NOVA IMS Information Management School

MEGA MARKET ANALYSIS

Analytic-based Table

Laura Matias 20221836
Marta Almendra 20221878
Matilde Casimiro 20221940
Teresa Simão 20221873

Data Preprocessing and Visualization

Table of Contents

Introduction

In the dynamic retail industry, prioritizing customer service and satisfaction is extremely vital if we want to gain a competitive advantage. Mega Market, a versatile retailer with an extensive product range, acknowledges the essencial role of comprehending and analyzing customer behavior as a key driver for success in the business.

This project is divided into three phases: data preprocessing, which includes data exploration, checking for outliers, missing value treatment, and verifying for coherence. Then ,building an ABT by generating additional variables; and lastly, creating meaningful and interactive dashboards in Power BI.

To achieve this goal, we will address the initial phase using the SAS Enterprise Miner platform. Within SAS Studio, we will assess coherence, convert data, and create derived variables. Additionally, we will also use SAS Studio to generate the customer-signature table, a form of Analytical-Based Table (ABT) designed for precise customer segmentation. Lastly, we will operate in Power BI to craft visualizations that will give us insights into customer behavior, product trends, and sales patterns.

Objective

Mega Market has assembled a dedicated team of data scientists, with a focus group working on data preprocessing (DP Team), whose goal is to harness the wealth of transactional data available in the company's information system to obtain comprehensive customer profiles, extract meaningful insights, and ultimately help the company grow and thrive.

The company hopes to enhance its decision-making processes, increase customer satisfaction, and improve overall business performance by taking a more educated approach.

Methodology

In order accomplish our objective, as mentioned before, the project was executed in various phases :

- **Data Preprocessing**

The initial step included diverse preprocessing techniques for raw data, involving preliminary visualizations, removal of evident outliers, and crucially, addressing discrepancies in the data.

Phase 1: First, we examined how each variable behaves in the dataset, and the inferences we may draw from it. This was executed using MultiPlot, Variable Clustering, StatExplore and GraphExplore nodes in SAS Miner.

Phase 2: The dataset was then filtered to detect outliers, and missing values were replaced. This was executed using the Filter and Impute node in SAS Miner.

Phase 3: We opted to alter some existing and add new variables to our model that we thought were relevant. This was executed using the Transform Variables node in SAS Miner.

After these phases are completed, the final version of the transactional table will be ready for use, and this version will be used in the Data Visualization stage.

- **Create the Analytic-Based Table**

In this point, the team will produce the Analytical Base Table (ABT). This was accomplished by adding up and structuring the data in the final transactional table, merging all of the existing pieces of information about the company's customers into a single table. This will be done with the SAS Studio software, which effectively employs SQL code to generate the final ABT.

- **Data Visualizations**

In this final stage, the team decided to use PowerBI to develop several graphics. This step enabled us to gain more insights and study in detail our dataset. This will help the company's employees to draw conclusions about their consumers and transactions more accuratly than before.

Data Preprocessing

• Initial Visualizations and Observations

As previously stated, the first round of preprocessing consisted of creating some early visualizations of the dataset as well as excluding some data points that were outliers to the overall pattern of the dataset. These processes were carried out with the SAS Enterprise Miner application, which generated the diagram shown below.

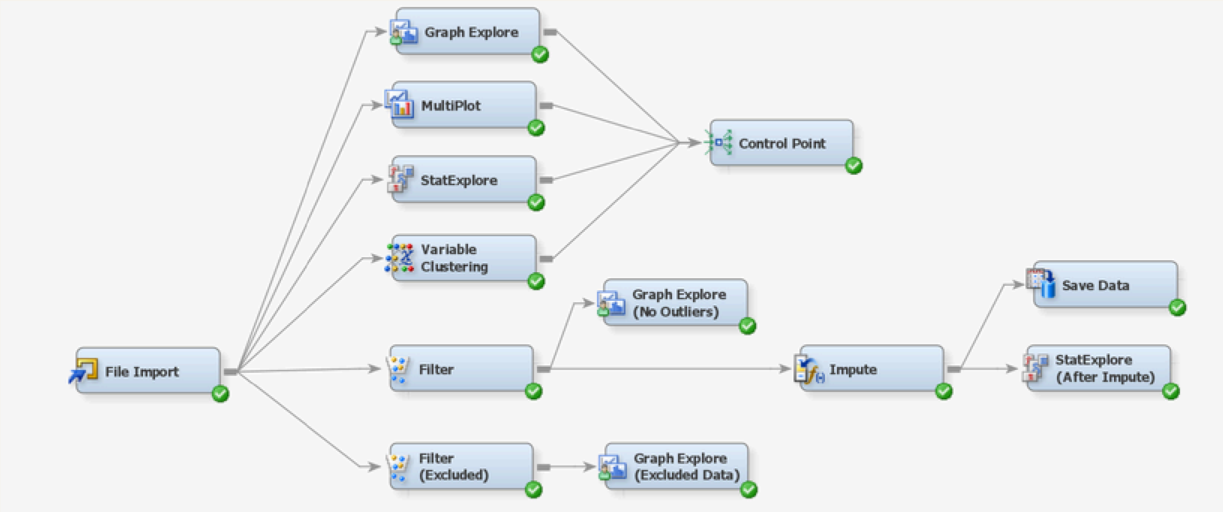
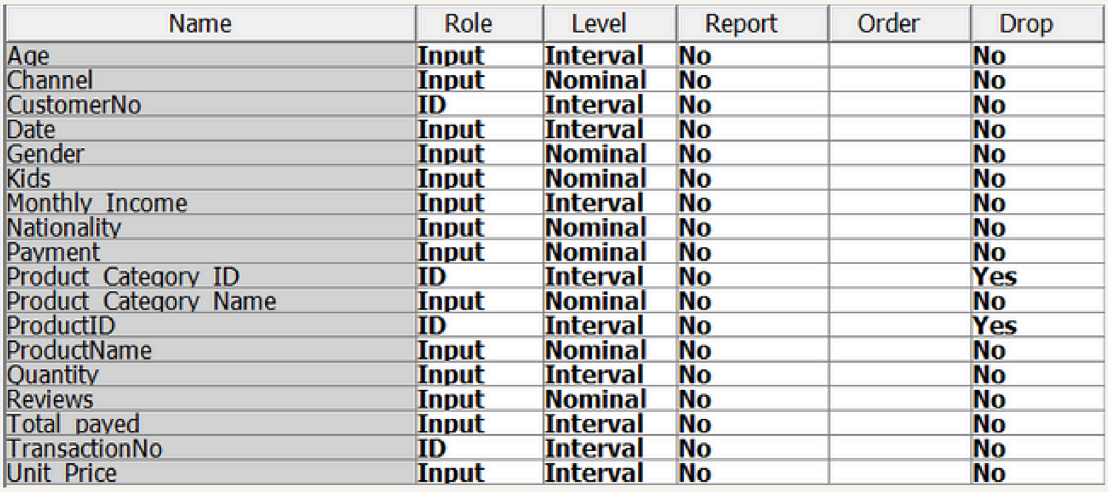


Fig. 1 - SAS Miner Diagram

The first node used had the basic (but critical) task of importing the dataset given by the company, Mega Market. Furthermore, we proceeded to update the automatic definitions of the variables and their roles, excluding the variables 'ProductID' and 'Product_Category_ID' from the dataset and future operations, since the information they offer is equivalent to that of other variables. After this modifications, this were the characteristics that we worked with:



Name	Role	Level	Report	Order	Drop
Age	Input	Interval	No		No
Channel	Input	Nominal	No		No
CustomerNo	ID	Interval	No		No
Date	Input	Interval	No		No
Gender	Input	Nominal	No		No
Kids	Input	Nominal	No		No
Monthly_Income	Input	Interval	No		No
Nationality	Input	Nominal	No		No
Payment	Input	Nominal	No		No
Product_Category_ID	ID	Interval	No		Yes
Product_Category_Name	Input	Nominal	No		No
ProductID	ID	Interval	No		Yes
ProductName	Input	Nominal	No		No
Quantity	Input	Interval	No		No
Reviews	Input	Nominal	No		No
Total_payed	Input	Interval	No		No
TransactionNo	ID	Interval	No		No
Unit_Price	Input	Interval	No		No

Fig. 2 - Variables' Level and Role Definition

Through the Multiplot node, we obtained the following graphics , that represent the distributions of some of the variables in the dataset, mainly those from which we could draw some useful conclusions. Here are some relevant comments about those variables

Fig.1 Age distribution

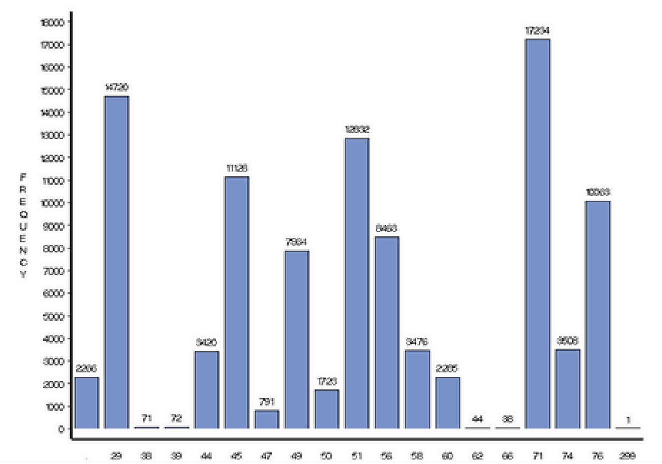
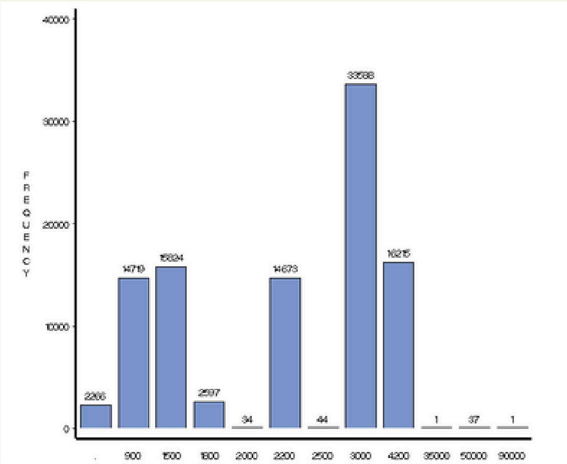


Fig. 2 Monthly Income distribution



- The variable **Age** has a noticeable outlier;
- The variable **Monthly_Income** contains 39 outliers

Fig.3 Gender distribution

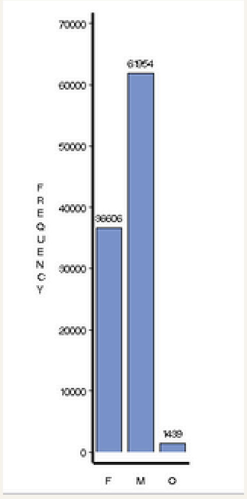


Fig. 4 Payment distribution

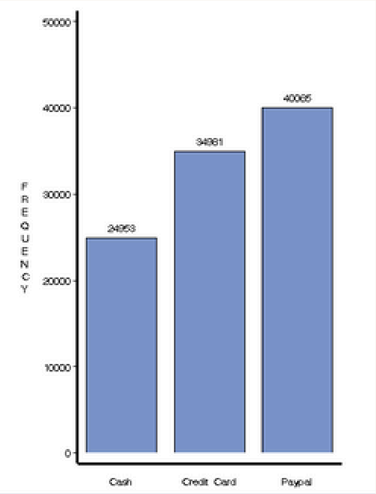
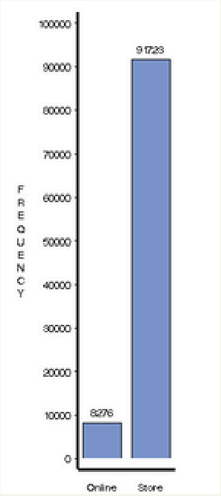


Fig. 5 Channel distribution



- As we can see, from the variable **Gender** Mega Market has more male costumers than female costumers;
- When it comes to the variabel **Payment_type**, the majority of clients favor PayPal;
- From the distribution of the variable **Channel** we can see that the majority of sales take place in store rather than online.

