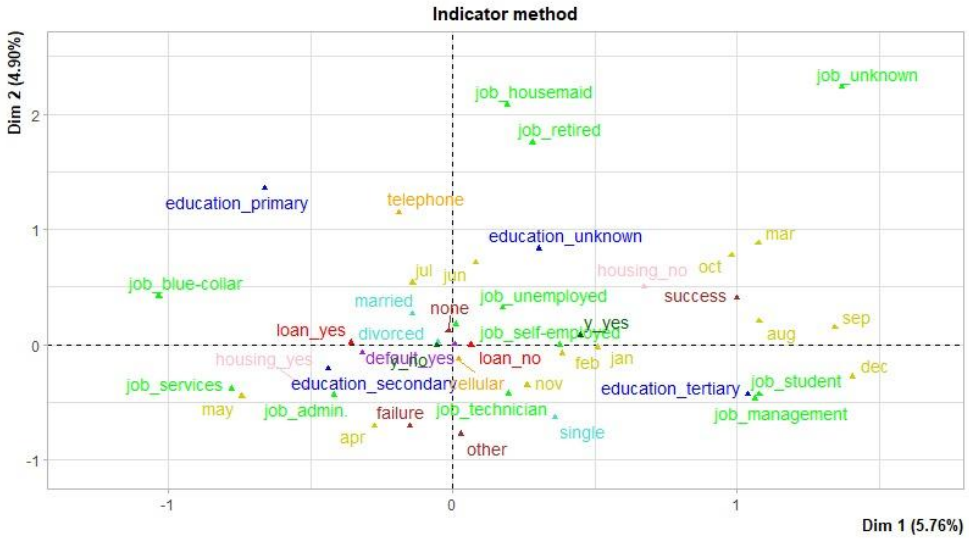

duration
yes
retired
age
housemaid
primary
divorced
unemployed
entrepreneur
day
unknown
unknown
married
blue-collar
secondary
balance
campaign
tertiary
services
technician
no
admin.
single
previous
student
name
education
job
marital
PCA axis
y
PC3
PC2
0.8
0.4
0.0
-0.4
-1.5
-1.0
-0.5
0.0
0.5

For all those factors we can conclude the following:

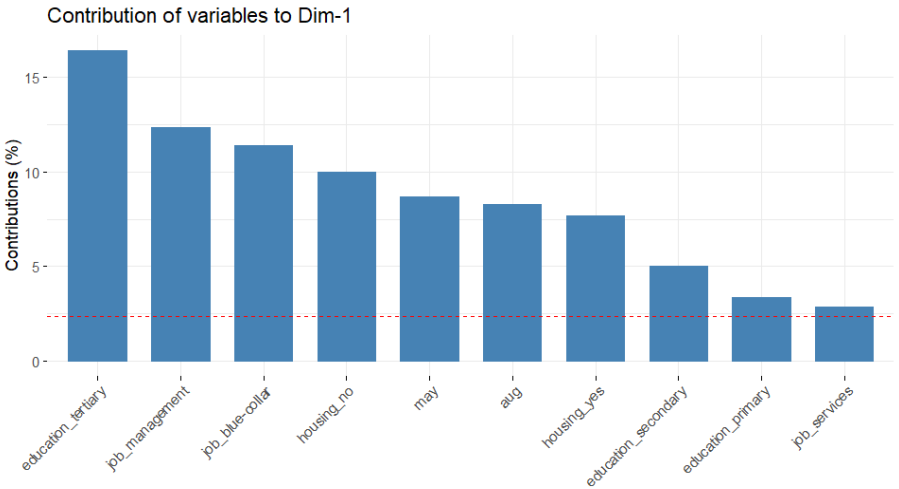
- Educational: There is a noticeable difference between the primary education and the other two levels according to the client profile.
- Job: There are three modalities that display a unique behaviour, from the rest. Those are students, housemaids and retired.
- Month: There is a slight separation between the months of autumn and winter versus spring and summer.
- Previous outcome: Although there is the noticeable cluster of the nonmodality on one of the sides aligned with previous, duration and balance. This was already expected as poutcome will always be none when the client wasn't contacted previously. (previous = 0)
- Marital status: There is a noticeable difference between people that have and haven't been married according to the client profile with two partially overlapping clusters.
- Has subscribed (y): Is most notable on call frequency and duration. And a positive result is more aligned with more previous contact, less now and longer calls.

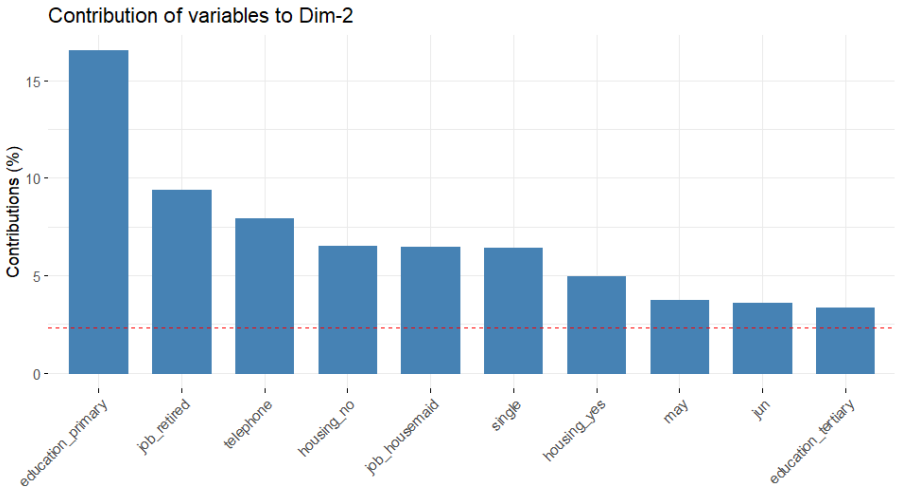
5. MULTIPLE CORRESPONDENCE ANALYSIS

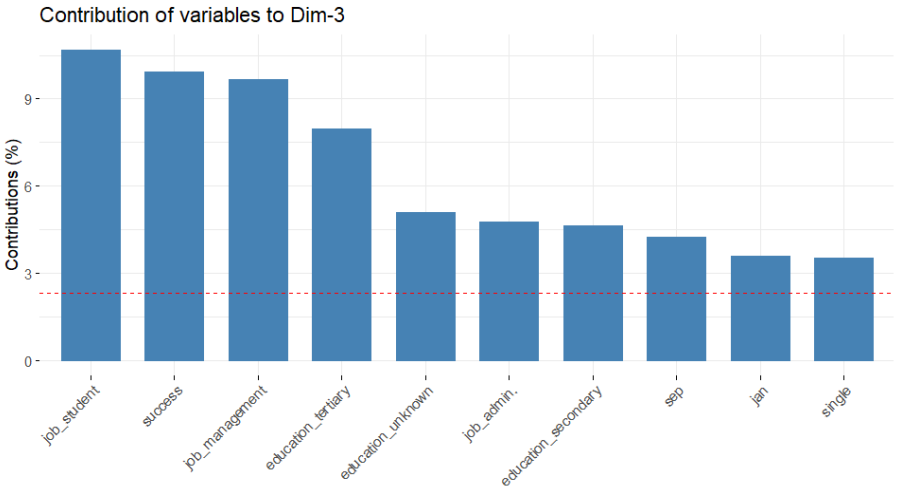
After completing the PCA which gave us insights about the correlations of numerical variables, this chapter will go through the Multiple Correspondence Analysis to gain knowledge on the relationships between categorical variables. First of all, we split the dataframe into numerical and categorical parts, as the MCA uses only categorical values, which leaves us with 9 explanatory variables and the response. We used both Indicator Analysis (Logic Table) and Burt Analysis (Burt Table) to check how they compare, which yielded similar results for our dataset.

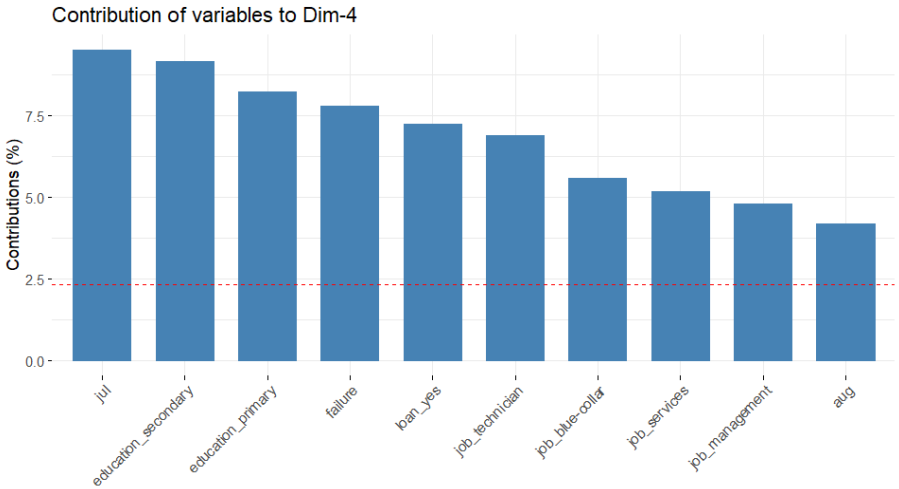


So, as both methods map the variables to the same relative locations, only changing the scale and the variance each dimension represents, we'll continue the analysis with the Burt method's results.

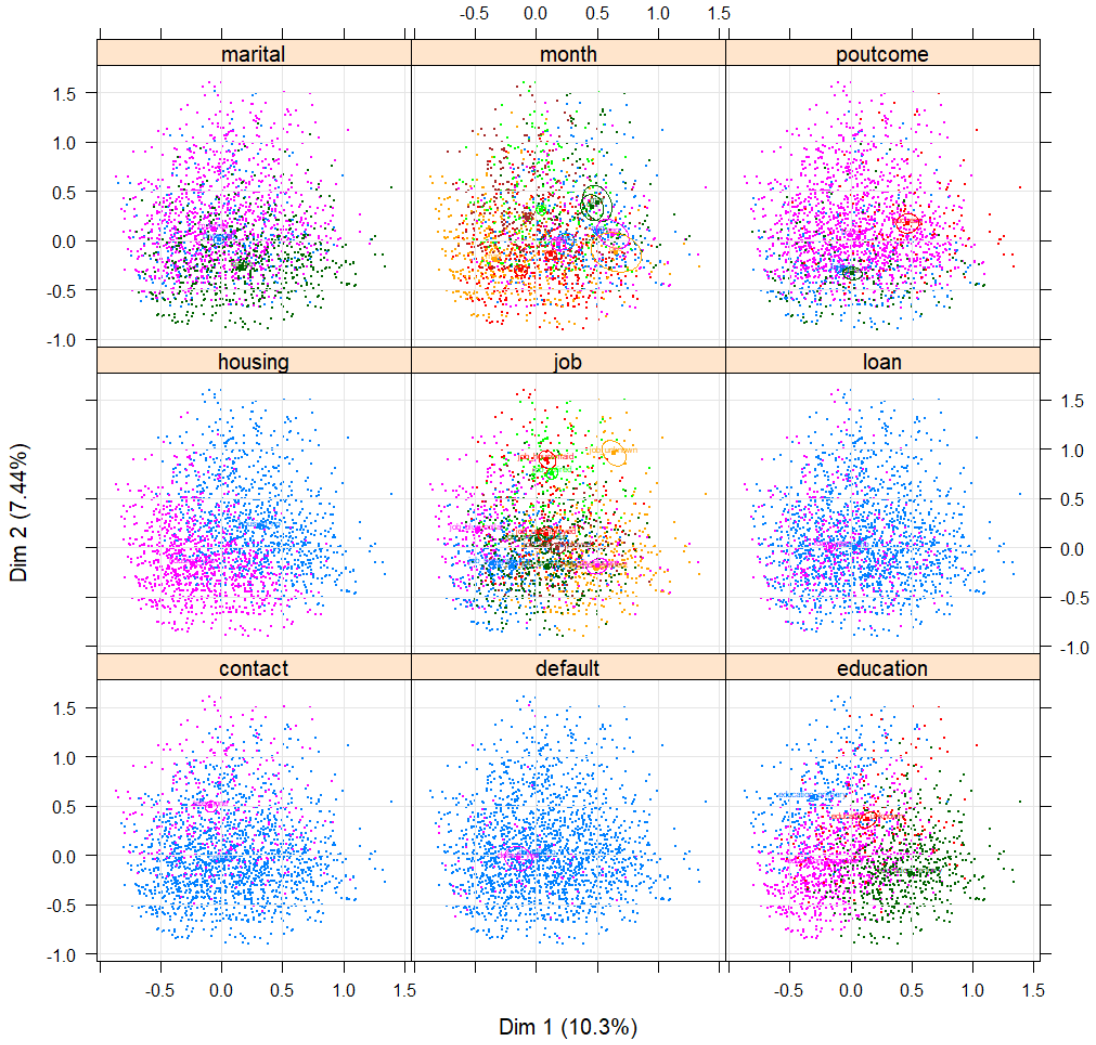

Contribution of variables to Dim-1
Contributions (%)
15
10
5
0
education_tertiary
job_management
job_blue-collar
housing_no
may
aug
housing_yes
education_secondary
education_primary
job_services


