Data Preprocessing Report

0. Table of Contents

X

1. Introduction

blablabla

X

2. Project Methodology

As this is a *data preprocessing* project, our pipeline concerns only the first 3 steps of the SEMMA process: Sample, Explore and Modify

- **Sample**: We will consider the transactional table a representative sample. The data will be imported with SAS Miner Enterprise.
- **Explore**: We will do exploratory data visualization on the data, to know which aspects of the dataset need particular attention.
- **Modify**: We will treat problems detected previously, mainly through two applications: SAS Miner Enterprise and SAS Guide.

#TODO GRAPHIC OF PARTIAL SEMMA

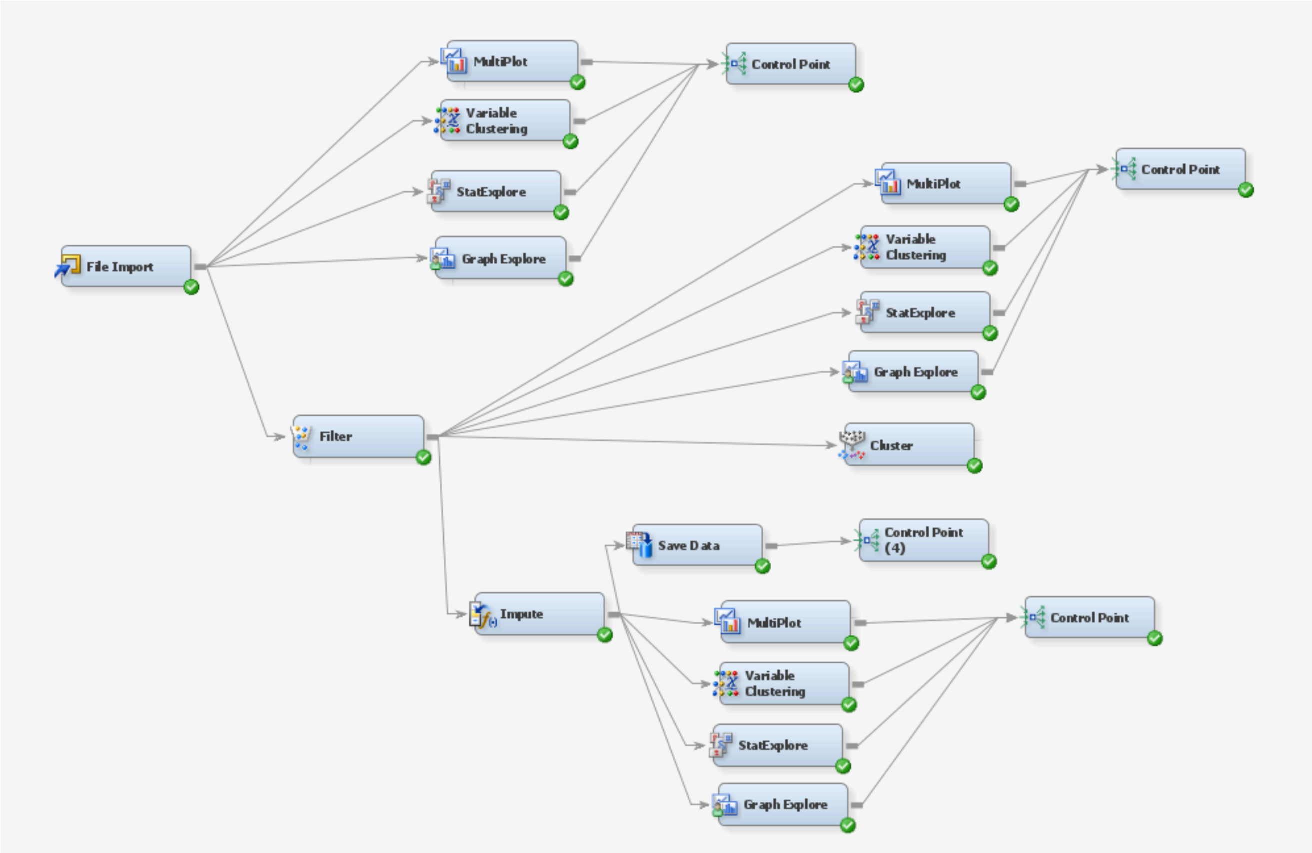
Then, we will also build an Analytic Base Table to obtain information about the customers.
In the end, we will perform data visualizations with PowerBI to gain business insights.

#TODO PIPELINE GRAPHIC

X

3. Data Exploration and Treatment

Let us present the workflow used to explore and treat data, in SAS Miner Enterprise.



(*fig 3.0.*, SAS Enterprise Miner's diagram for the project)

3.1. Phase 0: Exploratory Data Analysis

Metadata

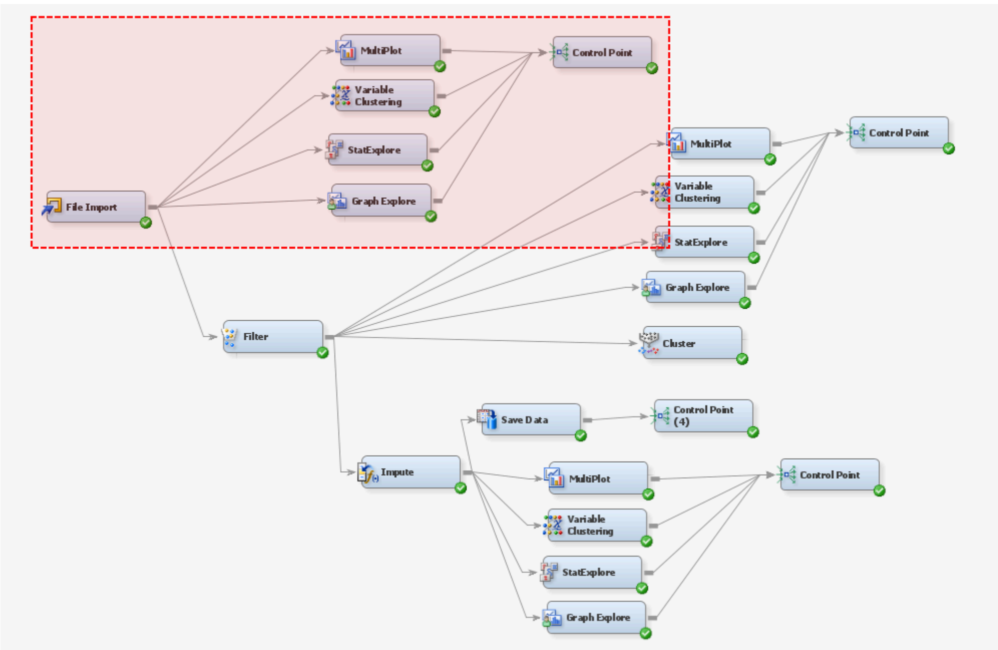
Before delving into technical details, we will explore by dataset by reading its metadata first (fig 3.1.), to gain an understanding of the business details.

Variable	Description
Patient ID	Unique identification of the patient
Age	Patient age
Gender	Patient gender (Male, Female, Other)
City of Residence	Patient city of residence
Profession	Patient profession
Insurance Provider	Patience insurance provider
Family History	Patient family history diseases
Education Level	Patient education level
Marital Status	Patient marital status
Visit Date	Date of the consultation
Department	Consultation department
Consultation Duration	Consultation duration in minutes
Satisfaction Level	Patient evaluation of the satisfaction level with the consultation (1-5)
Approximate Annual Income	Patient approximate annual income
Consultation Price	Consultation price (pounds)
Insurance Coverage	Amount of the consultation price covered by the insurance provider (pounds)

(*fig 3.1.*, metadata provided by project guidelines)

The initial dataset provided City Hospital is a transactional table containing information about each patient visit; therefore it is crucial to ensure that each transaction has correct values, in order to perform clustering on the transactions and patients. The dataset contains information about 10008 transactions.

EDA With SAS Miner Enterprise



(*fig 3.2.*, EDA with SAS Miner Enterprise)

Then we performed an initial inspection of the dataset through SAS Enterprise Miner, with the nodes marked in the red zone (fig 3.2.).

StatExplore

To get a good idea of the data, we took a quick glance at the variables' statistics through StatExplore.

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	City_of_Residence	INPUT	8	0	Birmingham	14.72	Belfast	14.10
TRAIN	Department	INPUT	13	0	Psychiatry	13.60	General Practice	13.30
TRAIN	Education_Level	INPUT	9	29	Undergraduate	41.57	Master	33.88
TRAIN	Family_History	INPUT	5	0	Heart Disease	22.33	Hypertension	20.47
TRAIN	Gender	INPUT	3	0	Other	34.14	Female	33.77
TRAIN	Insurance_Provider	INPUT	6	104	Provider D	21.63	Provider A	20.03
TRAIN	Marital_Status	INPUT	4	0	Divorced	28.91	Single	28.56
TRAIN	Profession	INPUT	10	0	Retired	35.98	Student	20.96
TRAIN	Satisfaction_Level	INPUT	6	0	2	18.89	4	18.87

(*fig 3.3.*, StatExplore on categorical variables)

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Age	INPUT	50.63565	31.18561	9952	56	0	52	195	0.28614	-0.21493
Approximate_Annual_Income	INPUT	43402.76	268142	9854	154	0	40874	11970900	42.21076	1835.703
Consultation_Duration	INPUT	67.80765	32.44714	10008	0	15	68	600	1.545225	21.11949
Consultation_Price	INPUT	187.263	862.8655	10008	0	50.03676	159.5248	39999.22	39.85869	1655.995
Insurance_Coverage	INPUT	115.4294	79.33776	9958	50	0	115.9291	421.8878	0.322906	-0.17236

(*fig 3.4.*, StatExplore on Numerical Variables)

Categorical Variables. In terms of category variability, all variables seem to not present any type of problem. In other words, there are no variables with a single class.
In terms of missing values, we have two problematic variables: `Education_Level` and `Insurance_Provider`

- `Education_Level` is potentially due to lack in measurements and it could be *"Missing at Random"*, as certain customers might have not been comfortable sharing such information.
- `Insurance_Provider` could be potentially due to non-applicability situations, meaning that some customers could have not had an insurance provider at all.
- There are six classes on `Satisfaction_Level`, when there should be five. This may suggest that a class which should not exist, is there (we will see later that it turns out to be level six)
  We will consider imputing missing variables with a classifier.

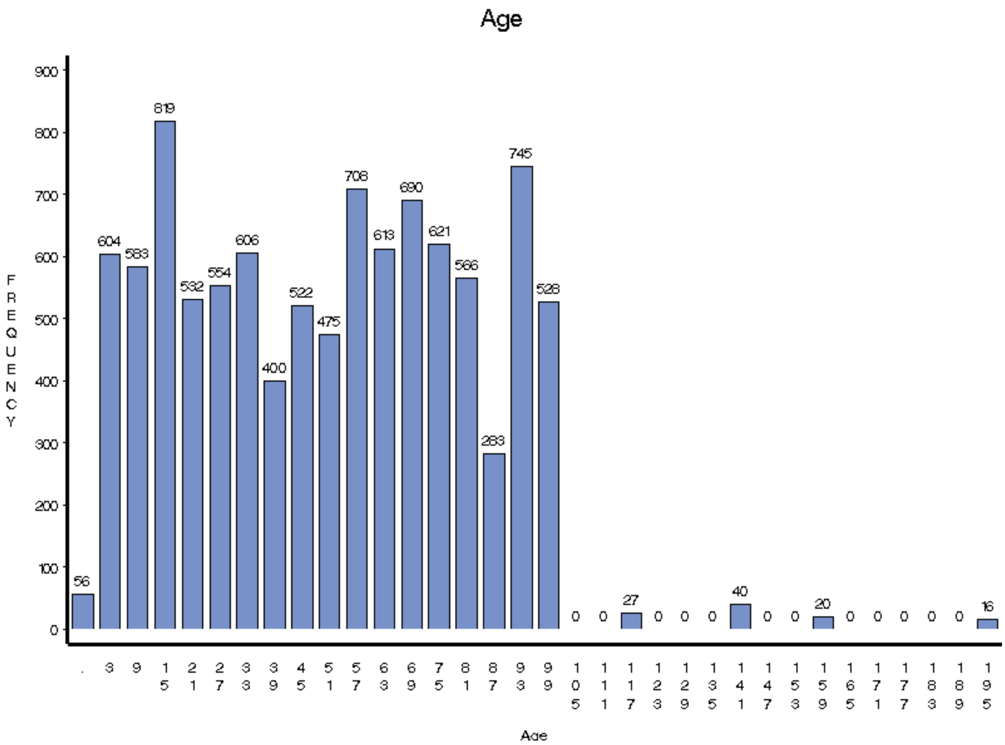
Numerical Variables. In the numerical variables we can already notice a few problems:

- In `Age` the maximum is 195, which is clearly an error in data measurement
- There are "extreme outliers" with `Approximate_Annual_Income` and `Consultation_Price`, as they have extremely high standard deviations: these could "ruin" our analysis of their distribution, which we will see in the next part.
- There are missing values in `Age`, `Approximate_Annual_Income` and `Insurance_Coverage`. They will be imputed through a regressor.

