Study over diabetic inpatients readmission rates

Data Mining Course, GEI-2018, FIB, UPC

Kaggle's Diabetes 130 US hospitals for years 1999-2008
(https://www.kaggle.com/brandao/diabetes/home)

Marc Badia Romero
Aleix Balletbó Gregorio
Bernat Gené Skrabec
Víctor Giménez Ábalos
Guillem Ferrer Nicolás
Daniel Tarrés Amselem

Index

Motivation of the work and description of the problem

Studies have shown the relationship between the management of hyperglycemia in hospitals and the subsequent outcome of the process. We want to study whether we can discover underlying factors that could potentially discover failures in the established protocols, relationship between factors such as the potential trouble factors that could bring the patient to suffer from their conditions.

We want to make a study over a set of recorded variables, in order to understand those relationships and predict problems such as readmission rates, which we are interested in studying deeply.

More specifically, our data covers the admission of ‘diabetic’ inpatients, which includes attributes potentially associated with the diabetic condition. These attributes are demographics, diagnoses, diabetic medications administered, information about medical visits and laboratory tests, but only for patients whose stay spanned between 1 and 14 days (which is useful for discarding routinary procedures which we are not interested in).

The dataset was built with the objective of forecasting patient readmission (which could potentially lower the deaths of diabetic individuals over hyperglycemia). The information on whether an inpatient was readmitted and the period of time between the admissions is included as a feature in the dataset.

Data source

Our data source comes from Kaggle, a webpage that hosts many datasets. Specifically, when we talk about our data matrix we refer to the one downloaded from the following link: https://www.kaggle.com/brandao/diabetes/version/1
To obtain it simply download from the above webpage's download button. The data is stored as a zip containing both a description over the variables and a .csv file with the data.

As explained before, it is a dataset over the admission of 'diabetic' inpatients, whose stay spanned between 1 and 14 days, with variables over demographics, diagnoses, administered drugs, information about procedures, payment codes, weight, measurements such as glucose in blood, the specialty that treated them etc. But also a readmission variable, which has values: *No*, *< 30* and *> 30* depending on whether he was readmitted into a hospital and when.

Formal data description

Data matrix explanation

Our dataset is constituted by 101766 individuals,which are encounters as described above:

- It is a hospital admission
- Of diabetic inpatients (meaning any kind of diabetes was entered to the system as a diagnosis)
- Whose stay lasted between 1 and 14 days
- With laboratory tests performed during the encounter
- With drugs administered during the encounter

Each data row contains information relating to that encounter, including information about the patient (*id, race, gender, age*…), their admission (*type, disposition, time_in_hospital*…), their laboratory procedures, their different diagnoses of their first three visits to the hospital (also includes the total number of diagnoses of that patient), the medications administered to them (24 generic medications discerning if they were diabetic medications or not) and a column indicating if they were readmitted after or before 30 days, or not at all.

Metadata table

In this section we include two tables that show the metadata behind the variables in our dataset. The first one belongs to the raw data, as downloaded initially. The other one belongs to the preprocessed dataset and therefore the one we work upon. In the next section we discuss the changes between one and the other.

Raw metadata table

Variable	Modalities	Meaning	Type	Measuring unit	Missing code	Measuring procedure	Range	Role
encounter_id		Unique identifier of an encounter	Numeric		?			Explanatory
patient_nbr		Unique identifier of a patient	Numeric		?			Explanatory
race	Caucasian, Asian,	Patient's origin place	Qualitative		?			Explanatory

	African American, Hispanic, Other							
gender	Male, Female, Unknown/invalid	Patient's gender	Qualitative		?			Explanatory
age	[0, 10), [10, 20), … [90, 100), Other	Patient's age	Qualitative	Years	?			Explanatory
weight	[25, 50), [50, 75), … [125, 150), Other	Patient's weight	Qualitative	Pounds	?			Explanatory
admission_type_id	1,2,...,9	Identifier of the patient's admission type	Qualitative		6			Explanatory
discharge_disposition_id	1,2,...,29	Identifier of the patient's discharge type	Qualitative		18			Explanatory
admission_source_id	1,2,...,26	Identifier of the patient's admission source	Qualitative		17			Explanatory
time_in_hospital		Days between admission and discharge	Numeric	Days	?		1 to 14	Explanatory
payer_code	BC, CH, CM, CP, ... (23 values)	Code of the paying method	Qualitative		?			Explanatory
medial_speciality	Cardiology, Oncology, Podiatry, Pediatrics, ... (84 values)	Specialty of the admitting physician	Qualitative		?			Explanatory
num_lab_procedures		Number of lab tests performed during the encounter	Numeric		?		1 to 132	Explanatory
num_procedures		Number of procedures (other than lab tests) performed during the encounter	Numeric		?		0 to 6	Explanatory
num_medications		Number of distinct generic drugs administered during the encounter	Numeric		?		1 to 81	Explanatory
number_outpatient		Number of outpatient visits of the patient in the year preceding the encounter	Numeric		?		0 to 42	Explanatory

number_ emergency		Number of emergency visits of the patient in the year preceding the encounter	Numeric		?		0 to 76	Explanatory
number_ inpatient		Number of inpatient visits of the patient in the year preceding the encounter	Numeric		?		0 to 21	Explanatory
diag_1	110, 112, 114, 115, ... (848 values)	Primary diagnosis (first three digits of ICD9)	Qualitative		?			Explanatory
diag_2	110, 112, 114, 115, ... (923 values)	Secondary diagnosis (first three digits of ICD9)	Qualitative		?			Explanatory
diag_3	110, 111, 112, 115, ... (954 values)	Additional secondary diagnosis (first three digits of ICD9)	Qualitative		?			Explanatory
number_ diagnoses		Number of diagnoses entered to the system	Numeric		?		1 to 16	Explanatory
max_glu_ serum	>200, >300, normal, none	Range of the glucose serum test result, or if not measured	Qualitative	mg/dL	?	Blood test		Explanatory
A1Cresult	>8, >7, normal, none	Range of the A1C test result, or if not mesured	Qualitative	%	?	Blood test		Explanatory
metformin	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
repaglinide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
nateglinide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
chlorpropamide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
glimepiride	up, down, steady,	Indicates whether the drug was prescribed or there was a change in	Qualitative		?			Explanatory

	no	the dosage						
acetohexamide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
glipizide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
glyburide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
tolbutamide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
pioglitazone	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
rosiglitazone	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
acarbose	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
miglitol	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
troglitazone	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
tolazamide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
examide	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
citoglipton	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
insulin	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
glyburide-m	up,	Indicates whether the	Qualitativ		?			Explanatory

etformin	down, steady, no	drug was prescribed or there was a change in the dosage	e					
glipizide-metformin	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
glimepiride-pioglitazone	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
metformin-rosiglitazone	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
metformin-pioglitazone	up, down, steady, no	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
change	ch, no	Indicates if there was a change in diabetic medication (either dosage or generic name)	Binary		?			Explanatory
diabetesMed	yes, no	Indicates if there was any diabetic medication prescribed	Binary		?			Explanatory
readmitted	<30, >30, No	Days to inpatient readmission	Qualitative	Days	?			Response

Preprocessed Metadata table

Variable	Modalities	Meaning	Type	Measuring unit	Missing code	Measuring procedure	Range	Role
x		Row number of preprocessed dataset.	N		?			Explanatory
encounter_id		Unique identifier of an encounter	Numeric		?			Explanatory
patient_n		Unique identifier of a patient	Numeric		?			Explanatory
race	AfricanAmerican, Asian, Caucasian, Other, race_unknown	Patient's origin place	Qualitative		?			Explanatory
gender	Male, Female	Patient's gender	Binary		?			Explanatory
age	[0, 10), [10, 20), … [90, 100), Other	Patient's age	Qualitative	Years	?			Explanatory
weight	[0-25), [25-50), … [175-200) >200	Patient's weight	Qualitative	Pounds	?			Explanatory
adm_type_id	1-6	Identifier of the patient's admission type	Qualitative		6			Explanatory
disch_id	1-7, 11,.13, 18	Identifier of the patient's discharge type	Qualitative		18			Explanatory
adm_source_id	1, 4-7, 17	Identifier of the patient's admission source	Qualitative		17			Explanatory
time_in_hpt		Days between admission and discharge	Numeric	Days	?		1-14	Explanatory
payer_code	payer_code_unknown, WC, UN, MC, CP, DM, BC, HM,	Code of the paying method	Qualitative		?			Explanatory

	OT, SP, MD, CM, CH							
specialty	Cardiology, InternalMedicine, Surgery-General, Family/General Practice, Pediatrics, ObstetricsandGynecology, Psychiatry, Dentistry, specialty_unknown	Specialty of the admitting physician	Qualitative		?			Explanatory
n_lab_proc		Number of lab tests performed during the encounter	Numeric		?		1-105	Explanatory
n_proc		Number of procedures (other than lab tests) performed during the encounter	Numeric		?		0-6	Explanatory
n_med		Number of distinct generic drugs administered during the encounter	Numeric		?		1-65	Explanatory
n_outp		Number of outpatient visits of the patient in the year preceding the encounter	Numeric		?		0-21	Explanatory
n_emerg		Number of emergency visits of the patient in the year preceding the encounter	Numeric		?		0-13	Explanatory
n_inp		Number of inpatient visits of the patient in the year preceding the encounter	Numeric		?		0-21	Explanatory
diag_1	Respiratory, Circulatory, Neoplasms, Digestive, Injury, Diabetes, Genitourinary, Musculoskeletal, Other	Primary diagnosis	Qualitative		?			Explanatory
diag_2	Respiratory, Circulatory, Neoplasms,	Secondary diagnosis	Qualitative		?			Explanatory

	Digestive, Injury, Diabetes, Genitourinary, Musculoskeletal, Other							
diag_3	Respiratory, Circulatory, Neoplasms, Digestive, Injury, Diabetes, Genitourinary, Musculoskeletal, Other	Additional secondary diagnosis	Qualitative		?			Explanatory
n_diag		Number of diagnoses entered to the system	Numeric		?		1-9	Explanatory
A1Cresult	None, >8, >7, Norm	Range of the A1C test result, or if not mesured	Qualitative	%	?	Blood test		Explanatory
metformin	No, Steady, Up, Down	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
insulin	No, Steady, Up, Down	Indicates whether the drug was prescribed or there was a change in the dosage	Qualitative		?			Explanatory
change	No, Ch	Indicates if there was a change in diabetic medication (either dosage or generic name)	Binary		?			Explanatory
diabetesMed	Yes, No	Indicates if there was any diabetic medication prescribed	Binary		?			Explanatory
readmitted	NO, >30, <30	Days to inpatient readmission	Qualitative	Days	?			Response
other_meds	no_more_meds, takes_more_meds	Indicates if another drug apart from insulin and metformin has been administered	Binary		?			Explanatory

Final scope of the study

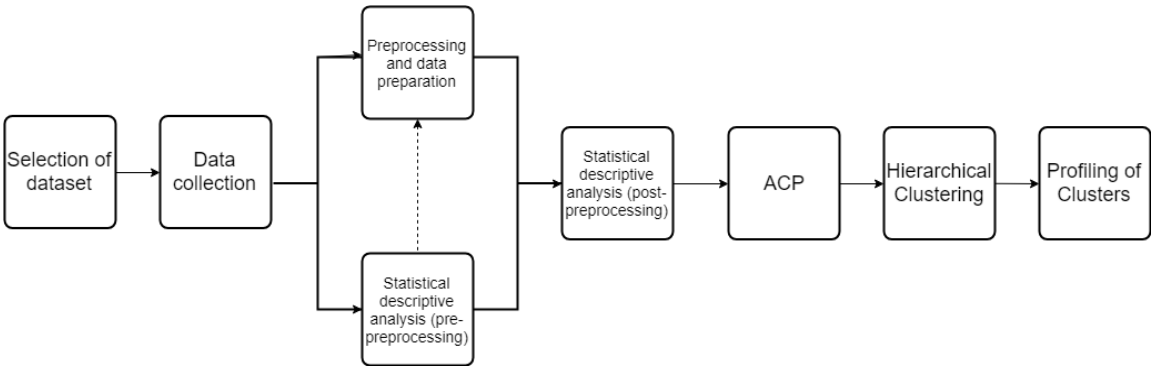
The first decision made is to only work with individuals whose weight has been recorded. In other words, we exclude the rows where *weight* is missing. The reasoning behind this is that the column *weight* has a lot of missing values, and we noticed that they might be structural instead of random. For this reason, our new criteria over the data is that each individual should have their weight recorded. This is our only row exclusion and after it we still have over 3000 individuals.

We also have some missing values on other columns, such as *race, payer_code, medical_specialty* and the diagnoses columns. However we have decided to include these columns and just input the missing values since the percentage of missing is nowhere close as that of *weight* and so we deem them as relevant to the study.

Next, we conclude that we have too many drugs variables at 24 different columns, which makes the dataset too crowded with data that is not that relevant to our study. Since *insulin* and *metformin* are the most important medications for diabetes and are well distributed over the factors, we decide to keep these two and group the others into a new binary column named *other_meds.*

Lastly, after a bit of discussion, we have decided that is best to include all three of the diagnoses columns. That is because we think is interesting to keep track of the differences in the diagnoses, so we can observe the evolution of the given patient. This reason and the fact that we have the total number of diagnoses will give us a broader view on the issue.

Datamining process



The first thing needed to do in any data mining study is to find a dataset that follows the established criteria. In our case one that has at least 500 records, seven numerical variables, two binary variables, five categorical ones and is about diabetic inpatients. we also have to collect the relevant data to our study. Again, in our case, it will be that of the dataset, with certain attention to the readmission variable.

Once a suitable dataset is found and the problem we want to study is determined, we start the actual process with the statistical descriptive analysis of the raw data. First, we select the variables that we deem are important to the study, such as *weight, DiabetesMed* and *Insulin* among others, and then we run the R script to get the different plots and statistics of these variables. We start the preprocessing in parallel to this phase, since the statistics are not needed to some parts of the preprocessing, such as eliminating the missing values of certain variables such as *race* or *weight*. We also start to change the *diagnosis* values at this moment.

After the preprocessing, we analyze again the chosen variables and generate the plots, with the intent of explaining the differences between the raw data (pre-preprocessing) and the processed data (post-preprocessing).

Next, we start the ACP analysis for numerical variables, where we check that we reach the necessary inertia for the data using scree plots, interpret the projections of the different factorial maps and observe the relationships that lie between variables (for example that there is a connection between being of the asian race and not being readmitted).

Then, in the “Hierarchical Clustering” phase, we group the data in clusters while getting the resulting dendrogram using an ascendent hierarchical method and Gower’s metric to obtain the distance matrix of the variables. In the “Profiling of Clusters” phase, we detect which variables are relevant in each cluster and the differences present in the different clusters using profiling graphs and GPGs and compare the results to those of the ACP phase.

Description of preprocessing and data preparation

First, we imported the data to R and ran an analysis over missing values. We noticed that we had four variables with important missing values which would need treatment. These were *race, weight, payer_code, medical_specialty* and the diagnoses (*diag_1, diag_2* and *diag_3*).

We noticed that particularly, *weight* had over 95% of missing values, meaning that other studies had immediately dropped the variable. Since we are interested in discovering new knowledge, we have decided to go the opposite way, instead dropping all the missing values and reducing our dataset from more than one hundred thousand encounters to exactly 3197. We account for the bias this decision might cause when dealing with our data, but we consider that this decision might bring different insights than those achieved from other studies, and that is the reason we do it.

After this decision, the percentage of missings varies over the rest of variables (particularly we note that '*diag_1*', the column consisting of the diagnosis of the first visit, has no missings now).

For these variables, we have decided to just create a new factor and consider the missings itselves relevant (not knowing the medical specialty could be related to the readmission variable and so on).

After these decisions we now have the data free from missing values, but we still have 50 variables. Knowing that the curse of dimensionality could result problematic for our study, we set off to study the variables themselves and see if there is irrelevant information or noise in our dataset. We find two things:

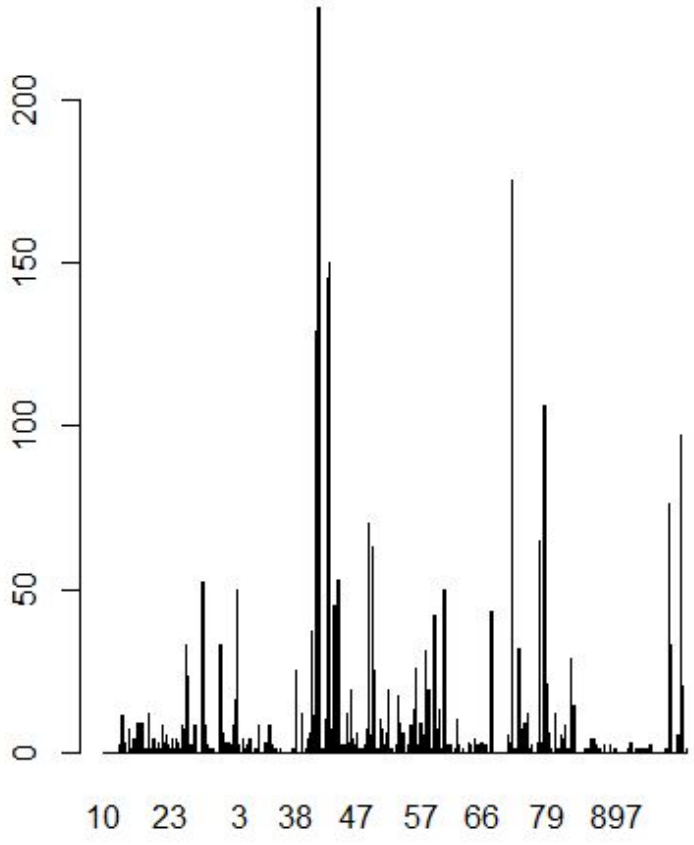
- Most of the drugs variables (namely all those that are not insulin and metformin) have in most cases the '*No'* factor. In some of those, it's even 100% of the cases, so it simply non-informative.
- The diagnosis variables have around 1000 factors, most of which are related to each other (for example, codes 390 to 459 and 785 are all related to circulatory problems). Our dataframe does not convey this in its current form, as 390 and 391 are as related as 390 and 800 (which corresponds to injury).

For the first problem, we decide that we can collapse all those drug variables (excluding insulin and metformin, which are really important when dealing with diabetes and thus we decide not to join them). For that, we will create a new variable '*other_meds*', whose factors will simply be '*takes_more_meds*' and '*no_more_meds*'. In particular, we collapse all the

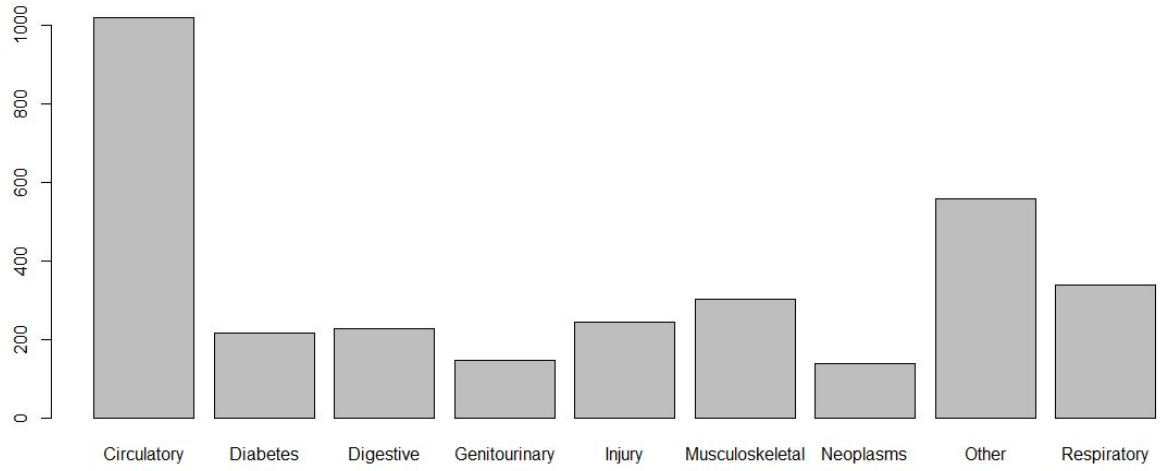
variables as: if all have value '*No*', then we will input '*no_more_meds*', otherwise we will input '*takes_more_meds*'. We consider this transformation to be meaningful in itself, as knowing whether a patient is taking other medicines is informative and interpretable, and also something experts need to know when dealing with this kind of problem.

Analysis over the results tells us that this variable is more or less split 50-50 between the two factors. We consider this positively entropic, since each instance has the same chances at ending at either, and thus it is informative. While it is less informative than having all the variables as we did before, this will greatly reduce our problems with the curse of dimensionality, while keeping some of the information we originally had.
After that, we drop the variables used for creating this new variable.

For the second problem, we have obtained a mapping done by experts which groups these factors into greater groups. Right now, the distribution of *diag_1* over factors is the following:



This is not very informative, but if we group them by the expert mapping, we will achieve this distribution:



Which is clearly much more interpretable and also improves semantics of the diagnosis, meaning that before we didn't know how 'close' 390 and 785 were, but now we know that they are both circulatory diagnosis, and very different from 800 which is injury-related.

Finally, and with the objective of embellishing the future PCA, we will reduce the names of our variables a bit, using the following mapping:

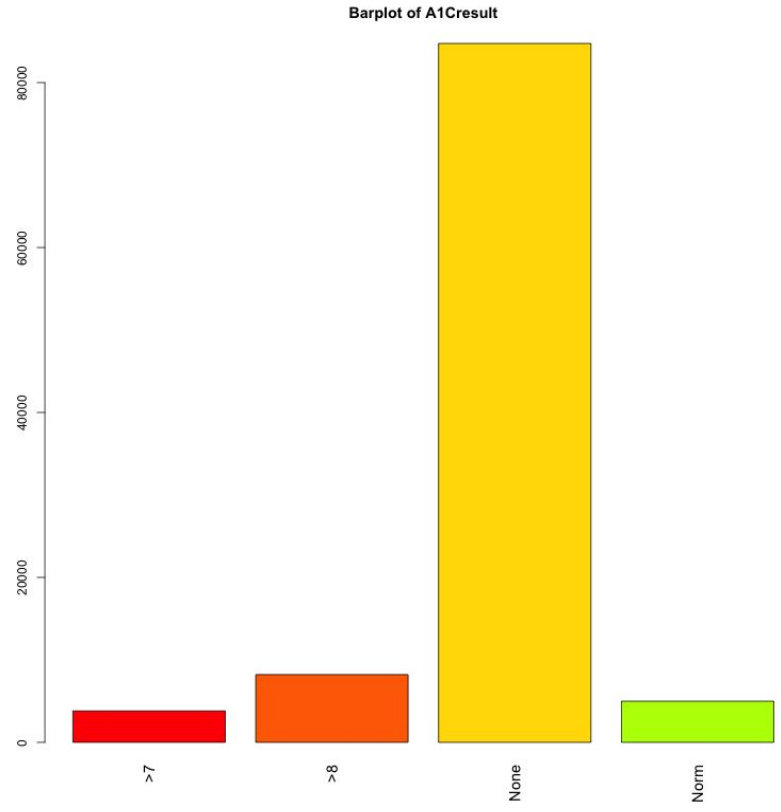
admission_type_id	adm_type_id
admission_source_id	adm_source_id
discharge_disposition_id	disch_id
time_in_hospital	time_in_hpt
medical_specialty	specialty
number_emergency	n_emerg
number_inpatient	n_inp
num_lab_procedures	n_lab_proc
num_procedures	n_proc
num_medications	n_med

With this we consider our preprocessing stage complete.

Basic statistical descriptive analysis

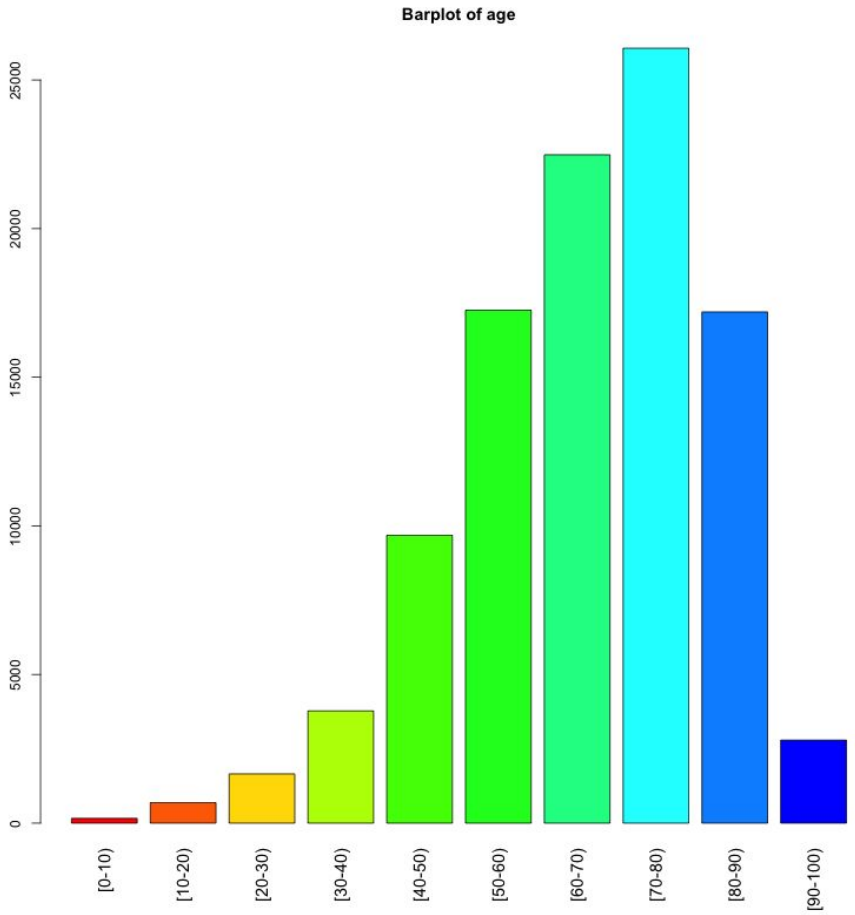
In this section we wish to gather a better understanding of the variables of our dataset by performing a statistical descriptive analysis. As seen in the previous section, several of the initial variables of the original dataset are irrelevant or escape the scope of this study, which means that having a statistical descriptive analysis for every one of them is rather pointless. Therefore, we will focus only on the relevant variables, and comment for those that were dropped or merged as to why we did so. Irrelevant variables such as the id of the patient or its row number are not being considered.

A1CResult



This variable measures the change in the A1C test result. It is a categorical variable that can take four possible variables: in the same order as the plot, >7, >8, None, Norm. The Barplot is valid for both the raw and preprocessed data, as it was unchanged by the preprocessing procedure. From it we can gather that most of the results are negative, therefore the relevance of this variable is questionable.

Age



The Age variable is categorical as it grouped different ages in 10 year range groups. It was already represented this way before our preprocessing. After it, the distribution did not change much. From the previous barplot, we can observe how most of our patients are of old age. Despite having the variable represented as categorical, given its numerical nature, we can estimate that the average age is around sixty years old. This variable is actually very informative and hints at the type of patients we are dealing with.

Change

This variable is meant to indicate whether there is any change in diabetic medication in any shape of form. It is a binary variable which can take the values of Ch, indicating a change, and No, indicating a lack of it.
Its distribution before and after preprocessing is the following one:

	Ch	No
Raw	0.462	0.538
Preprocessed	0.333	0.667

This indicates that the original dataset had a more even distribution of the two values. We can observe that having dropped some of the rows, the distribution moves towards the patients with no change.

DiabetesMed

This variable indicates whether a given patient is prescribed any diabetic medication at all. It is a binary variable that has either “No” or “Yes” as possible values.
Its distribution before and after preprocessing is shown in the following table.

	No	Yes
Raw	0.230	0.770
Preprocessed	0.312	0.688

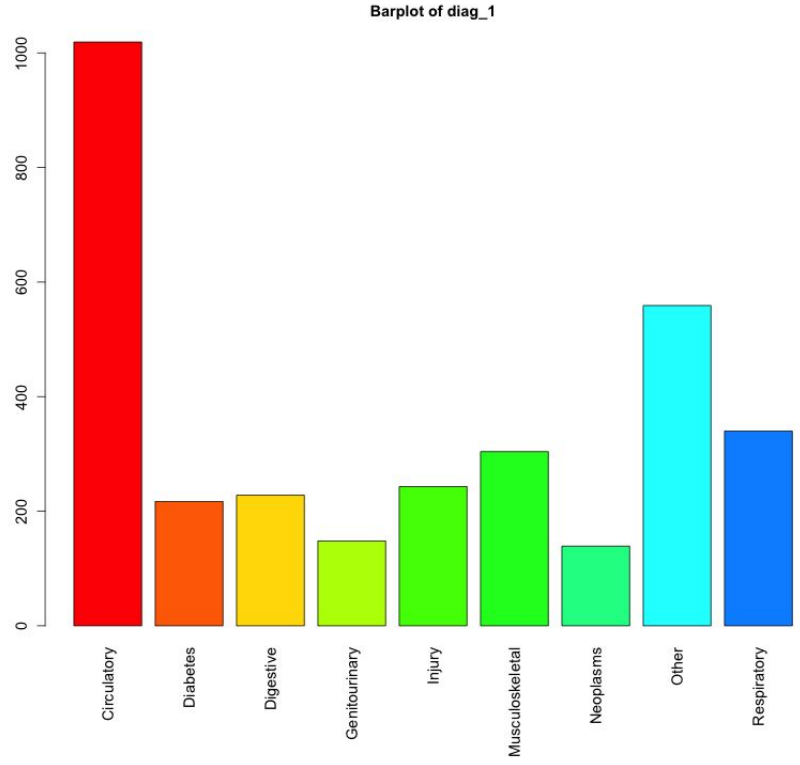
From the previous results we can see that the majority of our patients are taking medication for diabetics, which makes sense. However, we also observe that the preprocessing changed the proportion a little bit, increasing the number of patients without medication.

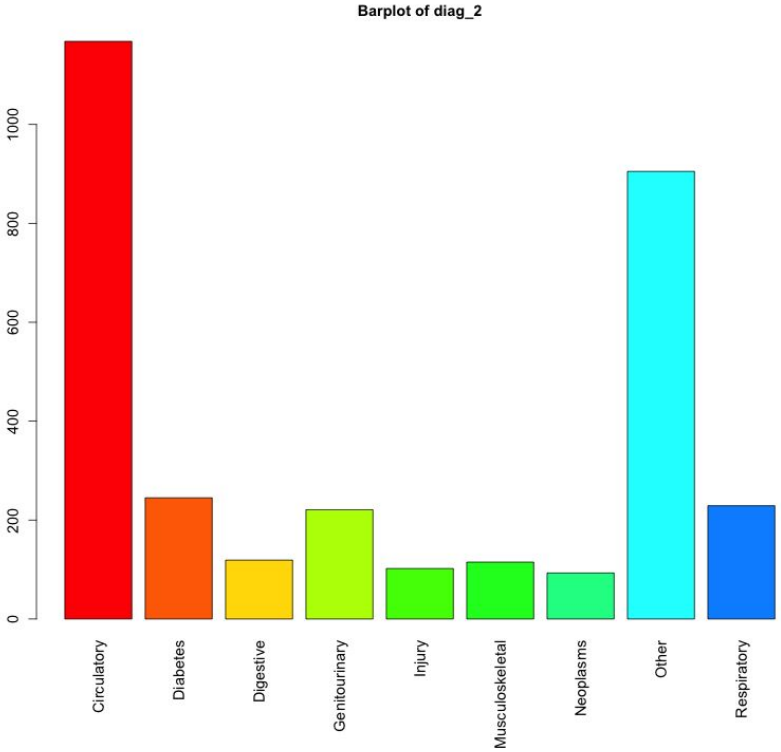
Diag1, Diag2, Diag3

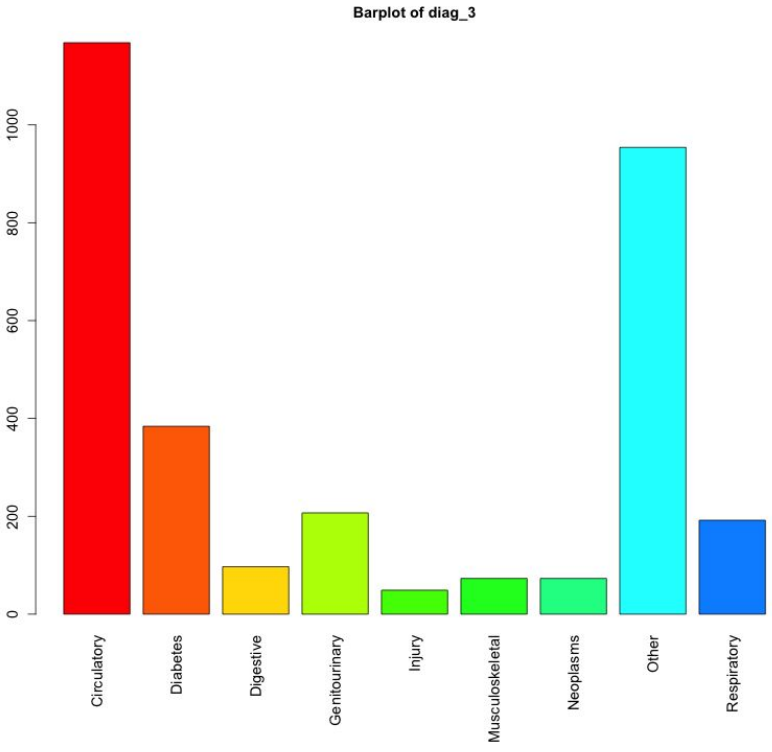
The diagnosis variables indicate what was cause of the encounter with the patient.

As seen in the preprocessing section, all three of the diagnosis variables were assigned cryptic numeric values that made it very difficult to gain any insight of the information they may convey. We grouped the values into the categories of diagnosis they belonged to and reduced the possible values to only nine: Circulatory, Diabetes, Digestive, Genitourinary, Injury, Musculoskeletal, Neoplasms, Other and Respiratory.
Given the already discussed problematic, we are only going to study the preprocessed data.

We think it is interesting to study all three diagnosis variables together, to see if we can observe any evolution among our patients. The following barplots show the distribution among the different diagnoses for all three of the variables.




Barplot of diag_2
0
200
400
600
800
1000
Circulatory
Diabetes
Digestive
Genitourinary
Injury
Musculoskeletal
Neoplasms
Other
Respiratory


Barplot of diag_3
0
200
400
600
800
1000
Circulatory
Diabetes
Digestive
Genitourinary
Injury
Musculoskeletal
Neoplasms
Other
Respiratory

At first glance, we can gather that the Circulatory diagnosis dominates in all three of the variables, together with Other. The rest of the diagnoses are somewhat evenly distributed in the first variable, but we see an increase in Diabetes diagnosis and a decrease in the rest of them in the following diagnoses.

Gender

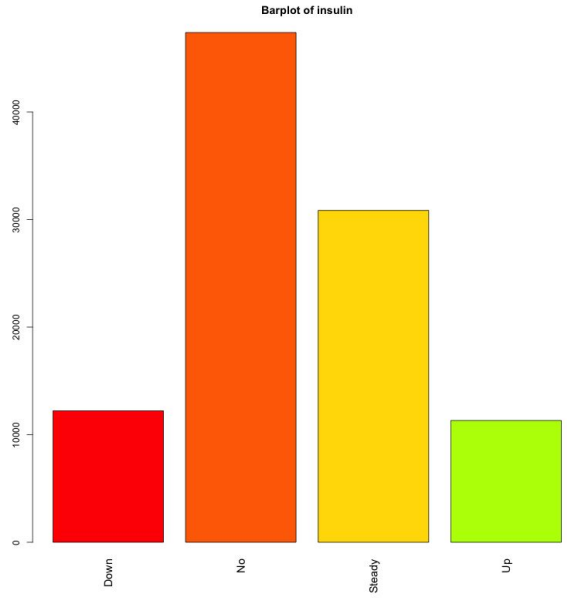
The gender variable is binary and refers to the sex of the patient. Its distribution before and after the preprocessing is the following one:

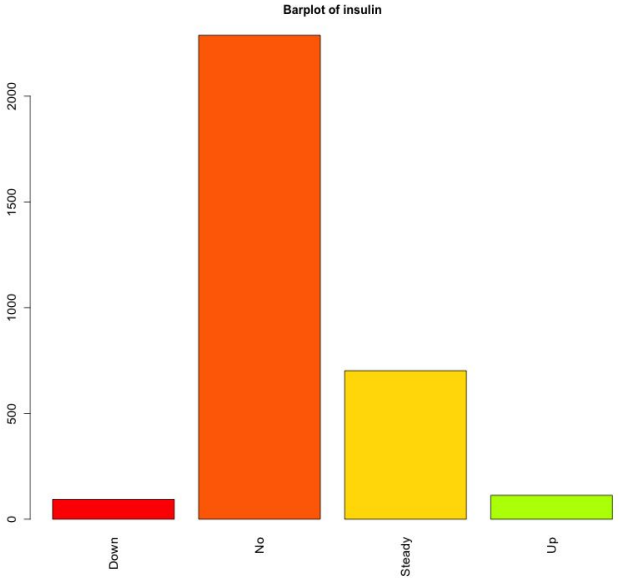
	Female	Male
Raw	0.538	0.462
Preprocessed	0.515	0.485

We can observe that in both cases the population is represented slightly more by females but is overall quite balanced. The proportion of females decreased somewhat after the preprocessing.

