SAS and MSc Business Analytics - AUEB

Joint Certificate in

SAS Programming and Data Mining

Milestone Project

Erasmia Kornelatou
f2821907

A. Objective of the project

This Milestone Project is part of the required procedure for obtaining the SAS Joint Certificate in SAS Programming and Data Mining.

The objective of the project is to apply techniques for accessing, processing, managing and mining of real world data and to provide solutions to business problems that today's organizations face through the use of Base SAS Programming and SAS Enterprise Miner.

In order to accomplish the above objectives we are given a set of real world POS data that are related to sales of a retail company along with other related data.

We are asked to analyze the given data through the use of Base SAS and SAS Enterprise Miner and to write a relevant report (deliverable) to be handed to the management team of the organization by answering the question that follow.

Datasets description

The datasets consist of POS data from a retail store.

The available data are included in the following tables. The first one of them is related to data about customers and is entitled Customer, the second and the third are related to POS data and are entitled Invoice & Basket respectively, the fourth contains the coding of the payment method done and is entitled Payment_Method, the fifth contains the coding of the promotional activities running and is entitled Promotions, the sixth contains the coding of the suppliers and is entitled Suppliers and finally the seventh contains the coding of the product origin and is entitled Product_Origin

Customer table

CustomerID	CustomerCountry	Day_Of_Birth	Month_Of_Birth	Year_Of_Birth	Gender
12431	Australia	7	9	1979	Male
12433	Norway	4	10	1987	Male
12583	France	13	11	1956	Male

This table is related to the data about the customers and contains the following columns:

- **CustomerID:** Customer ID, (unique for every customer)
- **CustomerCountry:** The country of origin of each customer
- **Day_Of_Birth:** The day when the customer was born
- **Month_Of_Birth:** The month when the customer was born
- **Year_Of_Birth:** The year when the customer was born
- **Gender:** The gender of the customer

Invoice table

InvoiceNo	InvoiceDate	InvoiceTime	CustomerID	Operation	Payment_Method
536365	12/1/2010	8:26 πμ	17850	500	2
536365	12/1/2010	8:26 πμ	17850	500	2
536365	12/1/2010	8:26 πμ	17850	500	2

This table contains data about the issued invoice (sale or return) and contains the following columns:

- **InvoiceNo:** The issue number of the invoice (unique for every invoice)
- **InvoiceDate:** The date when the invoice was issued
- **InvoiceTime:** The time when the invoice was issued
- **CustomerID:** Customer ID, (unique for every customer)
- **Operation:** Denotes whether the invoice is related to Sales (500) or Return (501)
- **Payment_Method:** The code of the payment method

We make the assumption that an invoice can be paid with more than one payment methods. The invoice_table data set contains duplicates. In order to proceed correctly with the project you have to remove them. The correct number of deduplicated observations is 24,517.

Basket table

InvoiceNo	SKU	Description	Product_Origin	Quantity	Unit_Price
536365	58720443050301	WHITE HANG	1	6	2,55
536365	85449120050011	WHITE METAL	2	6	3,39
536365	85449230050011	CREAM CUPID	2	8	2,75

This table contains the following columns:

- **InvoiceNo:** The issue number of the invoice (unique for every invoice)
- **SKU:** The Stock Keeping Unit of the product
- **Description:** The product name
- **Product_Origin:** The code that denotes the origin of the product
- **Quantity:** The quantity of the product sold
- **Unit_Price:** The price per unit of the product

Payment Method table

Code	Method
1	Cash
2	Credit Card
3	Pay Pal
4	Debit Card

This table contains the following columns:

- **Code:** The code of the payment method
- **Method:** The method used b the customer to pay the invoice

Promotion table

Promotion_Code	Promotion_Type

0	No Promotion
1	5% Off
2	10% Off

This table contains the following columns:

- **Promotion_Code:** The code of the promotion
- **Promotion_Type:** The type of the promotion

Suppliers table

Supplier_ID	Supplier_Name
1	J&J
….	….
9	Dragon

This table contains the following columns:

- **Supplier_ID:** The ID of the supplier
- **Supplier_Name:** The name of the supplier

Product Origin table

Region	Code
China	1
Asia (Except China)	2
Europe	3

This table contains the following columns:

- **Region:** The region of origin of the product
- **Code:** The code of the region of origin of the product

Customer table

CustomerID	CustomerCountry	Day_Of_Birth	Month_Of_Birth	Year_Of_Birth	Gender
12431	Australia	7	9	1979	Male
12433	Norway	4	10	1987	Male
12583	France	13	11	1956	Male

This table is related to the data about the customers and contains the following columns:

- **CustomerID:** Customer ID, (unique for every customer)
- **CustomerCountry:** The country of origin of each customer
- **Day_Of_Birth:** The day when the customer was born
- **Month_Of_Birth:** The month when the customer was born
- **Year_Of_Birth:** The year when the customer was born
- **Gender:** The gender of the customer

Invoice table

InvoiceNo	InvoiceDate	InvoiceTime	CustomerID	Operation	Payment_Method
536365	12/1/2010	8:26 πμ	17850	500	2
536365	12/1/2010	8:26 πμ	17850	500	2
536365	12/1/2010	8:26 πμ	17850	500	2

This table contains data about the issued invoice (sale or return) and contains the following columns:

- **InvoiceNo:** The issue number of the invoice (unique for every invoice)
- **InvoiceDate:** The date when the invoice was issued
- **InvoiceTime:** The time when the invoice was issued
- **CustomerID:** Customer ID, (unique for every customer)
- **Operation:** Denotes whether the invoice is related to Sales (500) or Return (501)
- **Payment_Method:** The code of the payment method

We make the assumption that an invoice can be paid with more than one payment methods. The invoice_table data set contains duplicates. In order to proceed correctly with the project you have to remove them. The correct number of deduplicated observations is 24,517.

Basket table

InvoiceNo	SKU	Description	Product_Origin	Quantity	Unit_Price
536365	58720443050301	WHITE HANG	1	6	2,55
536365	85449120050011	WHITE METAL	2	6	3,39
536365	85449230050011	CREAM CUPID	2	8	2,75

This table contains the following columns:

- **InvoiceNo:** The issue number of the invoice (unique for every invoice)
- **SKU:** The Stock Keeping Unit of the product
- **Description:** The product name
- **Product_Origin:** The code that denotes the origin of the product
- **Quantity:** The quantity of the product sold
- **Unit_Price:** The price per unit of the product

Payment Method table

Code	Method
1	Cash
2	Credit Card
3	Pay Pal
4	Debit Card

This table contains the following columns:

- **Code:** The code of the payment method
- **Method:** The method used b the customer to pay the invoice

Promotion table

Promotion_Code	Promotion_Type
0	No Promotion
1	5% Off
2	10% Off

This table contains the following columns:

- **Promotion_Code:** The code of the promotion
- **Promotion_Type:** The type of the promotion

Suppliers table

Supplier_ID	Supplier_Name
1	J&J
….	….
9	Dragon

This table contains the following columns:

- **Supplier_ID:** The ID of the supplier
- **Supplier_Name:** The name of the supplier

Product Origin table

Region	Code
China	1
Asia (Except China)	2
Europe	3

This table contains the following columns:

- **Region:** The region of origin of the product
- **Code:** The code of the region of origin of the product

B. Base SAS Programming

The following tasks require the use of Base SAS. Please take into account the following:

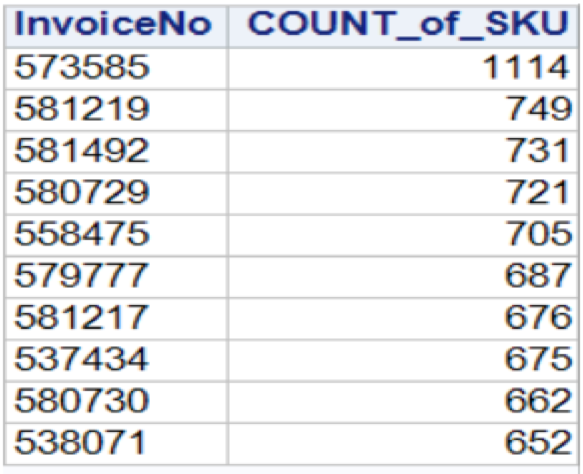
- The data sets should be transformed to SAS format with the use of the data step or through the File -- > Open (for Excel files) and File -- > Import (for raw data files).
- Proc sql can be used only in answering questions where it is explicitly mentioned, where as in any other case it is obligatory to use only the data step or any other procedure except proc sql (e.g. proc means).

Attention: In order to avoid errors when transforming data sets to SAS format, read the variables that will not be used as numbers (e.g. SKU, BasketID) in string type. Also all the new data set to be produced in Base SAS during the project should be stored in the RSULTS library.

We make the assumption that an invoice can be paid with more than one payment methods. The invoice_table data set contains duplicates. In order to proceed correctly with the project you have to remove them. The correct number of deduplicated observations is 24,257.

1. Data pre – processing:

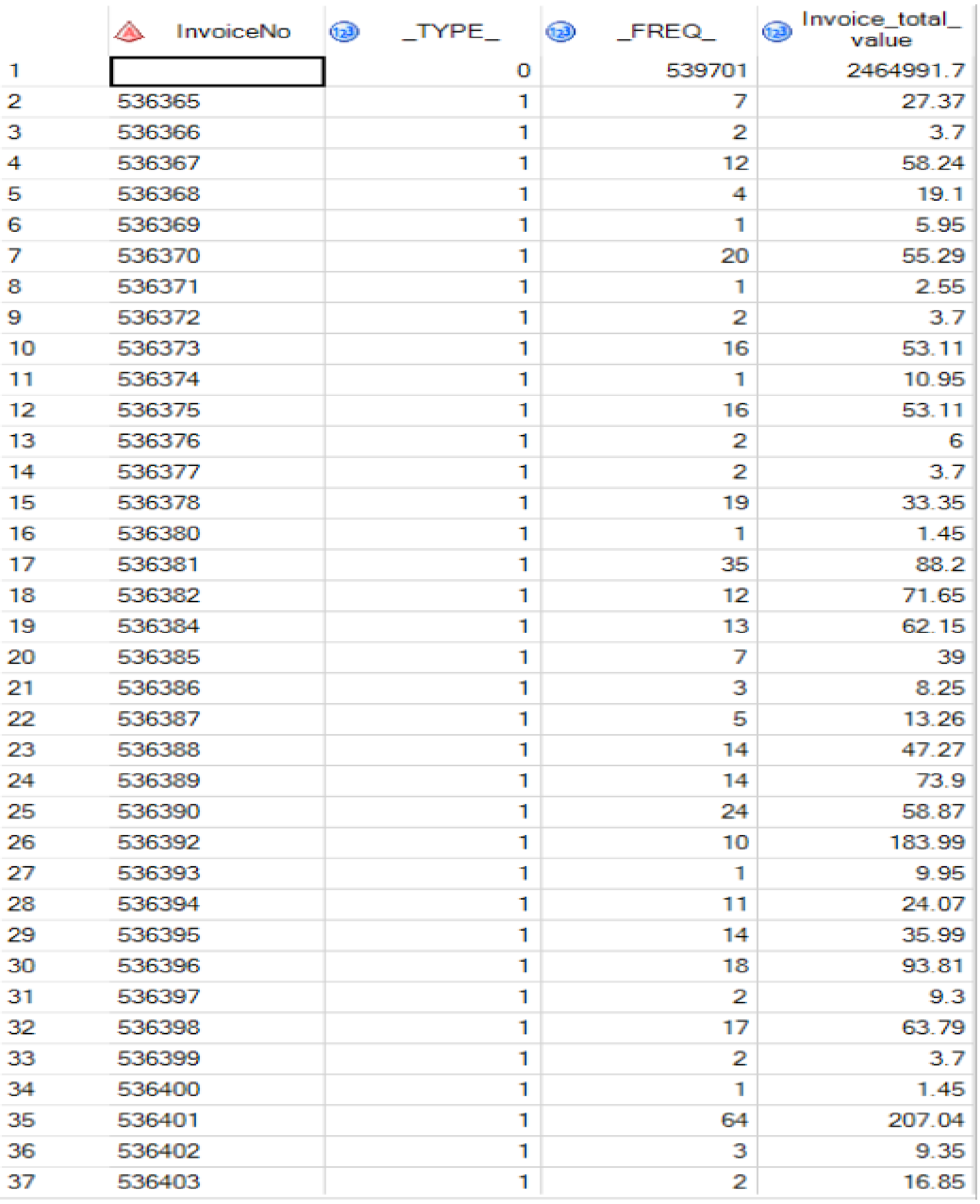
- For every invoice calculate the total number of SKU's that are related to it *'Invoice total items'*. Save the output in a new SAS data set and print the first 10 observations of it. Only the data step can be used for merging data sets. Proc sql can be used for the statistics. It is suggested to use noprint option in proc sql because the new data set will be large.



InvoiceNo	COUNT_of_SKU
573585	1114
581219	749
581492	731
580729	721
558475	705
579777	687
581217	676
537434	675
580730	662
538071	652

Figure 1 - Top 10 invoices containing most items

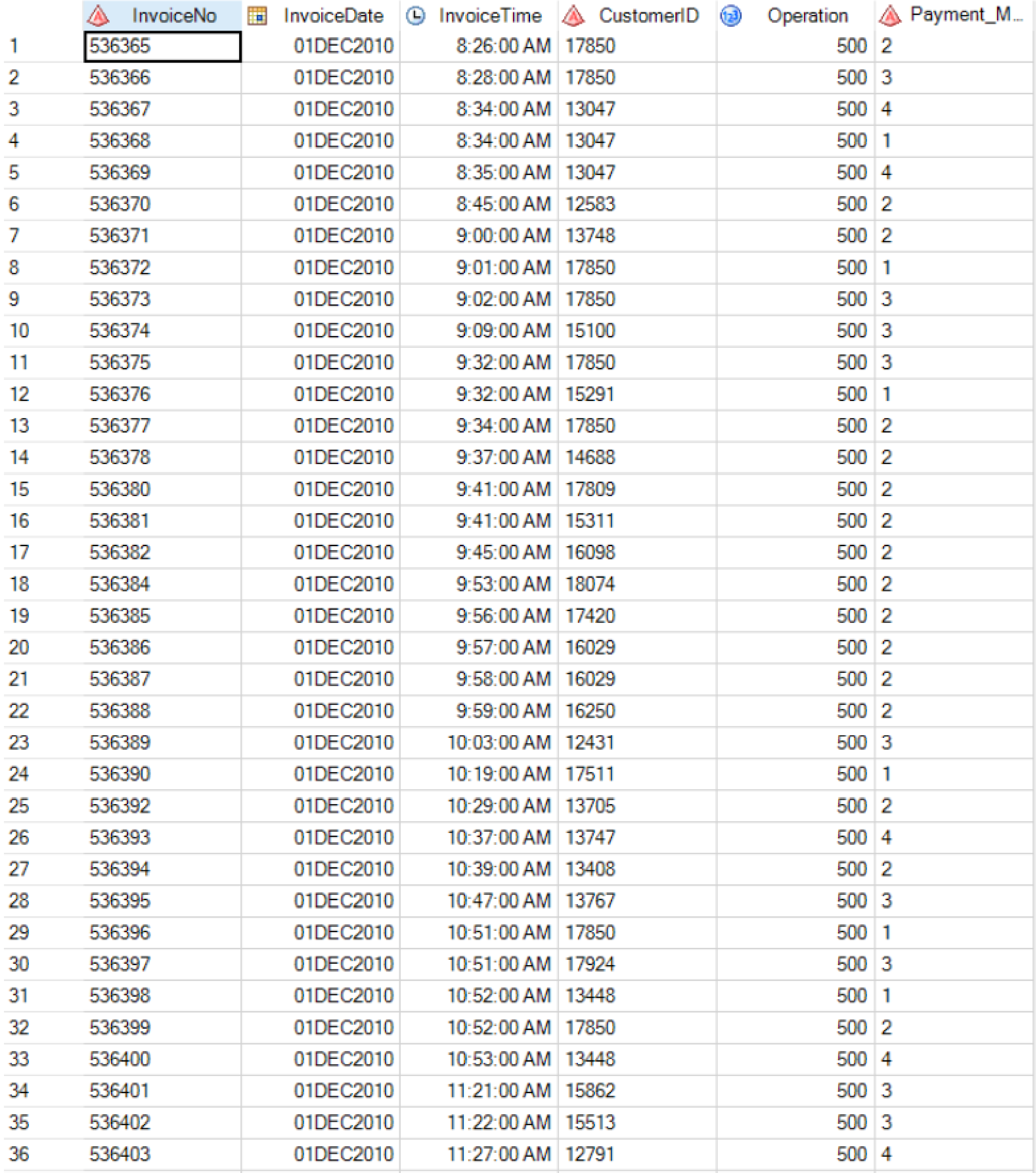
- For every invoice calculate the total value of the SKU's that are related to it *'Invoice total value*. Save the output in a new SAS data set. For this task use the proc means with the output statement.



	InvoiceNo	_TYPE_	_FREQ_	Invoice_total_value
1		0	539701	2464991.7
2	536365	1	7	27.37
3	536366	1	2	3.7
4	536367	1	12	58.24
5	536368	1	4	19.1
6	536369	1	1	5.95
7	536370	1	20	55.29
8	536371	1	1	2.55
9	536372	1	2	3.7
10	536373	1	16	53.11
11	536374	1	1	10.95
12	536375	1	16	53.11
13	536376	1	2	6
14	536377	1	2	3.7
15	536378	1	19	33.35
16	536380	1	1	1.45
17	536381	1	35	88.2
18	536382	1	12	71.65
19	536384	1	13	62.15
20	536385	1	7	39
21	536386	1	3	8.25
22	536387	1	5	13.26
23	536388	1	14	47.27
24	536389	1	14	73.9
25	536390	1	24	58.87
26	536392	1	10	183.99
27	536393	1	1	9.95
28	536394	1	11	24.07
29	536395	1	14	35.99
30	536396	1	18	93.81
31	536397	1	2	9.3
32	536398	1	17	63.79
33	536399	1	2	3.7
34	536400	1	1	1.45
35	536401	1	64	207.04
36	536402	1	3	9.35
37	536403	1	2	16.85

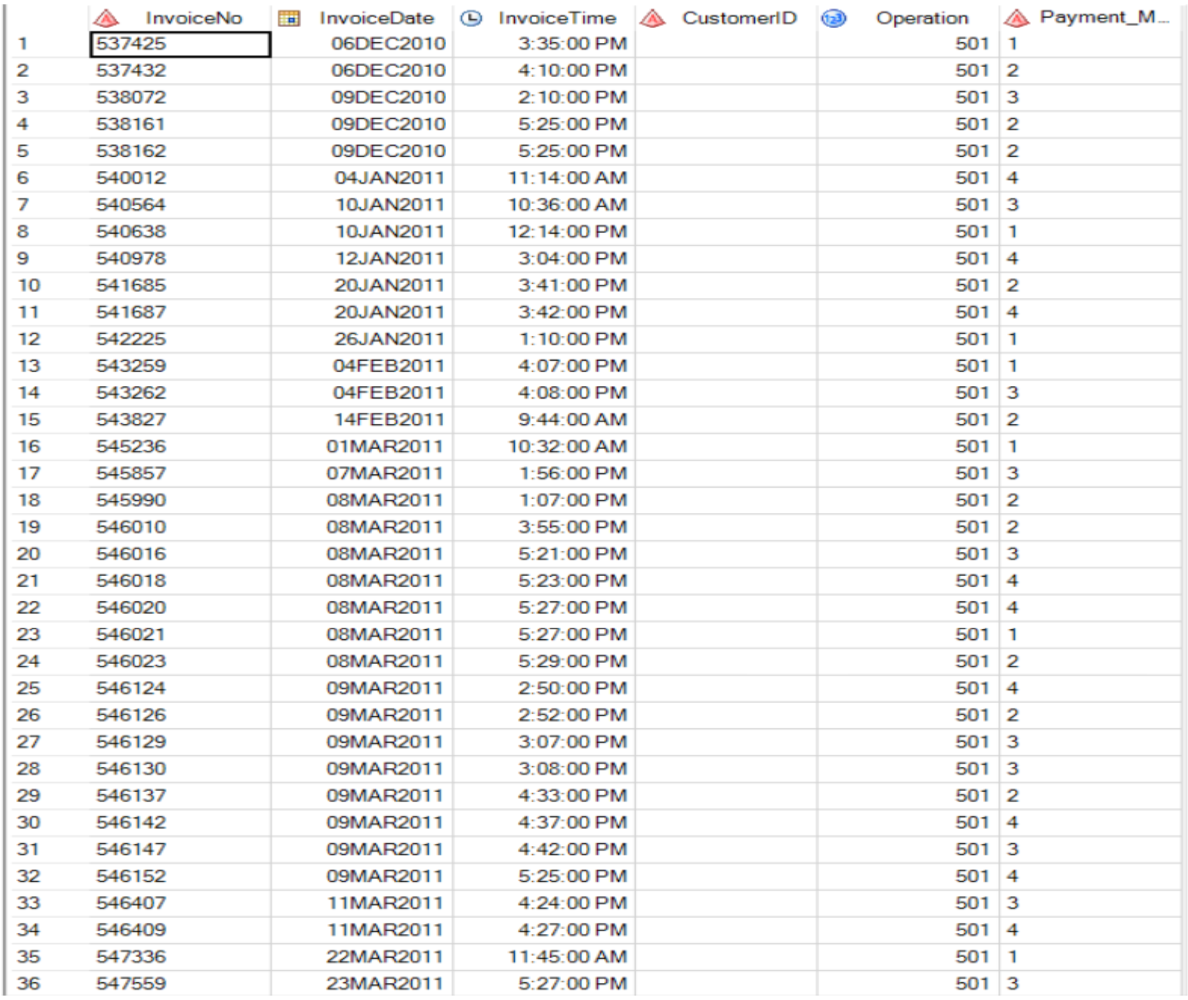
Figure 2 - Total Value Per Invoice

- Divide the observations of the table 'Invoice' into two new tables where in the one the Sales transactions will be stored where as in the second the Returns transactions will be stored. This division must be done using the variable 'Operation'.