MultiPlot

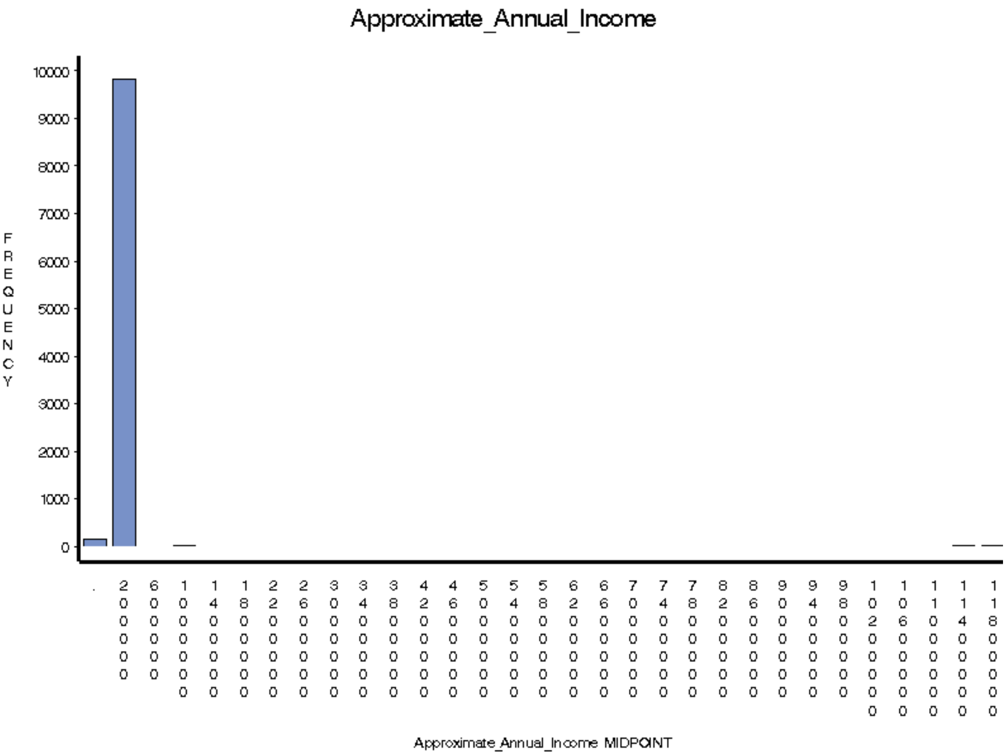
Successively, we took a look at the variables' distributions through the MultiPlot node. Therefore, we will proceed on a case-by-case basis to analyze each variable.

- **Age**: As detected before, there are outliers with patients that have age > 111. Also, there are missing variables (56). The variable does not seem to follow any particular type of distribution, more tending towards uniformity.

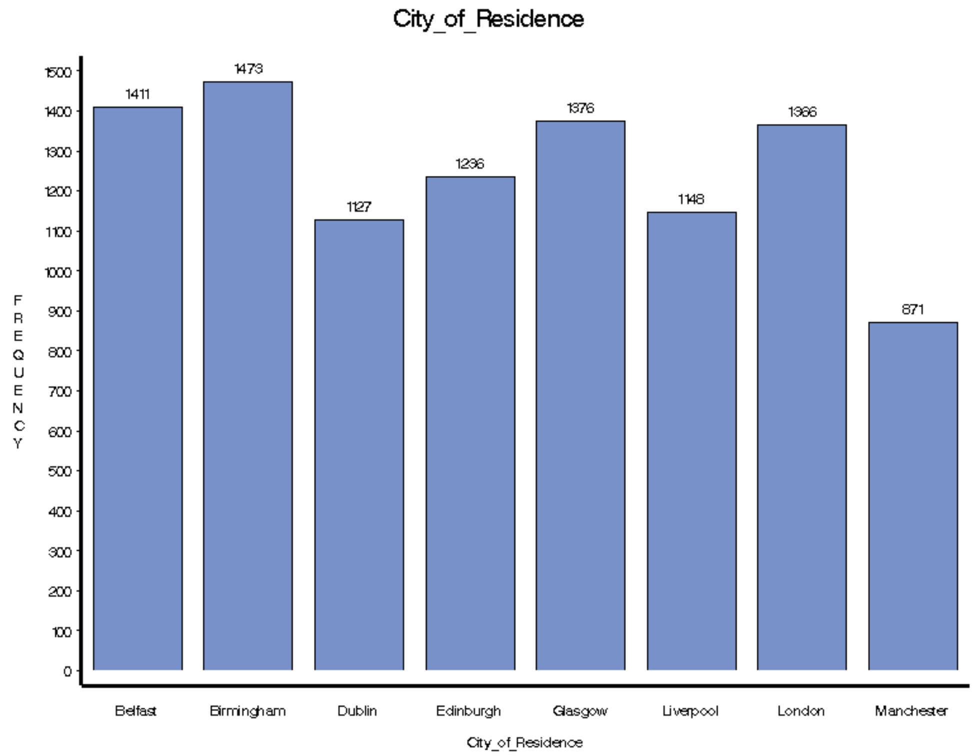


(*figure 3.6.*)

- **Approximate Annual Income**: In this case, the outliers are so "extreme" that it is impossible to analyze the variable's distribution; this is clearly a case of the "Bill Gates" effect. Also, as discussed previously, there are missing values.

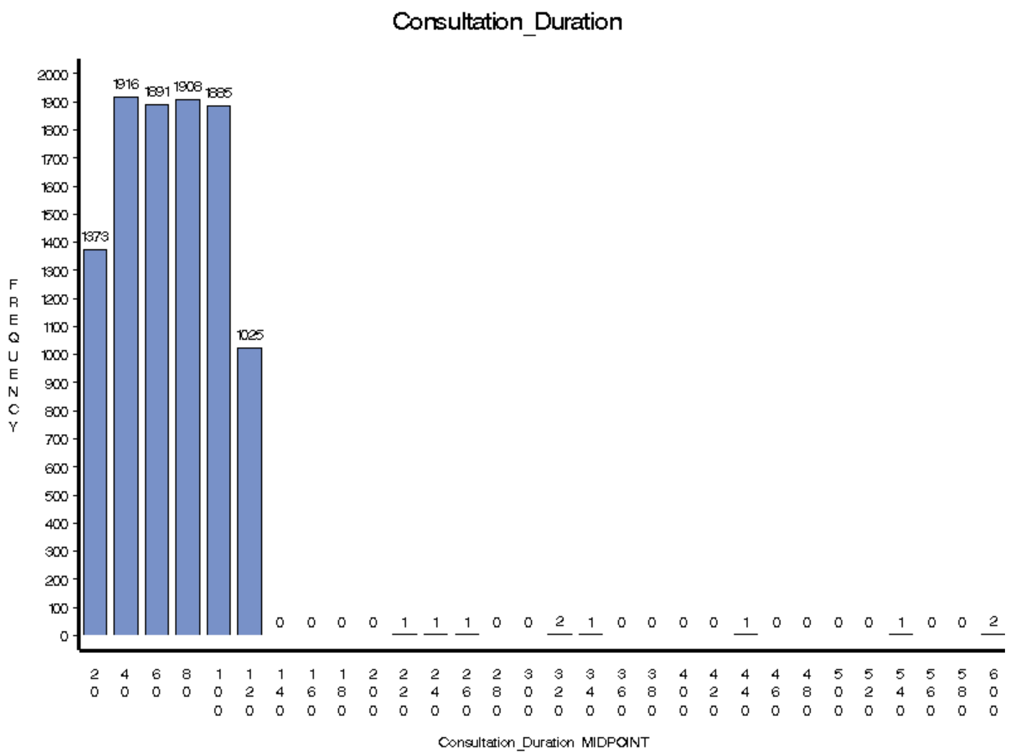


- **City of Residence**: No issues detected, cities of residence seem to be uniformly distributed between visitations.



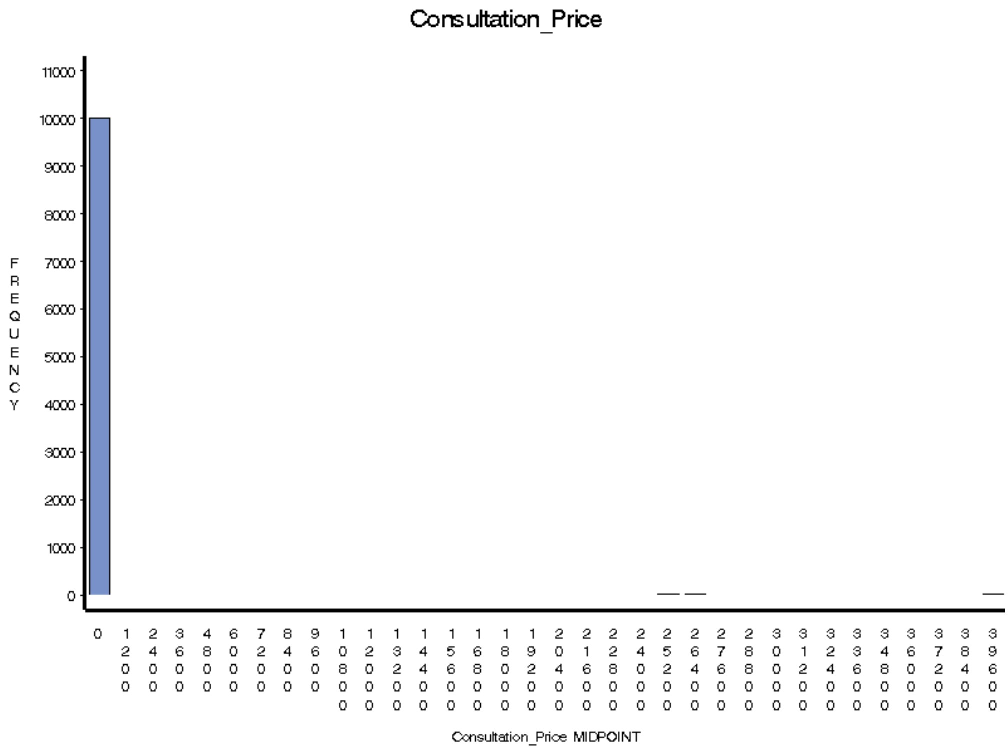
(*figure 3.7.*)

- **Consultation Duration**: Similarly to *Approximate Annual Income*, the outliers make it hard to analyze the variable's distribution: therefore we will postpone the distribution's analysis to post-cleaning analysis. It might seem that this follows some sort of normal distribution. There are no missing values detected here.



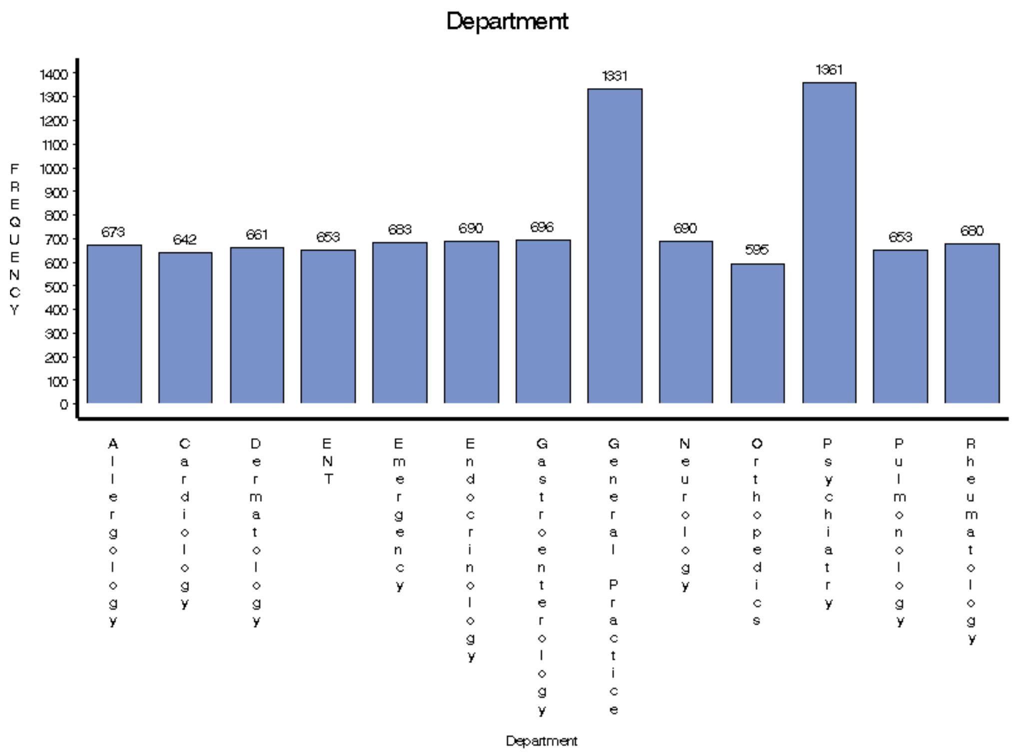
(*figure 3.8.*)

- **Consultation Price**: Same as above, the extreme outliers make it impossible to analyze the variable's distribution. So, we will postpone the analysis of this variable as nothing significant can be found.