Contribution of variables to Dim-2
Contributions (%)
15
10
5
0
education_primary
job_retired
telephone
housing_no
job_housemaid
single
housing_yes
may
jun
education_tertiary

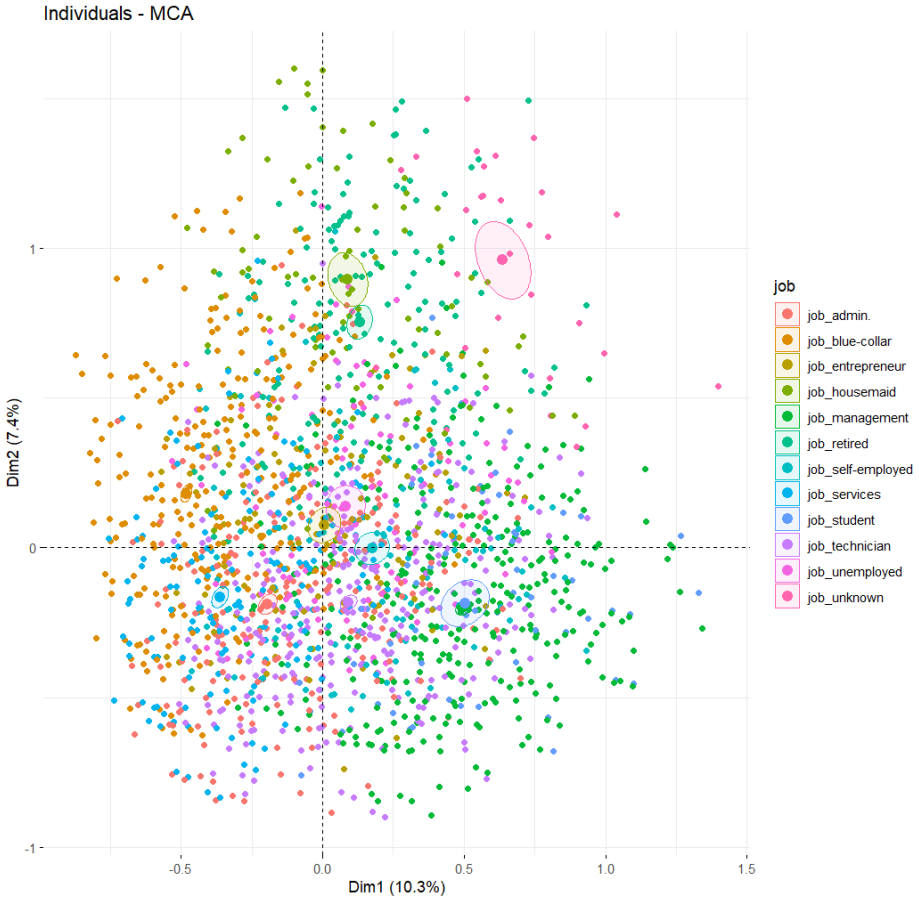

Contribution of variables to Dim-3
Contributions (%)
9
6
3
0
job_student
success
job_management
education_tertiary
education_unknown
job_admin.
education_secondary
ssp
jan
single

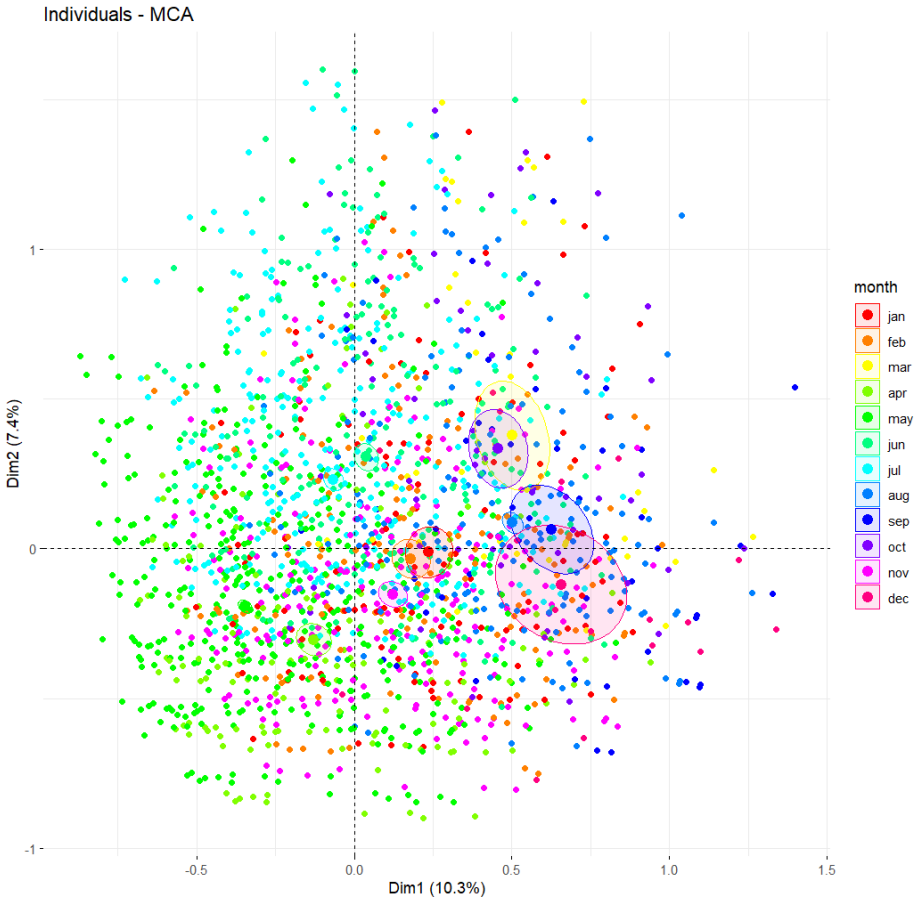

Contribution of variables to Dim-4
Contributions (%)
7.5
5.0
2.5
0.0
jul
education_secondary
education_primary
failure
loan_yes
job_technician
job_blue-collar
job_services
job_management
aug

For the first two plots, they represent the space after the MCA transformations for the first and second dimensions or third and fourth dimensions respectively. The position of each value shows its correlation with that dimension and the color shows its contribution for those two dimensions. With this plot we can see which variables may have a bigger leverage (with the color) and if their modalities are very similar (close to one another) and thus not very informative or more distinct (further away) meaning they may be useful categories. Because the contributions may not be fully clear, we also have a plot of the main contributors to each of the first four dimensions. Using this information, we can gather that the most relevant categories will end up being the education level, the job, if they have a housing loan, the previous outcome and maybe also the month, the marital status, and the contact method. If we plot the individuals coloured according to the different categories, we can check which of the factors (most likely the ones with higher contributions) may be the most useful later on.



The most notable ones are housing and education that show a clear distinction between the individuals of different modalities, then marital and poutcome to a lesser degree, and finally, for the plots of job and month, the data is so cluttered that we'll need a better view.

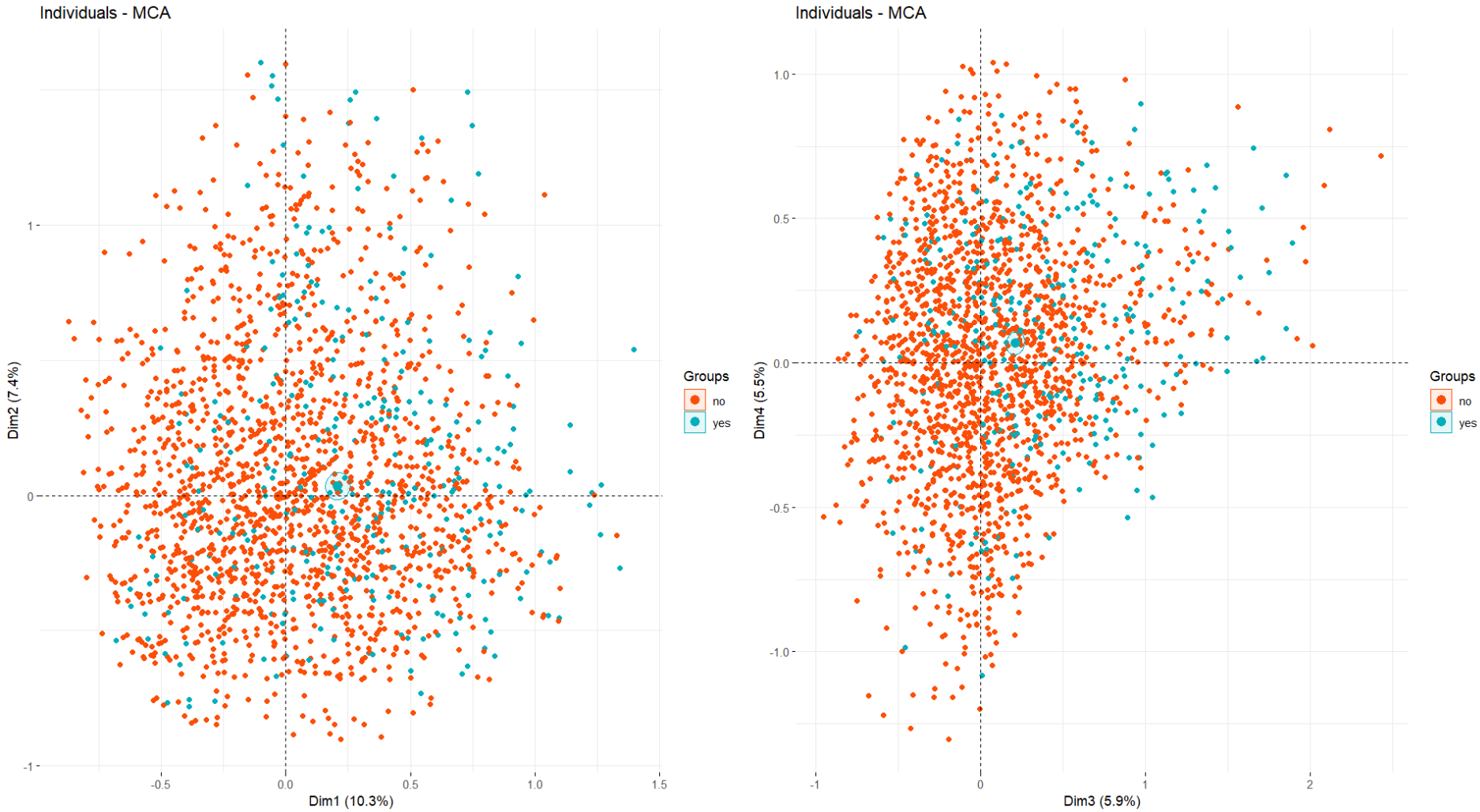

Individuals - MCA
Dim2 (7.4%)
Dim1 (10.3%)
job
job_admin.
job_blue-collar
job_entrepreneur
job_housemaid
job_management
job_retired
job_self-employed
job_services
job_student
job_technician
job_unemployed
job_unknown


Individuals - MCA
Dim2 (7.4%)
Dim1 (10.3%)
month
jan
feb
mar
apr
may
jun
jul
aug
sep
oct
nov
dec

For education, we see the clear difference according to the level with education increasing from the second quadrant towards the fourth. In housing loans, we see also the difference with no loans on the first quadrant and instead on the third. For jobs we see some of them are more remarkable: blue-collar, housemaid, retired, unknown, student, admin, and services. So, for this factor, if need be, it could be considered to join the rest of modalities. Regarding months, we have from April to July (spring to summer) on the left side, while on the right, the rest of them are mixed.