	InvoiceNo	InvoiceDate	InvoiceTime	CustomerID	Operation	Payment_M...
1	536365	01DEC2010	8:26:00 AM	17850	500	2
2	536366	01DEC2010	8:28:00 AM	17850	500	3
3	536367	01DEC2010	8:34:00 AM	13047	500	4
4	536368	01DEC2010	8:34:00 AM	13047	500	1
5	536369	01DEC2010	8:35:00 AM	13047	500	4
6	536370	01DEC2010	8:45:00 AM	12583	500	2
7	536371	01DEC2010	9:00:00 AM	13748	500	2
8	536372	01DEC2010	9:01:00 AM	17850	500	1
9	536373	01DEC2010	9:02:00 AM	17850	500	3
10	536374	01DEC2010	9:09:00 AM	15100	500	3
11	536375	01DEC2010	9:32:00 AM	17850	500	3
12	536376	01DEC2010	9:32:00 AM	15291	500	1
13	536377	01DEC2010	9:34:00 AM	17850	500	2
14	536378	01DEC2010	9:37:00 AM	14688	500	2
15	536380	01DEC2010	9:41:00 AM	17809	500	2
16	536381	01DEC2010	9:41:00 AM	15311	500	2
17	536382	01DEC2010	9:45:00 AM	16098	500	2
18	536384	01DEC2010	9:53:00 AM	18074	500	2
19	536385	01DEC2010	9:56:00 AM	17420	500	2
20	536386	01DEC2010	9:57:00 AM	16029	500	2
21	536387	01DEC2010	9:58:00 AM	16029	500	2
22	536388	01DEC2010	9:59:00 AM	16250	500	2
23	536389	01DEC2010	10:03:00 AM	12431	500	3
24	536390	01DEC2010	10:19:00 AM	17511	500	1
25	536392	01DEC2010	10:29:00 AM	13705	500	2
26	536393	01DEC2010	10:37:00 AM	13747	500	4
27	536394	01DEC2010	10:39:00 AM	13408	500	2
28	536395	01DEC2010	10:47:00 AM	13767	500	3
29	536396	01DEC2010	10:51:00 AM	17850	500	1
30	536397	01DEC2010	10:51:00 AM	17924	500	3
31	536398	01DEC2010	10:52:00 AM	13448	500	1
32	536399	01DEC2010	10:52:00 AM	17850	500	2
33	536400	01DEC2010	10:53:00 AM	13448	500	4
34	536401	01DEC2010	11:21:00 AM	15862	500	3
35	536402	01DEC2010	11:22:00 AM	15513	500	3
36	536403	01DEC2010	11:27:00 AM	12791	500	4

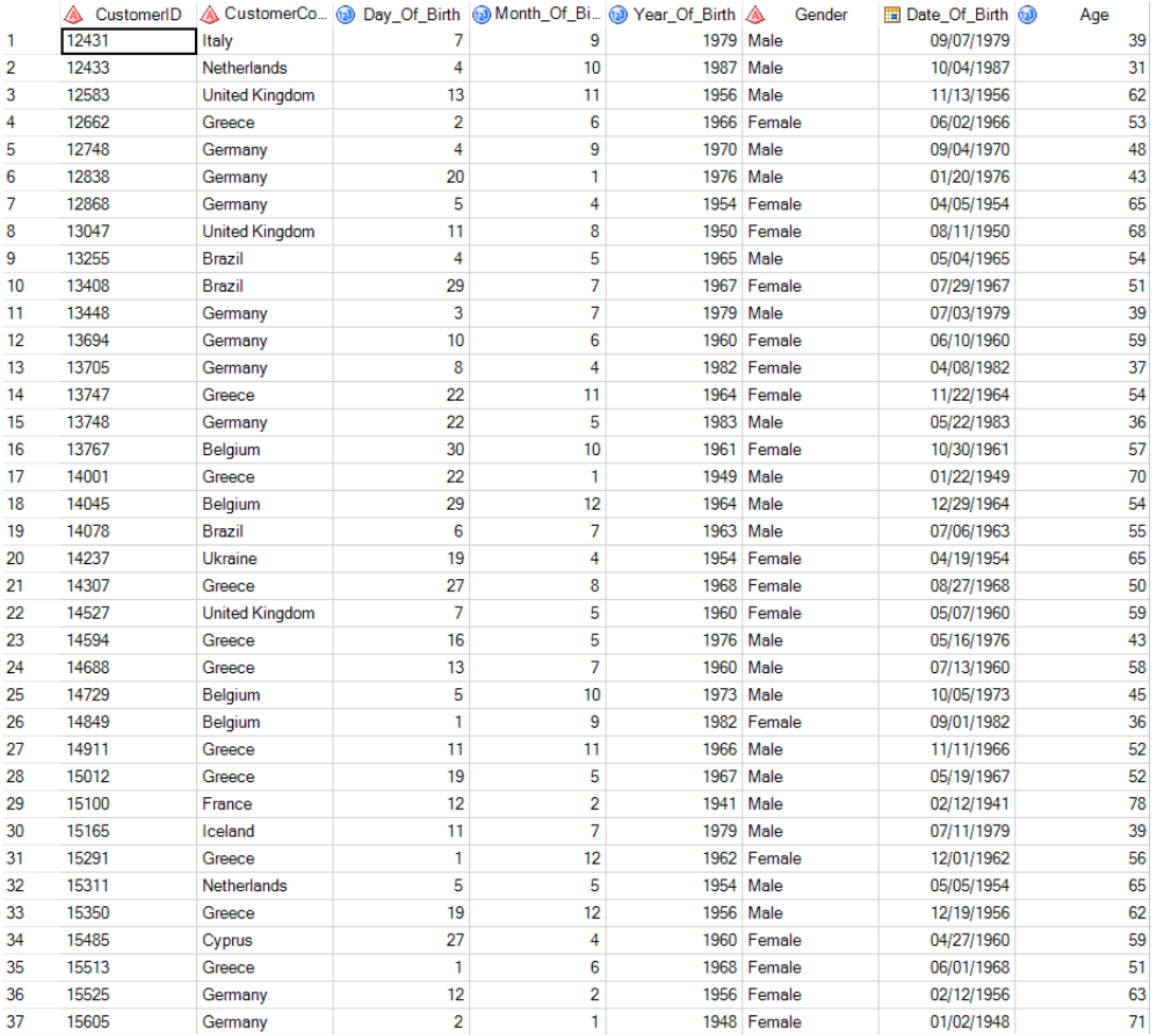
Figure 3 - Sales Transactions



	InvoiceNo	InvoiceDate	InvoiceTime	CustomerID	Operation	Payment_M...
1	537425	06DEC2010	3:35:00 PM		501	1
2	537432	06DEC2010	4:10:00 PM		501	2
3	538072	09DEC2010	2:10:00 PM		501	3
4	538161	09DEC2010	5:25:00 PM		501	2
5	538162	09DEC2010	5:25:00 PM		501	2
6	540012	04JAN2011	11:14:00 AM		501	4
7	540564	10JAN2011	10:36:00 AM		501	3
8	540638	10JAN2011	12:14:00 PM		501	1
9	540978	12JAN2011	3:04:00 PM		501	4
10	541685	20JAN2011	3:41:00 PM		501	2
11	541687	20JAN2011	3:42:00 PM		501	4
12	542225	26JAN2011	1:10:00 PM		501	1
13	543259	04FEB2011	4:07:00 PM		501	1
14	543262	04FEB2011	4:08:00 PM		501	3
15	543827	14FEB2011	9:44:00 AM		501	2
16	545236	01MAR2011	10:32:00 AM		501	1
17	545857	07MAR2011	1:56:00 PM		501	3
18	545990	08MAR2011	1:07:00 PM		501	2
19	546010	08MAR2011	3:55:00 PM		501	2
20	546016	08MAR2011	5:21:00 PM		501	3
21	546018	08MAR2011	5:23:00 PM		501	4
22	546020	08MAR2011	5:27:00 PM		501	4
23	546021	08MAR2011	5:27:00 PM		501	1
24	546023	08MAR2011	5:29:00 PM		501	2
25	546124	09MAR2011	2:50:00 PM		501	4
26	546126	09MAR2011	2:52:00 PM		501	2
27	546129	09MAR2011	3:07:00 PM		501	3
28	546130	09MAR2011	3:08:00 PM		501	3
29	546137	09MAR2011	4:33:00 PM		501	2
30	546142	09MAR2011	4:37:00 PM		501	4
31	546147	09MAR2011	4:42:00 PM		501	3
32	546152	09MAR2011	5:25:00 PM		501	4
33	546407	11MAR2011	4:24:00 PM		501	3
34	546409	11MAR2011	4:27:00 PM		501	4
35	547336	22MAR2011	11:45:00 AM		501	1
36	547559	23MAR2011	5:27:00 PM		501	3

Figure 4 - Return Transactions

- Create a new table that will contain customers for which there exists a birth date (no one of the fields Day_Of_Birth, Month_Of_Birth, Year_Of_Birth should be NULL). Then calculate the customer’s age based on the fact that today’s date is 01/01/2019 and store it into a new variable (check the validity of the dates e.g. birth year less than 1920).

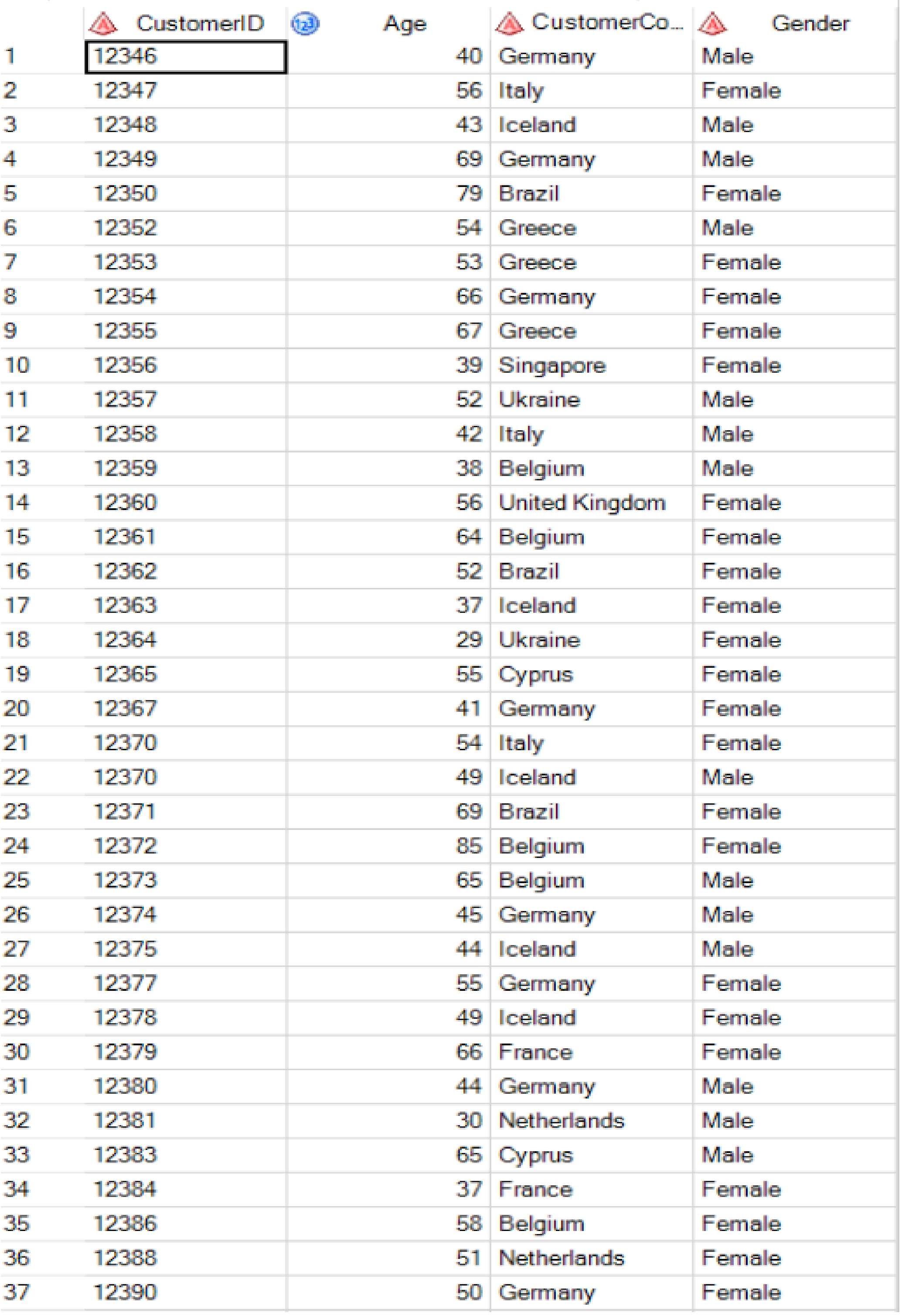


	CustomerID	CustomerCo...	Day_Of_Birth	Month_Of_Bi...	Year_Of_Birth	Gender	Date_Of_Birth	Age
1	12431	Italy	7	9	1979	Male	09/07/1979	39
2	12433	Netherlands	4	10	1987	Male	10/04/1987	31
3	12583	United Kingdom	13	11	1956	Male	11/13/1956	62
4	12662	Greece	2	6	1966	Female	06/02/1966	53
5	12748	Germany	4	9	1970	Male	09/04/1970	48
6	12838	Germany	20	1	1976	Male	01/20/1976	43
7	12868	Germany	5	4	1954	Female	04/05/1954	65
8	13047	United Kingdom	11	8	1950	Female	08/11/1950	68
9	13255	Brazil	4	5	1965	Male	05/04/1965	54
10	13408	Brazil	29	7	1967	Female	07/29/1967	51
11	13448	Germany	3	7	1979	Male	07/03/1979	39
12	13694	Germany	10	6	1960	Female	06/10/1960	59
13	13705	Germany	8	4	1982	Female	04/08/1982	37
14	13747	Greece	22	11	1964	Female	11/22/1964	54
15	13748	Germany	22	5	1983	Male	05/22/1983	36
16	13767	Belgium	30	10	1961	Female	10/30/1961	57
17	14001	Greece	22	1	1949	Male	01/22/1949	70
18	14045	Belgium	29	12	1964	Male	12/29/1964	54
19	14078	Brazil	6	7	1963	Male	07/06/1963	55
20	14237	Ukraine	19	4	1954	Female	04/19/1954	65
21	14307	Greece	27	8	1968	Female	08/27/1968	50
22	14527	United Kingdom	7	5	1960	Female	05/07/1960	59
23	14594	Greece	16	5	1976	Male	05/16/1976	43
24	14688	Greece	13	7	1960	Male	07/13/1960	58
25	14729	Belgium	5	10	1973	Male	10/05/1973	45
26	14849	Belgium	1	9	1982	Female	09/01/1982	36
27	14911	Greece	11	11	1966	Male	11/11/1966	52
28	15012	Greece	19	5	1967	Male	05/19/1967	52
29	15100	France	12	2	1941	Male	02/12/1941	78
30	15165	Iceland	11	7	1979	Male	07/11/1979	39
31	15291	Greece	1	12	1962	Female	12/01/1962	56
32	15311	Netherlands	5	5	1954	Male	05/05/1954	65
33	15350	Greece	19	12	1956	Male	12/19/1956	62
34	15485	Cyprus	27	4	1960	Female	04/27/1960	59
35	15513	Greece	1	6	1968	Female	06/01/1968	51
36	15525	Germany	12	2	1956	Female	02/12/1956	63
37	15605	Germany	2	1	1948	Female	01/02/1948	71

Figure 5 - Table containing Age Of Customers

2. Describe and explain using graphs who is your customer. What is the profile of the audience to which the company's products are targeted?

- What are the demographic characteristics i.e. age, gender and country of the company's customers?



	CustomerID	Age	CustomerCo...	Gender
1	12346	40	Germany	Male
2	12347	56	Italy	Female
3	12348	43	Iceland	Male
4	12349	69	Germany	Male
5	12350	79	Brazil	Female
6	12352	54	Greece	Male
7	12353	53	Greece	Female
8	12354	66	Germany	Female
9	12355	67	Greece	Female
10	12356	39	Singapore	Female
11	12357	52	Ukraine	Male
12	12358	42	Italy	Male
13	12359	38	Belgium	Male
14	12360	56	United Kingdom	Female
15	12361	64	Belgium	Female
16	12362	52	Brazil	Female
17	12363	37	Iceland	Female
18	12364	29	Ukraine	Female
19	12365	55	Cyprus	Female
20	12367	41	Germany	Female
21	12370	54	Italy	Female
22	12370	49	Iceland	Male
23	12371	69	Brazil	Female
24	12372	85	Belgium	Female
25	12373	65	Belgium	Male
26	12374	45	Germany	Male
27	12375	44	Iceland	Male
28	12377	55	Germany	Female
29	12378	49	Iceland	Female
30	12379	66	France	Female
31	12380	44	Germany	Male
32	12381	30	Netherlands	Male
33	12383	65	Cyprus	Male
34	12384	37	France	Female
35	12386	58	Belgium	Female
36	12388	51	Netherlands	Female
37	12390	50	Germany	Female

Figure 6 - Demographic Characteristics Of Customers

- Based on the age variable, create a new variable entitled Age_Range that takes the following values:

<18 -- > “Under 18”

18 - 25 -- > “Very Young”

26 - 35 -- > “Young”

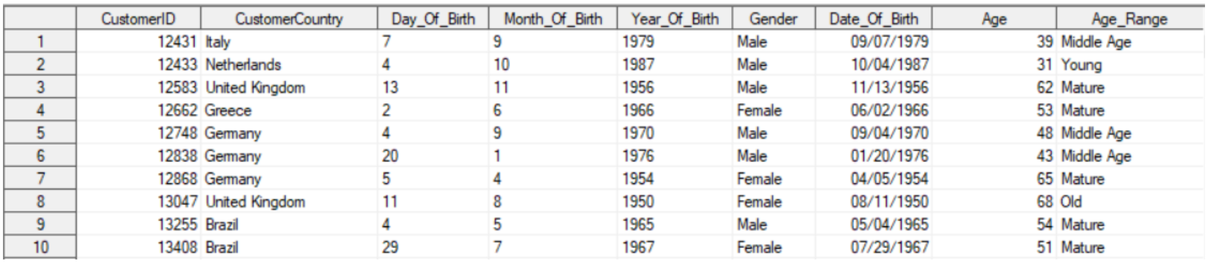
36 - 50 -- > “Middle Age”

51 - 65 -- > “Mature”

66 – 75 -- > “Old”

>= 76 -- > “Very Old”

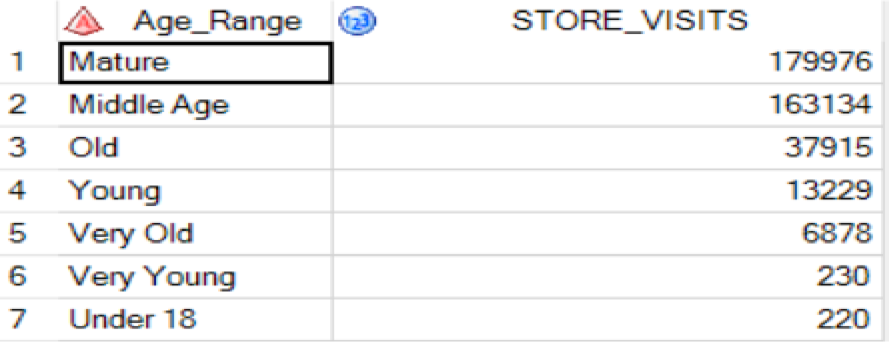
(Attention: do not format the values of the existing variable but create a new variable entitled Age _Range).



	CustomerID	CustomerCountry	Day_Of_Birth	Month_Of_Birth	Year_Of_Birth	Gender	Date_Of_Birth	Age	Age_Range
1	12431	Italy	7	9	1979	Male	09/07/1979	39	Middle Age
2	12433	Netherlands	4	10	1987	Male	10/04/1987	31	Young
3	12583	United Kingdom	13	11	1956	Male	11/13/1956	62	Mature
4	12662	Greece	2	6	1966	Female	06/02/1966	53	Mature
5	12748	Germany	4	9	1970	Male	09/04/1970	48	Middle Age
6	12838	Germany	20	1	1976	Male	01/20/1976	43	Middle Age
7	12868	Germany	5	4	1954	Female	04/05/1954	65	Mature
8	13047	United Kingdom	11	8	1950	Female	08/11/1950	68	Old
9	13255	Brazil	4	5	1965	Male	05/04/1965	54	Mature
10	13408	Brazil	29	7	1967	Female	07/29/1967	51	Mature

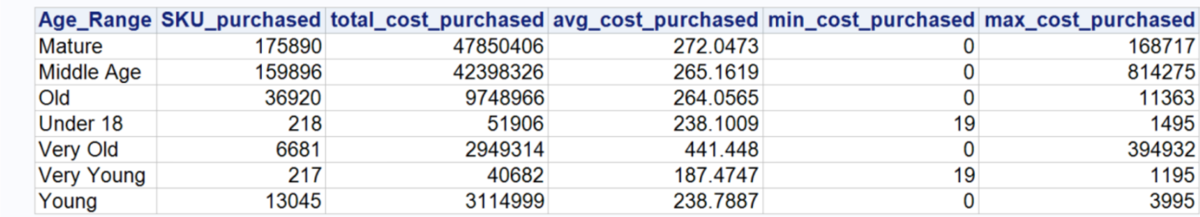
Figure 7 - Table Containing Age Range

- What are the behavioral characteristics of each age group? (visits to the stores, number of SKU’s purchased, total cost of purchases, average cost, minimum cost, maximum cost etc). The merging of the data sets must be done using exclusively the data step but the calculation of the statistics e.g. visits, total cost of purchases etc can be done using proc sql. Create a pie chart and a frequency table with the percentages of customers that belong to each age group. Augment your analysis by providing pie charts for the behavioral characteristics for each age group.



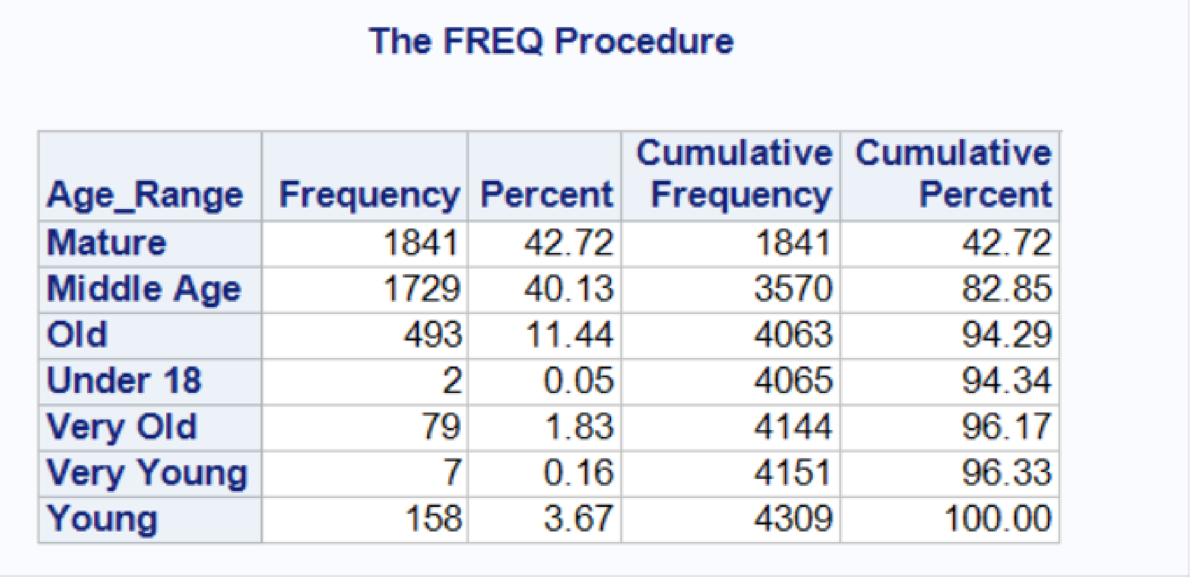
	Age_Range	STORE_VISITS
1	Mature	179976
2	Middle Age	163134
3	Old	37915
4	Young	13229
5	Very Old	6878
6	Very Young	230
7	Under 18	220

Figure 8 - Store Visits Per Age Group



Age_Range	SKU_purchased	total_cost_purchased	avg_cost_purchased	min_cost_purchased	max_cost_purchased
Mature	175890	47850406	272.0473	0	168717
Middle Age	159896	42398326	265.1619	0	814275
Old	36920	9748966	264.0565	0	11363
Under 18	218	51906	238.1009	19	1495
Very Old	6681	2949314	441.448	0	394932
Very Young	217	40682	187.4747	19	1195
Young	13045	3114999	238.7887	0	3995

Figure 9 - Cost Measures of Purchased Items Per Age Group



The FREQ Procedure

Age_Range	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Mature	1841	42.72	1841	42.72
Middle Age	1729	40.13	3570	82.85
Old	493	11.44	4063	94.29
Under 18	2	0.05	4065	94.34
Very Old	79	1.83	4144	96.17
Very Young	7	0.16	4151	96.33
Young	158	3.67	4309	100.00