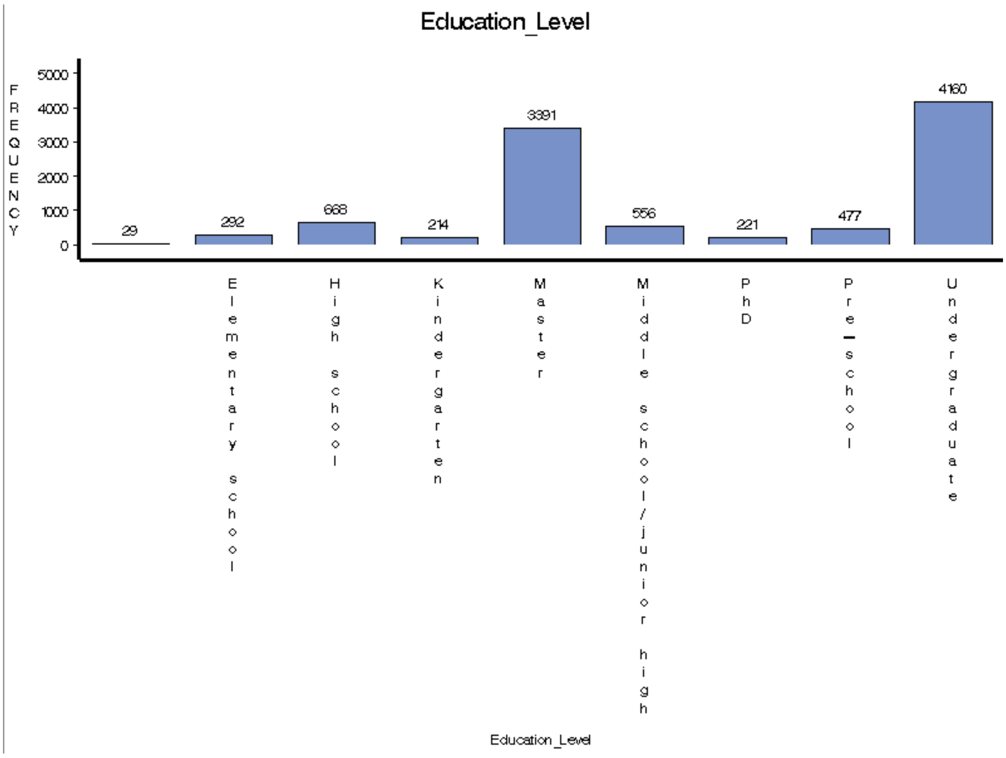
(*figure 3.9.*)

- **Department**: No issues detected. It seems to follow an uniform distribution, except for *General Practice* and *Psychiatry* departments as they have a slight peak, with more visits than everyone else.



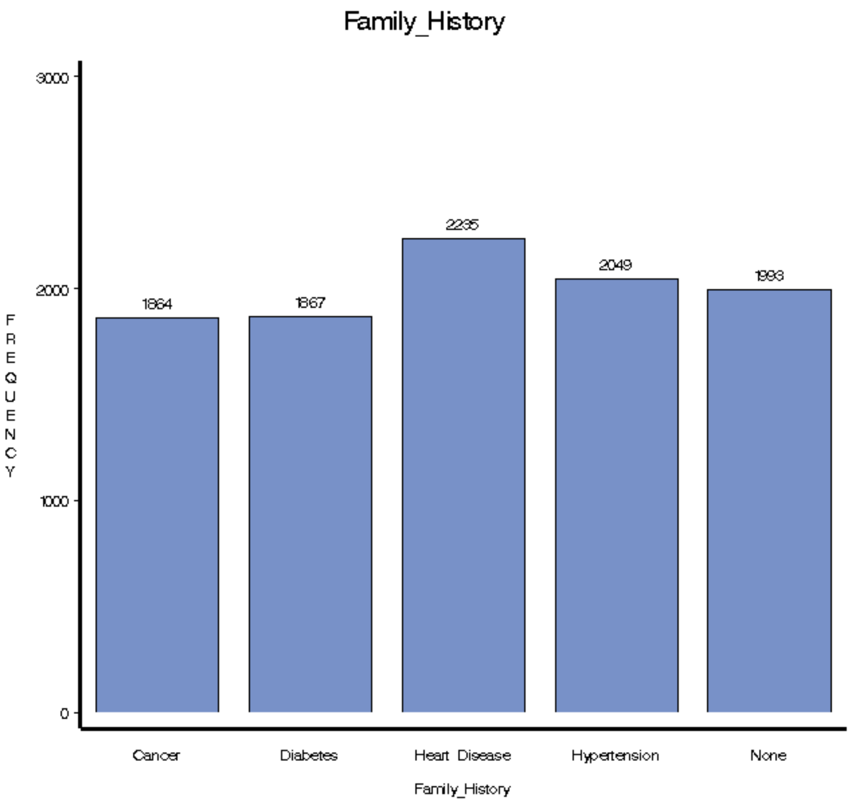
(*figure 3.10*)

- **Education Level**: Other than missing values (29), no problem is found. According to the distributions, it seems that there's a trend towards patients with Master's or Bachelor's degrees, occupying $\sim 77.45\%$ of the total transactions (or visits).



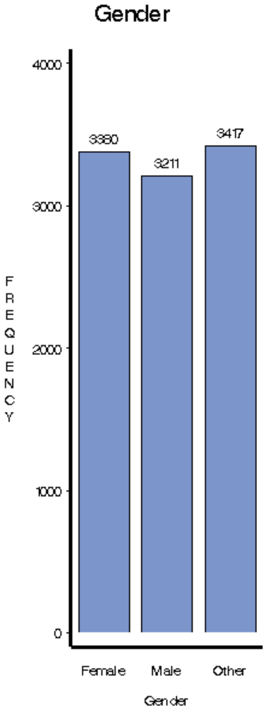
(*figure 3.11.*)

- **Family History**: The class *"None"* suggests that there might be missing values occupying a significant part of this variable (around 1/5ths); this might be a case of a missing variable being due to non-applicability, for instance cases of people whose family had no diseases. Therefore, if we consider *"None"* as a class of its own, we can say that this variable is uniformly distributed, with a slight trend towards heart disease.



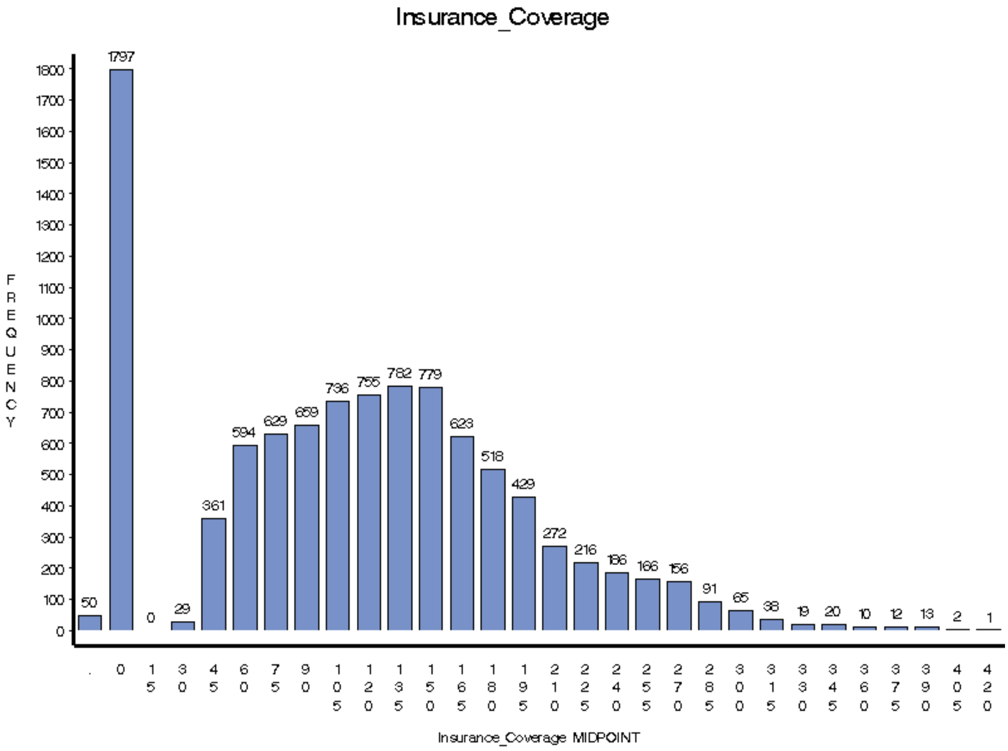
(*figure 3.12.*)

- **Gender**: No issues detected, variable follows an uniform distribution.



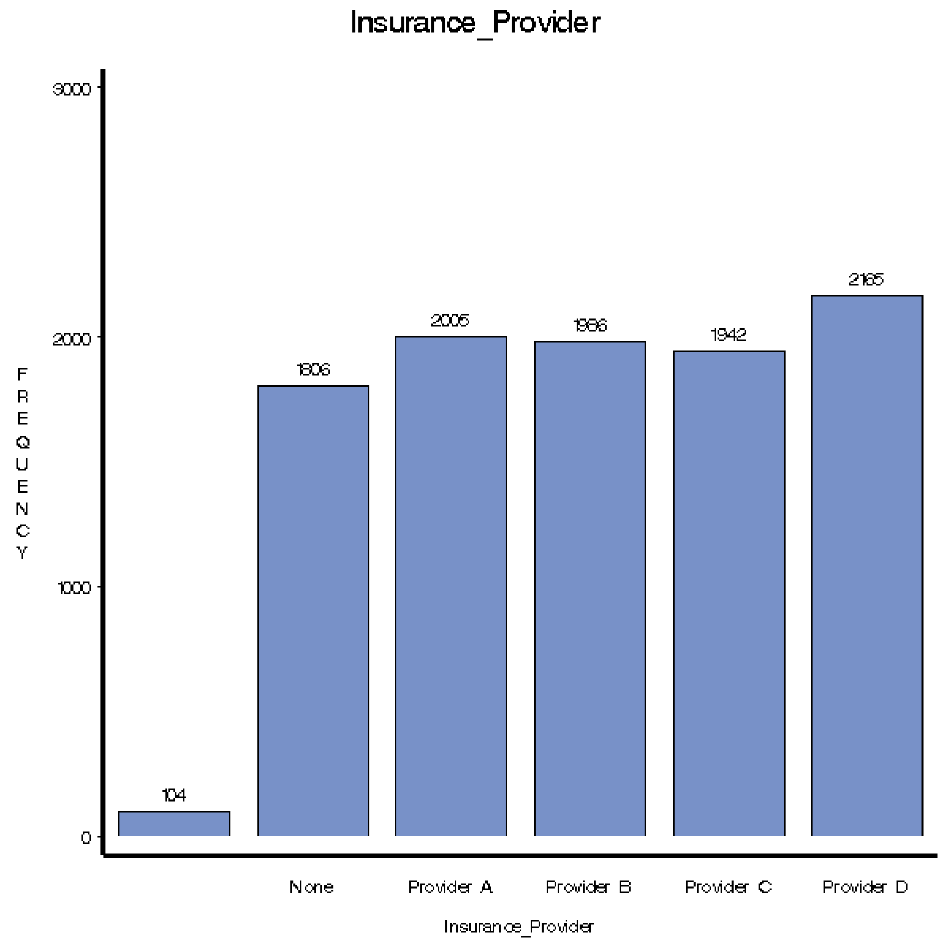
(*figure 3.13.*)

- **Insurance Coverage**: There are 50 missing values. We can gain an interesting insight about this variable: there's a peak of patients who had zero insurance coverage - potentially meaning that they had no insurance provider at all, as discussed previously - and ignoring this particular case, we have that the variable is slightly right-skewed.



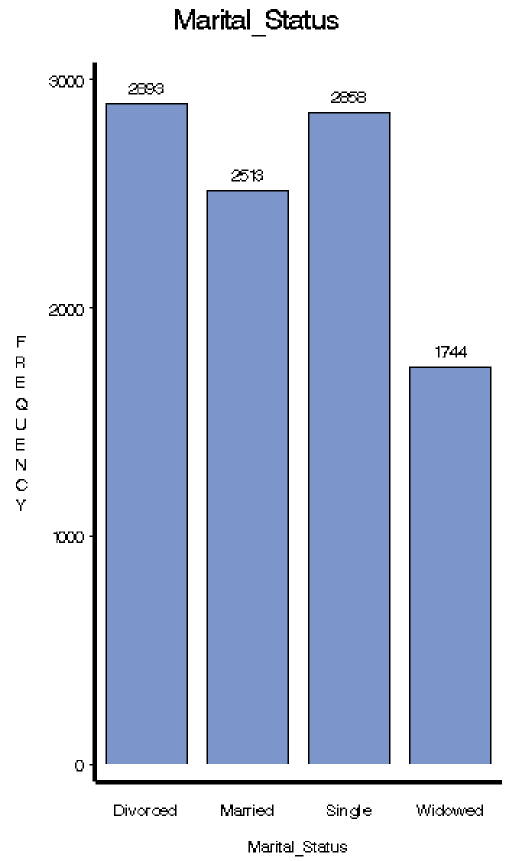
(*figure 3.14.*)

- **Insurance Provider**: There are missing values and also the *"None"* class: meaning that missing values are not necessarily to be *"None"* class, as they could be caused by errors in data measurement. Other than that, insurance providers seem to be uniformly distributed.