Insulin

This variable is one of the drug variables that we decided to keep separate for obvious reasons. It is a categorical variable that can take four possible values: Down; indicating a decrease in dosage, No; indicating that the drug was not prescribed, Steady; no changes in prescription and Up; an increase in dosage.

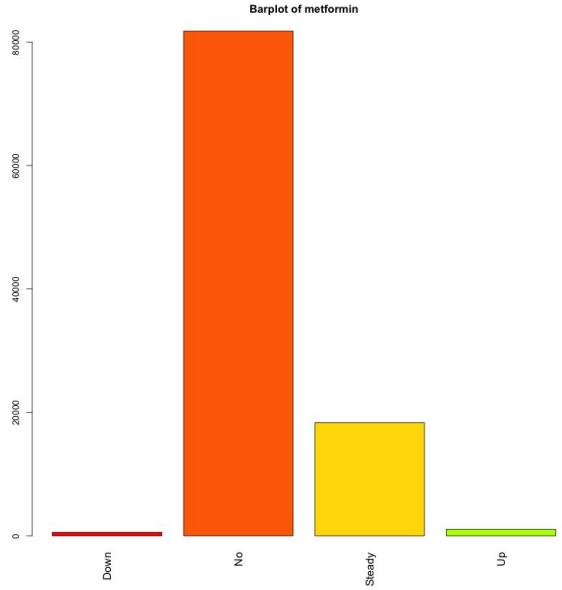


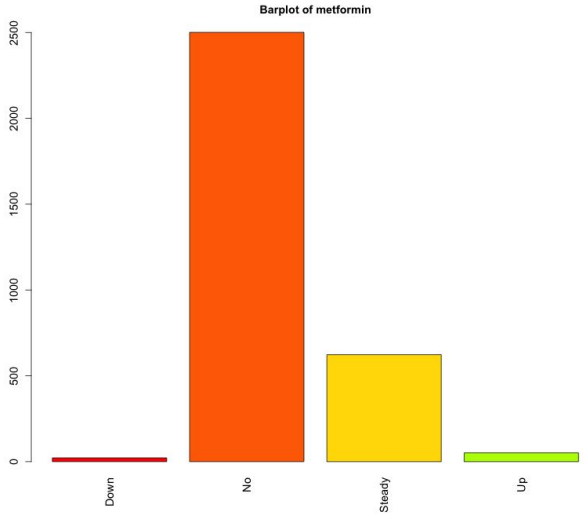


The barplots above, show the value distribution of the values before and after preprocessing. We can observe how most of the patients are not prescribed insulin in both cases, but after our preprocessing, the proportion of those in respect to all other increased significantly.

Metmorfin

This variable is the other medicament value we choose to keep separate from the rest as it is a drug used to treat type 2 diabetes. Its values are the same as the ones we had for insulin; down, no, steady, up.





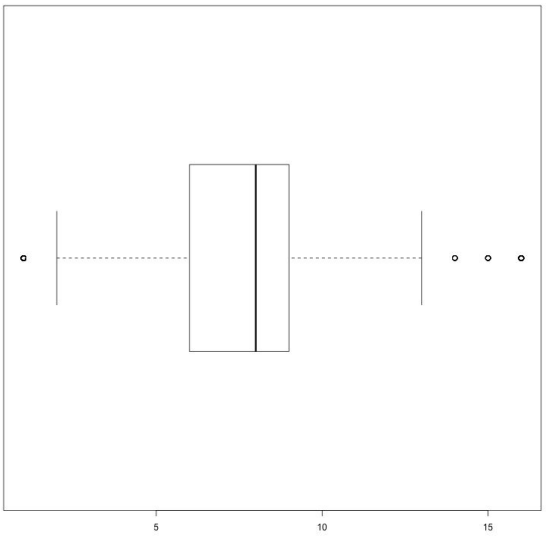
The previous barplots show the value distribution of the variable before and after preprocessing. Similar to what we observed in the insulin variable, the majority of the

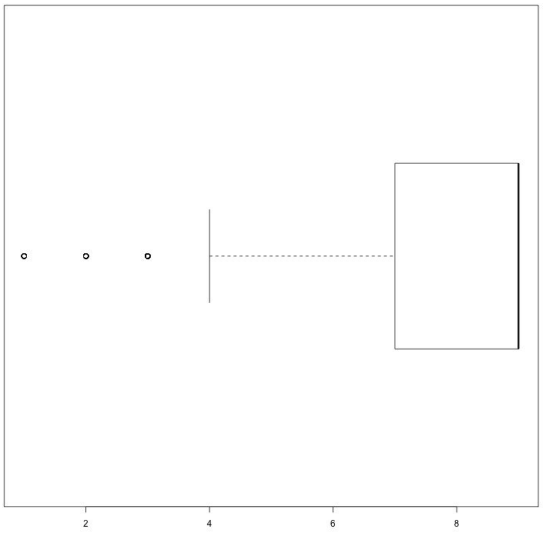
patients do not receive this medication. However, in this case the preprocessing decreased the proportion of such patients.

N_diag

This variable measures the total number of diagnoses that exist in the system for any given patient. It is a numerical variable with integer values that range from 1 to 16. However, after the preprocessing, the range of possible values decreased. In the following table we include the relevant parameters. Then, we include a boxplot for each case for a visual interpretation.

	Raw	Preprocessed
Min	1	1
Max	16	9
Median	8	9
Mean	7.4	7.98
SD	1.9	1.52



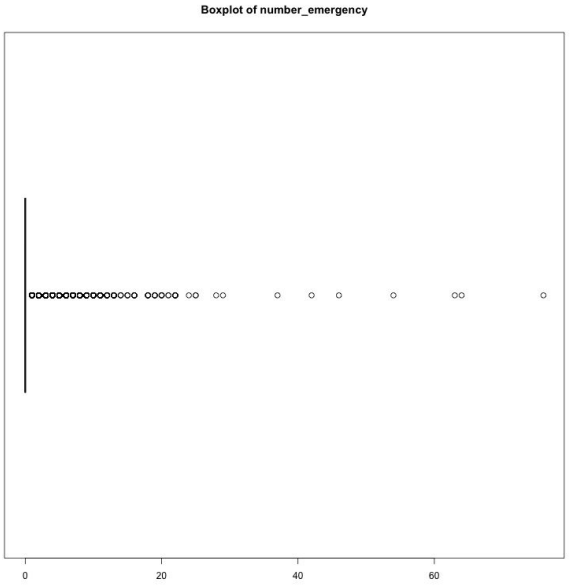


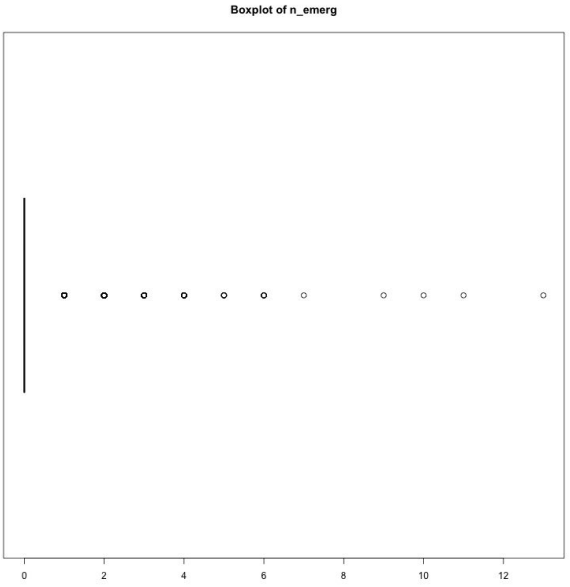
With the information above we can observe that our preprocessing procedure affected this variable greatly. In the original dataset, the variable was quite nicely distributed with a similar median and mean, in the middle of the maximum and minimum values, and a standard deviation that hints at a normal distribution. However, after dropping all patients whom their

weight was not registered in the system, the variable changed. It became greatly centered to the right, as we can observe in the boxplot. This means that the subset of patients that we are studying have had in an overwhelming majority between seven and nine procedures, but not more, which is rather weird. Maybe in the next sections we will gain some insight as to why is that.

N_emergency

This numerical variable indicates the number of emergency visits of the patient in the year preceding the encounter. The following boxplots will convey a visual understanding of the nature of this variable.





The figures above show the distribution for both the raw data and the preprocessed one. At first glance we see that they are very similar. The overwhelming majority is standing at the zero mark, meaning that most patients do not have an emergency visit in their record. Then we see a group of outliers that spreads out the further to the right we go. After preprocessing the density decreases and the maximum value too; in the raw data the most extreme outlier was standing at almost eighty visits, in the next one, the maximum value is thirteen.

N_inp, N_out

This two variables measure the number of inpatient visits, those visits that result in hospitalization, and the number of outpatient visits, those that don't. We believe that it is interesting to study this two variables together to observe their similarities and differences as

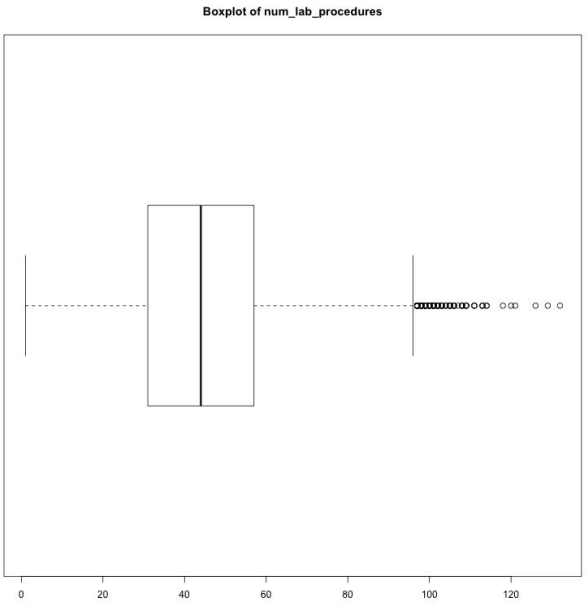
both are highly relevant to the study. We will focus only in the already preprocessed data, because there is no relevant change to the shape of the variable distribution after the preprocessing for either of the variables, and we are much more interested in the subset of patients we chose to study. In the following table we will include the relevant statistics for both variables.

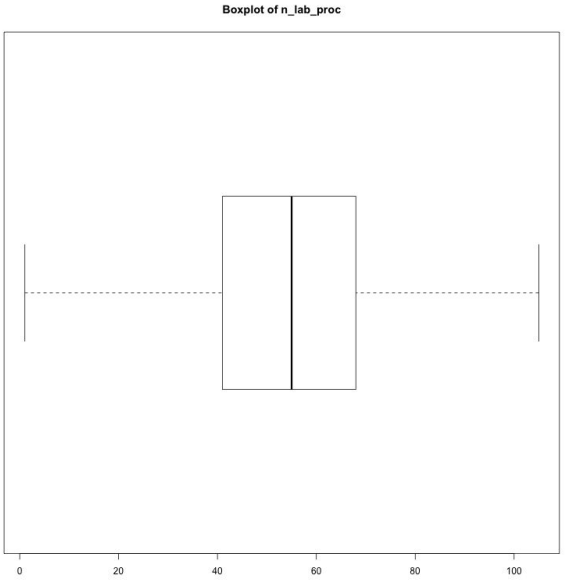
	N_inp	N_out
Min	0	0
Max	21	21
Median	0	0
Mean	0.57	1.09
SD	1.21	1.87

The previous table demonstrates that there is a connection between the two variables, as it should be expected. Both are highly skewed to the left, indicating that most of the patients do not require hospitalization. Their means, however, are somewhat different, with a greater mean of the number of outpatient visits.

N_lab_proc

This variable measures the number of lab tests realized upon the patient during the encounter. This includes blood tests and similar tests. It is an integer numerical variable that ranges between zero and a hundred procedures, with some going above that in the original dataset. The next to boxplots refer to the shape of this variable before and after preprocessing.

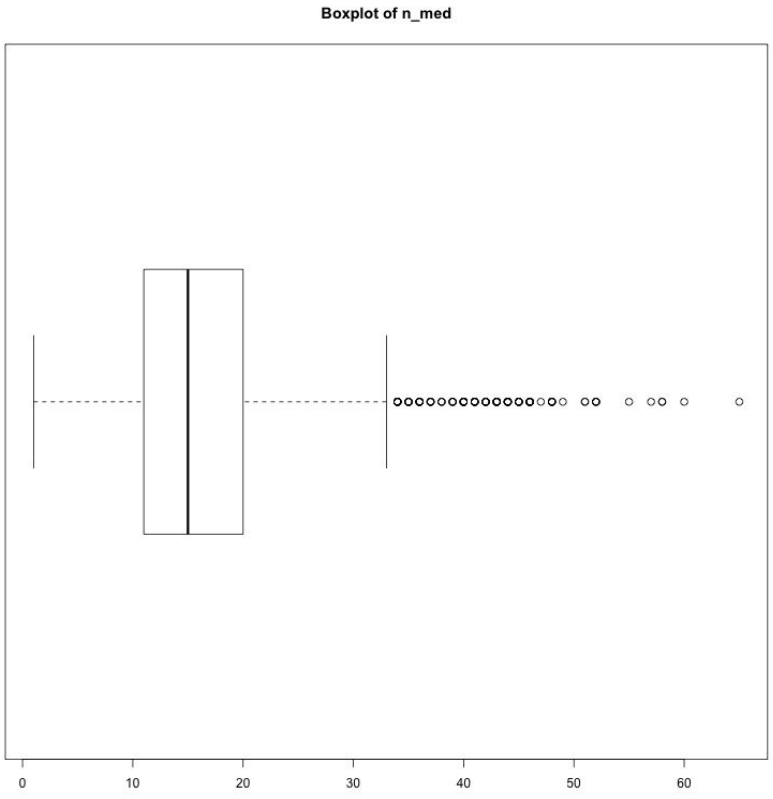




As conveyed visually by the above figures, the preprocessing procedure did not have a great impact in the distribution of this variable, we just lost some outliers among the way. The shape of the boxplot in both cases is quite nice; symmetrical with proportionate and even whiskers and a very similar mean and median. All of this hints at a probable normal distribution of the variable.

N_med

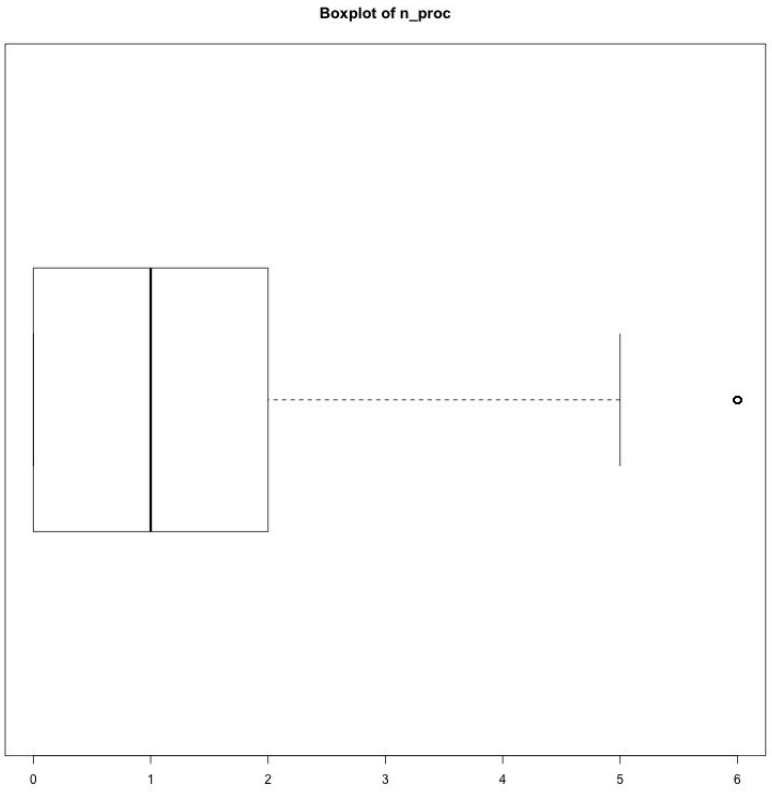
This variable indicates how many distinct generic drugs were administered to the patient during the encounter. Its values are integers and range between zero and forty, with some outliers going as far as eighty in the raw data and sixty in the preprocessed one. We will include only the boxplot belonging to the preprocessed data because there is not much change observed by the process other than the disappearance of some outliers.



The mean and median are 16.35 and 15 respectively. The max value is 65, but the previous boxplot tells as that this is an extreme value, together with some other outliers and that most of the values fall around the median, with a standard deviation of 7.89.

N_proc

This variable refers to the number of procedures that were carried upon the patient during the encounter, excluding lab tests (those were already measured by the n_lab_proc variable). It is a numerical variable that takes integer values ranging from zero to six in the selected subset of patients. We are going to include a boxplot and a summary of the variable for the preprocessed dataset to gain some understanding of its shape. We will omit the metadata for the original dataset regarding this variable as there is not much change to it and therefore lacks relevance.



The minimal and maximal values are 0 and 6, respectively. The median and mean are 1 and 1.49, respectively. The standard deviation is 1.78. From this information we can gather that usually a single procedure is carried upon each patient in any given encounter. However it is not uncommon to have more procedures than that, up to 2 still being usual, or no procedure at all.

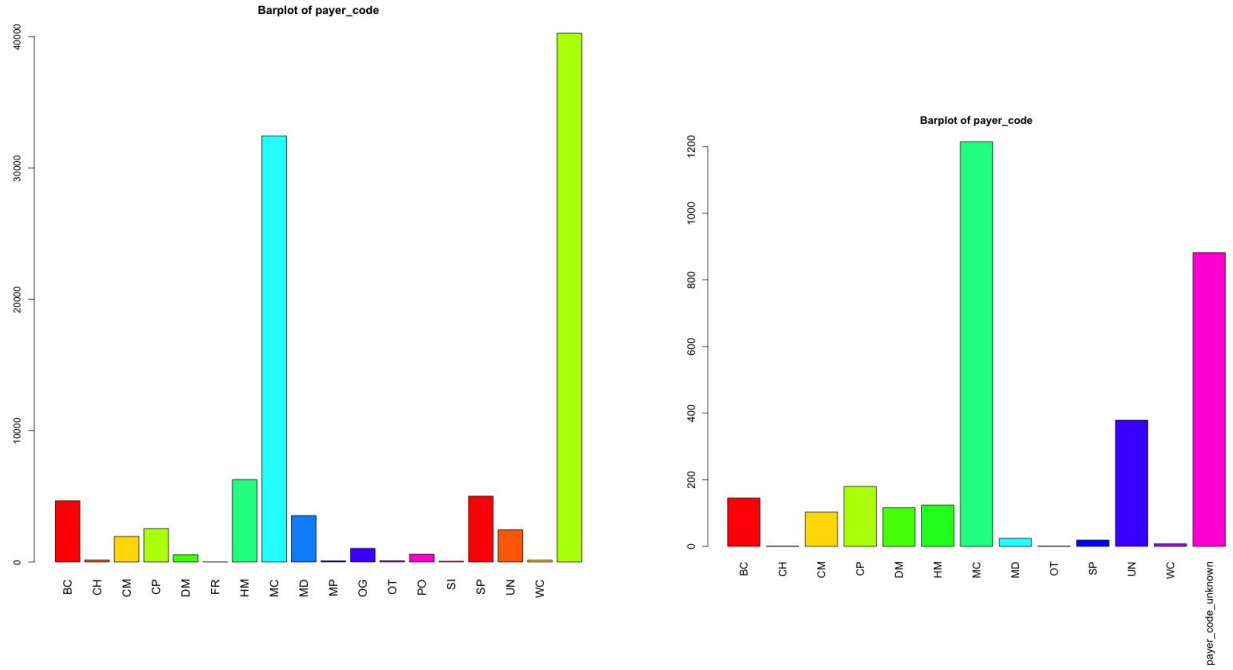
Other_meds

In the original dataset we had many variables (up to 24) for specific medications that could be administered to any given patient. The problem, other than having too many variables, was that most of them were not relevant at all, because the vast majority of patients did not take them. As it was explained in the preprocessing section, the solution we came up with was to separate the two drugs that specifically relate to diabetes, i.e. insulin and metformin, and grouped all other medicaments into a single variable, this one. It is a binary variable that takes either “no_more_meds” or “takes_more_meds” as values. If all of the original medication variables are negative for any given patient, then that patient is assigned no_more_meds. If any of them was positive, then it is assigned take_more_meds. The proportion of the two values is 0.557 and 0.443 respectively. Therefore, our decision to

group those variables made sense, because more than half of the patients do not take any of the twenty-four drugs that are available.

Payer_code

This variable is meant to indicate which is the entity or organization that pays for the patients visit. It is a categorical variable with thirteen possible values that refer to the code of an insurance or if the payer is unknown. Even if the specific meaning of each code escapes us, the variable is still relevant because it indicates something about the social status of the patient, and can be used to group the patients with similar profiles. Next, we include the barplots belonging to both the raw and preprocessed data.



The codes are in the same order in both plots, however some of the original codes that already had very low occurrences in the raw data disappeared after the selection of our subset. The rightmost column belongs to a missing value in the raw data, that was later changed to “unknown” after the preprocessing, because having a missing variable still conveys information about the individual being studied. The proportions in both plots are somewhat similar, with a prevalence for unknown codes and the “MC” code. However, we can observe some changes: Some codes disappeared, the amount of unknown values decreased significantly, and the “SP” code also decreased significantly, while “UN”

increased. All this may indicate the subset of patients we chose is not quite random regarding this variable.

Race

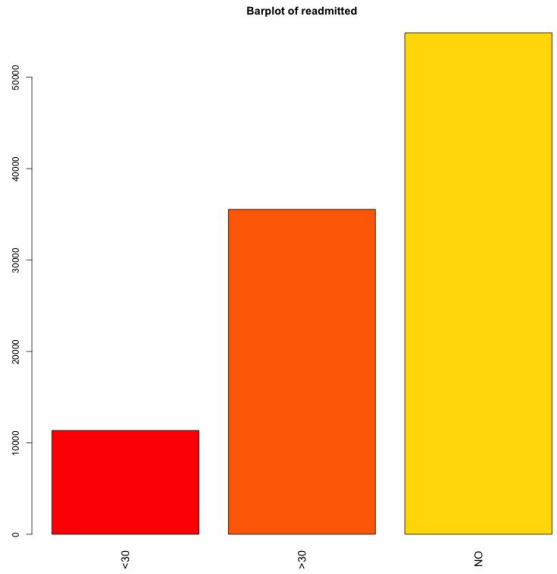
This variable classifies the patients according to their race. It is a categorical variable that originally had 6 possible values: African American, Asian, Caucasian, Hispanic, Other and NA, later changed to unknown. Some of the variables had very low representation, and disappeared after our subset selection. In the next table we will provide a numerical summary of the numbers, we think that in this case it may get the point across better than a barplot.

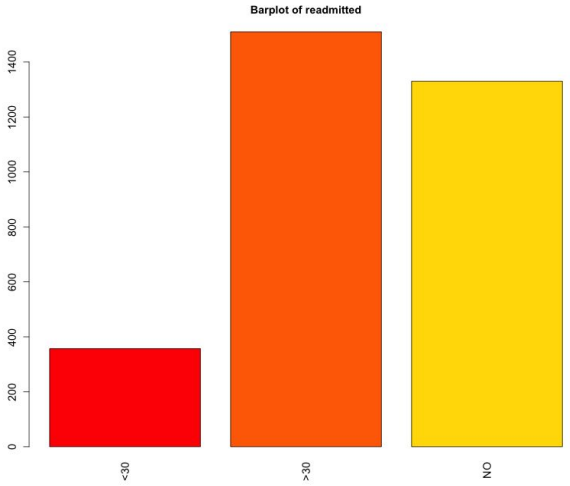
Category	Individuals, Raw	Individuals, Preprocessed	Percentage, Raw	Percentage, Preprocessed
African American	19210	113	18.87%	3.53%
Asian	641	9	0.63%	0.28%
Caucasian	76099	2907	74.78%	90.93%
Hispanic	2037	-	2.00%	-
Other	1506	30	1.48%	0.94%
NA - Unknown	2273	138	2.23%	4.32%

As we can observe in the previous table, the dataset is heavily biased towards caucasian individuals, and more so after our subset selection. Therefore, we must be careful with conclusions regarding the race of unrepresented individuals, such as those with Asian race.

Readmitted

This is our response variable. It indicates whether a patient was readmitted or not and if that happened before or after 30 days of their last encounter. The purpose of this study is to find any relation with the rest of the variables with this one, so it is important to understand how this variable behaves in our dataset. We will include the barplots for both the raw and preprocessed data as well as a numerical summary table for a more precise inspection.





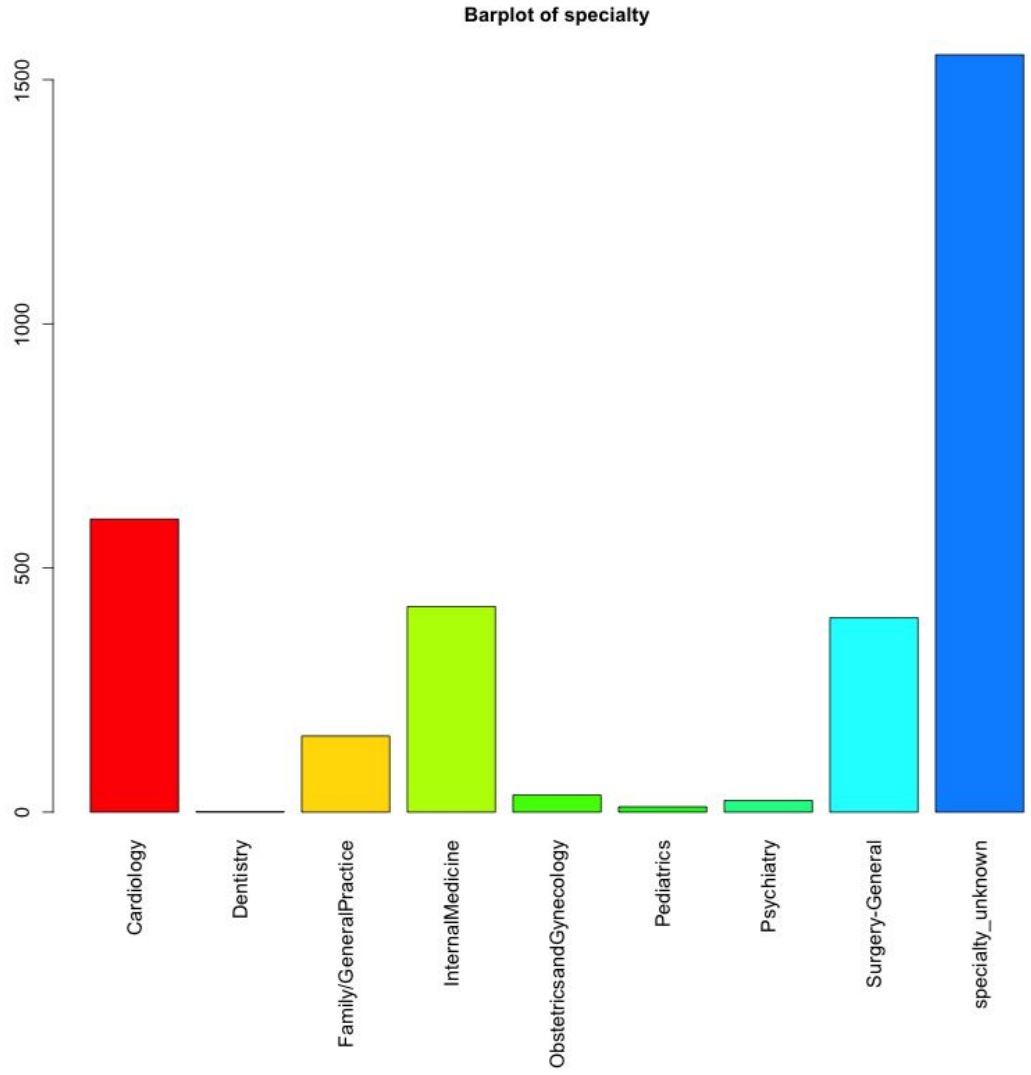
Category	Individuals, Raw	Individuals, Preprocessed	Percentage, Raw	Percentage, Preprocessed
<30	11357	357	11.16%	11.17%
>30	35545	1510	34.93%	47.23%
No	54864	1330	54.91%	41.60%

First of all, we see that the proportion of the categories were not maintained once the preprocessing was done. Therefore, our slice of the population is not exactly a random sample regarding this variable. However, we already justified the decision of selecting a concrete subset of the original raw data in the preprocessing section. Even more, we see that the proportion of readmitted patients in less than thirty days is actually maintained, and this is the most important category: patients readmitted in less than thirty days, maybe should not have been sent home, and our study aims to see if there is any cause that could help take the aforementioned decision.

Specialty

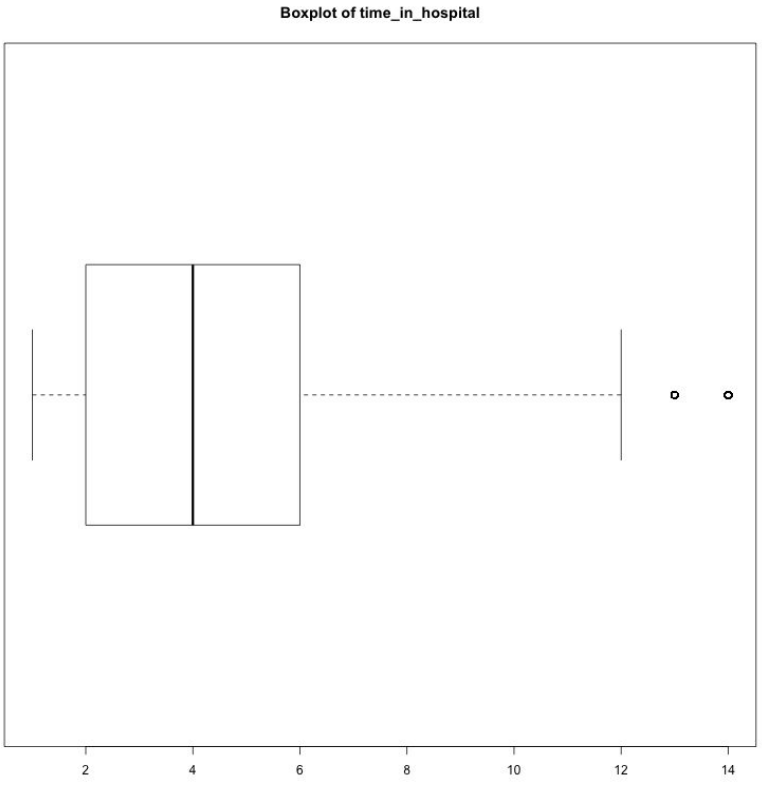
This variable specifies which is the specialty of the physician that admitted the patient. It is a categorical variable admitting nine possible values (in our final dataset). In the original dataset, the amount of possible values was confusingly high, with a lot of values having a

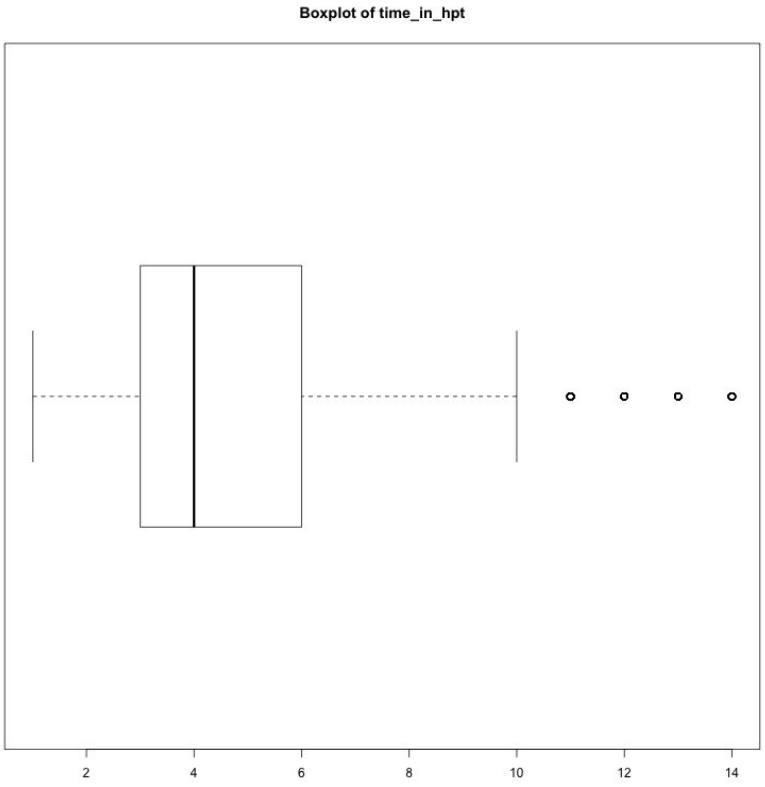
single or very few representatives. After the preprocessing procedure, most of the underrepresented specialties disappeared, leaving us with the most important ones. As we will see in the following plot, most of the specialties are unknown, with a high representation of cardiology, internal medicine and general surgery.



Time_in_hpt

This is a numerical variable that measures the time any given patient spent in the hospital between the admission and dismission dates. We will include the plots for both the original dataset and the final one.

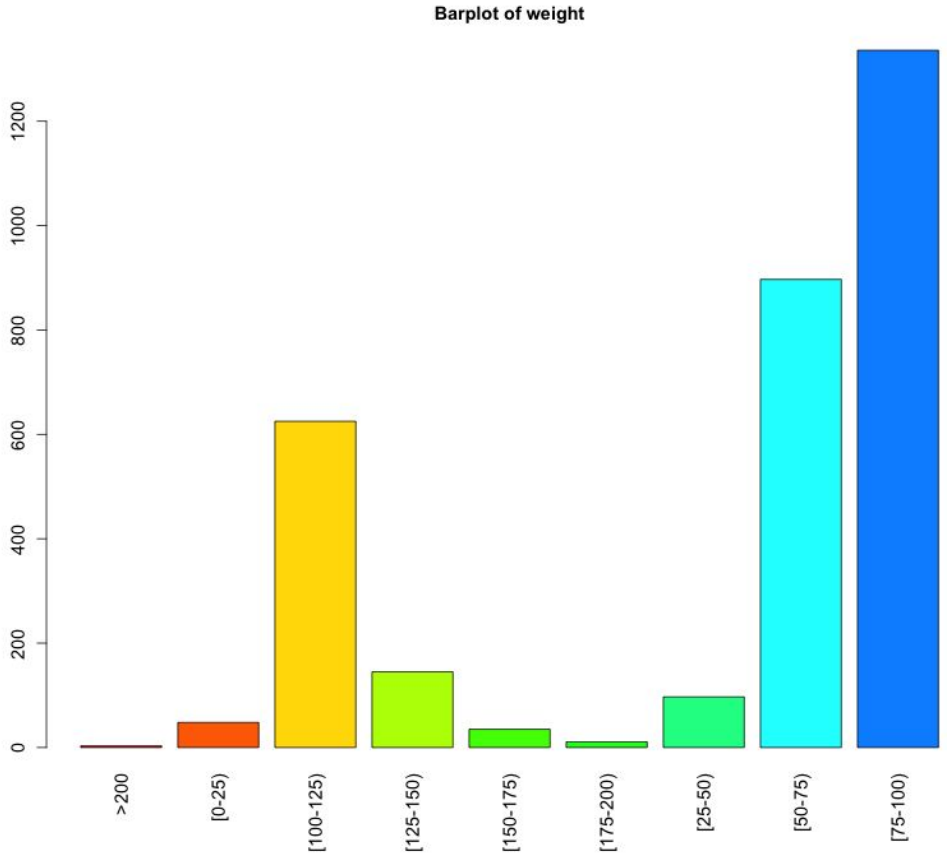




As it can be observed in the above boxplots, the value range was maintained, but after the preprocessing the data moved more towards its mean, leaving a more balanced boxplot.

