Algorithms for Analyzing and Forecasting in a Pharmaceutical Company

Lucelia DEDU
Database Master – Business Support
Faculty of Economic Cybernetics, Statistics and Informatics
Bucharest University of Economic Studies, Romania
luceliadedu@yahoo.com

This paper presents some of the utilities of using SAS code for analyze and forecast, in a pharmaceutical company. First, data must be cleaned, in order to obtain representative results. Once this stage done, the input can be evaluated and models can be created in SAS, to extract the most valuable information from the initial data. To sustain management decisions, SAS code allows creating different types of reports and has the capability to recode the initial variables into new ones, but keeping the most part of the information contained, through the technique called "principal components analysis". Another advantage of SAS code is its capability of classifying the initial variables into compact class. Based on scorecards and WOE (Weight of Evidence), notions used in banking analyses, it has been created a model which evaluates the suppliers of the company and ranks them, in order to find out what contracts must be continued and which contracts must be closed.
Keywords: *Reports, Principal Components Analysis, Clusters, Scorecard, Weight of Evidence*

1 Introduction

Data analysis involves collecting, analyzing and presenting large amounts of data, in order to extract the most valuable information from the initial data and to discover patterns and trends which are used in decision-making process. This means that based on many information, it is important to know how to extract the essence from the available data and how to use it in order to improve the future activity and the results obtained.

Data analysis is a process that involves statistical algorithms or logical techniques to describe and evaluate data. Many of the analyses performed become an iterative process where data is systematically collected and analyzed simultaneously, especially because patterns are discovered through the entire data collection 0.

Subjecting data to statistical operations can relieve the relationship among variables and the level of trust the answers are getting.

Data analysis is used to process and filter the information contained in the data studied, in order to capture the essence or extract the information contained. Another purpose is to highlight this information in an essentially form, more intelligible, suggestive, simplified and synthesizers. Achieving this goal involves the development of a sequence of transformations performed on the raw data and involves the use of specific methods and techniques. These changes aim to maximize the relevance and interpretability of data and requires, among other things, to eliminate redundant or meaningless information and generality. From this point of view, data analysis appears to be a specific transformation process that takes as input the raw data and as output the information synthesized 0.

Began in the late 60's, for analyzing the results of the experiments in agriculture, **SAS** (Statistical Analysis System) became a powerful programming language, providing data mining tools for Web analysis and development, for a wide range of industries.

Nowadays, SAS solutions are used in more than 130 countries, covering areas such as Analytics, Business Intelligence, Customer Intelligence, Data Integration & ETL, Fraud Prevention & Detection, Risk Management, Supply Chain Management, Cloud etc. 0.

SAS can import different types of data and creates outputs in diverse formats, being a flexible programming language, which provides statistical, descriptive, inferential

DOI: 10.12948/issn14531305/18.3.2014.09

and predictive analysis.
The base products of SAS are: BASE SAS (data management), SAS / STAT (statistical analysis), SAS/ GRAPH (graphics), SAS/OR (optimization and simulation), SAS/ ETS (time series analysis), SAS/ QC (quality control) and many others.
In SAS, statements are used to write a program, based on instructions, executed in order. SAS statements are grouped into "blocks", referred to as "steps". In a SAS program, we can have data steps or procedure steps. A data step reads and modifies data for analysis, creating a SAS data set, while a procedure step performs a specific utility action on a data set and produces results or print reports. A data set read from an external source and can include DO loops, IF-THEN-ELSE logic and an assortment of numeric and character functions, concatenation and match-merge.
According to 0, SAS procedures do everything from simple sorting and printing to analysis of variance and 3D graphics.
SAS is known for its flexibility - there aren't many rules about how to format a SAS program: statements can be in uppercase or lowercase, they can continue on the next line (with the condition that they don't split words in two) and they can start in any column. Almost all the statements begin with a keyword which identifies the type of the statement – an exception is the assignment statement which begins with a variable name. But the most important rule remains not to omit the semicolon at the end of the statement 0.
These statements specify how data are processed and analyzed, present the operations performed on data or detail the instructions about the analysis.