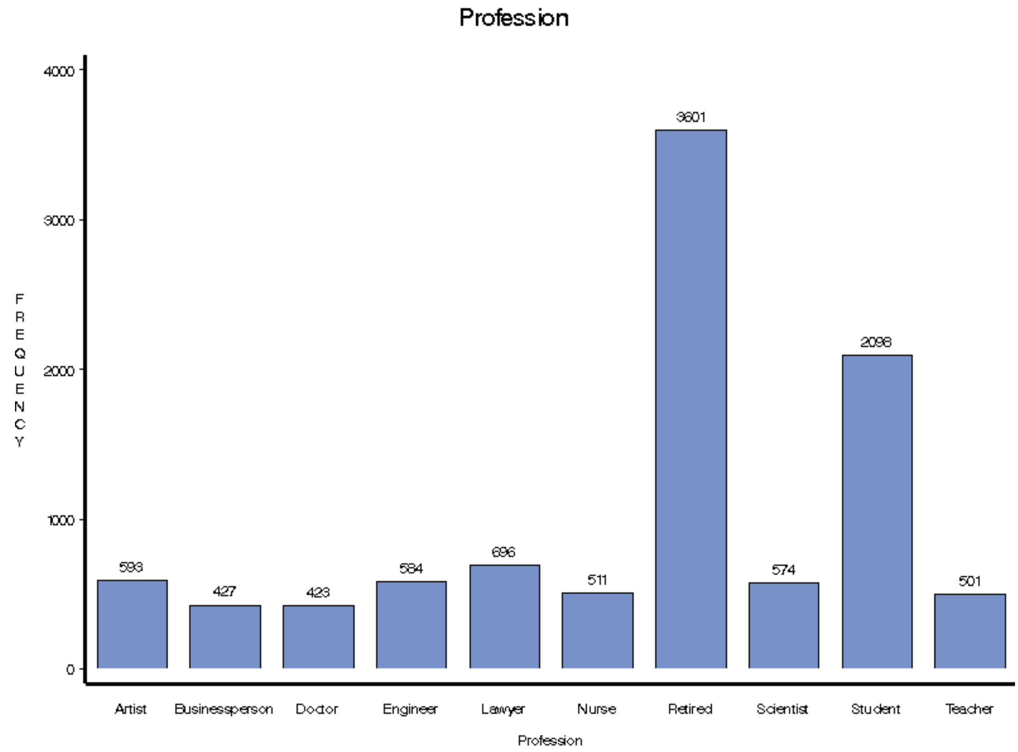
(*figure 3.15.*)

- **Marital Status**: No issues, there is a trend towards people who have been married (married, divorced and widowed). If we consider classes as their own, we cannot say anything about the classes distribution.



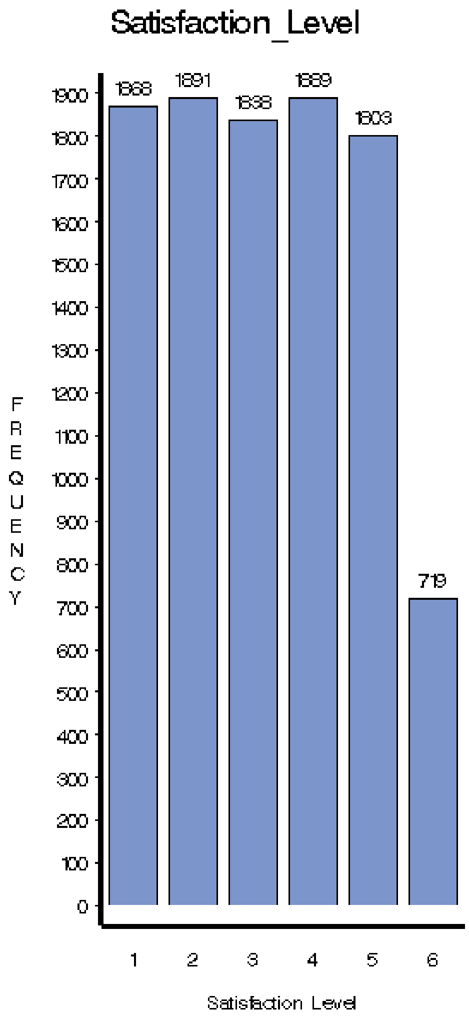
(*figure 3.16.*)

- **Profession**: No particular issues detected, classes other than "Retired" and "Student" seem to be uniformly distributed; there is a trend towards the two mentioned classes. This could suggest that most visits are either made by people of young or old age.



(*figure 3.17.*)

- **Satisfaction Level**: Classes seem to be uniformly distributed. However, there is an inconsistency between the existing classes and the metadata: the metadata suggests that levels should be from 1 to 5, meanwhile we actually have class "6", which should not exist. This could be due to errors in measurements, or this could be even a hidden missing value. In any case, this problem will be addressed in the part where we will check data inconsistencies.

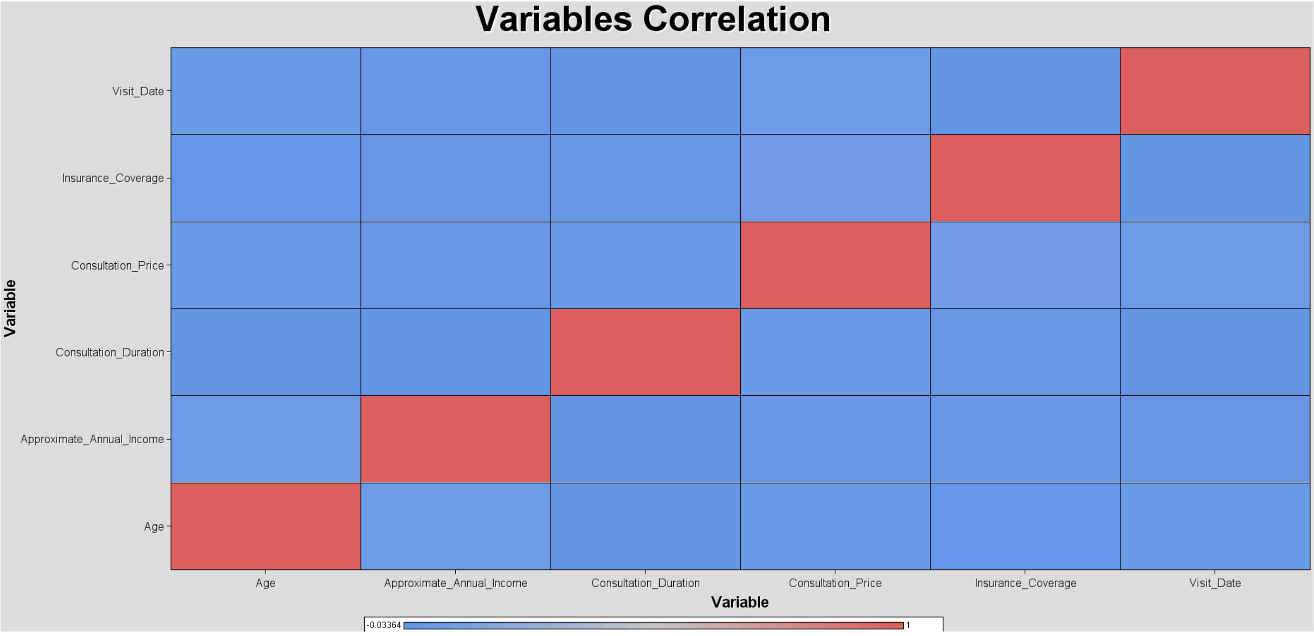


(*figure 3.18*)

Variable Clustering

Lastly, we took a look at the numerical variables' correlation with the *"Variable Clustering"* node. There seems to be no particular correlations, as all of them are inside the range $[-0.7, 0.7]$: all of the correlation values seem to be near 0.033 (figure 3.x.), which indicates a low amount of correlation between numerical variables.

However, this result is to be re-checked as we will clean the data from outliers and missing values.



(*figure 3.19.*, Correlation Matrix for numerical variables)

Python

With Pandas' library in Python we were able to extract information about the variable `Visit_Date`; it seems that all the visits happened in a time range from 1^{st} January 2024 to 6^{th} June 2024 (figure 3.y.). Therefore, we are talking about a time span of approximately 5 months; this insight will be relevant for data inconsistency checking purposes.

	Visit Date
count	10008
mean	2024-03-31 14:53:14.244604416
min	2024-01-01 00:00:00
25%	2024-02-15 00:00:00
50%	2024-03-31 00:00:00
75%	2024-05-16 00:00:00
max	2024-06-30 00:00:00

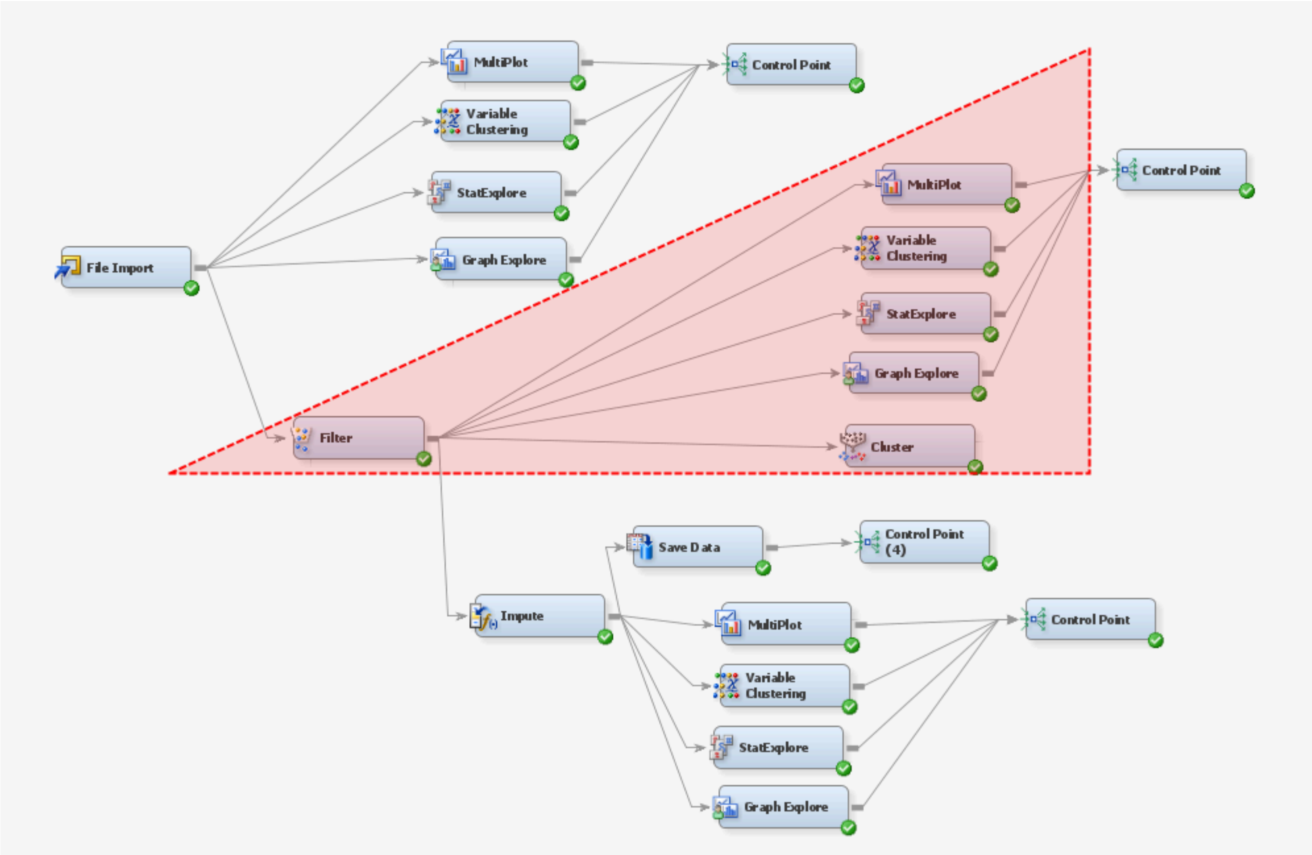
(*figure 3.20.*, Pandas' `.describe()` method on the `Visit_Date` variable)

3.2. Phase 1: Outliers and Missing Values Treatment

Let us remind the main problems with the data that have been detected previously:

- Outliers with Age
- Extreme outliers with Approximate Annual Income, Consultation Duration, Consultation Price
- Missing values with Age, Approximate Annual Income, Education Level, Insurance Coverage, Insurance Provider