Weight

This variable is very relevant to our study, because it is the one that was used to select the subset of patients that constitute the final dataset. In the original dataset, the weight variable had an overwhelming amount of missing values, probably because most of the encounters do not require to know the patients weight (for example, in medical specialties such as dentistry). However we thought that for a study over diabetic patients, it would be interesting to know their weight. As a result, we selected the subset of patients with a valid weight value. This variable is actually categorical, as it groups different weights in ranges of twenty-five kilograms. The following plot is in numerical order, so it makes it hard to see, at first glance, what is the range where most of the patients are found.

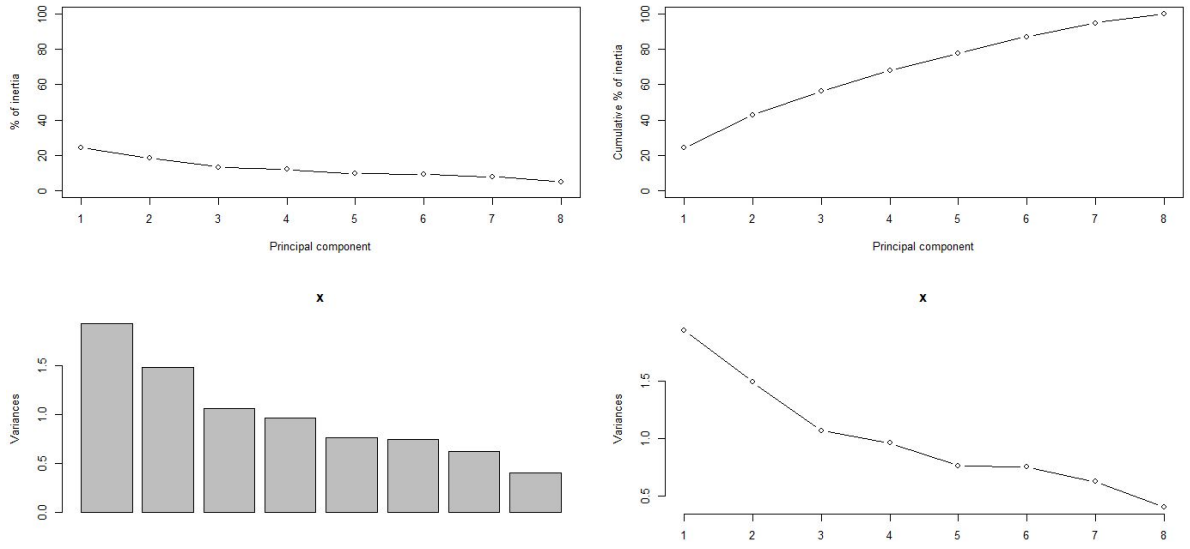


The mean weight is somewhere in between seventy-five and a hundred kilograms, with a bigger number of individuals below that range but a significant amount above that, too. The

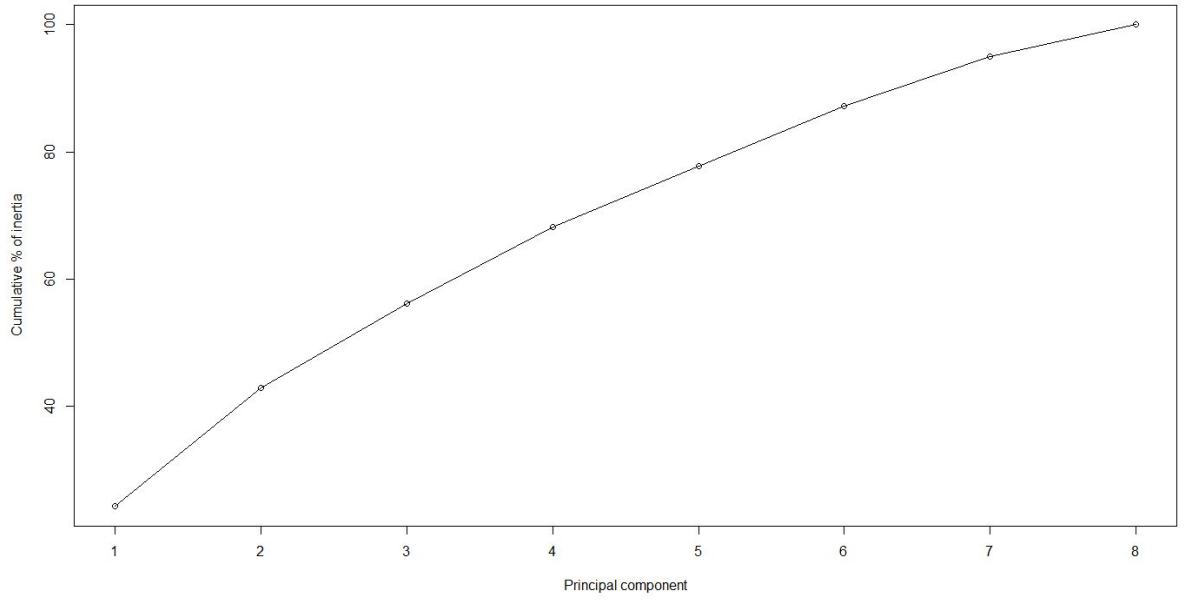
mean weight of U.S. citizens is around 80 kg, so our subset of patients is not particularly out of range regarding their body mass.

ACP

Running the scripts attached below give us the following scree plots:



In particular, we are interested in the cumulative percentage of variance explained:

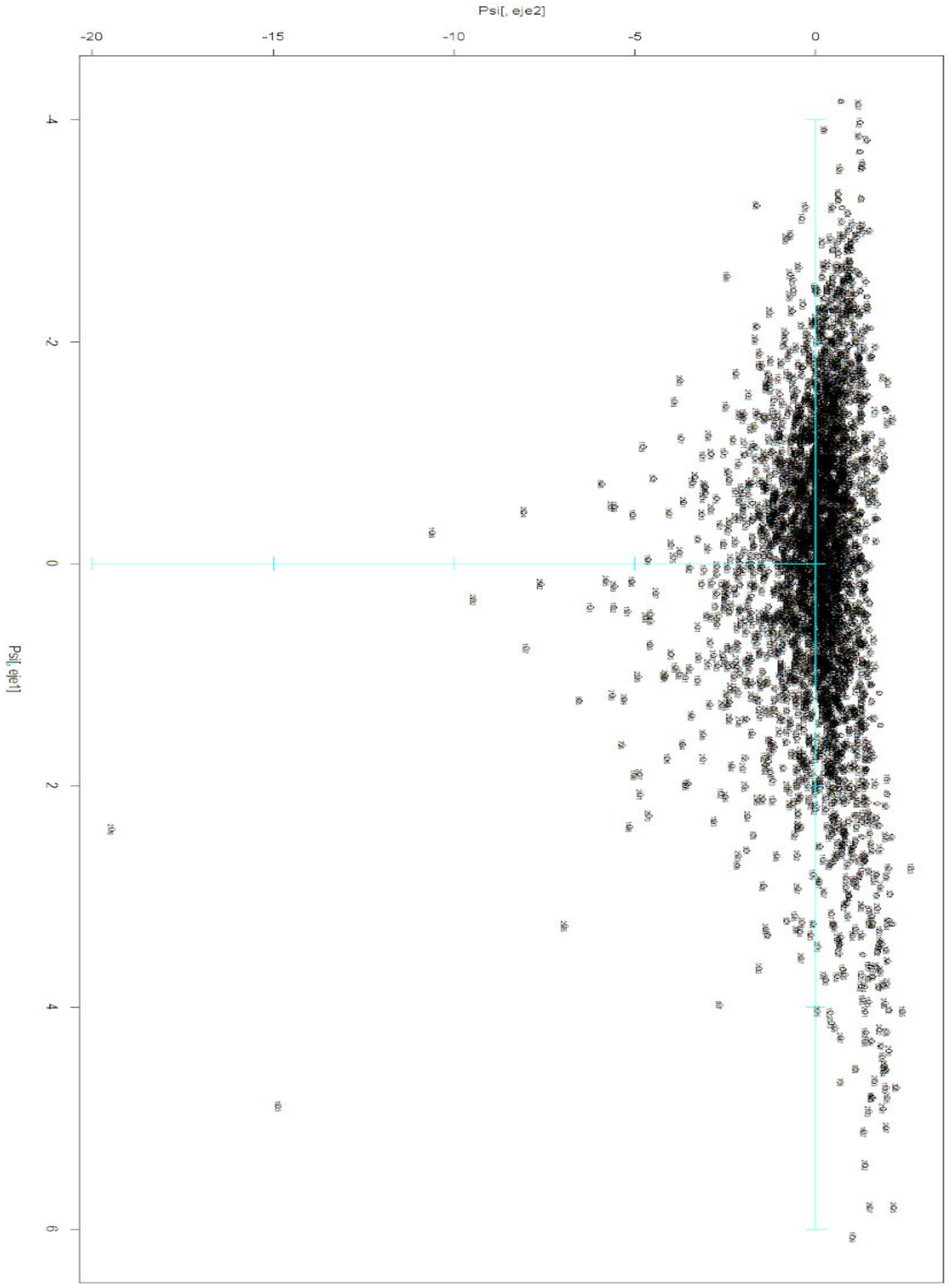


As we can see, we reach the necessary inertia (80%) between 5 and 6. The exact percentages are the following:

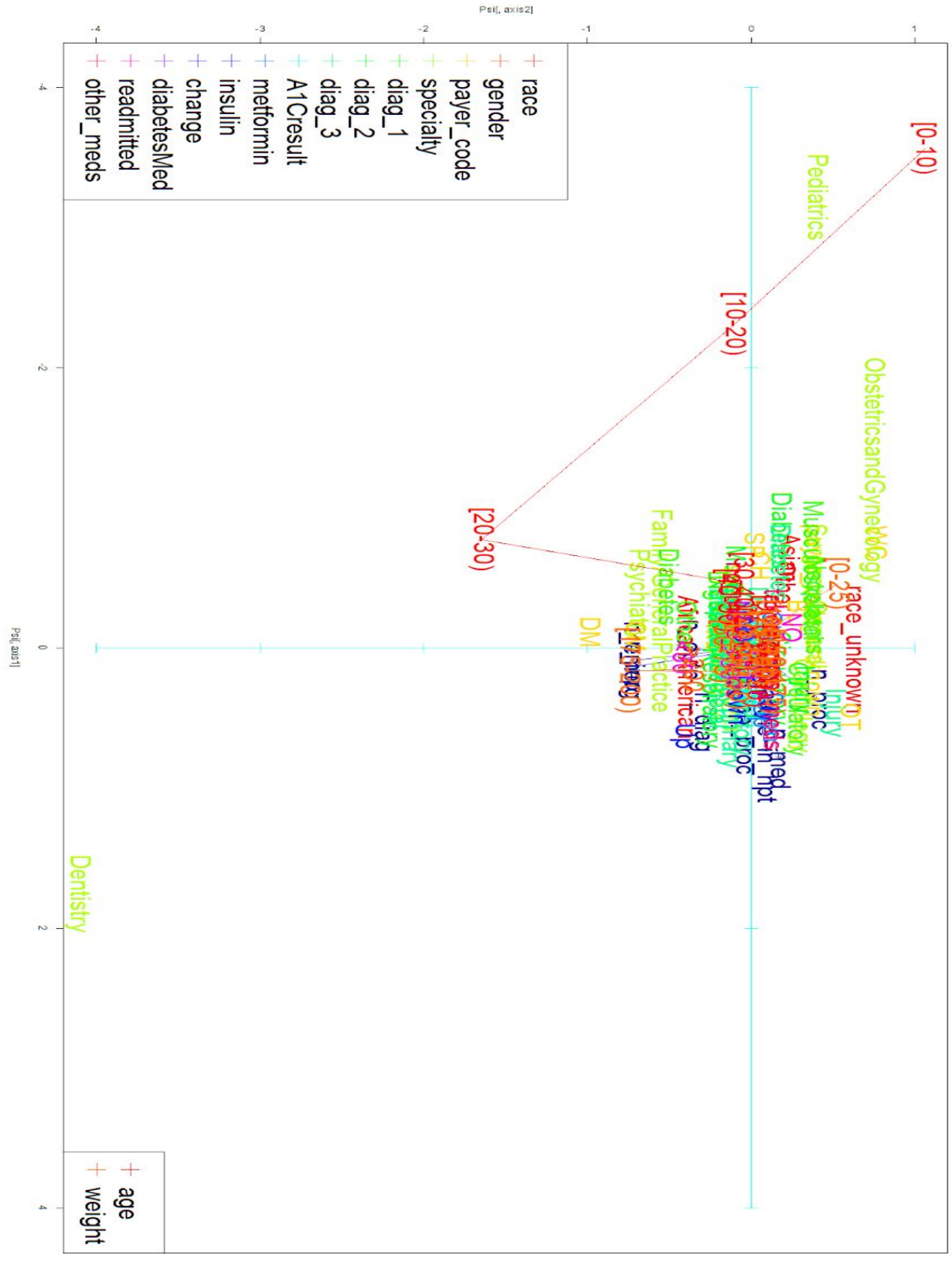
24.19869 42.81850 56.16501 68.20291 77.72693 87.12418 94.93362 100.00000

Therefore, we have decided to use 6 as our number of principal components.

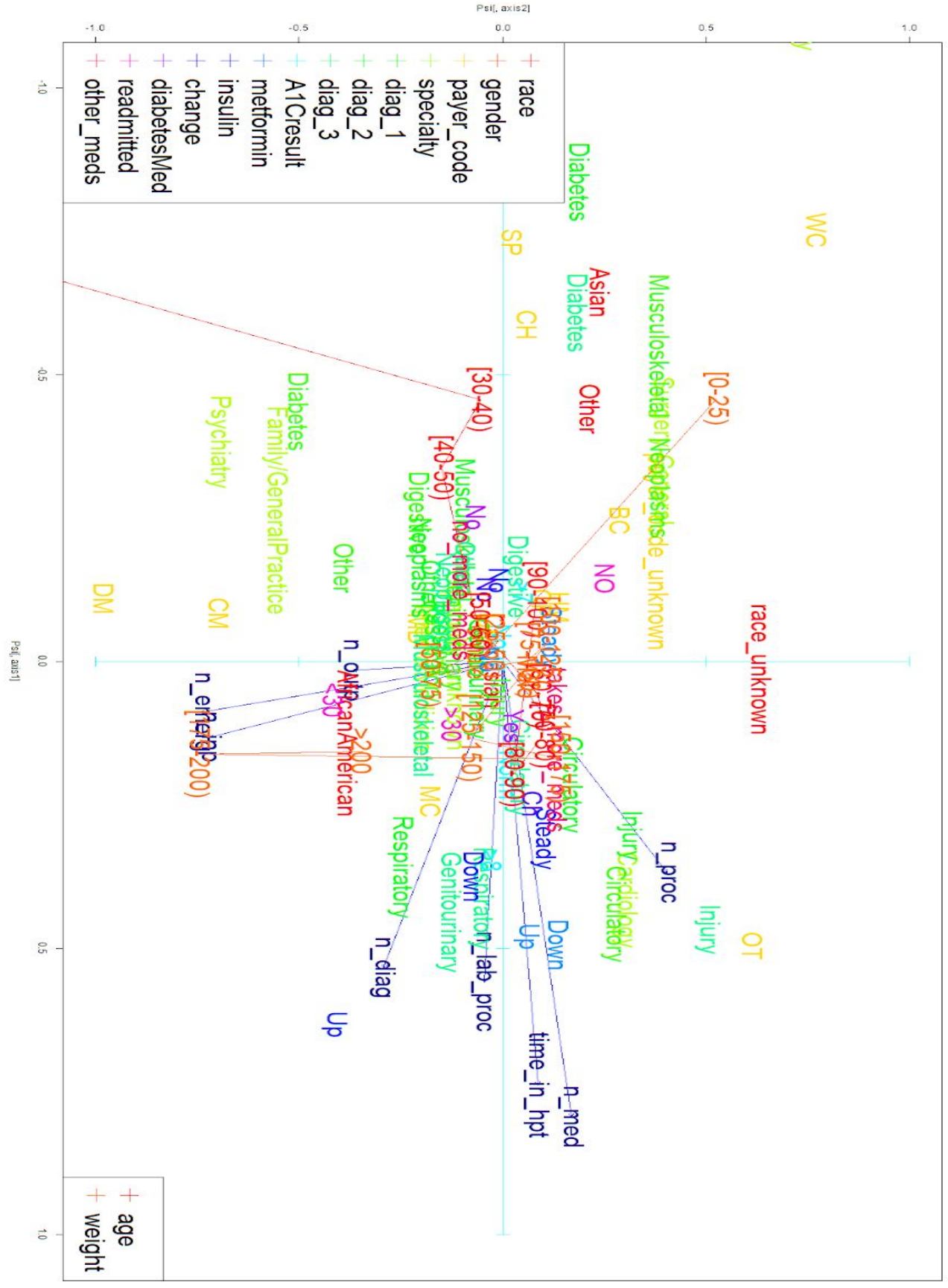
Axis 1 vs 2



Individuals plot



Common projection of variables



Common projection of variables (Zoom)

Interpretation

First we tried to understand the underlying logic for the axis, and the result seems quite easily inferrable.

- Axis 1 (horizontal) seems to be proportional to the ‘gravity’ of the admission. We deduce this from the variables that contribute positively to it, as admissions further on the right have more diagnosis, more procedures (lab and standard), more number of medications administered and longer time spent in the hospital.
- Axis 2 (vertical) seems to be inversely proportional to the ‘reincidence’ probability of the patient. Lower values come from those admissions of patients who have high number inpatient, number outpatient and number emergency. We recall that these variables indicate the number of times the patient had already visited the hospital in that year (be it being admitted into it, without needing to be admitted or in an emergency respectively). Especially so for patients who were readmitted or came in emergency, rather than outpatients.

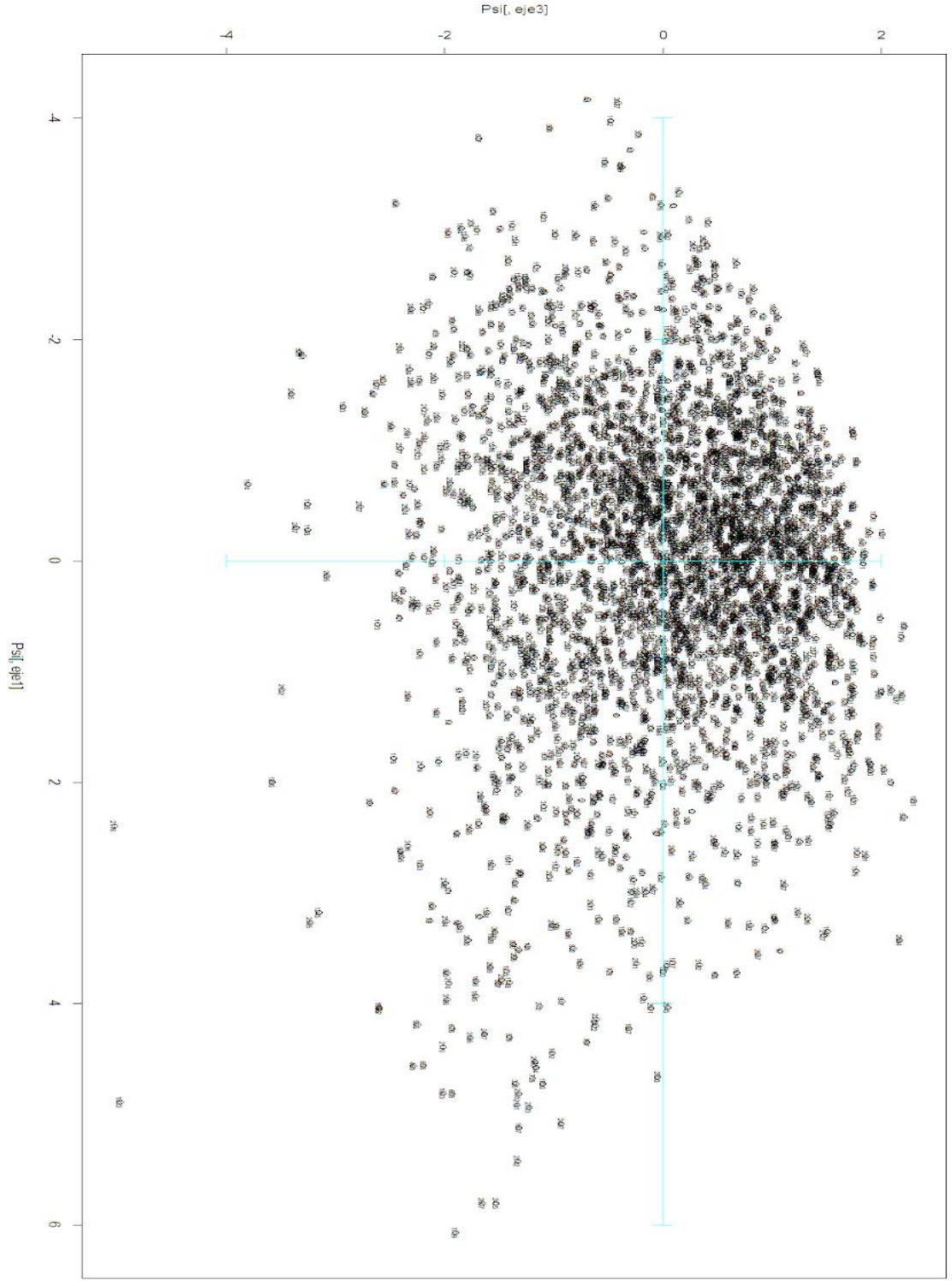
From this we can try to infer more about the modalities we see. We can link the raise in insulin drug administration with patients with a high number of diagnoses, which can also be related to respiratory admissions. Also, the number of procedures seems to be quite closely related with circulatory and injury problems.

There seems to be a relationship between high-weight patients and african-american patients, which also seem to be the ones that are readmitted the most (lower in the vertical axis).

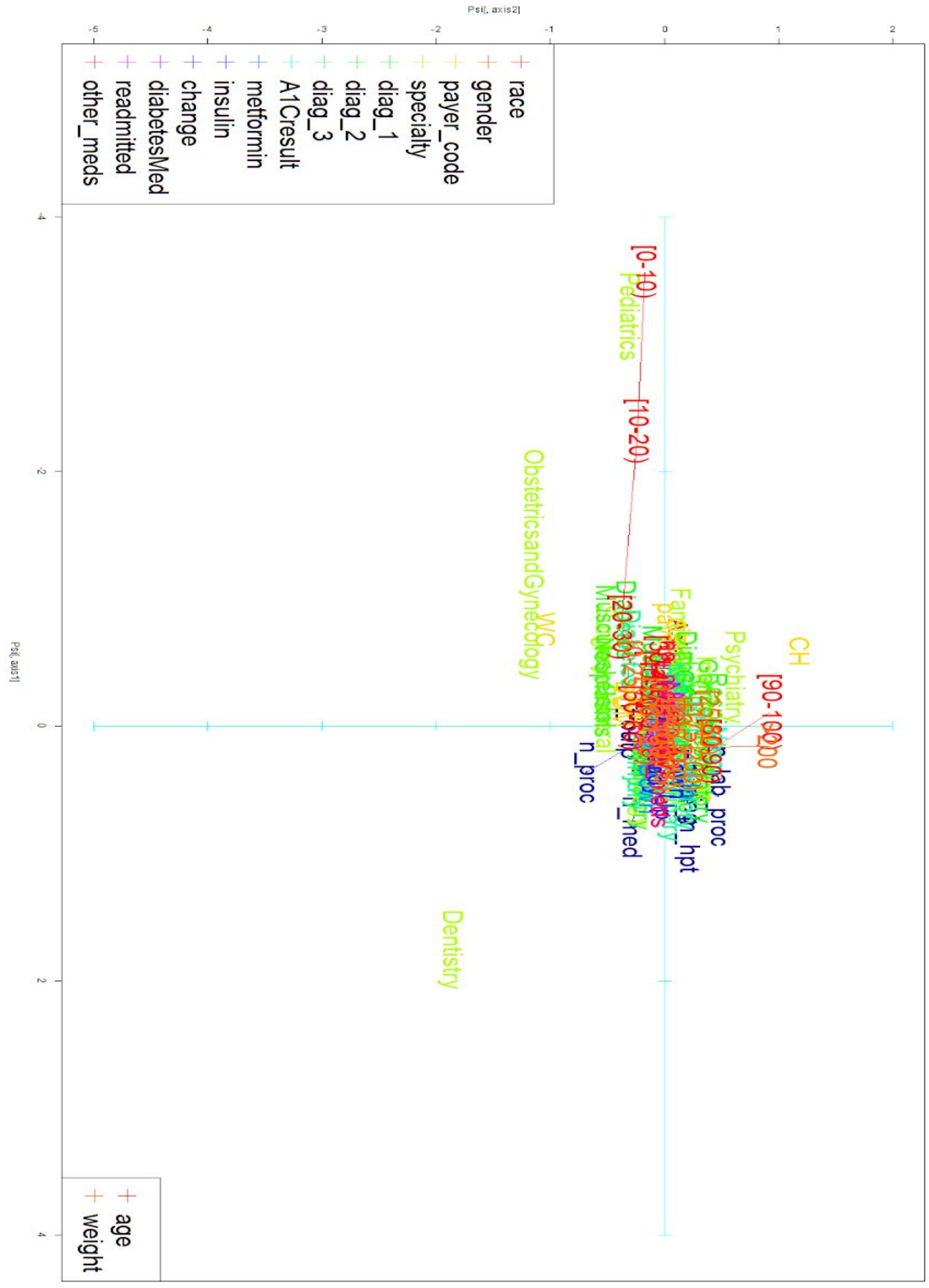
Now, focusing on our ‘target’ variable for study, which is the readmission possibility, we notice a distinct behaviour, which seems logical. The readmission closeness in time seems to be proportional to the second axis (AKA, inversely proportional to the ‘reincidence rate’). Furthermore, reincident patients (those belonging to modality <30 and >30) are both on the right side of the graph, while non-reincident patients are located to the left. We can understand that as being related to the ‘gravity’ of the admission cause. More grave problems are somewhat related to readmission, be it in short or long term, while less grave problems (with less procedures, medications, time in the hospital…)

Heavier weights (125-150) seem to be connected to readmission, since they follow a progression that nears both <30 and >30 values for readmission.

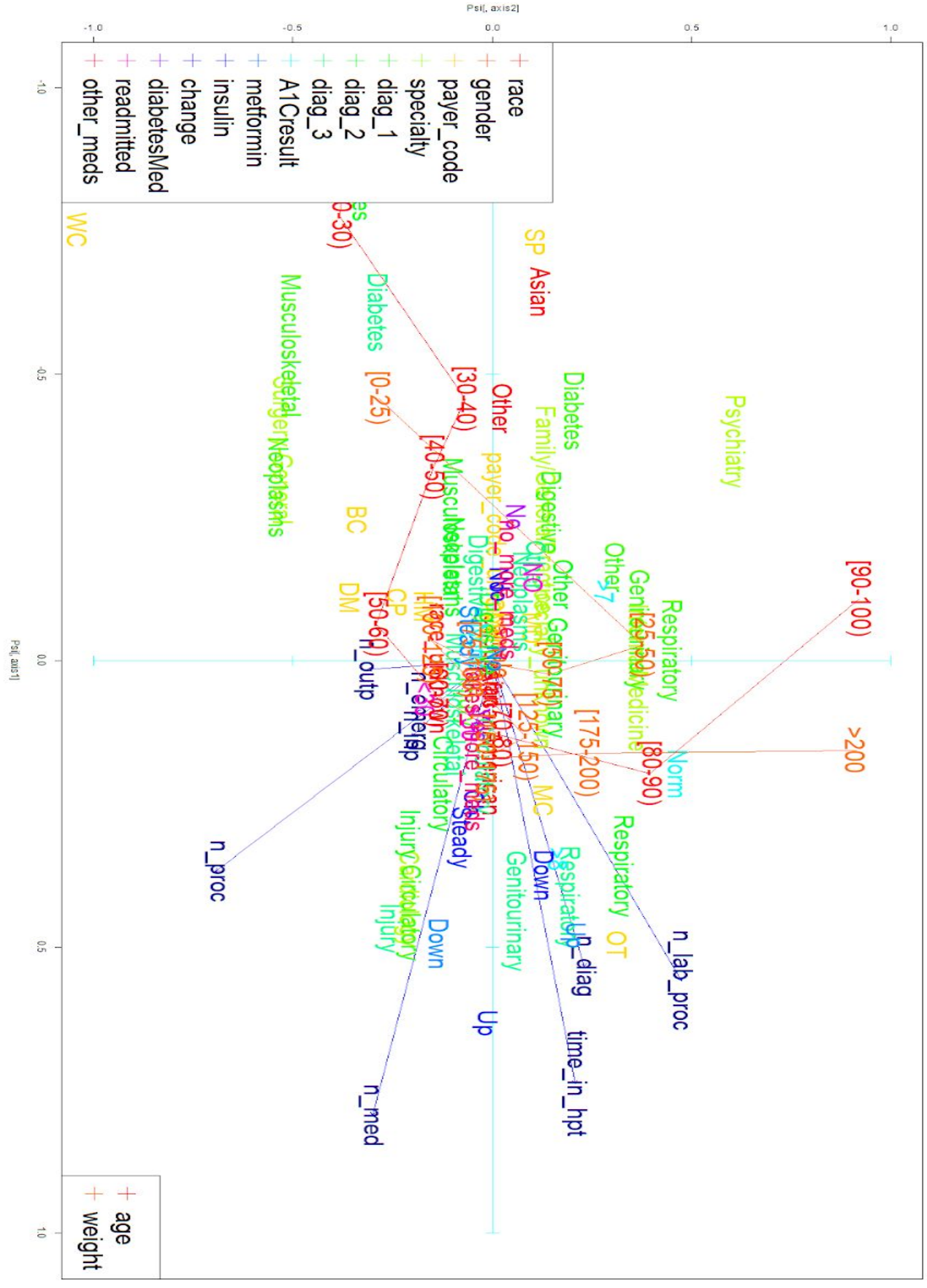
Axis 1 vs 3



Individuals plot



Common projection of variables



Common projection of variables (Zoom)

Interpretation

As before, our axis 1 (horizontal) is once again apparently connected to the ‘gravity’ of the admission.

The other axis explanation (3, vertical) is more difficult to determine. Its principal contributors are the number of procedures and the number of lab procedures, pointing in opposite senses. However, through the aid of our ordinal variables, we can trace a relationship between older age and greater weight and the axis. Therefore, we might be able to link this axis as something directly proportional to the ‘risk’ of the patient (but not necessarily the risk of the problem for which they were admitted). This risk goes hand in hand with procedures which require of lab procedures rather than standard procedures. Values closer to 0 we link to persons not looking too unhealthy, while values above it are those we deem ‘risky’ and require more complex procedures.

However, this ‘riskiness’ does not seem to be related to readmission rates as we would think. In fact it’s the other way: risky patients are less likely to be readmitted. Analysing it with the help of the second axis, we came up with a hypothesis why this seems to happen:

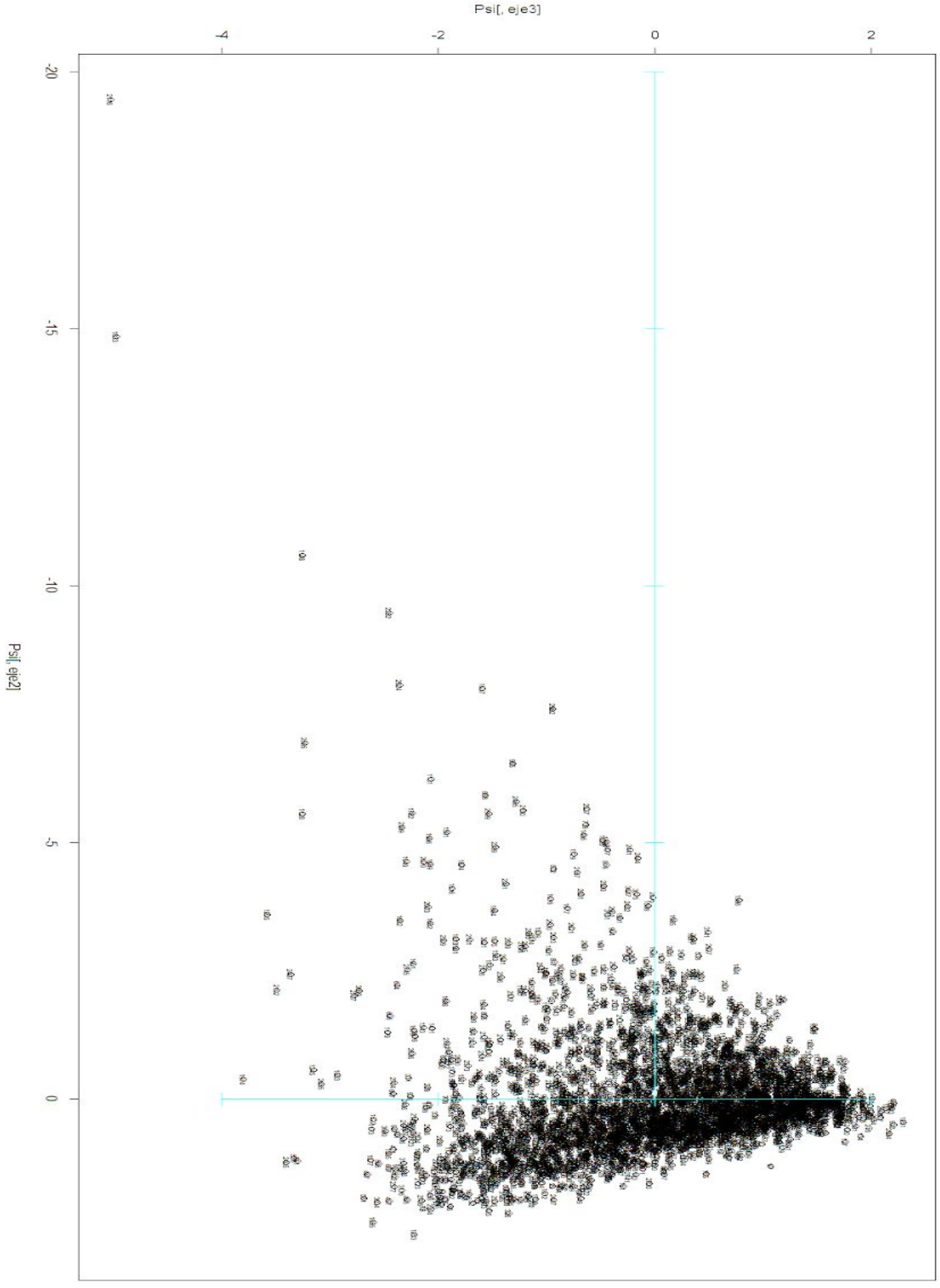
Patients are administered less medication(especially those of old age or lower weight (between 0 and 50 pounds)), are not likely to be readmitted. Those patients are also not kept on the hospital for long, nor undergo many procedures. From this, we have can link it to either not being a severe problem and thus did not need medication to be cured.

There seems to be some difference between taking and not taking medications (metformin and insulin included), as they are at a distance. Of these, we can link not taking medications to a lower chance of being readmitted, while those that do take more medications have a higher chance of being readmitted (specifically on the long term, seeing the closeness between >30 and takes_more_meds). This goes in line with what we just said before.