For the marital status we see again only single people are somewhat separate from the rest, benign more prominent on the third and fourth quadrants. Finally, the outcome of the previous campaign, projected on the third and fourth axis as that's where it has relatively the biggest relevance, shows three distinct (but still overlapping) groups. For the not contacted slightly towards the third quadrant, for the successful on the first and the other two together between the first and second quadrant.

To end the MCA we have to see how these new dimensions relate to the response variable. If we plot the samples along the four first dimensions, we see that the modalities for the variables are overlapping significantly, so the factors alone probably won't solve the problem alone. We also see that the two centroids are only separated in the first and third dimensions. This means that the people more likely to subscribe are of higher education, without house loans, with jobs either unknown or still students instead of blue-collar or services, not by the end of spring and that already said yes to the previous campaign.

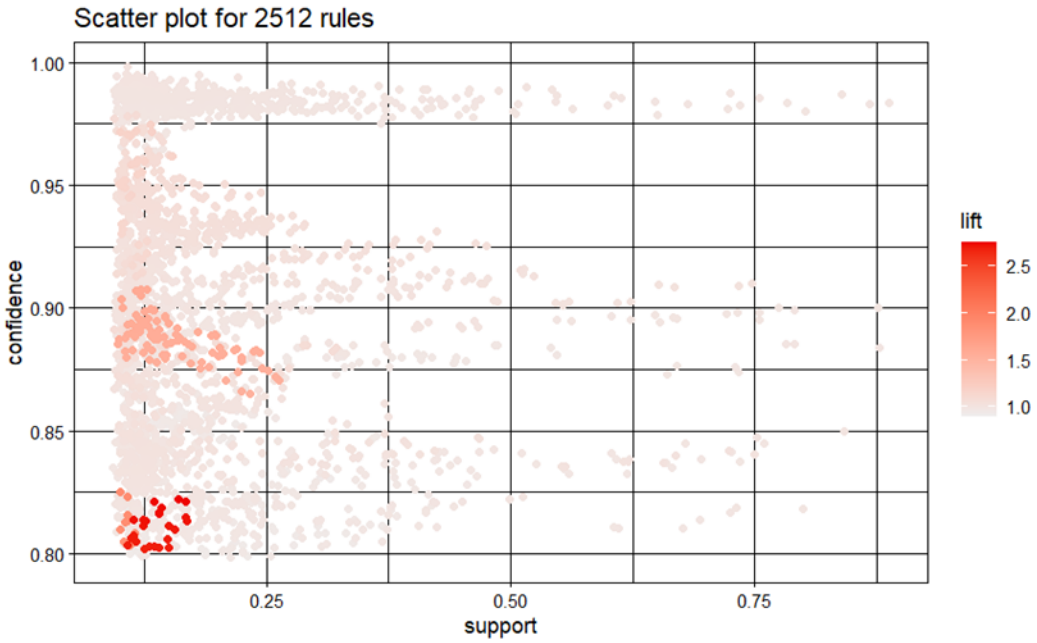

Individuals - MCA
Dim2 (7.4%)
Dim1 (10.3%)
Groups
no
yes
Individuals - MCA
Dim4 (5.5%)
Dim3 (5.9%)
Groups
no
yes

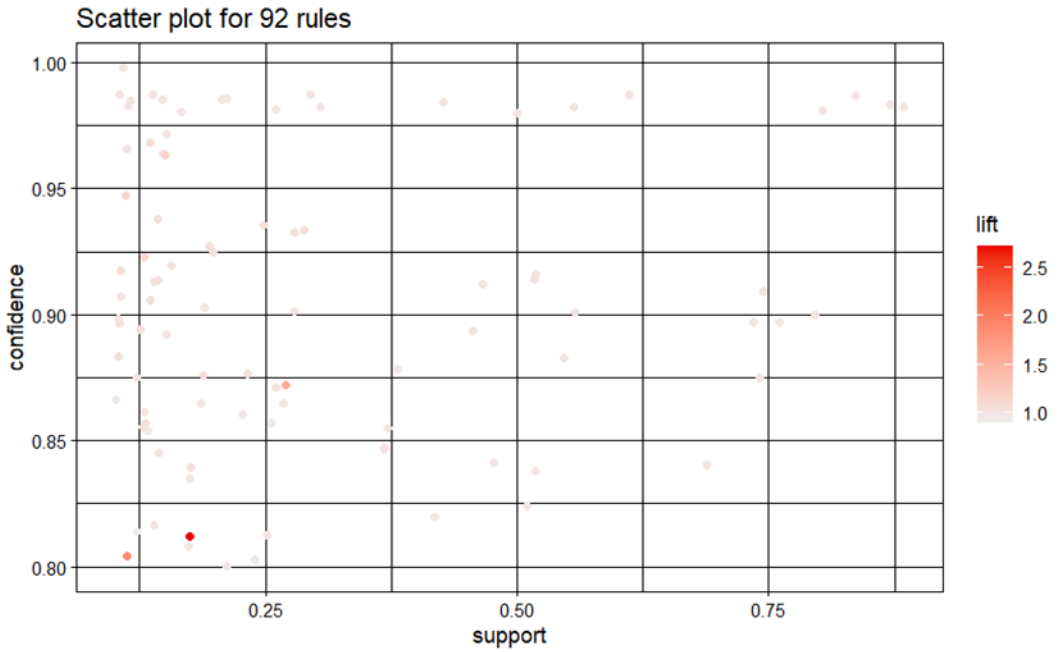
6. ASSOCIATION RULES

The association rule finds statistical association between categorical features in a dataset. To be able to find those in the project dataset we first convert all categorical columns to factors. Then, the apriori algorithm is used to identify all item sets that occur over a limit. The dataset is pre-processed, converted into a transaction format, and rules are mined with varying support and confidence thresholds.

To better know the rules in our dataset, we created a function that checks all combinations of support and confidence from 0.1 to 1 while storing the lift value. Then, we keep the rules that have a lift of over 1. Redundant rules are identified and removed to ensure a concise set of association rules. The pruned rules are sorted, leaving the rules over a lift of 1 and confidence over 0.8.

In the following plot, we visualise the representation of association rules based on support and lift. This visualisation captures the rules both before and after the pruning process.





And these are the top 20 results for the association rules, chose with the final combination that gives the best lift values (>1.0): in combination with the confidence (0.8):

	lhs <chr>	<chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>
[1]	{job=management}	=>	{education=tertiary}	0.1740765	0.8121775	0.2143331	2.7198922
[2]	{month=aug}	=>	{housing=no}	0.1125857	0.8041074	0.1400133	1.8528897
[3]	{month=may}	=>	{housing=yes}	0.2696306	0.8719599	0.3092236	1.5404966
[4]	{month=jul}	=>	{poutcome=none}	0.1504092	0.9631728	0.1561601	1.1753048
[5]	{month=jun}	=>	{poutcome=none}	0.1112586	0.9472693	0.1174519	1.1558987
[6]	{month=aug}	=>	{poutcome=none}	0.1291750	0.9225908	0.1400133	1.1257849
[7]	{y=yes}	=>	{loan=no}	0.1057288	0.9174664	0.1152400	1.0829936
[8]	{month=aug}	=>	{contact=cellular}	0.1355895	0.9684044	0.1400133	1.0765076
[9]	{loan=yes}	=>	{y=no}	0.1433311	0.9377713	0.1528423	1.0599161
[10]	{month=aug}	=>	{loan=no}	0.1251935	0.8941548	0.1400133	1.0554762

	lhs <chr>	<chr>	rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>
[11]	{month=may}	=>	{y=no}	0.2886530	0.9334764	0.3092236	1.0550617
[12]	{job=blue-collar}	=>	{y=no}	0.1939836	0.9270613	0.2092457	1.0478110
[13]	{loan=yes}	=>	{poutcome=none}	0.1309445	0.8567294	0.1528423	1.0454180
[14]	{education=primary}	=>	{poutcome=none}	0.1282902	0.8554572	0.1499668	1.0438656
[15]	{month=jun}	=>	{loan=no}	0.1037381	0.8832392	0.1174519	1.0425912
[16]	{marital=single}	=>	{contact=cellular}	0.2475116	0.9356187	0.2645432	1.0400620
[17]	{education=tertiary}	=>	{contact=cellular}	0.2784782	0.9325926	0.2986065	1.0366981
[18]	{marital=single}	=>	{loan=no}	0.2318071	0.8762542	0.2645432	1.0343460
[19]	{job=management}	=>	{loan=no}	0.1877903	0.8761610	0.2143331	1.0342360
[20]	{housing=no}	=>	{poutcome=none}	0.3676178	0.8470948	0.4339748	1.0336614

With these rules, we see that in rule number 18, the algorithm associates that the marital status is single to not having a loan, and in the number 1 associates the management job with a tertiary education. Despite these rules having a lot of sense, they are not very insightful of our target column: y.

To be able to look at that, we modify the initial function to filter by the column y. The results, when targeting y=yes (rhs) however, are not relevant enough for the final study, since the rules that have lift greater than one has a confidence lower than 0.2:

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>
[1]	{loan=no}	=>	{y=yes}	0.1057288	0.1248042	0.8471577	1.0829936
[2]	{default=no}	=>	{y=yes}	0.1132493	0.1151856	0.9831896	0.9995280
[3]	{contact=cellular}	=>	{y=yes}	0.1035169	0.1150725	0.8995797	0.9985469

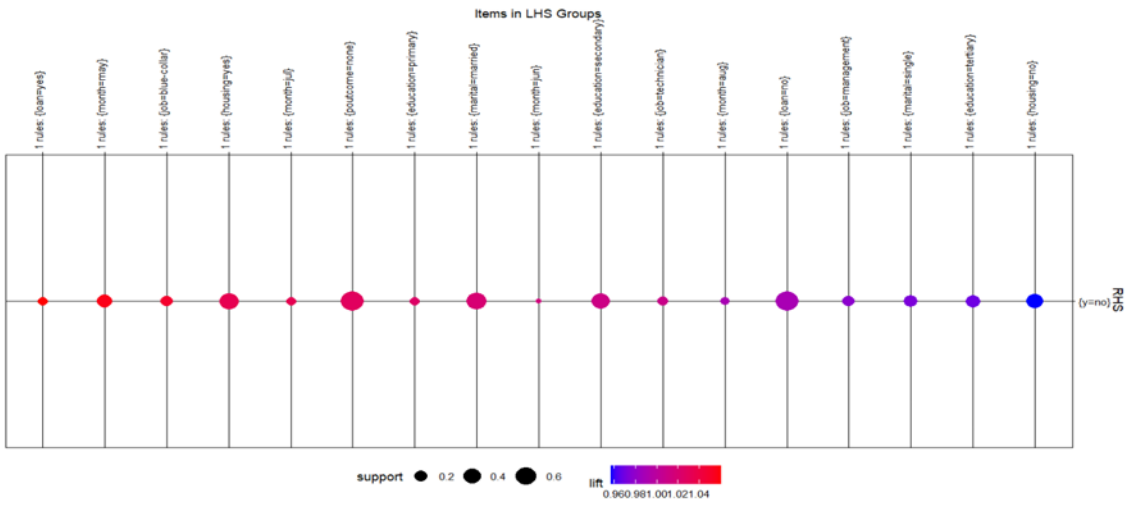
3 rows

Comparing with the rules that target y=no we see that the confidence is greater, and we have more rules applicable for the column. This is due to the fact of an unbalanced dataset, which results in predicting better when the answer to the target column has the greatest number of values in the actual dataset.