2 Accessing and Cleaning Data

In SAS, data can be accessed in 3 different ways:

- Using INFILE declaration;
- Reading data directly from SAS code (for small data sets);
- Using IMPORT procedure.

Table 1. Structure of initial data set

Variable	Product	Category	Type	Unitary_price	Range_series	Stock
Variable type	Char(50)	Char(8)	Char(10)	Float(5,2)	Integer(6)	Integer(10)
Variable	Sales	Production	Term	Substitutes	Form	Active_ingredients
Variable type	Integer(10)	Integer(10)	Integer(3)	Integer(2)	Char(6)	Integer(2)

Product	Category	Type	Unitary_price	Range_series	Stock
Char(50)	Char(8)	Char(10)	Float(5,2)	Integer(6)	Integer(10)
Sales	Production	Term	Substitutes	Form	Active_ingredients
Integer(10)	Integer(10)	Integer(3)	Integer(2)	Char(6)	Integer(2)

Table 1 presents the variables analyzed. Their description is presented below:
Category refers to the status of the pharmaceutical product. This variable can have the values "OTC" - without a prescription, "Rx" - by prescription, "SN" - a dietary supplement, "Cosmetic" or "Biocides".
Type describes the formulation of the product and can have the values "Tablets", "Capsule", "Syrup", "Solution" or "Spray".
Unitary_price, Range_series, Stock, Sales, Production refer to the production.
Term refers to the number of months of validity guaranteed for each product.
Substitutes contain the number of products substituted (have the same properties and the same curative effect) for each product.
Form refers to the physical nature of each pharmaceutical product. This variable can have the values "liquid" or solid ".
Active_ingredients contains the number of active ingredients in each product, which influences the shelf life of the products.
Data cleaning is one of the most important steps in processing data, by verifying that the data are correct or at least compliant with a

DOI: 10.12948/issn14531305/18.3.2014.09

certain set of rules. Data cleaning is concerned with detecting and removing errors and incompatibilities to improve data quality. Data quality problems are present in the individual data collections such as files and databases, caused by misspellings during data entry, lack of information or other types of errors.

In order to verify the accuracy of data, it has been created formats for the analyzed variables:

```
/* Creating formats */

proc format;
  value $Cat_f 'OTC','RX','SN','Cosmetic','Biocide'=
'Valid'
            ' '    = 'Gap'
            other  = 'INVALID';
  value $Tip_f
'Tablets','Capsules','Syrup','Liquid','Spray','Powder' =
'Valid'
            ' '    = 'Gap'
            other  = 'INVALID';
  value $Price_f '0.01'-'999.99' = 'Valid'
           ' '    = 'Gap'
           other  = 'INVALID';
  value $Range_f '1'-'99999' = 'Valid'
            ' '    = 'Gap';
  value $Stock_f '0'-'999999' = 'Valid'
            ' '    = 'Gap'
            other  = 'INVALID';
  value $Sales_f '0'-'999999' = 'Valid'
            ' '    = 'Gap'
            other  = 'INVALID';
  value $Prod_f '0'-'999999' = 'Valid'
            ' '    = 'Gap'
            other  = 'INVALID';
  value $Term_f '1'-'99' = 'Valid'
            ' '    = 'Gap'
            other  = 'INVALID';
  value $Subst_f '0'-'50' = 'Valid'
            ' '    = 'Gap';
  value $Form_f 'liquid','solid' = 'Valid'
            ' '    = 'Gap'
            other  = 'INVALID';
  value $active_f '1'-'99' = 'Valid'
            ' '    = 'Gap'
            other  = 'INVALID';
run;
```

To eliminate records with duplicate values, we can use the code below:

```
proc sort data=formats.products out=temp;
  by Product;
run;

data formats.duplicate formats.nonduplicate;
  set temp;
  by Product;
  if first.Product and last.Product then output
formats.nonduplicate;
  else output formats.duplicate;
run;

title "Duplicate values: ";
proc print data=formats.duplicate;
  id Product;
run;

title "Unique values: ";
proc print data=formats.nonduplicate;
  id Product;
run;
```

3 Reports

Once the stage of accessing and cleaning data done, data can be summarized, analyzed and evaluated. Before transform data, it can be created reports to show the initial state. Regarding that data are unique, without missing values and with no incorrect format, the results obtained will be representatives.