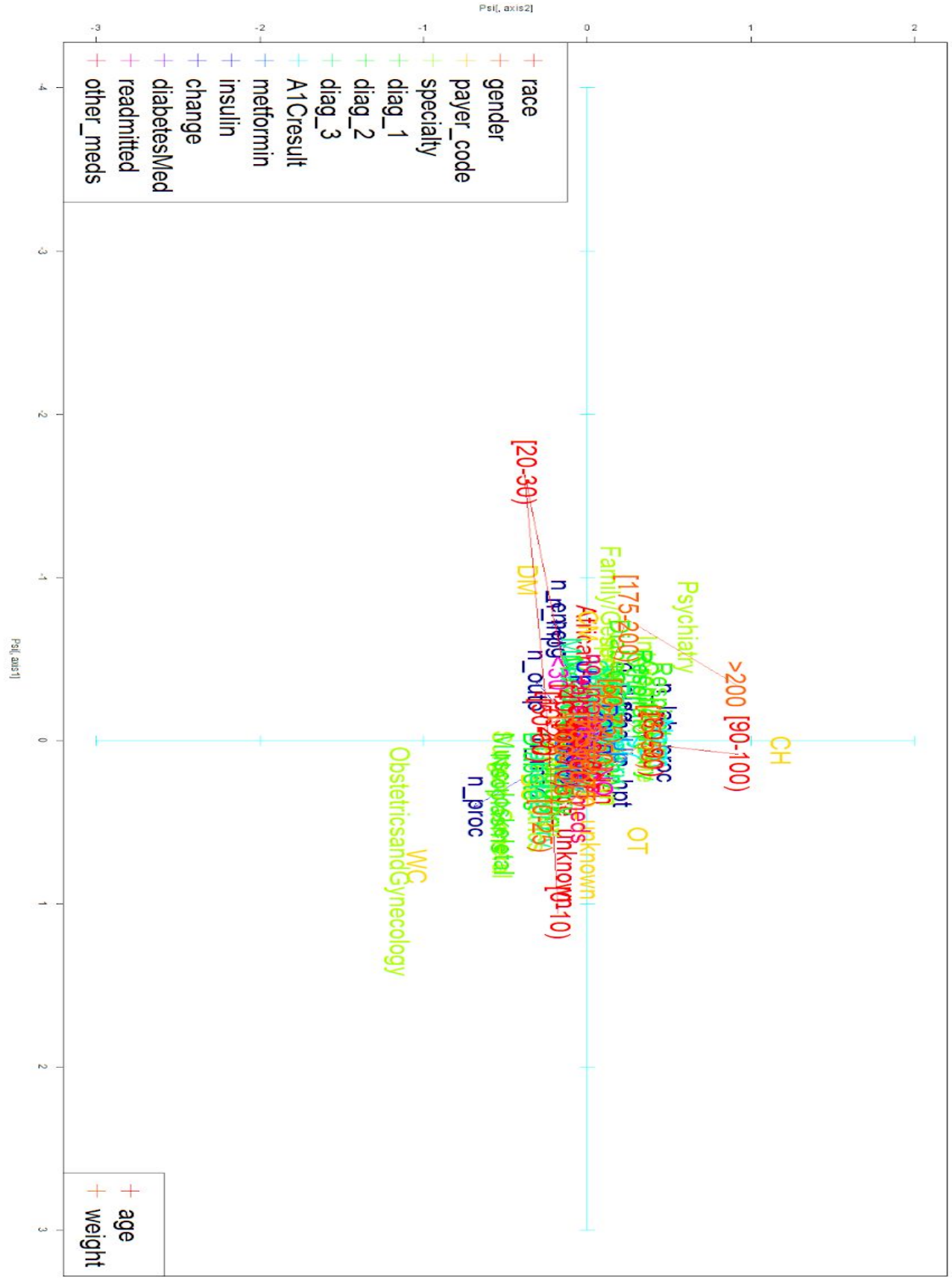
We are not surprised we once again can link the respiratory admissions and the number of diagnosis though insulin does not go up (in fact it’s closer to being lowered) and metformin is used instead.

There seems to be a connection between being slightly old (60-70), being readmitted in less than 30 days, the number of previous visits in the year preceding the encounter (both emergency, inpatient and outpatient). This once again goes in line with our previous discovery on the first plot, aside from the age which previously we saw as unrelated.

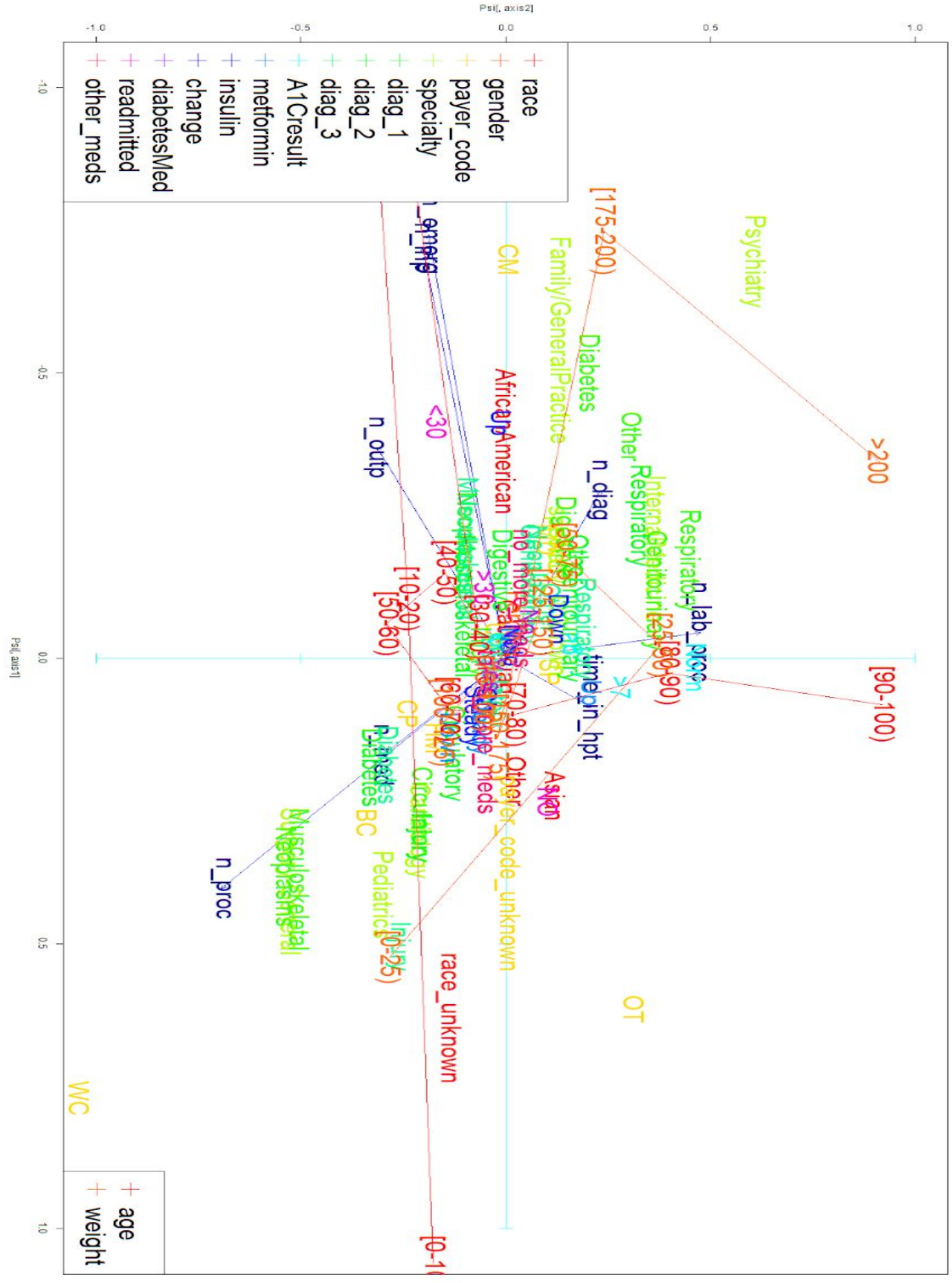
Axis 2 vs 3



Individuals plot



Common projection of variables



Common projection of variables(Zoom)

Interpretation

Both axis have already been interpreted previously, the horizontal one being axis 2 (reincidence) and vertical being axis 3 (riskiness and lab vs normal procedures).

Having diagnosis of diabetic problems is strongly connected to the number of meds administered during the encounter and the number of procedures performed during the encounter.

There seems to be a strong connection between being of the asian race and not being readmitted.

There's a connection between being of the african american race and having the dosage of insulin increased.

Here we can also link pediatrics with younger age, since it is crossed by the line from 0-20 years old.

Being readmitted, both after less than 30 days and after more than 30 days, is strongly related to the number of emergencies, inpatient visits and outpatient visits in the year preceding the encounter (our so called 'reincidence' from the 1-2 plot); the more of these, the more likely to be readmitted, especially so with being readmitted soon. Also, not being readmitted is inversely proportional to the reincidence, as one would expect.

There's not much more we can extract from this last plot as expected, since the expected information goes down as we increase the axis' index. We decide to stop our analysis at this point, as further plots could give misinformation, and we will trust the first plot the most of all.

Hierarchical Clustering

Data description

In order to perform the hierarchical clustering, we started from the outcoming data from the preprocessing step. Since not every variable was adequate to be included in this step, we did the following modifications to the dataset.

First of all, the identifier variables had to be removed from the analysis. To do so, we discarded the row identifiers, and variables *encounter_id* and *patient_n*. The next variable that we discarded was *payer_code*, since it is a factorial one with a high percentage of unknown values. And the last variable removed was the response variable, named *readmitted*.

Once we applied all these modifications, the dataset that had to be analysed was formed by these variables:

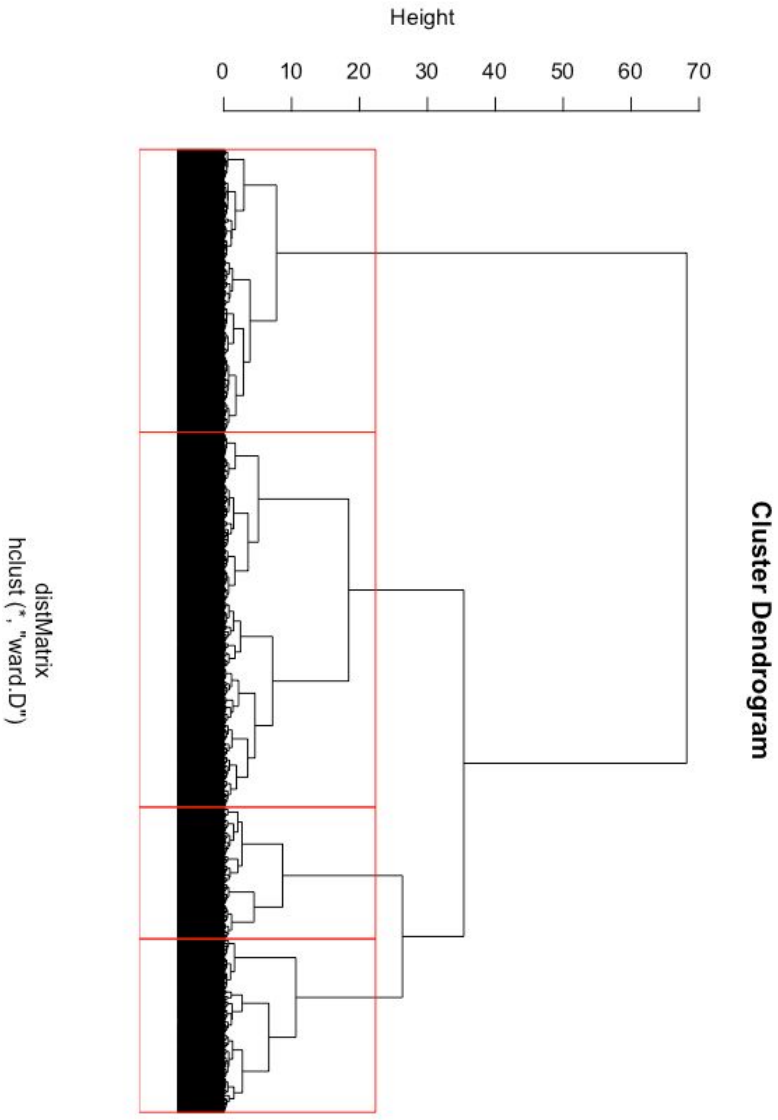
race	gender	age	weight	adm_type_id
disch_id	adm_source_id	time_in_hpt	specialty	n_lab_proc
n_proc	n_med	n_outp	n_emerg	n_inp
diag_1	diag_2	diag_3	n_diag	A1Cresult
metformin	insulin	change	diabetesMed	other_meds

Clustering method used, metrics and aggregation criteria

The clustering was performed using an ascendent hierarchical method. In order to obtain the distance matrix of the variables, Gower's metric was used, since we are dealing simultaneously with numerical and qualitative data. Regarding the aggregation criteria, we used Ward's method, which groups classes giving minimal inter-class inertia loss.

Resulting dendrogram

After running the hierarchical clustering algorithm with our data, here is the resulting dendrogram, divided in 4 clusters. The rationale behind this decision is explained in the next section.



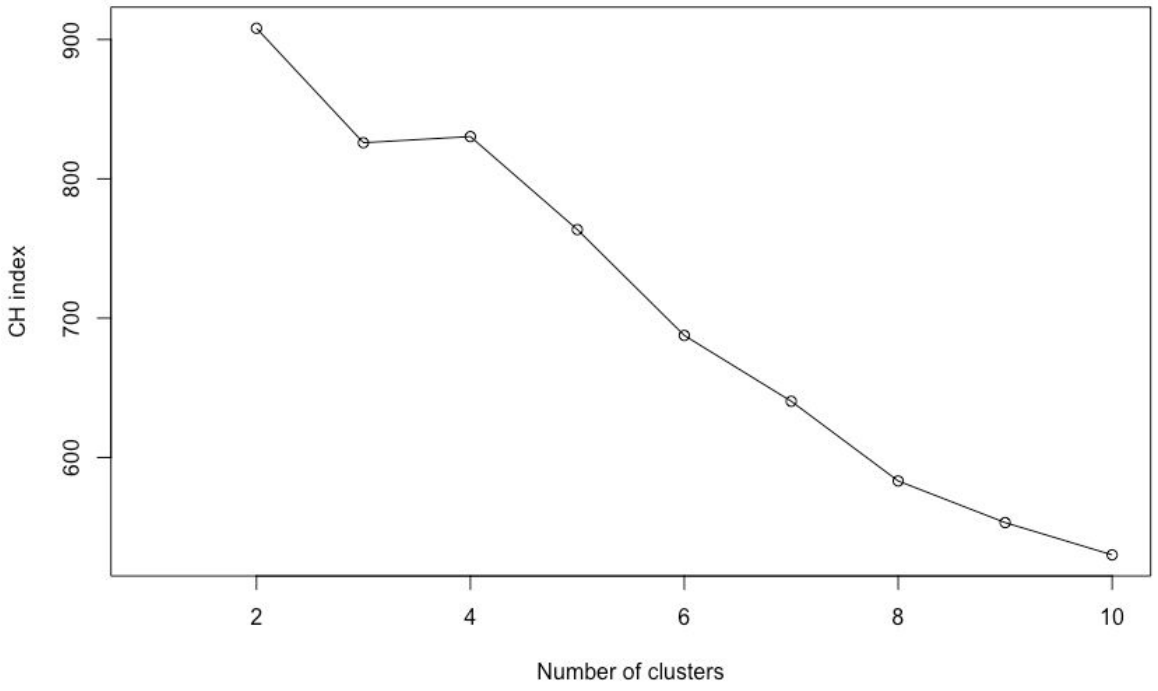
Discuss about how to get the final number of clusters

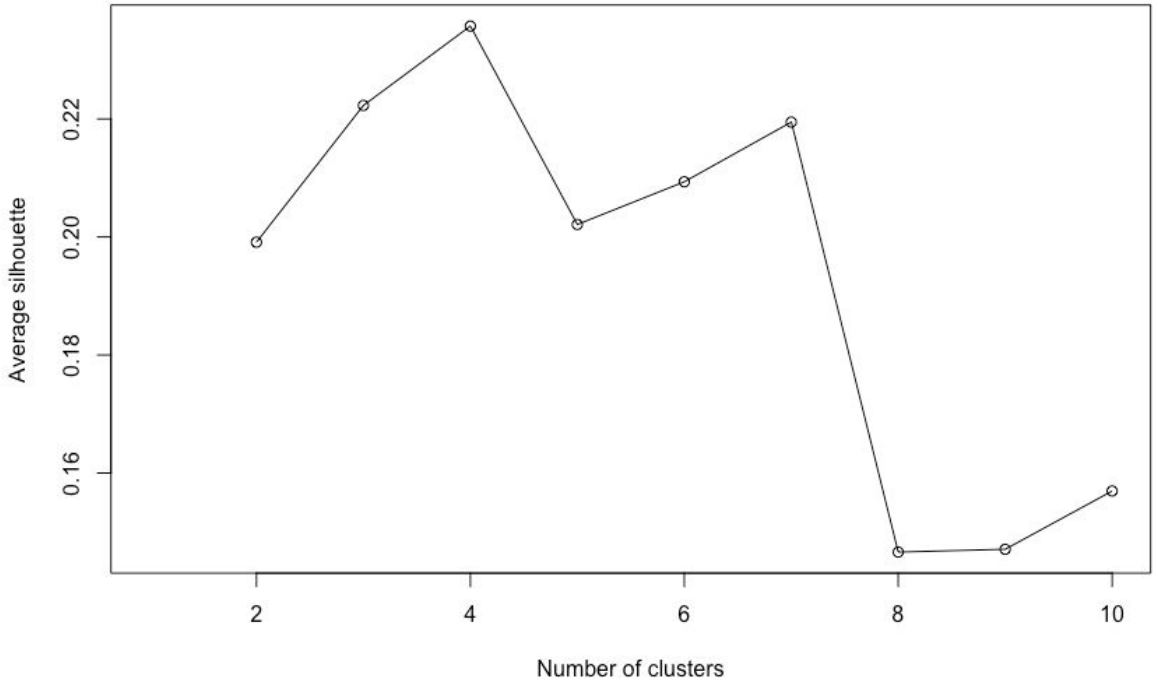
In order to cut the dendrogram, our first approach was to look at its morphology, selecting the cut with the longer vertical branches. However, we wanted to find some metrics that helped us with that decision, gaining insight on the best possible solution.

For that purpose, we have selected two different measures. The first one is the Calinski-Harabasz Index, and the second is the average Silhouette.

The **Calinski-Harabasz Index** is a between-cluster and within-cluster variance ratio criterion giving some insight into the structure of the points. The **Silhouette** value is a measure of how similar an object is to its own cluster, compared with other clusters. By obtaining the average Silhouette of all the data, we obtain a measure of how tightly grouped the points in the clusters are, determining how appropriately the data have been clustered.

For both of the measures, a high value means that the number of clusters is well chosen. Thus, we calculated those metrics for different partitions of the dendrogram, going from 2 to 10. The results were the following plots:





We discarded the 2 clusters partition, since it is not informative enough. For both metrics, the next maximum value was set at **4 clusters**. This confirmed our first thoughts observing the dendrogram, so finally we picked this partition.

Table with a description of the clusters size

After dividing the dataset into four clusters we analyzed those clusters as following. First of all, we compared the size of those. Having into account that after the preprocessing we had a total of 3197 observations, a balanced cut would have about 800 observations each cluster. However, the results are the following:

Cluster	**1**	**2**	**3**	**4**	**Aggregate**
#Observations	437	939	1245	576	**3197**
Percentage	13.67%	29.37%	38.94%	18.02%	**100.00%**

Appart from the size, and before passing into the profiling analysis, we decided to go one step further on the cluster division and compare all representative variables with each other taking into account the cluster. To do so we generated a 25x25 grid image with the comparison of all variables. The result was a huge image which we couldn't open with

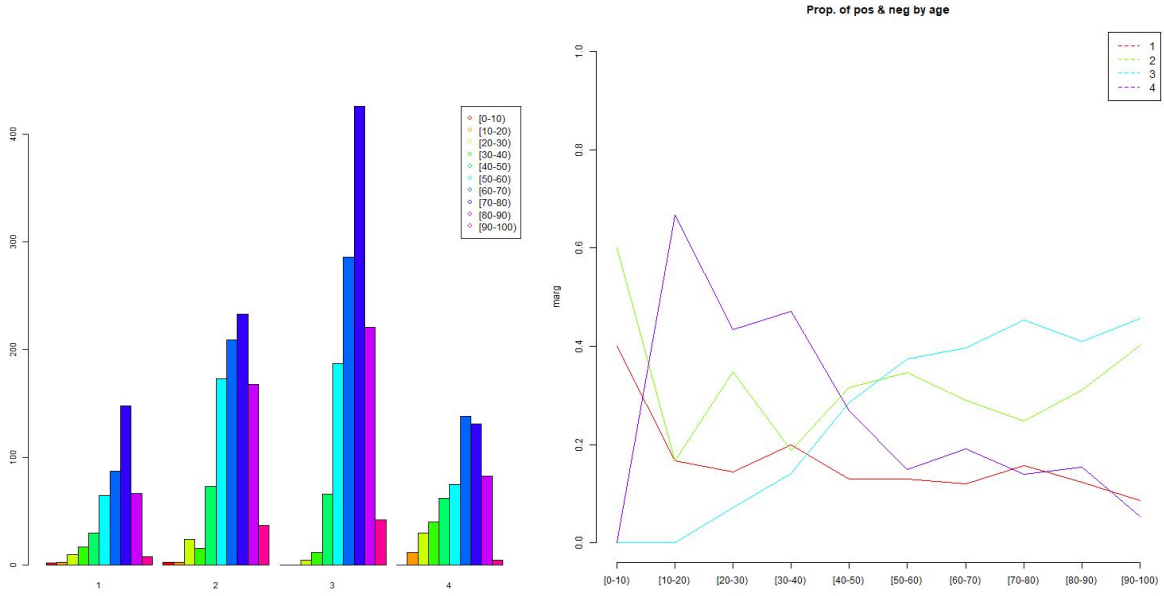
standard image viewers. After analyzing the image with online image viewers and realizing that because many of our variables are discrete our comparison couldn't be complete we took the ones that could be more interesting to study and generated another plot with those. It is the following. Again, many of those cannot be analyzed properly because they are discrete variables and result in predictable conclusions. A deeper analysis is done on the next section.

Profiling of Clusters

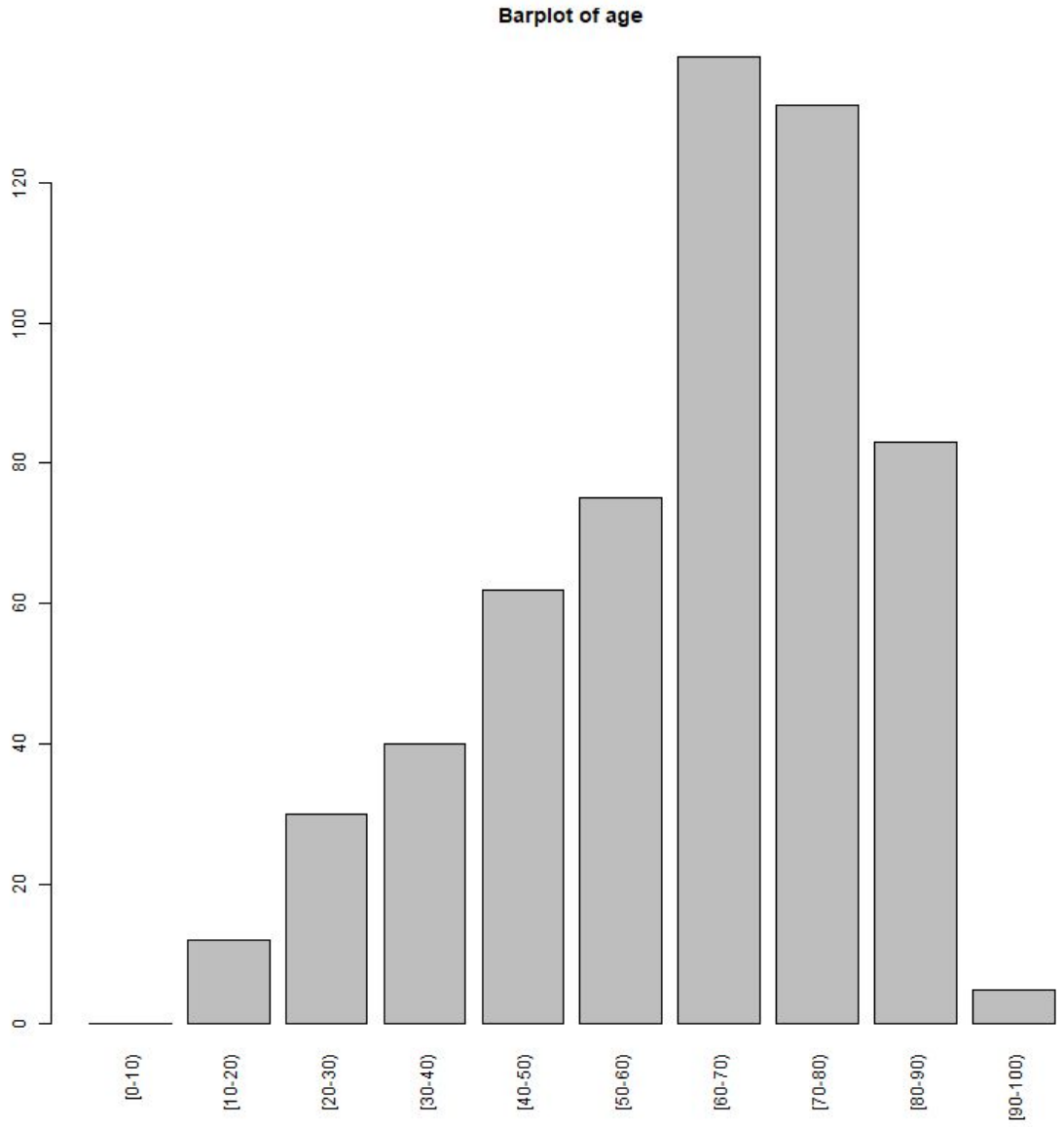
After computing the clusters we need to compare them in order to extract relevant and meaningful conclusions. For this, we do the profiling graphs and CPGs of the variables used for the hierarchical clustering plus the response variable, *readmitted*.
Next, we will discuss the most relevant graphs.

Profiling plots

Age

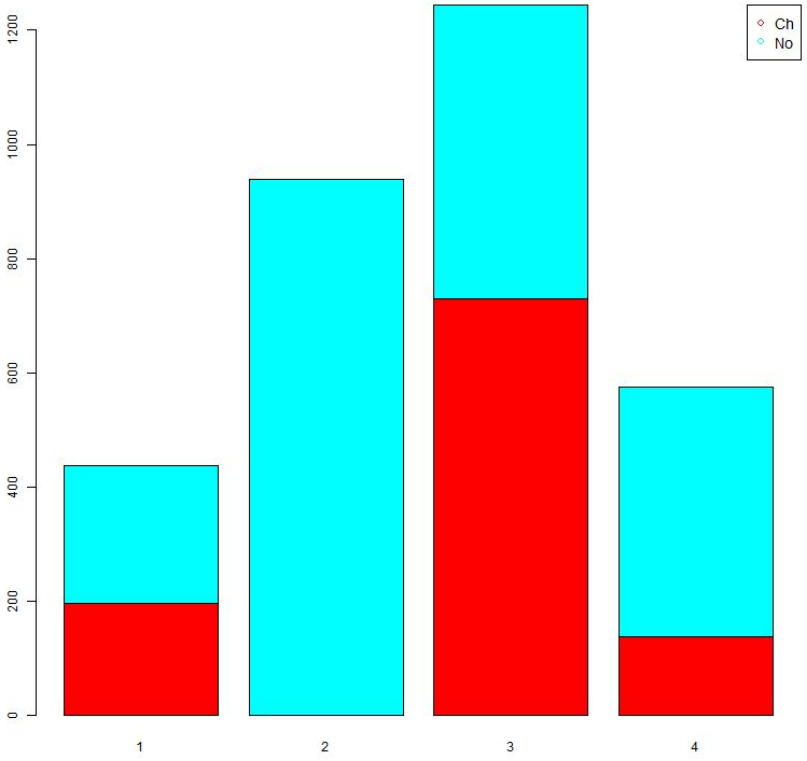


The first relevant variable that we find is age. The proportion of young people (not kids) is bigger in the fourth cluster, while the third cluster has the most percentage of old people. Meanwhile, the first and second have all the kids and are more well-distributed.
This means that we can begin to see the differences between the clusters, and with the information of the following variables will be able to draw relevant conclusions to the study.

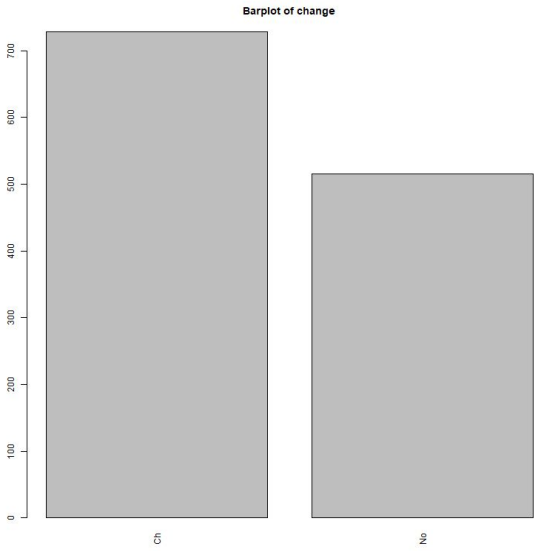


It's important to note that even though the fourth cluster has the most relative percentage of young people (again, not kids) the number of old people is still greater. This is because the number of old people present in the dataset is so much bigger than the number of young people, which makes sense since old people tend to go more to the hospital.

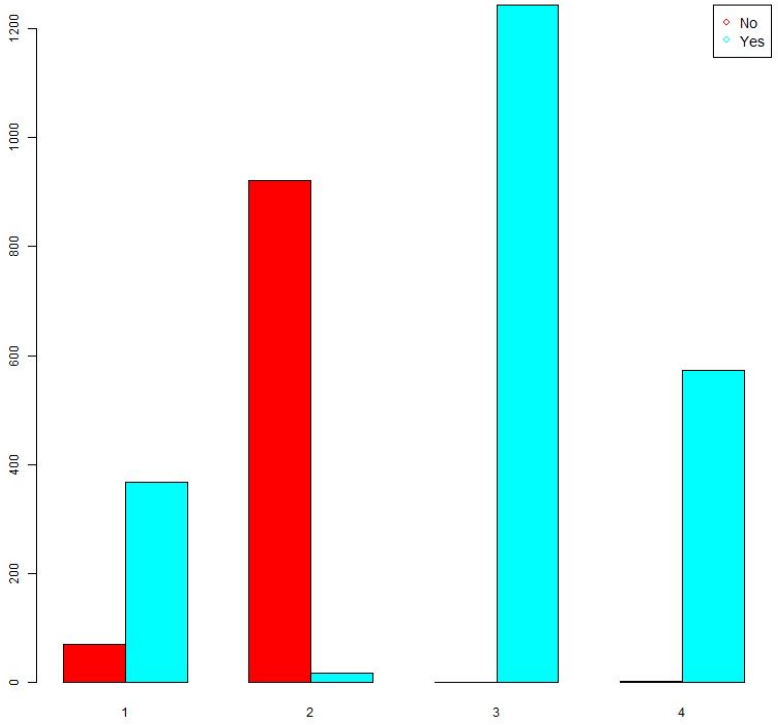
Change

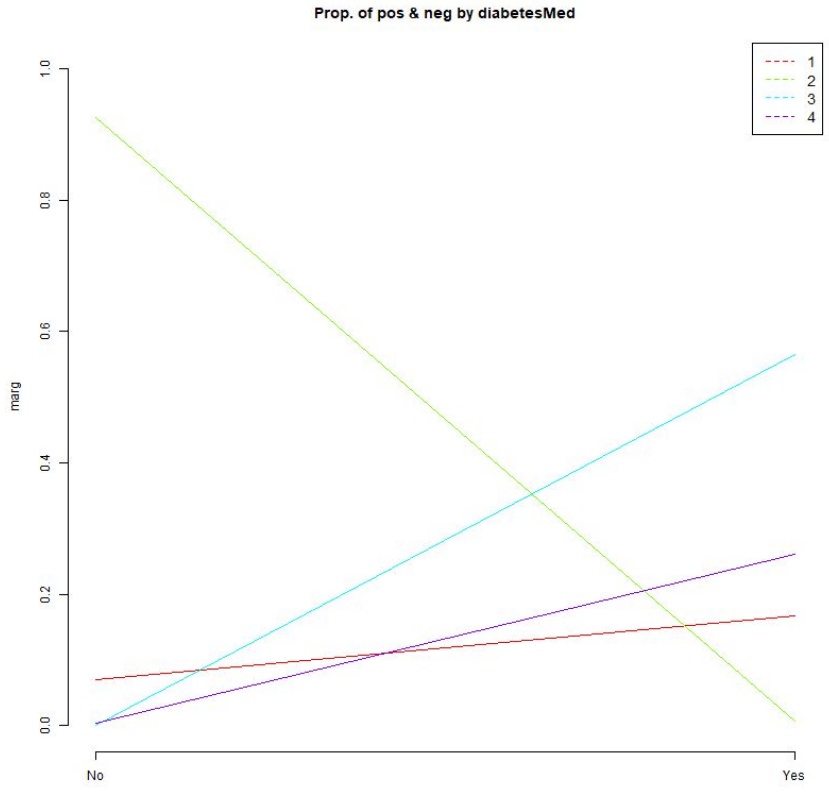


Change, which is meant to indicate if a patient has change of diabetic medication or not, is one of the most interesting variables we have observed. It's obvious from the beginning that the second cluster has those patients that do not have a change of diabetic medication, while the third one is the only that has more patients with a change:



DiabetesMed



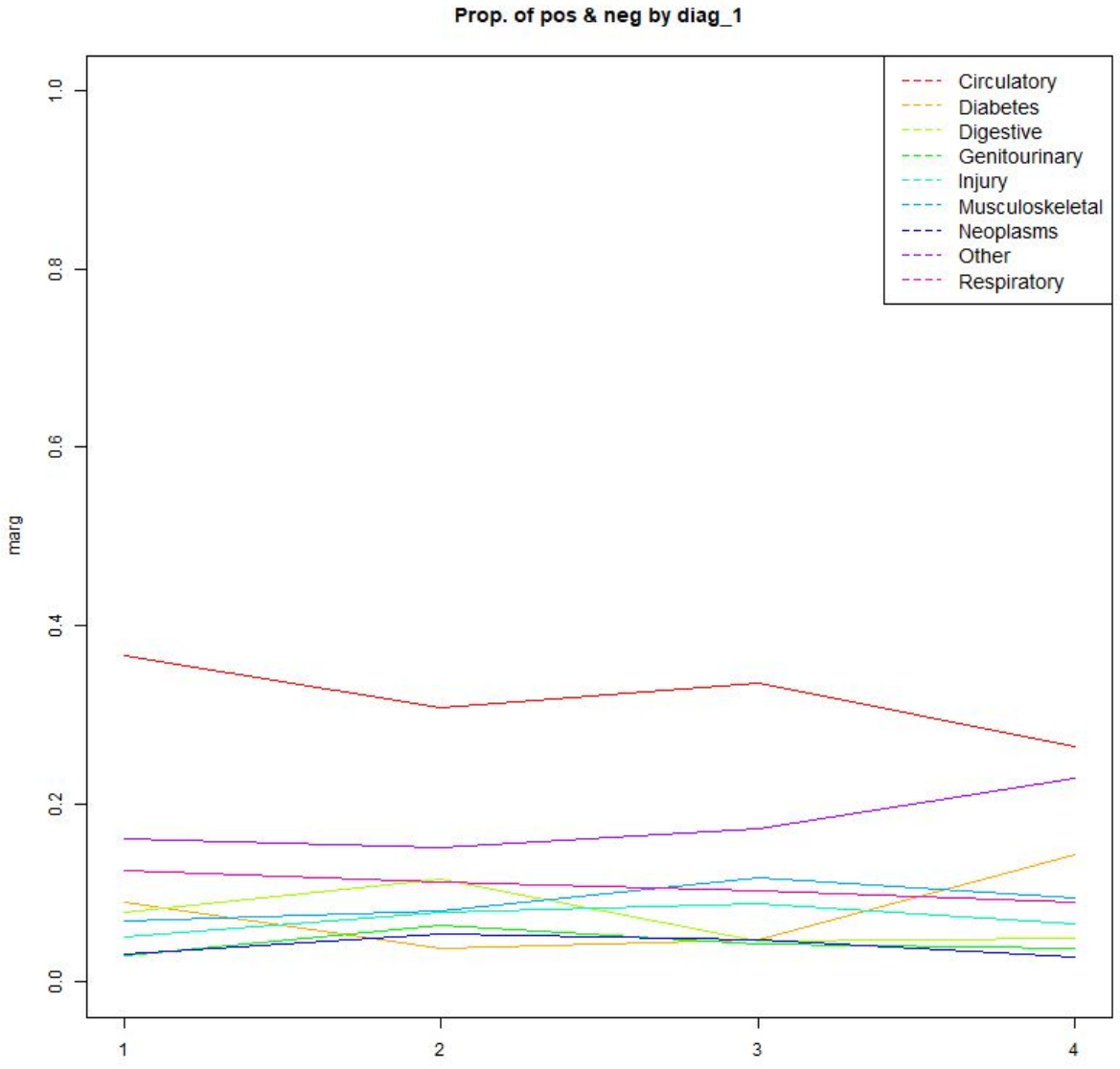


Here we can clearly see another variance between the clusters. The third and fourth ones are almost all patients that were prescribed diabetic medications, while the second cluster is the other way around as seen in the plots above.