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>
[1]	{loan=yes}	=>	{y=no}	0.1433311	0.9377713	0.1528423	1.0599161
[2]	{month=may}	=>	{y=no}	0.2886530	0.9334764	0.3092236	1.0550617
[3]	{job=blue-collar}	=>	{y=no}	0.1939836	0.9270613	0.2092457	1.0478110
[4]	{housing=yes}	=>	{y=no}	0.5173634	0.9140289	0.5660252	1.0330812
[5]	{month=jul}	=>	{y=no}	0.1426676	0.9135977	0.1561601	1.0325938
[6]	{poutcome=none}	=>	{y=no}	0.7449679	0.9090418	0.8195090	1.0274445
[7]	{education=primary}	=>	{y=no}	0.1358107	0.9056047	0.1499668	1.0235597
[8]	{marital=married}	=>	{y=no}	0.5573988	0.9009653	0.6186684	1.0183161
[9]	{month=jun}	=>	{y=no}	0.1052864	0.8964218	0.1174519	1.0131808
[10]	{education=secondary}	=>	{y=no}	0.4558726	0.8937554	0.5100641	1.0101671

With that we conclude, for example, that having a loan is the most significant variable associated with the final objective of the research, that appears in both y=yes and y=no. We also see that for the months of May, June, July and may also have a tendency of y=no.

The following plot visually represents association rules that involve specific conditions in the dataset. We selected the rules: y=no on the right side and plotted the rules associated with it grouped by method, where the colour is determined by the lift value.



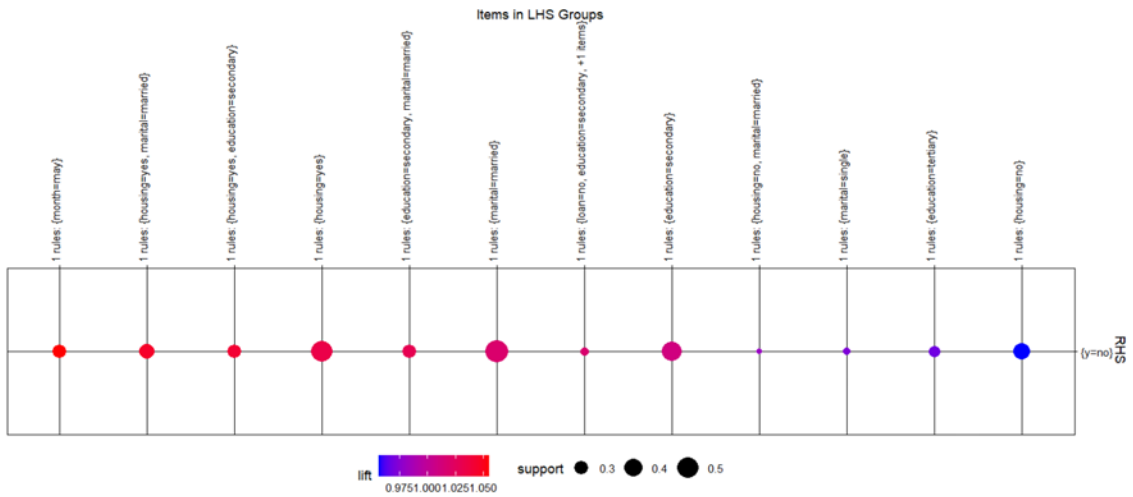
Finally, we leveraged the power of the ECLAT (Equivalence Class Transformation) algorithm to uncover frequent itemsets within our dataset. Similar to the apriori algorithm, ECLAT is a popular method for association rule mining, specifically designed for vertical data formats. To examine the itemsets we follow the same parameters as the ones done previously and get the following results.

	lhs <chr>		rhs <chr>
[1]	{contact=cellular, month=may, poutcome=none}	=>	{housing=yes}
[2]	{month=may, poutcome=none, y=no}	=>	{housing=yes}
[3]	{contact=cellular, month=may, y=no}	=>	{housing=yes}
[4]	{contact=cellular, month=may}	=>	{housing=yes}
[5]	{month=may, poutcome=none}	=>	{housing=yes}
[6]	{loan=no, contact=cellular, month=may}	=>	{housing=yes}
[7]	{month=may, y=no}	=>	{housing=yes}
[8]	{loan=no, month=may, y=no}	=>	{housing=yes}
[9]	{month=may}	=>	{housing=yes}
[10]	{loan=no, month=may}	=>	{housing=yes}
[11]	{housing=no, y=no}	=>	{poutcome=none}
[12]	{month=may}	=>	{y=no}
[13]	{marital=married, housing=yes}	=>	{y=no}
[14]	{education=secondary, housing=yes}	=>	{y=no}
[15]	{marital=single}	=>	{contact=cellular}
[16]	{marital=single, y=no}	=>	{contact=cellular}
[17]	{education=tertiary}	=>	{contact=cellular}
[18]	{education=tertiary, y=no}	=>	{contact=cellular}
[19]	{education=tertiary, poutcome=none}	=>	{contact=cellular}
[20]	{marital=single}	=>	{loan=no}

While both have the same goal, they do not deliver similar results, due to the different ways of generating the itemsets. A priori generates itemsets by joining and pruning candidate sets while Eclat generates by intersecting transactions containing each item.

Like apriori, when examining the 'y' column, it is noteworthy that the ECLAT library does not yield any associations with 'y=yes.' However, it does reveal associations with 'y=no,' as evidenced by the following associated values. These results are similar to the ones with the apriori results but obtain better lift values and confidence levels.

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>	lift <dbl>
[1]	{month=may}	=>	{y=no}	0.2886530	0.9334764	1.0550617
[2]	{marital=married, housing=yes}	=>	{y=no}	0.3324486	0.9249231	1.0453943
[3]	{education=secondary, housing=yes}	=>	{y=no}	0.2915284	0.9216783	1.0417269
[4]	{housing=yes}	=>	{y=no}	0.5173634	0.9140289	1.0330812
[5]	{marital=married, education=secondary}	=>	{y=no}	0.2875470	0.9110021	1.0296601
[6]	{marital=married}	=>	{y=no}	0.5573988	0.9009653	1.0183161
[7]	{marital=married, education=secondary, loan=no}	=>	{y=no}	0.2298164	0.9003466	1.0176168
[8]	{education=secondary}	=>	{y=no}	0.4558726	0.8937554	1.0101671
[9]	{marital=married, housing=no}	=>	{y=no}	0.2249502	0.8677474	0.9807715
[10]	{marital=single}	=>	{y=no}	0.2276045	0.8603679	0.9724308



7. HIERARCHICAL CLUSTERING

For this section we use all the variables, both numerical and categorical, by converting the categorical to factors. So, our variables are: "age" , "job", "marital" , "education", "default", "balance" "housing", "loan", "contact", "day", "month", "duration", "campaign", "previous" , "poutcome", "y", with “y” being the response variable.

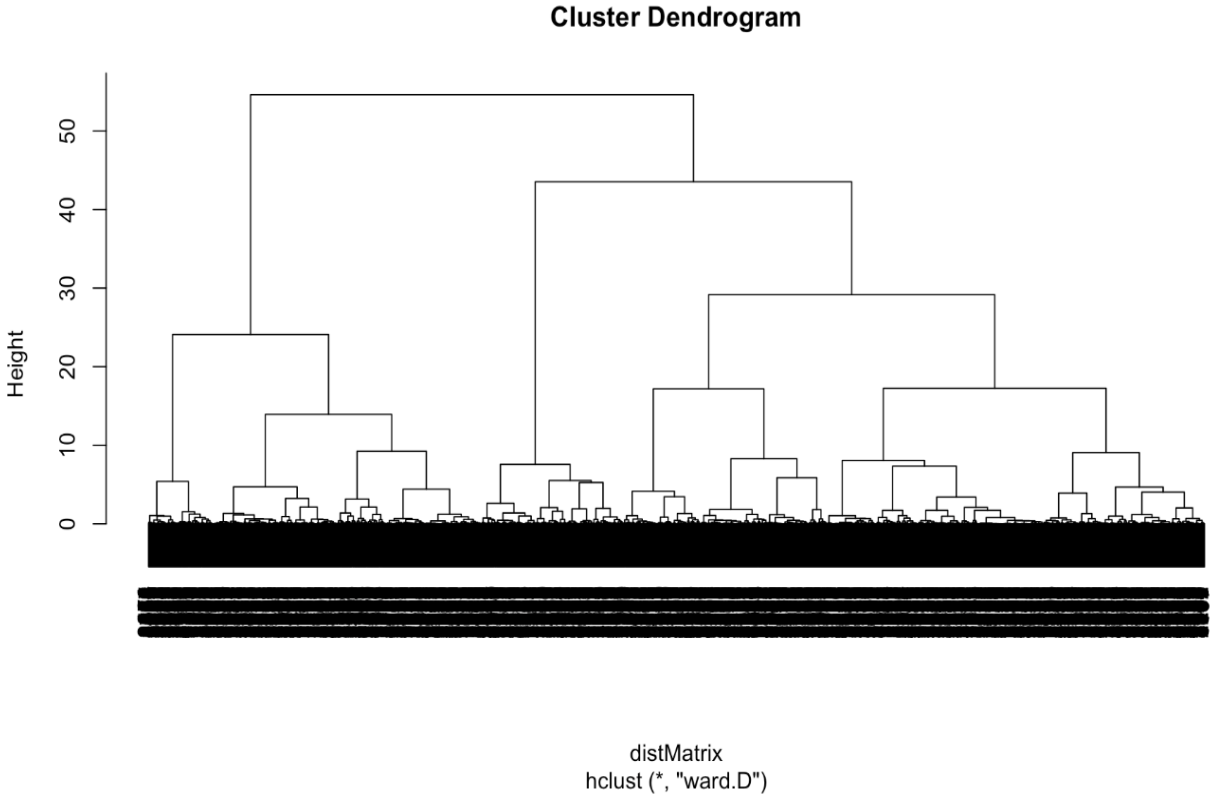
In the hierarchical clustering analysis, we applied Ward's minimum variance method to our pre-standardized dataset to identify the natural groupings within the data. A dissimilarity matrix was first computed using Gower's coefficient, which caters to mixed data types, and then squared to serve as the input for the clustering algorithm.

The dendrogram produced from this analysis visualizes the data's hierarchical structure, with the height of the merges indicating the level of dissimilarity. The choice of the number of clusters is somewhat subjective, as it can depend on the specific context and the granularity of clustering that is desired.

Subsequently, we partitioned the hierarchical tree into five distinct clusters, revealing the composition of our dataset: 1152 observations in Cluster 1, 610 in Cluster 2, 1614 in Cluster 3, 871 in Cluster 4, and 274 in Cluster 5. This clustering has provided a quantitative insight into the dataset's structure, which will underpin further analysis and interpretation of the data characteristics within each group. We think that merging any further (3 clusters) would involve combining clusters that are quite dissimilar (as indicated by the height of the merge on the dendrogram), which can be counterproductive to the clustering goal. Also, going further down the dendrogram (choosing 7 or 9 clusters) would result in some clusters being insignificant. After the profiling of 5 clusters, we concluded that the distribution of data into clusters and the profiling of them was adequate, providing a significant outcome. If we were about to use only numerical data, we could evaluate the clusters, by checking the total inertia gain, with functions like hcpc from FactoMineR, but since we wanted to use categorical data as well that does not suit us.

Table of cluster sizes

1	2	3	4	5
1152	610	1614	871	274



8. PROFILING OF CLUSTER

In our profiling of clusters analysis, we introduced two functions for significance testing. The ValorTestXnum function calculates p-values to assess whether numerical variables differ significantly across clusters. The ValorTestXquali function examines categorical variables for significant associations with clusters. Applying these functions will identify variables that are statistically significant in distinguishing between clusters, critical for understanding cluster characteristics and differences.

Next in our analysis we include ValorTestXnum and ValorTestXquali to evaluate statistical significance across clusters. ValorTestXnum is applied to numerical variables, this function calculates p-values using the Student's t-distribution to assess if the mean values of these variables significantly differ among clusters. ValorTestXquali is for categorical variables. It uses a contingency table approach to determine if their distribution varies significantly across clusters, deriving p-values from z-scores. The dataset is segmented into 5 distinct clusters.