Fig. 6 Reviews distribution

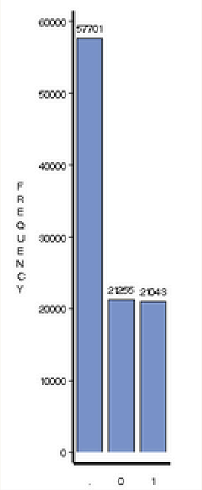


Fig. 7 Quantity distribution

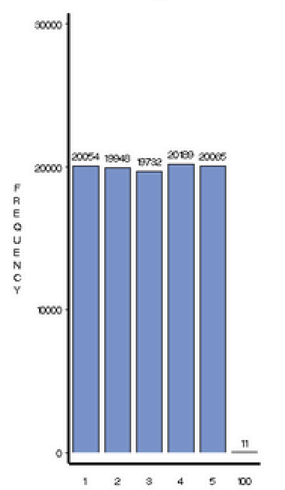
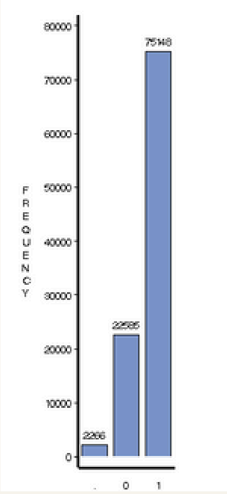
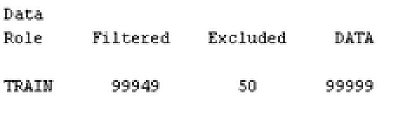


Fig. 8 Kids distribution



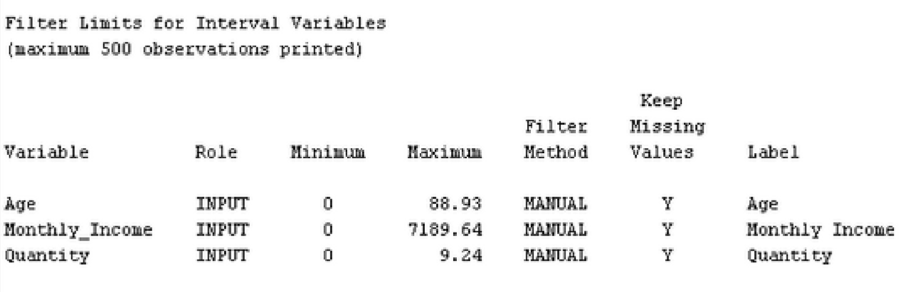
- We have a lot of missing data in the variable **Reviews**, so it is hard to take conclusions in this phase;
- There are 11 outliers in the **Quantity** variable;
- As we can see from the variable **Kids**, The majority of clients have kids.

• Outliers



```
Data
Role     Filtered   Excluded    DATA

TRAIN     99949        50       99999
```

50 outliers excluded



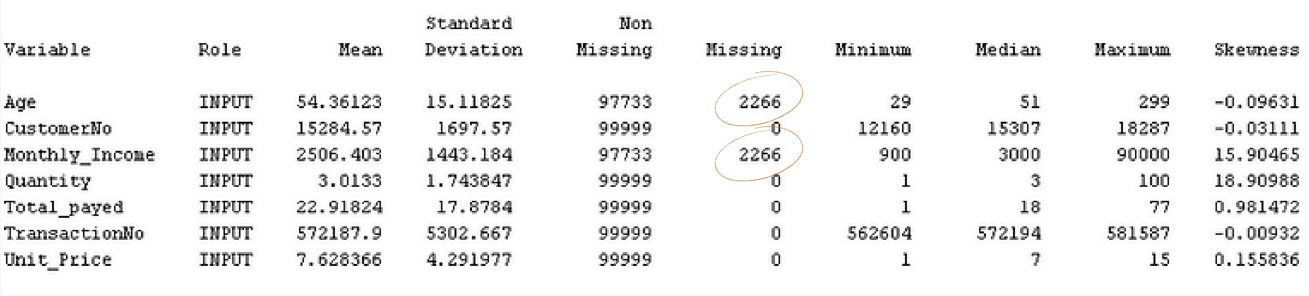
```
Filter Limits for Interval Variables
(maximum 500 observations printed)

                                                        Keep
                                               Filter   Missing
Variable          Role     Minimum    Maximum  Method   Values    Label

Age               INPUT       0        88.93   MANUAL     Y       Age
Monthly_Income    INPUT       0      7189.64   MANUAL     Y       Monthly Income
Quantity          INPUT       0         9.24   MANUAL     Y       Quantity
```

Fig. 9 Filter limits for outliers

The data set filters that we used are displayed in the previous spread sheet. These filters were used to detect certain observations, such as severe outliers, unusual class values, missing values, and erroneous data.
As you can see, each variable has a positive or zero lower limit since these variables should not have negative values, and we filtered the highest values to find and exclude outliers that might disturb our analysis.
These outliers were completely removed since they don't represent a great quantity of the data and would probably wrongly influence the data science algorithms that will be applied to this dataset later on. [Annex 5]

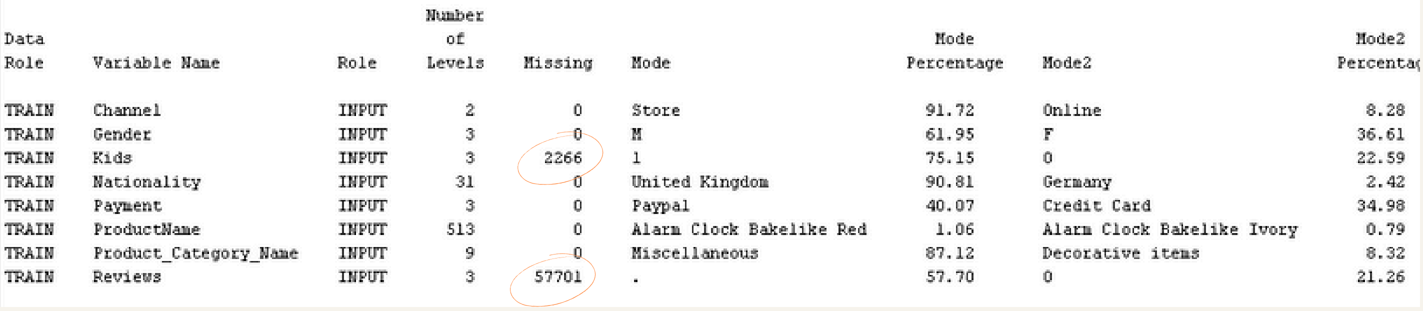
- ## Missing Values

Next, we dealt with missing values, which can be caused by data collection mistakes, partial client replies, and so on. Without these values, we would have risked missing out on useful or critical information that was still recorded in the non-missing variables.



Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness
Age	INPUT	54.36123	15.11825	97733	2266	29	51	299	-0.09631
CustomerNo	INPUT	15284.57	1697.57	99999	0	12160	15307	18287	-0.03111
Monthly_Income	INPUT	2506.403	1443.184	97733	2266	900	3000	90000	15.90465
Quantity	INPUT	3.0133	1.743847	99999	0	1	3	100	18.90988
Total_payed	INPUT	22.91824	17.8784	99999	0	1	18	77	0.981472
TransactionNo	INPUT	572187.9	5302.667	99999	0	562604	572194	581587	-0.00932
Unit_Price	INPUT	7.628366	4.291977	99999	0	1	7	15	0.155836

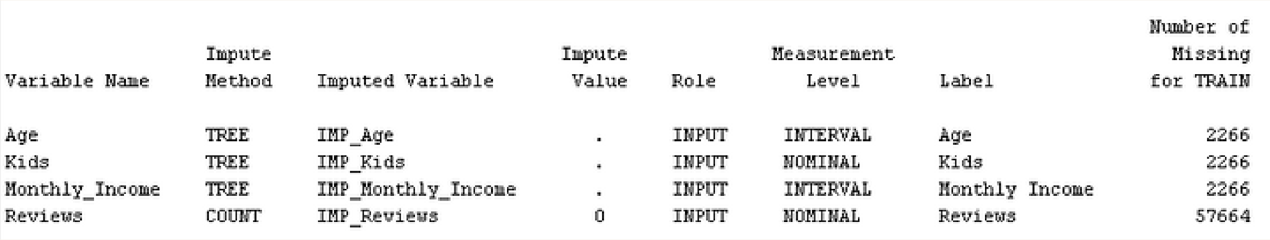
Fig. 10 Descriptive Statistics on Categorical Variables



Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentag
TRAIN	Channel	INPUT	2	0	Store	91.72	Online	8.28
TRAIN	Gender	INPUT	3	0	M	61.95	F	36.61
TRAIN	Kids	INPUT	3	2266	1	75.15	0	22.59
TRAIN	Nationality	INPUT	31	0	United Kingdom	90.81	Germany	2.42
TRAIN	Payment	INPUT	3	0	Paypal	40.07	Credit Card	34.98
TRAIN	ProductName	INPUT	513	0	Alarm Clock Bakelike Red	1.06	Alarm Clock Bakelike Ivory	0.79
TRAIN	Product_Category_Name	INPUT	9	0	Miscellaneous	87.12	Decorative items	8.32
TRAIN	Reviews	INPUT	3	57701	.	57.70	0	21.26

Fig. 11 Descriptive Statistics on Numerical Variables

As a result, all of the numerical variables were converted to the IMP_variable format, and the missing values were filled using the tree method (replacement values are determined by considering each input as a target and utilizing the remaining input and rejected variables as predictors), except for the variable 'Reviews', in which null data points were replaced with the default value 0. The reason we decided to implement this transformation was the variable's nature, since this column represents whether or not a customer left a review, it's safe to assume that they didn't if there is no positive record of it.



Variable Name	Impute Method	Imputed Variable	Impute Value	Role	Measurement Level	Label	Number of Missing for TRAIN
Age	TREE	IMP_Age	.	INPUT	INTERVAL	Age	2266
Kids	TREE	IMP_Kids	.	INPUT	NOMINAL	Kids	2266
Monthly_Income	TREE	IMP_Monthly_Income	.	INPUT	INTERVAL	Monthly Income	2266
Reviews	COUNT	IMP_Reviews	0	INPUT	NOMINAL	Reviews	57664

Fig. 12 Imputed Variables

Furthermore, we discovered 468 duplicate values in our dataset, which we proceeded to remove to avoid an inaccurate portrayal of the data.