Reports are tools for management, which influence the decision making process. SAS offers the possibility of creating such reports, parameterize and customize them, as well as save the resulted reports in PDF or HTML.

The database analyzed belongs to a pharmaceutical company, which means that the information analyzed refer to clients, manufacture, products and suppliers.

For example, we can create a report based on information about products – unit price, category, type, unit stock, unit sales etc.

To create a report about products and the situation of units, considering the category and the type for each product, we can use the code below:

```
/************************************/
/** REPORT  - CATEGORY and TYPE   ***/
/************************************/
ods pdf
file="C:\Users\DELL\Desktop\Date\Output\Report_cate
gory_type.pdf" style=sigla;
proc report data=formats.products nowd headline
headskip;
column category type product unitary_price sales stock
production;
define category/ group 'Category';
define type / group 'Type';
define product / 'Product';
define unitary_price /group 'Price';
define sales / analysis sum;
define stock / analysis sum;
define production / analysis sum;
break after category / color=red summarize suppress
skip;
break after type / color = blue summarize suppress skip;
```

DOI: 10.12948/issn14531305/18.3.2014.09

```
compute after;
line 'Total sales:' sales.sum '  lei (for 2013) ';
line 'Total stock:' stock.sum '  units  (for 2013) ';
line 'Total production :' production.sum '  units (for
2013) ';
endcomp;
compute sales;
endcomp;
compute stock;
endcomp;
compute production;
endcomp;
where sales > 0;
title "Products and total units of each product";
title2 "by category and type";
run;
ods pdf close;
```

BIOFARM

Products and total units of each product by category and type 11:10 Thursday, August 25, 2014

Category	Type	Product	Price	Sales	Stock	Production
Biocide	Liquid	BETAGIN sol.ext. 10% 500ml x1	20	2082	2329	2500
		BETAGIN sol.ext. 10% 30ml x1	6.5	13745	24485	15000
				15827	*26814*	*17500*
	Tablets	DEZINFECTANT FORTE compr. x50	5.79	3678	25436	7000
				3678	*25436*	*7000*
				19505	*52250*	*24500*
Cosmetic	Capsules	CARMOL BABY caps. x10	6.3	14299	32925	10000
		INHALANT BABY caps. x10		19861	61503	20000
				34160	*94428*	*30000*
	Liquid	OSSIDENTA Liquid 50ml x1	3.5	65400	138022	80000
		INHALANT sol.inh. 10ml x1	3.6	151580	164822	160000
		OTET AROMAT Liquid x1	3.7	12237	85240	12500
		CARMOL BABY sol.ext. 100ml x1	4.2	14771	90881	25000
		OSSIDENTA Liquid 250ml x1 Cirese.Menta	5.35	19428	50746	24000
		OSSIDENTA Liquid 250ml x1 Menta		19428	25226	16000
				282844	*554937*	*317500*
	Spray	OSSIDENTA spray 30ml x1 Menta	6.5	5732	5889	7000
		OSSIDENTA spray 30ml x1 Cirese.Menta	6.65	5732	46181	7000
				11464	*52070*	*14000*
				328468	*701435*	*361500*

Fig. 1. Report about products and the situation of stock, sales and production units, considering the category and the type for each product

Fig. 1 presents the report which lists by category and type, products and unit prices, while achieving total units sold, the total stock and total units produced. For example, in the category "Biocide", there are 2 types of products – liquid and tablets; the liquid products are Betagin sol. ext. 10% 500ml x1 and Betagin sol. ext. 10% 30ml x1 and the tablets are Dezinfectant Forte compr.x50. For every product are listed total units sold, the total stock and total units produced.