Numerical Variables Analysis:

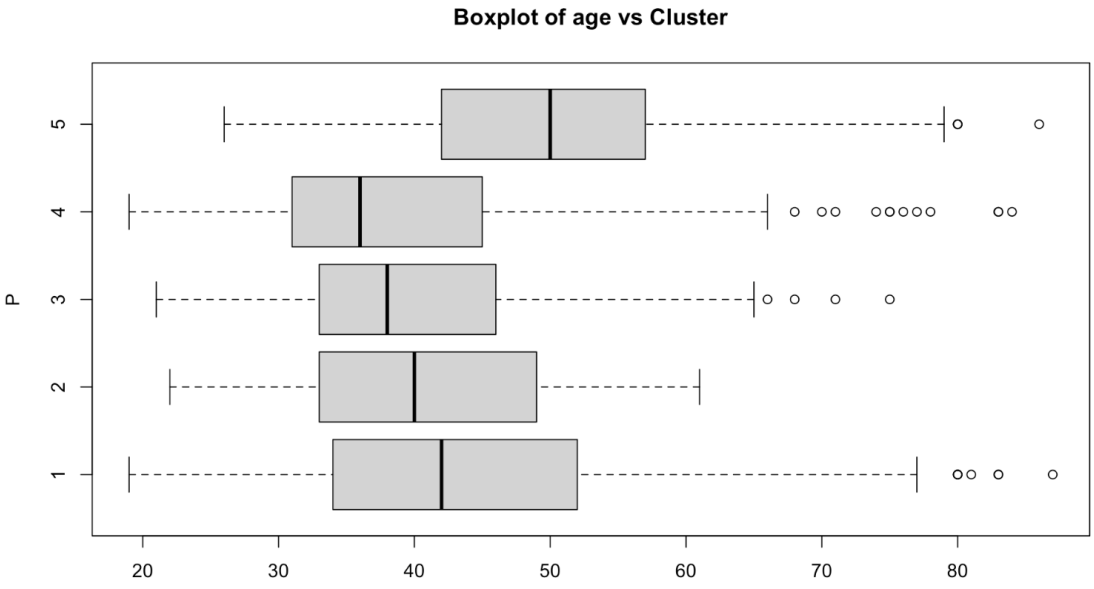
- The age, balance, duration, day, previous, and campaign variables are analyzed for their distribution across clusters.
- For each of these numerical variables, the script performs a one-way ANOVA test (o<-oneway.test) and a Kruskal-Wallis test (kw<-kruskal.test) to check if their distribution significantly varies across clusters.
- The very low p-values (e.g., 4.12e-60 for ANOVA on age) suggest strong evidence of differences in these variables across clusters.
- Specific p-values for each cluster is calculated using ValorTestXnum function.

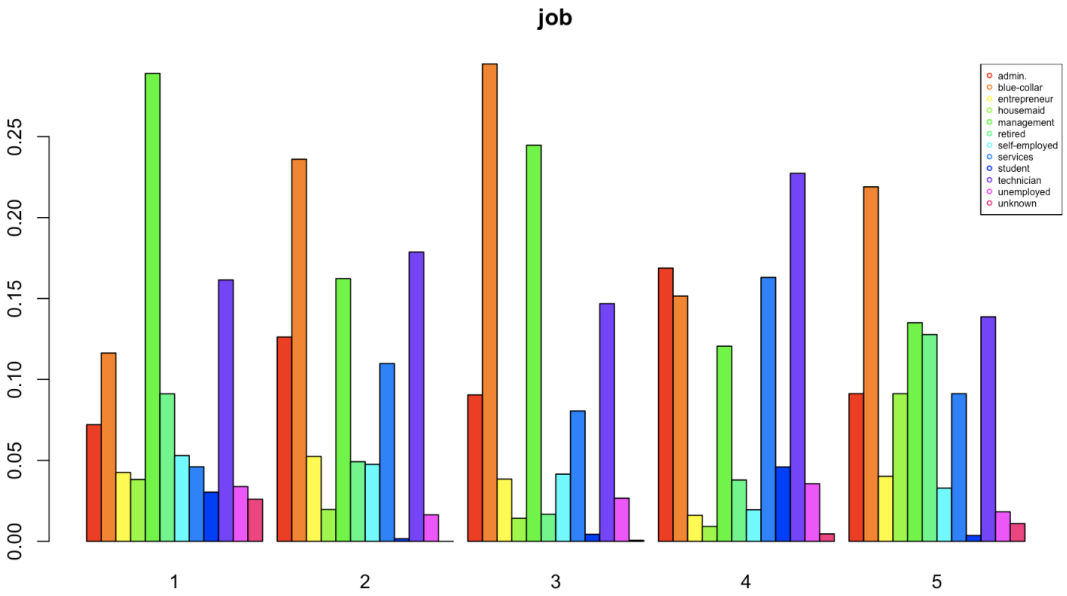
Categorical Variables Analysis:

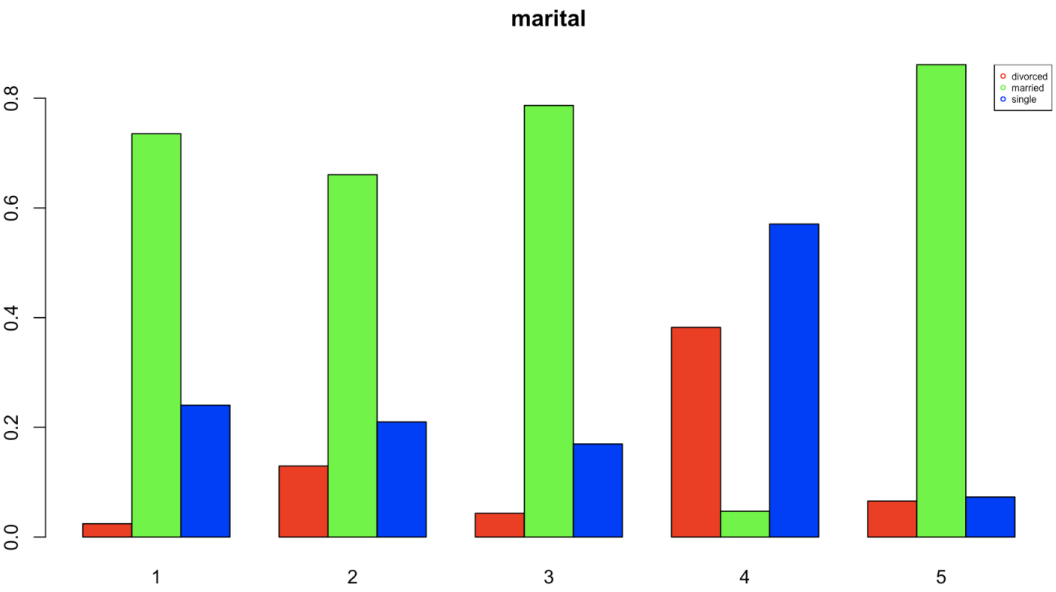
- Variables such as job, marital, education, default, housing, loan, contact, and y are analyzed for their distribution across clusters.
- Chi-squared tests are performed to see if the distribution of these categorical variables significantly differs among clusters.
- Extremely low p-values (e.g., <2e-16 for job) indicate significant differences in the distribution of these categorical variables across clusters.
- For each categorical variable and cluster, specific p-values are calculated using the ValorTestXquali function.

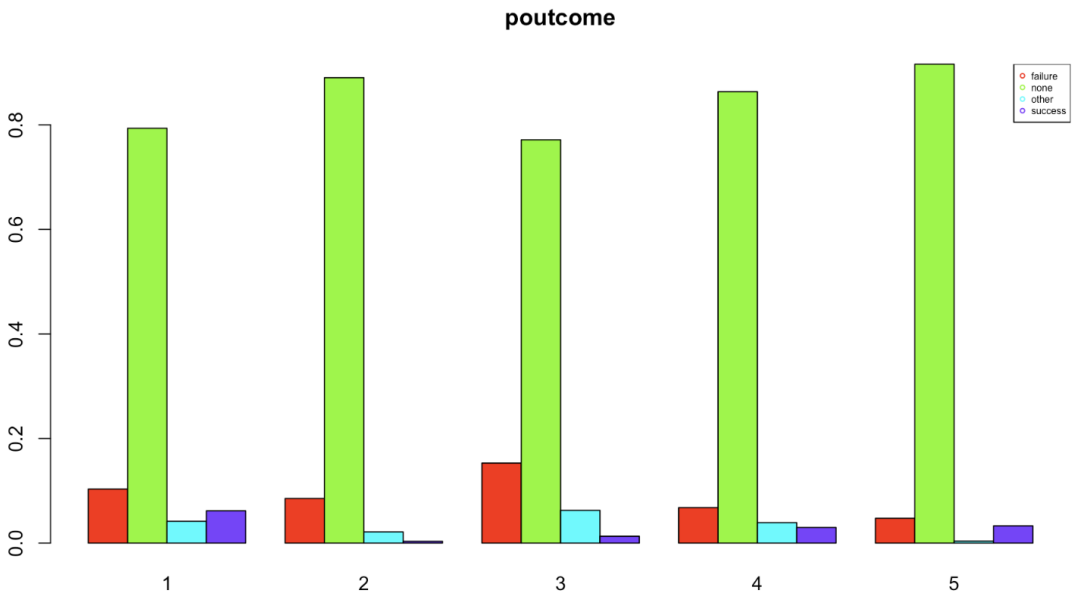
Significance in Clusters:

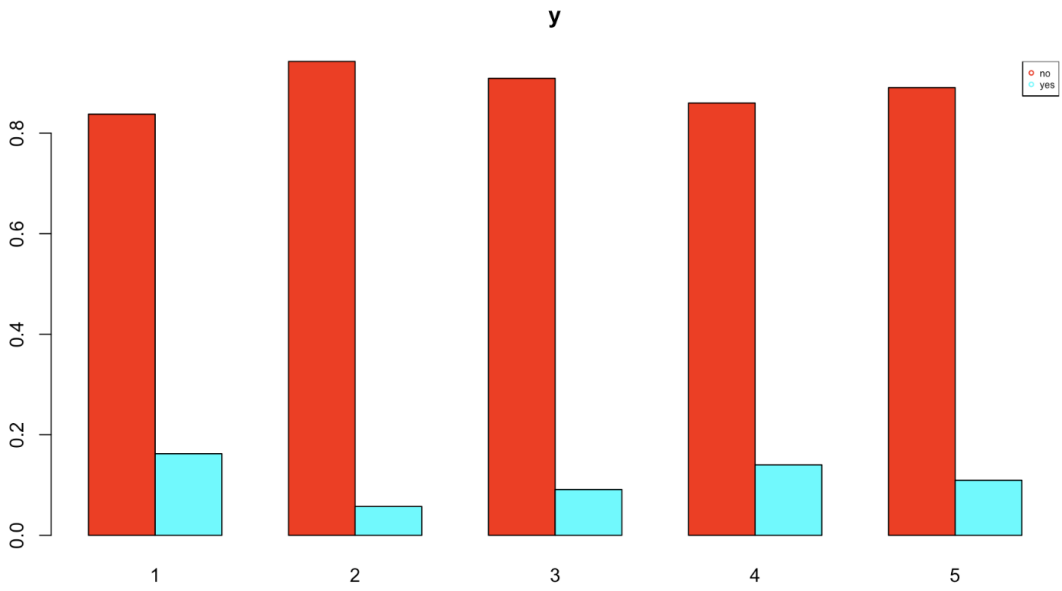
- The final output lists p-values for each variable within each cluster, providing a view of which variables are most significant in distinguishing each cluster.
- The sorted p-values for each cluster show which variables have the most significant differences. For instance, in Cluster 1, variables like job, marital, and education have p-values of 0.00e+00, indicating they are highly significant in defining this cluster.




job
admin.
blue-collar
entrepreneur
housemaid
management
retired
self-employed
services
student
technician
unemployed
unknown
0.00
0.05
0.10
0.15
0.20
0.25
1
2
3
4
5


marital
divorced
married
single
0.0
0.2
0.4
0.6
0.8
1
2
3
4
5





9. DECISION TREE

Decision Trees

The objective of this section is to train a decision tree capable of predicting the 'y' variable given an observation. The library used for this is 'rpart' and the approaches used to find the best tree for our data are the following:

1. Limiting the tree apriori with the 'minsplit', 'minbucket' and 'maxdepth' parameters of 'rpart'
2. Limiting the tree, a posteriori by pruning according to the complexity parameter

In our data, ‘y’ is the target variable, which indicates if the client has subscribed a term deposit or not. However, our data is unbalanced due to ‘y’ being a ‘no’ 4000 times and ‘yes’ only 521 times. In order for decision trees to be used we have to balance our data by either applying weights to the samples or undersampling and/or oversampling. To decide which of these is better in our case we constructed multiple trees with both strategies and compared their results. Although in the best case both can deliver a very similar model, the over and under sampling (using the ‘ROSE’ package) provides more consistent results.