Figure 10 – Customers Per Age Group

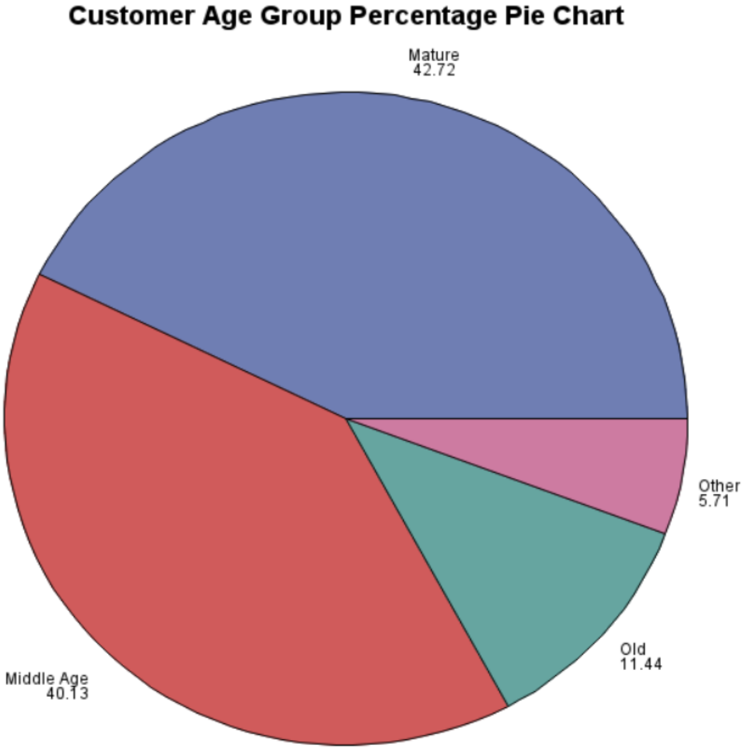


Figure 11 - Customers Per Age Group

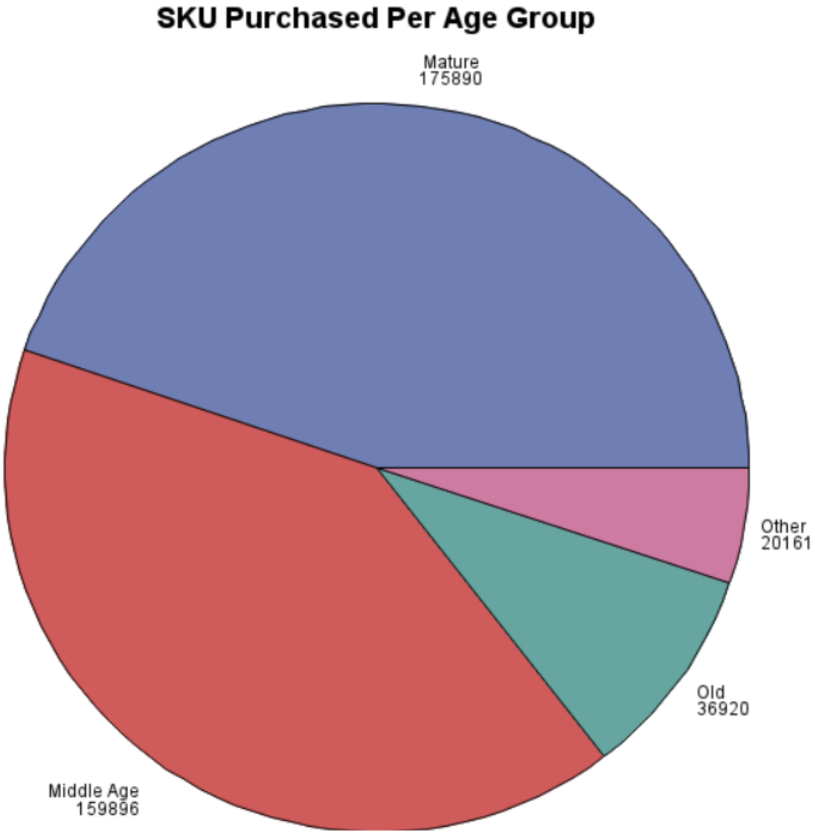


Figure 12 - Items Purchased Per Age Group

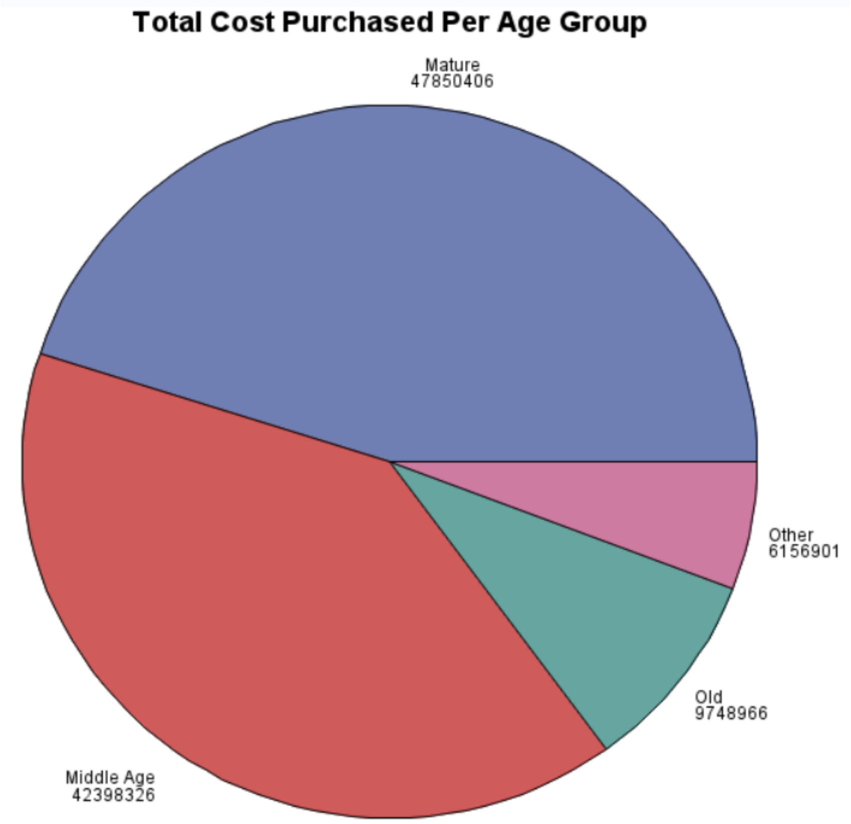


Figure 13 - Total Cost of Items Purchased Per Age Group

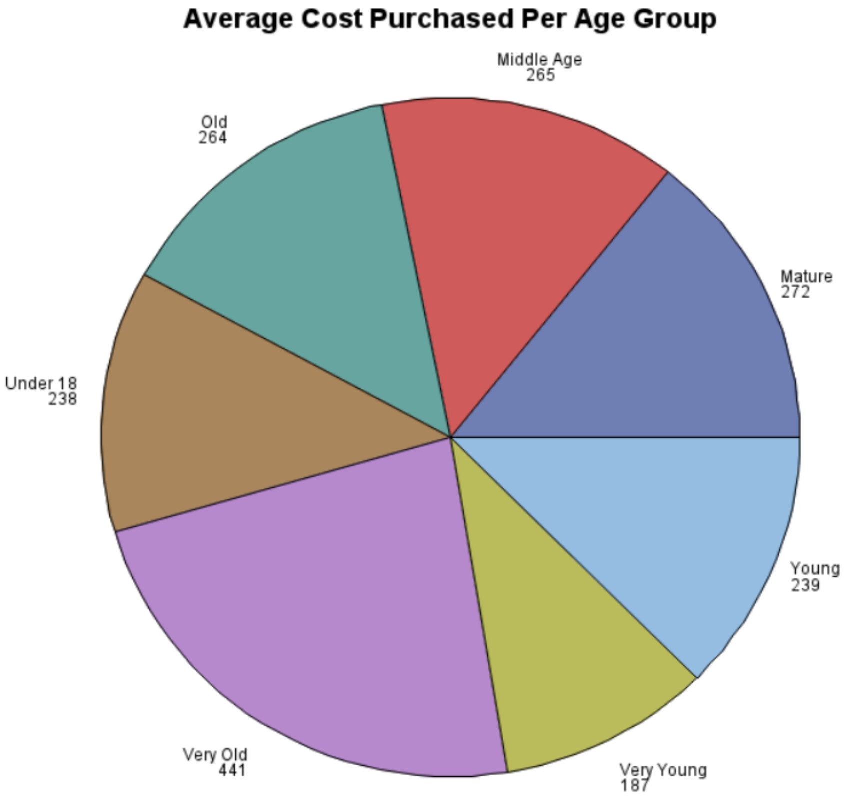


Figure 14 - Average Cost of Items Purchased Per Age Group

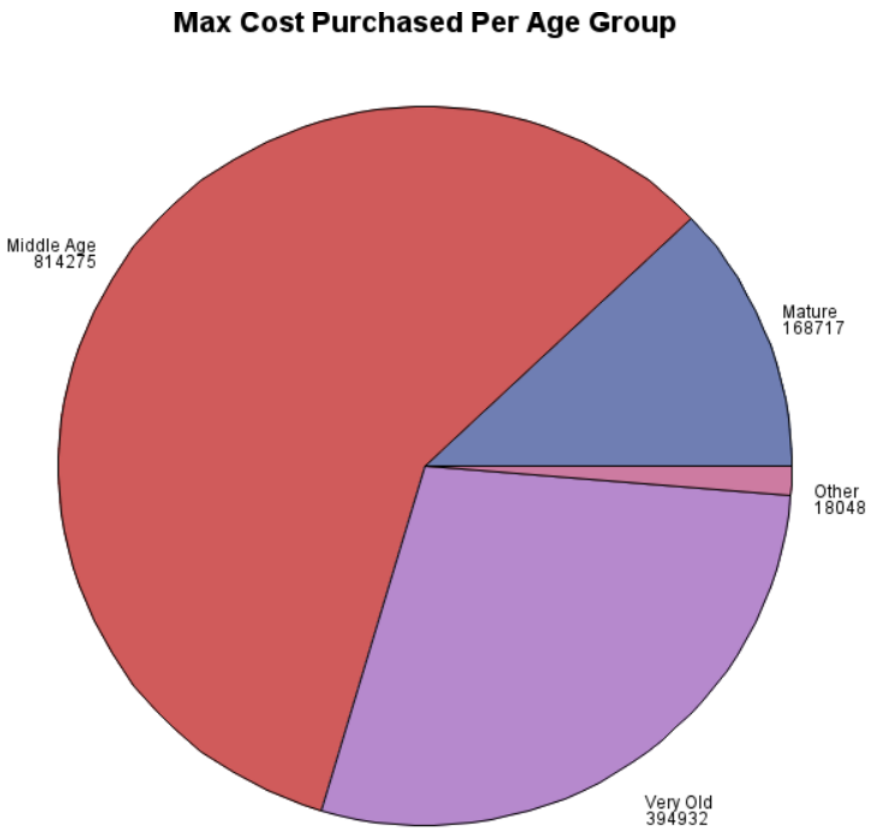


Figure 15 - Max Cost of Items Purchased Per Age Group

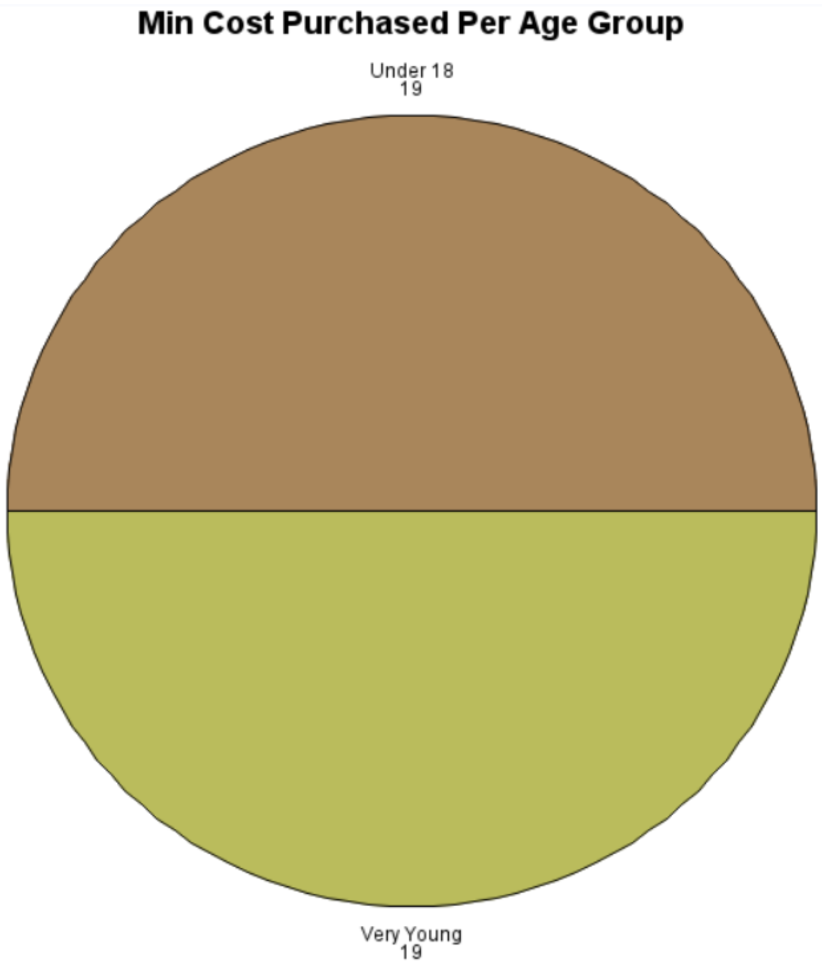


Figure 16 - Min Cost of Items Purchased Per Age Group

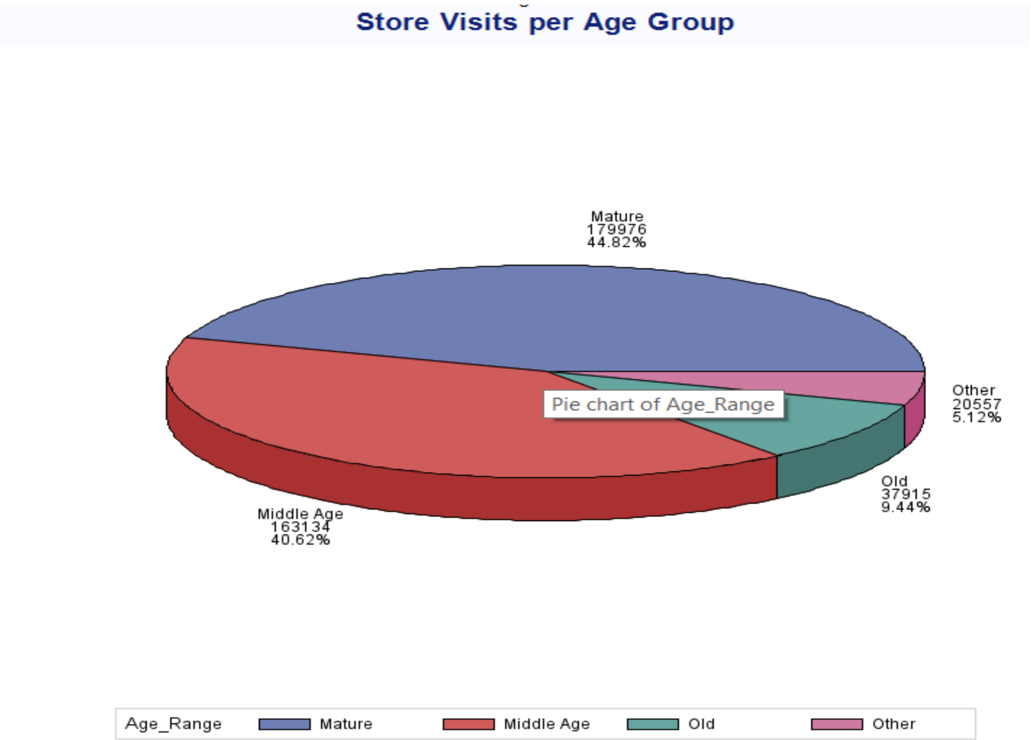


Figure 17 - Store Visits Per Age Group

From Figure 10 and Figure 11, it is clear that most of the customers are mature whereas the fewest are under 18.
From Figure 9 and Figure 12 – Figure 16, we observe that mature customers have purchased the most items with the biggest total and average cost. However, the maximum cost has been made by middle Age ‘whereas the biggest min cost has been made byvery young customers and customers under 18. This may occur due to outliers.
Based on Figure 8 and Figure 17, we consider that the most visits have been made by middle-aged people while the fewest by minors.

3. Exploration and understanding of sales:

- What was the level of Sales and Returns? The variable ‘Operation´ of the 'Invoice table takes the values '500 and 501'. Create a bar chart with the monetary values and a frequency table by creating a custom format for which 500=Sale and 501=Cancellation.

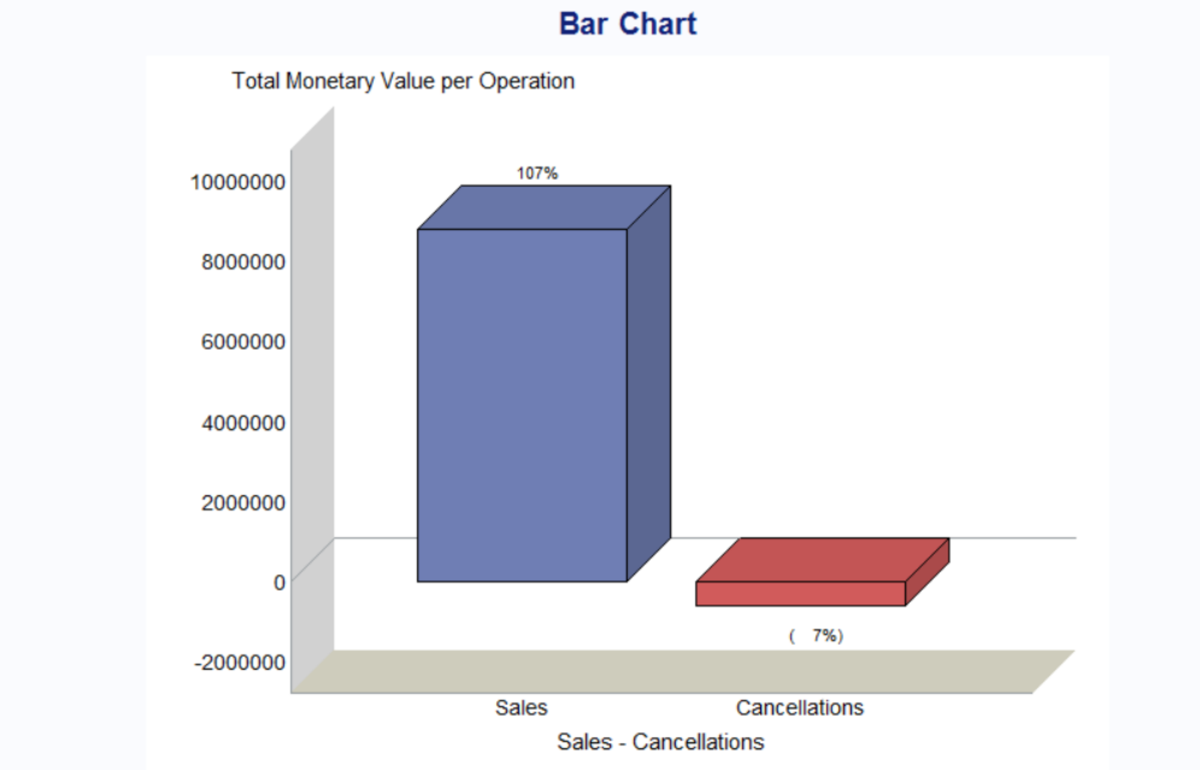
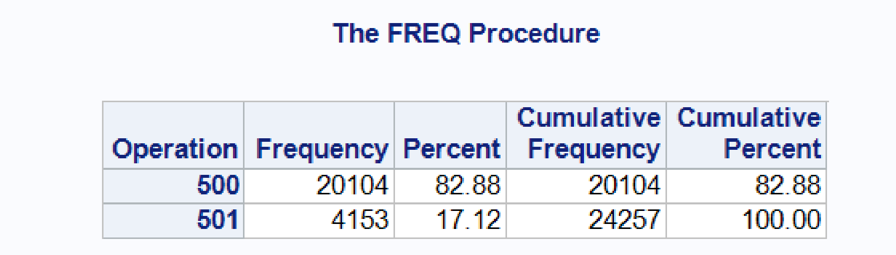


Figure 18 - Total Monetary Value per Operation



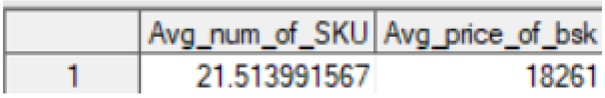
The FREQ Procedure

Operation	Frequency	Percent	Cumulative Frequency	Cumulative Percent
500	20104	82.88	20104	82.88
501	4153	17.12	24257	100.00

Figure 19 - Frequency Table

From figures 18 and 19,it seems that in comparison with the sales, there are few cancellations and due to that the total monetary value of them is low.

Create graphs for the average basket size i.e. number of SKU's, total monetary value, etc and comment on your findings. Proc sql can be used only for the calculation of the statistics e.g. of the average basket and not e.g. for merging data sets (for this data step should be used).



	Avg_num_of_SKU	Avg_price_of_bsk
1	21.513991567	18261

Figure 20 - Average Basket Size

The average basket has 22 items and costs 18261.

4. Zoom in further to the sales transactions and describe the way that the customers pay for their purchases (cash, credit card, pay pal, debit card) and identify their preferences concerning the origin of the products they buy (irrespective of their age group). We make the assumption that an invoice can be paid by more than one payment method.

   - Create pie charts for the variable 'Payment Method' for every value of the variable 'Movement' of the 'Basket' table. In the graphs show the type of the payment method and not its code e.g. credit card 20%.

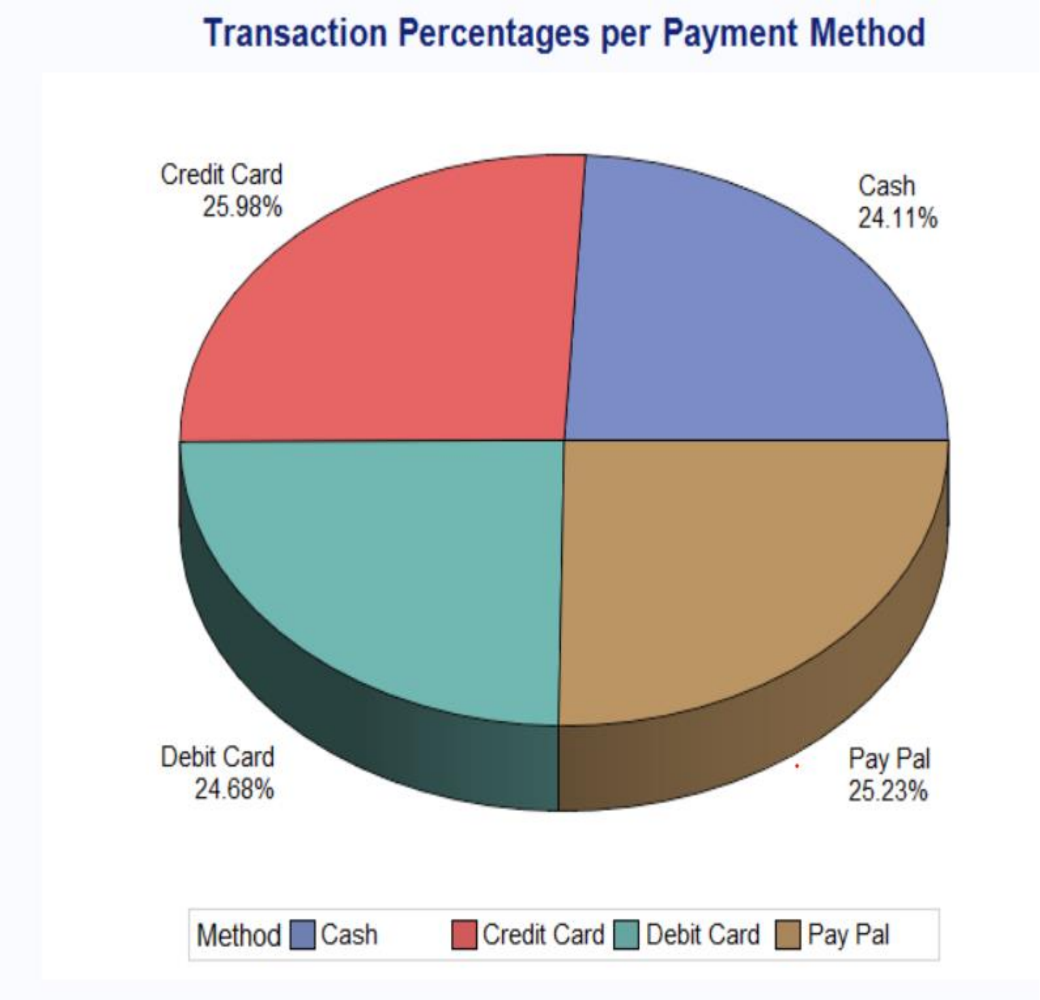


Figure 21 - Frequency Per Payment Method

   - Which payment method is the most popular (it is used most by the customers)? Which payment method brings the biggest revenues for the company? (Use graphs).

From figures 21 and 22 it appears that customers slightly prefer debit card payments and due to that the revenue is a bit higher than other payment methods.