4 Scorecard for Ranking Suppliers

Scorecard is a technique that allows a company to monitor and manage performance, considering some targets. Measurements are usually financial performance, customer value, internal business processes, the performance of innovation or employee performance 0.

Suppliers have an important role in the proper functioning of the Biofarm Company. If suppliers do not fulfill their obligations under the contract, increases the risk that Biofarm reduce or even stop their work due to lack of raw materials or packaging.

Scorecard is used in activities related to risk management of non-commitment to a supplier, in order to decide to conclude a new contract or not to continue to work with suppliers for which risk and losses are related. For these reasons, it is useful to

DOI: 10.12948/issn14531305/18.3.2014.09

review suppliers that Biofarm contracts. The analyze made regarded the top 20 suppliers by value of purchases made by the company.
The analysis regards 20 vendors for which 9 attributes are known: Acquisition_value, Term (delivery time agreed in contract), Term(achieved) (delivery time realized), Product_category (the category for each product), Orders_history (the status for each order), Company_type (the type of every supplier), Contract (the length of the collaboration between Biofarm and each supplier), No_of_contracts (number of contracts closed by Biofarm with each supplier), Performance (show if a supplier is performing or not).
Fig. 2 presents the formula for WOE (Weight of Evidence), which represents the power of discrimination of each attribute and helps to differentiate the model risk and must be calculated for each variable.

$$\mathbf{WOE} = \ln \frac{\frac{\#\ Performing\ suppliers\ from\ category\ X\ for\ atribut\ Y}{\#\ Performing\ suppliers}}{\frac{\#\ Underperforming\ suppliers\ from\ category\ X\ for\ atribut\ Y}{\#\ Underperforming\ suppliers}}$$

Fig. 2. Weight Of Evidence formula

Information Value formula, shown in *Fig*. 3, has the purpose to select variables during analysis and model building. This concept reflects the overall predictive power of the variables considered, so it can be used for comparing the predictive power among competing variables.

$$IV = \sum \left(\frac{\#\ Performing\ suppliers\ from\ category\ X\ for\ atribut\ Y}{\#\ Performing\ suppliers} - \frac{\#\ Underperforming\ suppliers\ from\ category\ X\ for\ atribut\ Y}{\#\ Underperforming\ suppliers} \right) * \ln \frac{\frac{\#\ Performing\ suppliers\ from\ category\ X\ for\ atribut\ Y}{\#\ Performing\ suppliers}}{\frac{\#\ Underperforming\ suppliers\ from\ category\ X\ for\ atribut\ Y}{\#\ Underperforming\ suppliers}}$$

Fig. 3. Information Value formula

Regarding the notions explained above **Error! Reference source not found.**, the following macro has been defined in order to calculate the Weight of Evidence and the Information Value for each attribute 0:

```
/*************************************/
/*******   MACRO WOE AND IV    *******/
/*************************************/

%macro WOE(var1);
data formats.woe_suppliers;
set formats.suppliers;  run;

proc sort data=formats.suppliers out = formats.sort;
        by performance id;
run;
proc transpose data=formats.sort out=formats.sort;
        by performance id;
        var &var1;
run;
proc freq data=formats.sort;
        by _name_ performance;
        tables col1/out=formats.sort;
run;
proc sort data=formats.sort;
        by _name_ col1;
run;
proc transpose data=formats.sort out=formats.sort2;
        by _name_ col1;
        id performance;
        var percent;
run;
data formats.WOE_table  (drop =  _name_ _label_);
        set formats.sort2;
        rename col1=&var1;
        Performing=Performing/100;
        UnderPerforming= UnderPerforming/100;

/* WOE*/
        woe=log(Performing/UnderPerforming)*100;
        IV=(Performing-UnderPerforming)*woe;
run;
proc sort data = formats.suppliers;
by &var1;
run;
proc sort data = formats.WOE_table;
        by &var1;
run;
```