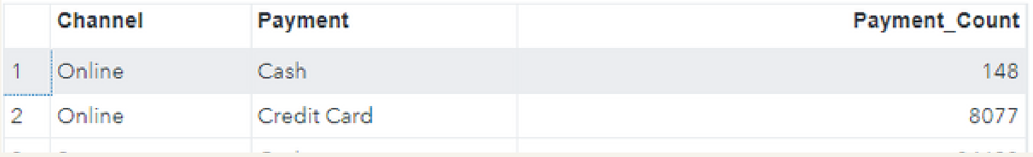
Coherence

The next step before proceeding with any additional analysis was to check the data for inconsistencies that might lead to incorrect results.
For this reason, we wrote and implemented code on SAS guide to check for the possible inconsistencies in the list below and develop solutions. The code can be found in the annexes of this document. [Annex 7]

- The product of Unit_price and Quantity variables should be equal to Total_payed;
- Total_payed should not be superior to annual income (sum of monthly incomes in the time period of the data set);
- Equal ProductName variables should always have the same Unit_Price;
- Normally, customers should only have 1 Nationality (cases when this is false can be analysed individually);
- Kids variable can only assume the values 1 and 0;
- Reviews variable can only assume the values 1 and 0;
- Each CustomerNo must have only one gender;
- Product_Category_Name variable can only assume the values: miscellaneous, beauty and accessories, candles and lights, decorative items, entryway items, kitchenware, beauty office supplies, socks and sombrero;
- Channel variable can only assume the values Online and Store;
- Payment variable can only assume the values: Cash, Paypal and Credit Card;
- Unit_Price, Quantity, Total_payed and Monthly_income variables must have a positive value;
- The Payment cannot be Cash if the Channel is Online;
- Age should not be below 18;
- Every TransactionNo must correspond to only one CustomerNo;
- Every TransactionNo must be completed on the same day;
- Every TransactionNo must be done through only one Channel;
- Every TransactionNo must have only one Payment;

We found the following inconsistencies in our data:

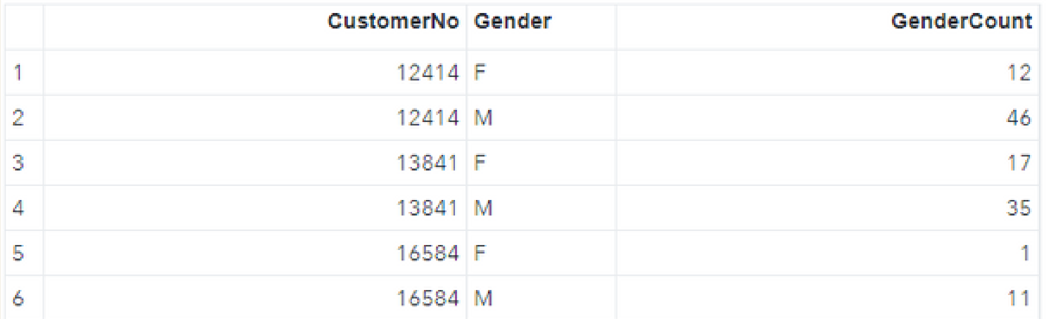
- **Purchases made online but paid for in cash**
  - this issue was resolved by dropping the rows with this error from the dataset.



	Channel	Payment	Payment_Count
1	Online	Cash	148
2	Online	Credit Card	8077

Fig. 13 Payment inconsistency

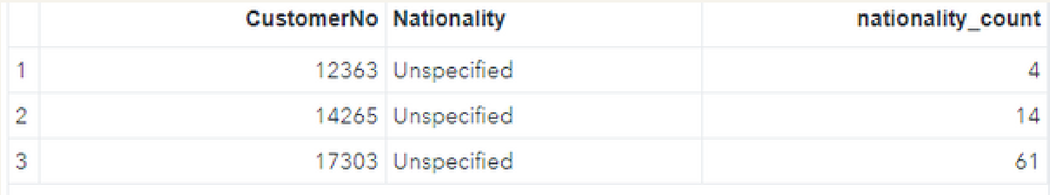
- **Same customer with more than one gender assigned**
    - this issue was resolved by deleting this records, since it was a small number of customers (only 3).



	CustomerNo	Gender	GenderCount
1	12414	F	12
2	12414	M	46
3	13841	F	17
4	13841	M	35
5	16584	F	1
6	16584	M	11

Fig. 14 Gender inconsistency

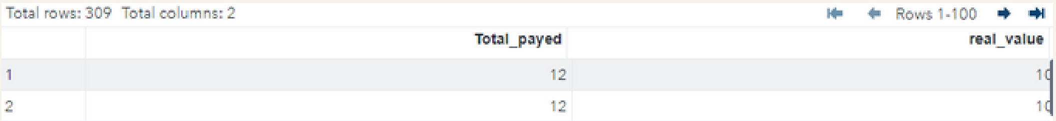
- **Some records contain 'Unspecified' as the nationality value**
    - There were only 3 clients that had this discrepancy, we examined the original data set and client 12363 was the only one that had a lot of missing values, and an extremely odd combination of purchased products. Therefore we only deleted this client. [Annex 8]



	CustomerNo	Nationality	nationality_count
1	12363	Unspecified	4
2	14265	Unspecified	14
3	17303	Unspecified	61

Fig. 15 Nationality inconsistency

- **Inconsistent values in the total payed variable**
    - The values in the column did not correspond to the actual value payed by the customer, since the quantity times the unit price was different than the value displayed. Therefore, we replaced the total payed for the real price.



Total rows: 309 Total columns: 2

	Total_payed	real_value
1	12	10
2	12	10

Fig. 16 Total Payed inconsistency

Analytic-Based Table

Once the data is processed, we move on to creating an Analytical Base Table (ABT), particularly a customer-signature table. New variables are generated in SAS Studio to enhance efficiency and gain insights into client engagement, purchasing habits and patterns, allowing a more effective categorization and understanding of customer behaviour.
Our main idea was to have a single row per customer that includes all possible and relevant information related to them. [Annex 9]

Variables	Description
CustomerNo	Customer ID
Age	Customer's age
Gender	Customer's gender
Nationality	Customer's nationality
Monthly_Income	Customer's monthly income
Kids	If customers have kids (1-yes, 0-no)
Total_reviews	Number of reviews left by customer
Total_Spent	Total money spent
First_purchase	Date of first purchase
Last_purchase	Date of last purchase
Total_quantity	Total number of units bought
Total_transactions	Total number of transactions made
Average_purchase	Average money spent per transaction
Miscellaneous_monetary	Amount of money spent on miscellaneous category
Beauty_Accessories_monetary	Amount of money spent on beauty and accessories category
Candles_Lights_monetary	Amount of money spent on candles and lights category
Decorative_items_monetary	Amount of money spent on decorative items category
Entryway_items_monetary	Amount of money spent on entryway items category
Kitchenware_monetary	Amount of money spent on kitchenware category
Office_supplies_monetary	Amount of money spent on beauty office supplies category
Socks_monetary	Amount of money spent on socks category

Sombrero_monetary	Amount of money spent on sombrero category
Miscellaneous_quantity	Number of items bought from miscellaneous category
Beauty_Accessories_quantity	Number of items bought from beauty and accessories category
Candles_Lights_quantity	Number of items bought from candles and lights category
Decorative_items_quantity	Number of items bought from decorative items category
Entryway_items_quantity	Number of items bought from entryway items category
Kitchenware_quantity	Number of items bought from kitchenware category
Office_supplies_quantity	Number of items bought from office supplies category
Socks_quantity	Number of items bought from socks category
Sombrero_quantity	Number of items bought from sombrero category
Online_monetary	Amount of money spent online
Store_monetary	Amount of money spent in store
Cash_monetary	Amount payed with cash
PayPal_monetary	Amount payed with PayPal
Credit_Card_monetary	Amount payed with Credit Card
Online_transactions	Number of transactions made online
Store_transactions	Number of transactions made in store
Cash_transactions	Number of transactions payed with cash
PayPal_transactions	Number of transactions payed with PayPal
Credit_Card_transactions	Number of transactions payed with Credit Card
Recency_in_days	Number of days since last purchase

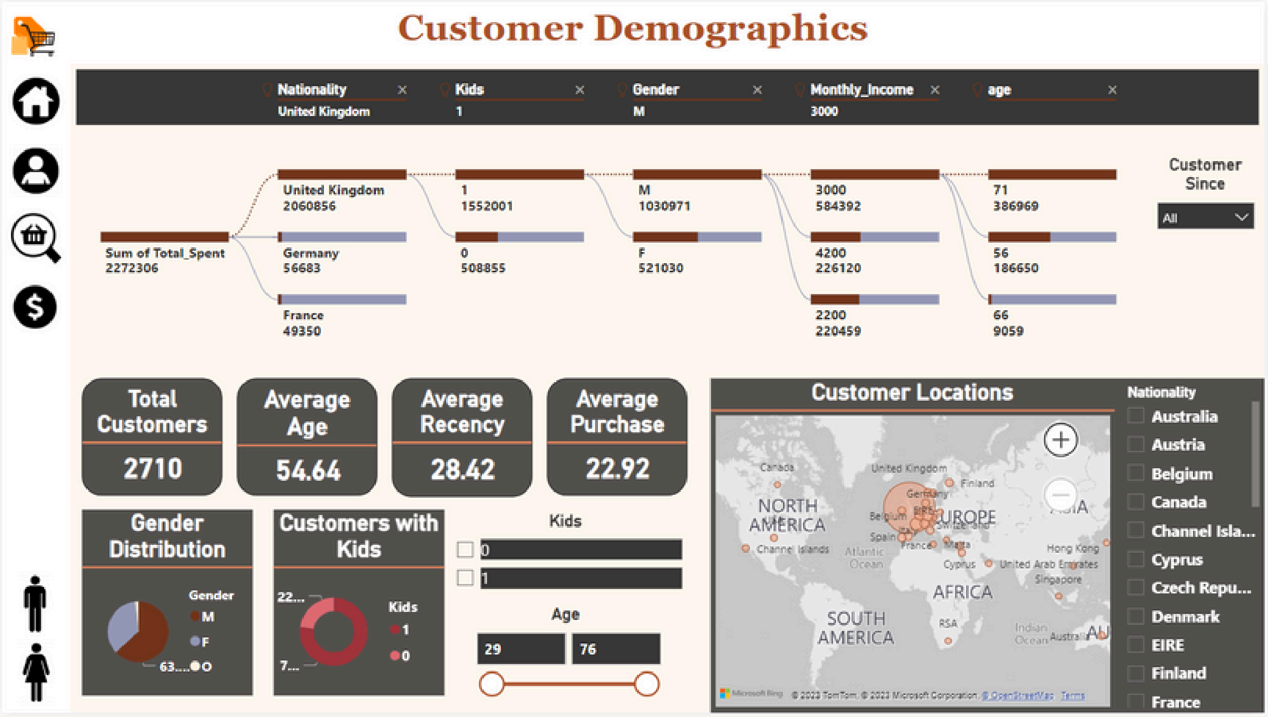
Data Visualizations

In this final section of our research, we utilized PoweBi to build some vital data visualizations. These will be valuable to Mega Market since they allow conclusions to be derived from the dataset provided.

Before beginning to develop the visualizations, we imported both the analytical-based table and the initial data processed from SAS Miner, since if we just used our ABT, the data would be shown in a form that was far too limiting for future visualization creation.

Furthermore, we grouped the visualizations into three dashboards to allow for a more thorough examination of each topic:

- **Dashboard 1 - Customer Demographics**
    - Analyses consumers using visualizations that address personal information, as well as filters, such as age, gender (icons on the right bottom) and whether or not they have children, that allow Mega Market personnel to select which customers' information will be displayed in the visualizations.
    - It was determined that the majority of clients are males with children, aged around 56 years old. They are also mostly found throughout Europe, notably the United Kingdom.
    - Included a visualization that depicts the sort of client that spends the most, and hence the target market to study or possibly advertise to. These customers are from the United Kingdom, are male, have children, have a monthly salary of 3000 euros, and are mostly 71 years old.