Outliers

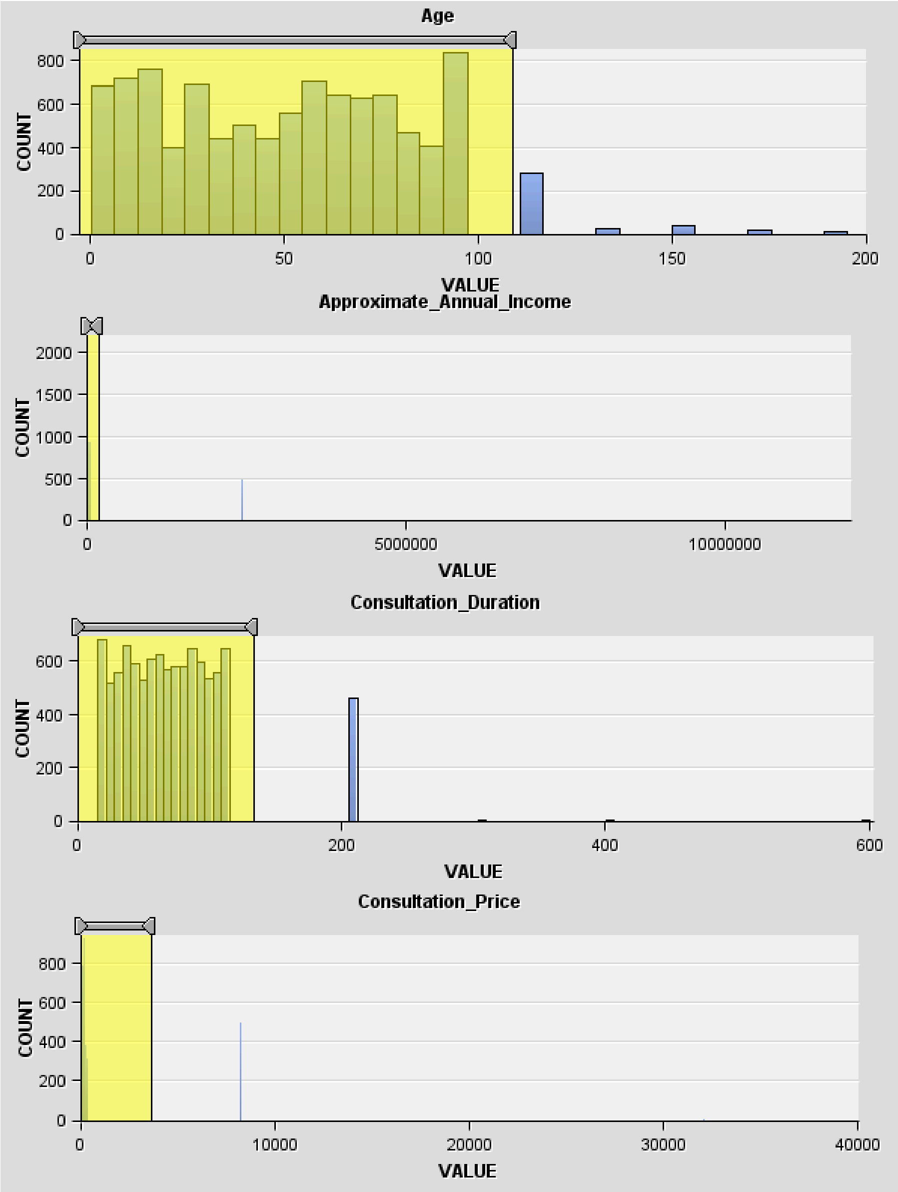


(*figure 3.21.*, nodes used for outliers filtering)

Let us address the outliers first, to not cause any biased predictions during the imputation of missing values.

To deal with one-dimensional outliers, we manually defined a limit for each variable as a "filter range". In other words, we arbitrarily defined a range for which the variables would be classified as an outlier and thus be filtered from the main dataset. To do this, we used the *"Filter"* node (fig 3.21.).
In specifics, we have decided the following ranges (fig 3.22.):

- Age: $R \approx [0, 108]$
- Approximate Annual Income: $R \approx [0, 186740]$
- Consultation Duration: $R \approx [0, 133]$
- Consultation Price: $R \approx [0, 3636]$



(*fig 3.22.*, arbitrarily defined ranges)

As a result of this filtering, around 141 observations have been excluded from the dataset, which is approximately $\sim 1.41\%$ of the observations in the whole dataset. We can consider this as a good number of observations to filter.

```
Number Of Observations

Data
Role      Filtered    Excluded     DATA

TRAIN       9867         141       10008
```

(*fig 3.23.*, summary of the filter)

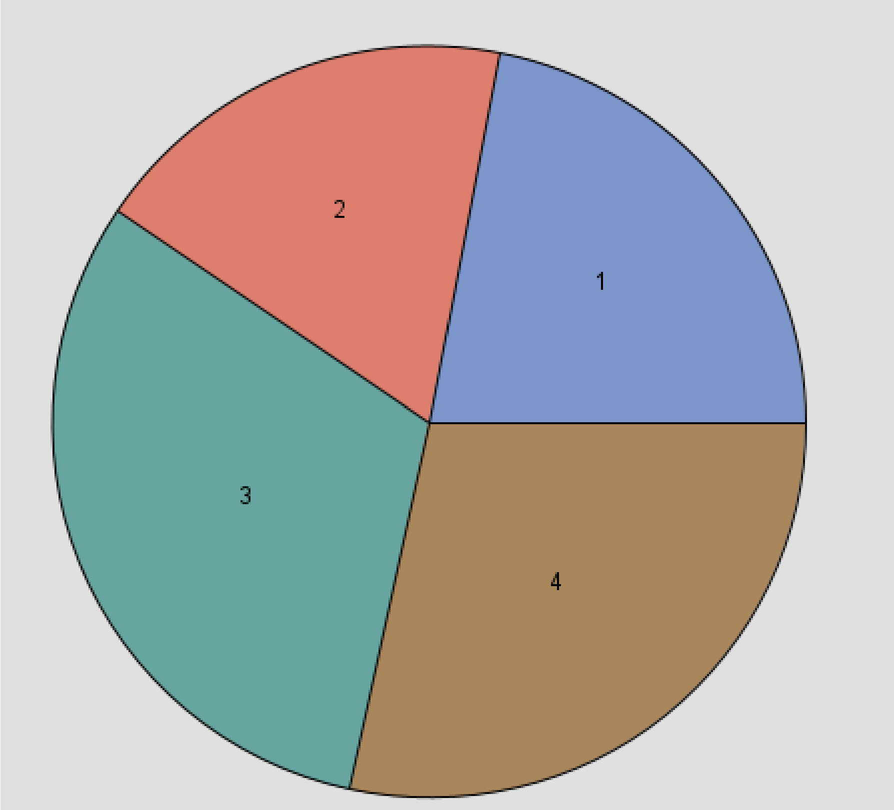
Multidimensional Outliers

Before we impute values, we still need to check for multidimensional outliers. To do it, we used the *"Cluster Node"* (fig 3.21.) which performs K-means clustering on the dataset. This can be effective in finding these multidimensional outliers, as K-means is sensitive to them. More precisely, this node does the following:

- Standardizes the numerical variables

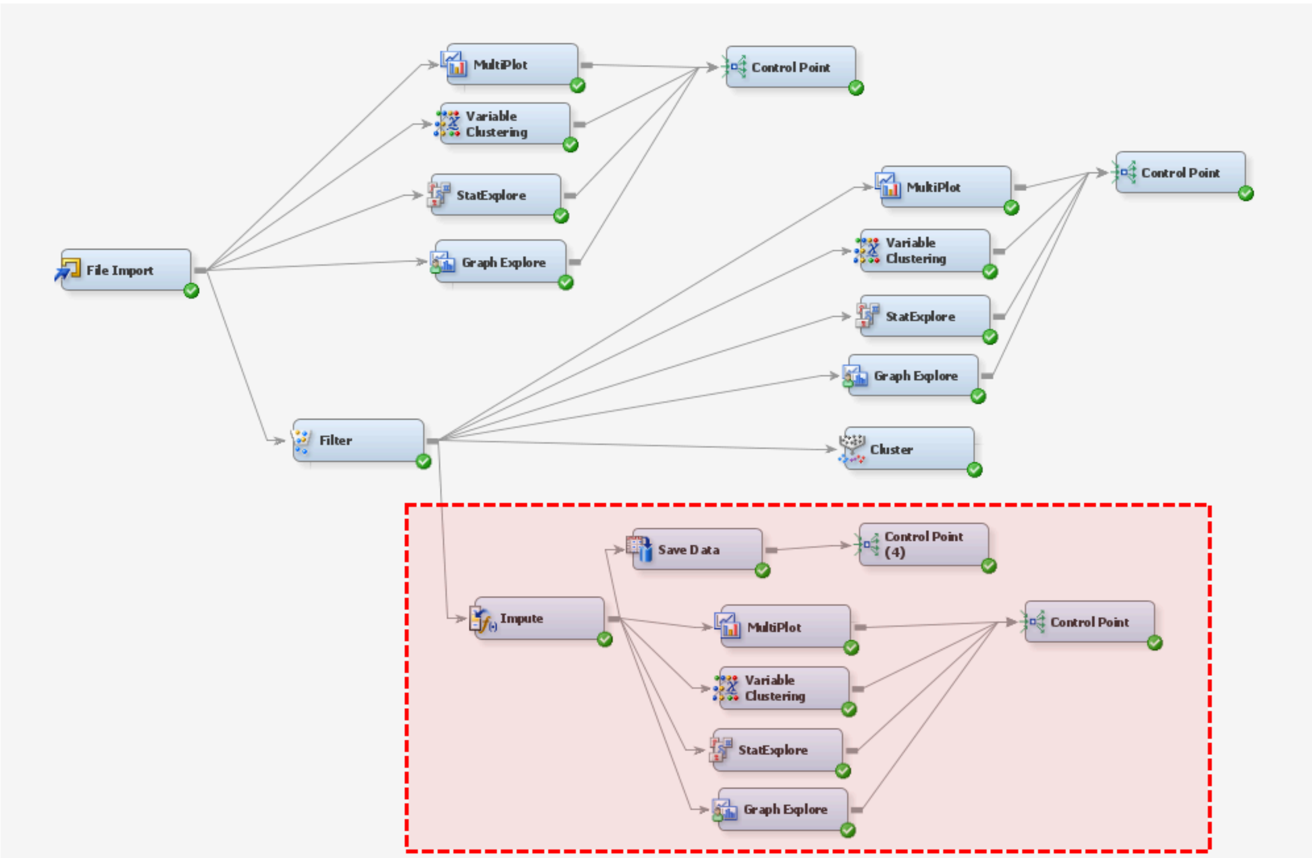
- Initializes the seed with Princomp method, reducing the number of necessary iterations for the clustering process
- Makes four clusters; so $K = 4$

  As a result, four almost equally-sized clusters were formed, meaning there are no multidimensional outliers detected (fig 3.24.).



(*fig 3.24.*, result of 4-means clustering)

Missing Values



(*fig 3.25.*, nodes used for missing values imputation)

Having made sure that our data is clean from outliers, we can proceed to deal with missing values.

We have to decided to impute the missing values through *decision trees*, which are able to impute both numerical and categorical variables. We have not used KNN to perform imputation, as it is unavailable in the SAS Miner Enterprise program.
To perform this imputation, we used the *"Impute"* node (fig 3.25.), setting the method to "Decision Tree". It is worth noting that the imputed variables have been renamed to IMP_<variable> .

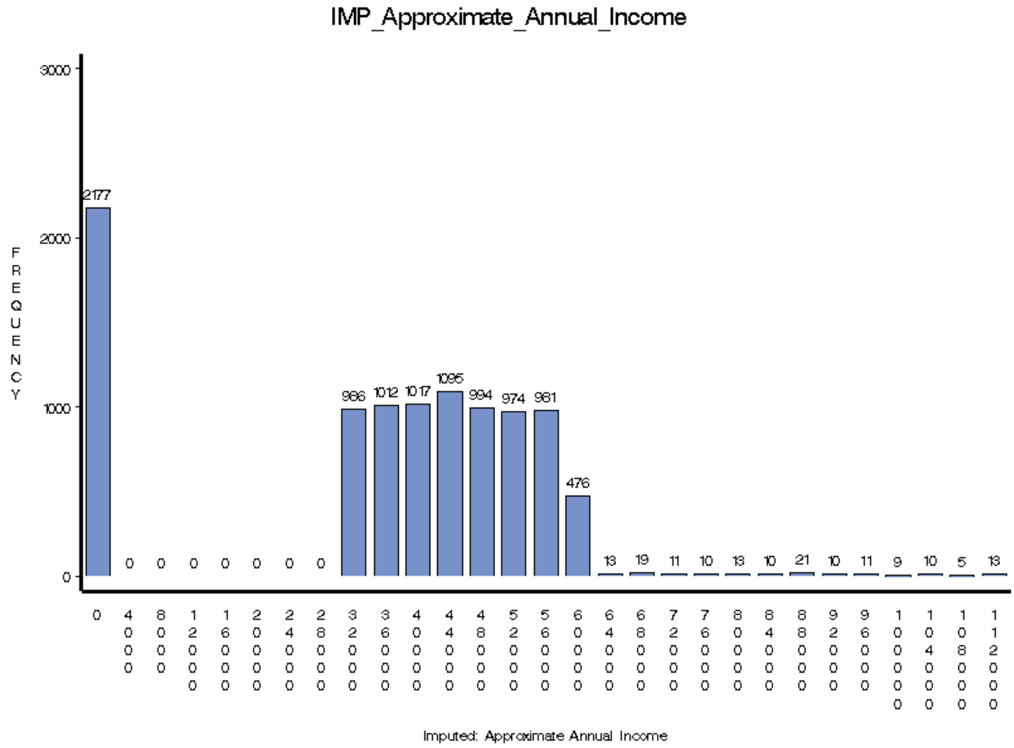
Variable Name	Impute Method	Imputed Variable	Impute Value	Role	Measurement Level	Label	Number of Missing for TRAIN
Age	TREE	IMP_Age	.	INPUT	INTERVAL	Age	55
Approximate_Annual_Income	TREE	IMP_Approximate_Annual_Income	.	INPUT	INTERVAL	Approximate Annual Income	153
Education_Level	TREE	IMP_Education_Level	.	INPUT	NOMINAL	Education Level	29
Insurance_Coverage	TREE	IMP_Insurance_Coverage	.	INPUT	INTERVAL	Insurance Coverage	50
Insurance_Provider	TREE	IMP_Insurance_Provider	.	INPUT	NOMINAL	Insurance Provider	104

(*fig 3.26.*, results of tree imputation)