DOI: 10.12948/issn14531305/18.3.2014.09

```
data formats.WOE_SUPPLIERS;
            merge formats.suppliers formats.woe_table;
            by &var1;
            rename woe=&var1;
            rename IV=IV_&var1;
            drop &var1;
run;
proc sort data=formats.WOE_SUPPLIERS;
by id; run;

goptions reset=all ctitle=black ftitle=swissb
       ctext=black htext=1 htitle=2 ftext=swissb
       colors=(cx00cccc cxcd0369 cx5b768d
            cx594f4a cx008080 cxff8f71)
       transparency

iback='C:\Users\DELL\Desktop\bio2.png' imagestyle=fit;
title 'WOE -- ' &var1;
axis1 label=(&var1);
axis2 label= ('WOE');

proc gchart data=formats.WOE_table;
  hbar3d &var1 / sumvar = woe maxis=axis1 raxis=axis2
discrete;
run;
quit;
%mend WOE;

%WOE(Acquisition_value);
%WOE(Term);
%WOE(Term_achieved);
%WOE(Product_category);
%WOE(Orders_history);
%WOE(Company_type);
%WOE(Contract);
%WOE(No_of_contracts);
```

As it is shown in *Fig.* 4, the biggest negative value for the Acquisition_value variable is obtained for the "300-600 mil" category, which means that the category of dealers who have obtained orders worth 300-600 thousand, there are more underperforming suppliers than performing. The ratio of non-performing providers were able to breach of contract is lower for suppliers regarding Acquisition_value attribute than the whole portfolio.

The graphic shows that the most reliable and profitable suppliers are those who obtained orders worth more than 1 million or less than 300 thousand.

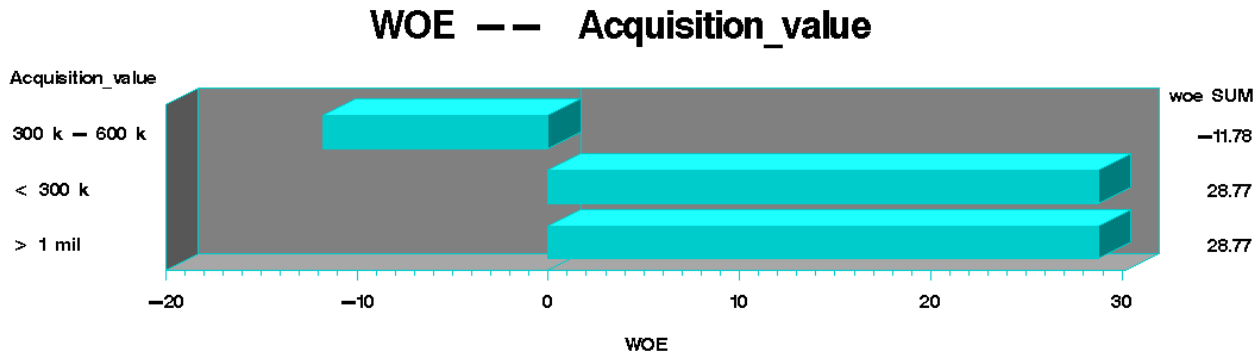

Fig. 4. Weight of Evidence for the variable Acquisition_value

As it is presented in *Fig.* 5, the biggest negative value for the variable Term is obtained for the "45" category, which means that for the category of providers who should deliver in 45 days, there are more underperforming suppliers than performing ones. The chart above shows that the most profitable are providers who delivered the order within 30 days.