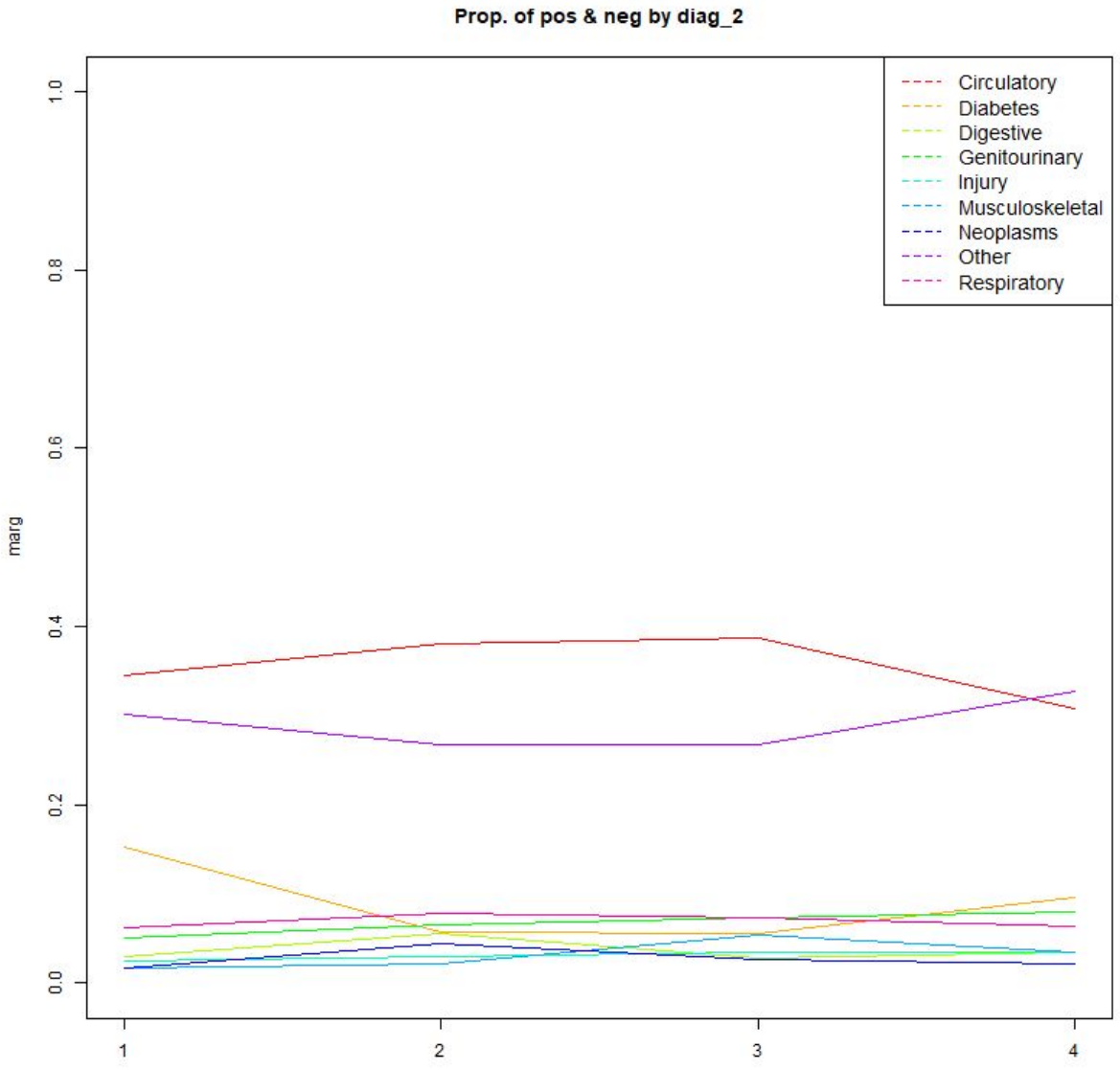
If we combine this knowledge with the analysis of the *change* variable, we can see the first real connection in the clusters even if it was obvious from the beginning; the second cluster is composed of those patients that do not require a diabetic medication, and thus do not need a change in said medication. However, the third cluster presents those patients that need diabetic medication and have needed a change.

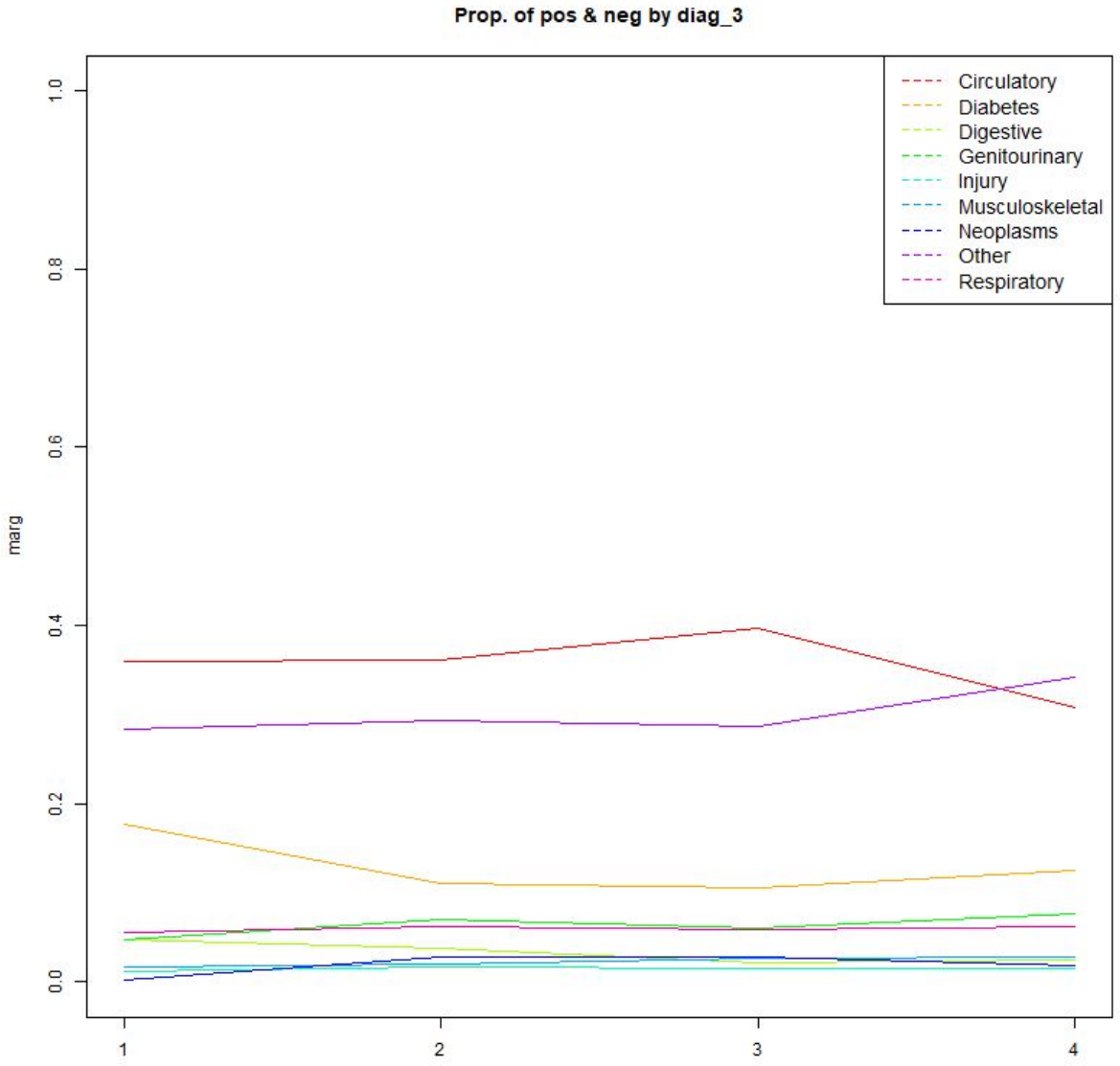
diag_1, diag_2, diag_3 and n_diag

It's important to analyze these three variables at the same time because they are closely related.

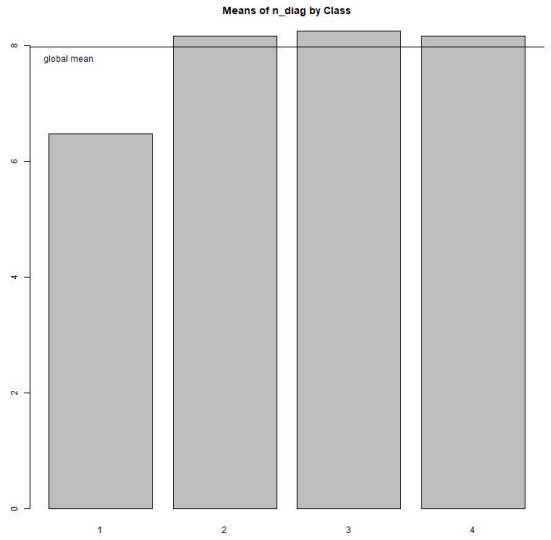


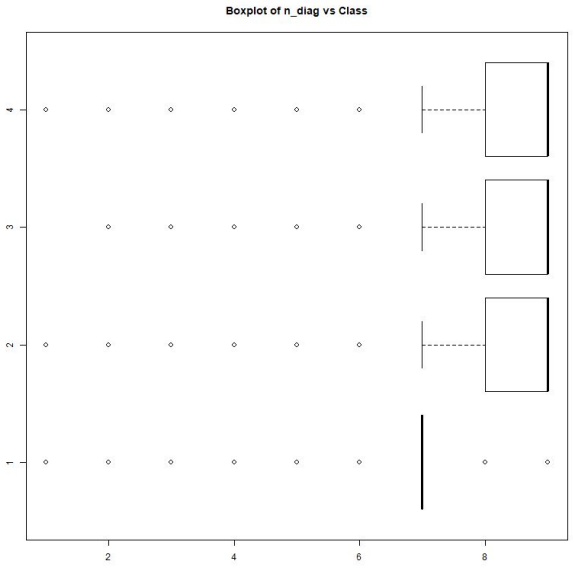
Circulatory is the most common cause of the first diagnosis among all clusters and *Diabetes* is most prevalent in the fourth cluster.





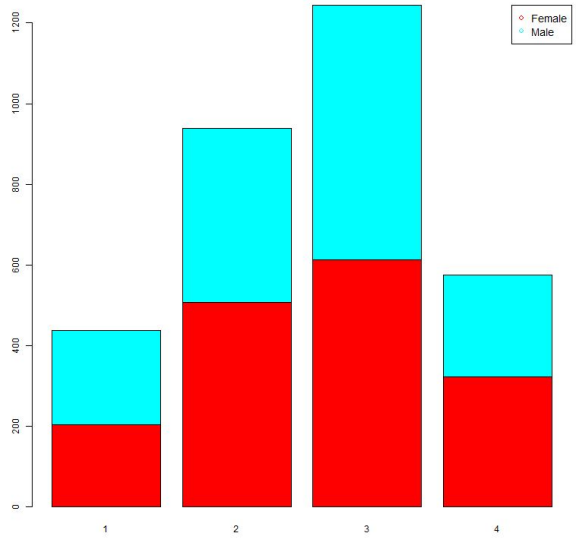
Things are more or less the same in the second and third diagnosis, with the difference that *Diabetes* is slightly more prevalent in the first cluster and *Other* in the fourth this time.

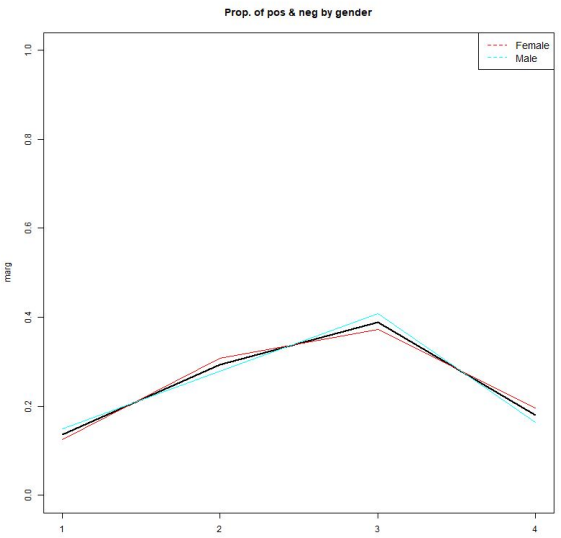




The *n_diag* variable is pretty well-rounded around the 8 and 9 values in three of the clusters. However, in the first is significantly lower, with a value slightly above 6. Since *Diabetes* is stronger in the first cluster in the second and third diagnosis, we could correlate that and say that a diagnosis of diabetes means less total number of diagnosis.

Gender

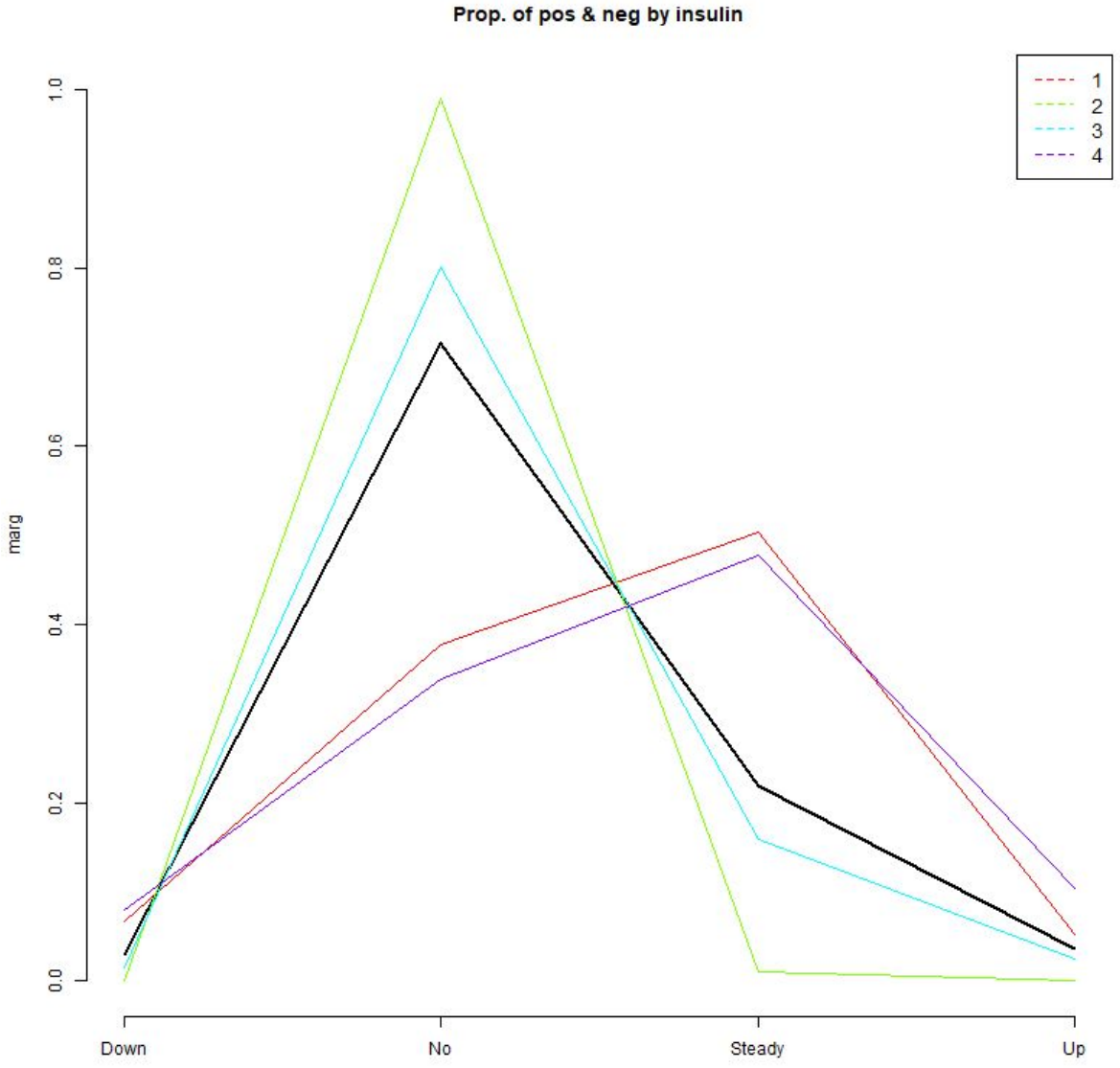




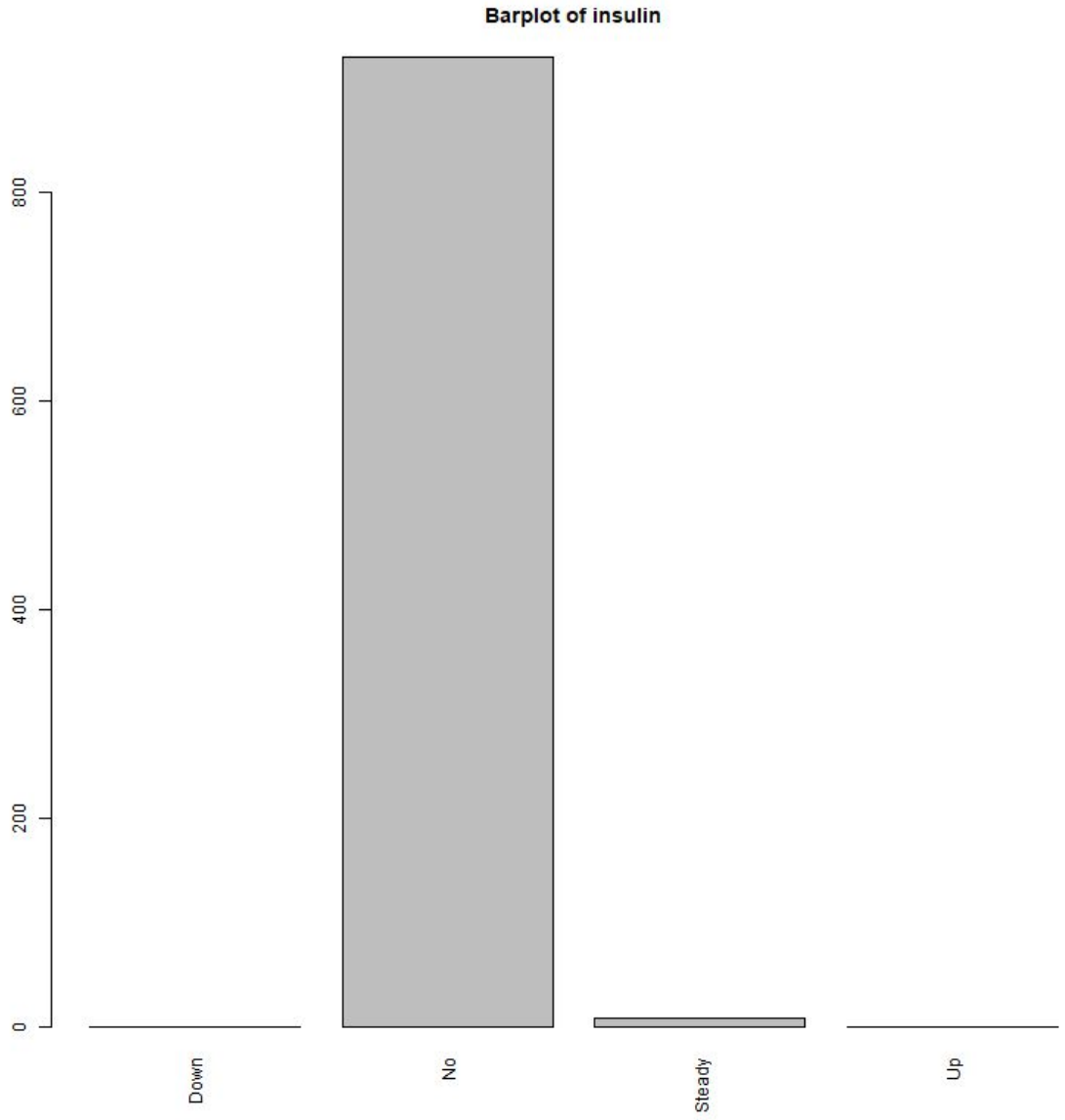
We do not think this variable is significant to the clusters. The only thing worth mentioning is that the third cluster has the most difference between genders (more males than females).

Insulin, metformin and other_meds

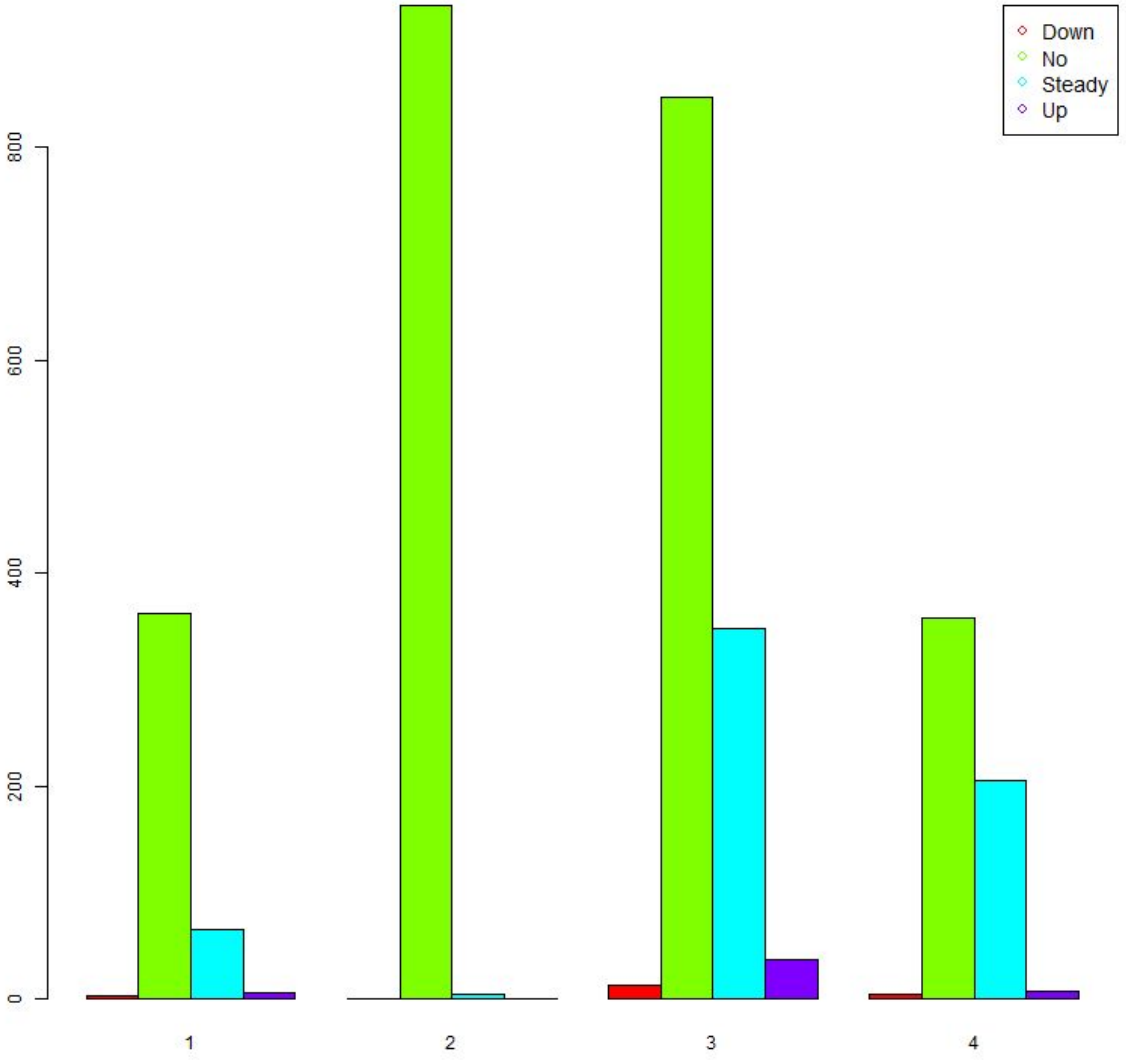
Like the diagnosis variables, these are better explained together. As explained in the preprocessing we grouped all the not-so relevant medicaments in the *other_meds* variables, but left *Insulin* and *Metformin* apart, because they are clearly the most popular.



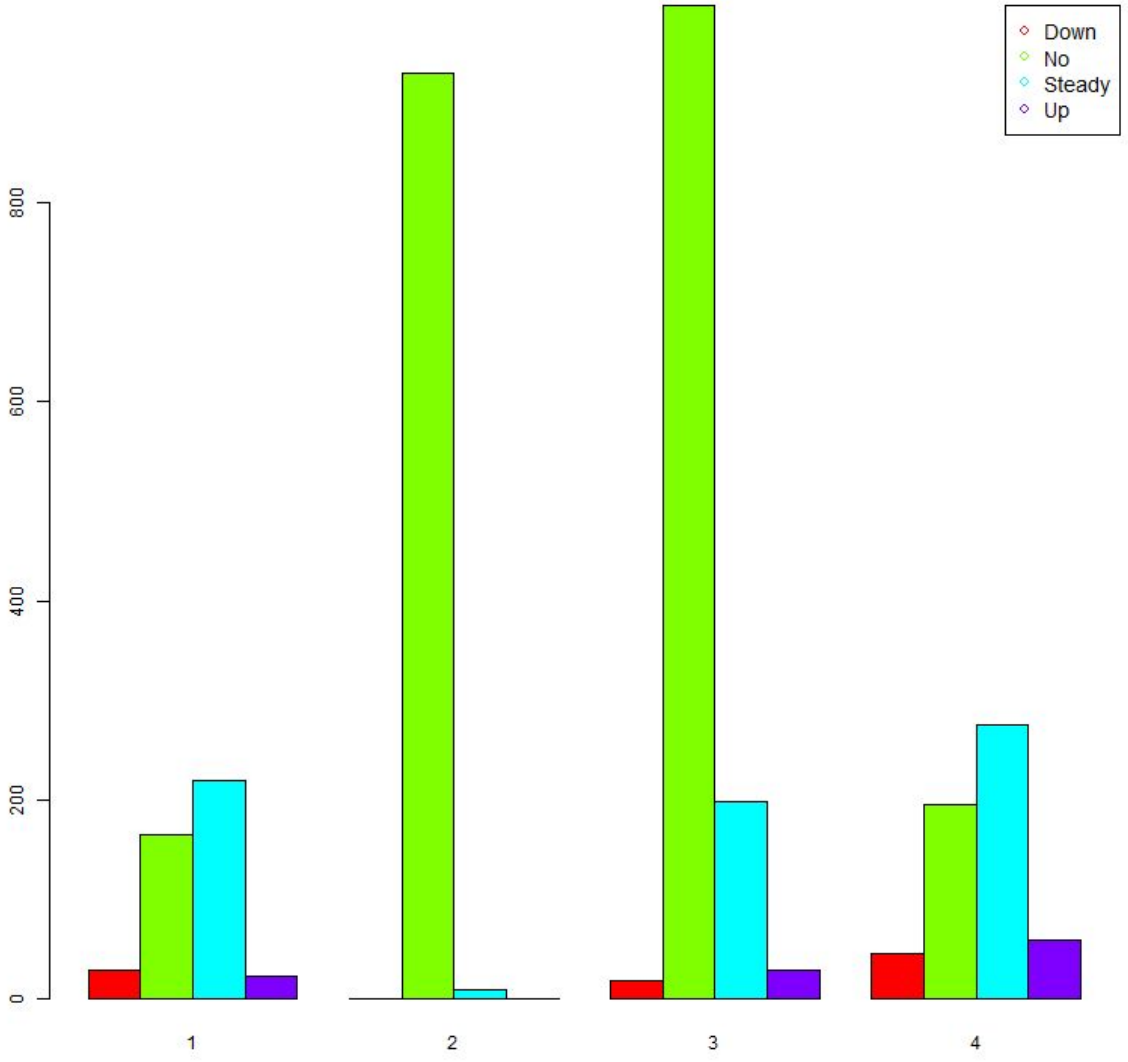
We can clearly see that the first and fourth cluster are really similar, having the most relative percentage of steady and up medication (it's worth noting that in all the clusters most people do not take insulin). What really stands out here is the second cluster, with a really big peak of those who do not get medicated insulin. The third si similar to the second but it is not that big of a peak. In the GPG below we can see the second cluster more clearly:



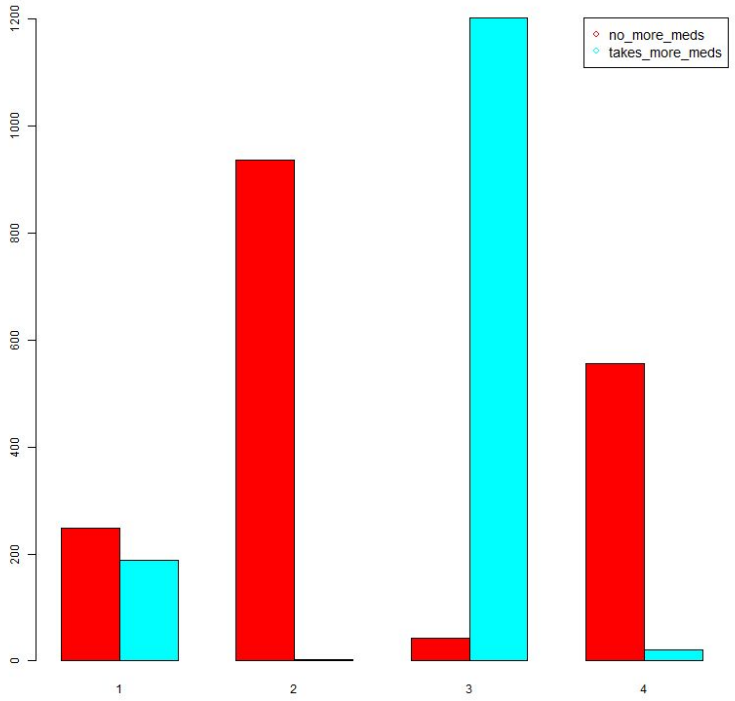
Comparing with the metformin graphs, we see that not a lot of people take it as opposed to metformin, but those who do are grouped in the third and fourth clusters:

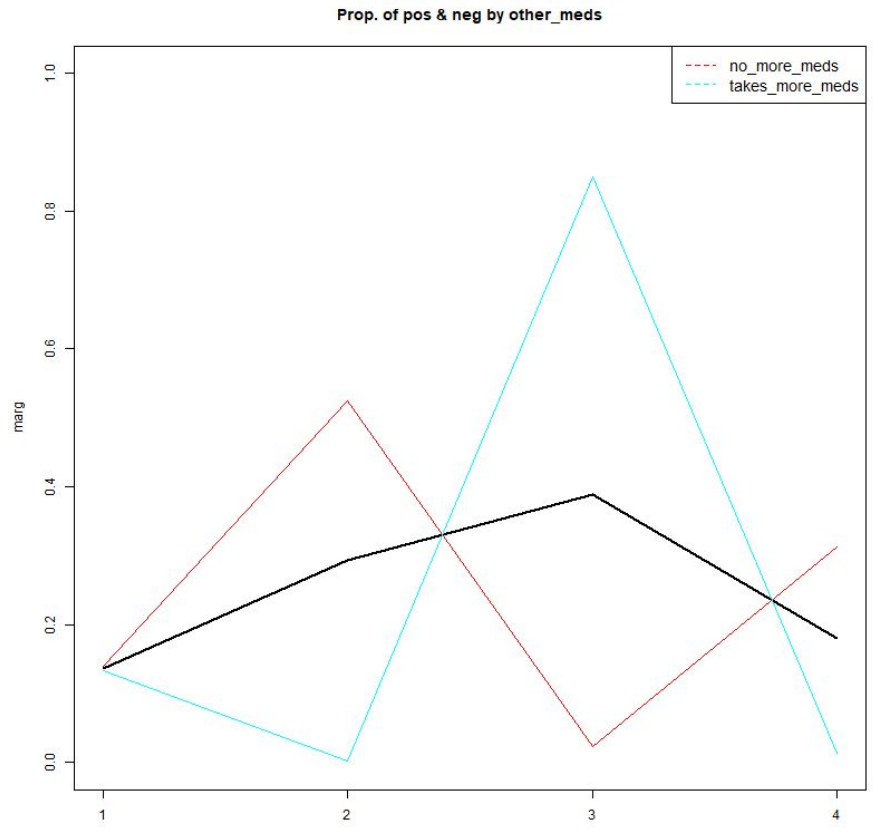


Going back to the insulin, the patients in clusters one and fourth do take insulin:



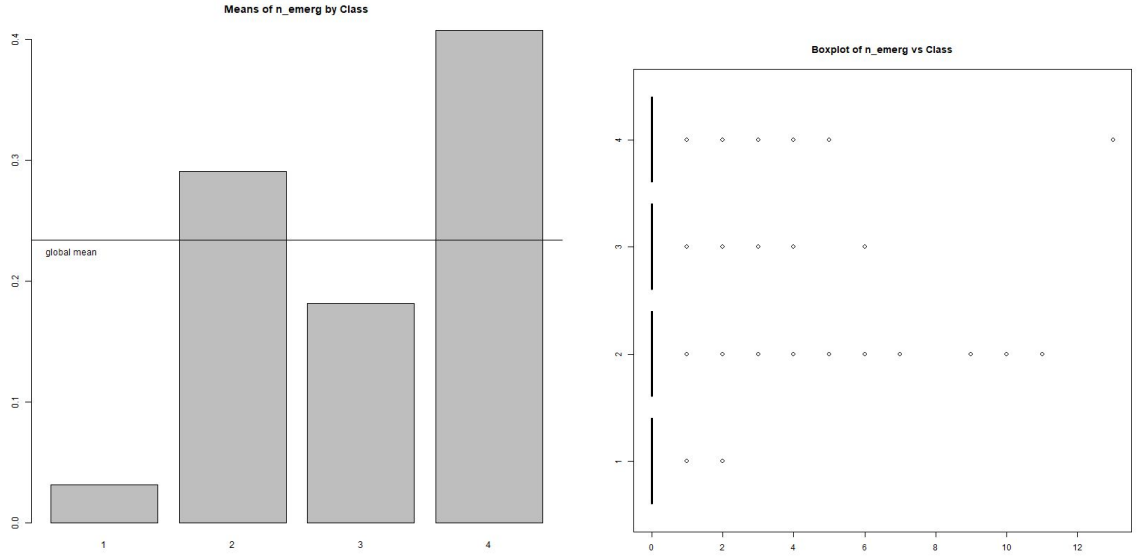
This means that people in the second cluster do not take those medicines, reapproving that they are the patients that do not medicate for diabetes. However, as presented in the *change* and *diabetesMed* plots, people in the first, third and fourth do medicate, more concretely the third cluster is the one where people takes the most medicines, as shown as in the *other_meds* plots:





It also seems that patients in the first cluster medicate more with metformin and other medications while the fourth cluster has people medicating with both insulin and metformin.