Customer Demographics

Nationality	Kids	Gender	Monthly_Income	age
United Kingdom	1	M	3000	

Sum of Total_Spent 2272306

Nationality	Kids	Gender	Monthly_Income	age
United Kingdom 2060856	1 1552001	M 1030971	3000 584392	71 386969
Germany 56683	0 508855	F 521030	4200 226120	56 186650
France 49350			2200 220459	66 9059

Customer Since: All

Total Customers	Average Age	Average Recency	Average Purchase
2710	54.64	28.42	22.92

Gender Distribution

Customers with Kids

Kids: □ 0 □ 1

Age: 29 – 76

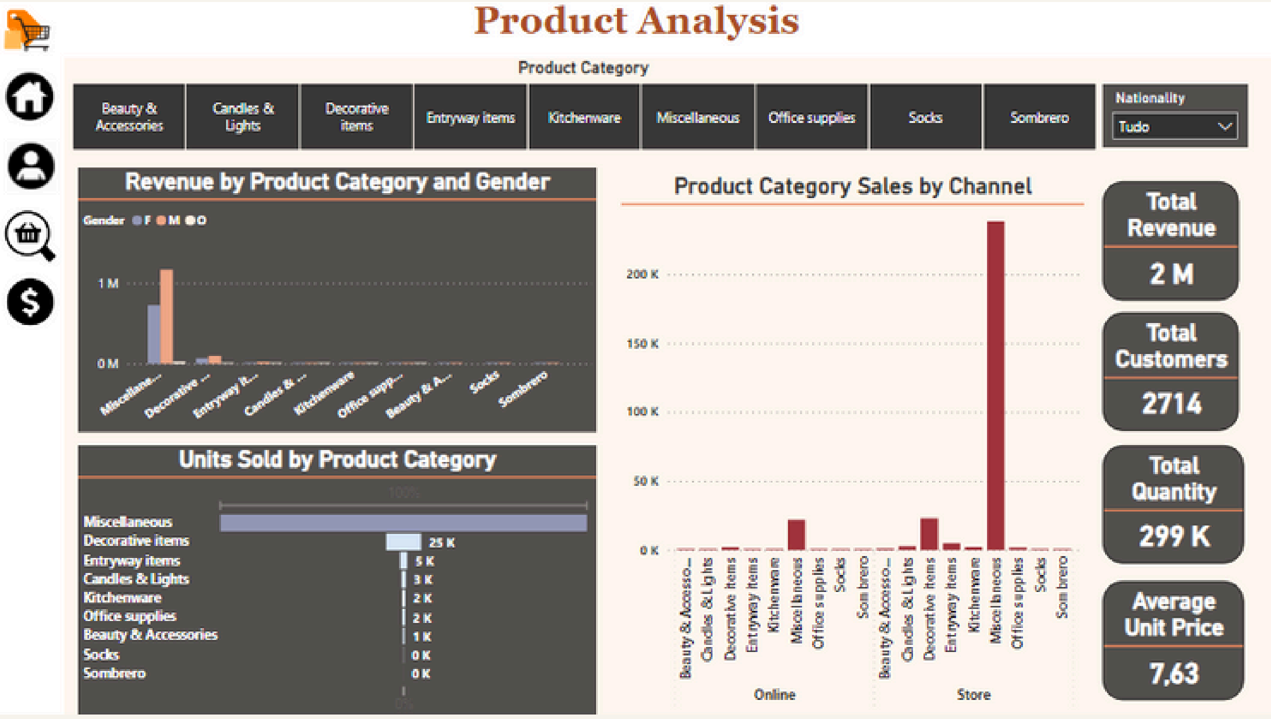
Customer Locations

Nationality: □ Australia □ Austria □ Belgium □ Canada □ Channel Isla... □ Cyprus □ Czech Repu... □ Denmark □ EIRE □ Finland □ France

Fig. 17 Power BI (Costumer Demographics)

- ## Dashboard 2 - Product Analysis

  - Includes relevant information about Mega Market's products, with the ability to filter by product category and customer nationality. Allows for the display of unit sales as well as how a customer's gender and channel type relate to specific product sales.
  - Furthermore, total revenue, quantity, number of consumers, and average unit pricing may be seen by category using our filter, allowing for a deeper knowledge of each of them.
  - It was determined that the most profitable product category was miscellaneuos, which was mostly purchased by men.
  - Furthermore, there is a significant difference between in-store and online sales, however the most profitable category remains the same on both of these channels.



Total Revenue 2 M

Total Customers 2714

Total Quantity 299 K

Average Unit Price 7,63

Fig. 18 Power BI (Product Analysis)

- **Dashboard 3 - Financial Report**
    - The visualizations in this final dashboard relate to all of the financial data in our dataset, with a filter that allows the dashboard user to narrow down the results by transaction date.
    - As previously stated, the Miscellaneous product category generates the greatest capital, with the "Hot Water Bottle" product ranking first.
    - The strongest month for sales, with a transaction high of 93, was determined to be December, which might be explained by the Christmas season's influence. Furthermore, the majority of sales are made in-store and paid for with paypal, followed by credit card, and finally cash.
    - Lastly, the country that brings a greater revenue is the United Kingdom.

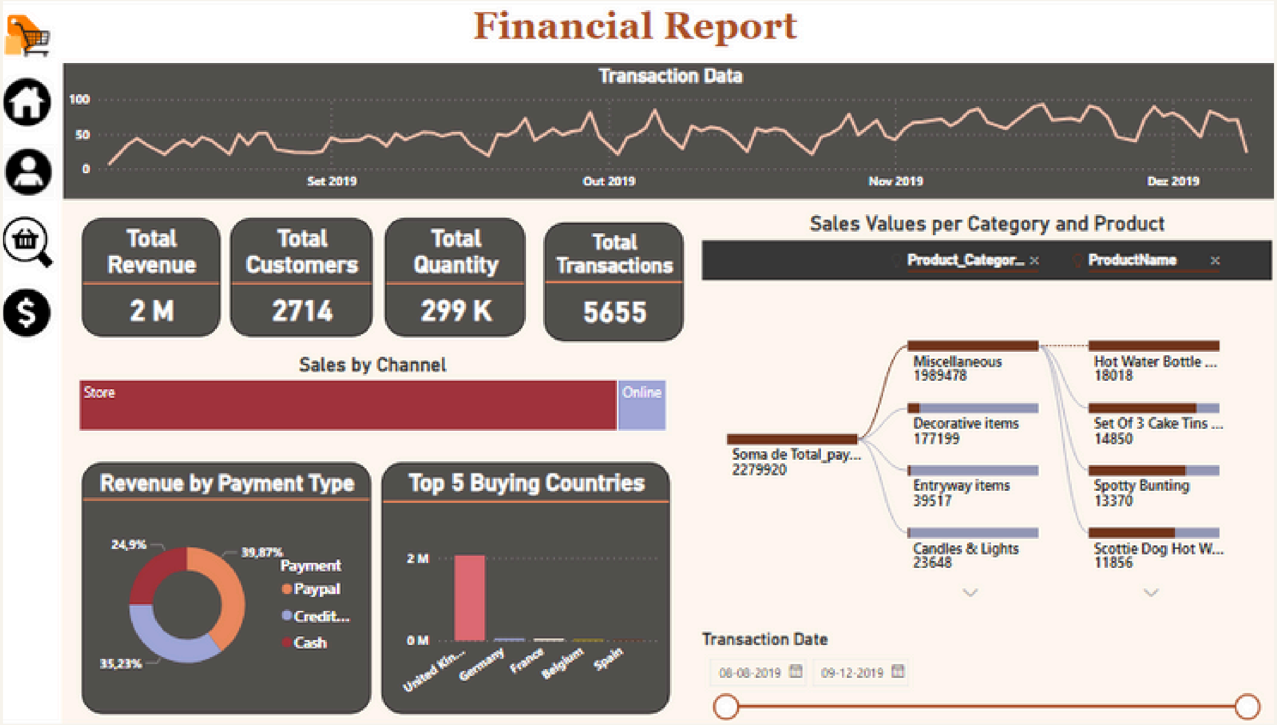


Fig. 19 Power BI (Financial Report)

Conclusion

After executing our investigation, we applied diverse analyses, methodologies, and visual representations to reveal patterns and formulate conclusions regarding Mega Market.

We used different methods and tools like SAS Enterprise Miner, SAS Studio, Excel, and Power BI to reach our project goals. We wanted to make the most of each tool to create a customer signature table(ABT) and different dashboards. This helps us understand more about Mega Market's customers and sales.

We initiated the process by addressing data pre-processing tasks, such as handling outliers, missing values, and inconsistencies. After that, in the development of the customer signature table, our goal was to create meaningful variables that would contribute with valuable insights and essential information about the customers and their behaviors.

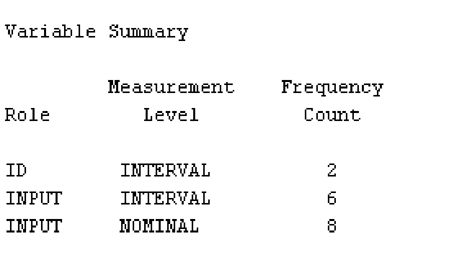
When it comes to our Power BI dashboards, our main objective was to craft visually attractive representations that simultaneously gather information about customers, products, and sales.

Now that our analysis is done, we can answer questions about customer behavior and help with the making of informed business decisions based on the facts we found.

Concluding, we were able to prepare the data for sophisticated analytic methods and provide some business insights while also resolving the absence of information on Mega Market’s activities and clients' buying habits.

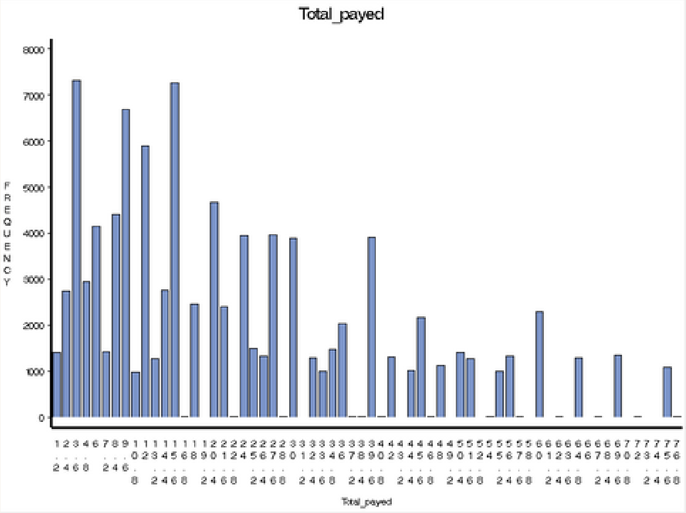
Annexes

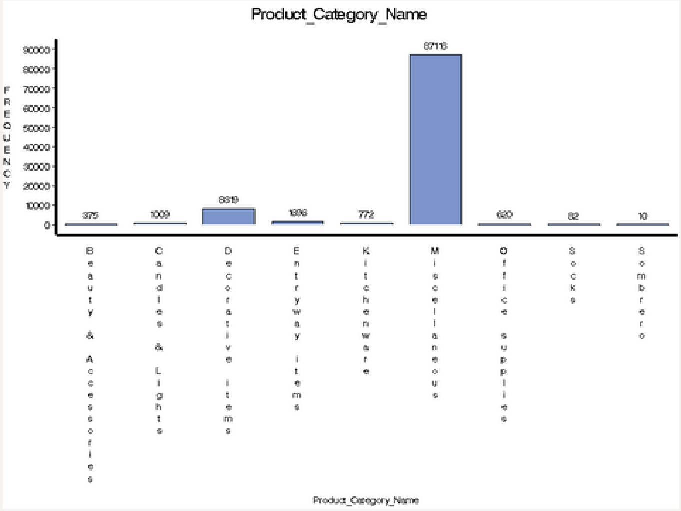
Annex 1 - Variable Summary

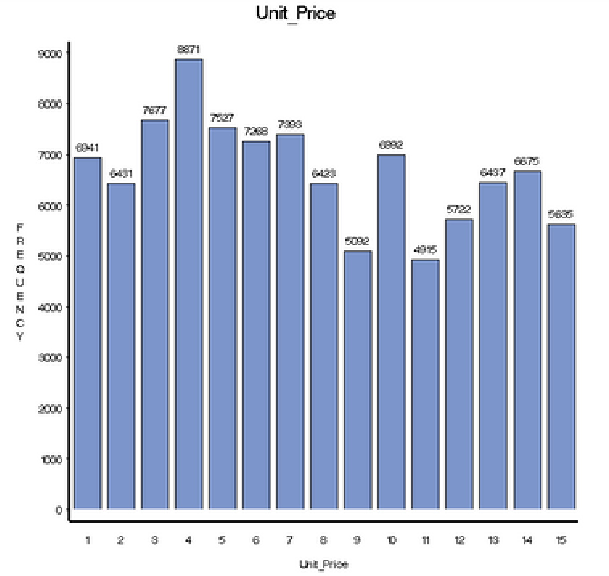


```
Variable Summary

         Measurement    Frequency
Role        Level         Count

ID         INTERVAL         2
INPUT      INTERVAL         6
INPUT      NOMINAL          8
```