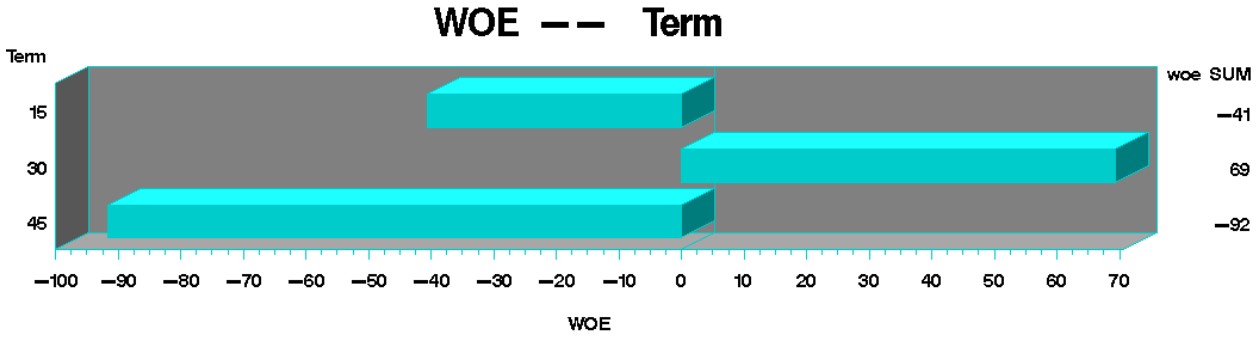

Fig. 5. Weight of Evidence for the variable Term

DOI: 10.12948/issn14531305/18.3.2014.09

According to 0 and 0, in order to eliminate the missing values from the table created, the code below must be run:

```
proc stdize data=fz.WOE_FZ reponly missing=0
out=fz.WOE_FZ;
var _numeric_; run;
data fz.WOE_global; set fz.WOE_FZ; drop performant;
drop neperformant; IV=IV_Val_Achizitii +
IV_Termen_plata + IV_Termen_realizat +
IV_Cat_Produse + IV_Istoric_plata + IV_Tip_societate
+ IV_Durata_Contract + IV_Nr_Contracte; run;
```

ID	Supplier	IV
1	Bostocke Trading LTD	50.028
2	Teva Czech Industries s.r.o.	88.820
3	XIAMEN FOREVER GREEN SOURCE BIOCHEM	60.843
4	Colorcon Limited	61.496
5	Indena SAS	95.638
6	PHYTOPHARM KLEKA SA	93.675
7	Selectchemie AG	135.388
8	S & D Chemicals LTD	112.066
9	ICE SpA	131.113
10	AIS PRODIMPEX S.R.L.	104.568
11	BASF CHEM TRADE GMBH	85.393
12	MORGAN THORPE SA	113.077
13	Linnea SA	39.316
14	DSM Nutritional Products Europe LTD	37.192
15	Barentz Romania SRL	70.032
16	TIPOGRAFIA POLIROM SRL	114.251
17	Kunststoffwerk Kremsmunster GMBH	83.977
18	ARGOROM PLASTICS	42.971
19	SGD(SAINT GOBAIN)	99.265
20	ONGROPACK KFT.BC-ONGROPACK LTD	126.097

Fig. 6. Information Value, for every supplier

By summing the Information Value for each attribute, as in *Fig.* 6, Information Value can be calculated globally as running the following code, suggested in 0:

```
proc sql;
create table formats.RANK
as select id, supplier, performance, IV from
formats.WOE_global;
run;

proc rank data= formats.RANK out= formats.RANK
descending;
var iv;
ranks rank_number;
run;

proc sort data= formats.RANK out= formats.RANK;
by rank_number;
run;
data formats.RANK2;
set formats.RANK;
rename rank_number = rank;
run;

proc template;
   define style sigla;
   parent=styles.printer;
   style header from header /
        background=_undef_;
   style body from document /
        background=_undef_

backgroundimage="C:\Users\DELL\Desktop\sigla.jpg";
end;
run;

ods pdf
file="C:\Users\DELL\Desktop\Date\Output\Hierarchy.pdf"
style=sigla;

title 'Suppliers hierarchy';
title2 ' - using Information Value criteria -';

proc print data= formats.RANK;
run;
goptions reset=all ctitle=black ftitle=swissb
       ctext=black htext=1 htitle=2 ftext=swissb
       colors=(cx00cccc cxcd0369 cx5b768d
            cx594f4a cx008080 cxff8f71)
       transparency

iback='C:\Users\DELL\Desktop\bio2.png' imagestyle=fit;
title 'Suppliers hierarchy';
axis1 label=('Rank');
axis2 label= ('Information Value');

proc gchart data= formats.RANK;
  hbar3d rank_number supplier / sumvar = iv maxis=axis1
raxis=axis2 discrete;
run;
quit;

ods pdf close;
```