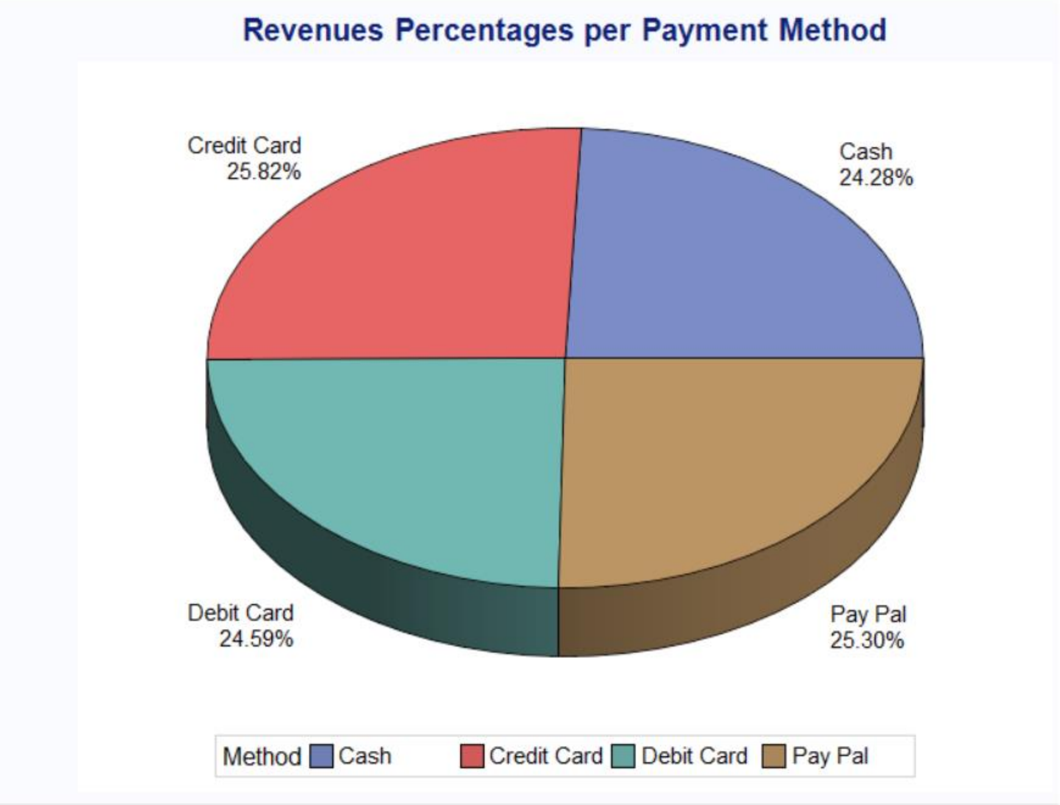


Figure 22 - Revenues Per Payment Method

- Products of what origin do the customers prefer (use graphs)? Products of which origin bring the biggest revenues to the retailer and what is the amount of the revenues for each origin (use graphs and tables).

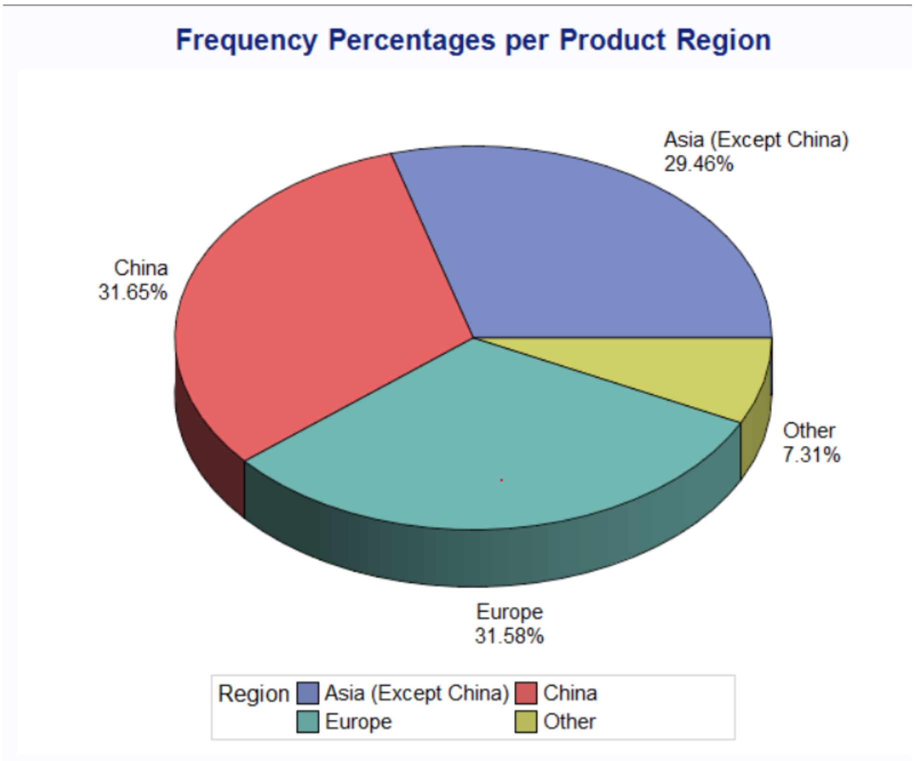


Figure 23 - Frequency Of Products Purchased Per Product Origin

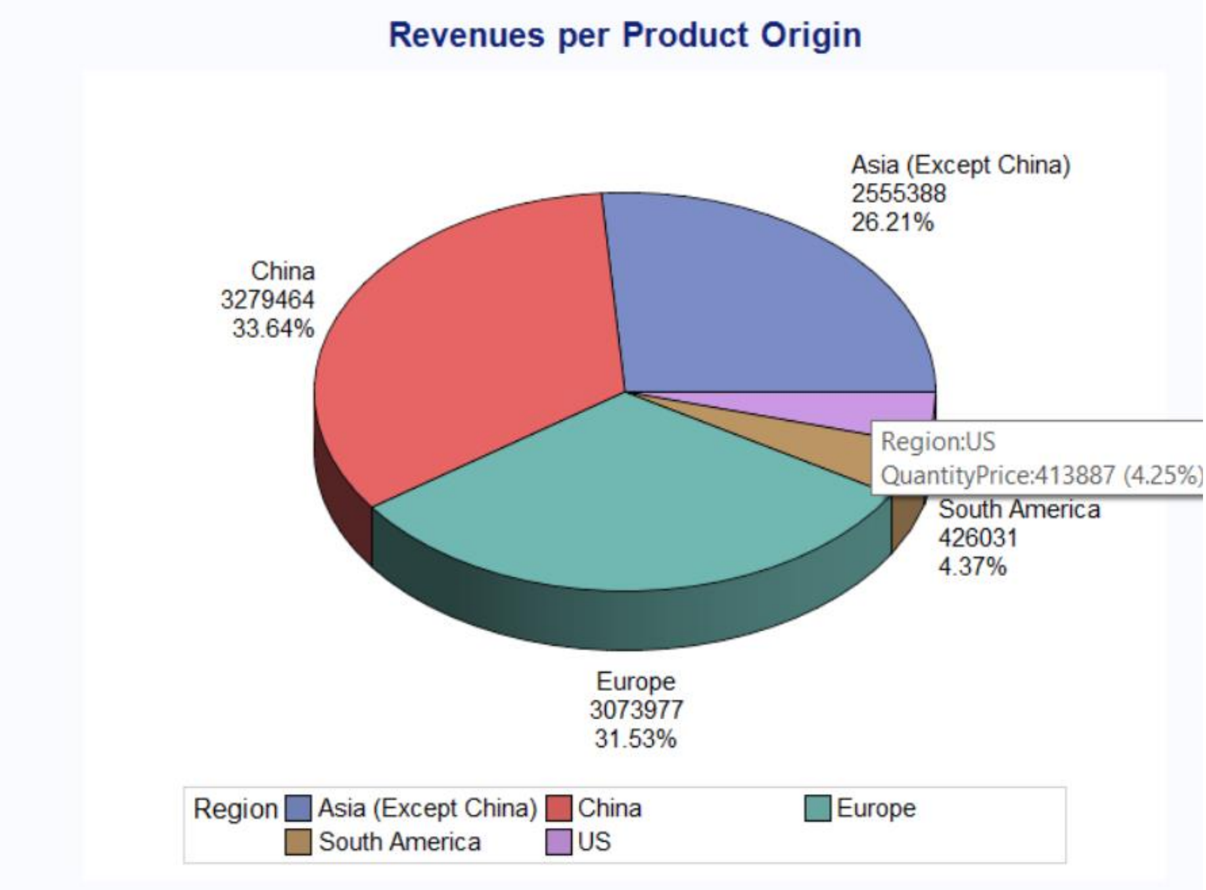
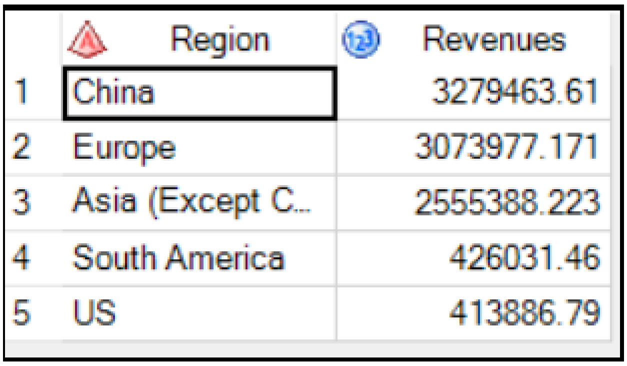


Figure 24 - Revenues Of Products Purchased Per Product Origin



	Region	Revenues
1	China	3279463.61
2	Europe	3073977.171
3	Asia (Except C...	2555388.223
4	South America	426031.46
5	US	413886.79

Figure 25 - Table Of Revenues Per Product Origin

From Figures 23, 24 and 25 , it seems that most of the purchased products are made in China and these products bring the biggest revenues.

5. It should be mentioned that the SKU of each product contains "hidden" information. The twelfth (12th) digit indicates the promotional activity that is attached to the product. In order to unhide this piece of information use relevant functions and then store it to a new column. If we assume that an SKU is 58720443050301, then the promotional activity code is 3.

- What is the percentage of products that are sold without promotion and what is the percentage of products sold with promotion (use graphs).

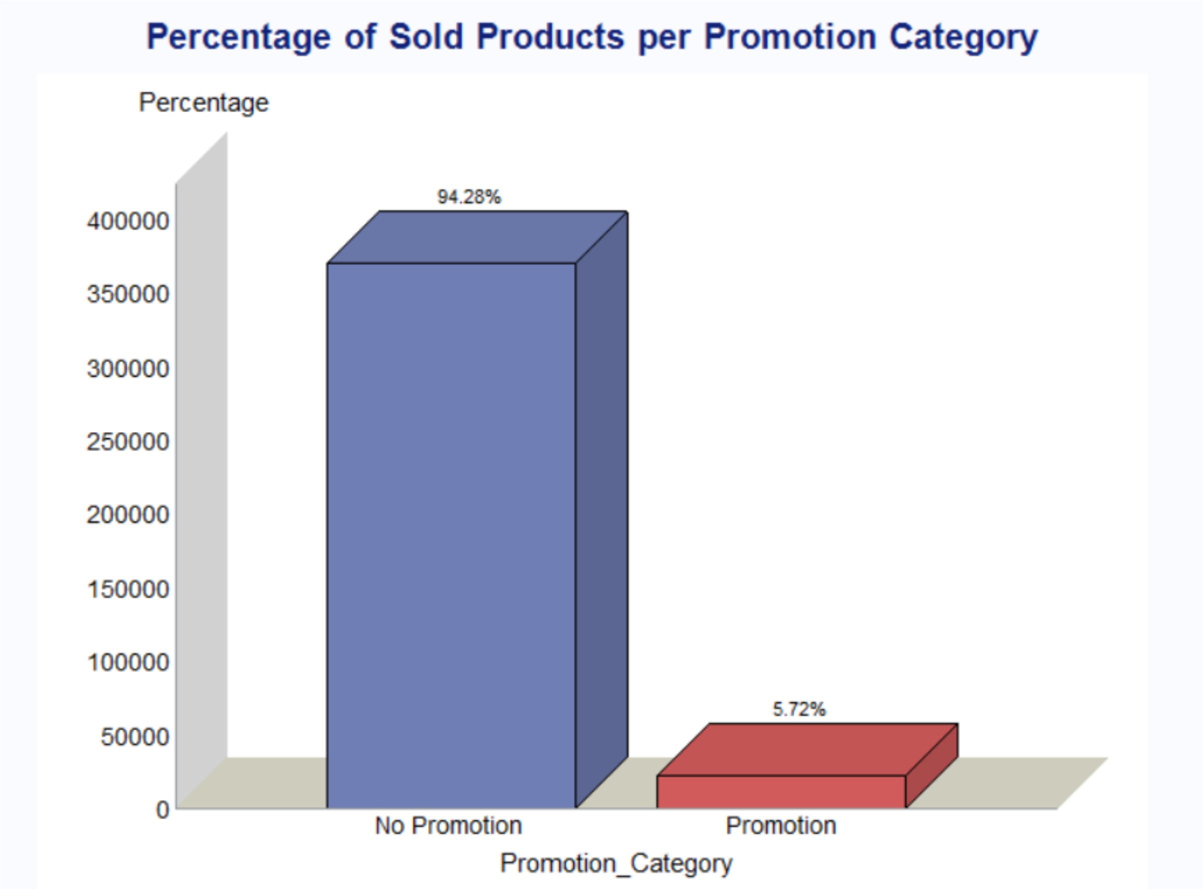


Figure 26 - Frequency of Purchased Products Per Promotion Category

The percentage of products sold without promotion is equal to 94.28 % whereas the percentage of products sold with promotion is equal to 5.72 % .

- Create pie charts to show the percentage of products that are sold on each promotion type (use the description of the promotion and not its code). Do not include the products sold without promotion.

From Figure 27, we conclude that when a product has promotion , there is much likelihood of having a 15% discount.

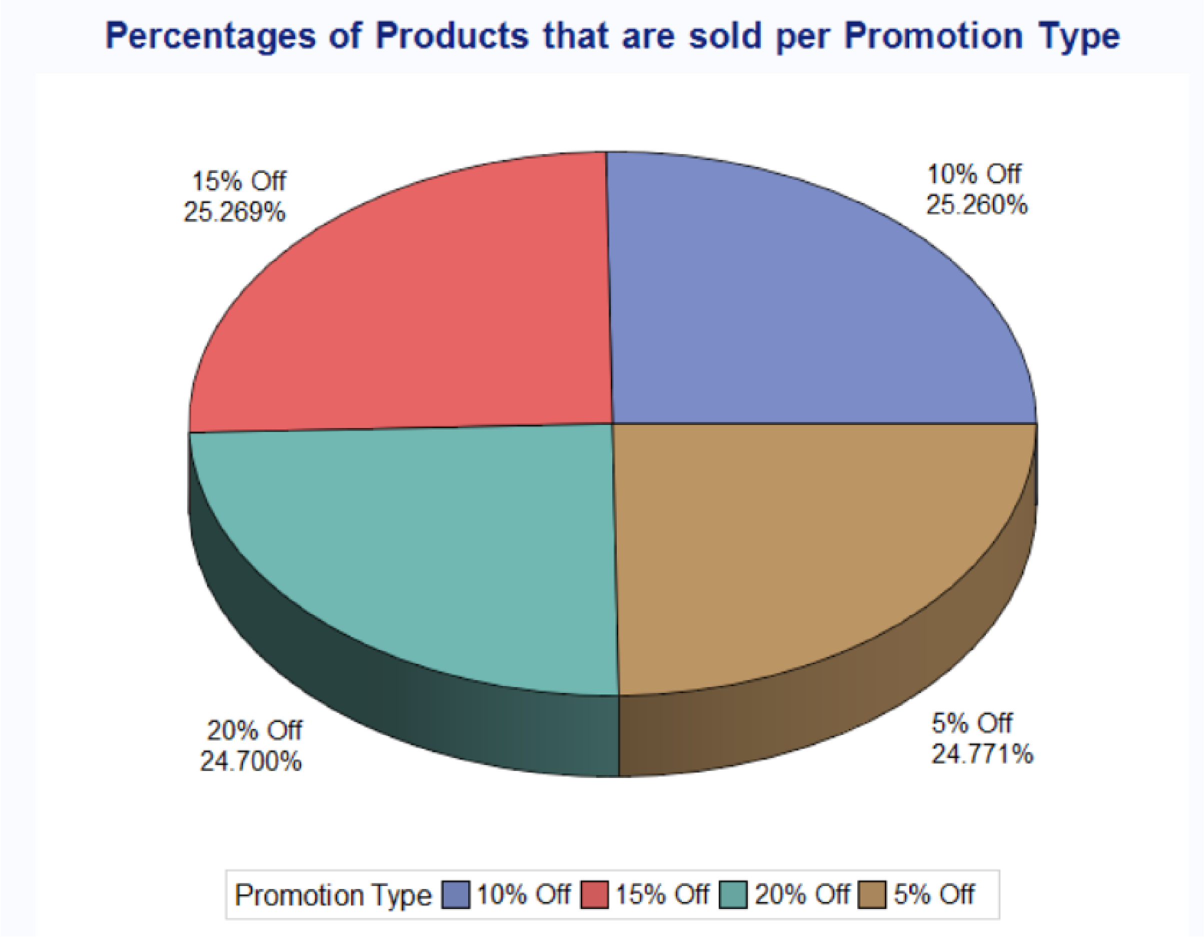
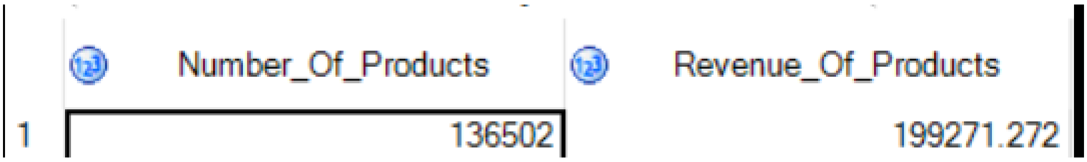


Figure 27 - Percentages of purchased Products Per Promotion Type

- How many products are sold with promotion over or equal to 15%? What is the revenue of the sale of these products?

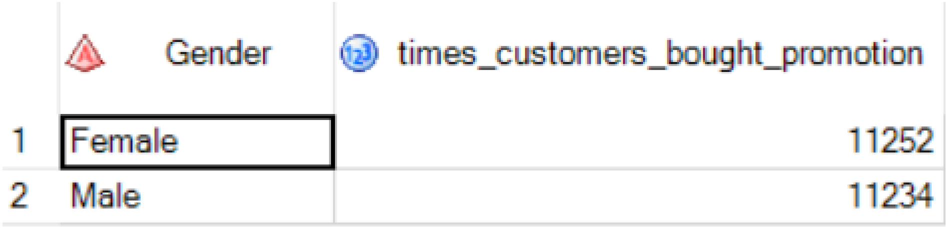


	Number_Of_Products	Revenue_Of_Products
1	136502	199271.272

Figure 28 - Frequency and Revenues of Products With more than 15 % discount

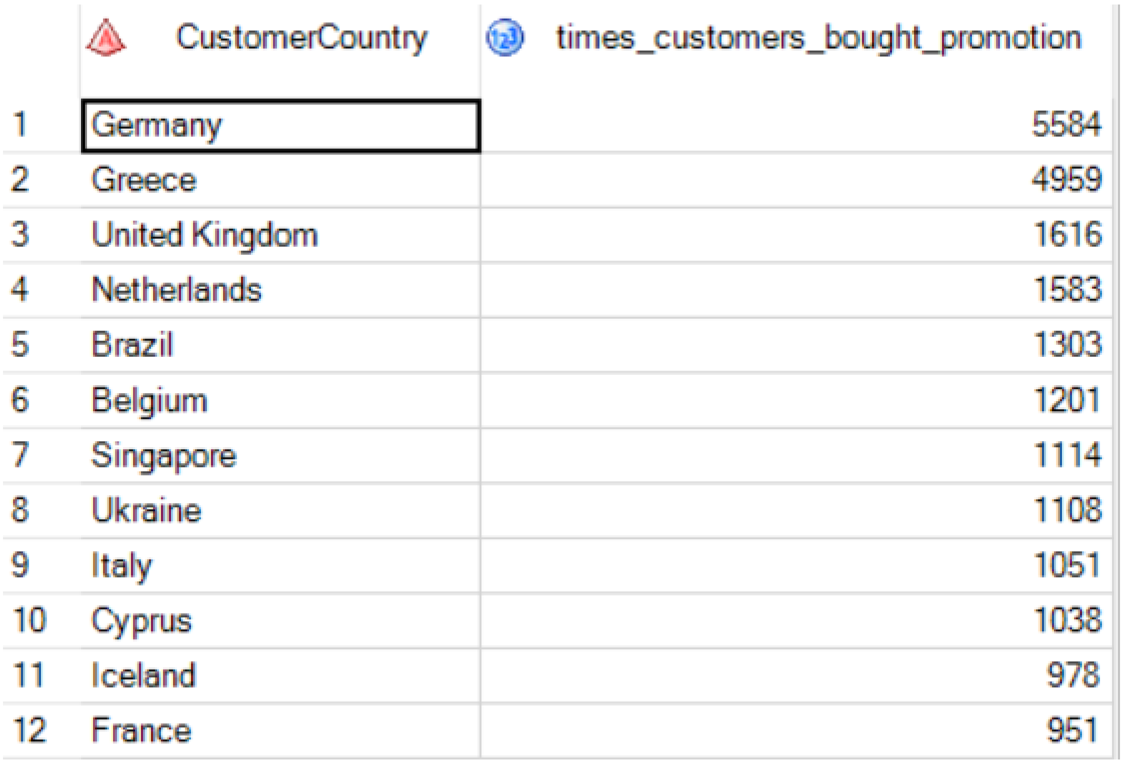
Based on Figure 28, 136502 products are sold with more than 15% discount with total revenue equal to 199271.

- Which customers buy more times products that are on promotion? Provide their demographic characteristics (i.e. gender, age group, country).



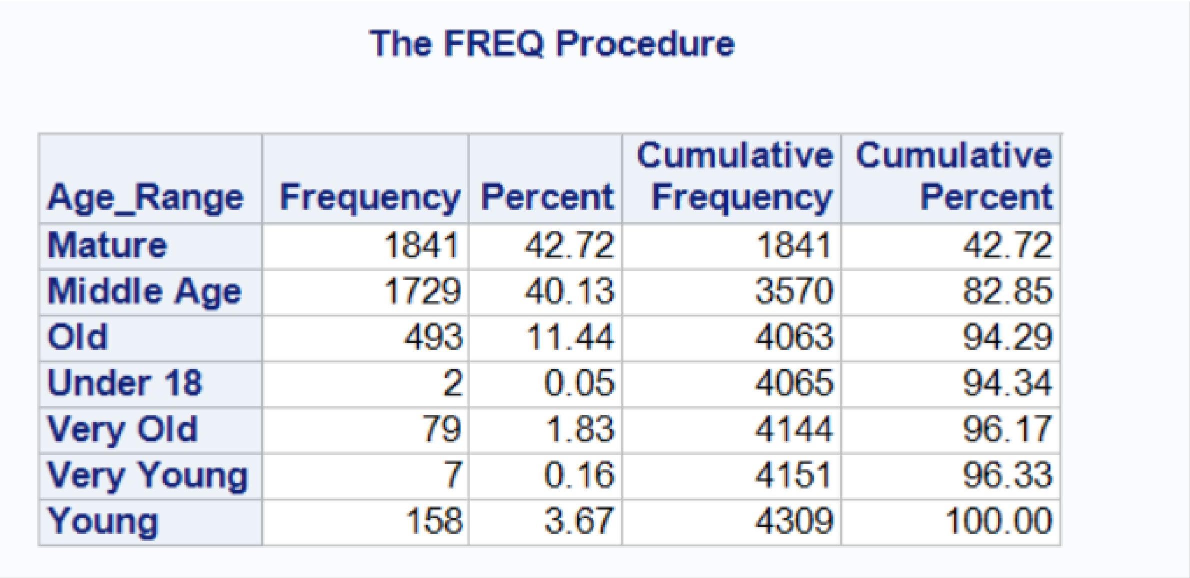
	Gender	times_customers_bought_promotion
1	Female	11252
2	Male	11234

Figure 29 - Frequency Of Sold Products per Gender



	CustomerCountry	times_customers_bought_promotion
1	Germany	5584
2	Greece	4959
3	United Kingdom	1616
4	Netherlands	1583
5	Brazil	1303
6	Belgium	1201
7	Singapore	1114
8	Ukraine	1108
9	Italy	1051
10	Cyprus	1038
11	Iceland	978
12	France	951

Figure 30 - Frequency Of Sold Products per Customer's Country



The FREQ Procedure

Age_Range	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Mature	1841	42.72	1841	42.72
Middle Age	1729	40.13	3570	82.85
Old	493	11.44	4063	94.29
Under 18	2	0.05	4065	94.34
Very Old	79	1.83	4144	96.17
Very Young	7	0.16	4151	96.33
Young	158	3.67	4309	100.00

Figure 31 - Frequency of Sold Products per Age Group

From Figures 29, 30 and 31 , we conclude that Mature German women buy most products at a discount.