With our strategy decided, we firstly split the data in the train set (70%) and the test set (30%). Then, we use the ‘ROSE’ package to oversample the minority class (‘yes’) and undersample the majority class (‘no’) in the train set. The over and under sampling is set to achieve a target proportion of 50% between ‘yes’ and ‘no’. The train dataset before ‘ROSE’ had 2805 ‘no’ and 376 ‘yes’, and after applying ‘ROSE’ to balance it, it had 1588 ‘no’ and 1593 ‘yes’.

For our first strategy, instead of just using its default parameters, a grid search for minsplit, minbucket and maxdepth parameters takes place with the help of ‘rpart.control’. The below combinations for these 3 parameters were tested: minsplit_values <- c(3, 5, 7, 10, 12, 15, 20, 30, 50, 100), minbucket_values <- c(3, 5, 7, 10, 12, 15, 20, 30, 50, 100), maxdepth_values <- c(3, 5, 7, 10, 12, 15, 20, 30). For each different combination of these 3 parameters, a confusion matrix is used to evaluate the model. More specifically, ‘Accuracy’, ‘Sensitivity’, ‘Specificity’, ‘Pos Pred Value’ and ‘Neg Pred Value’ are kept. The results are ordered by accuracy and sensitivity and the evaluation is also done in the training set, so it is compared with the testing, in order to check for overfitting.

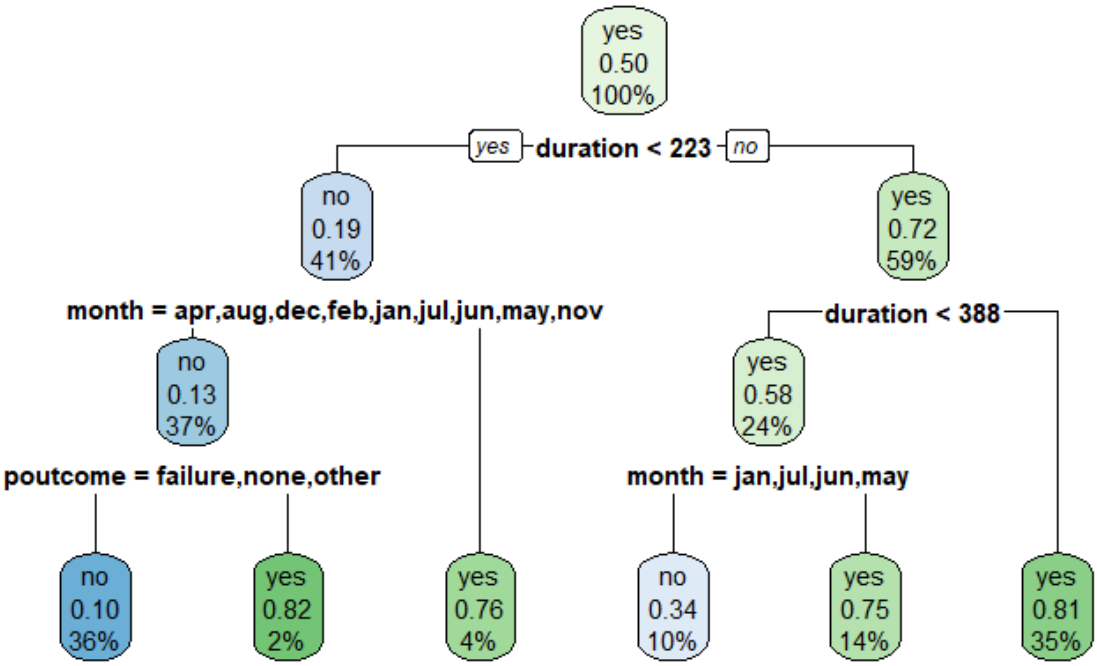
It seems that most of the parameters combinations have similar results with many of them having the best one. The best result can be seen with maxdepth=5, minsplit=3 and minbucket=3, but it also can be seen with minbucket changed to 5 or 7. There are many combinations with the same outcome. A possible reason for many combinations having the same result could be the oversampling impact. The use of oversampling techniques, such as those provided by the ROSE package, might influence the model in a way that makes it less sensitive to changes in the specified parameters. Oversampling can introduce synthetic data points, and the algorithm may find similar decision boundaries for different parameter values.

The metrics for the best model are: Accuracy 0.761194 / Sensitivity 0.8551724 / Specificity 0.7497908 / Pos Pred Value 0.2931442 / Neg Pred Value 0.9770992. Interpretation of these results: The model has a reasonably high accuracy of 76.12%, meaning it correctly predicts the outcome for about three-fourths of the instances. The sensitivity of 85.52% indicates that the model is good at capturing instances of the positive class, suggesting that it has a relatively low false negative rate.The specificity of 74.98% suggests that the model is reasonably good at correctly identifying instances of the negative class. The positive predictive value (precision) of 29.31% indicates that among the instances predicted as positive, only about 29.31% are true positives. This suggests that the positive predictions should be interpreted with caution.The negative predictive value of 97.71% is high, indicating that the model is very good at correctly identifying instances of the negative class.

The training results are presented for comparisons: Accuracy 0.8277271 / Sensitivity 0.8926554 / Specificity 0.7625945 / Pos Pred Value 0.7904391 / Neg Pred Value 0.8762663. Although the scores for accuracy, sensitivity and specificity are similar (or slightly higher than those in the testing set), we see a considerable difference in the

positive predicted value dropping from 79.04% in the training environment to 29.31% in the testing environment, indicating a possible overfitting in the positive class.

Now the resulting tree has to be explored.



The leaves respectively indicate that:

- The first leaf (36% of the balanced cases) are observations which have duration lower than 223 seconds, not in March, September or October and the previous outcome was not successful. From those, 10% are predicted to have subscribed.
- The second leaf (2% of the balanced cases) are observations which have duration lower than 223 seconds, not in March, September or October and the previous outcome was successful. From those, 82% are predicted to have subscribed.
- The third leaf (4% of the balanced cases) are observations which have duration lower than 223 seconds in March, September or October. From those, 76% are predicted to have subscribed.
- The fourth leaf (10% of the balanced cases) are observations which have duration between 222 and 338 seconds and the month is January, July, June or may. From those, 34% are predicted to have subscribed.
- The fifth leaf (14% of the balanced cases) are observations which have duration between 222 and 338 seconds and the month is not January, July, June or may. From those, 75% are predicted to have subscribed.
- The sixth leaf (35% of the balanced cases) are observations which have duration longer than 338 seconds. From those, 81% are predicted to have subscribed.

As a reminder the duration is the duration of last contact in seconds, month is the month of the last contact and poutcome is the outcome of the previous marketing campaign with this client. Only these 3 variables end up relevant in the decision tree and all of them have to do with the last communication between the bank and the client.

Insights into the tree:

- Clients with shorter duration interactions (less than 223 seconds) and specific months with poutcome being other but a success are less likely (10%) to subscribe to the term deposit. This could indicate that quick interactions with specific previous outcomes in these months are not strong predictors of positive outcomes.

- Longer interactions in specific months (jan, jul, jun, or may) are associated with a higher likelihood (34%) of clients subscribing, but still not favouring a new subscription.

- Longer interactions (223 to 388 secs) in a broader set of months (apr, aug, dec, feb, mar, nov, oct, or sep) are even more strongly associated with positive outcomes (75%). These months might indicate a more general trend where extended interactions tend to result in positive responses.

- Shorter interactions in specific months (mar, oct, or sep) have a relatively high likelihood (76%) of positive outcomes. This could suggest that quick engagements in these months are particularly effective in securing positive responses.

- Longer interactions, specifically those lasting 388 seconds or more, are strongly associated with positive outcomes (81%). Investing more time in interactions appears to be a key factor in achieving positive responses.

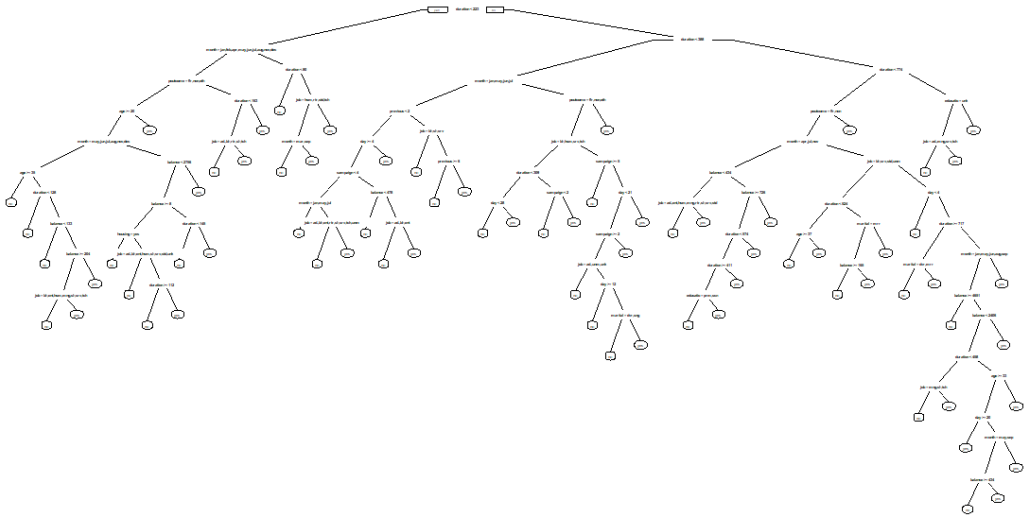
- Shorter interactions in specific months (apr, aug, dec, feb, jan, jul, jun, may, or nov) with a successful previous outcome have a very high likelihood (82%) of positive responses. Quick engagements with successful historical outcomes in these months are extremely promising for positive future outcomes.

So:

1. Timing Matters: Timing, as represented by the month of the last contact, plays a crucial role. Certain months are more favourable for successful marketing campaigns.
2. Previous Outcome Significance: The outcome of the previous marketing campaign (poutcome) is a significant predictor. A successful previous outcome strongly correlates with positive responses.
3. Investment in Longer Interactions: Longer interactions generally result in higher positive outcomes. Considering personalised and extended interactions with clients might enhance campaign effectiveness.
4. Quick Engagements with Success: Quick engagements in specific months with a history of success are highly indicative of positive outcomes. Focusing on these specific scenarios can be a strategic approach.

For our second strategy, instead of limiting the growth of the tree, we build it as big as possible (by setting the complexity parameter to zero) and then trim the resulting tree at the point where it had the best predictive value as calculated in its CP table. As the measure of the predictive value is not perfect (it has some noise) we can either just take

the optimal point even if it's the best just because of the noise or the first that is within some deviation from the best (in our case 95% of error from the optimal value).