Based on nine attributes recorded for the 20 suppliers, the Weight of Evidence has been calculated for each attribute and the information has been reflected by each feature separately. Considering the products, their category, type, unitary_price, range_series, stock, sales, production, term, substitutes, form and active_ingredients, the Information Value compares the predictive power across those 9 attributes and the Weight of Evidence reflects how behaves every variable considered.

As shown in Fig. 7, the information aggregated globally has revealed a hierarchy of suppliers, after descending the Information Value field.

DOI: 10.12948/issn14531305/18.3.2014.09

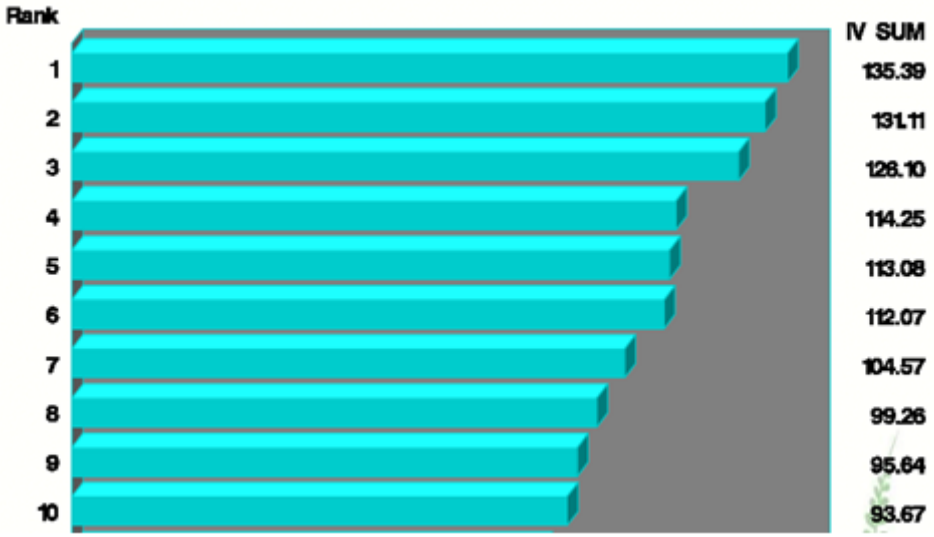

Fig. 7. Suppliers hierarchy, by Information Value

The report in *Fig.* 8 highlight the suppliers with whom Biofarm need to maintain their collaboration, due to good results obtained in the past.

The report presented above facilitates management decisions, providing a clear situation of the suppliers, considering the variables analyzed.

BIOFARM

Suppliers hierarchy
- using Information Value criteria -

Obs	ID	*Supplier*	IV	rank_number
1	7	Selectchemie AG	135.388	1
2	9	ICE SpA	131.113	2
3	20	ONGROPACK KFT.BC-ONGROPACK LTD	126.097	3
4	16	TIPOGRAFIA POLIROM SRL	114.251	4
5	12	MORGAN THORPE SA	113.077	5
6	8	S & D Chemicals LTD	112.066	6
7	10	AIS PRODIMPEX S.R.L.	104.568	7
8	19	SGD(SAINT GOBAIN)	99.265	8
9	5	Indena SAS	95.638	9
10	6	PHYTOPHARM KLEKA SA	93.675	10
11	2	Teva Czech Industries s.r.o.	88.820	11
12	11	BASF CHEM TRADE GMBH	85.393	12
13	17	Kunststoffwerk Kremsmunster GMBH	83.977	13
14	15	Barentz Romania SRL	70.032	14
15	4	Colorcon Limited	61.496	15
16	3	XIAMEN FOREVER GREEN SOURCE BIOCHEM	60.843	16
17	1	Bostocke Trading LTD	50.028	17
18	18	ARGOROM PLASTICS	42.971	18
19	13	Linnea SA	39.316	19
20	14	DSM Nutritional Products Europe LTD	37.192	20