6. It should be also mentioned that the SKU of each product contains more “hidden” information. The sixth (6th) digit indicates the company that supplied the product (supplier). In order to unhide this piece of information use relevant functions and then store it to a new column. If we assume that an SKU is 58720443050301, then the supplier code is 4.

- Create graphs to show the percentage of products sold by each supplier (use the name of the supplier and not its code).

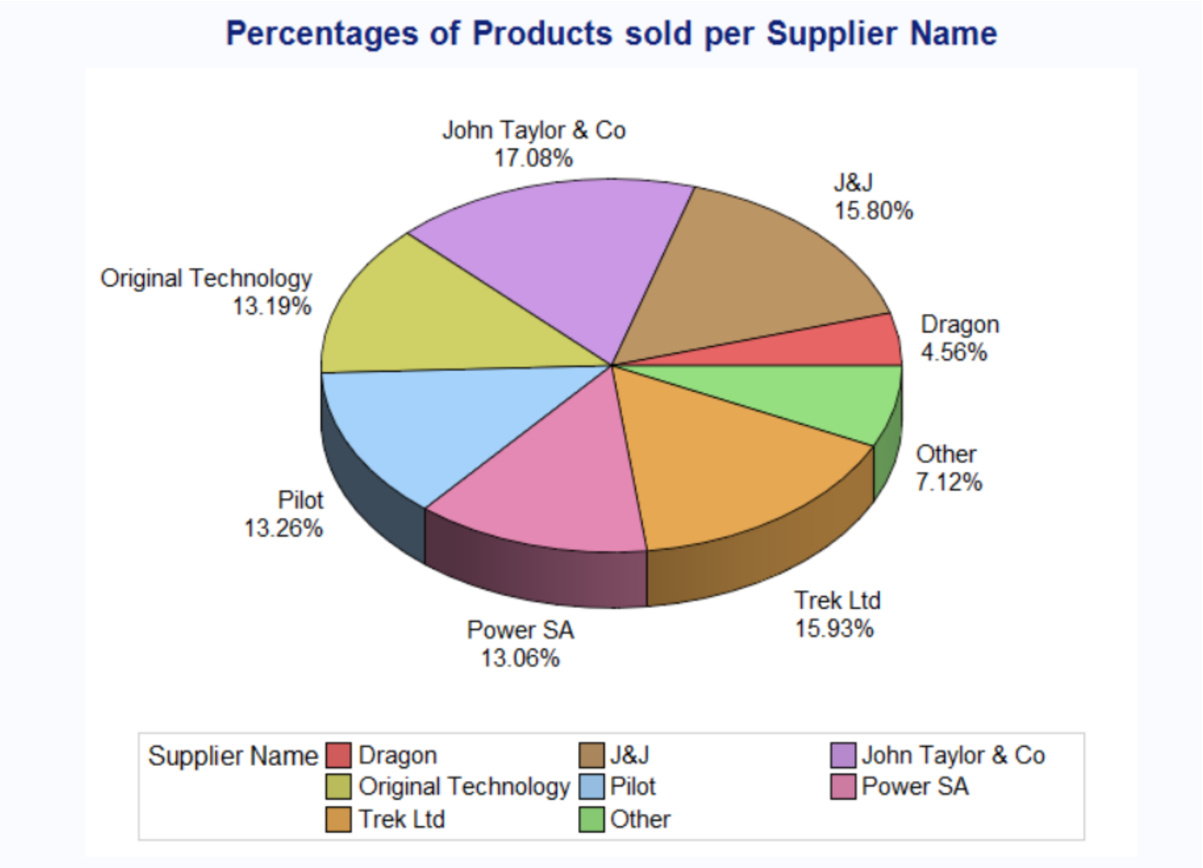


Figure 32 - Purchased Products Per Supplier Name

- Create graphs to show the percentage and actual revenues of products sold by each supplier (use the name of the supplier and not its code).

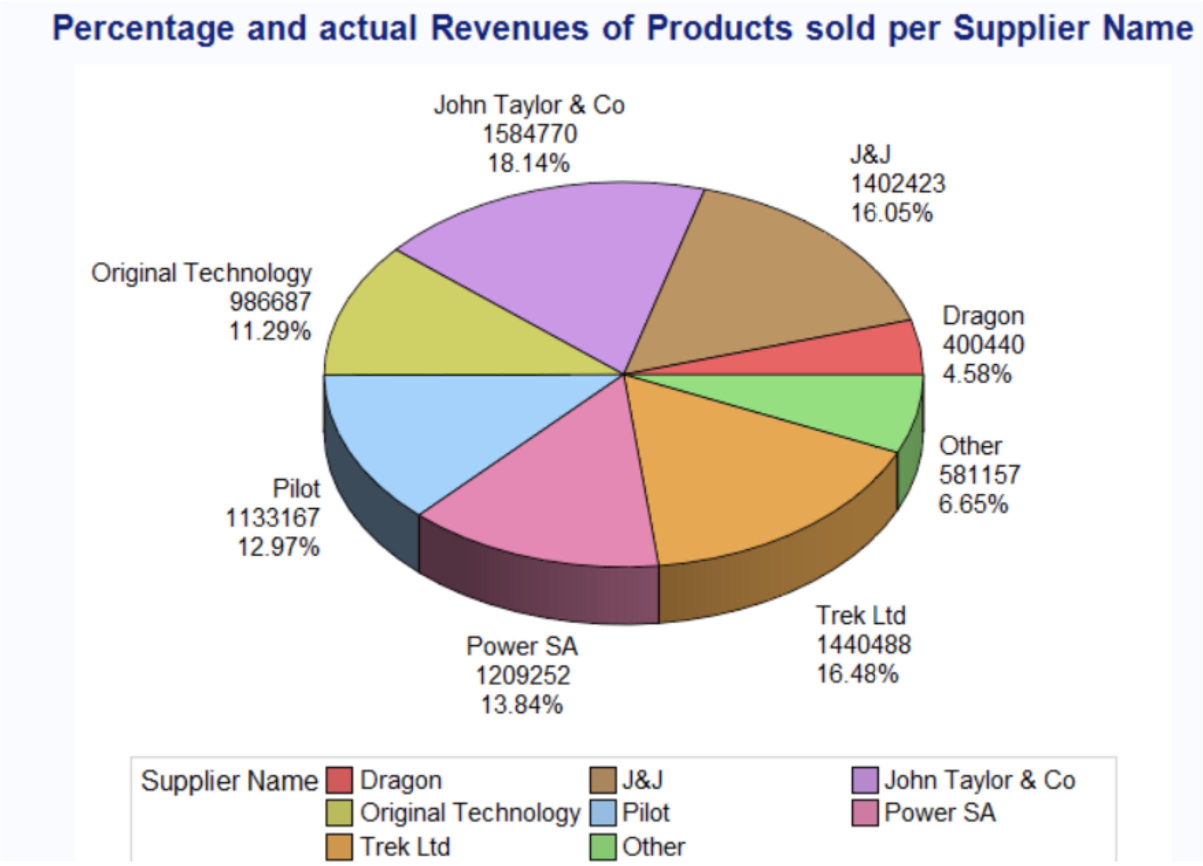
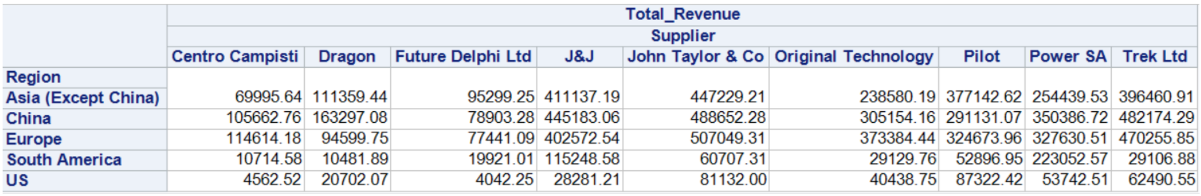


Figure 33 - Purchased Product Revenues Per Supplier Name

- Create a cross tabulation table to show the total revenue of the company with respect to the origins of the products sold by each supplier (Use the names of the suppliers and the names of the countries of origins and not their codes. Put the total revenue in the middle of the cross tabulation, the origin in the rows and the suppliers in the columns). For this task you have to use proc tabulate (find relevant instructions in the web or in sas help).



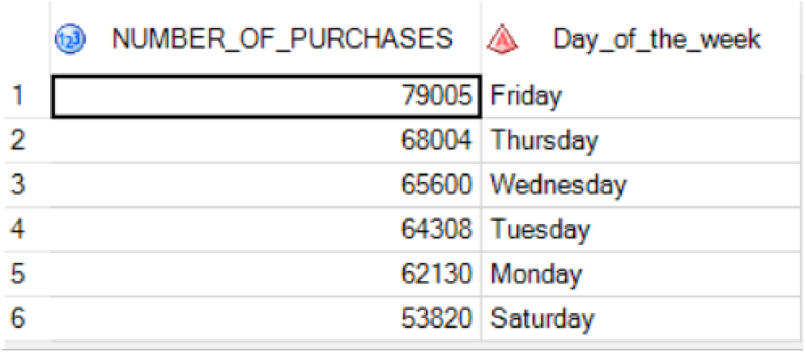
	Total_Revenue								
	Supplier								
	Centro Campisti	Dragon	Future Delphi Ltd	J&J	John Taylor & Co	Original Technology	Pilot	Power SA	Trek Ltd
Region									
Asia (Except China)	69995.64	111359.44	95299.25	411137.19	447229.21	238580.19	377142.62	254439.53	396460.91
China	105662.76	163297.08	78903.28	445183.06	488652.28	305154.16	291131.07	350386.72	482174.29
Europe	114614.18	94599.75	77441.09	402572.54	507049.31	373384.44	324673.96	327630.51	470255.85
South America	10714.58	10481.89	19921.01	115248.58	60707.31	29129.76	52896.95	223052.57	29106.88
US	4562.52	20702.07	4042.25	28281.21	81132.00	40438.75	87322.42	53742.51	62490.55

Figure 34 - Tabulation Table Product Origins - Supplier Names

Based on Figures 32,33 and 35, most of the Products are supplied by John Taylor & Co and most of them are made in Europe. The biggest revenues are earned by them, as well.

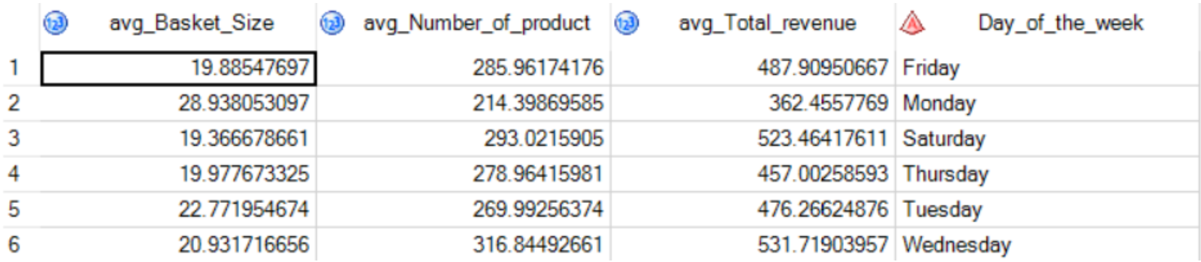
7. The company wants to focus on its sales so as to conduct promotional activities in store. What days would you propose that these activities should take place and why?

 What is the distribution of purchases per day of the week? Is there any difference among the various days (e.g. basket size, number of products per invoice etc). In order to find the day of the week when the sale takes place use the weekday function.



	NUMBER_OF_PURCHASES	Day_of_the_week
1	79005	Friday
2	68004	Thursday
3	65600	Wednesday
4	64308	Tuesday
5	62130	Monday
6	53820	Saturday

Figure 35 - Sales Per Day of Week



	avg_Basket_Size	avg_Number_of_product	avg_Total_revenue	Day_of_the_week
1	19.88547697	285.96174176	487.90950667	Friday
2	28.938053097	214.39869585	362.4557769	Monday
3	19.366678661	293.0215905	523.46417611	Saturday
4	19.977673325	278.96415981	457.00258593	Thursday
5	22.771954674	269.99256374	476.26624876	Tuesday
6	20.931716656	316.84492661	531.71903957	Wednesday

Figure 36 - Basket Size Per Day of Week

Based on Figure 35, most purchases are made on Friday which means that there is more traffic. From Figure 36, we consider that Friday contributes a lot to the turnover. For the above reasons, I would recommend promotional activities in store to be conducted on Friday.

8. The company wants to profile its customers based on their importance so as to offer them personalized services and products. The customer segmentation is asked to be done based on the three parameters of the RFM model. Before the application of the RFM model the RFM data set should be created. It is reminded that the RFM model is based on the following three parameters:

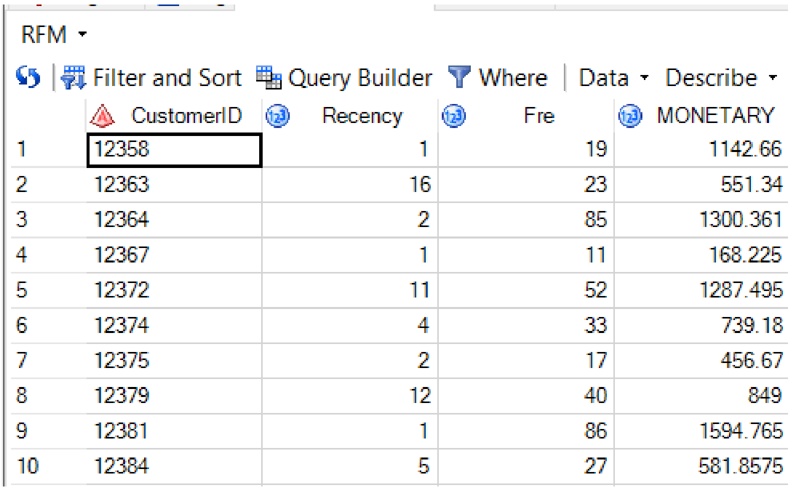
Recency - How recently did the customer purchase?

Frequency - How often do they purchase?

MonetaryValue - How much do they spend?

For this task proc sql can be used. For the calculation of R, F, M the following functions will be useful: max, sum, count and intck (For the intck use the argument week and the argument 16/12/2011 for today's date).

For the creation of the variable Monetary, the price, quantity and promotion variables should be used.



	CustomerID	Recency	Fre	MONETARY
1	12358	1	19	1142.66
2	12363	16	23	551.34
3	12364	2	85	1300.361
4	12367	1	11	168.225
5	12372	11	52	1287.495
6	12374	4	33	739.18
7	12375	2	17	456.67
8	12379	12	40	849
9	12381	1	86	1594.765
10	12384	5	27	581.8575

Figure 37 – Recency, Frequency, Monetary Value

D. SAS Enterprise Miner (In some questions Base SAS Programming should also be used)

9. Create customer segments by analyzing the RFM data set from the previous question using SAS Enterprise Miner and the three parameters of the RFM model. In order to access the RFM data set you must create a library named RESULTS in SAS Enterprise Miner that should be connected with the same path that the RESULTS library in SAS Studio is connected. It should be underlined that in order for the cluster analysis to produce logical results the customers with extreme values of the variables R, F, M should be excluded from the analysis. In order to do that, descriptive statistics tasks (e.g. proc univariate with the percentiles output) should be used in Base SAS. After the clusters are created in SAS Enterprise Miner the RFM data set with the newly created cluster column should be exported to a library and then by using Base SAS Programming the demographic data (age, gender and country) of the two most important clusters (justify why the selected ones are the most important) should be described.

Based on percentile tables, we keep only rows where recency is less than or equal to 38 and frequency is less than or equal to 208 and monetary value is less than or equal to 3619.424. This way, we are getting rid of extreme values.

After that , we load the data into Enteprise Miner (RFM button),we do clustering (Cluster button) and we save the results (Save Data button).

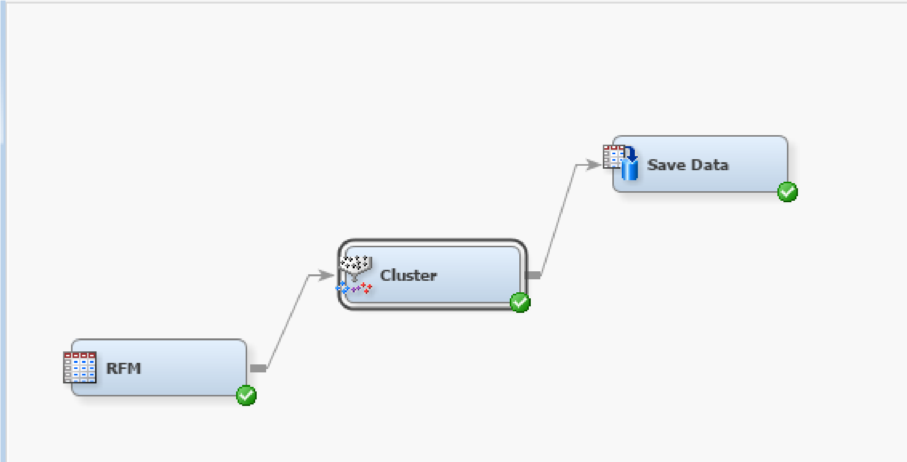
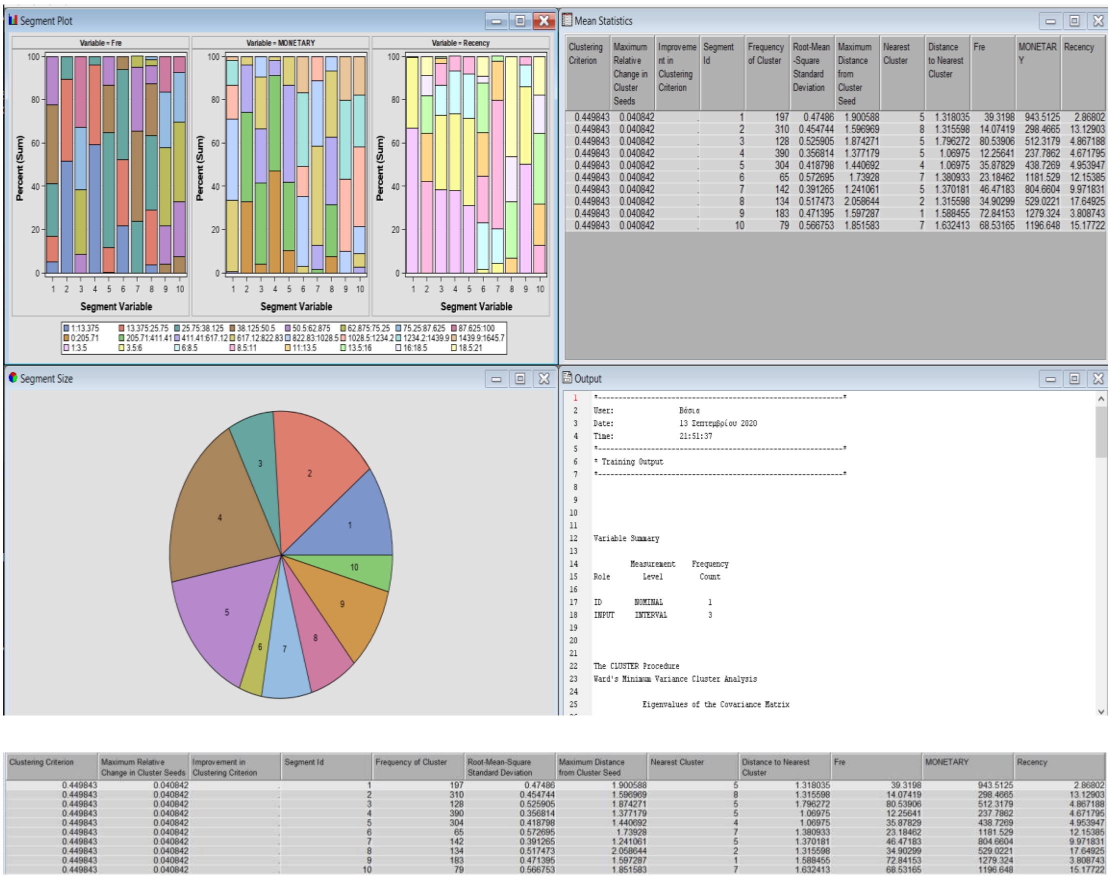


Figure 38 - Enterprise Miner - Clustering



Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster	Distance to Nearest Cluster	Fre	MONETARY	Recency
0.449843	0.040842	.	1	197	0.47486	1.900588	5	1.318035	39.3198	943.5125	2.86802
0.449843	0.040842	.	2	310	0.454744	1.596969	8	1.315598	14.07419	298.4665	13.12903
0.449843	0.040842	.	3	128	0.525905	1.874271	5	1.796272	80.53906	512.3179	4.867188
0.449843	0.040842	.	4	390	0.356814	1.377179	5	1.06975	12.25641	237.7862	4.671795
0.449843	0.040842	.	5	304	0.418798	1.440692	4	1.06975	35.87829	438.7269	4.953947
0.449843	0.040842	.	6	65	0.572695	1.73928	7	1.380933	23.18462	1181.529	12.15385
0.449843	0.040842	.	7	142	0.391265	1.241061	5	1.370181	46.47183	804.6604	9.971831
0.449843	0.040842	.	8	134	0.517473	2.058644	2	1.315598	34.90299	529.0221	17.64925
0.449843	0.040842	.	9	183	0.471395	1.597287	1	1.588455	72.84153	1279.324	3.808743
0.449843	0.040842	.	10	79	0.566753	1.851583	7	1.632413	68.53165	1196.648	15.17722

```
*------------------------------------------------------------*
User:                Βάσια
Date:                13 Σεπτεμβρίου 2020
Time:                21:51:37
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*




Variable Summary

         Measurement    Frequency
Role        Level         Count

ID        NOMINAL           1
INPUT     INTERVAL          3



The CLUSTER Procedure
Ward's Minimum Variance Cluster Analysis

            Eigenvalues of the Covariance Matrix
```

Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster	Distance to Nearest Cluster	Fre	MONETARY	Recency
0.449843	0.040842	.	1	197	0.47486	1.900588	5	1.318035	39.3198	943.5125	2.86802
0.449843	0.040842	.	2	310	0.454744	1.596969	8	1.315598	14.07419	298.4665	13.12903
0.449843	0.040842	.	3	128	0.525905	1.874271	5	1.796272	80.53906	512.3179	4.867188
0.449843	0.040842	.	4	390	0.356814	1.377179	5	1.06975	12.25641	237.7862	4.671795
0.449843	0.040842	.	5	304	0.418798	1.440692	4	1.06975	35.87829	438.7269	4.953947
0.449843	0.040842	.	6	65	0.572695	1.73928	7	1.380933	23.18462	1181.529	12.15385
0.449843	0.040842	.	7	142	0.391265	1.241061	5	1.370181	46.47183	804.6604	9.971831
0.449843	0.040842	.	8	134	0.517473	2.058644	2	1.315598	34.90299	529.0221	17.64925
0.449843	0.040842	.	9	183	0.471395	1.597287	1	1.588455	72.84153	1279.324	3.808743
0.449843	0.040842	.	10	79	0.566753	1.851583	7	1.632413	68.53165	1196.648	15.17722

Figure 39 - Clustering Results

After the clusters are created in SAS Enterprise Miner , we create the below table:

Cluster	Recency	Frequency	Monetary	Description
1	↓	↑	↑	Churners
2	↑	↓	↓	First Time Uncertain
3	↓	↑	↑	Churners
4	↓	↓	↓	Worst
5	↓	↓	↓	Worst
6	↑	↓	↑	First Time Uncertain
7	↑	↑	↑	Best
8	↑	↓	↓	First Time Uncertain
9	↓	↑	↑	Churners
10	↑	↑	↑	Best

Figure 40 - Cluster Descriptions

Worst Customers: The customers with lowest frequency of purchases in the past six months, and who spend less money than average on their transactions. The probability to gain this cluster of people back are minimal, and their habits does not make it worth to take actions in order to approach them.

First Time Uncertain: Customers, who made a recent purchase but in the past six months have not made many of them. For the specific segment, we could make promotion offers to let them better evaluate our company's pros and build a more stable relationship.

Churners: Customers who spend frequently more money than average, but they have quite a time to make a purchase. We have to send them a questionnaire in order to figure out how their experience in our shop is, whether they have some complaints and do not want to buy from us anymore.

Best Customers: Customers who spend the most money, most frequently than any other cluster. The marketing strategy for this group of customers could be to send a letter to express our appreciation for their preference to us, with a reward discount coupon for their next purchase.

We are going to visualize the demographic characteristics of cluster 2 (first time uncertain) and cluster 9 (churner) which fall into different categories. We consider them as important since they are more likely to become good customers. It does not make sense to visualize best customers clusters (cluster 7 and cluster 10) since they are already good. We will not choose worst customers clusters (cluster 4 and cluster 5) as it is more difficult to turn them into good customers.