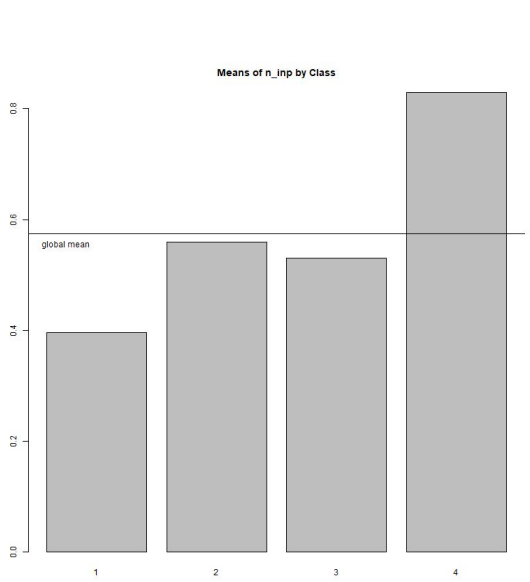
n_emerg

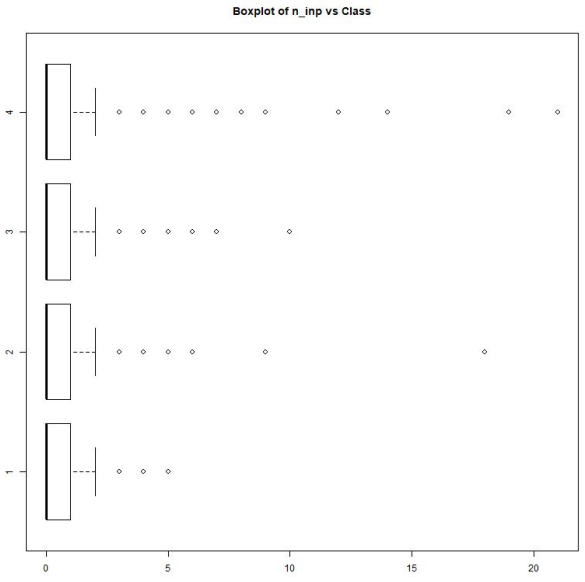


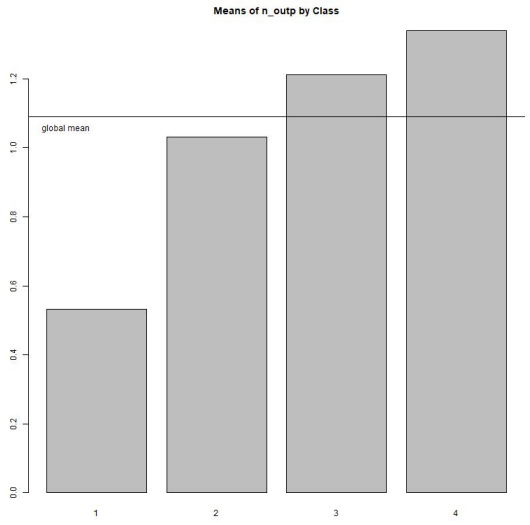
These plots show that even though the vast majority of patients do not have any emergency visit in their record, those that do are in the second and fourth clusters, while patients in the first cluster have the least amount of emergency visits. Since the actual number of people with emergency visits is so small, we think that the only relevant conclusion is that patients in the first clusters are those with few emergency visits (the fourth cluster *has* the most emergencies but there is also an outlier present).

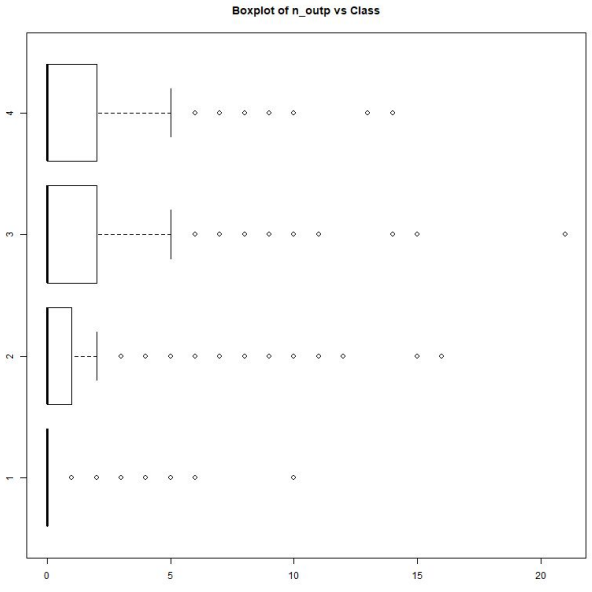
n_inp, n_outp

We will again analyze these two variables together.







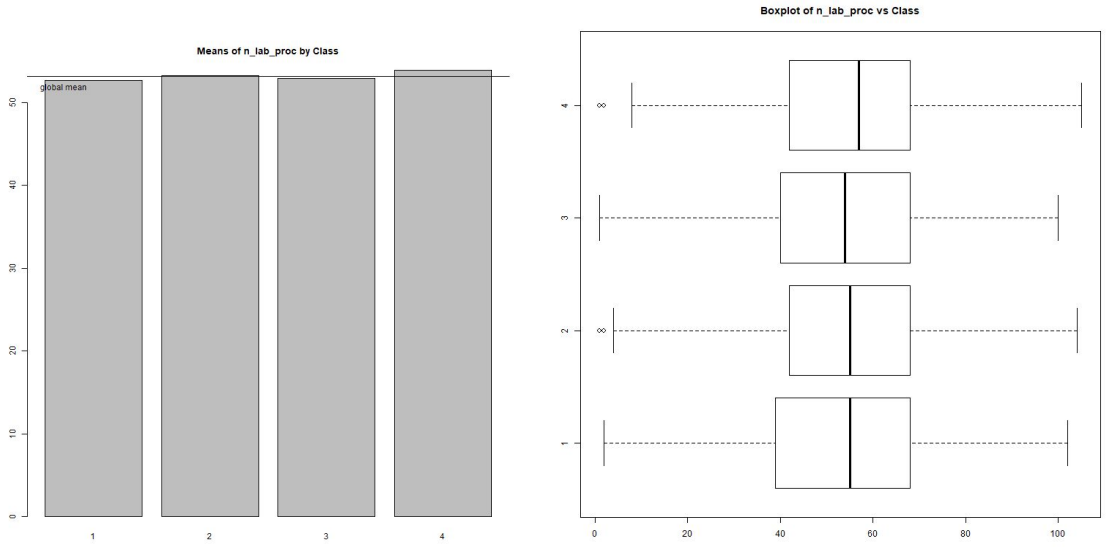


As explained in the basic statistical descriptive analysis, inpatients visits are those that result in hospitalization, while outpatient visits are those those that do not.

At first glance, both variables seem pretty similar, only with minor differences. However, it's interesting to note that the patients in the fourth clusters have the most inpatient and outpatient visits, since they also are the ones with more emergency visits. The same can be said for the first cluster, they have the least amount of visits in all three of the variables.

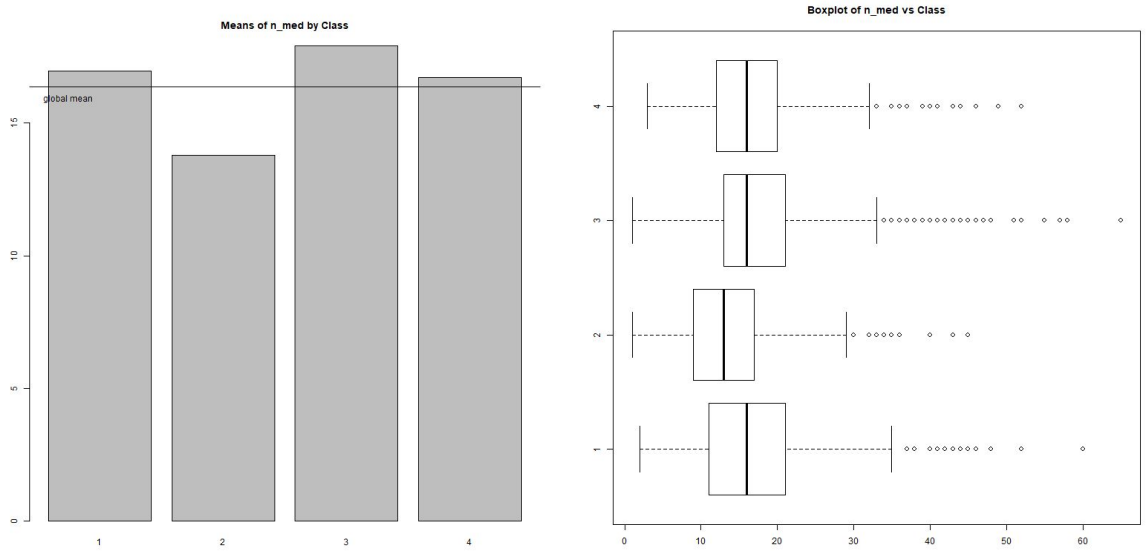
Apart from that, the only notable difference is that the first cluster is the only one with more inpatients visits than outpatients.

n_lab_proc



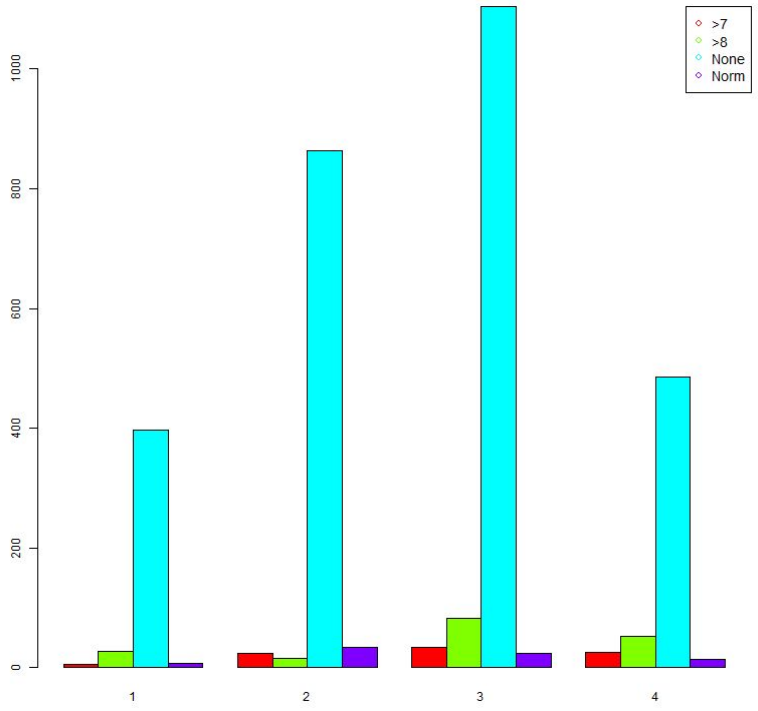
Not much to say about this variable. All the clusters are well-balanced and really similar.

n_med

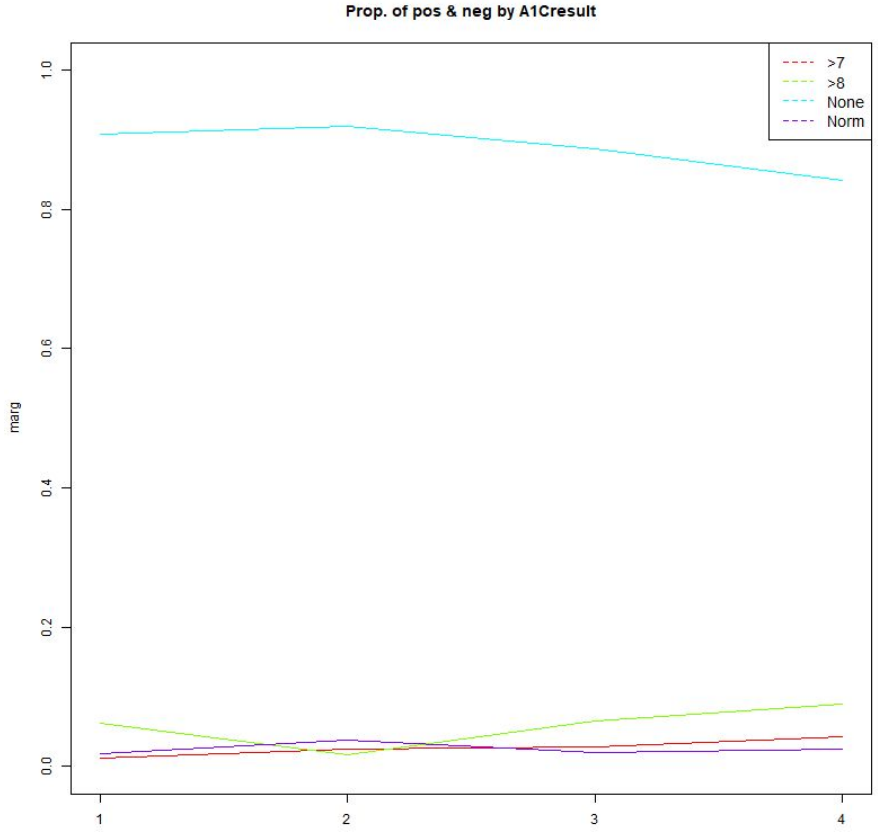


This one also has very similar clusters. The only thing worth mentioning is that the second cluster is the one where patients were administered the less drugs during the encounter, which can be linked to other variables where the second cluster was also the one where patients did not take medications such as *Insulin, Metformin, diabetes_meds* or *other_meds*.

A1Cresult



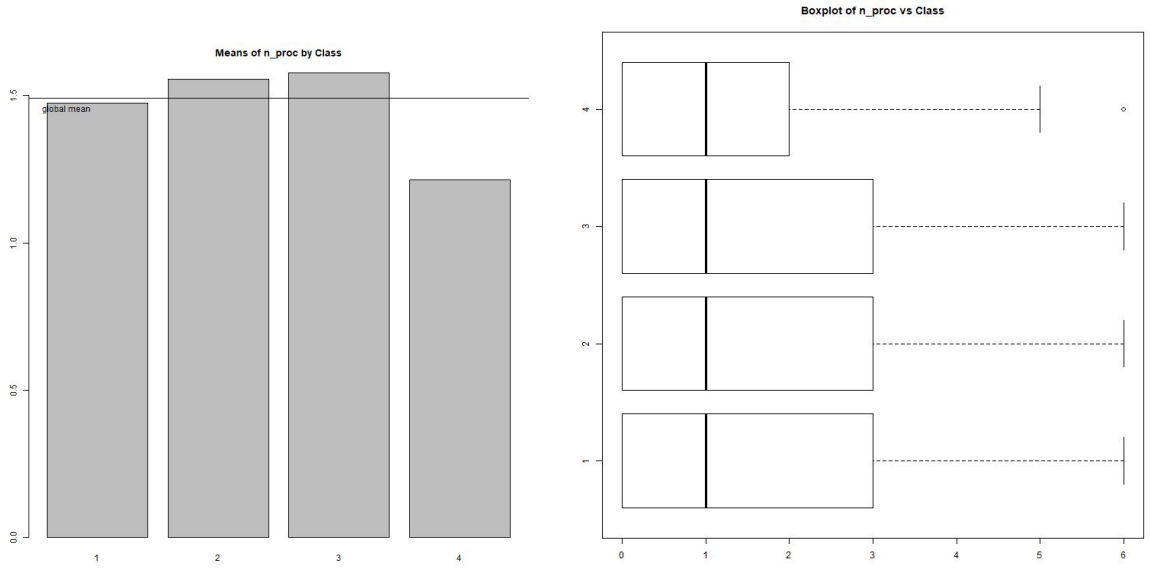
With this barplot we can clearly see that the majority in all clusters is that this test was negative and there is not much deviance with the other results at first glance.



In this other graph we see that the fourth cluster is the one with the least relative percentage of failed test.

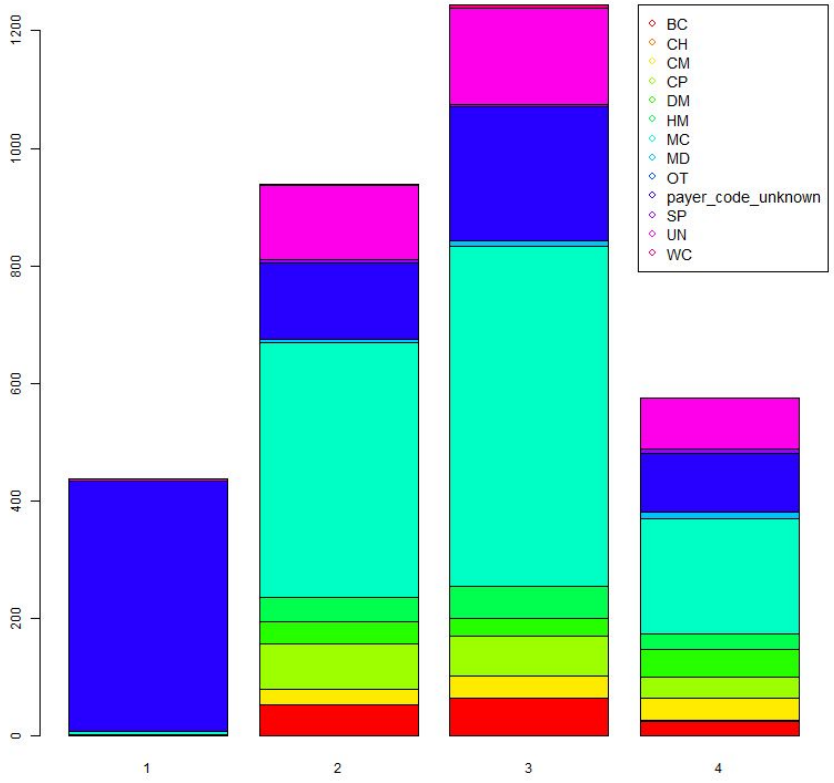
That said we believe that this variable is not relevant to the study.

n_proc

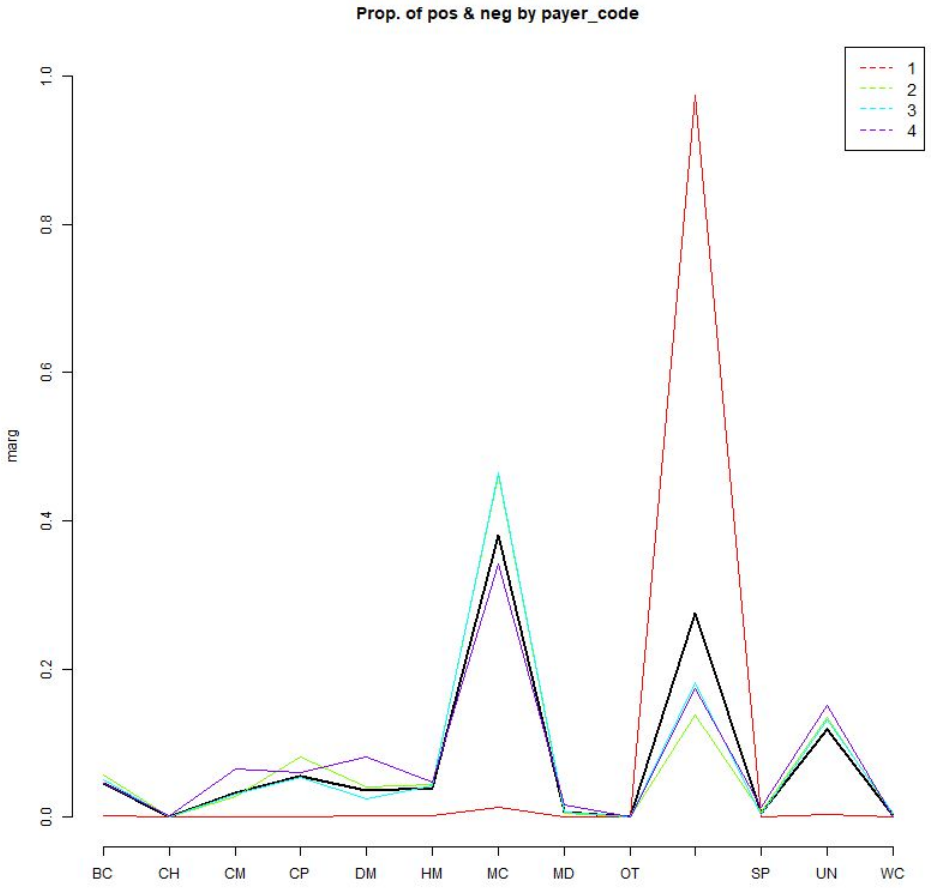


The first three clusters are really similar but we can see a clear deviation in the fourth, where the least amount of procedures were carried upon the patient.

payer_code



Here we can see a difference between the clusters. The second and third are more inclined for the MC value, the fourth is more balanced but the first is overwhelmingly unknown. This variable might not have that much relevance in the global scope of the study, but it is definitely interesting that those with no or unknown insurance are grouped in the first cluster. Obviously there is a large amount of patients with no or unknown insurance in the other clusters too, but the first is the only one with such disparity.

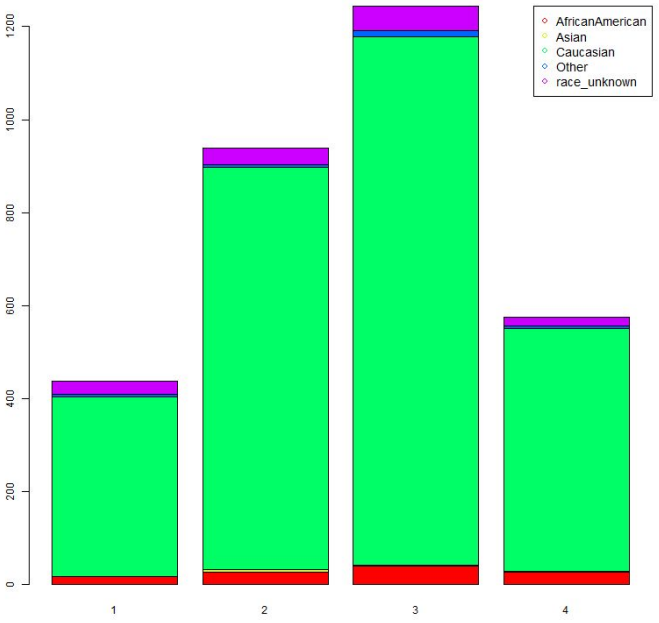


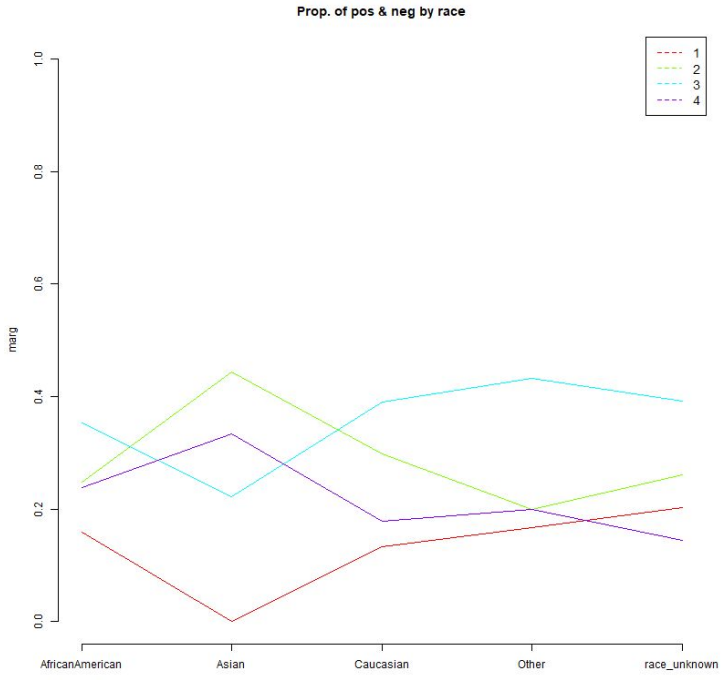
In this plot we can also see that the biggest peak is the first cluster over the unknown payer code, while the other clusters peak higher in the MC, so it reaffirms that those in the first cluster are those with no or unknown payer code.

If we combine this information with the fact that those in the first cluster are the ones with less emergency visits (and both inpatient and outpatient visits), it's easy to conclude that people with no payer code will not seek medical attention unless it is really necessary. To put this in context, we have to remember that the data id from the US, where it is not cheap to go to the doctor if you have no insurance.

Also, even if the first cluster has more people medicating with metformin than not, it's the second cluster with less people medicating (as shown as in *diabetesMed*), only behind the second cluster, which we have already concluded that are those patients that do not need medication. This, again, makes sense because they would have to pay the medication themselves.

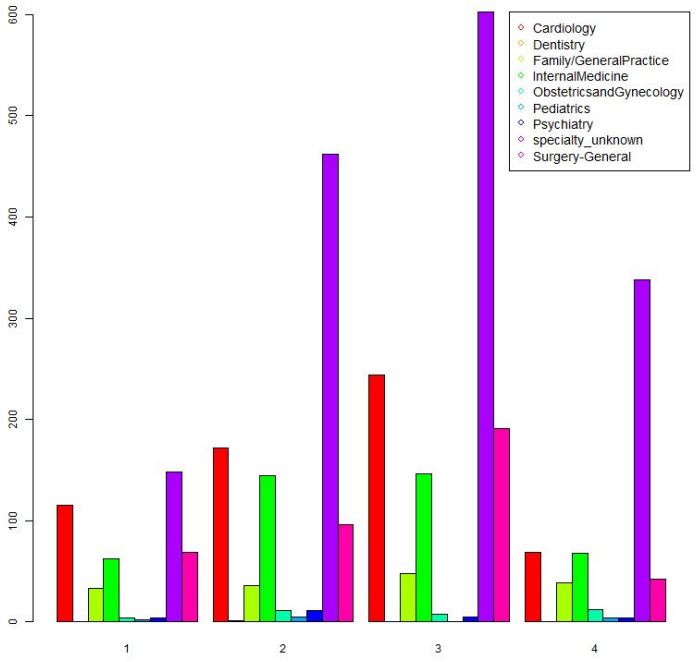
race

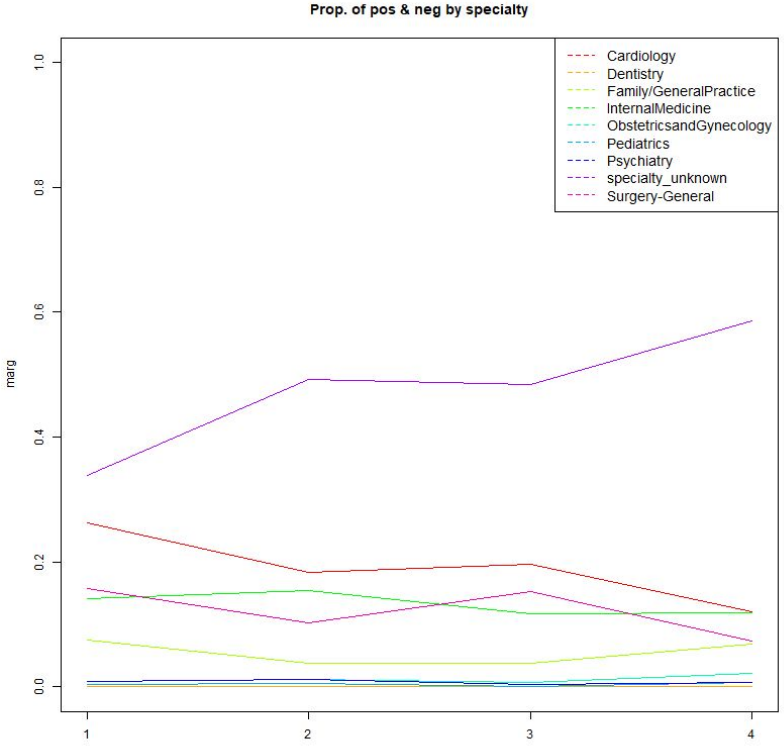




Not much to say about this variable other than the second cluster is the one with the most asian representation. In the end it's not that big of a difference to draw any conclusion since the number of asian people in the dataset is nine.

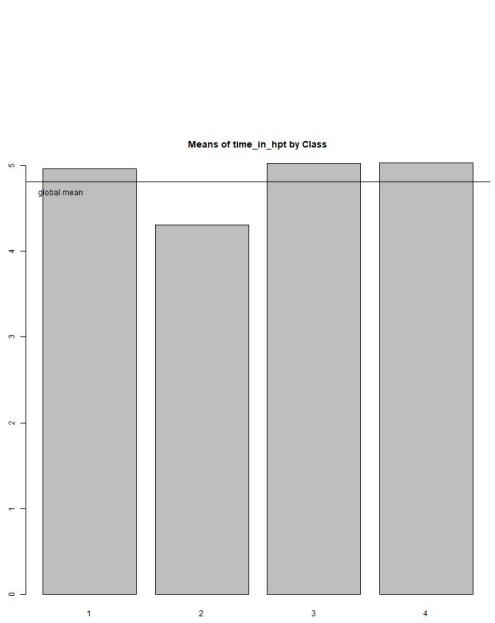
speciality

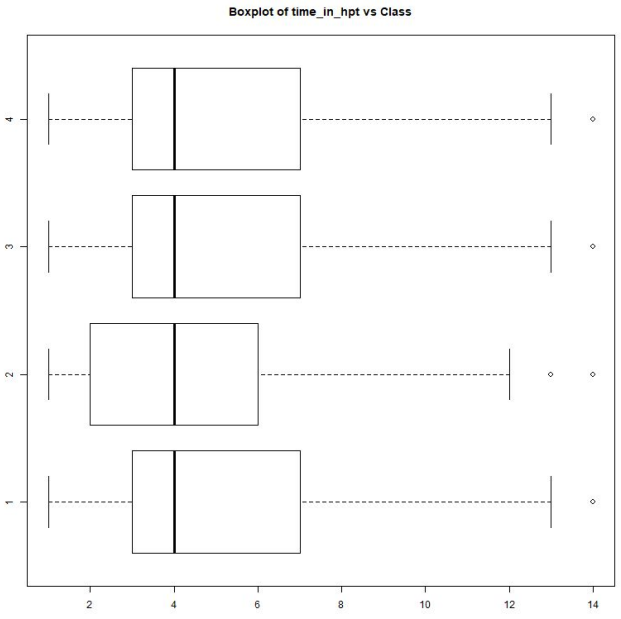




Again, not much to say about this variable. It is irrelevant to the global scope of the study and the plots don't say much other than the first cluster is the one with less percentage of unknowns.

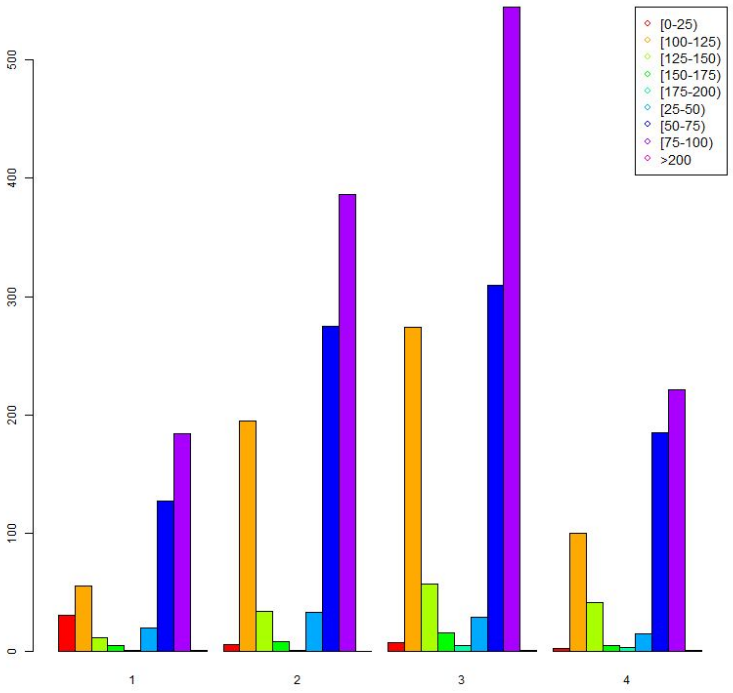
time_in_hpt





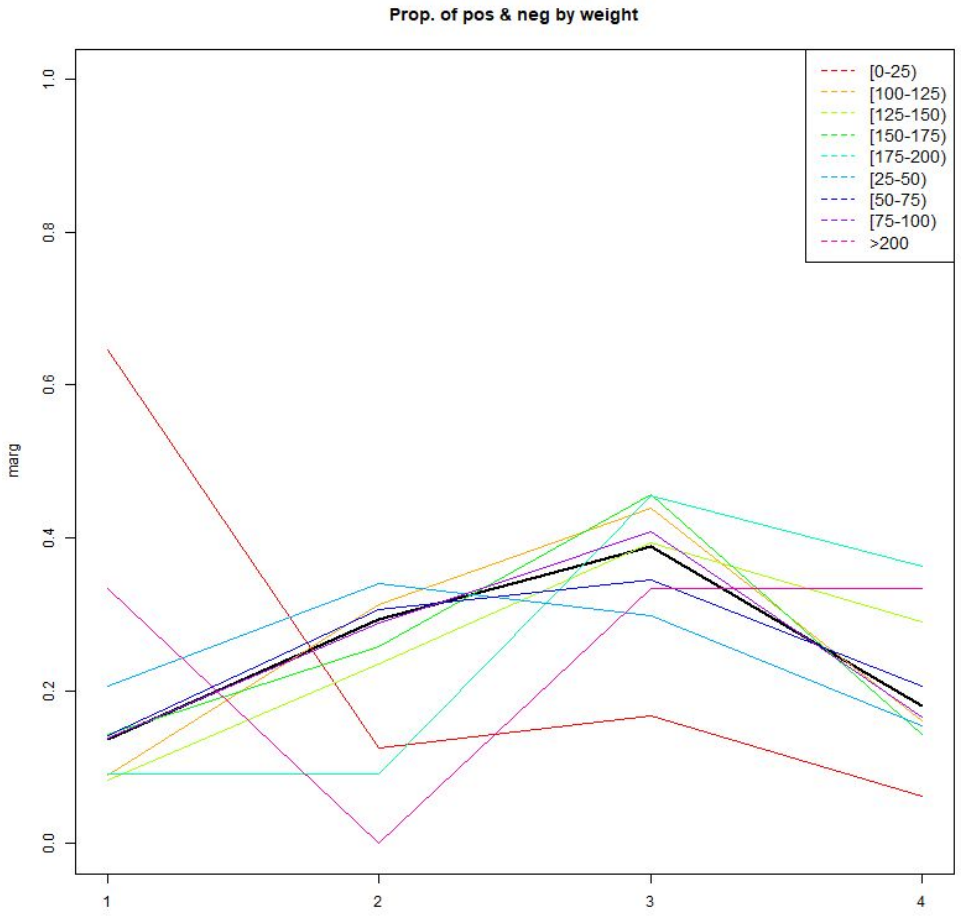
The clusters are all really similar in this variable. Only outlier is the second cluster with slightly less time spent in the hospital for the patients, which can be correlated to the fact that patients in the second cluster are those who do not need to take medicines, so it makes sense for them to not spend much time in the hospital.

Weight



Even though weight is of extreme importance in diabetic patients, as explained in the basic statistical descriptive analysis, our dataset is not particularly out of range regarding the body mass of the patients.

At first glance, it seems more or less balanced, with cluster one having the most patients in the 0-25 range and cluster three having the most overweight patients, which will make it interesting if the relative percentage of overweight people is also superior.

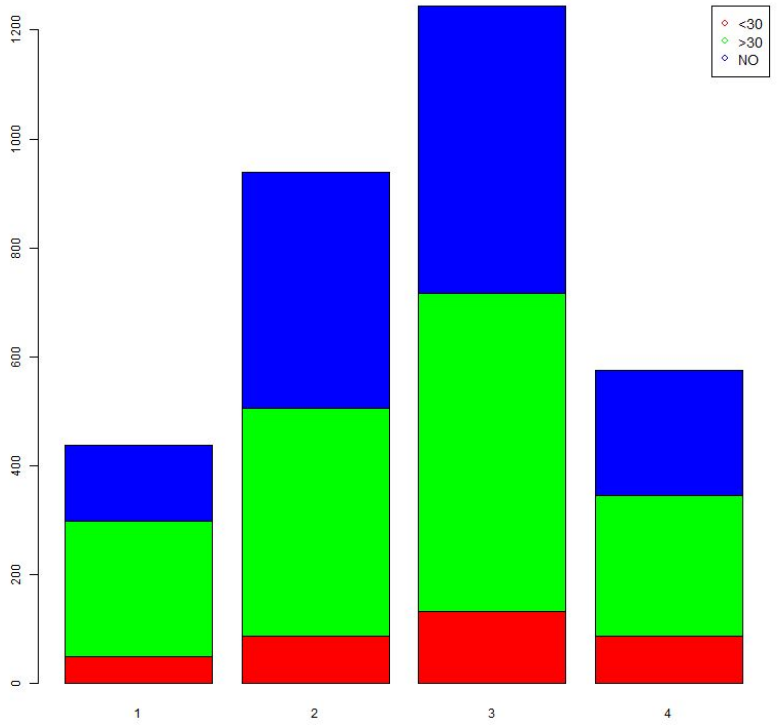


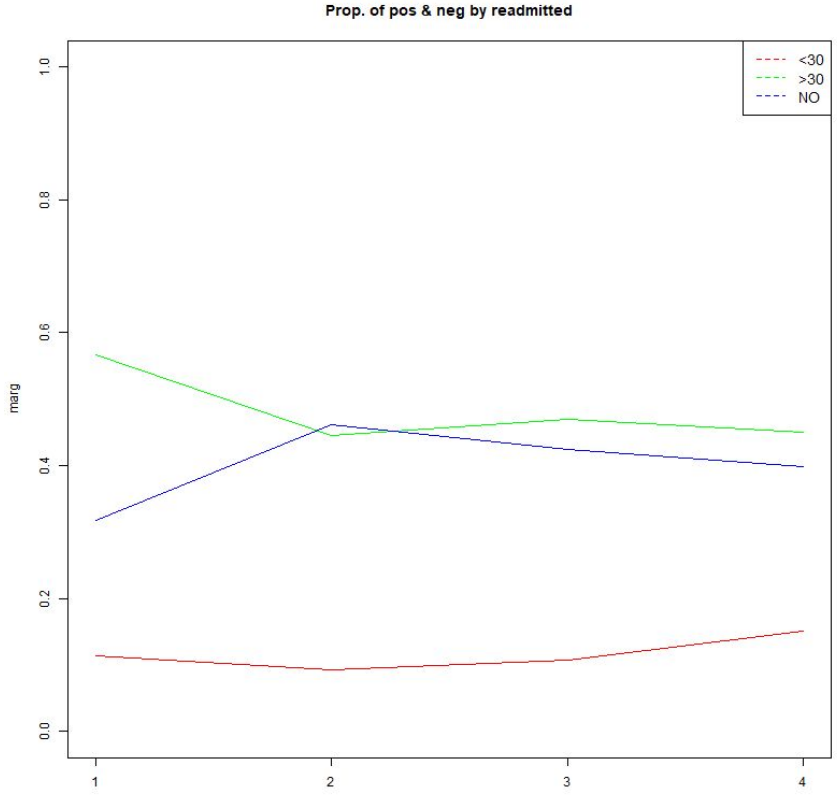
And it definitely is. With this we conclude that overweight and obese patients get grouped in the third cluster more than in the rest.