Post-Cleaning Analysis

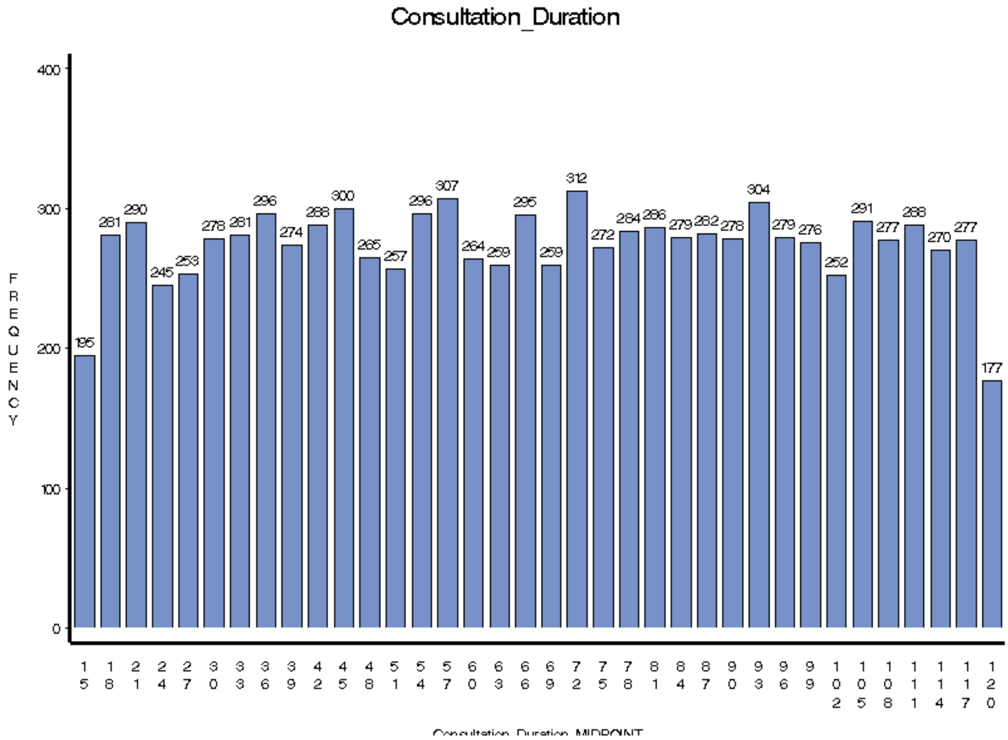
Having a clean dataset from outliers and missing values, we can check its statistics again. As remarked before, we will focus on the variables which were impossible to analyze due to extreme outliers - that is Approximate Annual Income, Consultation Duration and Consultation Price - and gain significant insights on the dataset.

- **Approximate Annual Income**: We can see an interesting fact: there is a neat separation between people with no income and people with income $>$ 32.000. This could tell us that some of the patients were people who had no income at all, such as children. Other than that, the variable seems to be uniformly distributed, with some low-frequent values on the high range (they will not be considered as outliers as they are not "too far" from the values).



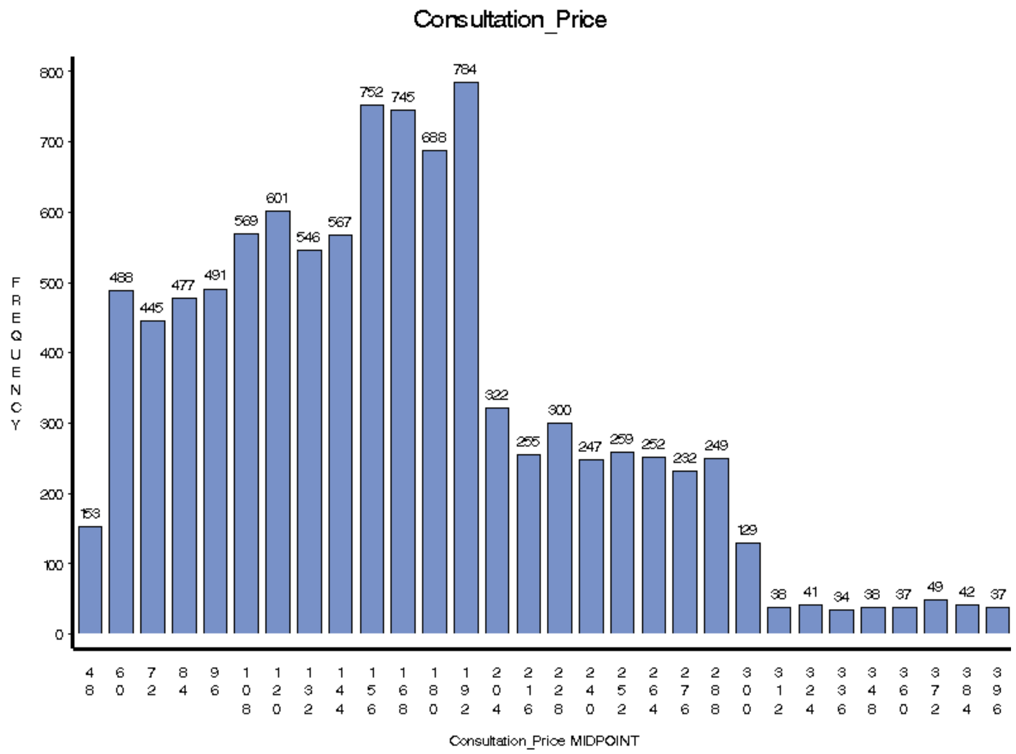
(*fig 3.27.*, cleaned)

- **Consultation Duration**: Without outliers, the consultation durations seem to be more or less uniformly distributed, except for "extreme values" (first and last bin) which have a lower frequency.



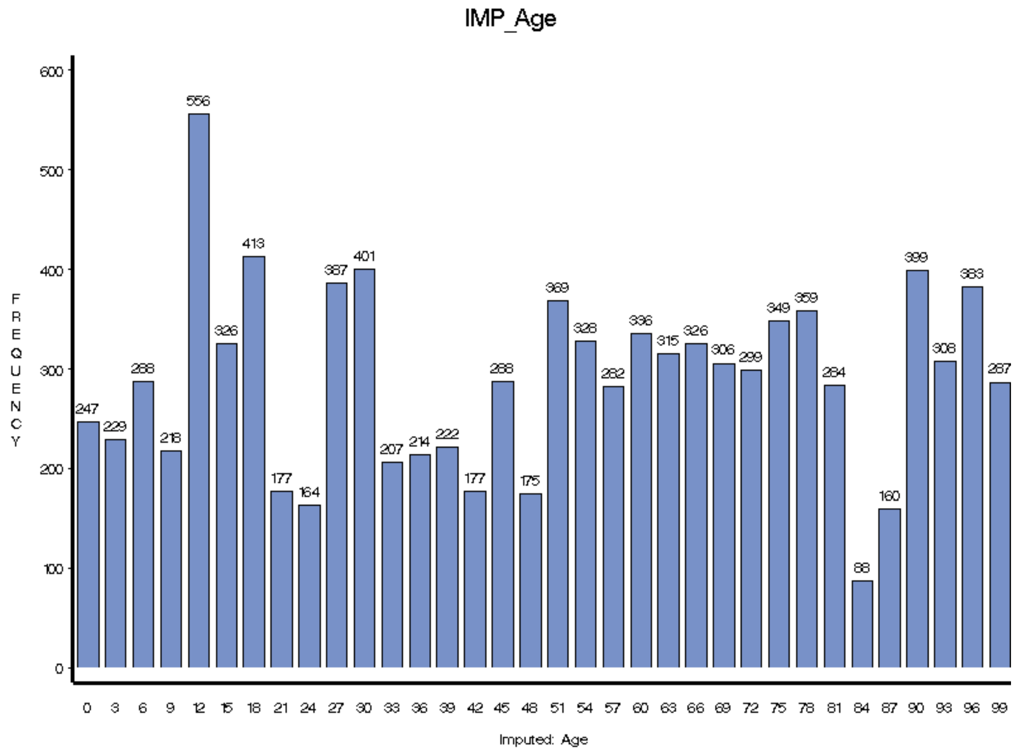
(*fig 3.28.*, cleaned)

- **Consultation Price**: The consultation prices seem to be distributed with a right-skew; this tells us that higher prices are rare (such as >312 pounds), whereas it's common to be charged around 150-200 pounds.



(*fig 3.29.*, cleaned)

- **Age**: Without outliers, we cannot define a precise distribution for age; however, we can say that there is a trend towards people of young age (in particular $\in [12, 15]$); this confirms our previous hypothesis as we analyzed the approximate annual income, where most patients were underaged children who cannot have an income.

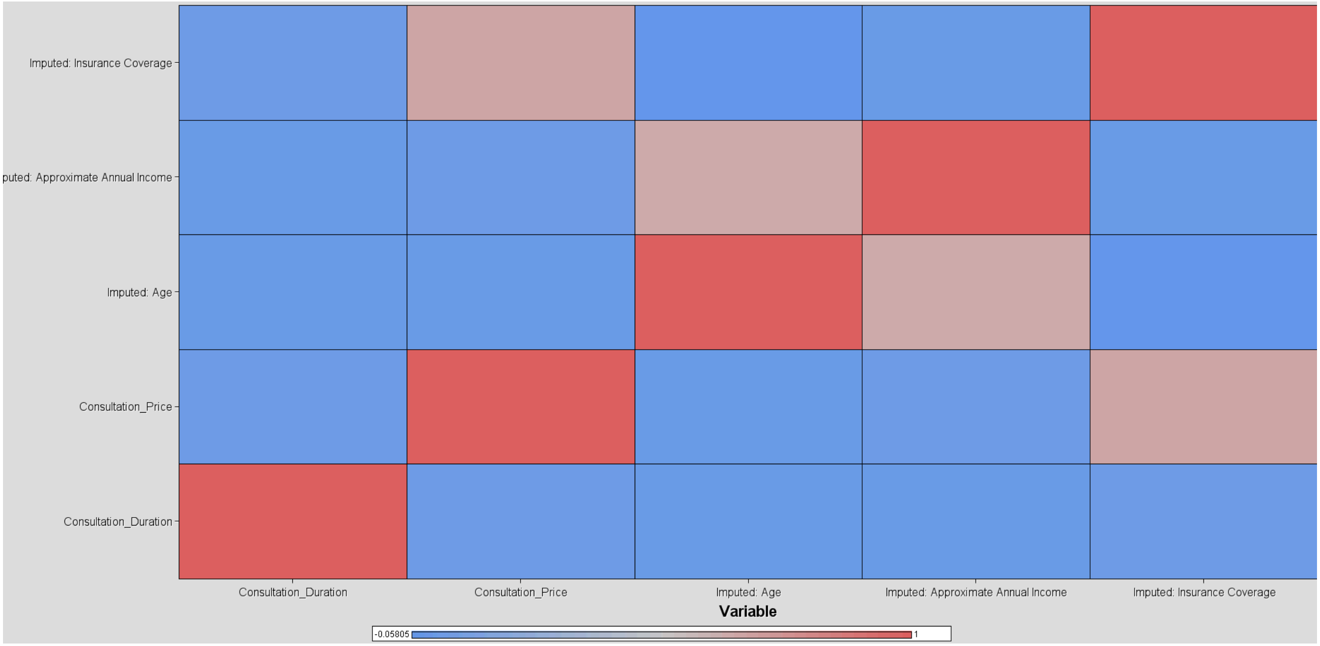


(*fig 3.30.*, cleaned)

Concerning the other variables, we can make the same conclusions as the ones we did previously (in *Phase 0*).

However, the situation becomes different if we check again the correlation between numerical variables. Here we obtain that there exist significant correlations. In fact, we can see that there is a significant amount of correlation between *Insurance Coverage* and *Consultation Price* (0.63), as well between *Approximate Annual Income* and *Age* (0.61) (fig 3.30.α). Although they're still inside the range $[-0.7, 0.7]$, we still have potential grounds to consider these variables to be correlated enough.

As the project guidelines instructed, we will not do anything about the correlation and simply make it known in the report.



(*fig 3.30.α.*, correlation of variables post-data cleaning)

Final note: Variables Transformation

As specified in the guidelines, we will not standardize numerical variables. Moreover, we will not transform categorical variables to numerical with o*fine-hot encoding (or dummy transformation), as this could cause an inflation in amount of variables.

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	City_of_Residence	INPUT	8	0	Birmingham	14.46	Belfast	14.28
TRAIN	Department	INPUT	13	0	Psychiatry	13.65	General Practice	13.30
TRAIN	Family_History	INPUT	5	0	Heart Disease	22.33	Hypertension	20.25
TRAIN	Gender	INPUT	3	0	Female	33.95	Other	33.81
TRAIN	IMP_Education_Level	INPUT	8	0	Undergraduate	41.83	Master	34.08
TRAIN	IMP_Insurance_Provider	INPUT	5	0	Provider D	21.54	Provider B	20.53
TRAIN	Marital_Status	INPUT	4	0	Divorced	28.65	Single	28.54
TRAIN	Profession	INPUT	10	0	Retired	36.25	Student	20.96
TRAIN	Satisfaction_Level	INPUT	6	0	2	18.89	4	18.86

(*fig 3.31.*, summary statistics of categorical variables)

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Consultation_Duration	INPUT	67.52133	30.47866	9867	0	15	68	120	-0.00403	-1.19091
Consultation_Price	INPUT	164.9322	71.24999	9867	0	50.03676	159.459	398.737	0.649235	0.182921
IMP_Age	INPUT	49.66622	29.67209	9867	0	0	52	100	0.00729	-1.23504
IMP_Approximate_Annual_Income	INPUT	35641.06	21094.31	9867	0	0	40979	113120	-0.56547	-0.20714
IMP_Insurance_Coverage	INPUT	115.0282	79.56521	9867	0	0	115.5196	421.8878	0.324597	-0.19115

(*fig 3.32.*, summary statistics of quantitative variables)

3.3. Phase 2: Data Inconsistencies Treatment

This phase of data treatment will make use of *SAS Guide* software, as this process may involve making some SQL queries.