After building our tree we get that the lowest error is 0.2594458 with a standard deviation of 0.01192553 so we'll try cutting at the CP with the lowest error and at the biggest CP in a 95% interval of the optimal error. If we cut at the optimal value (0.0005247691) we get a tree so big that trying to draw it makes everything illegible. On the other hand the results from our model are an improvement from before:

	Training	Testing
Accuracy	85.16%	80.00%
Sensitivity	91.49%	68.28%
Specificity	84.31%	81.42%
Positive Predictive Value	43.88%	30.84%
Negative Predictive Value	98.66%	95.49%

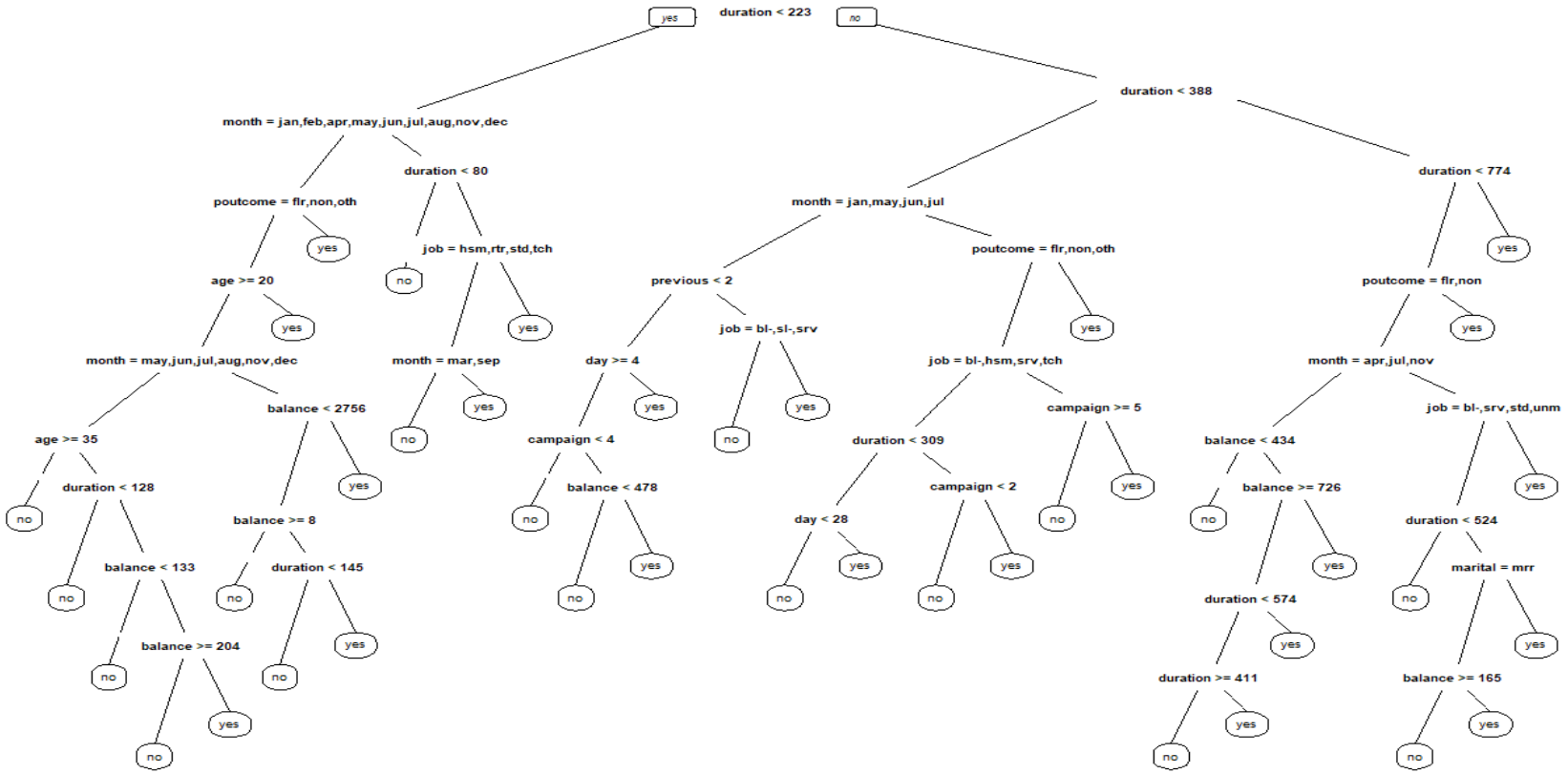
Additionally, we see that there isn't a drop as big in our indicators as in the first model, but with the drop we have in sensitivity and positive predictive value we may still be in front of an overfitting case.

To solve the size problem we can try the tree cut at the biggest CP with an error in the 95% interval of the best value (0.002518892). With this smaller tree we can now see what the conditions are at each split and that if we stopped at the second and third layer of the tree, we would get the first tree we got. To ensure this reduction in our trees size (and increase in readability) doesn't incur in a heavy loss in predictive power we take a look at how its results change from the complete one:

	Training	Testing
Accuracy	83.50%	80.45%
Sensitivity	90.43%	74.48%
Specificity	82.57%	81.17%
Positive Predictive Value	41.01%	32.43%
Negative Predictive Value	98.47%	96.33%

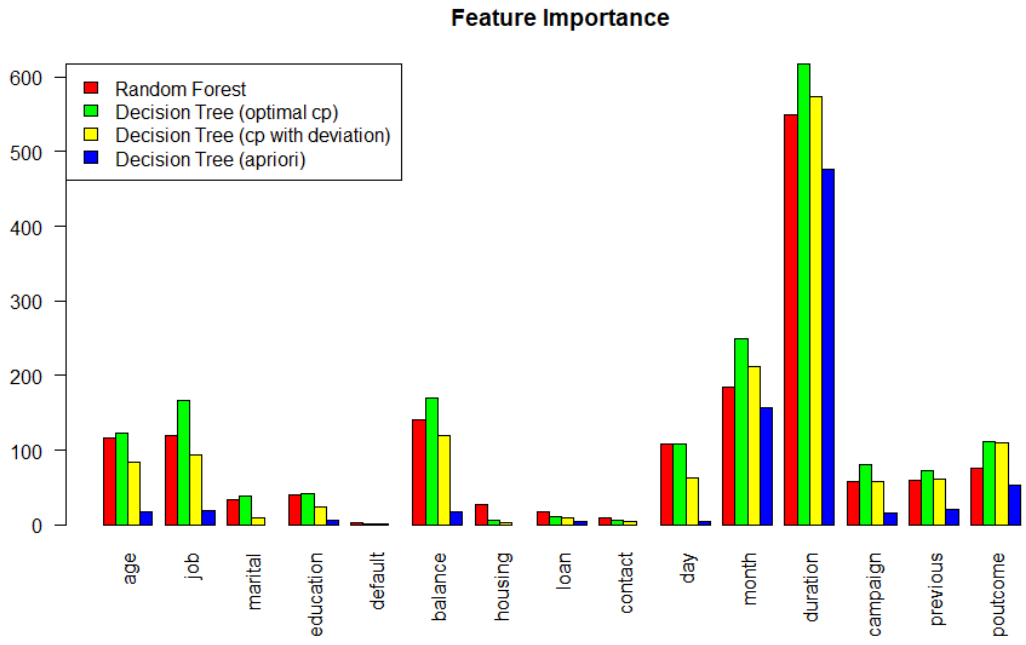
We continue to see a drop in both sensitivity and positive prediction, although a 4~8% smaller than the previous, and what is more, all indicators in the testing stay the same or improve, indicating that we aren't overfitting as much

Finally, as the positive predictive value of our trees is still considerably low (only one out of three of the customers predicted will subscribe) we also tried to use a random forest for the prediction which yielded the following results:



	Training	Testing
Accuracy	95.03%	88.81%
Sensitivity	99.47%	67.59%
Specificity	94.44%	91.38%
Positive Predictive Value	70.57%	48.76%
Negative Predictive Value	99.92%	95.87%

The random forest, similar to the big tree, ends up with a notable drop in both the sensitivity and the positive predictive value. To make a final comparison between the four models we made a plot of the feature importance to check what parameters weighed more when making the decisions. Which showed the peaks we expected from the first tree we made in duration, month and the previous outcome with four new peaks; age, job, balance and day; shared by only the other three models.



From these results, we can conclude that if the model we want is something we have to evaluate in the moment, the best one will be the first as it can be evaluated with a glance; but if we want something more robust, either use the third model (the a posteriori tree with the error in the 95%) or use a different type of models for the tasks.

10. DISCRIMINANT ANALYSIS

Linear Discriminant Analysis or LDA seeks to find a linear combination of features that best separates different classes while preserving the variance within each class. Unlike some other dimensionality reduction techniques, LDA explicitly considers the distribution of data for each class, making it particularly effective when dealing with classification tasks.

The fundamental premise of LDA involves maximizing the ratio of the between-class variance to the within-class variance. In other words, it aims to project the data into a lower-dimensional space where the distances between class centroids are maximized, and the spread of data points within each class is minimized.

To use this method, we used the following steps:

- Standardize the data of numerical variables, making it a standard normal distribution (mean at 0).

- Split the data between train and test, on 80/20 rate.

- Since LDA requires a balanced dataset, we used ROSE function to balance our target column y. Giving the following results:

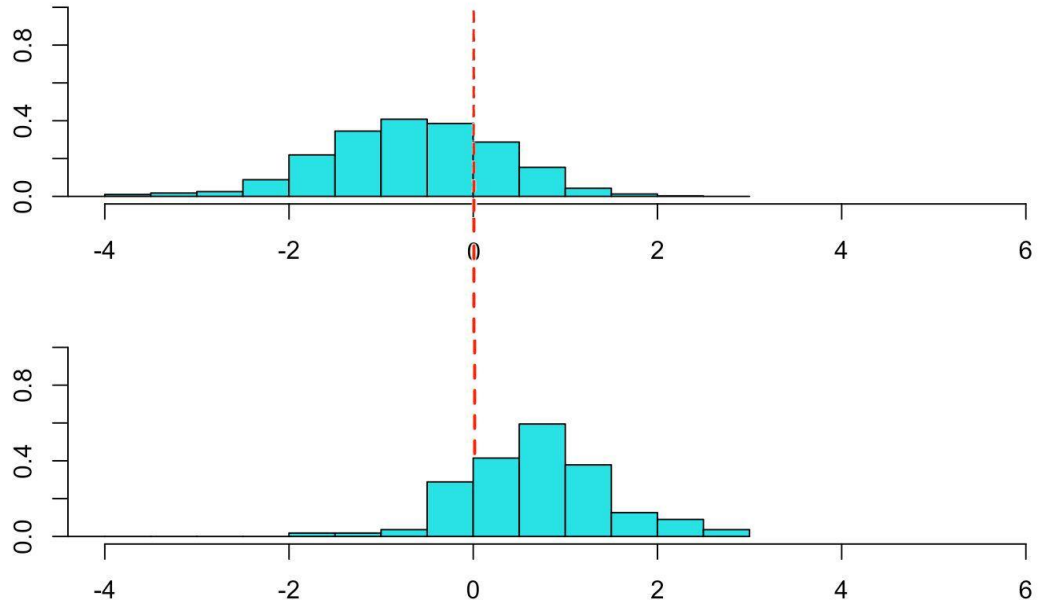
	Y=no	Y=yes
Before ROSE	3202	414
After ROSE	1809	1807

- After applying LDA to the pre-processed dataset, we calculate the coefficients of linear discriminants that represent the weight assigned to each feature in linear combination to discriminate between classes. With this, we conclude that the most relevant feature, with the biggest influence on prediction, is by far the duration of the call, followed by balance, previous, and campaign. This means that customers with higher balance who were contacted for previous campaigns with longer duration of the calls are more likely to have subscribed a term deposit. Furthermore, people who were not called as much for the active campaign and people contacted at the beginning of the month are also more likely to have a positive outcome.

```
Coefficients of linear discriminants:
                LD1
age        0.065315044
balance    0.278305164
day       -0.008557396
duration   0.948793119
campaign  -0.106302252
previous   0.134509721
```

Now we try to classify our outcome, first using rules and then using probabilities received from the LDA analysis.