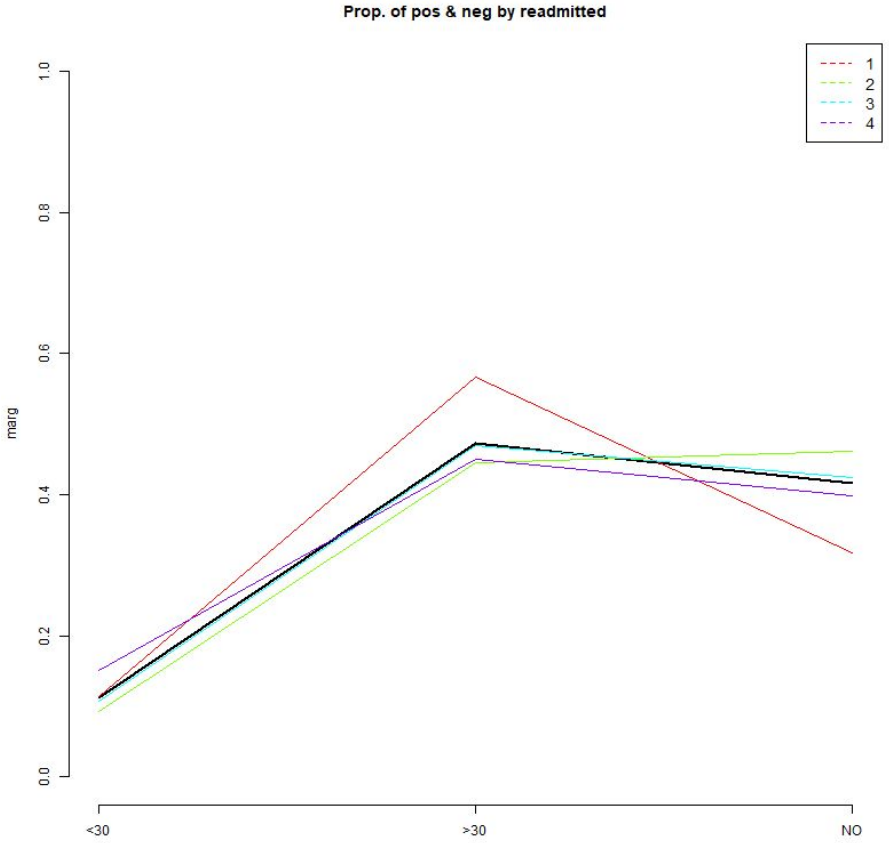
We already knew that patients in the third cluster consumed the most medications (as shown as in the variables *DiabetesMed* and *OtherMeds*), so this shows that overweight and obese patients will need to take more meds than the rest.

Readmitted

Lastly, we will analyze our response variable.


0
200
400
600
800
1000
1200
1
2
3
4
<30
>30
NO


Prop. of pos & neg by readmitted
marg
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
<30
>30
NO



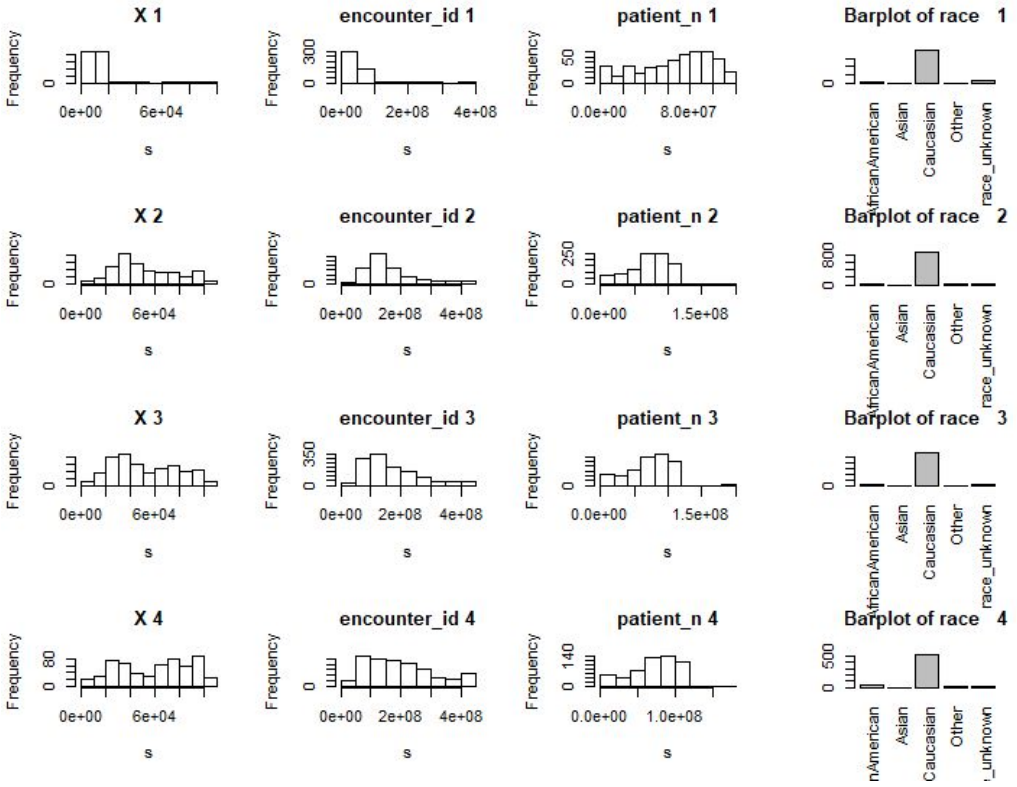
The three plots above are the most relevant to this variable. At first it seems that the clusters are well-balanced, but there is a big peak showing that the first cluster is the one with more percentage of late admission (after 30 days), while being the one with less no readmissions (both total number and percentage), so the patients in the first cluster are more likely to be readmitted, which is definitely interesting considering that we had established that those in the first cluster tend to avoid visiting the hospital. However, one way to rationalize this information is noting that we are talking about *late* readmissions, so it probably means that they wait until the last moment to be readmitted to the hospital, since they have no insurance, or at least they payment method is unknown, and some readmissions could have been avoided if they came earlier (and thus it's the cluster with more readmissions).

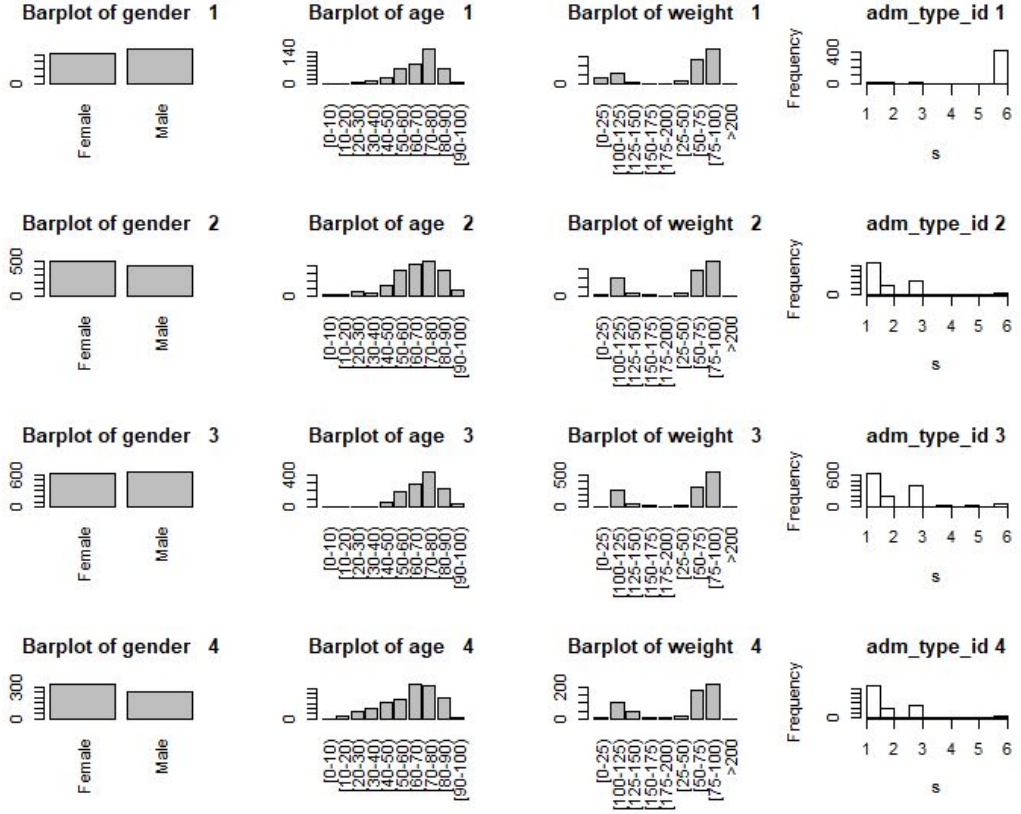
On the other hand the patients in the second cluster are the most likely to NOT be readmitted, continuing the impression that the patients of this cluster are those that do not need medication and thus are more likely to not need to be readmitted to the hospital.

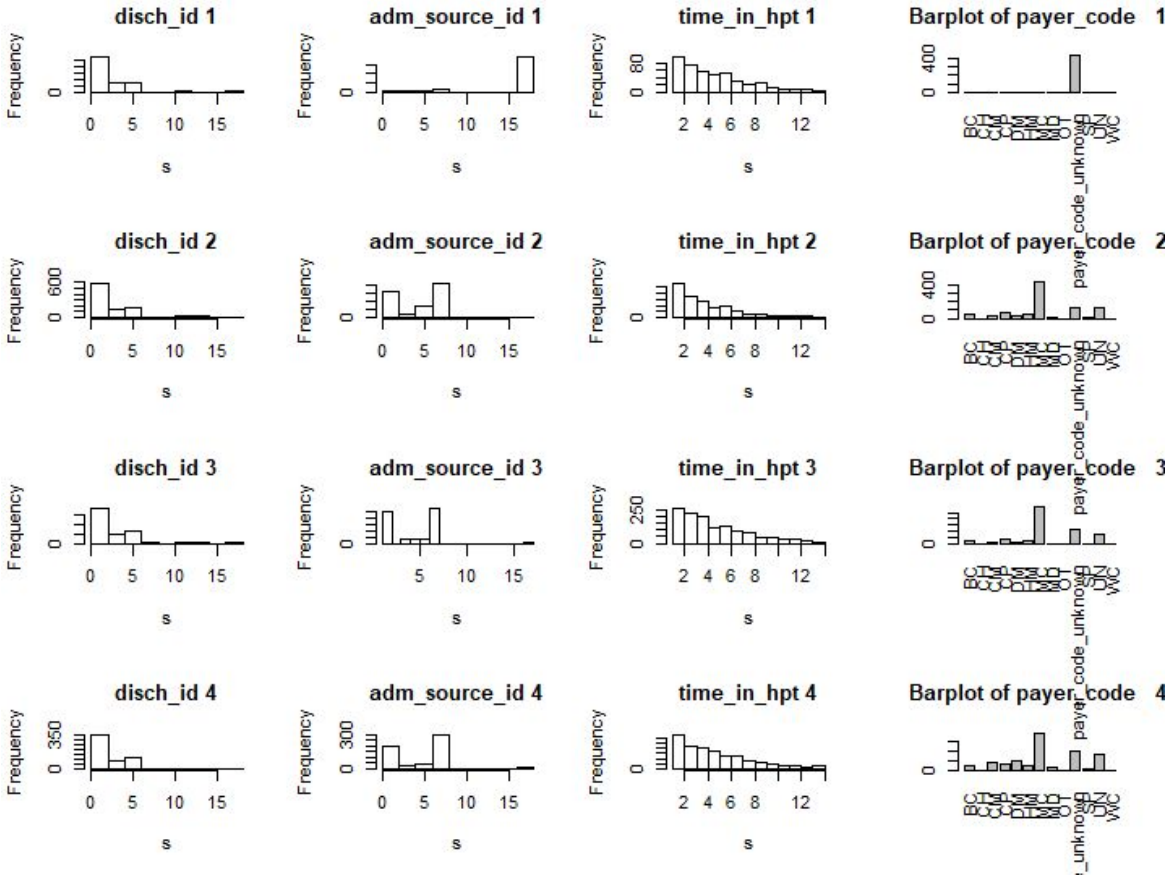
The p-values show that all the results are significant. (p-value < 0.05).

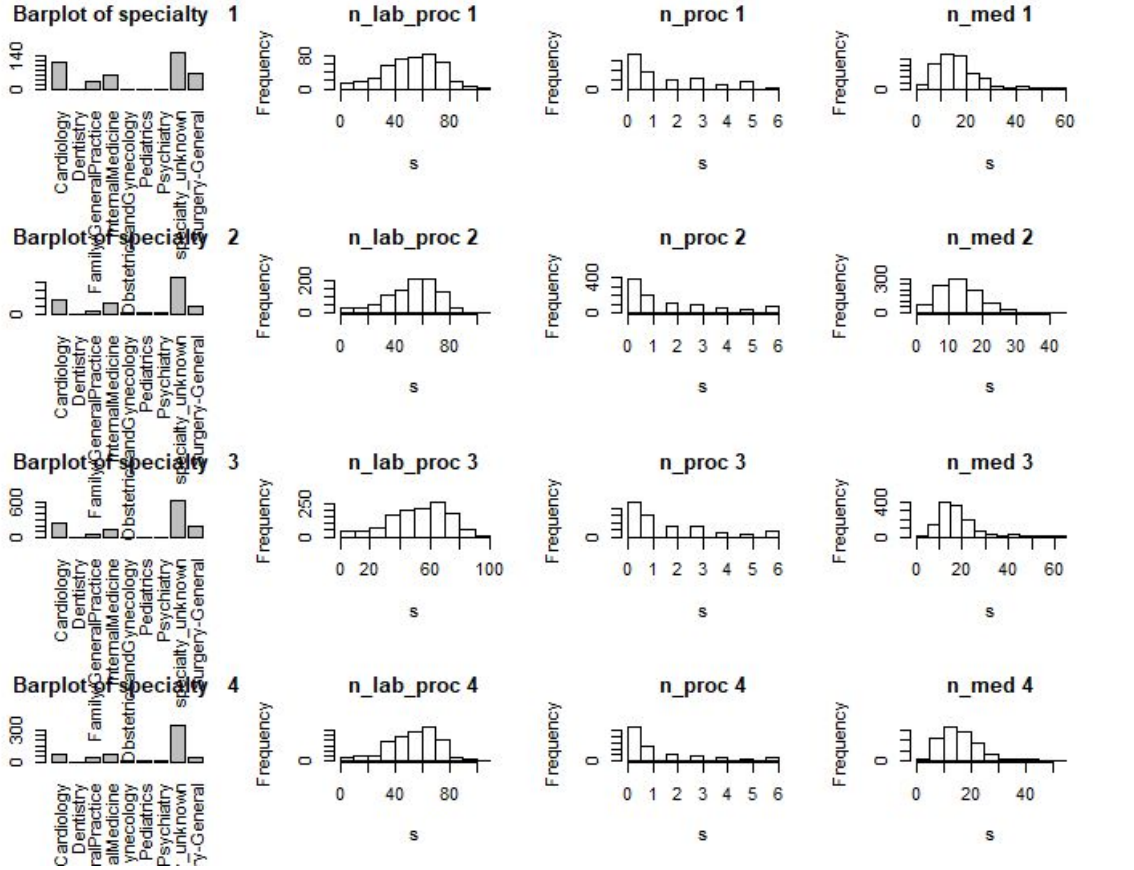
CPGs

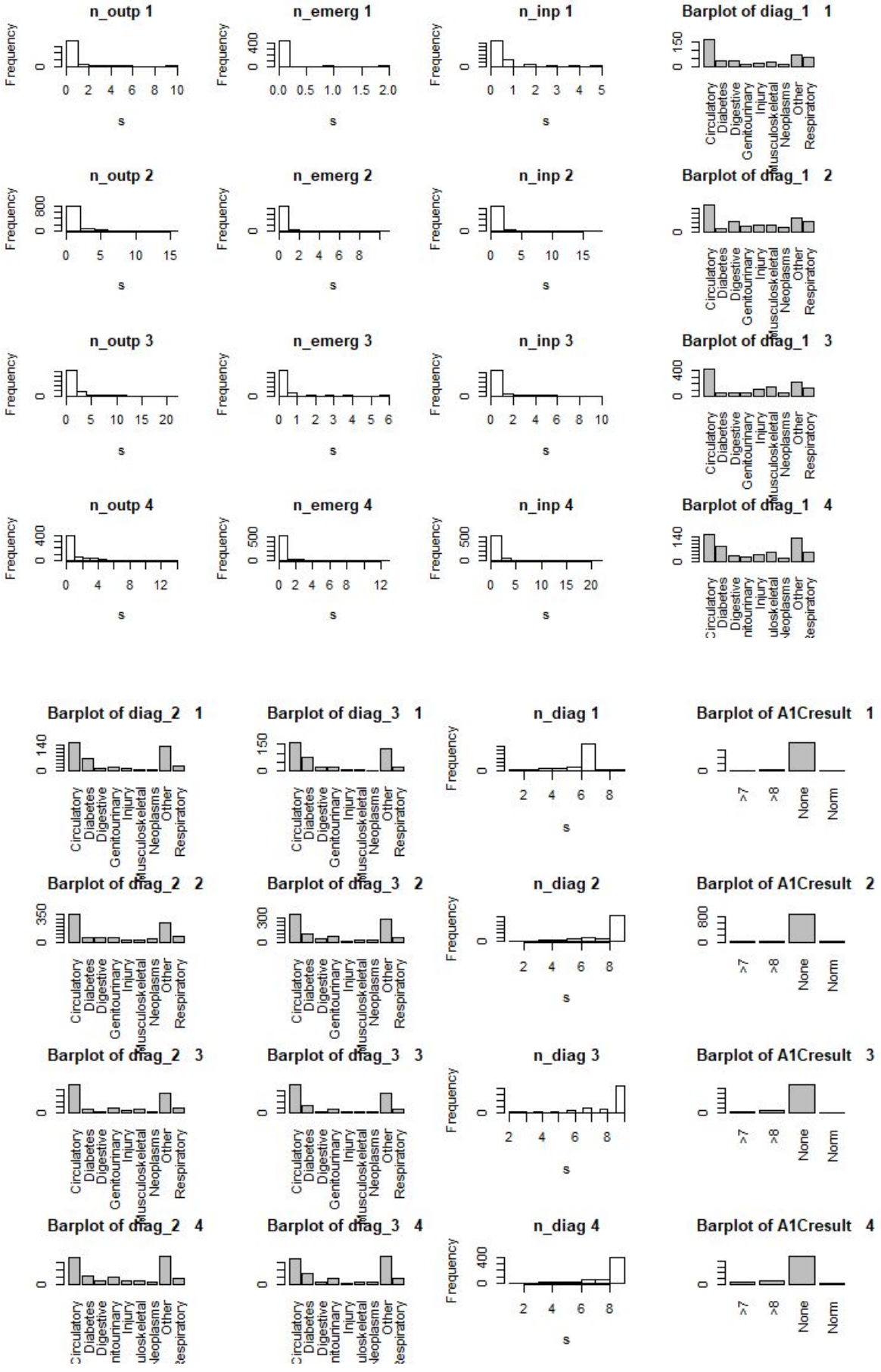
Here are all the CPGs of the variables:

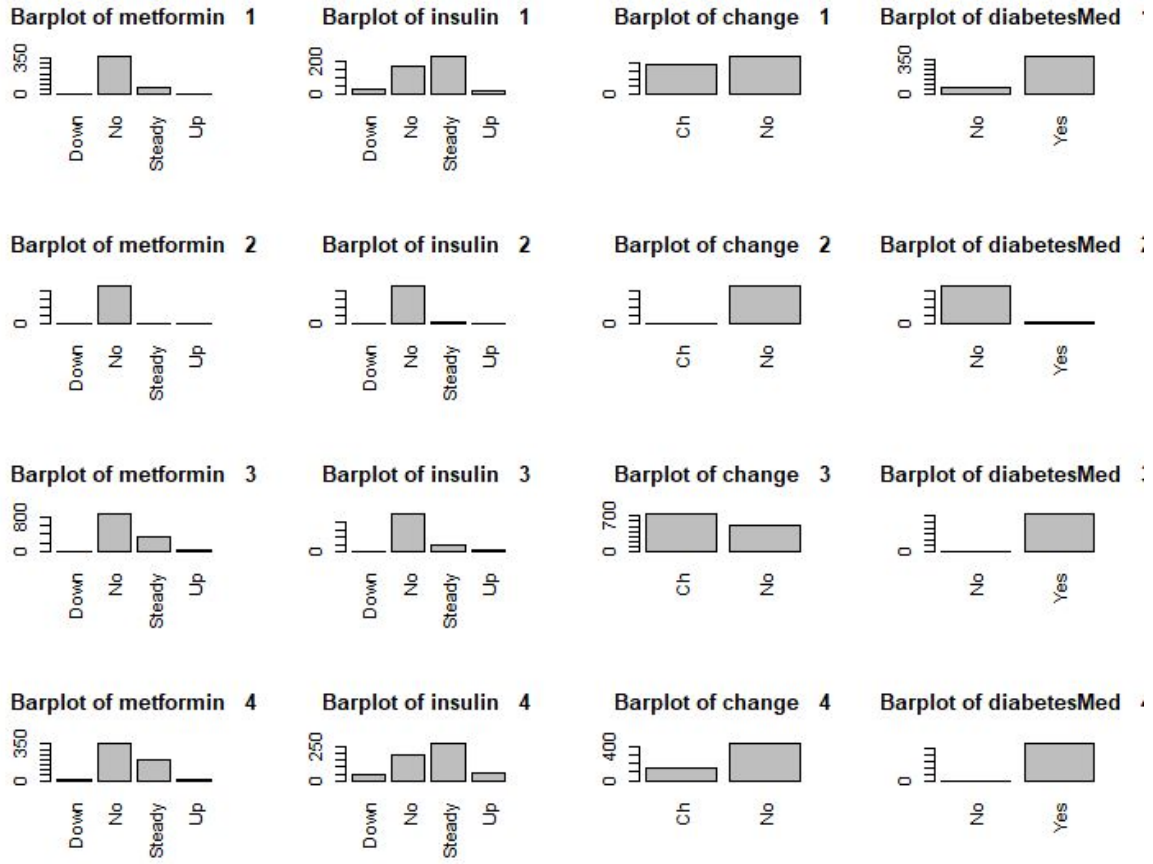


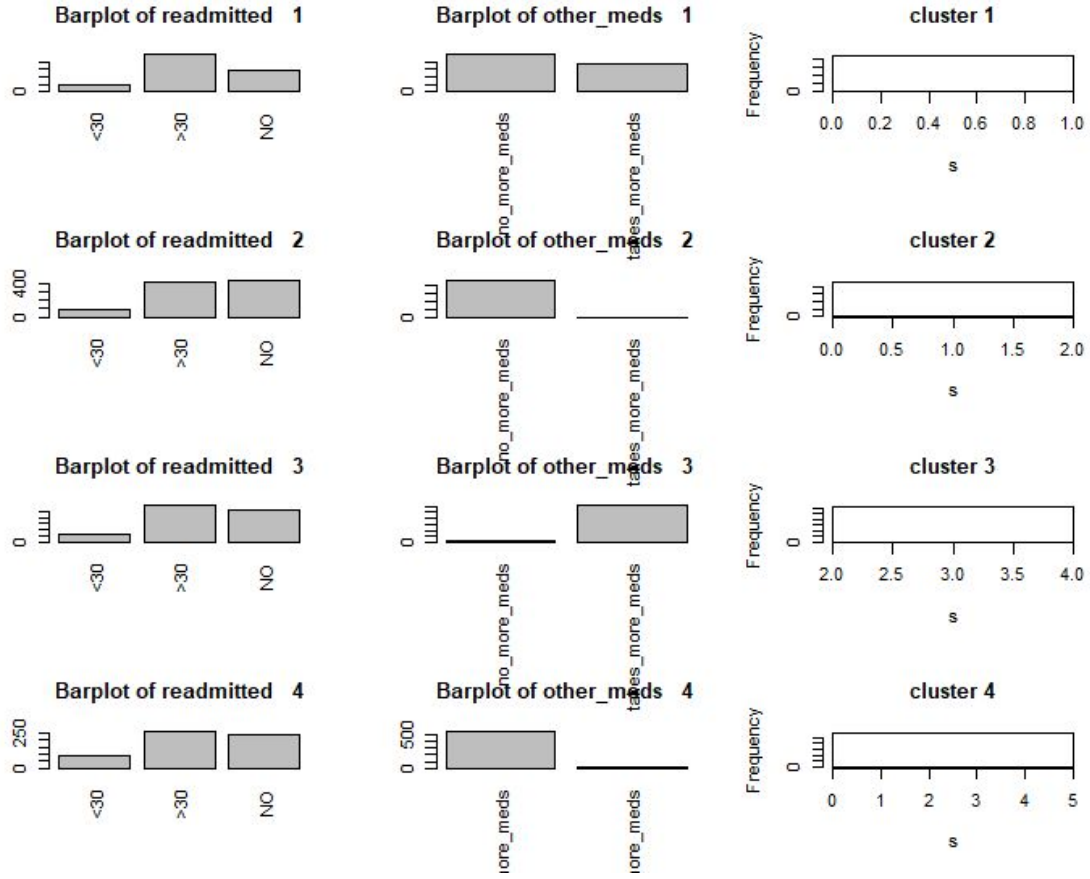



disch_id 1
adm_source_id 1
time_in_hpt 1
Barplot of payer_code 1
disch_id 2
adm_source_id 2
time_in_hpt 2
Barplot of payer_code 2
disch_id 3
adm_source_id 3
time_in_hpt 3
Barplot of payer_code 3
disch_id 4
adm_source_id 4
time_in_hpt 4
Barplot of payer_code 4
Frequency
s


Barplot of specialty 1
n_lab_proc 1
n_proc 1
n_med 1
Barplot of specialty 2
n_lab_proc 2
n_proc 2
n_med 2
Barplot of specialty 3
n_lab_proc 3
n_proc 3
n_med 3
Barplot of specialty 4
n_lab_proc 4
n_proc 4
n_med 4
Frequency
s


n_outp 1
n_emerg 1
n_inp 1
Barplot of diag_1 1
n_outp 2
n_emerg 2
n_inp 2
Barplot of diag_1 2
n_outp 3
n_emerg 3
n_inp 3
Barplot of diag_1 3
n_outp 4
n_emerg 4
n_inp 4
Barplot of diag_1 4
Barplot of diag_2 1
Barplot of diag_3 1
n_diag 1
Barplot of A1Cresult 1
Barplot of diag_2 2
Barplot of diag_3 2
n_diag 2
Barplot of A1Cresult 2
Barplot of diag_2 3
Barplot of diag_3 3
n_diag 3
Barplot of A1Cresult 3
Barplot of diag_2 4
Barplot of diag_3 4
n_diag 4
Barplot of A1Cresult 4
Frequency
s
Circulatory
Diabetes
Digestive
Genitourinary
Injury
Musculoskeletal
Neoplasms
Other
Respiratory
>7
>8
None
Norm


Barplot of metformin 1
Barplot of insulin 1
Barplot of change 1
Barplot of diabetesMed 1
Barplot of metformin 2
Barplot of insulin 2
Barplot of change 2
Barplot of diabetesMed 2
Barplot of metformin 3
Barplot of insulin 3
Barplot of change 3
Barplot of diabetesMed 3
Barplot of metformin 4
Barplot of insulin 4
Barplot of change 4
Barplot of diabetesMed 4
Down
No
Steady
Up
Ch
Yes


Barplot of readmitted 1
Barplot of other_meds 1
cluster 1
Barplot of readmitted 2
Barplot of other_meds 2
cluster 2
Barplot of readmitted 3
Barplot of other_meds 3
cluster 3
Barplot of readmitted 4
Barplot of other_meds 4
cluster 4
<30
>30
NO
no_more_meds
takes_more_meds
Frequency
s

Classes Interpretation

First cluster

One of the most relevant characteristics of the first cluster is their distribution of the payer code; almost all of its members have a payer code unknown, which may indicate that either they do not have insurance or that they do not specify it, which at the same time may indicate a lower social status of the individuals of this class. It has the highest proportion of kids, which can be seen both by the age and low weight. The change variable, which was one of the most relevant because in its difference among clusters, is balanced roughly at fifty percent. They do take insulin and other medicines, but not predominantly. They have a lower number of diagnoses and visits compared to the other clusters, which again may indicate that the population of this cluster may avoid going to the doctor because they can't pay it. They are the population with the highest representations of late readmissions, that again may support our theory that the individuals of this class postpone going to the doctor as much as possible.

Second cluster

The most defining characteristic of this cluster is that they do not medicate, at all. They do not take insulin, metformin nor any other of the drugs that were recorded for. Logically, their change variable, which indicated their change in medication dosage, is virtually zero. Overall, they also spend less time in the hospital and their readmission rates are the lower. Their age is not representative or the cluster, as they have individuals from all ages. Having all this in mind, we can conclude that the members of this cluster represent the ones in less severe conditions.

Third cluster

This is by far the biggest cluster, it has the greatest number of members compared to the other ones. It has a high prevalence of old people, compared to other age ranges. Therefore, it makes sense for it to be the bigger cluster, as the complete population of our dataset was predominantly of old age. All of its members are taking some form of diabetic medication, as indicated by the Diabetes Med variable, but not particularly insulin. They also have a very high use of other medicines, their values for the other_meds variable are very predominantly positive. All of their members have a valid payer code, however, there is no code that dominates over the others, apart from MC, which dominates in all clusters except the first.

They have the highest number of overweight individuals. This class may represent more severe patients compared to the previous two clusters. Their readmission rates are balanced compared to other clusters, so it is hard to say anything of value regarding this variable. Because of their ages, weight and metformin values, most of the patients in this cluster may present a case of type two diabetes.

Fourth cluster

This cluster has the highest representation of young people (other than kids). All of them take diabetic medications, but mostly insulin. They do not take any other kind of medication. They all have a valid payer code with similar proportions as the two previous clusters (all do except the first). They present the least amount of procedures. However, they are the group most likely to be readmitted early. From the age of the group and their medication it may be safe to assume that most of the population of this class presents type one diabetes.

Global Discussion

In cluster 1 we can see young and less readmission short-term, along with not taking many medicines nor changing them predominantly. We can somewhat see this in the ACP plots, specifically mapping the low weight and age to no readmission or readmission in more than 30 days. However, we can't see things such as the payer code being related in the ACP.

In cluster 2 we saw characteristics related to not taking medication, less time in the hospital and low readmission rate. We notice this group matches several things in the ACP, particularly less medication is correlated to not being readmitted and to spending less time in the hospital too.

In cluster 3, we match old, overweight people with changes in medicine, taking diabetic medication and metformin, but with lower levels of insulin administration. We are seeing a group that is on the upper side of our 3rd axis, the so called 'risky' axis, and in the ACP plots we can find several of these relationships quite close by. But, as with ACP's second plot, we cannot directly link age and weight to readmission. This cluster is probably conformed by patients with type 2 diabetes.

In cluster 4, however, we find the most important relationship that has been overarching over this study. Cluster 4 contains youngsters (but no kids) taking insulin but no other meds, with the least amount of procedures and with known payer code. Also, it is the cluster with the greatest record in visits (*n_inp, n_out, n_emer*). It is also the cluster with the greatest amount of early readmission, and from this we can link the major point in our axis 1 of the ACP, which is that more visits earlier during the year is very correlated to an earlier readmission. Here we also see that link between less medications taken is related to less procedures taken which seems obvious.

However, recalling ACP we remember several relationships we saw that are not present here. In particular, we are talking about:

- Weight being linked to readmission
- Rise in insulin administration linked to many diagnoses
- Slightly old patients have a higher number of visits
- Diabetic diagnosed patients generally take more meds
- Asian patients do not get readmitted
- African american patients have a higher chance to get a raise in insulin dosage
- A 'respiratory admission' is linked to having many diagnoses.

We don't see the first relationship as high weight and high number of visits are in separate clusters, while the ACP is projecting the two things together, which can be seen in the second factorial plane.

Of the others, we can understand how some of them are not visible in the clustering. Particularly, those related to lower granularity factors, such as Asian or African American race, along with respiratory admissions and a diagnosis of diabetes, are very difficult to see when splitting a whole dataset into 4 clusters. They might be together in one of the clusters, but profiling them out is unfeasible. Furthermore, more in-depth analysis brings out the fact that the Asian collective in our dataset is extremely small, with less than 10 instances.

The rise in insulin and slightly old patients are not as straightforward to see, but we have some base to reject the second one when we revise the first factorial plane and see no real relationship there. We can also hypothesize that given that the rise in insulin administration is a factor that does not fall into a single cluster predominantly, and it is not quite common in the dataset. Therefore, it does not get represented in the clustering enough to see this relationship.

Conclusions

After having realized this study we have come across several pieces of not so evident new knowledge.

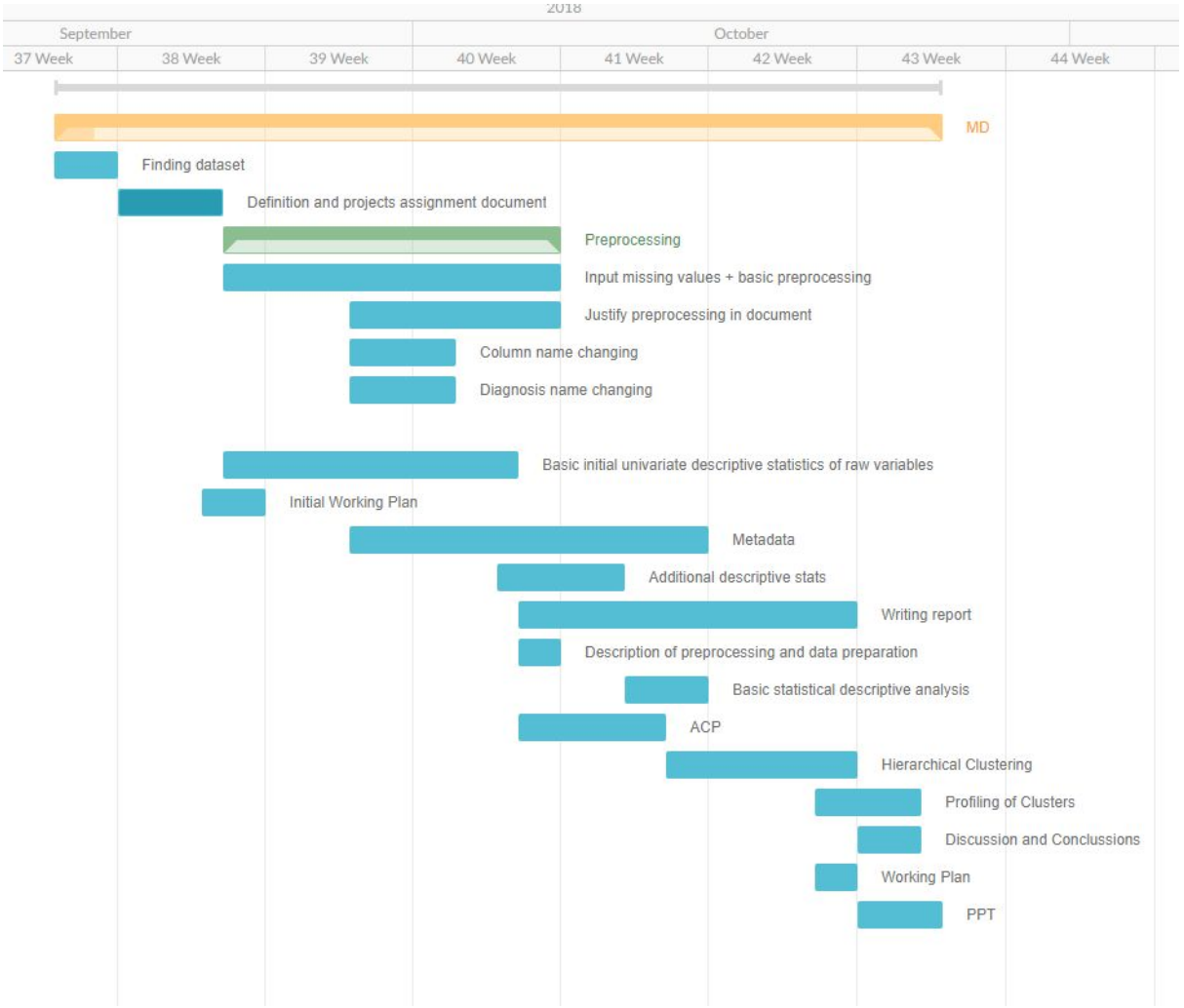
Relative to the problem, we have analyzed a heavily studied dataset but from a different point of view: not discarding the weight variable. From it, we tried to study the behaviour of the readmission over diabetic inpatients, but against what one would expect we found no reason to link heavier weights and older age to readmission. Instead, we are drawn to the conclusion that young people are the less fortunate in this regard, to be specific we have inferred that young people with type 1 diabetes are the most likely to be readmitted in short periods of time. Furthermore so, we have also found correlation between kids and being readmitted but after a period 30 days. However, the most important point we have finally uncovered is that patients with a medical history with more visits to the hospital previously during the year are the most likely to come back. Therefore, a patient should be considered under the risk of readmission when he has visited the hospital many times previously, even more so if this person is a type 1 diabetic.