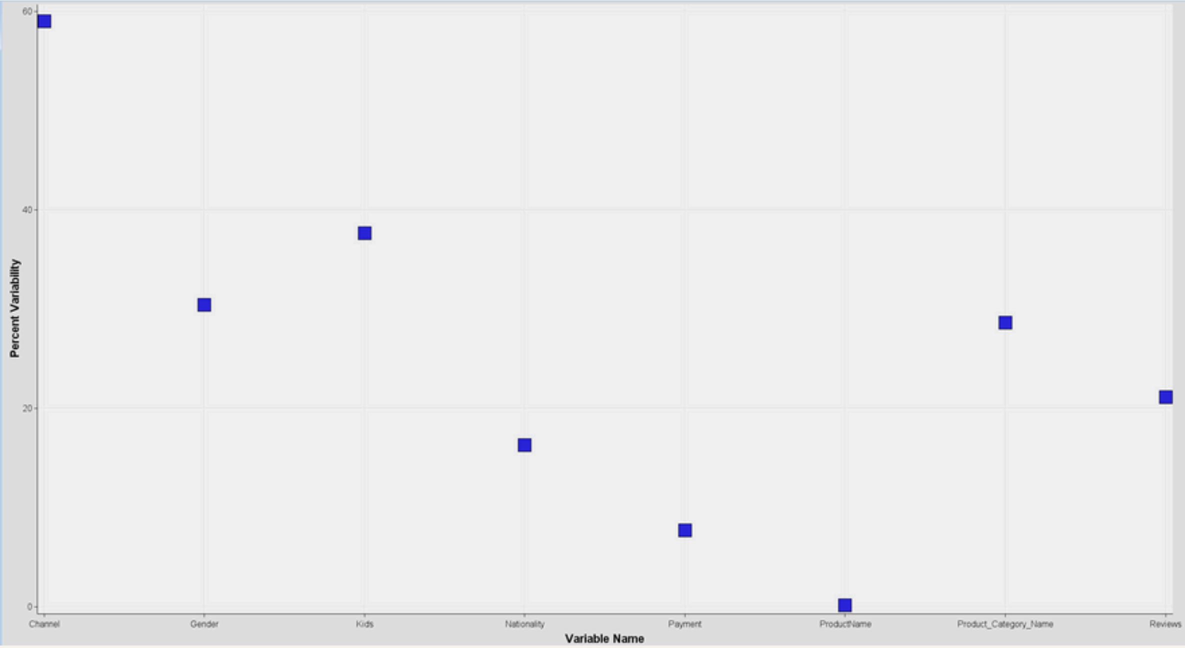
Annex 2 - Multiplot Graphs



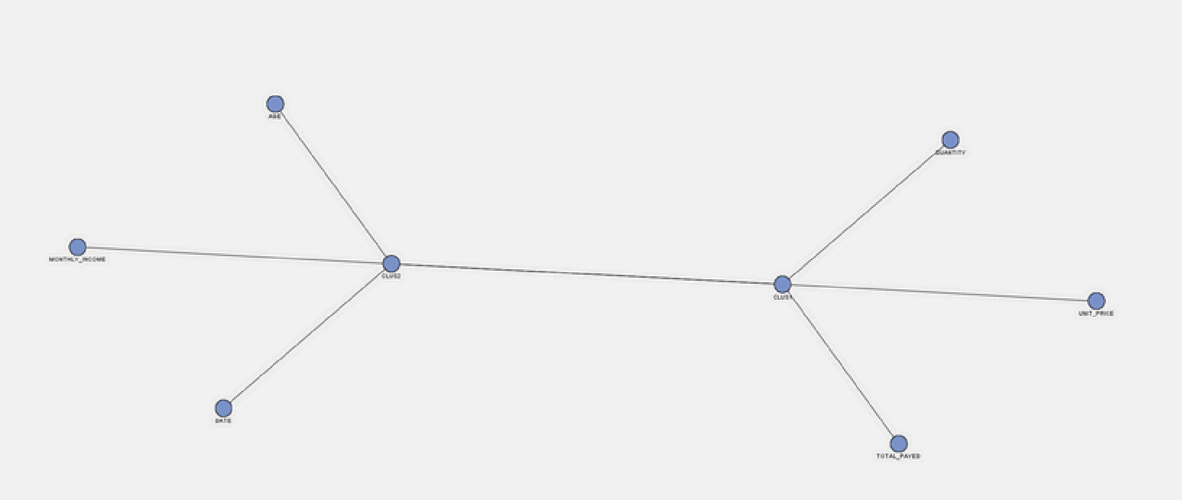




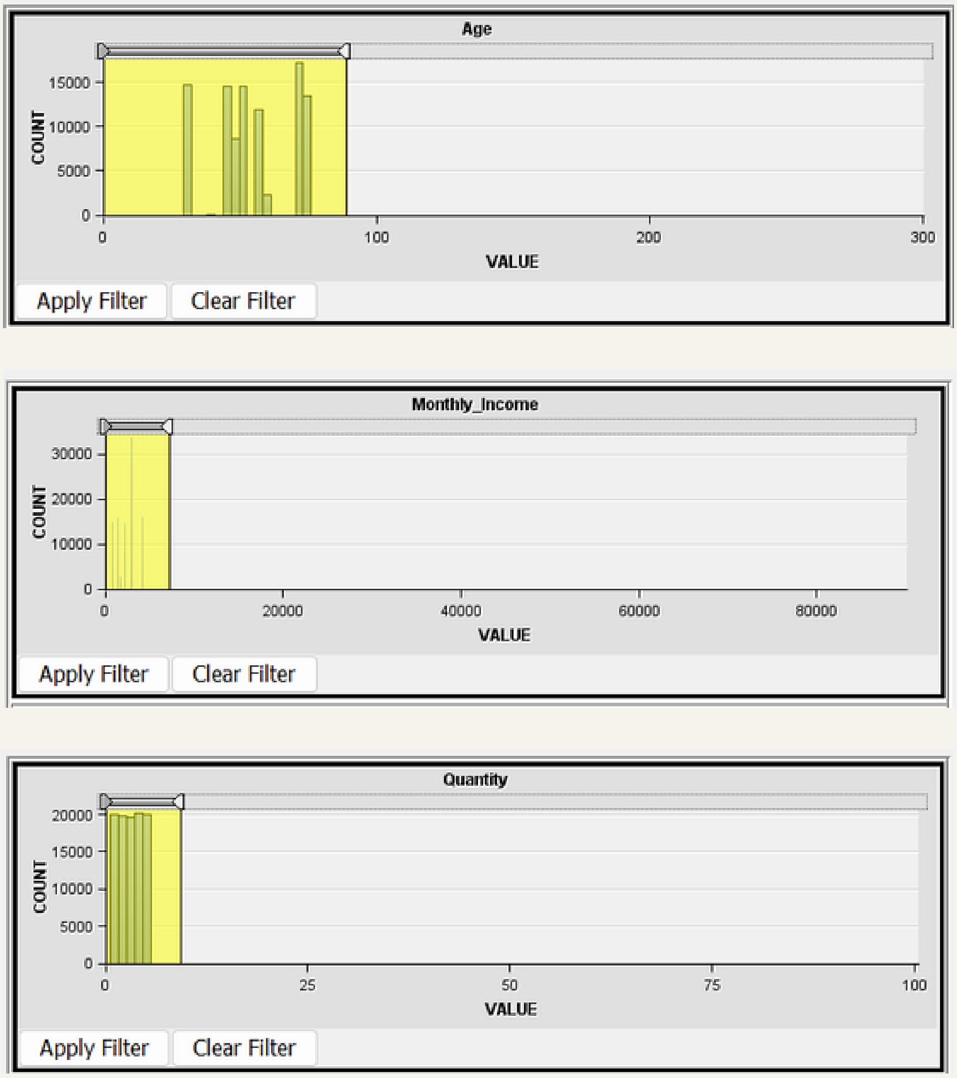
Annex 3 - Variance of Variables before Imputing