Fig. 8. Suppliers hierarchy, by Information Value

As stated in the report above, top 5 suppliers for Biofarm are Selectchemie AG, ICE SpA,

DOI: 10.12948/issn14531305/18.3.2014.09

Ongropack Kft BC, Tipografia Polirom SRL and Morgan Thorpe SA. Considering the results obtained, Biofarm will continue to collaborate with those companies and can close the contracts with the suppliers which didn't fulfill their responsibilities, such as Xiamen Forever, Bostocke, Argorom Plastics, Linnea SA or DSM Nutritional Products Europe LTD.

5 Conclusions

In the beginning of this article, SAS has been presented as an important data analysis software tool, dedicated to highlight the power and importance of using such programs for statistical analysis of the data in a company.

Following, it has been shown the importance of data cleaning and validation; this phase cannot miss because the analyze need accurate data in order to obtain representative results. For that, it has been created data validation formats and the records which didn't comply have been removed.

Once the stage of accessing and cleaning data done, data can be summarized, analyzed and evaluated. Before transform data, it can be created reports to show the initial state. Regarding that data are unique, without missing values and with no incorrect format, the results obtained will be representatives.

Reports defined in SAS allow visualization of key information about products and are useful tools for sustaining management decisions. SAS offers the possibility of creating such reports, parameterize and customize them, as well as save the resulted reports in PDF or HTML.

Scorecard created for the company's suppliers has the purpose to sustain management decisions. The hierarchy resulted takes into account the performance indicators obtained in the last year, and historical data, such as length collaboration between the company and suppliers, the number of contracts and the degree of respect the contract by each supplier.

References

[1] F. D. Delwiche and S. J. Slaughter, "The Little SAS Book: A Primer: A Programming Approach", 2012

[2] G. Ruxanda, "Data Mining", Bucharest, 2010

[3] Information about SAS products, available:http://www.sas.com/en_us/company-information.html#stats

[4] G. Der and B. S. Everitt, "A Handbook of Statistical Analyses using SAS", 2002

[5] M. Evans, "SAS Manual For Introduction to the Practice of Statistics", 2010

[6] Information about variable reduction in SAS, using WOE (Weight of Evidence) and IV (Information Value): http://support.sas.com/resources/papers/proceedings13/095-2013.pdf

[7] Information about WOE (Weight of Evidence) and IV (Information Value): http://documentation.statsoft.com/STATISTICAHelp.aspx?path=WeightofEvidence/WeightofEvidenceWoEIntroductoryOverview

[8] F. E. Pratter, "Web Development with SAS® by Example", 2006

[9] J. Shostak, "SAS® Programming in the Pharmaceutical Industry", 2005

[10] R. Cody, "Learning SAS® by Example - A Programmer's Guide", 2007

[11] SAS Institute Inc. Step-by-Step Programming with Base SAS® Software, 2001

DOI: 10.12948/issn14531305/18.3.2014.09

Lucelia DEDU has graduated the Faculty of Economic Cybernetics, Statistics and Informatics from the Bucharest Academy of Economic Studies in 2012 and she has graduated in 2014 the courses of "Databases for Business Support" master program organized by the Academy of Economic Studies of Bucharest. During her second year at the master program, she followed the course Credit Risk Management, held by BCR, where she discovered new features of using SAS code. Her career is centered on payroll domain, database's queries, ETL Tools for data warehouses and COBOL development. Her areas of interest are: Data Warehouses, Data Modeling, Risk Management, Databases and SAS development.

DOI: 10.12948/issn14531305/18.3.2014.09