Relative to the data mining techniques, we are surprised to see such good behaviour between the profiled classes and our ACP analysis, since we had not expected them to match so thoroughly at first. On the other hand, we have also come to the conclusion that, for hierarchical clustering, the use of an index such as Calinski-Harabasz should be highly encouraged. At first we were rather confident in picking 3 as our number of clusters, since that was also the number of factors of our target variable and the dendogram was ‘apparently’ good for 3. However, after seeing the expressivity we have achieved picking 4 instead we are rather satisfied with this more informed choice.

Working Plan

Initial Gantt Diagram

Final Gantt Diagram



The initial Gantt was planned taking into account that we had to finish the work by October 19th. However, the deadline was changed to October 26th, so we invested more time than originally planned in some of the tasks such as the *Hierarchical Clustering* or *Metadata*.

We also got confused over when was a suitable time to do the *Basic initial univariate descriptive statistics of raw variables*. We thought we could do it after the preprocessing, but after revising what we had to do, we advanced that phase.

Apart from these two things, we believe we have followed the initial working plan with no major deviations.

Division of tasks

	Marc Badia	Aleix Balletbó	Bernat Gené	Víctor Giménez	Guillem Ferrer	Daniel Tarrés
Finding dataset	X	X	X	X	X	X
Definition and projects assignment document	X	X	X	X	X	X
Initial Working Plan	X		X			
Metadata		X				X
Univariate descriptive statistics of raw variables		X				X
Input missing values + basic preprocessing		X		X		
Column name changing					X	
Diagnosis name changing	X		X			X
Additional descriptive stats				X	X	
Justify preprocessing in document	X	X	X	X	X	X
Writing report (motivation, data source, formatting, etc)	X		X	X	X	
Description of preprocessing and data preparation				X	X	
Basic statistical descriptive analysis	X		X			
ACP				X	X	
Hierarchical Clustering		X				X
Profiling of Clusters	X		X			
Discussion and Conclusions	X	X	X	X	X	X
Working Plan		X				X
PPT	X	X	X	X	X	X

The division of tasks was the same as originally planned, so everyone did the tasks assigned to them.

Risk contingency plan

Risk 1: A team member leaves the course.
How to prevent: All the tasks have more than one member assigned.
How to manage: Reassign uncompleted tasks and balance again.

Risk 2:. A task ends up being much more or less work than was previously thought.
How to prevent: Discuss with the group how much time will each task take. Use agile methods such as “Planning Poker” to do it.
How to manage: Reassign other tasks so the work is well-balanced within the team. If necessary re-estimate other tasks with the acquired information.

Risk 3: There are errors in the dataset.
How to prevent: Examine the dataset to clear errors.
How to manage: Delete the errors and start again from the beginning.

Risk 4: A team member does a poor job in a task he was assigned.
How to prevent: All tasks have more than one member assigned and are revised by even more members.
How to manage: The rest of the team talks with the team member involved to revisit his work. In case the job is still unsatisfactory, the task will be reassigned and the team member removed from the team. Depending on the severity of the case, we may involve the teacher.

No additional risk has appeared during the course of this project.

R Scripts

DiabeticPreprocessing.Rmd

```
```{r}
#Set working directory to the one with the data in the following way:
setwd("YOUR/PATH")  //NOTICE ORIENTATION OF BAR
diabetic_data <- read.csv("diabetic_data.csv", na.strings = c("?"))
names(which(sapply(diabetic_data, anyNA)))
```


First we want to see the % of missings that each variable has


```{r echo=FALSE}
print(paste0("race: ",(sum(is.na(diabetic_data$race))/dim(diabetic_data)[1])*100))
print(paste0("weight:
",(sum(is.na(diabetic_data$weight))/dim(diabetic_data)[1])*100))
print(paste0("payer_code:
",(sum(is.na(diabetic_data$payer_code))/dim(diabetic_data)[1])*100))
print(paste0("medical_specialty:
",(sum(is.na(diabetic_data$medical_specialty))/dim(diabetic_data)[1])*100))
print(paste0("diag_1:
",(sum(is.na(diabetic_data$diag_1))/dim(diabetic_data)[1])*100))
print(paste0("diag_2:
",(sum(is.na(diabetic_data$diag_2))/dim(diabetic_data)[1])*100))
print(paste0("diag_3:
",(sum(is.na(diabetic_data$diag_3))/dim(diabetic_data)[1])*100))
```


The first decision is to work only with individuals whose weight has been recorded. We notice that these NAs might not be random but structural, but nevertheless we choose this scope of work for our study, as a new criteria over our data: "Each individual should have their weight recorded."
```{r}
diabetic_data <- diabetic_data[!is.na(diabetic_data$weight), ]
n<-dim(diabetic_data)[1]
```


Our dataset has been significantly reduced, and now we have `r n` observations. We re-evaluate the missing values once again.
```{r echo=FALSE}
names(which(sapply(diabetic_data, anyNA)))
print(paste0("race: ",(sum(is.na(diabetic_data$race))/dim(diabetic_data)[1])*100))
print(paste0("payer_code:
",(sum(is.na(diabetic_data$payer_code))/dim(diabetic_data)[1])*100))
print(paste0("medical_specialty:
",(sum(is.na(diabetic_data$medical_specialty))/dim(diabetic_data)[1])*100))
print(paste0("diag_2:
",(sum(is.na(diabetic_data$diag_2))/dim(diabetic_data)[1])*100))
```
```

```
print(paste0("diag_3:
",(sum(is.na(diabetic_data$diag_3))/dim(diabetic_data)[1])*100))
```

Our next step is to input missing values and, since those variables with NAs are all categorical, we will create a new factor named <var>_unknown for each one of them, substituting the missing values by it. This will help for computation and visualization procedures.

```{r}
diabetic_data$race <- factor(diabetic_data$race, levels=
c(levels(diabetic_data$race),"race_unknown"))
diabetic_data$race[is.na(diabetic_data$race)] <- "race_unknown"


diabetic_data$payer_code <- factor(diabetic_data$payer_code, levels=
c(levels(diabetic_data$payer_code),"payer_code_unknown"))
diabetic_data$payer_code[is.na(diabetic_data$payer_code)] <- "payer_code_unknown"


diabetic_data$medical_specialty <- factor(diabetic_data$medical_specialty, levels=
c(levels(diabetic_data$medical_specialty),"specialty_unknown"))
diabetic_data$medical_specialty[is.na(diabetic_data$medical_specialty)] <-
"specialty_unknown"

diabetic_data$diag_2 <- factor(diabetic_data$diag_2, levels=
c(levels(diabetic_data$diag_2),"diag_2_unknown"))
diabetic_data$diag_2[is.na(diabetic_data$diag_2)] <- "diag_2_unknown"


diabetic_data$diag_3 <- factor(diabetic_data$diag_3, levels=
c(levels(diabetic_data$diag_3),"diag_3_unknown"))
diabetic_data$diag_3[is.na(diabetic_data$diag_3)] <- "diag_3_unknown"
```


Now we have no NAs (or at least, they have been reskinned). We want to see the levels of our factors:

```{r}
summary(diabetic_data)
```


We notice that some variables have not been imported correctly as factors(admission_source_id, admission_type_id, discharge_disposition_id). We fix it.
```{r}
diabetic_data$admission_source_id = as.factor(diabetic_data$admission_source_id)

diabetic_data$admission_type_id = as.factor(diabetic_data$admission_type_id)

diabetic_data$discharge_disposition_id =
as.factor(diabetic_data$discharge_disposition_id)
```


We notice several things from this last summary. Firstly, we have too many medicines, most of them apparently uninformative for the subset of data we are dealing with. We also have too many factors over diagnostics, and after a look at the meaning of those factors we notice they are not even ordinal and would not be too useful in their current form (too disperse and the factors' semantical relationship is not expressed).

From these observations, we decide that the next steps for modelling our data are the following:
 - Collapse the drugs variables, droping most of the unsignificant ones. Our criteria has been to maintain insuline and metformine as they are (since they are both important for diabetes and well distributed over the factors) and to group all of the rest into a new variable 'other_meds', which will be binary, with value

'no_more_meds' if all of the other drugs have 'No' value and 'takes_more_meds' otherwise.
 - Reduce factors over diagnostics, grouping them semantically. That is, instead of using specific diagnostics, we will map them to the general area they are related to, such as circulatory, respiratory etc. For that, we already have a table made by experts of the field.
 - Rename the variables and factors if they are too long, for the sake of having a cleaner look.

We will first work on collapsing the drugs. To recapitulate, we have decided to name this new variable 'other_meds', which will be a binary variable with factors:'takes_more_meds' and 'no_more_meds' (which will help distinguish it from other qualitative vars when we do the PCA).

```{r}
diabetic_data$other_meds <- as.factor(with(diabetic_data, ifelse(repaglinide=='No' & nateglinide == 'No'& chlorpropamide == 'No'& glimepiride == 'No'& acetohexamide == 'No'& glipizide == 'No'& glyburide == 'No'& tolbutamide == 'No'& pioglitazone == 'No'& rosiglitazone == 'No'& acarbose == 'No'& miglitol == 'No'& troglitazone == 'No'& tolazamide == 'No'& examide == 'No' & citoglipton == 'No', 'no_more_meds', 'takes_more_meds')))

table(diabetic_data$other_meds)

```

As we see above, this new variable is quite well balanced (which implies it will probably be informative as well). By reducing the number of variables (which were imbalanced between their factors), we hope to reduce the execution time of our algorithms while keeping a (now not-so-informative as before) record over the rest of medicines which appear to be more punctual than metformin and insuline.
Having checked that, we now drop the compiled medicine variables from the dataset. We'll also drop the variable max_glu_serum, since we have observed it has no descriptive capabilities (all values are 'None').

```{r echo=FALSE}
cat("Previous data:")

colnames(diabetic_data)

diabetic_data_1 <- diabetic_data[, -which(names(diabetic_data) %in% c("repaglinide","nateglinide","chlorpropamide","glimepiride","acetohexamide","glipizide","glyburide","tolbutamide","pioglitazone","rosiglitazone","acarbose","miglitol","troglitazone","tolazamide","examide","citoglipton","glyburide.metformin","glipizide.metformin","glimepiride.pioglitazone","metformin.rosiglitazone","metformin.pioglitazone", "max_glu_serum"))]

cat("\nNew data:\n")

colnames(diabetic_data_1)

diabetic_data <- diabetic_data_1
```

We will now reduce the factors, using the table provided by experts. We first look over the factor distribution of diag_1
```{r}
plot(diabetic_data$diag_1)
```

As we can see, we have many factors and not much we can interpret from them.

First we will need to transform these values from factors to characters (strings), to work more easily
```{r echo=FALSE}
diabetic_data$diag_1 <- as.character(diabetic_data$diag_1)

diabetic_data$diag_2 <- as.character(diabetic_data$diag_2)
```

```
diabetic_data$diag_3 <- as.character(diabetic_data$diag_3)
```

Next, we change these values to their group names following the guidelines provided in the Table 2 of the description PDF.

```{r echo=FALSE}
#Circulatory
for (i in 390:459){
 diabetic_data$diag_1[diabetic_data$diag_1 == as.character(i)] <- "Circulatory"
 diabetic_data$diag_2[diabetic_data$diag_2 == as.character(i)] <- "Circulatory"
 diabetic_data$diag_3[diabetic_data$diag_3 == as.character(i)] <- "Circulatory"
}
diabetic_data$diag_1[diabetic_data$diag_1 == "785"] <- "Circulatory"
diabetic_data$diag_2[diabetic_data$diag_2 == "785"] <- "Circulatory"
diabetic_data$diag_3[diabetic_data$diag_3 == "785"] <- "Circulatory"

#Respiratory
for (i in 460:519){
 diabetic_data$diag_1[diabetic_data$diag_1 == as.character(i)] <- "Respiratory"
 diabetic_data$diag_2[diabetic_data$diag_2 == as.character(i)] <- "Respiratory"
 diabetic_data$diag_3[diabetic_data$diag_3 == as.character(i)] <- "Respiratory"
}
diabetic_data$diag_1[diabetic_data$diag_1 == "786"] <- "Respiratory"
diabetic_data$diag_2[diabetic_data$diag_2 == "786"] <- "Respiratory"
diabetic_data$diag_3[diabetic_data$diag_3 == "786"] <- "Respiratory"

#Digestive
name <- "Digestive"
for (i in 520:579){
 diabetic_data$diag_1[diabetic_data$diag_1 == as.character(i)] <- name
 diabetic_data$diag_2[diabetic_data$diag_2 == as.character(i)] <- name
 diabetic_data$diag_3[diabetic_data$diag_3 == as.character(i)] <- name
}
diabetic_data$diag_1[diabetic_data$diag_1 == "787"] <- name
diabetic_data$diag_2[diabetic_data$diag_2 == "787"] <- name
diabetic_data$diag_3[diabetic_data$diag_3 == "787"] <- name


#Injury
name <- "Injury"
for (i in 800:999){
 diabetic_data$diag_1[diabetic_data$diag_1 == as.character(i)] <- name
 diabetic_data$diag_2[diabetic_data$diag_2 == as.character(i)] <- name
 diabetic_data$diag_3[diabetic_data$diag_3 == as.character(i)] <- name
}


#Musculoskeletal
name <- "Musculoskeletal"
```

```
for (i in 710:739){
 diabetic_data$diag_1[diabetic_data$diag_1 == as.character(i)] <- name
 diabetic_data$diag_2[diabetic_data$diag_2 == as.character(i)] <- name
 diabetic_data$diag_3[diabetic_data$diag_3 == as.character(i)] <- name
}


#Genitourinary
name <- "Genitourinary"
for (i in 580:629){
 diabetic_data$diag_1[diabetic_data$diag_1 == as.character(i)] <- name
 diabetic_data$diag_2[diabetic_data$diag_2 == as.character(i)] <- name
 diabetic_data$diag_3[diabetic_data$diag_3 == as.character(i)] <- name
}
diabetic_data$diag_1[diabetic_data$diag_1 == "788"] <- name
diabetic_data$diag_2[diabetic_data$diag_2 == "788"] <- name
diabetic_data$diag_3[diabetic_data$diag_3 == "788"] <- name


#Neoplasms
name <- "Neoplasms"
for (i in 140:239){
 diabetic_data$diag_1[diabetic_data$diag_1 == as.character(i)] <- name
 diabetic_data$diag_2[diabetic_data$diag_2 == as.character(i)] <- name
 diabetic_data$diag_3[diabetic_data$diag_3 == as.character(i)] <- name
}


#Diabetes
name <- "Diabetes"
diabetic_data$diag_1[diabetic_data$diag_1 >="250." & diabetic_data$diag_1 <=
"250.999999999"] <- name
diabetic_data$diag_2[diabetic_data$diag_2 >="250." & diabetic_data$diag_2 <=
"250.999999999"] <- name
diabetic_data$diag_3[diabetic_data$diag_3 >="250." & diabetic_data$diag_3 <=
"250.999999999"] <- name
diabetic_data$diag_1[diabetic_data$diag_1 =="250"] <- name
diabetic_data$diag_2[diabetic_data$diag_2 =="250"] <- name
diabetic_data$diag_3[diabetic_data$diag_3 =="250"] <- name


#Other
name <- "Other"
diabetic_data$diag_1[diabetic_data$diag_1 != "Circulatory" & diabetic_data$diag_1 !=
"Respiratory" & diabetic_data$diag_1 != "Digestive" & diabetic_data$diag_1 !=
"Diabetes" & diabetic_data$diag_1 != "Injury" & diabetic_data$diag_1 !=
"Musculoskeletal" & diabetic_data$diag_1 != "Genitourinary" & diabetic_data$diag_1
!= "Neoplasms"] <- name
diabetic_data$diag_2[diabetic_data$diag_2 != "Circulatory" & diabetic_data$diag_2 !=
"Respiratory" & diabetic_data$diag_2 != "Digestive" & diabetic_data$diag_2 !=
"Diabetes" & diabetic_data$diag_2 != "Injury" & diabetic_data$diag_2 !=
"Musculoskeletal" & diabetic_data$diag_2 != "Genitourinary" & diabetic_data$diag_2
!= "Neoplasms"] <- name
```

```
diabetic_data$diag_3[diabetic_data$diag_3 != "Circulatory" & diabetic_data$diag_3 != "Respiratory" & diabetic_data$diag_3 != "Digestive" & diabetic_data$diag_3 != "Diabetes" & diabetic_data$diag_3 != "Injury" & diabetic_data$diag_3 != "Musculoskeletal" & diabetic_data$diag_3 != "Genitourinary" & diabetic_data$diag_3 != "Neoplasms"] <- name


diabetic_data$diag_1 = as.factor(diabetic_data$diag_1)

diabetic_data$diag_2 = as.factor(diabetic_data$diag_2)

diabetic_data$diag_3 = as.factor(diabetic_data$diag_3)
```

The new distribution it follows, as we can see below, is quite more informative:

```{r}
plot(diabetic_data$diag_1)
```

Now the transformation over the three variables is done, we will want to see the summary:

```{r}

print(dim(diabetic_data))

summary(diabetic_data)
```

We will now reduce the names of our variables for better readibility on analysis such as PCA, with the following mapping
```{r echo=FALSE}
keys <- c("admission_type_id", "admission_source_id","discharge_disposition_id",

"time_in_hospital", "medical_specialty", "number_emergency",

"number_inpatient", "num_lab_procedures", "num_procedures",

"num_medications", "number_outpatient","number_diagnoses", "patient_nbr")

shortened <- c("adm_type_id", "adm_source_id","disch_id",

"time_in_hpt", "specialty", "n_emerg",

"n_inp", "n_lab_proc", "n_proc",

"n_med", "n_outp", "n_diag","patient_n")

names(shortened) <- keys

print(data.frame("abbreviated"=shortened))


currCols <- colnames(diabetic_data)

for (currCol in 1:length(currCols)){

 colName <- currCols[currCol]

 if (colName %in% keys){

   colnames(diabetic_data)[currCol] <- shortened[[colName]]

 }

}
```

Now that the deed is done, we take a final look at our data:

```{r}

summary(diabetic_data)
```

```
```

And store it already processed, so we don't have to run this code again.

```{r}
write.csv(diabetic_data, file="processed_data.csv")
```
```

# ACPCode.r

```
READING CREDSCO_BIN
#
load("d:/karina/docencia/sreferenciesppt/16.AssociatiusVisualitzacio/MultivariateAnalysis/PracticaR/cr
edscok_bin")

# setwd("D:/karina/docencia/areferenciesPPT/DadesPractiques/CREDSCO")
# dd <- read.table("credscoClean.csv",header=T, sep=";");
setwd("C:/Users/guill/Documents/GitHub/DataMiningOverDiabetics")
dd <- read.csv("processed_data.csv")

objects()
attributes(dd)

#
VISUALISATION OF DATA
#
PRINCIPAL COMPONENT ANALYSIS OF CONTINcUOUS VARIABLES, WITH Dictamen PROJECTED AS ILLUSTRATIVE
#

CREATION OF THE DATA FRAME OF CONTINUOUS VARIABLES

attach(dd)
names(dd)

#set a list of numerical variables

dcon <- data.frame (n_diag,n_inp,n_outp,n_emerg,n_med,n_proc,n_lab_proc,time_in_hpt)

PRINCIPAL COMPONENT ANALYSIS OF dcon

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

print(pc1)

WHICH PERCENTAGE OF THE TOTAL INERTIA IS REPRESENTED IN SUBSPACES?

pc1$sdev
inerProj<- pc1$sdev^2
inerProj
totalIner<- sum(inerProj)
totalIner
pinerEix<- 100*inerProj/totalIner
pinerEix
png(filename="ACP/InertiaBarplot.png", width=1920, height=1080,units="px")
barplot(pinerEix)
dev.off()
```
```

```
#Cummulated Inertia in subspaces, from first principal component to the 8th dimension subspace
png(filename="ACP/Inertia2x2.png", width=1920, height=1080,units="px")
par(mfrow=c(2,2))
plot(pinerEix,xlab="Principal component", ylab="% of inertia", ylim=c(0,100), type='b')
plot(cumsum(inerProj/sum(inerProj)*100),xlab="Principal component", ylab="Cumulative % of inertia",
ylim=c(0,100), type='b')
screeplot(pc1)
screeplot(pc1,type="l")
dev.off()
png(filename="ACP/Inertia1x1.png", width=1920, height=1080,units="px")
par(mfrow=c(1,1))

tmp <- 100*cumsum(pc1$sdev[1:dim(dcon)[2]]^2)/dim(dcon)[2]
names(tmp) <- c(1,2,3,4,5,6,7,8)
barplot(tmp)
plot(c(1:8),tmp,type='o',xlab="Principal component", ylab="Cumulative % of inertia")
dev.off()
percInerAccum<-tmp
percInerAccum


# SELECTION OF THE SINGIFICNT DIMENSIONS (keep 80% of total inertia)

nd = 6

# STORAGE OF THE EIGENVALUES, EIGENVECTORS AND PROJECTIONS IN THE nd DIMENSIONS


Psi = pc1$x[,1:nd]

# STORAGE OF LABELS FOR INDIVIDUALS AND VARIABLES

iden = row.names(dcon)
etiq = names(dcon)
ze = rep(0,length(etiq)) # WE WILL NEED THIS VECTOR AFTERWARDS FOR THE GRAPHICS


# PLOT OF INDIVIDUALS

#select your axis
eje1<-1
eje2<-2
eje3<-3
png(filename="ACP/Individual1.png", width=1920, height=1080,units="px")
plot(Psi[,eje1],Psi[,eje2])
text(Psi[,eje1],Psi[,eje2],labels=iden, cex=1) # vvv changed cex from 0.5 vvv
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")
dev.off()
png(filename="ACP/Individual2.png", width=1920, height=1080,units="px")
plot(Psi[,eje1],Psi[,eje3])
text(Psi[,eje1],Psi[,eje3],labels=iden, cex=1)
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")
dev.off()
png(filename="ACP/Individual3.png", width=1920, height=1080,units="px")
plot(Psi[,eje2],Psi[,eje3])
text(Psi[,eje2],Psi[,eje3],labels=iden, cex=1)
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")
dev.off()

#Projection of variables
```

```
Phi = cor(dcon,Psi)


plot.with.zooms <- function(minx,maxx,miny,maxy, axis1, axis2){
  globalCEX = 2.5
  X<-Phi[,axis1]
  Y<-Phi[,axis2]
  #all qualitative together with zooms
  plot(Psi[,axis1],Psi[,axis2],type="n",xlim=c(minx,maxx), ylim=c(miny,maxy))
  axis(side=1, pos= 0, labels = F, col="cyan")
  axis(side=3, pos= 0, labels = F, col="cyan")
  axis(side=2, pos= 0, labels = F, col="cyan")
  axis(side=4, pos= 0, labels = F, col="cyan")

  arrows(ze, ze, X, Y, length = 0.07,col="blue")
  text(X,Y,labels=etiq,col="darkblue", cex=globalCEX) #changed cex from 0.7

  #nominal qualitative variables
  dcat<-c(4,5,12,13,20,21,22,24,25,26,27,28,29,30)
  #divide categoricals in several graphs if joint representation saturates
  #build a palette with as much colors as qualitative variables
  colors<-rainbow(length(dcat))
  c<-1
  for(k in dcat){
    seguentColor<-colors[c]
    fdic1 = tapply(Psi[,axis1],dd[,k],mean)
    fdic2 = tapply(Psi[,axis2],dd[,k],mean)

    text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=globalCEX) #changed cex from 0.6
    c<-c+1
  }
  # ordinal
  dordi<-c(6,7)
  dd[,dordi[2]] <- factor(dd[,dordi[2]], ordered=TRUE,  levels= c("[0-25)", "[25-50)", "[50-75)",
"[75-100)", "[100-125)", "[125-150)", "[150-175)", "[175-200)",">200"))
  c<-1
  col <- 1
  for(k in dordi){
    seguentColor<-colors[col]
    fdic1 = tapply(Psi[,axis1],dd[,k],mean)
    fdic2 = tapply(Psi[,axis2],dd[,k],mean)

    #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
    #connect modalities of qualitative variables
    lines(fdic1,fdic2,pch=16,col=seguentColor)
    text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=globalCEX) #changed cex from 0.6
    c<-c+1
    col<-col+1
  }
  legend("bottomleft",names(dd)[dcat],pch=3.25,col=colors, cex=globalCEX) #vvv changed cex from 0.6,
pch from 1
  legend("bottomright",names(dd)[dordi],pch=3.25,col=colors[1:length(dordi)], cex=globalCEX)

}


# Axis 1,2
png(filename="ACP/PlotWithZooms1-2(1).png", width=1920, height=1080,units="px")
plot.with.zooms(-4,4,-4,1,1,2)
dev.off()
png(filename="ACP/PlotWithZooms1-2(2).png", width=1920, height=1080,units="px")
plot.with.zooms(-1,1,-1,1,1,2)
dev.off()
png(filename="ACP/PlotWithZooms1-2(3).png", width=1920, height=1080,units="px")
plot.with.zooms(-2,2,-1,1,1,2)
dev.off()

# Axis 1,3
png(filename="ACP/PlotWithZooms1-3(1).png", width=1920, height=1080,units="px")
plot.with.zooms(-5,4,-5,2,1,3)
dev.off()
```

```
png(filename="ACP/PlotWithZooms1-3(2).png", width=1920, height=1080,units="px")
plot.with.zooms(-1,1,-1,1,1,3)
dev.off()

# Axis 2,3
png(filename="ACP/PlotWithZooms2-3(1).png", width=1920, height=1080,units="px")
plot.with.zooms(-4,3,-3,2,2,3)
dev.off()
png(filename="ACP/PlotWithZooms2-3(2).png", width=1920, height=1080,units="px")
plot.with.zooms(-2,2,-2,2,2,3)
dev.off()
png(filename="ACP/PlotWithZooms2-3(3).png", width=1920, height=1080,units="px")
plot.with.zooms(-1,1,-1,1,2,3)
dev.off()
```

Clustering.R

```
base_path <- "C:/Users/danie/Documents/MD/diab/DataMiningOverDiabetics"
setwd(file.path(base_path))
diabetic_data <- read.csv("processed_data.csv", na.strings = c("?"))

names(diabetic_data)
dim(diabetic_data)
summary(diabetic_data)

attach(diabetic_data)

#hierarchical clustering

library(cluster)
#install.packages("fpc")
library(fpc)

#dissimilarity matrix
actives<-c(4:11,13:28,30) #exclude row identifiers, non significant variables and response variable
dissimMatrix <- daisy(diabetic_data[,actives], metric = "gower", stand=TRUE)

distMatrix<-dissimMatrix^2

h1 <- hclust(distMatrix,method="ward.D")

K<-10 #we want to see 10 partitions
CHIndexes <- array(dim=10)
Silhouettes <- array(dim=10)
for (k in 2:K) {
  ck <- cutree(h1,k)
  stats <- cluster.stats(distMatrix, ck)
  CHIndexes[k] <- stats$ch
  Silhouettes[k] <- stats$avg.silwidth
}
plot(CHIndexes, type="o", xlab="Number of clusters", ylab="CH index")
plot(Silhouettes, type="o", xlab="Number of clusters", ylab="Average silhouette")

#The number of clusters is the max of CH indexes and Silhouette (excluding the 2 clusters partition)
n_clusters = 4

c1 <- cutree(h1,n_clusters)

plot(h1, labels = FALSE)
rect.hclust(h1, k = n_clusters)
```

```
#insert again the response variable


dcon <- data.frame (race, gender, age, weight, adm_type_id, disch_id, adm_source_id, time_in_hpt,
specialty, n_lab_proc, n_proc, n_med, n_outp, n_emerg, n_inp, diag_1, diag_2, diag_3, A1Cresult,
metformin, insulin, change, diabetesMed, readmitted, other_meds)
png("all_vars_pairs.png", width=20, height=20, units="in", res=500)
pairs(dcon[,1:25], col=c1)
dev.off()

dcon <- data.frame (age, n_lab_proc, n_med, time_in_hpt, n_outp, disch_id)
png("some_vars_pairs2.png", width=20, height=20, units="in", res=500)
pairs(dcon[,1:6], col=c1)
dev.off()


plot(n_med, n_lab_proc,col=c1,main="Clustering of credit data in 3 classes")
plot(n_med, age,col=c1,main="Clustering of credit data in 3 classes")
plot(n_med, time_in_hpt,col=c1,main="Clustering of credit data in 3 classes")
plot(weight, adm_type_id,col=c1,main="Clustering of credit data in 3 classes")
plot(n_med, disch_id,col=c1,main="Clustering of credit data in 3 classes")

#trying to display the discrete variables as continuous to avoid problems
for (row in 1:nrow(diabetic_data)) {
  diabetic_data[row, "disch_id"] <- diabetic_data[row, "disch_id"] + runif(1, -1.0, 1.0)
  #disch_id <- disch_id + runif(1, -1.0, 1.0)
}
```

CPG.R

```
createCPG<- function(data, response)
{
  if (!is.factor(response))
  {
    cat("The variable ", names(response), " must be a factor" )
  }
  else
  {
    #alerta! el maxim es 7 per fila
    #sembla que 3 per columna ho fa amb numeriques. Mes ja no se. Qualis donen problemes
    plotConditionalTable(data, response)
  }#end else
}#endcreateCPG



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

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

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

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

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

  for (k in 1:K){
    dir.create(file.path(names(dades)[k]))
    setwd(file.path(names(dades)[k]))
    Vnum<-data[,k]
    for(niv in levels(response)){
      print(niv)
      s<-subset(Vnum, response==niv)
      if(is.numeric(data[,k]))
      {
        png(filename=paste(names(data)[k],"_histograma_", niv ,".png",sep=""), width=800, height=800)
        hist(s, main=paste(names(data)[k], niv))
        dev.off()
        #evenctually add other summary statistics, like vc
      }else{
        png(filename=paste(names(data)[k],"_barplot", niv, ".png",sep=""), width=800, height=800)
        barplot(table(s), las=3, cex.names=1, main=paste("Barplot of", names(data)[k]))
        dev.off()
      }#endifelse
    }#end for niv
    setwd(file.path(".."))
    }#end for k
}#end plot conditional table
```

Profiling.R

```
setwd("E:/Marc/Cole/Uni/7eQ/MD/DataMiningOverDiabetics");
dd <- read.csv("processed_data.csv", na.strings = c("?"))
names(dd)

attach(dd)

#Dictamen    <- as.factor(Dictamen)
#levels(Dictamen) <- c(NA, "positiu","negatiu")

actives<-c(4:11,13:30)

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

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


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

P<-c1
nc<-length(levels(as.factor(P)))
pvalk <- matrix(data=0,nrow=nc,ncol=K, dimnames=list(levels(P),names(dades)))
nameP<-"Class"
n<-dim(dades)[1]


setwd(file.path("./Profiling"))
for(k in 1:K){
  dir.create(file.path(names(dades)[k]))
  setwd(file.path(names(dades)[k]))

  if (is.numeric(dades[,k])){
    print(paste("An�lisi per classes de la Variable:", names(dades)[k]))

    png(filename=paste(names(dades)[k],"_Boxplot",".png",sep=""), width=800, height=800)
    boxplot(dades[,k]~P, main=paste("Boxplot of", names(dades)[k], "vs", nameP ), horizontal=TRUE,
cex=1.2)
    dev.off()


    png(filename=paste(names(dades)[k],"_Barplot",".png",sep=""), width=800, height=800)
    barplot(tapply(dades[[k]], P, mean),main=paste("Means of", names(dades)[k], "by", nameP ))

    abline(h=mean(dades[[k]]))
    legend(0,mean(dades[[k]]),"global mean",bty="n")
    dev.off()

    print("Estad�stics per groups:")
    for(s in levels(as.factor(P))) {print(summary(dades[P==s,k]))}
    o<-oneway.test(dades[,k]~P)
    print(paste("p-valueANOVA:", o$p.value))
    kw<-kruskal.test(dades[,k]~P)
    print(paste("p-value Kruskal-Wallis:", kw$p.value))
    pvalk[,k]<-ValorTestXnum(dades[,k], P)
    print("p-values ValorsTest: ")
    print(pvalk[,k])
  }else{
    #qualitatives
    print(paste("Variable", names(dades)[k]))
    table<-table(P,dades[,k])
    #   print("Cross-table")
    #   print(table)
    rowperc<-prop.table(table,1)

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

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

#with legend
png(filename=paste(names(dades)[k],"_pos&neg",".png",sep=""), width=800, height=800)
plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(dades[,k])))
for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }
legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=1.2)
dev.off()

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

#with legend
png(filename=paste(names(dades)[k],"_pos&negCondClasse",".png",sep=""), width=800, height=800)
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(dades[,k])))
for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }
legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=1.2)
dev.off()

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

#with legend
png(filename=paste(names(dades)[k],"_pos&negGirat",".png",sep=""), width=800, height=800)
plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c])}
legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=1.2)
dev.off()

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

#with legend
png(filename=paste(names(dades)[k],"_pos&negCondColumn",".png",sep=""), width=800, height=800)
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c])}
legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=1.2)
dev.off()

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

#diagrames de barres apilades

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

png(filename=paste(names(dades)[k],"_barplotApilades",".png",sep=""), width=800, height=800)
barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )
legend("topright",levels(as.factor(dades[,k])),pch=1,cex=1.2, col=paleta)
dev.off()
```

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

    png(filename=paste(names(dades)[k],"_barplotAdosades",".png",sep=""), width=800, height=800)
    barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta)
    legend("topright",levels(as.factor(dades[,k])),pch=1,cex=1.2, col=paleta)
    dev.off()

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

    print("valorsTest:")
    print( ValorTestXquali(P,dades[,k]))
  }
  setwd(file.path(".."))
}#endfor
setwd(file.path(".."))
################ Arreglar pvalues


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

#afegir la informacio de les modalitats de les qualitatives a la llista de pvalues i fer ordenacio
global

#saving the dataframe in an external file
#write.table(dd, file = "credscoClean.csv", sep = ";", na = "NA", dec = ".", row.names = FALSE,
col.names = TRUE)


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

#Fer gran la finestra del R
#createCPG(dd[,active], Dictamen)

#comparar una variable amb les altres

detach(dd)
detach(dades)
detach(diabetic_data)

dades<-dd[,actives]
attach(dades)
plotConditionalTable(dades[,1:2], readmitted)


#cada numero es la columna
#fer creixer la finestra de plots
#control - per fer menor el tipus de lletra en R
#cambiar els valors d'active per mostrar les diferents variables
active<-c(24:25)
createCPG(dades[,active], as.factor(c1))
```