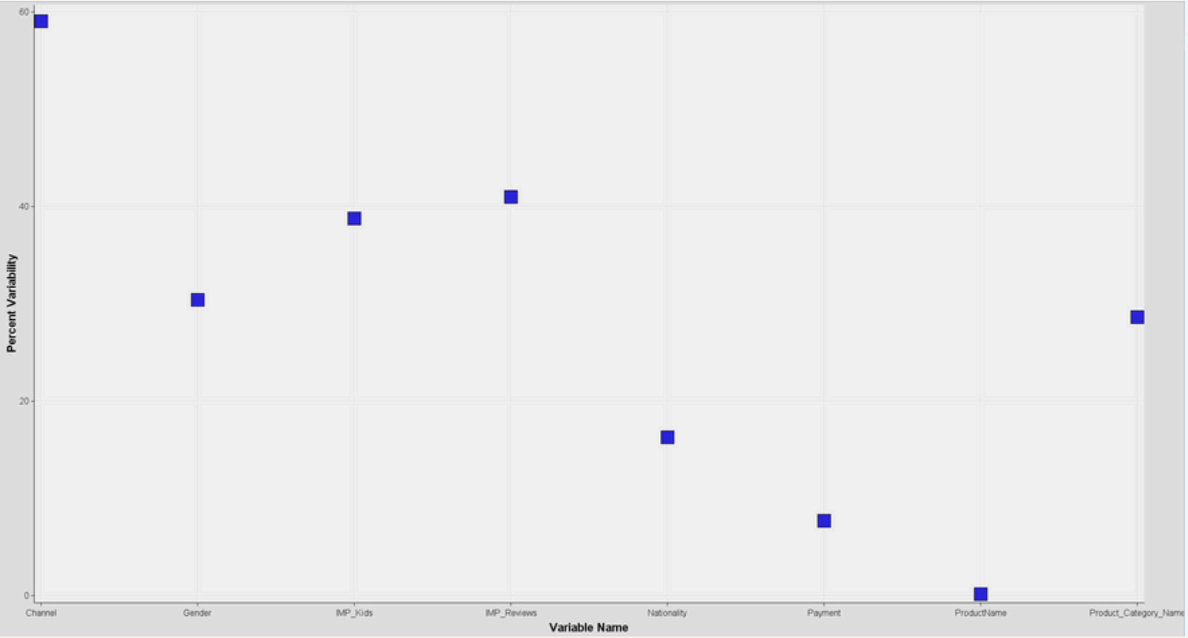
Annex 4 - Variable Clusters



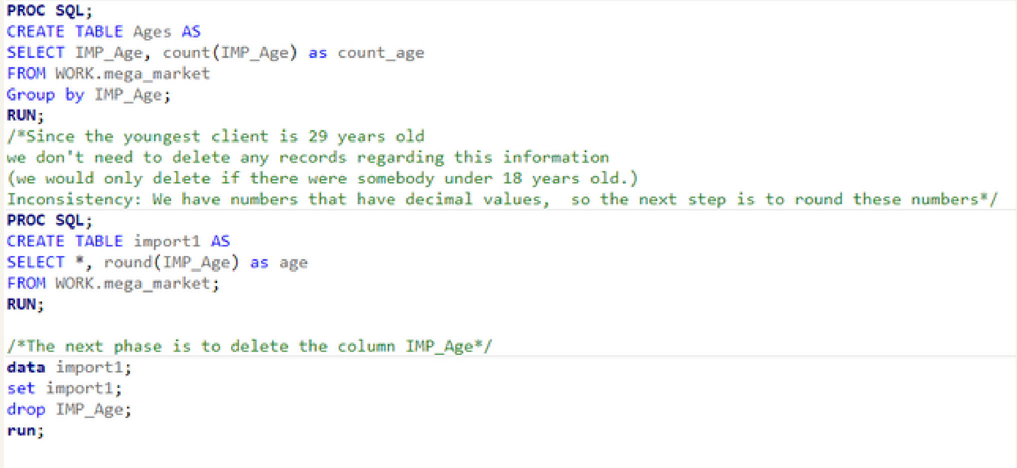
Annex 5 - Variable Filters



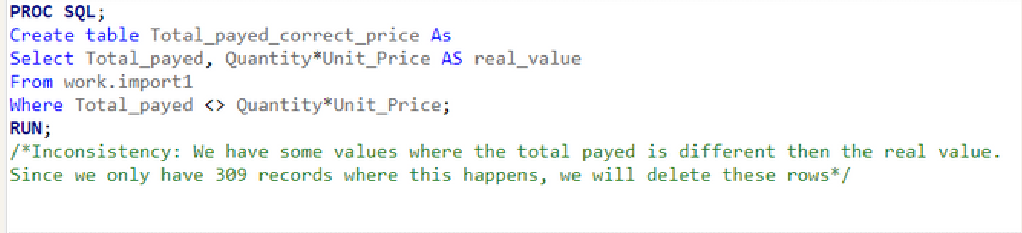
Annex 6 - Variance of Variables after Imputing



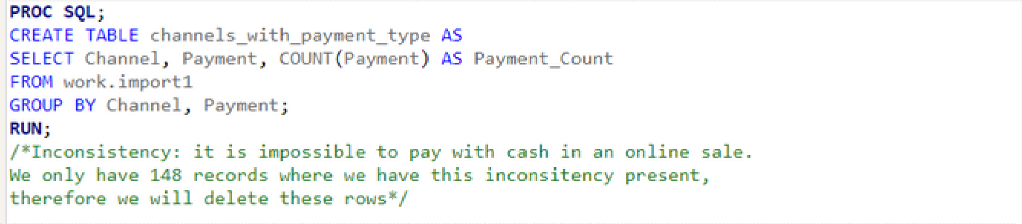
Annex 7 - Coherence Testing



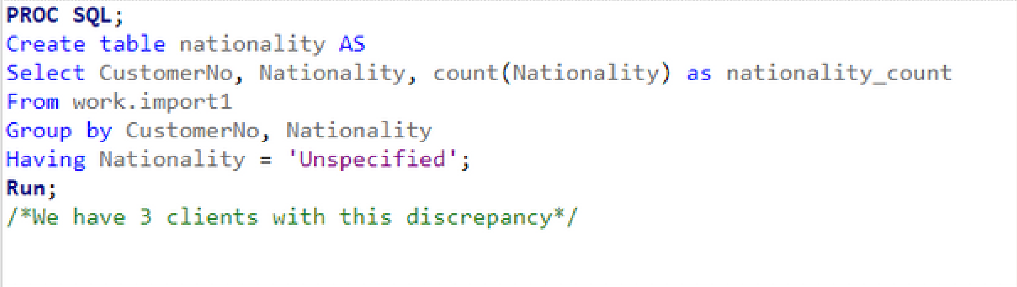
```
PROC SQL;
CREATE TABLE Ages AS
SELECT IMP_Age, count(IMP_Age) as count_age
FROM WORK.mega_market
Group by IMP_Age;
RUN;
/*Since the youngest client is 29 years old
we don't need to delete any records regarding this information
(we would only delete if there were somebody under 18 years old.)
Inconsistency: We have numbers that have decimal values,  so the next step is to round these numbers*/
PROC SQL;
CREATE TABLE import1 AS
SELECT *, round(IMP_Age) as age
FROM WORK.mega_market;
RUN;

/*The next phase is to delete the column IMP_Age*/
data import1;
set import1;
drop IMP_Age;
run;
```



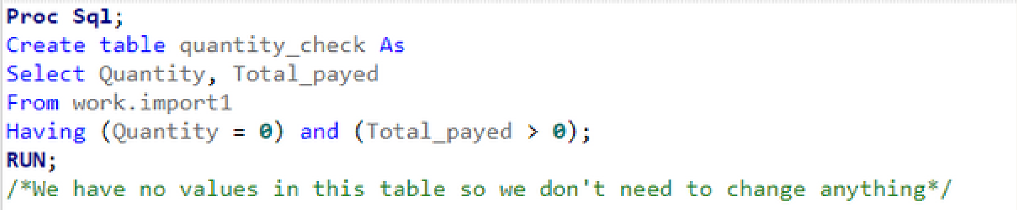
```
PROC SQL;
Create table Total_payed_correct_price As
Select Total_payed, Quantity*Unit_Price AS real_value
From work.import1
Where Total_payed <> Quantity*Unit_Price;
RUN;
/*Inconsistency: We have some values where the total payed is different then the real value.
Since we only have 309 records where this happens, we will delete these rows*/
```



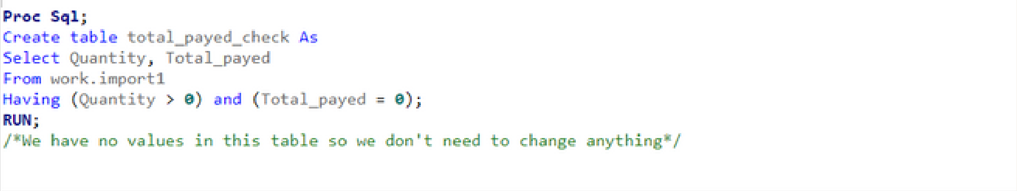
```
PROC SQL;
CREATE TABLE channels_with_payment_type AS
SELECT Channel, Payment, COUNT(Payment) AS Payment_Count
FROM work.import1
GROUP BY Channel, Payment;
RUN;
/*Inconsistency: it is impossible to pay with cash in an online sale.
We only have 148 records where we have this inconsitency present,
therefore we will delete these rows*/
```



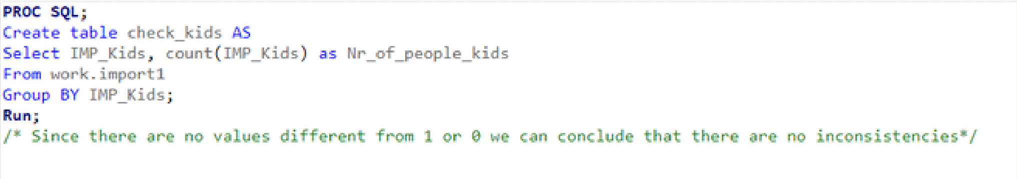
```
PROC SQL;
Create table nationality AS
Select CustomerNo, Nationality, count(Nationality) as nationality_count
From work.import1
Group by CustomerNo, Nationality
Having Nationality = 'Unspecified';
Run;
/*We have 3 clients with this discrepancy*/
```



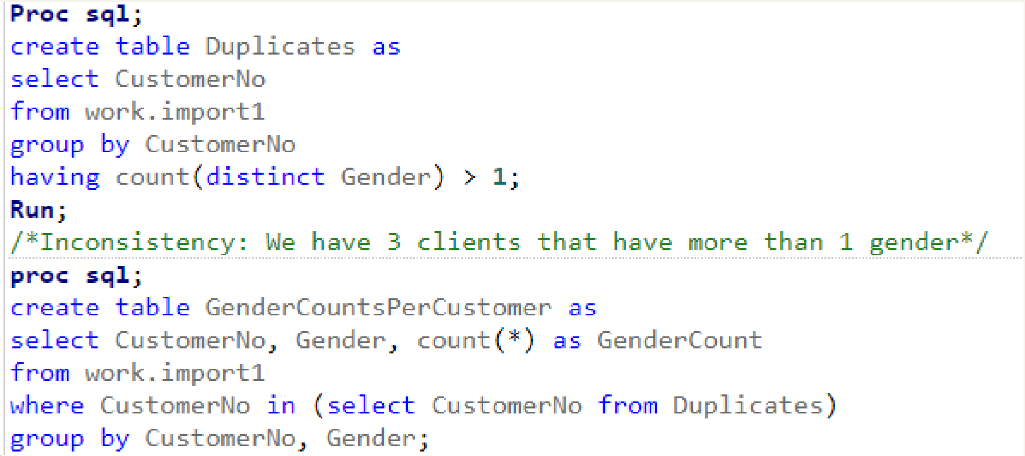
```
Proc Sql;
Create table quantity_check As
Select Quantity, Total_payed
From work.import1
Having (Quantity = 0) and (Total_payed > 0);
RUN;
/*We have no values in this table so we don't need to change anything*/
```



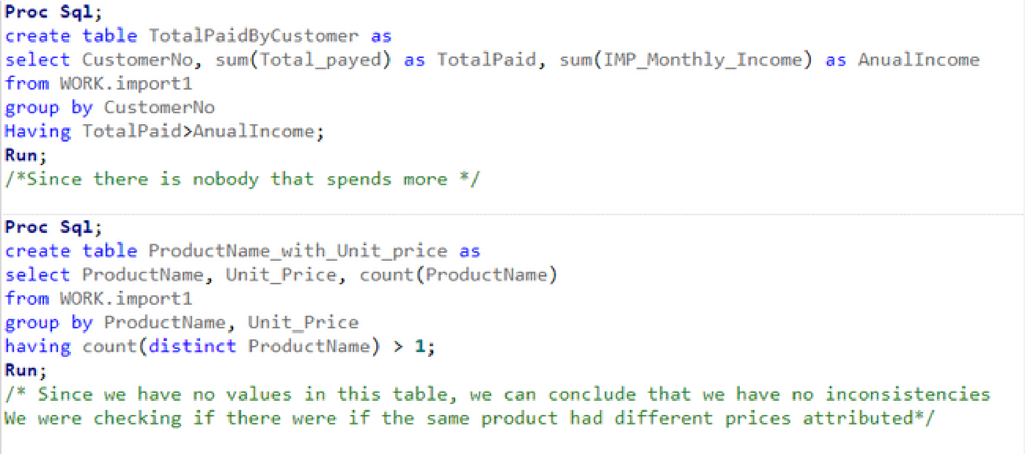
```
Proc Sql;
Create table total_payed_check As
Select Quantity, Total_payed
From work.import1
Having (Quantity > 0) and (Total_payed = 0);
RUN;
/*We have no values in this table so we don't need to change anything*/
```