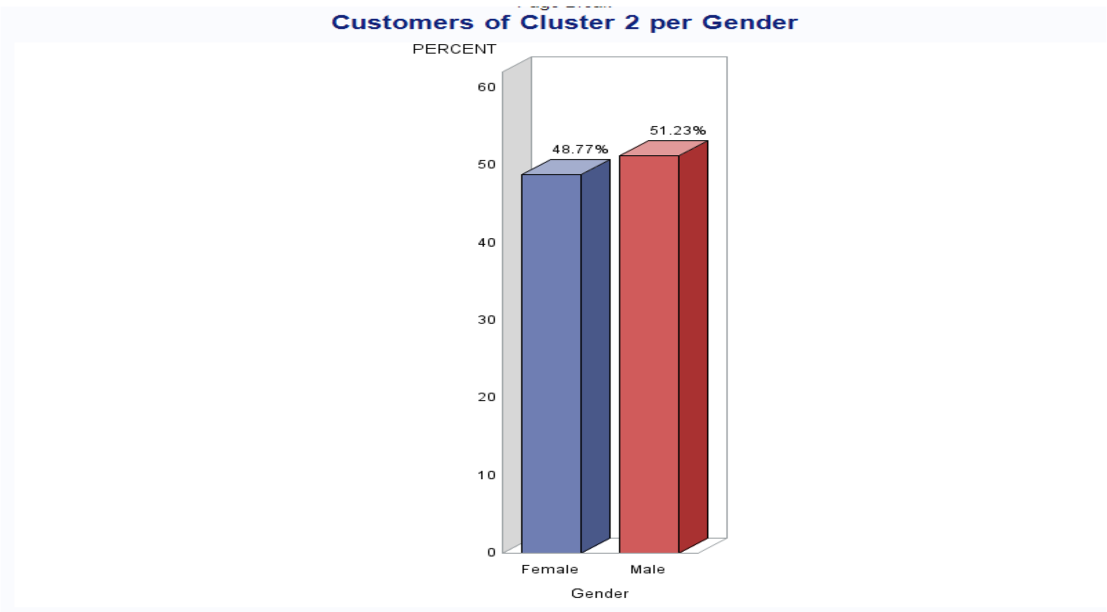


Figure 41 - Customers of Cluster 2 per Gender

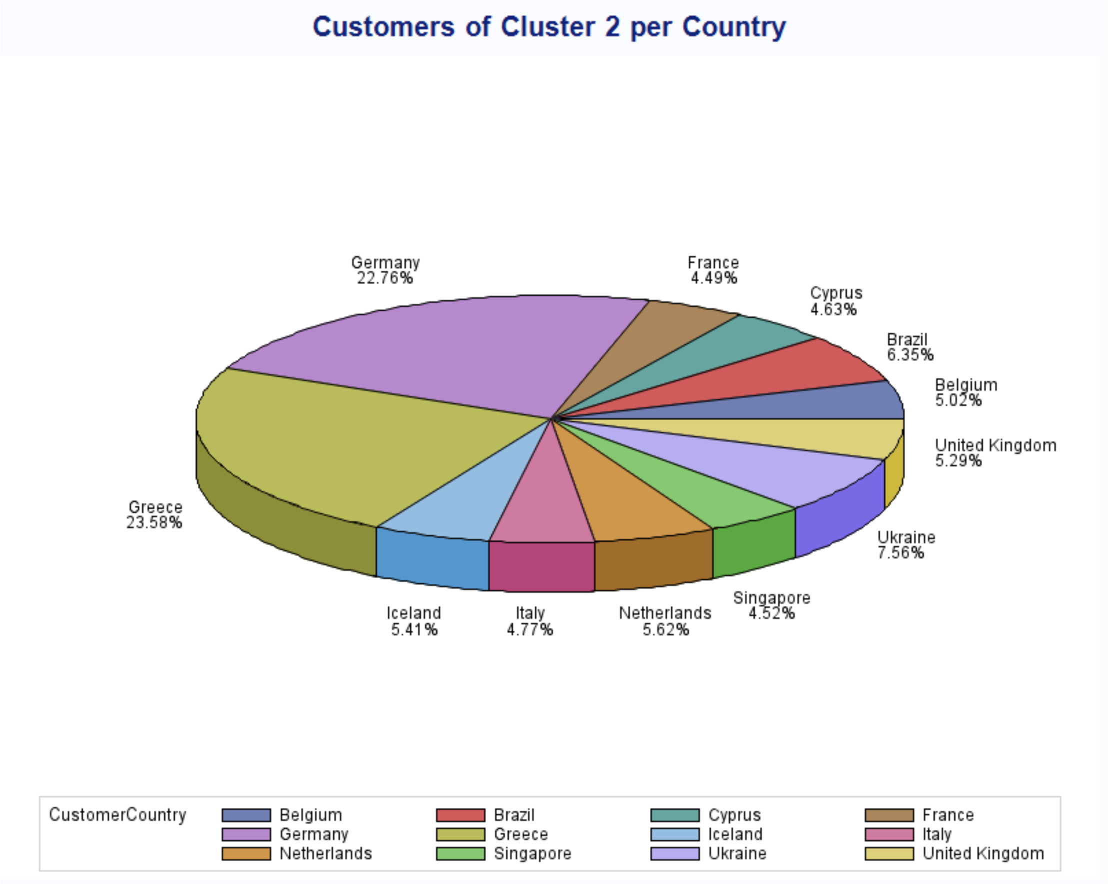


Figure 42 - Customers of Cluster 2 per Country

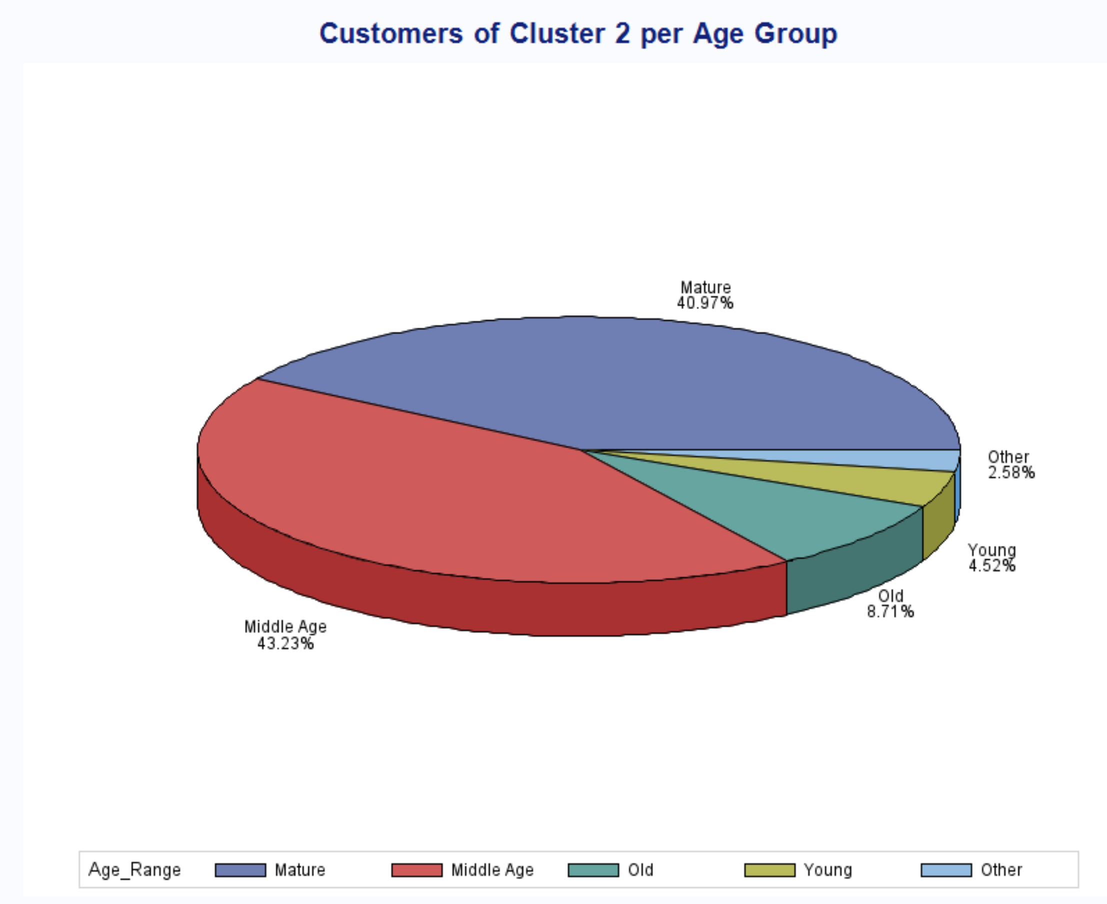


Figure 43 - Customers of Cluster 2 per Age Group

From figures 41-43, we consider that the majority of first time uncertain customers are middle-aged men from Greece.

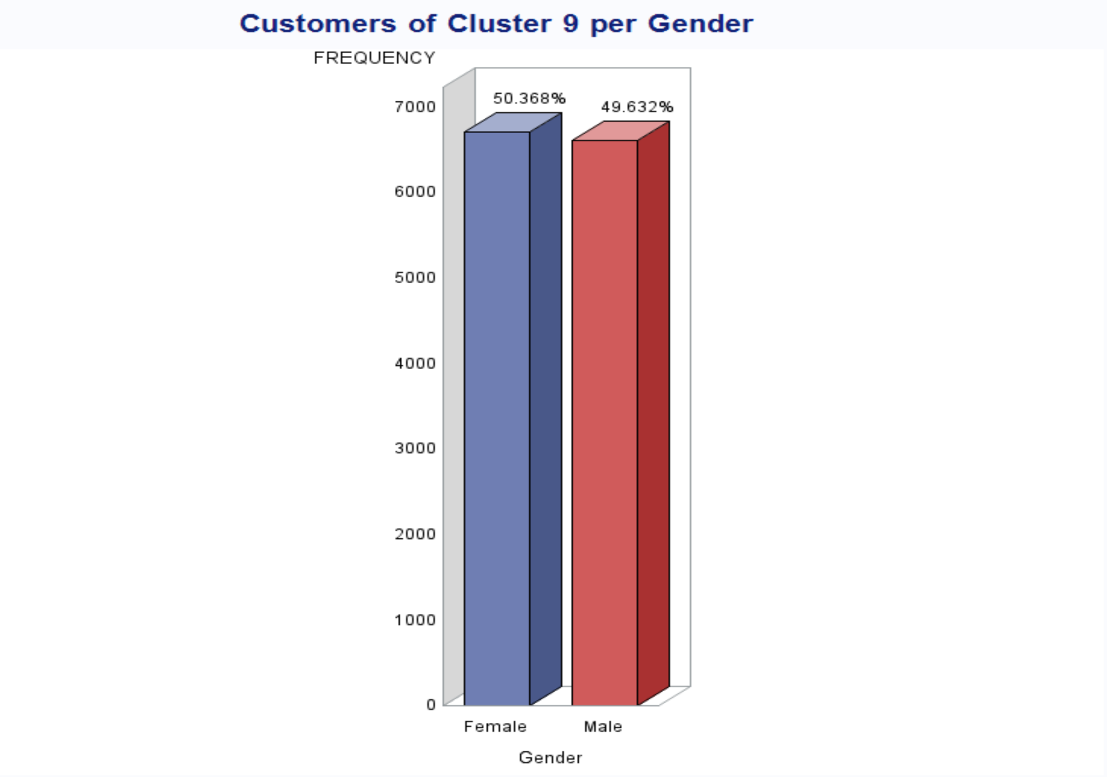


Figure 44 - Customers of Cluster 9 per Gender

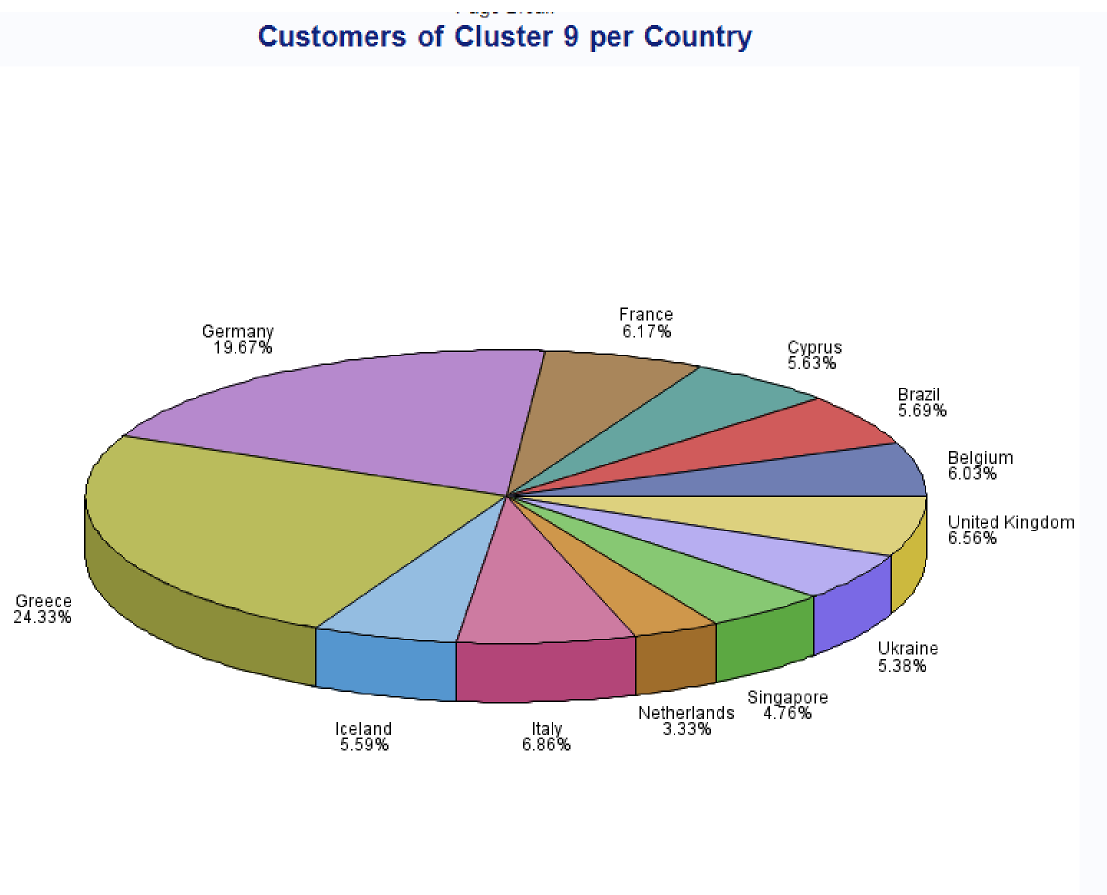


Figure 45 - Customers of Cluster 9 per Country

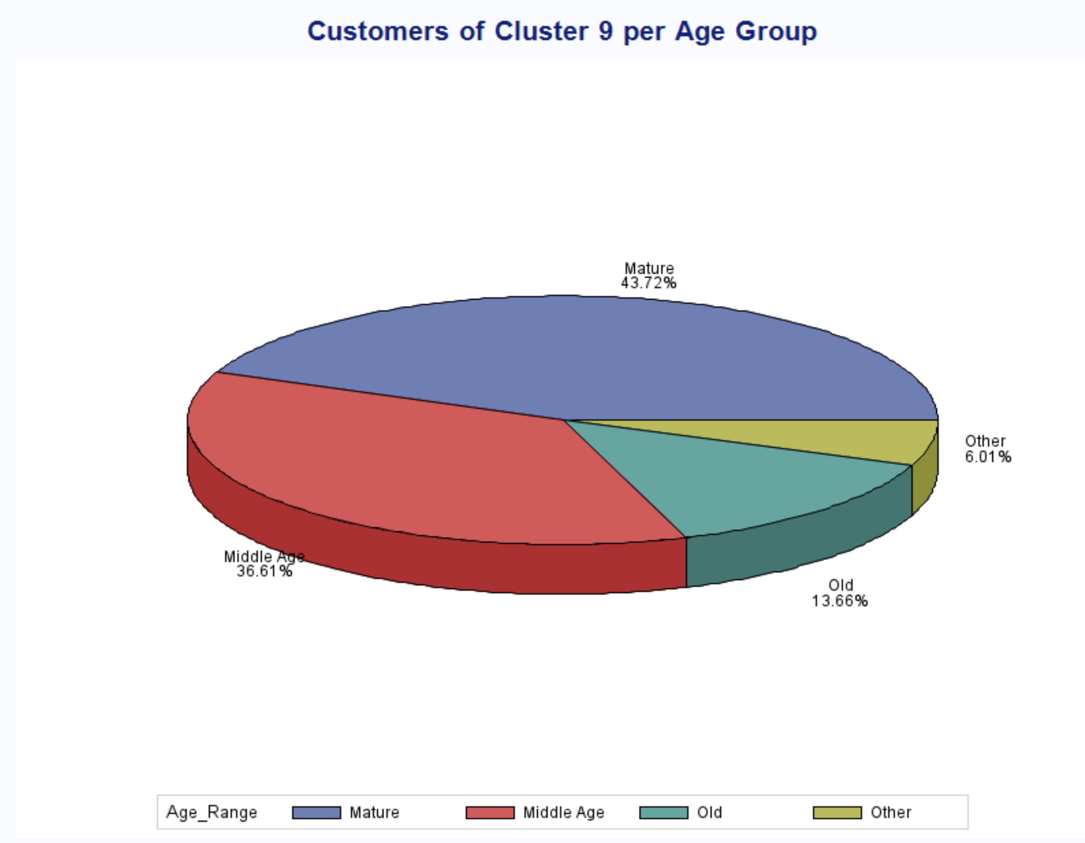
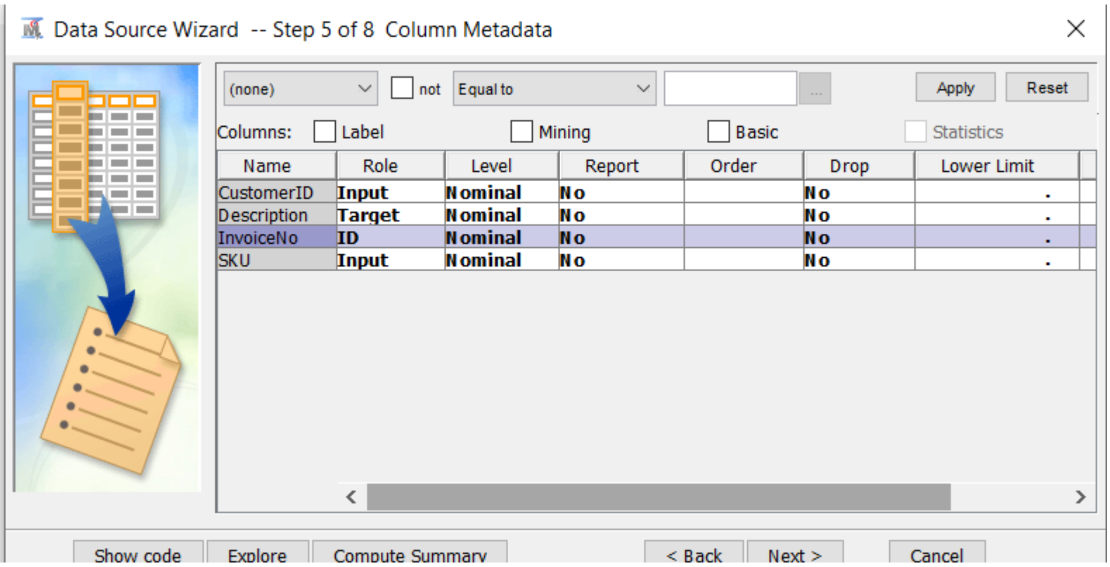


Figure 46 - Customers of Cluster 9 per Age Group

From figures 44-46 , we consider that the majority of churner customers are mature women from Greece.

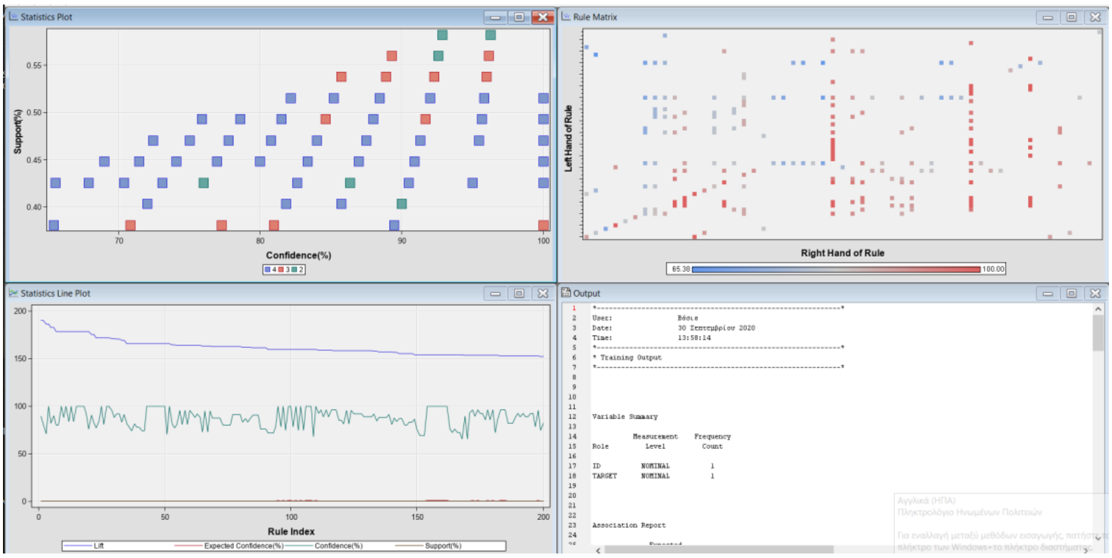
10. The company is interested to change internally the store based on the products that tend to be bought together. In order to apply this initiative the company must be sure about the associations among the product names. You are asked to find which products are bought together (associations of product names) in the whole data set. Then find the associations among products in the two most important clusters (according to your business thinking) previously identified so if a customer is found to belong in one of them to receive the most suitable/ best proposals/ offers. For this task Base SAS should be used to filter the customers that belong to the two most important clusters, create the two relevant data sets and then these data sets to be analyzed using association rules through SAS Enterprise Miner.