For the approach with rules, we plot histograms of the linear discriminant when y=no and when y=yes. Then we try to find a vertical line that best separates the response groups. We did it by taking the mean of the medians of the train dataset when y=yes and when y=no. So, we create a vertical line (our rule) at that point of the discriminant, that will separate those two groups:



Following the graph above we conclude that the rule that separates y=no from y=yes is:

```
[1] "Median when y=no"
[1] -0.6588295
[1] "Median when y=yes"
[1] 0.6904398
[1] "Rule (mean of the medians)"
[1] 0.01580511
```

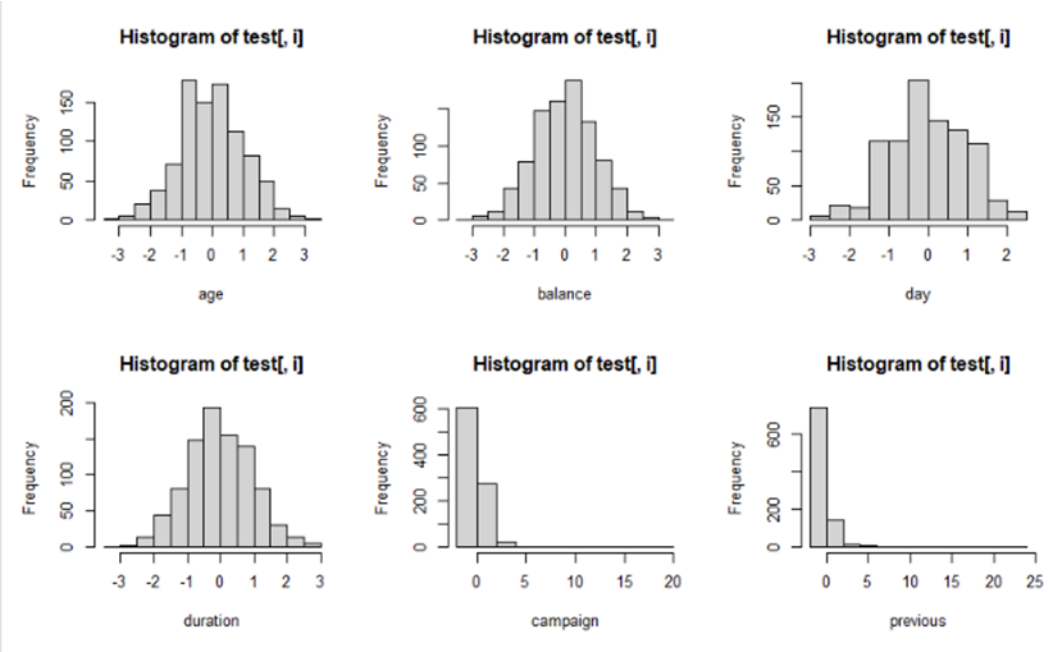
That means that any value is on the right of this line will mean that y=no, and everything on the left means that y=yes. After calculating the accuracy on the dataset se we that, in the confusion matrix we obtain a rate of 76.4% of accuracy.

```
     1   2
1 600 194
2  20  91
[1] 76.35359
```

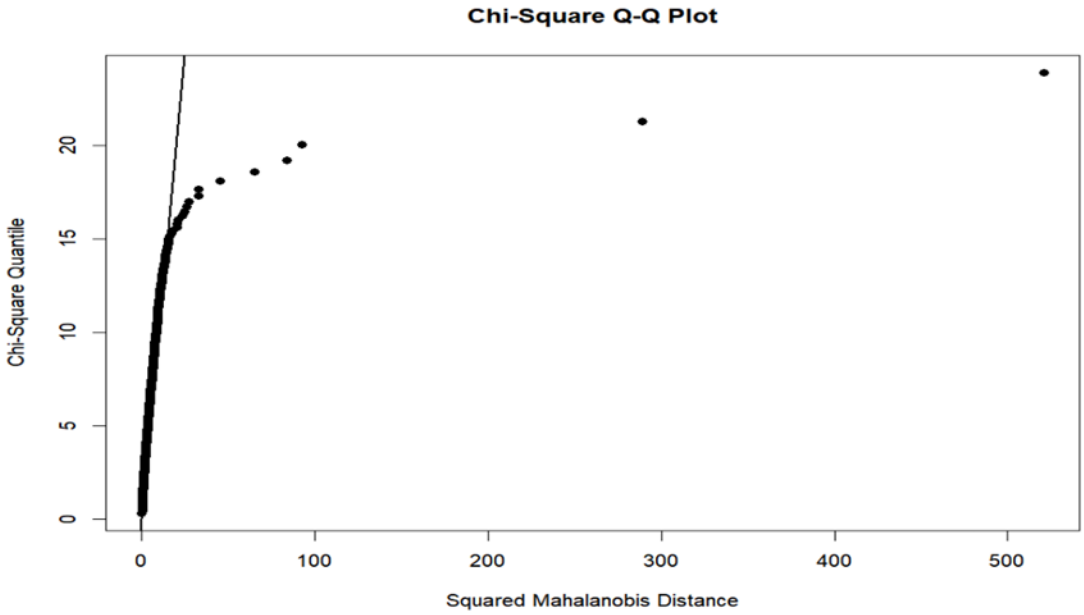
To compare to this prediction, we now evaluate the confusion matrix for LDA using probability.

For that we first test the 3 conditions for probability prediction:

- Univariate Distribution: We use the Shapiro-Wilk normality test on each column of the test dataset and create histograms to see the distribution of the values in each column. With that we obtain that the dataset is not normally distributed, as it is seen in the following plots:



- Multivariate Normality: We check for Multivariate Gaussian Conditions (Royston test and/or Henze-Zirkler test) and obtain the results that both p values for both tests are equal to zero. This test has also failed.



- Covariance Conditions: We perform a test of equality of covariance matrices for different groups within data. The result is that covariance is not equal between the yes and no groups. So, the test failed.

Despite failing all three tests we still calculate use probability, and we obtain the following results: accuracy of 76% and the following matrix:

```
      no yes
no   597 197
yes   20  91
[1] 76.0221
```

11. DISCUSSION AND CONCLUSION

The Bank Marketing dataset has the objective of predicting whether a client will subscribe to a term deposit. This classification task relies on various categorical features, such as the client's education level, occupation, and marital status. Additionally, numerical features like the client's age and average account balance are considered.

To evaluate this dataset, firstly we **pre-processed the data**, filling in the missing data using the MICE algorithm, that, compared with other methods, performed better, for each of the three variables we wanted to impute. As for the outliers, removing these outliers might result in losing insights about a crucial segment of the client base. As the dataset's objective is to predict term deposit subscription, understanding the behaviour of high-value clients might be vital, it has been decided to keep these in the dataset.

Principal Component Analysis (PCA) and **Multiple Correspondence Analysis (MCA)** conducted on the dataset have yielded valuable insights into various factors influencing client profiles. The key findings from PCA include distinctions in client profiles based on educational levels, employment status, months, marital status, and subscription status. MCA further highlighted notable variations in education, housing loans, jobs, months, marital status, and the outcome of the previous campaign as key variables.

The **association rules analysis** found statistical association between categorical features in a dataset, and its main objective was to find the association between the target column y with these variables. This study concluded that the main categorical features that influence the target column are the loan variable, together with the month and the job variable. This is coherent when comparing it with the previous analysis of PCA and MCA.

With **hierarchical clustering analysis**, we applied Ward's minimum variance method to our pre-standardized dataset to identify the natural groupings within the data. The number of clusters chosen is 5.

The **profiling of clusters** found significant differences across clusters. Numerical variables showed variance in ANOVA and Kruskal-Wallis tests, and categorical variables differed in Chi-squared tests. This resulted in: Cluster 1 with married professionals, Cluster 2 with unique financial traits, Cluster 3 with age-specific demographics, Cluster

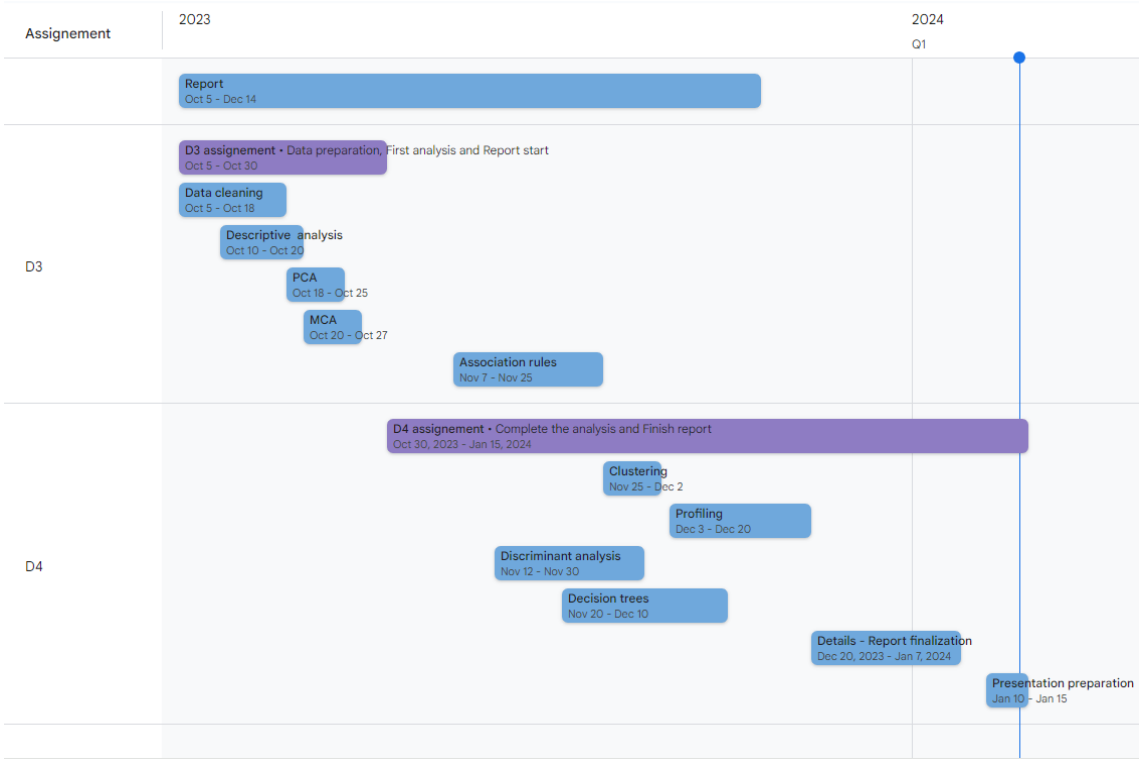
4 with a focus on higher education, and Cluster 5 characterized by specific loan behaviors.

Next, a **decision tree** capable of predicting the 'y' variable given an observation was created, with four different models. All models showed a peak in duration, month and the previous outcome as parameters weighed more when making the decisions.

Finally, a **discriminant analysis** was done to seek a linear combination of numerical features that best separates different classes while preserving the variance within each class. It has done so using two different methods, with similar accuracy of 76%. The results also show that the most relevant feature, with the biggest influence on prediction, is by far the duration of the call, followed by balance, previous, and campaign.

In conclusion, by combining different analytical methods, we gained a deeper understanding of the Bank Marketing dataset. The discovered patterns and significant variables provide valuable insights for making strategic decisions in marketing and client engagement. Our comprehensive analysis offers a complete view, helping stakeholders make well-informed decisions grounded in a nuanced understanding of client behaviour and predictive factors.

Final Executed Gantt



Risk management results

Regarding the risks involved in the production of this report, there have been only two. The first risk arose at the very start, when one of the members was absent for the second deliverable and two more weeks into the timetable of the third deliverable. We communicated the issue and reassigned the tasks in a way that the rest of the group

would handle the D2 tasks and the first tasks of the D3 and the absent member would compensate by having more tasks in late D3 and D4. The problem was handled, and all the deliverables were finished in time.

The second issue we have faced is the different order of materials covered in class opposed to the order in the original project statement. Since the initial task distribution was created with the timetables in order of the project statement, they have had to reorganize as the project went on. This way, the tasks were done in order of the material being covered in class. This issue did not pose a significant problem.

Task assignment grid

	Andreja A	David C	Nayara C	Julie O	Alex T
D1	x	**x**	x	x	x
Metadata				**x**	
Grantt diagram		**x**			
Task division	x	x	x	x	**x**
Risk management			**x**		x
Report	x	x	x	x	x
D3 assignement	x	x	x	x	x
Data cleaning				x	**x**
Descriptive analysis			**x**	x	
PCA	x	**x**	x		
MCA	x	x		**x**	
D4 assignement	x	x	x	x	x
Association rules			x		**x**
Clustering	**x**				x
Profiling		x		**x**	
Discriminant analysis	**x**		x		
Decision Trees		**x**			x
Presentation preparation	x	x	**x**	x	x