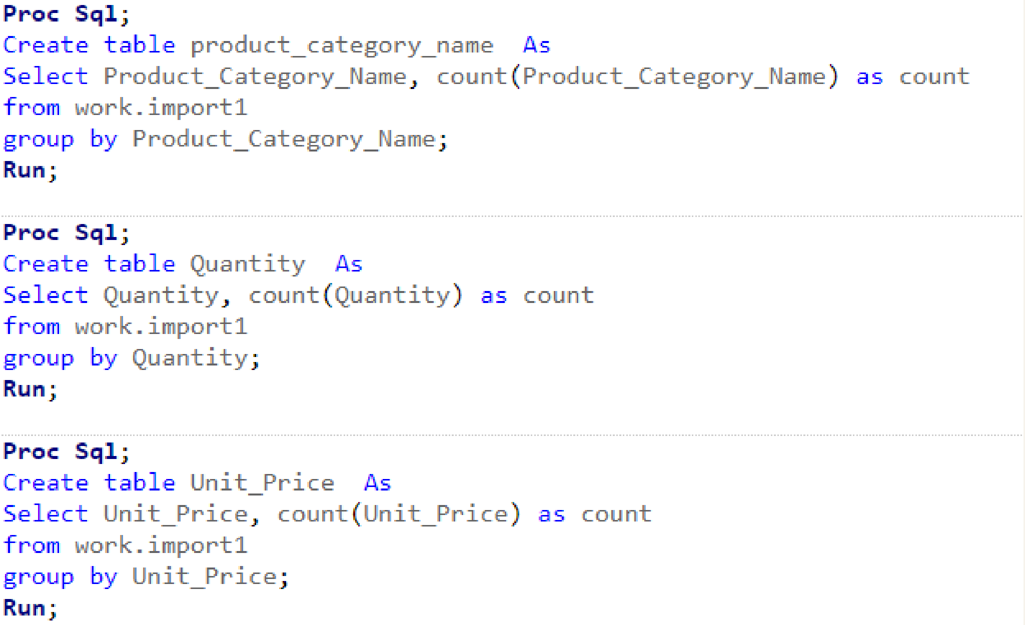
```
PROC SQL;
Create table check_kids AS
Select IMP_Kids, count(IMP_Kids) as Nr_of_people_kids
From work.import1
Group BY IMP_Kids;
Run;
/* Since there are no values different from 1 or 0 we can conclude that there are no inconsistencies*/
```



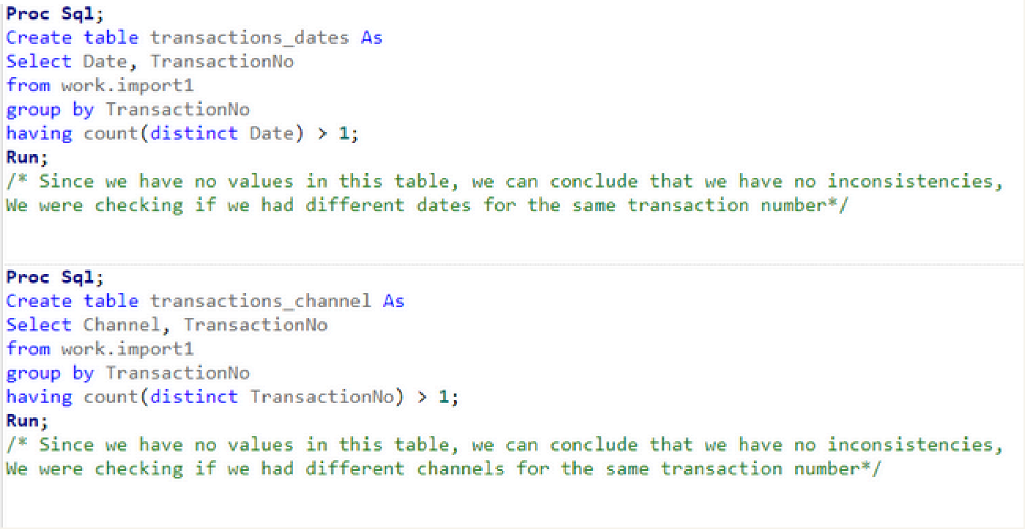
```
Proc sql;
create table Duplicates as
select CustomerNo
from work.import1
group by CustomerNo
having count(distinct Gender) > 1;
Run;
/*Inconsistency: We have 3 clients that have more than 1 gender*/
proc sql;
create table GenderCountsPerCustomer as
select CustomerNo, Gender, count(*) as GenderCount
from work.import1
where CustomerNo in (select CustomerNo from Duplicates)
group by CustomerNo, Gender;
```



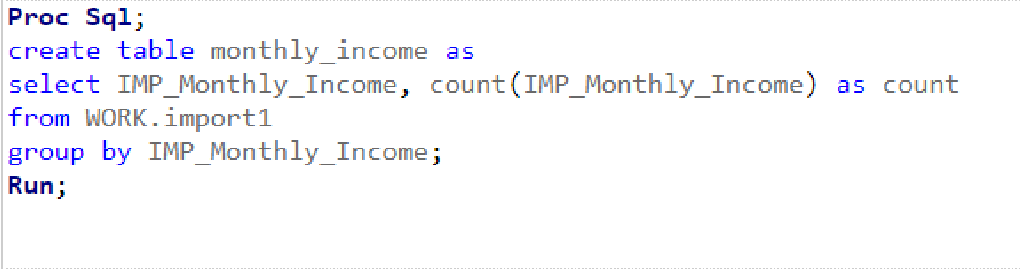
```
Proc Sql;
create table TotalPaidByCustomer as
select CustomerNo, sum(Total_payed) as TotalPaid, sum(IMP_Monthly_Income) as AnualIncome
from WORK.import1
group by CustomerNo
Having TotalPaid>AnualIncome;
Run;
/*Since there is nobody that spends more */

Proc Sql;
create table ProductName_with_Unit_price as
select ProductName, Unit_Price, count(ProductName)
from WORK.import1
group by ProductName, Unit_Price
having count(distinct ProductName) > 1;
Run;
/* Since we have no values in this table, we can conclude that we have no inconsistencies
We were checking if there were if the same product had different prices attributed*/
```



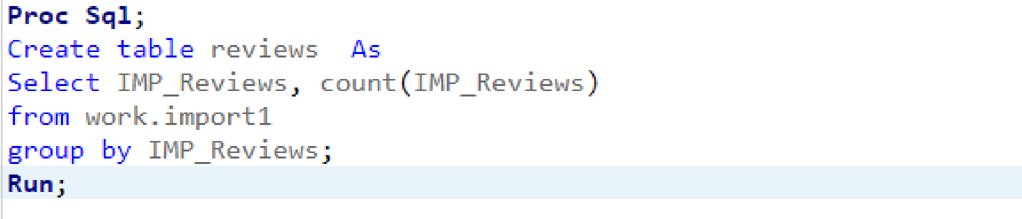
```
Proc Sql;
Create table product_category_name  As
Select Product_Category_Name, count(Product_Category_Name) as count
from work.import1
group by Product_Category_Name;
Run;

Proc Sql;
Create table Quantity  As
Select Quantity, count(Quantity) as count
from work.import1
group by Quantity;
Run;

Proc Sql;
Create table Unit_Price  As
Select Unit_Price, count(Unit_Price) as count
from work.import1
group by Unit_Price;
Run;
```



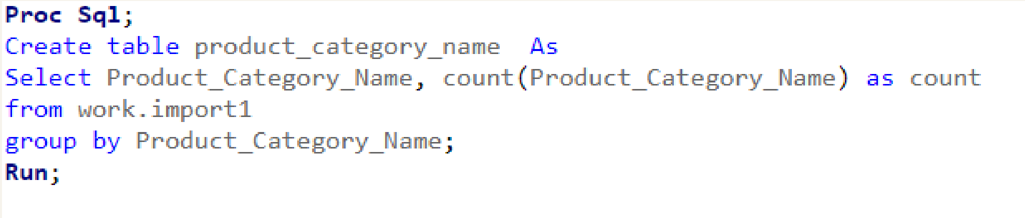
```
Proc Sql;
Create table transactions_dates As
Select Date, TransactionNo
from work.import1
group by TransactionNo
having count(distinct Date) > 1;
Run;
/* Since we have no values in this table, we can conclude that we have no inconsistencies,
We were checking if we had different dates for the same transaction number*/

Proc Sql;
Create table transactions_channel As
Select Channel, TransactionNo
from work.import1
group by TransactionNo
having count(distinct TransactionNo) > 1;
Run;
/* Since we have no values in this table, we can conclude that we have no inconsistencies,
We were checking if we had different channels for the same transaction number*/
```



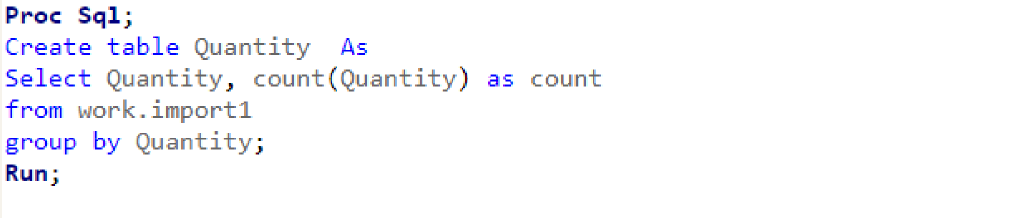
```
Proc Sql;
create table monthly_income as
select IMP_Monthly_Income, count(IMP_Monthly_Income) as count
from WORK.import1
group by IMP_Monthly_Income;
Run;
```



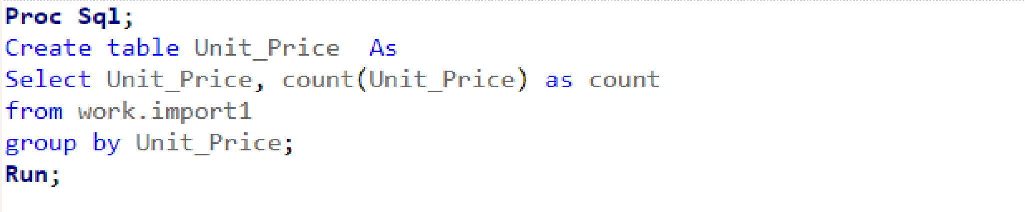
```
Proc Sql;
Create table reviews  As
Select IMP_Reviews, count(IMP_Reviews)
from work.import1
group by IMP_Reviews;
Run;
```



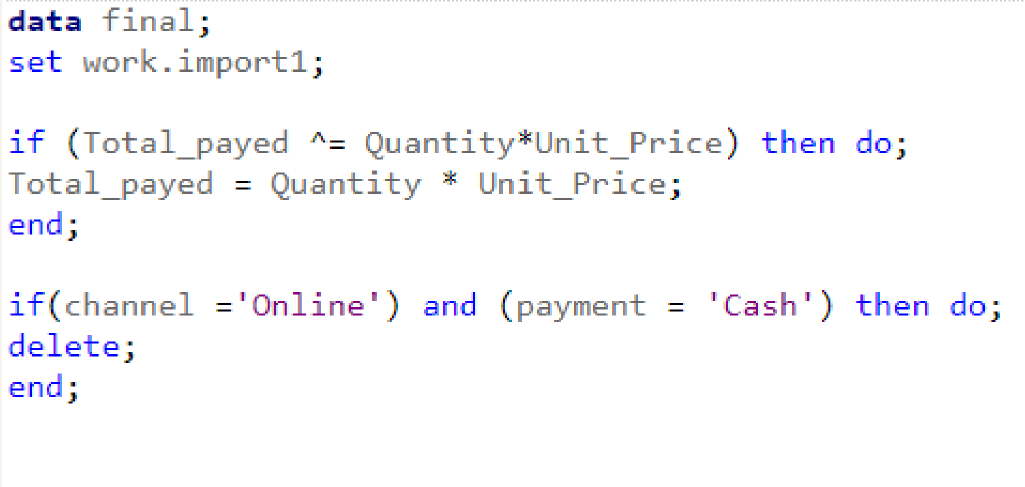
```
Proc Sql;
Create table product_category_name  As
Select Product_Category_Name, count(Product_Category_Name) as count
from work.import1
group by Product_Category_Name;
Run;
```