Data Source Wizard -- Step 5 of 8 Column Metadata

(none) □ not Equal to ... Apply Reset

Columns: □ Label □ Mining □ Basic □ Statistics

Name	Role	Level	Report	Order	Drop	Lower Limit
CustomerID	Input	Nominal	No		No	.
Description	Target	Nominal	No		No	.
InvoiceNo	ID	Nominal	No		No	.
SKU	Input	Nominal	No		No	.



```
*------------------------------------------------------------*
User:                Βάσια
Date:                30 Σεπτεμβρίου 2020
Time:                13:58:14
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*




Variable Summary

          Measurement    Frequency
Role         Level         Count

ID          NOMINAL          1
TARGET      NOMINAL          1




Association Report
```

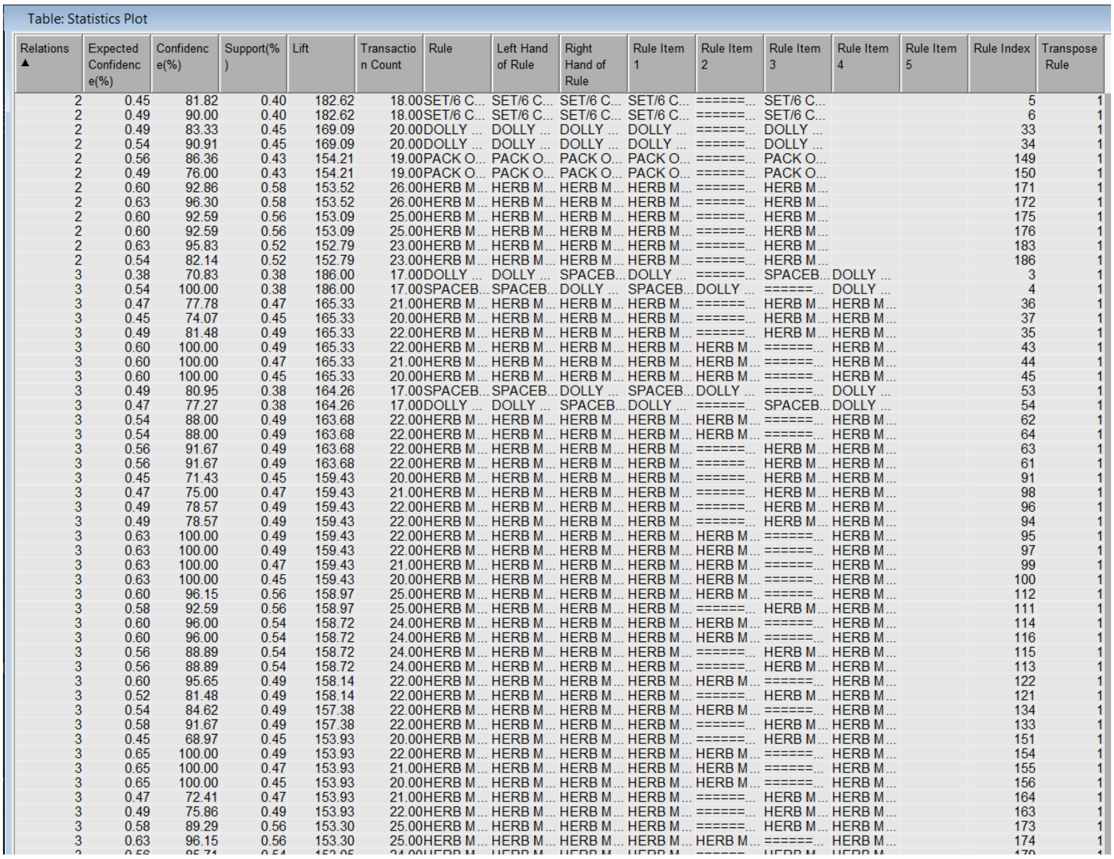
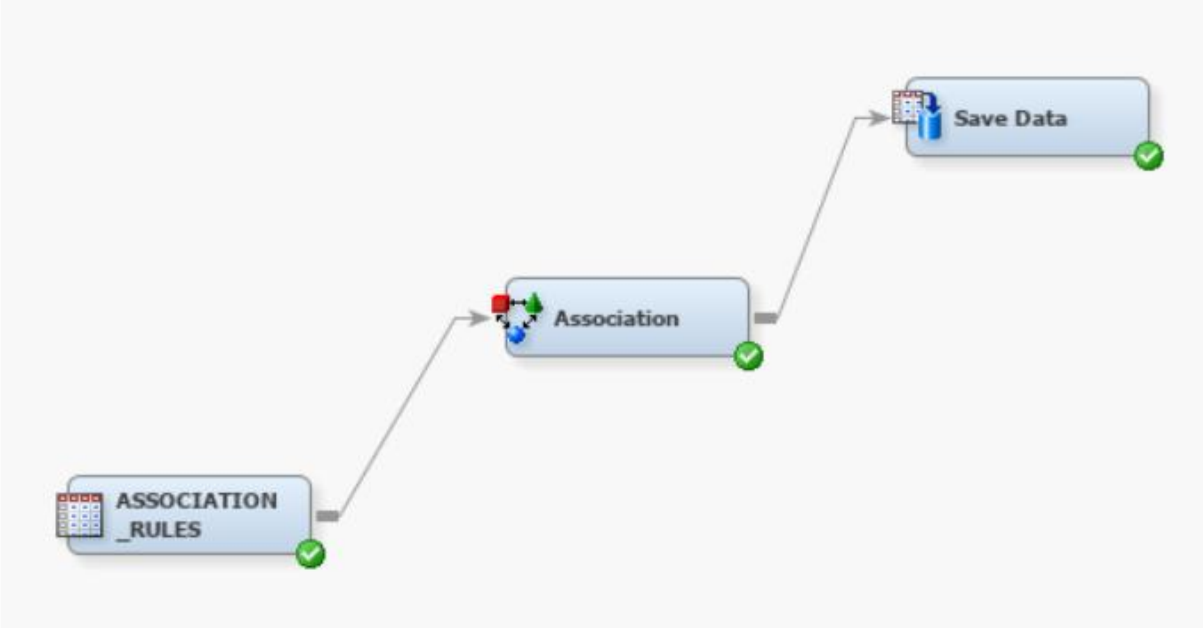


Table: Statistics Plot

Relations ▲	Expected Confidence(%)	Confidence(%)	Support(%)	Lift	Transaction Count	Rule	Left Hand of Rule	Right Hand of Rule	Rule Item 1	Rule Item 2	Rule Item 3	Rule Item 4	Rule Item 5	Rule Index	Transpose Rule
2	0.45	81.82	0.40	182.62	18.00	SET/6 C...	SET/6 C...	SET/6 C...	SET/6 C...	======...	SET/6 C...			5	1
2	0.49	90.00	0.40	182.62	18.00	SET/6 C...	SET/6 C...	SET/6 C...	SET/6 C...	======...	SET/6 C...			6	1
2	0.49	83.33	0.45	169.09	20.00	DOLLY ...	DOLLY ...	DOLLY ...	DOLLY ...	======...	DOLLY ...			33	1
2	0.54	90.91	0.45	169.09	20.00	DOLLY ...	DOLLY ...	DOLLY ...	DOLLY ...	======...	DOLLY ...			34	1
2	0.56	86.36	0.43	154.21	19.00	PACK O...	PACK O...	PACK O...	PACK O...	======...	PACK O...			149	1
2	0.49	76.00	0.43	154.21	19.00	PACK O...	PACK O...	PACK O...	PACK O...	======...	PACK O...			150	1
2	0.60	92.86	0.58	153.52	26.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...			171	1
2	0.63	96.30	0.58	153.52	26.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...			172	1
2	0.60	92.59	0.56	153.09	25.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...			175	1
2	0.60	92.59	0.56	153.09	25.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...			176	1
2	0.63	95.83	0.52	152.79	23.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...			183	1
2	0.54	82.14	0.52	152.79	23.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...			186	1
3	0.38	70.83	0.38	186.00	17.00	DOLLY ...	DOLLY ...	SPACEB...	DOLLY ...	======...	SPACEB...	DOLLY ...		3	1
3	0.54	100.00	0.38	186.00	17.00	SPACEB...	SPACEB...	DOLLY ...	SPACEB...	DOLLY ...	======...	DOLLY ...		4	1
3	0.47	77.78	0.47	165.33	21.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		36	1
3	0.45	74.07	0.45	165.33	20.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		37	1
3	0.49	81.48	0.49	165.33	22.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		35	1
3	0.60	100.00	0.49	165.33	22.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		43	1
3	0.60	100.00	0.47	165.33	21.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		44	1
3	0.60	100.00	0.45	165.33	20.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		45	1
3	0.49	80.95	0.38	164.26	17.00	SPACEB...	SPACEB...	DOLLY ...	SPACEB...	DOLLY ...	======...	DOLLY ...		53	1
3	0.47	77.27	0.38	164.26	17.00	DOLLY ...	DOLLY ...	SPACEB...	DOLLY ...	======...	SPACEB...	DOLLY ...		54	1
3	0.54	88.00	0.49	163.68	22.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		62	1
3	0.54	88.00	0.49	163.68	22.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		64	1
3	0.56	91.67	0.49	163.68	22.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		63	1
3	0.56	91.67	0.49	163.68	22.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		61	1
3	0.45	71.43	0.45	159.43	20.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		91	1
3	0.47	75.00	0.47	159.43	21.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		98	1
3	0.49	78.57	0.49	159.43	22.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		96	1
3	0.49	78.57	0.49	159.43	22.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		94	1
3	0.63	100.00	0.49	159.43	22.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		95	1
3	0.63	100.00	0.49	159.43	22.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		97	1
3	0.63	100.00	0.47	159.43	21.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		99	1
3	0.63	100.00	0.45	159.43	20.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		100	1
3	0.60	96.15	0.56	158.97	25.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		112	1
3	0.58	92.59	0.56	158.97	25.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		111	1
3	0.60	96.00	0.54	158.72	24.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		114	1
3	0.60	96.00	0.54	158.72	24.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		116	1
3	0.56	88.89	0.54	158.72	24.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		115	1
3	0.56	88.89	0.54	158.72	24.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		113	1
3	0.60	95.65	0.49	158.14	22.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		122	1
3	0.52	81.48	0.49	158.14	22.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		121	1
3	0.54	84.62	0.49	157.38	22.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		134	1
3	0.58	91.67	0.49	157.38	22.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		133	1
3	0.45	68.97	0.45	153.93	20.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		151	1
3	0.65	100.00	0.49	153.93	22.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		154	1
3	0.65	100.00	0.47	153.93	21.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		155	1
3	0.65	100.00	0.45	153.93	20.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		156	1
3	0.47	72.41	0.47	153.93	21.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		164	1
3	0.49	75.86	0.49	153.93	22.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		163	1
3	0.58	89.29	0.56	153.30	25.00	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...	HERB M...		173	1
3	0.63	96.15	0.56	153.30	25.00	HERB M...	HERB M...	HERB M...	HERB M...	HERB M...	======...	HERB M...		174	1



Performing Market Basket Analytics, we can find any associations between the customers' purchases. We checked to node 'Association' in Semma's toolbar Explore tab of the diagram we already produced. From the results, we checked the View-> Rules-> Rules Table, to produce the table showing the 'Left Hand of Rule', 'Right Hand of Rule' and the metrics 'Support Confidence %', 'Expected Confidence %' and 'Lift'. Those data were imported to Excel. Using filters we could identify the associations we are interested in. The associations were sorted judged based on 'Lift'. This is because it measures the strength of the association between two products, by how many times more possible it is for a customer who bought a product for the left relationship, to buy a product on the right hand of the relationship, compared to all the other products. This process is performed for cluster 2 and cluster 9 which were considered as important in question 9. The findings based on the metric described above are:

- HERB MARKER THYME ==> HERB MARKER ROSEMARY & HERB MARKER CHIVES & HERB MARKER BASIL (lift 168.00, cluster 2)

- REGENCY TEA PLATE ROSES & REGENCY MILK JUG PINK ==> REGENCY TEA PLATE GREEN & REGENCY SUGAR BOWL GREEN (lift 122.00, cluster 9)