Possible Inconsistencies

We thought out nine scenarios of data inconsistency, and they are:

1. Age should be above 0
2. Satisfaction level should be in the range $[1, 5]$
3. Legal marriage in the United Kingdom is 18; so anyone under 18 who presents marital status other than single is considered as an anomaly
4. School leaving age is defined to be 16 in the United Kingdom; therefore anyone $\leq$ 16-aged customers should be a student
5. Insurance coverage should be always smaller (or equal) than the consultation price
6. People without an insurance provider should not have insurance coverage at all
7. Some professions might require some degrees; in our case, we considered Engineers, Lawyers and Scientists to be at least undergraduates (or higher).
8. Students should not possess an income
9. Ages and education level should coincide; in particular, some education levels have an intrinsic "minimum age". We considered them as the following:

- You need to be at least 16 to have a high school diploma
- You need to be at least 21 to have a bachelor's degree
- You need to be at least 22 to have a master's degree; in United Kingdom master's degrees last one year
- You need to be at least 25 to have a PhD
  In all cases except the one about satisfaction level, rows will be deleted. For the exception, we will replace values: if a satisfaction level is < 0, then replace it with 0. If > 5, then replace it with 5.

Moreover, some variables should remain constant between patients, which are the following: Profession, Age, Gender, Family History, Insurance Provider, Marital Status and City of Residence should remain the same. To do this, we will use SQL queries and proceed on a case-by-case basis.
The reason we are checking this, is that the timespan of the dataset is around five months (fig 3.20.), and the previously mentioned variables tend not to vary in such a timespan.
As an end-result, this makes possible to built ABTs without any type of inconsistencies.

Results

The code to treat the first eight scenarios of data inconsistency was written in SAS code, and we filtered out inconsistent data in the following order:

1. Age

2. Satisfaction Level
3. Age and Marriage
4. Age and Profession=Student
5. Satisfaction Value
6. Age and Marital Status
7. Insurance Coverage and Consultation Cost
8. Insurance Provider and Insurance Coverage
9. Education Level and Profession
10. Profession=Student and Approximate Annual Income
11. Age and Education Level

As a result, we have the following sequence which represents the decrease in instance as we check for inconsistent rows:

$$9867 \xrightarrow{1.} 9724 \xrightarrow{2.} 9724 \xrightarrow{3.} 9719 \xrightarrow{4.} 9610 \xrightarrow{5.} 9599 \xrightarrow{6.} 9599$$
$$9599 \xrightarrow{7.} 9599 \xrightarrow{8.} 9599 \xrightarrow{9.} 9599 \xrightarrow{10.} 9599 \xrightarrow{11.} 9191$$

Therefore, from these series of controls we have deleted 676 rows, reducing the dataset to 93.15% of the original size.

```
/* Program to check for basic consistency in the transactional table, inconsistencies end up in deletion */

DATA CONSISTENT_TRANTABLE;
SET WORK.PREABT; /* File import */

/* Age has to be >0 */
IF (IMP_Age<0 OR IMP_Age=0) THEN DO;
	DELETE;
END;
/* 9867 -> 9724 */

/* Satisfaction value must be in [1, 5]. If <0, set to 0; If >5, set to 5.*/
IF (Satisfaction_Level < 1) THEN DO;
	Satisfaction_Level=1;
END;

IF (Satisfaction_Level > 5) THEN DO;
	Satisfaction_Level=5;
END;

/* Legal age for marriage in UK is 18, so any rows not respecting this is considered as an inncosistency */
IF (IMP_Age<18 AND NOT(Marital_Status='Single')) THEN DO;
	DELETE;
END;
/* 9724 -> 9719 */

/* School leaving age is legally defined to be 16, therefore anyone with age <=16 must be a student */
```

```
IF (IMP_Age<17 AND NOT(Profession='Student')) THEN DO;
      DELETE;
END;
/* 9719 -> 9610 */

/* Insurance coverage should be always smaller than consultation cost */
IF (Consultation_Price < IMP_Insurance_Coverage) THEN DO;
      DELETE;
END;
/* 9610 -> 9599*/

/* People without insurance should not have insurance coverage */
IF (IMP_INSURANCE_COVERAGE > 0 AND IMP_Insurance_Provider='None') THEN DO;
      DELETE;
END;
/* 9599 -> 9599 /*

/* Check professions according to their degree required
      Lawyer, Engineer, Scientist -> At least high school
      Others won't be checked as some of them might have more specific requirements
*/
IF (
      (PROFESSION='ENGINEER' OR PROFESSION='Lawyer' or PROFESSION='Scientist') AND
      NOT(IMP_Education_Level='PhD' or IMP_Education_Level='Master' or
      IMP_Education_Level='Undergraduate' or IMP_Education_Level='High school')
) THEN DO;
      DELETE;
END;
/* 9599 -> 9599 */

/* Students should not have an income (we will not count cases of part-time jobs or irregular work) */
IF (PROFESSION='Student' AND IMP_Approximate_Annual_Income > 0) THEN DO;
      DELETE;
END;
/* 9599 -> 9599 */

/* Compare age with education level; exclusing exceptional cases of people who skipped grades, it should
be that
      High School: must be at least 16, compulsory education ends at that age
      Undergraduate: must be at least 21 (three years to complete a BsC degree)
      Master's: must be at least 22 (in UK master's last one year)
      PhD: 25 (3 years)
      The rest won't be checked as the cases can vary
*/
IF ( (IMP_EDUCATION_LEVEL='High school' AND IMP_AGE < 16 ) OR
       (IMP_EDUCATION_LEVEL='Undergraduate' AND IMP_AGE < 21) OR
       (IMP_EDUCATION_LEVEL='Master' AND IMP_AGE < 22 ) OR
       (IMP_EDUCATION_LEVEL='PhD' AND IMP_AGE < 25 )
) THEN DO;
      DELETE;
```

```
END;
/* 9599 -> 9191 */

/*
    RESULTS
    -------
    10 Queries
    9867 -> 9191 rows
    676 deleted rows
*/
```

(*snippet 3.1.*, SAS code for checking and treating data consistency)

Concerning the controls about the "constant" variables between patients' IDs, we have the following result:

- **Profession:** Two patients had inconsistencies in profession: they are the ones with ID 1488 and 1496. Looking at their age, it is clear that their profession should be corrected to "Student"; it might be that there were visits where his profession was erroneously classified as "Retired". We will manually correct them to be defined as "Student" in another SAS script.
- **Age**: There were a lot of inconsistencies in age, mainly due to tree-imputation. As the values are "close to each other", we can consider doing nothing about them and taking the mean while building the ABT.
- **Gender**: There were five patients with inconsistent genders: 1050, 1307, 1349, 1447, 1490. The fact the difference in genders do not follow a timeline (meaning that from a certain date they switched genders) suggests that this is due to a registration error, rather than gender transitioning. Therefore, their genders will be replaced by the mode of each patient's gender.
- **Family History**: No inconsistencies detected
- **City of Residence**: No inconsistencies detected
- **Marital Status**: The following patients had inconsistent marital status: 1140, 1322, 1332, 1382. By analyzing their marital statuses row-by-row, we have found out that each patient with inconsistent marital status had only one row with inconsistent information. Therefore, we ruled this to be due to registration error; so the inconsistencies will be replaced with the correct value.
- **Insurance Provider**: Interestingly enough, there are a good amount of patients with different insurance providers for each visit. There are 31 patients with different insurance providers, and they make up 612 rows of the dataset (so around 6.20% of the total). It is possible to separate them into another date for special analysis, as they make up a significant amount of data. For our ABT, we will filter these rows out. In other words, we will only keep patients who kept only one insurance provider. (***Note: to ask for validation to professor, just to see if our reasoning makes sense***)

All of this has been done with SAS code (see *snippet 3.2., 3.3.*)

Patient_ID	Profession	Profession	Visit_Date	IMP_Age
1488	Retired	Student	15APR2024:00:00:00	11
1488	Retired	Student	08MAY2024:00:00:00	11
1488	Student	Retired	02JAN2024:00:00:00	11
1488	Student	Retired	02FEB2024:00:00:00	11
1488	Student	Retired	11FEB2024:00:00:00	11
1488	Student	Retired	23FEB2024:00:00:00	11
1488	Student	Retired	26FEB2024:00:00:00	11
1488	Student	Retired	13MAR2024:00:00:00	11
1488	Student	Retired	15MAR2024:00:00:00	11
1488	Student	Retired	19MAR2024:00:00:00	11
1488	Student	Retired	09APR2024:00:00:00	11
1488	Student	Retired	12APR2024:00:00:00	11
1488	Student	Retired	06MAY2024:00:00:00	11
1488	Student	Retired	09MAY2024:00:00:00	11
1488	Student	Retired	10MAY2024:00:00:00	11
1488	Student	Retired	17MAY2024:00:00:00	11
1488	Student	Retired	30MAY2024:00:00:00	11
1488	Student	Retired	10JUN2024:00:00:00	11
1496	Retired	Student	05JAN2024:00:00:00	4
1496	Retired	Student	25JAN2024:00:00:00	4
1496	Retired	Student	21FEB2024:00:00:00	4
1496	Retired	Student	11MAR2024:00:00:00	4
1496	Retired	Student	22MAR2024:00:00:00	4
1496	Retired	Student	28MAR2024:00:00:00	4
1496	Retired	Student	15APR2024:00:00:00	4
1496	Retired	Student	22APR2024:00:00:00	4

(*figure 3.33*, customers with inconsistent profession)

Patient_ID	Gender	Visit_Date
1050	Female	08JAN2024:00:00:00
1050	Female	15JAN2024:00:00:00
1050	Female	13FEB2024:00:00:00
1050	Female	23FEB2024:00:00:00
1050	Female	29FEB2024:00:00:00
1050	Female	09MAR2024:00:00:00
1050	Female	27MAR2024:00:00:00
1050	Female	29MAR2024:00:00:00
1050	Female	23APR2024:00:00:00
1050	Female	03MAY2024:00:00:00
1050	Female	10JUN2024:00:00:00
1050	Female	13JUN2024:00:00:00
1050	Female	14JUN2024:00:00:00
1050	Female	15JUN2024:00:00:00
1050	Male	15JAN2024:00:00:00
1050	Male	28JAN2024:00:00:00
1050	Male	13MAR2024:00:00:00

(*figure 3.34.*, example of a customer with inconsistent gender)

Patient_ID	Marital_Status	Marital_Status	Visit_Date
1140	Single	Widowed	10JAN2024:00:00:00
1140	Single	Widowed	29JAN2024:00:00:00
1140	Single	Widowed	30JAN2024:00:00:00
1140	Single	Widowed	04FEB2024:00:00:00
1140	Single	Widowed	10FEB2024:00:00:00
1140	Single	Widowed	19FEB2024:00:00:00
1140	Single	Widowed	18MAR2024:00:00:00
1140	Single	Widowed	03APR2024:00:00:00
1140	Single	Widowed	06APR2024:00:00:00
1140	Single	Widowed	11APR2024:00:00:00
1140	Single	Widowed	17APR2024:00:00:00
1140	Single	Widowed	29APR2024:00:00:00
1140	Single	Widowed	20MAY2024:00:00:00
1140	Single	Widowed	07JUN2024:00:00:00
1140	Single	Widowed	18JUN2024:00:00:00
1140	Single	Widowed	29JUN2024:00:00:00
1140	Widowed	Single	22FEB2024:00:00:00
1322	Married	Single	25JAN2024:00:00:00
1322	Married	Single	23JUN2024:00:00:00
1322	Single	Married	05JAN2024:00:00:00
1322	Single	Married	12JAN2024:00:00:00
1322	Single	Married	13JAN2024:00:00:00
1322	Single	Married	19JAN2024:00:00:00
1322	Single	Married	22JAN2024:00:00:00
1322	Single	Married	19FEB2024:00:00:00
1322	Single	Married	28FEB2024:00:00:00