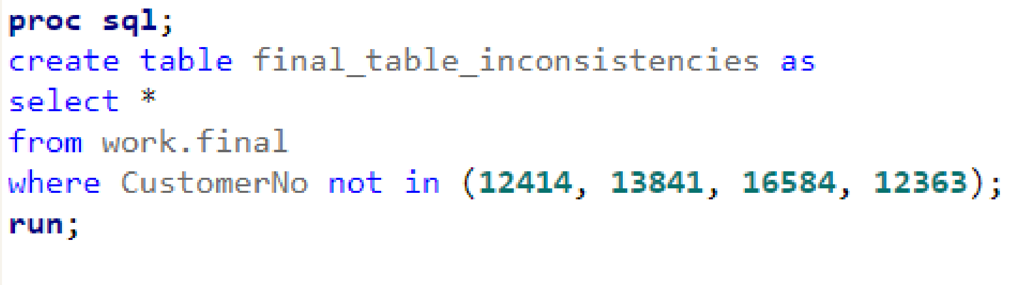
```
Proc Sql;
Create table Quantity  As
Select Quantity, count(Quantity) as count
from work.import1
group by Quantity;
Run;
```



```
Proc Sql;
Create table Unit_Price  As
Select Unit_Price, count(Unit_Price) as count
from work.import1
group by Unit_Price;
Run;
```

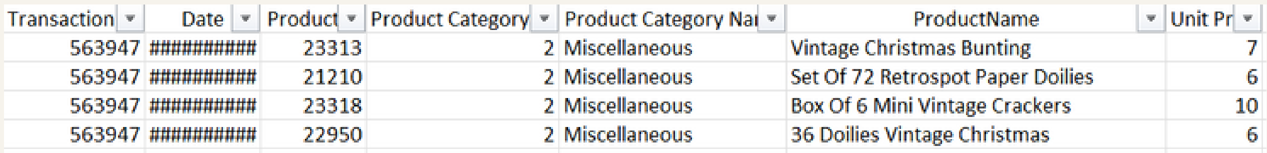


```
data final;
set work.import1;

if (Total_payed ^= Quantity*Unit_Price) then do;
Total_payed = Quantity * Unit_Price;
end;

if(channel ='Online') and (payment = 'Cash') then do;
delete;
end;
```

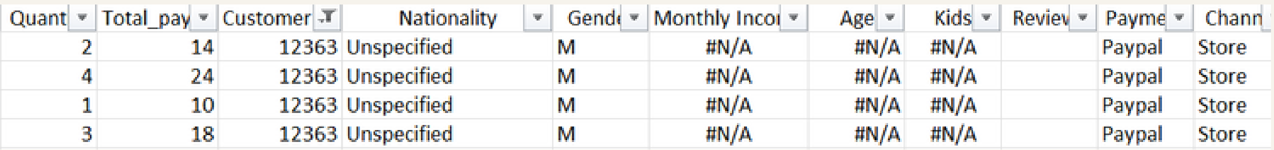


```
proc sql;
create table final_table_inconsistencies as
select *
from work.final
where CustomerNo not in (12414, 13841, 16584, 12363);
run;
```

Annex 8 - Unusual Customer (removed)

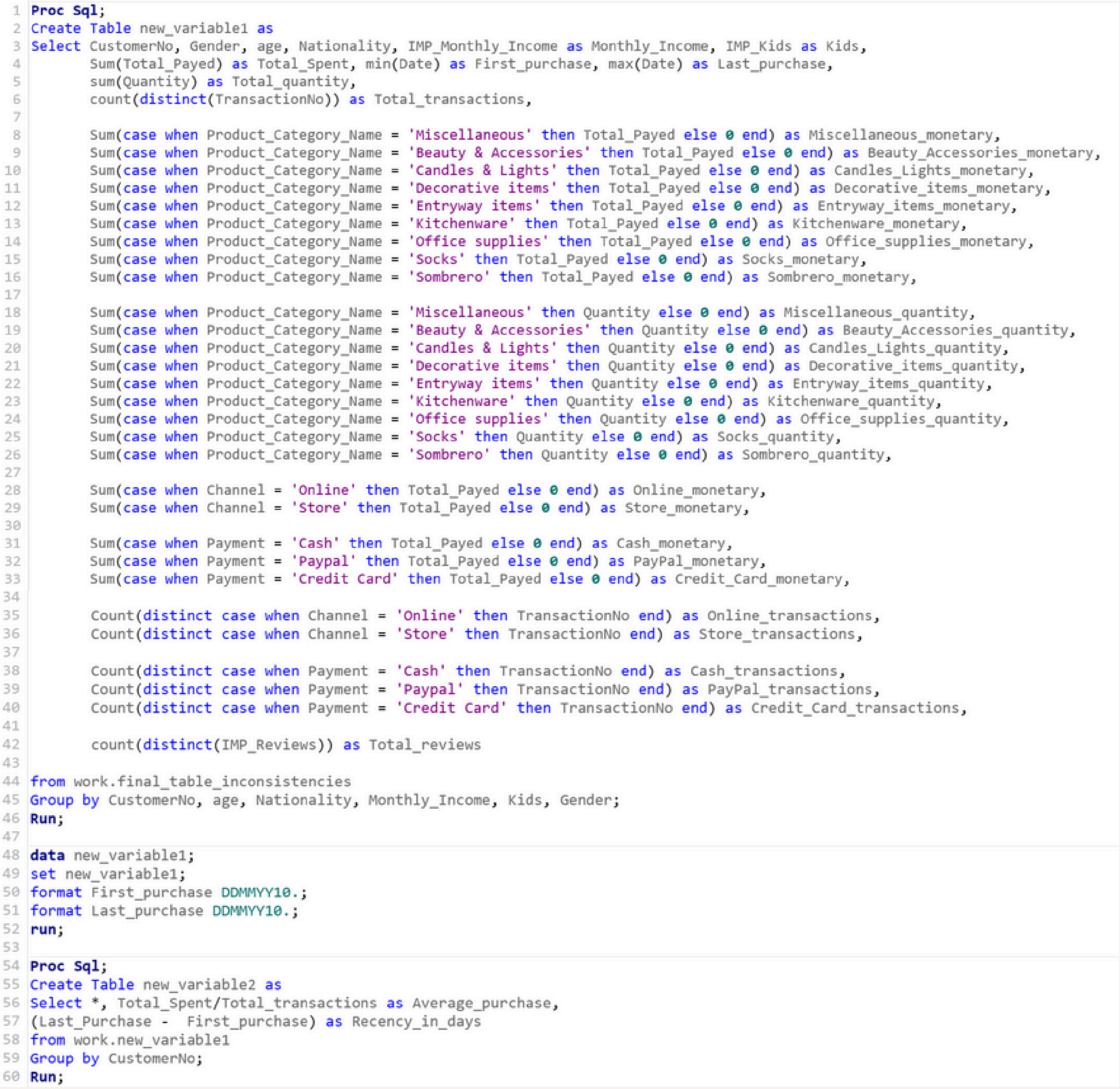


Transaction	Date	Product	Product Category	Product Category Nai	ProductName	Unit Pr
563947	##########	23313	2	Miscellaneous	Vintage Christmas Bunting	7
563947	##########	21210	2	Miscellaneous	Set Of 72 Retrospot Paper Doilies	6
563947	##########	23318	2	Miscellaneous	Box Of 6 Mini Vintage Crackers	10
563947	##########	22950	2	Miscellaneous	36 Doilies Vintage Christmas	6



Quant	Total_pay	Customer	Nationality	Gende	Monthly Incoi	Age	Kids	Reviev	Payme	Chann
2	14	12363	Unspecified	M	#N/A	#N/A	#N/A		Paypal	Store
4	24	12363	Unspecified	M	#N/A	#N/A	#N/A		Paypal	Store
1	10	12363	Unspecified	M	#N/A	#N/A	#N/A		Paypal	Store
3	18	12363	Unspecified	M	#N/A	#N/A	#N/A		Paypal	Store

Annex 9 - Creation of Analytical Base Table and New Variables



```
Proc Sql;
Create Table new_variable1 as
Select CustomerNo, Gender, age, Nationality, IMP_Monthly_Income as Monthly_Income, IMP_Kids as Kids,
       Sum(Total_Payed) as Total_Spent, min(Date) as First_purchase, max(Date) as Last_purchase,
       sum(Quantity) as Total_quantity,
       count(distinct(TransactionNo)) as Total_transactions,

       Sum(case when Product_Category_Name = 'Miscellaneous' then Total_Payed else 0 end) as Miscellaneous_monetary,
       Sum(case when Product_Category_Name = 'Beauty & Accessories' then Total_Payed else 0 end) as Beauty_Accessories_monetary,
       Sum(case when Product_Category_Name = 'Candles & Lights' then Total_Payed else 0 end) as Candles_Lights_monetary,
       Sum(case when Product_Category_Name = 'Decorative items' then Total_Payed else 0 end) as Decorative_items_monetary,
       Sum(case when Product_Category_Name = 'Entryway items' then Total_Payed else 0 end) as Entryway_items_monetary,
       Sum(case when Product_Category_Name = 'Kitchenware' then Total_Payed else 0 end) as Kitchenware_monetary,
       Sum(case when Product_Category_Name = 'Office supplies' then Total_Payed else 0 end) as Office_supplies_monetary,
       Sum(case when Product_Category_Name = 'Socks' then Total_Payed else 0 end) as Socks_monetary,
       Sum(case when Product_Category_Name = 'Sombrero' then Total_Payed else 0 end) as Sombrero_monetary,

       Sum(case when Product_Category_Name = 'Miscellaneous' then Quantity else 0 end) as Miscellaneous_quantity,
       Sum(case when Product_Category_Name = 'Beauty & Accessories' then Quantity else 0 end) as Beauty_Accessories_quantity,
       Sum(case when Product_Category_Name = 'Candles & Lights' then Quantity else 0 end) as Candles_Lights_quantity,
       Sum(case when Product_Category_Name = 'Decorative items' then Quantity else 0 end) as Decorative_items_quantity,
       Sum(case when Product_Category_Name = 'Entryway items' then Quantity else 0 end) as Entryway_items_quantity,
       Sum(case when Product_Category_Name = 'Kitchenware' then Quantity else 0 end) as Kitchenware_quantity,
       Sum(case when Product_Category_Name = 'Office supplies' then Quantity else 0 end) as Office_supplies_quantity,
       Sum(case when Product_Category_Name = 'Socks' then Quantity else 0 end) as Socks_quantity,
       Sum(case when Product_Category_Name = 'Sombrero' then Quantity else 0 end) as Sombrero_quantity,

       Sum(case when Channel = 'Online' then Total_Payed else 0 end) as Online_monetary,
       Sum(case when Channel = 'Store' then Total_Payed else 0 end) as Store_monetary,

       Sum(case when Payment = 'Cash' then Total_Payed else 0 end) as Cash_monetary,
       Sum(case when Payment = 'Paypal' then Total_Payed else 0 end) as PayPal_monetary,
       Sum(case when Payment = 'Credit Card' then Total_Payed else 0 end) as Credit_Card_monetary,

       Count(distinct case when Channel = 'Online' then TransactionNo end) as Online_transactions,
       Count(distinct case when Channel = 'Store' then TransactionNo end) as Store_transactions,

       Count(distinct case when Payment = 'Cash' then TransactionNo end) as Cash_transactions,
       Count(distinct case when Payment = 'Paypal' then TransactionNo end) as PayPal_transactions,
       Count(distinct case when Payment = 'Credit Card' then TransactionNo end) as Credit_Card_transactions,

       count(distinct(IMP_Reviews)) as Total_reviews

from work.final_table_inconsistencies
Group by CustomerNo, age, Nationality, Monthly_Income, Kids, Gender;
Run;

data new_variable1;
set new_variable1;
format First_purchase DDMMYY10.;
format Last_purchase DDMMYY10.;
run;

Proc Sql;
Create Table new_variable2 as
Select *, Total_Spent/Total_transactions as Average_purchase,
(Last_Purchase -  First_purchase) as Recency_in_days
from work.new_variable1
Group by CustomerNo;
Run;
```