Appendix

```
/* getting rid of duplicate lines in Invoice table */

proc sort data = RSULTS.INVOICE nodupkey;
by _all_;
run;


/* Question1 */
/* a) •    For every invoice calculate the total number of SKU's that are related to it 'Invoice total items'.
   Save the output in a new SAS data set and print the first 10  observations of it. Only the data step
   can be used for merging data sets. Proc sql can be used for the statistics. It is suggested to use
   noprint option in proc sql because the new data set will be large. */

proc sql noprint ;
CREATE TABLE RSULTS.Invoice_total_items AS
select InvoiceNo, COUNT(SKU) AS COUNT_of_SKU
            from RSULTS.BASKET
             group by InvoiceNo
    order by COUNT_of_SKU desc;
QUIT;


/* printing the first 10 observations of the above table.*/
proc sql outobs=10;
select InvoiceNo,COUNT_of_SKU
            from RSULTS.Invoice_total_items
QUIT;
```

```
/* b) •    For every invoice calculate the total value of the SKU's that are related to it 'Invoice total value.
        Save the output in a new SAS data set. For this task use the proc means with the output statement. */


proc means data=RSULTS.Basket noprint;
var UnitPrice;
class InvoiceNo;
Output Out= RSULTS.Invoice_total_value sum= Invoice_total_value;
run;

/* c) •    Divide the observations of the table 'Invoice' into two new tables where
     in the one the Sales transactions will be stored where as in the second the Returns
     transactions will be stored. This division must be done using the variable 'Operation'. */

data RSULTS.Sales RSULTS.Returns;
  set RSULTS.INVOICE;
  if Operation = '500' then output RSULTS.Sales;
  else output RSULTS.Returns;
run;

/* d  •    Create a new table that will contain customers for which there exists a birth date (no one of the
     fields Day_Of_Birth, Month_Of_Birth, Year_Of_Birth should be NULL). Then calculate the customer's age
based
    on the fact that today's date is 01/01/2019 and store it into a new variable (check the validity of the dates
e.g.
     birth year less than 1920).*/

/* converting month, day and year of birth to date of birth  */
DATA RSULTS.DATES;
   SET RSULTS.CUSTOMER;
   Date_Of_Birth = MDY(Month_Of_Birth, Day_Of_Birth, Year_Of_Birth );
   FORMAT Date_Of_Birth MMDDYY10.;
            If Date_Of_Birth= '-' or Year_Of_Birth< '1920' then delete;
RUN;

/* calculating age and rounding it*/
DATA RSULTS.DATES;
   SET RSULTS.DATES;
   Age = YRDIF(Date_Of_Birth,input('01/01/2019', mmddyy10.), 'Actual');
   Age = strip(put(round(Age,1),10.1));
RUN;



/* Question2 */

/* a) What are the demographic characteristics i.e. age, gender and country of the
     company's customers? */

proc sql noprint;
create table RSULTS.Demographic as
select CustomerID, AGE, CustomerCountry, Gender
            FROM RSULTS.DATES
            ORDER BY CustomerID;
quit;


/* b) •    Based on the age variable, create a new variable entitled Age_Range that takes
     the following values:
```

```
<18 -- > “Under 18”

18 - 25 -- > “Very Young”

26 - 35 -- > “Young”

36 - 50 -- > “Middle Age”

51 - 65 -- > “Mature”

66 – 75 -- > “Old”

>= 76      -- > “Very Old”

(Attention: do not format the values of the existing variable but create a new variable entitled Age _Range).
 */


data RSULTS.DATES;
  SET RSULTS.DATES;
  length Age_Range $12;
  if Age<18 then Age_Range='Under 18';
  else if Age<=25 then Age_Range='Very Young';
  else if Age<=35 then Age_Range='Young';
  else if Age<=50 then Age_Range='Middle Age';
  else if Age<=65 then Age_Range='Mature';
  else if Age<=75 then Age_Range='Old';
  else if Age>75 then Age_Range='Very Old';
run;


/* c) •    What are the behavioral characteristics of each age group? (visits to the stores,
  number of SKU’s purchased, total cost of purchases, average cost, minimum cost, maximum cost etc).
  The merging of the data sets must be done using exclusively the data step but the calculation of the
  statistics e.g. visits, total cost of purchases etc can be done using proc sql. Create a pie chart and
  a frequency table with the percentages of customers that belong to each age group. Augment your analysis
  by providing pie charts for the behavioral characteristics for each age group.*/

PROC SORT DATA=RSULTS.DATES;
 BY CustomerID;
RUN;

PROC SORT DATA=RSULTS.INVOICE;
 BY CustomerID;
RUN;

data RSULTS.DATES_INVOICE;
merge RSULTS.DATES(IN=A) RSULTS.INVOICE(IN=B);
by CUSTOMERID;
IF A AND B;
run;

PROC SORT DATA=RSULTS.SALES;
 BY CustomerID;
RUN;

data RSULTS.DATES_SALES;
merge RSULTS.DATES(IN=A) RSULTS.SALES(IN=B);
by CUSTOMERID;
```

```
IF A AND B;
run;


PROC SORT DATA=RSULTS.BASKET;
 BY InvoiceNo;
RUN;

PROC SORT DATA=RSULTS.DATES_INVOICE;
 BY InvoiceNo;
RUN;

data RSULTS.BASKET_DATES_INVOICE;
merge RSULTS.BASKET(IN=A) RSULTS.DATES_INVOICE(IN=B);
by InvoiceNo;
IF A AND B;
run;

PROC SORT DATA=RSULTS.DATES_SALES;
 BY InvoiceNo;
RUN;

data RSULTS.BASKET_DATES_SALES;
merge RSULTS.BASKET(IN=A) RSULTS.DATES_SALES(IN=B);
by InvoiceNo;
IF A AND B;
run;

PROC SORT DATA=RSULTS.BASKET_DATES_INVOICE;
 BY Age_Range;
RUN;

proc sql noprint;
CREATE TABLE RSULTS.Visits AS
select Age_Range, COUNT(InvoiceNo) AS STORE_VISITS
        from RSULTS.BASKET_DATES_INVOICE
         group by Age_Range
   order by STORE_VISITS desc;
quit;






proc sql noprint;
CREATE TABLE RSULTS.Purchase_metrics AS
select Age_Range, COUNT(SKU) AS SKU_purchased,
    SUM(UNITPrice) AS total_cost_purchased, AVG(UNITPrice) AS avg_cost_purchased,
          MIN(UNITPrice) AS min_cost_purchased, MAX(UNITPrice) AS max_cost_purchased
        from RSULTS.BASKET_DATES_SALES
         group by Age_Range
   order by Age_Range;
QUIT;

proc freq data = RSULTS.DATES;
tables AGE_RANGE;
run;
```

```
PROC GCHART DATA = RSULTS.DATES;

            PIE         Age_Range /
            SUMVAR=Percent
            TYPE=SUM
            NOLEGEND
            SLICE=OUTSIDE
            PERCENT=NONE
            VALUE=OUTSIDE
            OTHER=4
            OTHERLABEL="Other"
            COUTLINE=BLACK
   NOHEADING;

RUN;
QUIT



/* CALCULATE FREQUENCIES */

proc freq data = RSULTS.DATES;
ods output onewayfreqs=RSULTS.frequency_table;
tables AGE_RANGE;
run;

/* Customer Age Group Percentage Pie Chart */
PROC SQL;
            CREATE VIEW WORK.SORTTempTableSorted AS
                        SELECT T.Age_Range, T.Percent
            FROM RSULTS.FREQUENCY_TABLE as T
;
QUIT;
TITLE;
TITLE1 "Customer Age Group Percentage Pie Chart";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
;
            PIE         Age_Range /
            SUMVAR=Percent
            TYPE=SUM
            NOLEGEND
            SLICE=OUTSIDE
            PERCENT=NONE
            VALUE=OUTSIDE
            OTHER=4
            OTHERLABEL="Other"
            COUTLINE=BLACK
NOHEADING
;
RUN; QUIT;



/* SKU Purchased Per Age Group */
PROC SQL;
            CREATE VIEW WORK.SORTTempTableSorted AS
                        SELECT T.Age_Range, T.SKU_purchased
            FROM RSULTS.Purchase_metrics as T
;
```

```
QUIT;
TITLE;
TITLE1 "SKU Purchased Per Age Group";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
;
        PIE     Age_Range /
        SUMVAR=SKU_purchased
        TYPE=SUM
        NOLEGEND
        SLICE=OUTSIDE
        PERCENT=NONE
        VALUE=OUTSIDE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;
RUN; QUIT;



/* Total Cost Purchased Per Age Group */
PROC SQL;
        CREATE VIEW WORK.SORTTempTableSorted AS
                SELECT T.Age_Range, T.total_cost_purchased
        FROM RSULTS.Purchase_metrics as T
;
QUIT;
TITLE;
TITLE1 "Total Cost Purchased Per Age Group";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
;
        PIE     Age_Range /
        SUMVAR=total_cost_purchased
        TYPE=SUM
        NOLEGEND
        SLICE=OUTSIDE
        PERCENT=NONE
        VALUE=OUTSIDE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;
RUN; QUIT;


/* Avg Cost Purchased Per Age Group */
PROC SQL;
        CREATE VIEW WORK.SORTTempTableSorted AS
                SELECT T.Age_Range, T.avg_cost_purchased
        FROM RSULTS.Purchase_metrics as T
;
QUIT;
TITLE;
TITLE1 "Average Cost Purchased Per Age Group";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
```

```
;
        PIE        Age_Range /
        SUMVAR=avg_cost_purchased
        TYPE=SUM
        NOLEGEND
        SLICE=OUTSIDE
        PERCENT=NONE
        VALUE=OUTSIDE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;
RUN; QUIT;

/* Max Cost Purchased Per Age Group */
PROC SQL;
        CREATE VIEW WORK.SORTTempTableSorted AS
                SELECT T.Age_Range, T.max_cost_purchased
        FROM RSULTS.Purchase_metrics as T
;
QUIT;
TITLE;
TITLE1 "Max Cost Purchased Per Age Group";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
;
        PIE        Age_Range /
        SUMVAR=max_cost_purchased
        TYPE=SUM
        NOLEGEND
        SLICE=OUTSIDE
        PERCENT=NONE
        VALUE=OUTSIDE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;
RUN; QUIT;

/* Min Cost Purchased Per Age Group */
PROC SQL;
        CREATE VIEW WORK.SORTTempTableSorted AS
                SELECT T.Age_Range, T.min_cost_purchased
        FROM RSULTS.Purchase_metrics as T
;
QUIT;
TITLE;
TITLE1 "Min Cost Purchased Per Age Group";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
;
        PIE        Age_Range /
        SUMVAR=min_cost_purchased
        TYPE=SUM
        NOLEGEND
        SLICE=OUTSIDE
        PERCENT=NONE
        VALUE=OUTSIDE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;
RUN; QUIT;


/* store visits per Age Group */
PROC SQL;
        CREATE VIEW WORK.SORTTempTableSorted AS
                SELECT T.Age_Range, T.STORE_VISITS
        FROM RSULTS.VISITS as T
;
QUIT;
Legend1
        FRAME
        POSITION = (BOTTOM CENTER OUTSIDE)
        ;
TITLE;
TITLE1 "Store Visits per Age Group";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
;
        PIE3D   Age_Range /
        SUMVAR=STORE_VISITS
        TYPE=SUM
        LEGEND=LEGEND1
        SLICE=OUTSIDE
        PERCENT=OUTSIDE
        VALUE=OUTSIDE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;

RUN;
QUIT;




/* Question3 */

/* What was the level of Sales and Returns? The variable ‘Operation´ of the 'Invoice table takes the values '500
and 501'.
Create a bar chart with the monetary values and a frequency table by creating a custom format for which
500=Sale and 501=Cancellation.*/

/* a) make new column with Sales and Cancellations based on 500 and 501 operations */
proc format;
value names 500 = 'Sales'
            501 = 'Cancellations';
run;

 DATA RSULTS.BASKET_DATES_INVOICE;
 SET RSULTS.BASKET_DATES_INVOICE;
 Operation_name = Operation;
 format Operation_name names. ;
 run;


 /* level of Sales and Returns */
PROC SQL;
           CREATE VIEW WORK.SORTTEMPTABLESORTED_0003 AS
                     SELECT T.Operation_name
           FROM RSULTS.BASKET_DATES_INVOICE as T
;
QUIT;
Axis1
           STYLE=1
           WIDTH=1
           MINOR=
           (NUMBER=1
           )


;
Axis2
           STYLE=1
           WIDTH=1


;
TITLE;
TITLE1 "Level Of Sales and Returns";
FOOTNOTE;
PROC GCHART DATA=WORK.SORTTEMPTABLESORTED_0003
;
           VBAR
            Operation_name
 /
           CLIPREF
FRAME   TYPE=FREQ
           COUTLINE=BLACK
           RAXIS=AXIS1
           MAXIS=AXIS2
;
RUN; QUIT;

/* creating a frequency table custom format for which 500=Sale and
501=Cancellation */

proc freq data = RSULTS.INVOICE;
tables OPERATION;
run;



/* Create graphs for the average basket size i.e. number of SKU’s, total monetary value, etc and
comment on your findings. Proc sql can be used only for the calculation of
the statistics e.g. of the average basket and not e.g. for merging data sets (for this data step should be used).*/

proc sql noprint;
CREATE TABLE RSULTS.basket_statistics_tmp AS
select count(SKU) AS Number_of_SKU, count(UnitPrice) AS Total_price_of_Bsk
```

```
            FROM RSULTS.BASKET_DATES_SALES
            where Description <> 'Adjust bad debt'
            GROUP BY InvoiceNo;
quit;

proc sql print;
CREATE TABLE RSULTS.basket_size AS
select avg(Number_of_SKU) AS Avg_num_of_SKU, count(Total_price_of_Bsk) AS Avg_price_of_bsk
            FROM RSULTS.basket_statistics_tmp
            ORDER BY Avg_num_of_SKU;
quit;









/* Question4 */

/* Create pie charts for the variable 'Payment Method’ for every value of the variable 'Movement'
the 'Basket' table.In the graphs show the type of the payment method and not its code e.g. credit card 20%.
• Which payment method is the most popular (it is used most by the customers)? Which payment method brings the
biggest revenues for the company? (Use graphs).
• Products of what origin do the customers prefer (use graphs)? Products of which origin bring the biggest revenues to
the retailer and what is the amount of the revenues for each origin (use graphs and tables).   */

/* calculating Quantity price   */
DATA RSULTS.BASKET_DATES_SALES;
  SET RSULTS.BASKET_DATES_SALES;
  QuantityPrice = Quantity*UnitPrice;
RUN;

PROC SORT DATA=RSULTS.BASKET;
 BY       InvoiceNo;
RUN;

PROC SORT DATA=RSULTS.INVOICE;
 BY InvoiceNo;
RUN;

data RSULTS.BASKET_INVOICE;
merge RSULTS.BASKET(IN=A) RSULTS.INVOICE(IN=B);
by InvoiceNo;
IF A AND B;
run;

PROC SORT DATA=RSULTS.Payment_Method;
 BY       Code;
RUN;

PROC SORT DATA=RSULTS.BASKET_INVOICE;
 BY Payment_Method;
RUN;

data RSULTS.BASKET_INVOICE_PAYMENT ;
```

```
Merge  RSULTS.BASKET_INVOICE RSULTS.PAYMENT_METHOD (rename=(CODE=PAYMENT_METHOD));
by PAYMENT_METHOD;
run;


PROC SORT DATA=RSULTS.BASKET;
  BY 'Product Origin'n;
RUN;

PROC SORT DATA=RSULTS.Product_Origin ;
  BY Code;
RUN;

data RSULTS.BASKET_PRODUCT ;
Merge  RSULTS.BASKET RSULTS.PRODUCT_ORIGIN (rename=(CODE='Product Origin'n));
by 'Product Origin'n;
run;

proc sql noprint;
create table RSULTS.Revenues_Origin as
select Region, sum(QuantityPrice) AS Revenues
            FROM RSULTS.Basket_Product
   GROUP BY Region
            ORDER BY Revenues DESC;
quit;


/* Transaction Percentages per Payment Method */

PROC SQL;
            CREATE VIEW RSULTS.SORTTempTableSorted AS
                        SELECT T.Method
            FROM RSULTS.BASKET_INVOICE_PAYMENT as T
;
QUIT;
Legend1
            FRAME
            POSITION = (BOTTOM CENTER OUTSIDE)
            ;
TITLE;
TITLE1 "Transaction Percentages per Payment Method";
FOOTNOTE;
PROC GCHART DATA =RSULTS.SORTTempTableSorted
;
            PIE3D     Method /
            TYPE=PCT
            LEGEND=LEGEND1
            SLICE=OUTSIDE
            PERCENT=OUTSIDE
            VALUE=NONE
            OTHER=4
            OTHERLABEL="Other"
            COUTLINE=BLACK
NOHEADING
;

RUN;
```

```
QUIT;
/* Revenues Percentages per Payment Method */
PROC SQL;
          CREATE VIEW RSULTS.SORTTempTableSorted AS
                    SELECT T.Method, T.QuantityPrice
          FROM RSULTS.BASKET_INVOICE_PAYMENT as T
;
QUIT;
Legend1
          FRAME
          POSITION = (BOTTOM CENTER OUTSIDE)
          ;
TITLE;
TITLE1 "Revenues Percentages per Payment Method";
FOOTNOTE;
PROC GCHART DATA =RSULTS.SORTTempTableSorted
;
          PIE3D     Method /
          SUMVAR=QuantityPrice
          TYPE=SUM
          LEGEND=LEGEND1
          SLICE=OUTSIDE
          PERCENT=OUTSIDE
          VALUE=NONE
          OTHER=4
          OTHERLABEL="Other"
          COUTLINE=BLACK
NOHEADING
;
RUN;
QUIT;

/* Frequency Percentages per Product Region */
PROC SQL;
          CREATE VIEW WORK.SORTTempTableSorted AS
                    SELECT T.Region
          FROM RSULTS.BASKET_PRODUCT as T
;
QUIT;
Legend1
          FRAME
          POSITION = (BOTTOM CENTER OUTSIDE)
          ;
TITLE;
TITLE1 "Frequency Percentages per Product Region";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
;
          PIE3D     Region /
          LEGEND=LEGEND1
          SLICE=OUTSIDE
          PERCENT=OUTSIDE
          VALUE=NONE
          OTHER=4
          OTHERLABEL="Other"
```

```
    COUTLINE=BLACK
NOHEADING
;

RUN;
QUIT;

/* Revenues per Product Origin */
PROC SQL;
    CREATE VIEW WORK.SORTTempTableSorted AS
        SELECT T.Region, T.QuantityPrice
    FROM RSULTS.BASKET_PRODUCT as T
;
QUIT;
Legend1
    FRAME
    POSITION = (BOTTOM CENTER OUTSIDE)
    ;
TITLE;
TITLE1 "Revenues per Product Origin";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTempTableSorted
;
    PIE3D     Region /
    SUMVAR=QuantityPrice
    TYPE=SUM
    LEGEND=LEGEND1
    SLICE=OUTSIDE
    PERCENT=OUTSIDE
    VALUE=OUTSIDE
    OTHER=4
    OTHERLABEL="Other"
    COUTLINE=BLACK
NOHEADING
;

RUN;
QUIT;








/* Question5 */

/*   It should be mentioned that the SKU of each product contains “hidden” information. The twelfth (12th)
digit indicates the promotional activity that
is attached to the product. In order to unhide this piece of information use relevant functions and then store it
to a new column. If we assume that an SKU is 58720443050301,
then the promotional activity code is 3.
• What is the percentage of products that are sold without promotion and what is the percentage of products
sold with promotion (use graphs).
```

```
• Create pie charts to show the percentage of products that are sold on each promotion type (use the
description of the promotion and not its code). Do not include
the products sold without promotion.
• How many products are sold with promotion over or equal to 15%? What is the revenue of the sale of these
products?
• Which customers buy more times products that are on promotion? Provide their demographic
characteristics (i.e. gender, age group, country).*/

data RSULTS.BASKET_DATES_SALES;
 SET RSULTS.BASKET_DATES_SALES;
 'Promotion Code'n = SUBSTR(SKU,12, 1);
 SupplierID = SUBSTR(SKU,6, 1);
run;

data RSULTS.BASKET_DATES_SALES;
  SET RSULTS.BASKET_DATES_SALES;
  if 'Promotion Code'n =0 then Promotion_Category ='No Promotion';
  else Promotion_Category ='Promotion';
  run;

PROC SORT DATA=RSULTS.BASKET_DATES_SALES;
 BY 'Promotion Code'n;
RUN;

data RSULTS.BASKET_DATES_SALES;
merge RSULTS.BASKET_DATES_SALES(IN=A) RSULTS.PROMOTION(IN=B);
by 'Promotion Code'n;
IF A AND B;
run;

proc sql noprint;
create table RSULTS.promoted as
select 'Promotion Code'n, 'Promotion Type'n
            FROM RSULTS.Basket_Dates_Sales
   WHERE 'Promotion Code'n NE '0';
quit;

data RSULTS.Basket_Dates_Sales;
            SET RSULTS.Basket_Dates_Sales;
            dis= scan('Promotion Type'n,1,'%');
run;


data RSULTS.BASKET_DATES_SALES;
set RSULTS.BASKET_DATES_SALES ;
if dis NE 'No Promotion' then Discount =input(dis,2.0);
else Discount = 0;
Real_Price = QuantityPrice*(1- Discount/100);
drop dis;
RUN;

proc sql noprint;
create table RSULTS.PRODUCTSDIS15 as
select SKU AS Upper15Dis__Sold_Products,sum(Quantity) AS Number_Of_Products,
     sum(Real_Price) AS Revenue_Of_Products
            FROM RSULTS.Basket_Dates_Sales
   WHERE Discount >= 15
   GROUP BY SKU
            ORDER BY Revenue_Of_Products DESC;
quit;
```

```
/* The demographic characteristics with promotion: age, gender and country of the
customer . */
proc sql noprint;
create table RSULTS.DemographicProm as
select CustomerID, AGE, CustomerCountry, Gender
            FROM RSULTS.BASKET_DATES_SALES
            Where 'Promotion Type'n = 'Promotion'
            ORDER BY CustomerID;
quit;


proc sql noprint;
create table RSULTS.prom_age as
select Age_Range,count(customerID) as times_customers_bought_promotion
            FROM RSULTS.Basket_Dates_Sales
   WHERE  Promotion_Category = 'Promotion'
   Group by Age_Range
            order by times_customers_bought_promotion DESC;
quit;

proc sql noprint;
create table RSULTS.prom_gender as
select Gender,count(customerID) as times_customers_bought_promotion
            FROM RSULTS.Basket_Dates_Sales
   WHERE  Promotion_Category = 'Promotion'
   Group by Gender
            order by times_customers_bought_promotion DESC;
quit;

proc sql noprint;
create table RSULTS.prom_Cust_Country as
select CustomerCountry,count(customerID) as times_customers_bought_promotion
            FROM RSULTS.Basket_Dates_Sales
   WHERE  Promotion_Category = 'Promotion'
   Group by CustomerCountry
            order by times_customers_bought_promotion DESC;
quit;




/*Question6 */


/* It should be also mentioned that the SKU of each product contains more “hidden” information. The sixth
(6th) digit indicates the company that supplied the product (supplier). In order to unhide this piece
of information use relevant functions and then store it to a new column. If we assume that an SKU is
58720443050301, then the supplier code is 4.
• Create graphs to show the percentage of products sold by each supplier (use the name of the supplier and
not its code).
• Create graphs to show the percentage and actual revenues of products sold by each supplier (use the name
of the supplier and not its code).
• Create a cross tabulation table to show the total revenue of the company with respect to the origins of the
products sold by each supplier (Use the names of the suppliers and the names of the countries
of origins and not their codes. Put the total revenue in the middle of the cross tabulation, the origin in the
rows and the suppliers in the columns). For this task you have to use proc tabulate (find relevant instructions
in the web or in sas help).   */

/* a */
```

```
PROC SORT DATA=RSULTS.BASKET_DATES_SALES;
  BY SupplierID;
RUN;

data RSULTS.BASKET_DATES_SALES;
merge RSULTS.BASKET_DATES_SALES  RSULTS.SUPPLIER;
by SupplierID;
run;


/* Pie Chart Percentage of products Per Supplier Name */

PROC SQL;
        CREATE VIEW RSULTS.SORTTEMPTABLESORTED_0000 AS
                SELECT T."Supplier Name"n
        FROM RSULTS.BASKET_DATES_SALES as T
;
QUIT;
Legend1
        FRAME
        POSITION = (BOTTOM CENTER OUTSIDE)
        ;
TITLE;
TITLE1 "Percentages of products sold per Supplier Name";
FOOTNOTE;
PROC GCHART DATA =RSULTS.SORTTEMPTABLESORTED_0000
;
        PIE3D   "Supplier Name"n /
        TYPE=PCT
        LEGEND=LEGEND1
        SLICE=OUTSIDE
        PERCENT=OUTSIDE
        VALUE=NONE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;
RUN;
QUIT;



/* b */

/* Pie Chart Percentage and Revenue of products Per Supplier Name */

PROC SQL;
        CREATE VIEW RSULTS.SORTTEMPTABLESORTED_0000 AS
                SELECT T."Supplier Name"n, T.Real_Price
        FROM RSULTS.BASKET_DATES_SALES as T
;
QUIT;
Legend1
        FRAME
        POSITION = (BOTTOM CENTER OUTSIDE)
        ;
TITLE;
TITLE1 "Percentage & Actual Revenues of Products sold per Supplier Name";
```

```
FOOTNOTE;
FOOTNOTE1 "Generated by the SAS System (&_SASSERVERNAME, &SYSSCPL) on %TRIM(%QSYSFUNC(DATE(),
NLDATE20.)) at %TRIM(%SYSFUNC(TIME(), TIMEAMPM12.))";
PROC GCHART DATA =RSULTS.SORTTEMPTABLESORTED_0000
;
            PIE3D      "Supplier Name"n /
            SUMVAR=Real_Price
            TYPE=SUM
            LEGEND=LEGEND1
            SLICE=OUTSIDE
            PERCENT=OUTSIDE
            VALUE=OUTSIDE
            OTHER=4
            OTHERLABEL="Other"
            COUTLINE=BLACK
NOHEADING
;
RUN;
QUIT;




/* c  tabulate table */


PROC SORT DATA=RSULTS.BASKET_DATES_SALES ;
 BY 'Product Origin'n;
RUN;

data RSULTS.BASKET_DATES_SALES ;
Merge  RSULTS.BASKET_DATES_SALES RSULTS.PRODUCT_ORIGIN (rename=(CODE='Product Origin'n));
by 'Product Origin'n;
run;

proc tabulate data=RSULTS.BASKET_DATES_SALES;
 class 'Supplier Name'n Region;
 var Real_Price;
 table Region='Region',
      sum=' 'Real_Price='Total_Revenue''Supplier Name'n='Supplier';
run;










/* Question7 */

/* What is the distribution of purchases per day of the week?
Is there any difference among the various days (e.g. basket size,
number of products per invoice etc). In order to find the day of the
week when the sale takes place use the weekday function. */

/* a */
data RSULTS.BASKET_DATES_SALES;
```

```
          SET RSULTS.BASKET_DATES_SALES;
          DayOfWeek = weekday(InvoiceDate);
run;
PROC SQL;
          CREATE TABLE RSULTS.BASKET_DATES_SALES_CLEAN AS
          SELECT *
          FROM RSULTS.BASKET_DATES_SALES
                    where DayOfWeek <> .;
QUIT;

 data RSULTS.BASKET_DATES_SALES_CLEAN;
  SET RSULTS.BASKET_DATES_SALES_CLEAN;
  length Day_of_the_week $12;
  if DayOfWeek = 0 then Day_of_the_week='Sunday';
  else if DayOfWeek= 1  then Day_of_the_week='Monday';
  else if DayOfWeek = 2 then Day_of_the_week='Tuesday';
  else if DayOfWeek = 3 then Day_of_the_week='Wednesday';
  else if DayOfWeek = 4 then Day_of_the_week='Thursday';
  else if DayOfWeek = 5 then Day_of_the_week='Friday';
  else if DayOfWeek = 6 then Day_of_the_week='Saturday';
run;
PROC SQL;
          CREATE TABLE RSULTS.DATE_RESULTS AS
          SELECT COUNT(InvoiceNo) AS NUMBER_OF_PURCHASES, Day_of_the_week
          FROM RSULTS.BASKET_DATES_SALES_CLEAN
          GROUP BY Day_of_the_week
          ORDER BY NUMBER_OF_PURCHASES desc;
QUIT;



/* b */

PROC SQL;
          CREATE TABLE RSULTS.DATES_BASKET AS
          SELECT COUNT(SKU) AS Basket_Size, SUM(Quantity) AS Number_of_product,
  SUM(Real_price) AS Total_revenue, Day_of_the_week
          FROM RSULTS.BASKET_DATES_SALES_CLEAN
          GROUP BY Day_of_the_week, InvoiceNo;
quit;
PROC SQL;
          CREATE TABLE RSULTS.AVERAGE_BASKET_RESULTS AS
          SELECT AVG(Basket_Size) AS avg_Basket_Size, AVG(Number_of_product) AS
avg_Number_of_product,
  AVG(Total_revenue) AS avg_Total_revenue, Day_of_the_week
          FROM RSULTS.DATES_BASKET
          GROUP BY Day_of_the_week;
quit;





/* Question8 */

 /* The company wants to profile its customers based on their importance so as
to offer them personalized services and products. The customer segmentation is asked
to be done based on the three parameters of the RFM model. Before the application of the
RFM model the RFM data set should be created. It is reminded that the RFM model is based on
```

```
the following three parameters:

Recency - How recently did the customer purchase?
Frequency - How often do they purchase?
MonetaryValue - How much do they spend?

For this task proc sql can be used. For the calculation of R, F, M the following functions will
be useful: max, sum, count and intck (For the intck use the argument week and the argument 16/12/2011
for today’s date).For the creation of the variable Monetary, the price, quantity and promotion variables
should be used. */


proc sql noprint;
create table RSULTS.RFM as

select CustomerID, intck('WEEK', max(InvoiceDate),'16DEC2011'd) AS Recency, COUNT(*) AS Frequency,
SUM(RealPrice) AS Monetary
        FROM RSULTS.BASKET_DATES_SALES
        GROUP BY CustomerID
        ORDER BY CustomerID;

quit;


/* Question 9*/

/*
 It should be underlined that in order for the cluster analysis to produce logical results the customers with
extreme values
of the variables R, F, M should be excluded from the analysis. In order to do that, descriptive statistics tasks
(e.g. proc univariate with the percentiles output)
should be used in Base SAS. After the clusters are created in SAS Enterprise Miner the RFM data set with the
newly created cluster column should be exported to a library
and then by using Base SAS Programming the demographic data (age, gender and country) of the two most
important clusters (justify why the selected ones are the most important)
should be described.
*/

PROC UNIVARIATE data=RSULTS.RFM ;
VAR RECENCY FREQUENCY MONETARY;
RUN;

DATA RSULTS.RFM;
SET RSULTS.RFM;
IF RECENCY <= 38 AND FREQUENCY <= 208 AND MONETARY <= 3619.424;
RUN;



/* clusters 2 and 9 and demographics*/


proc sql noprint;
create table RSULTS.CHOSEN_CLUSTERS as
select *
        FROM RSULTS.rfmresults_train
        where _SEGMENT_ = 2 OR _SEGMENT_ = 9;
quit;
```

```
PROC SORT DATA= RSULTS.CHOSEN_CLUSTERS;
 BY CustomerID;
RUN;
PROC SORT DATA=RSULTS.BASKET_DATES_SALES_CLEAN ;
  BY CustomerID;
RUN;
data RSULTS.CLUSTER_DEMOGRAPHICS;
merge RSULTS.CHOSEN_CLUSTERS(IN=A) RSULTS.BASKET_DATES_SALES_CLEAN(IN=B);
by CustomerID;
IF A AND B;
run;

proc sql noprint;
create table RSULTS.DEMOGRAPHICS_OF_CLST  as
select AGE_Range, GENDER, CUSTOMERCOUNTRY, _SEGMENT_
            FROM RSULTS.CLUSTER_DEMOGRAPHICS;
quit;
proc sql noprint;
create table RSULTS.DEMOGRAPHICS_OF_CLST2  as
select AGE_Range, GENDER, CUSTOMERCOUNTRY, _SEGMENT_
FROM RSULTS.CLUSTER_DEMOGRAPHICS
where _segment_=2;
quit;

proc sql noprint;
create table RSULTS.DEMOGRAPHICS_OF_CLST9  as
select AGE_range, GENDER, CUSTOMERCOUNTRY, _SEGMENT_
FROM RSULTS.CLUSTER_DEMOGRAPHICS
where _segment_=9;
quit;

/*Charts for segment 2*/

/* Customers of Cluster 2 per Country */
PROC SQL;
            CREATE VIEW WORK.SORTTEMPTABLESORTED_0001 AS
                        SELECT T.CustomerCountry
            FROM RSULTS.DEMOGRAPHICS_OF_CLST2 as T
;
QUIT;
Legend1
            FRAME
            POSITION = (BOTTOM CENTER OUTSIDE)
            ;
TITLE;
TITLE1 "Customers of Cluster 2 per Country";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTEMPTABLESORTED_0001
;
            PIE3D     CustomerCountry /
            TYPE=PCT
            LEGEND=LEGEND1
            SLICE=OUTSIDE
            PERCENT=OUTSIDE
            VALUE=NONE
```

```
    OTHER=4
    OTHERLABEL="Other"
    COUTLINE=BLACK
NOHEADING
;
RUN;
QUIT;



/* Customers of Cluster 2 per Gender */
PROC SQL;
    CREATE VIEW WORK.SORTTEMPTABLESORTED_0001 AS
        SELECT T.Gender
    FROM RSULTS.DEMOGRAPHICS_OF_CLST2 as T
;
QUIT;
Axis1
    STYLE=1
    WIDTH=1
    MAJOR=NONE
    MINOR=NONE


;
Axis2
    STYLE=1
    WIDTH=1


;
TITLE;
TITLE1 "Customers of Cluster 2 per Gender";
FOOTNOTE;
PROC GCHART DATA=WORK.SORTTEMPTABLESORTED_0001
;
    VBAR3D
     Gender
 /
    SHAPE=BLOCK
FRAME  TYPE=PCT
PCT
    LEGEND=LEGEND1
    COUTLINE=BLACK
    RAXIS=AXIS1
    MAXIS=AXIS2
PATTERNID=MIDPOINT
;
RUN;
QUIT;


/* Customers of Cluster 2 per Age Group */
PROC SQL;
```

```
        CREATE VIEW WORK.SORTTEMPTABLESORTED_0000 AS
                SELECT T.Age_Range
        FROM RSULTS.DEMOGRAPHICS_OF_CLST2_10 as T
;
QUIT;
Legend1
        FRAME
        POSITION = (BOTTOM CENTER OUTSIDE)
        ;
TITLE;
TITLE1 "Customers of Cluster 2 per Age Group";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTEMPTABLESORTED_0000
;
        PIE3D    Age_Range /
        LEGEND=LEGEND1
        SLICE=OUTSIDE
        PERCENT=OUTSIDE
        VALUE=NONE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;

RUN;
QUIT;








/*Customers of Cluster 9 per Gender */
PROC SQL;
        CREATE VIEW WORK.SORTTEMPTABLESORTED_0001 AS
                SELECT T.Gender
        FROM RSULTS.DEMOGRAPHICS_OF_CLST9 as T
;
QUIT;
Axis1
        STYLE=1
        WIDTH=1
        MAJOR=NONE
        MINOR=NONE


;
Axis2
        STYLE=1
        WIDTH=1
```

```
;
TITLE;
TITLE1 "Customers of Cluster 9 per Gender";
FOOTNOTE;
PROC GCHART DATA=WORK.SORTTEMPTABLESORTED_0001
;
            VBAR3D
             Gender
 /
            SHAPE=BLOCK
FRAME  TYPE=FREQ
PCT
            LEGEND=LEGEND1
            COUTLINE=BLACK
            RAXIS=AXIS1
            MAXIS=AXIS2
PATTERNID=MIDPOINT
;
RUN;
QUIT;



/* Customers of Cluster 9 per Country */
PROC SQL;
            CREATE VIEW WORK.SORTTEMPTABLESORTED_0001 AS
                        SELECT T.CustomerCountry
            FROM RSULTS.DEMOGRAPHICS_OF_CLST9 as T
;
QUIT;
TITLE;
TITLE1 "Customers of Cluster 9 per Country";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTEMPTABLESORTED_0001
;
            PIE3D     CustomerCountry /
            NOLEGEND
            SLICE=OUTSIDE
            PERCENT=OUTSIDE
            VALUE=NONE
            OTHER=4
            OTHERLABEL="Other"
            COUTLINE=BLACK
NOHEADING
;
RUN;
QUIT;

/* Customers of Cluster 9 per Age Group */

PROC SQL;
            CREATE VIEW WORK.SORTTEMPTABLESORTED_0001 AS
                        SELECT T.Age_Range
            FROM RSULTS.DEMOGRAPHICS_OF_CLST9_10 as T
;
```

```
QUIT;
Legend1
        FRAME
        POSITION = (BOTTOM CENTER OUTSIDE)
        ;
TITLE;
TITLE1 "Customers of Cluster 9 per Age Group";
FOOTNOTE;
PROC GCHART DATA =WORK.SORTTEMPTABLESORTED_0001
;
        PIE3D   Age_Range /
        LEGEND=LEGEND1
        SLICE=OUTSIDE
        PERCENT=OUTSIDE
        VALUE=NONE
        OTHER=4
        OTHERLABEL="Other"
        COUTLINE=BLACK
NOHEADING
;

RUN;
QUIT;


/* Question 10*/

/*The company is interested to change internally the store based on the
products that tend to be bought together. In order to apply this initiative
the company must be sure about the associations
among the product names. You are asked to find which products are bought
together (associations of product names) in the whole data set. Then find
the associations among products in the two most important
clusters (according to your business thinking) previously identified so if
a customer is found to belong in one of them to receive the most suitable/
best proposals/ offers. For this task Base SAS should be
used to filter the customers that belong to the two most important
clusters, create the two relevant data sets and then these data sets to be
analyzed using association rules through SAS Enterprise Miner.*/


PROC SORT DATA=RSULTS.BASKET_SALES;
  BY CUSTOMERID;
RUN;
PROC SORT DATA=RSULTS.rfmresults_TRAIN;
  BY CustomerID;
RUN;
data RSULTS.BASKET_SALES_RFM;
merge RSULTS.BASKET_SALES(IN=A) RSULTS.rfmresults_TRAIN(IN=B);
by CustomerID;
if a and b;
run;
proc sql noprint;
create table RSULTS.SEGMENT_2 as
select InvoiceNo, SKU, Description, CustomerId
      FROM RSULTS.BASKET_SALES_RFM
      where _SEGMENT_ = 2;
quit;
```

```
proc sql noprint;
create table RSULTS.SEGMENT_9 as
select InvoiceNo, SKU, Description, CustomerId
      FROM RSULTS.BASKET_SALES_RFM
      where _SEGMENT_ = 9;
quit;
```