(*figure 3.33*, example of a customer with inconsistent marital status)

```
PROC SQL;
    SELECT DISTINCT T1.PATIENT_ID, T1.PROFESSION, T2.PROFESSION
    FROM WORK.PREABT T1 LEFT JOIN WORK.PREABT T2
        ON T1.PATIENT_ID = T2.PATIENT_ID
    WHERE
        T1.PROFESSION <> T2.PROFESSION;
RUN;

PROC SQL;
    SELECT DISTINCT T1.PATIENT_ID, T1.IMP_AGE, T2.IMP_AGE
    FROM WORK.PREABT T1 LEFT JOIN WORK.PREABT T2
        ON T1.PATIENT_ID = T2.PATIENT_ID
    WHERE
        T1.IMP_AGE <> T2.IMP_AGE;
RUN;

PROC SQL;
    SELECT DISTINCT T1.PATIENT_ID, T1.GENDER, T2.GENDER
    FROM WORK.PREABT T1 LEFT JOIN WORK.PREABT T2
        ON T1.PATIENT_ID = T2.PATIENT_ID
    WHERE
        T1.GENDER <> T2.GENDER;
RUN;

PROC SQL;
    SELECT DISTINCT T1.PATIENT_ID
    FROM WORK.PREABT T1 LEFT JOIN WORK.PREABT T2
        ON T1.PATIENT_ID = T2.PATIENT_ID
    WHERE
        T1.IMP_INSURANCE_PROVIDER <> T2.IMP_INSURANCE_PROVIDER;
```

```
RUN;

PROC SQL;
    SELECT DISTINCT T1.PATIENT_ID, T1.FAMILY_HISTORY, T2.FAMILY_HISTORY
    FROM WORK.PREABT T1 LEFT JOIN WORK.PREABT T2
        ON T1.PATIENT_ID = T2.PATIENT_ID
    WHERE
        T1.FAMILY_HISTORY <> T2.FAMILY_HISTORY;
RUN;

PROC SQL;
    SELECT DISTINCT T1.PATIENT_ID, T1.MARITAL_STATUS, T2.MARITAL_STATUS
    FROM WORK.PREABT T1 LEFT JOIN WORK.PREABT T2
        ON T1.PATIENT_ID = T2.PATIENT_ID
    WHERE
        T1.MARITAL_STATUS <> T2.MARITAL_STATUS;
RUN;

PROC SQL;
    SELECT DISTINCT T1.PATIENT_ID, T1.CITY_OF_RESIDENCE, T2.CITY_OF_RESIDENCE
    FROM WORK.PREABT T1 LEFT JOIN WORK.PREABT T2
        ON T1.PATIENT_ID = T2.PATIENT_ID
    WHERE
        T1.CITY_OF_RESIDENCE <> T2.CITY_OF_RESIDENCE;
RUN;
```

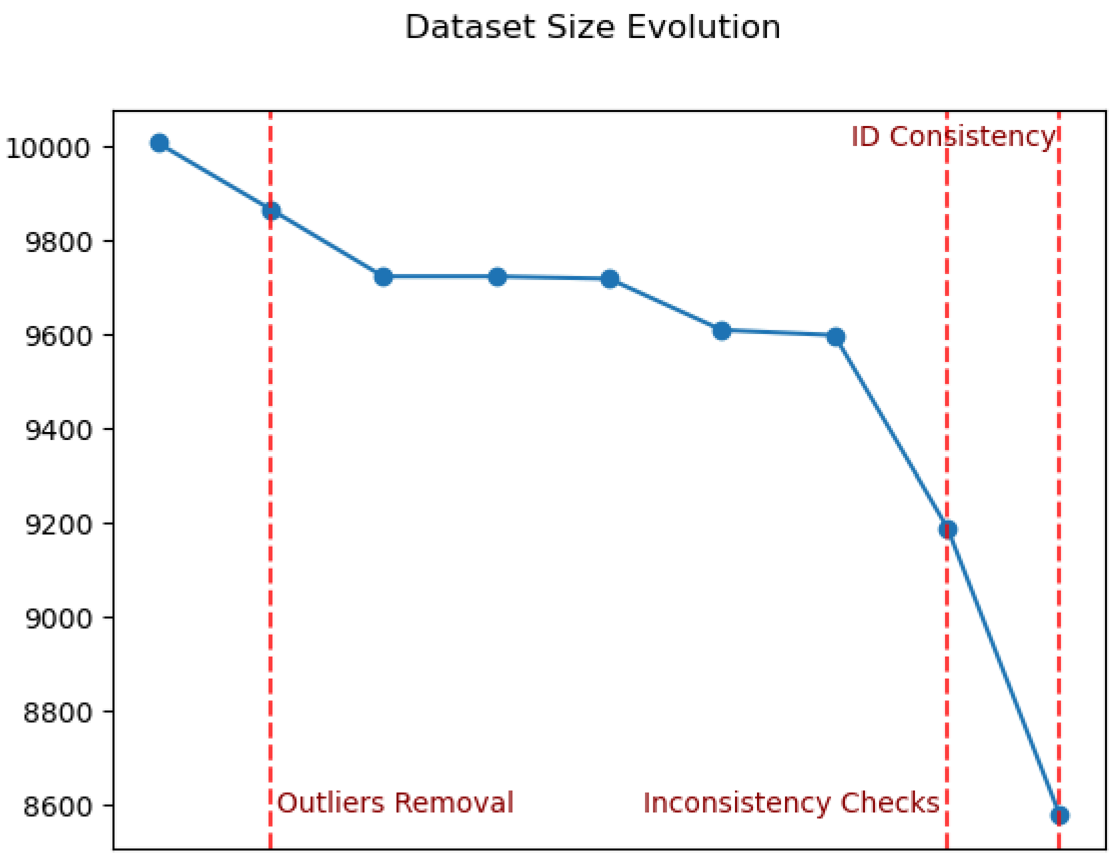
(*snippet 3.2.*, code for checking inconsistencies between IDs)

```
IF ( PATIENT_ID=1488 AND NOT(PROFESSION='Student')) THEN DO;
        PROFESSION='Student';
END;

IF ( PATIENT_ID=1496 AND NOT(PROFESSION='Student')) THEN DO;
  PROFESSION='Student';
END;

IF ( PATIENT_ID=1050 AND NOT(GENDER='Female')) THEN DO;
  GENDER='Female';
END;

IF ( PATIENT_ID=1307 AND NOT(GENDER='Male')) THEN DO;
  GENDER='Male';
END;

IF ( PATIENT_ID=1349 AND NOT(GENDER='Male')) THEN DO;
  GENDER='Male';
END;

IF ( PATIENT_ID=1447 AND NOT(GENDER='Female')) THEN DO;
```

```
   GENDER='Female';
END;

IF ( PATIENT_ID=1490 AND NOT(GENDER='Female')) THEN DO;
   GENDER='Female';
END;

IF ( PATIENT_ID=1140 AND NOT(MARITAL_STATUS='Single')) THEN DO;
   MARITAL_STATUS='Single';
END;

IF ( PATIENT_ID=1322 AND NOT(MARITAL_STATUS='Single')) THEN DO;
   MARITAL_STATUS='Single';
END;

IF ( PATIENT_ID=1332 AND NOT(MARITAL_STATUS='Single')) THEN DO;
   MARITAL_STATUS='Single';
END;

IF ( PATIENT_ID=1382 AND NOT(MARITAL_STATUS='Single')) THEN DO;
   MARITAL_STATUS='Single';
END;

IF (
      PATIENT_ID in(
1013,
1014,
1015,
1028,
1031,
1034,
1089,
1092,
1100,
1105,
1135,
1143,
1234,
1245,
1248,
1260,
1261,
1266,
1285,
1294,
1302,
1308,
1317,
1340,
1343,
```

```
1381,
1449,
1455,
1485,
1490,
1498
)
)
THEN DO;
    DELETE;
END;
```

(*snippet 3.3.*, SAS code for manually correcting inconsistent rows)



(*figure 3.34.*, evolution of dataset size)

X

4. ABT Construction

The modified dataset obtained remains a *transactional table*, meaning we still have no insights about the *customers itself*. To obtain a source of data where we can glean insights about customers, we'll have to transform the transactional table into an analytic-base table.

To do this, we will use the following methods:

Pivoting: We can directly transpose some variables to each customer, which we assumed to be unique. They are namely gender, profession, marital status, city of residence, family history and insurance provider.

Aggregation: We can get frequency, recency, membership and monetary of the customer.

- *Frequency* is the total amount of transactions linked to a patient
- *Recency* is the amount of days since the last visit
- *Membership* is the amount of days since the first visit
- *Monetary* is the total sum of consultation price

Summarization: We can get the following averages:

- *Average Approximate Annual Income*
- *Average Age*: there were some mismatch in age, due to imputations. As previously established, we can do this as the values are "near" enough.
- *Average Satisfaction Level*
- *Average Consultation Duration*

Proportions. We can get the following proportion:

- *Total insurance coverage* respect to *total charged amount* for all visits of a patient

Segmentation of Departments: We can segment each department visit to get the following information:

- Amount of consultations done, relative to the frequency of a patient
- Proportion of prices, relative to monetary of a patient

```
PROC SQL;
CREATE TABLE BIO_INFO AS
    SELECT DISTINCT PATIENT_ID, GENDER, PROFESSION, FAMILY_HISTORY, CITY_OF_RESIDENCE,
MARITAL_STATUS, IMP_INSURANCE_PROVIDER
    FROM WORK.PREABTCONSISTENT /* IMPORTANT !!! */
    GROUP BY PATIENT_ID;
RUN;
/* ^^ Directly transposes some biographical/anagraphical information ^^ */
    /* such as gender, profession, family history, which are supposed to be unique. */

/* ========================================================== */
CREATE TABLE AGE AS
    SELECT DISTINCT PATIENT_ID, avg(IMP_Age) as Age
    FROM WORK.PREABTCONSISTENT
    GROUP BY PATIENT_ID;
RUN;
/* As there are inconsitencies in the imputed ages, we will simply take their average */
/* ========================================================== */
```

```
PROC SQL;
CREATE TABLE STEP1 AS
    SELECT X.PATIENT_ID, X.DEPARTMENT, (sum(X.Consultation_Price)/T.MON) as TotAmt
    FROM WORK.PREABTCONSISTENT as X, (
        SELECT PATIENT_ID, sum(AUX.CONSULTATION_PRICE) as MON
        FROM WORK.PREABTCONSISTENT AS AUX
        GROUP BY AUX.PATIENT_ID) as T
    WHERE T.PATIENT_ID = X.PATIENT_ID
    GROUP BY X.PATIENT_ID, X.DEPARTMENT;
RUN;

PROC SORT DATA=STEP1 OUT=STEP2;
    BY PATIENT_ID;
RUN;

PROC TRANSPOSE DATA=STEP2 OUT=SEGMENTED_PRICE
    PREFIX=proportion_price_;
    ID DEPARTMENT;
    BY PATIENT_ID;
RUN;

/* ^^ Segments total consultation price by department in form of proportion ^^ */

/* ========================================================= */
PROC SQL;
CREATE TABLE STEP1 AS
    SELECT PATIENT_ID, DEPARTMENT, count(*) as Freq
    FROM WORK.PREABTCONSISTENT
    GROUP BY PATIENT_ID, DEPARTMENT;
RUN;

PROC SORT DATA=STEP1 OUT=STEP2;
    BY PATIENT_ID;
RUN;

PROC TRANSPOSE DATA=STEP2 OUT=SEGMENTED_FREQ
    PREFIX=freq_;
    ID DEPARTMENT;
    BY PATIENT_ID;
RUN;
/* ^^ same as above but with frequency */

/* ========================================================= */
proc sql;
CREATE TABLE TIME_DATA AS
    select distinct PATIENT_ID, (DATETIME()-min(Visit_Date))/86400 as membership_days
    from WORK.PREABTCONSISTENT
    group by PATIENT_ID;
run;
```

```
data TIME_DATA_FORMATTED;
set TIME_DATA;
format first_visit DTDATE.;
FORMAT
run;
/* Get date of first visit */

/* ========================================================= */
proc sql;
CREATE TABLE RECENCY AS
	select distinct PATIENT_ID, (DATETIME()-max(Visit_Date))/86400 as recency_days
	from WORK.PREABTCONSISTENT
	group by PATIENT_ID;
run;
/* Get recency */

/* ========================================================= */
PROC SQL;
CREATE TABLE AGGREGATED_INFO AS
	SELECT PATIENT_ID,
		avg(Consultation_Duration) as avg_duration,
		avg(Satisfaction_Level) as avg_satisfaction_level,
		sum(Consultation_Price) as monetary,
		avg(IMP_Approximate_Annual_Income) as avg_recorded_income,
		count(*) as total_frequency
	FROM WORK.PREABTCONSISTENT
	GROUP BY PATIENT_ID;
RUN;

/* Get important aggregated variables*/
	/* Namely: -total amount of money spent; -mode of department; -satisfaction, duration, ANI avg. */

/* ========================================================= */
PROC SQL;
CREATE TABLE PROPORTION_COVERAGE AS
	SELECT DISTINCT X.PATIENT_ID, sum(X.IMP_Insurance_Coverage)/T.MON as CoverageProportion
	FROM WORK.PREABTCONSISTENT as X, (
		SELECT PATIENT_ID, sum(AUX.CONSULTATION_PRICE) as MON
		FROM WORK.PREABTCONSISTENT AS AUX
		GROUP BY AUX.PATIENT_ID) as T
	WHERE T.PATIENT_ID = X.PATIENT_ID
	GROUP BY X.PATIENT_ID
;
RUN;




/* ========================================================= */
```

```
DATA PRE_FINAL;
	MERGE BIO_INFO AGE PROPORTION_COVERAGE SEGMENTED_PRICE SEGMENTED_FREQ
TIME_DATA_FORMATTED RECENCY AGGREGATED_INFO;
	BY PATIENT_ID;
RUN;

DATA PRE_PRE_FINAL;
	SET PRE_FINAL;
	DROP _NAME_;
RUN;

DATA FINAL_ABT;
	SET PRE_PRE_FINAL;
	ARRAY change _numeric_;
		DO OVER change;
		IF change=. THEN change=0;
	END;
RUN;
```

(*snippet 4.1.*, code for creating the finalized ABT table)

X

5. Data Analysis with PowerBI

TBD

X

6. Conclusion

Some yapping metrics about the company results blablabla

X

7. Appendix

IDEA: Might be a smart idea to move codes here, so they don't obstruct the document