MASTER'S DEGREE THESIS

Interuniversity Master in Statistics and Operations Research

Title: Business Intelligence's Self-Service tools evaluation

Author: Jordina Orcajo

Advisor: Pau Fonseca

Department: Statistics and Operative Research

University: UPC-UB

Academic year: 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya

Master's degree thesis

Business Intelligence's Self-Service tools evaluation

Jordina Orcajo Hernández

Director: Pau Fonseca

Department of Statistics and Operational Research

1 Abstract

This project proposes a comparison analysis between four different tools, called Self-Service tools, from the Business Intelligence area. The comparison was done adapting a Systemic Quality Model, already, formalized and using a database simulated with R. In order to assess the quality of this tipe of software, seven (7) characteristics and eighty-two (82) metrics were considered.

Index

2 Introduction

This study belongs to the business sector. In particular the Business Intelligence sector, where I participated doing this Master's degree thesis.

Business Intelligence (BI) is the name associated to the set of tools and techniques for the transformation of raw data into meaningful and useful information for business analysis purposes. BI technologies are capable of handling large amounts of unstructured data to help identify, develop and otherwise create new strategic business opportunities. And the main goal of BI is to allow the easy interpretation of these large volumes of data. In particular, the Self-Service BI aims to boost that the company is able to get useful information from their own data.

The idea behind deploying self-service software, is to empower business people to analyze and understand data without specialized expertise. There are many benefits that can be derived through the implementation of a self-service BI system. Functional workers can make, faster, better decisions because they no longer have to wait during long reporting backlogs. At the same time, technical teams will be freed from the burden of satisfying end user report requests, so they can focus their efforts on more strategic IT initiatives.

There are many Self-Service BI tools in the market, and before recommending a particular one, a depth analysis of the available tools on the market must be done, according to own requirements. And because of this, the aim of this thesis is to build a comparative assessment of Self-Service BI tools, adapting a Systemic Quality Model (SQMO) and apply this methodology in the evaluation of four (4) particular tools.

In order to accomplish this, first of all we had to learn how to use Self-Service BI tools in order to know its operation, what they can do and understand how useful they are for the BI sector. Knowing, with a minimum level of depth, tools in order to evaluate them, demands spending much time in addition to technical and functional knowledge. And because of this , we have done this work together with the department of Business Intelligence from INDRA S.A and under the tutelage of Dr. Pan Fonseca. Secondly, we adapted the SQMO to particular aims and finally four (4) tools were evaluated. They were Tableau, MicroStrategy Analytics, QlikView and SAP Lumira.

As it has been pointed before, in the BI world there are many Self-Service tools, making this thesis interesting within this sector, because, probably, not all of them fulfil the requirements for all type of projects. It has to take into account that the concept "best tool" is difficult to apply in this ambit. And for this reason, it is more usual to choose an appropriate solution for a particular project.

At this moment, many consultant companies are interested in knowing which are the tool/s closer to their clients' requirements. Particularly, INDRA was interested in determine which of the four (4) evaluated tools is/are closer to its clients' requirements. INDRA S.A was also interested to apply this evaluation method on further comparisons, with other Self-Service BI tools. It means that, from this thesis, can result an applicable method to determine which tool has to be chosen in each particular project.

Nowadays, the term Business Intelligence it is also known as Business Analysis (BA). This change is due to the implemented techniques added in order to extract more information from

business data. BA is defined as the skills, technologies and practices for continuous iterative exploration and investigation of past business performance to gain insight and drive business planning, based on data and statistical methods. To gain future vision of the business, predictive modelling takes an important role. It helps to get different scenarios depending on different possible business paths. The implementation of predictive modelling can be considered the biggest difference between Business Intelligence and Business Analysis. Although predictive techniques are not in the pure definition of Business Intelligence, offering predictive techniques will be positively evaluated on this thesis, because it is considered that those tools must also evolve with the needs and interests of the companies.

2.1 Approach

To carry out an assessment, a series of steps must be followed. First of all, the responsible of preparing the assessment known as the evaluator, must know the area of use. Then, the evaluator has to fix a methodology and adapt it to the particular area of use. The adaption implies decide the interesting metrics which will be evaluated. Users can advice to the evaluator about the interesting metrics, and the evaluator have to design a questionnaire to enclose the interesting metrics. Additionally, the evaluator has to fix the area of application, in order to not misuse the methodology. Following, evaluator has to send a questionnaire to the users, in order to get opinions from experienced people in the area. Moreover, the evaluator has to provide every item required to do the evaluation (questionnaires, data, applications ...). Finally, the evaluator collects the questionnaires and proceeds to evaluate the results according to the chosen methodology .

In this particular thesis, the used methodology consists in the adaption of a Systemic Quality Model (SQMO), which is a model to evaluate software. The Systemic Quality Model and its adaption is explained, in detail, in **chapter 3**. On the other hand, in **chapter 4**, there is the method used to select the applications, which can be evaluated. Finally, in order to evaluate particular tools, data and a questionnaire, which should be provided to users, were built. A database called **20141220_Initial_test** was simulated, and it is explained in **chapters 5.** Moreover, the R scripts built to simulated it are in **Annex 1**. The questionnaire, resulting on the adaption of the SQMO, is in **Annex 2**. Finally, the answered questionnaires were analyzed, and the results are explained in **chapter 6**. Finally, in **Annex 7**, there some graphs built by Self-Service BI tools in order to introduce them to the reader.

Recalling, that the first step in a assessment is to know the area of use, and in this particular case it is the Business Intelligence area. For this reason, the terminology used in the thesis can be specific from the BI area. And reading the following **sub-chapter 3.2** is recommended to understands the terminology used a long the thesis.

2.2 Introduction to BI systems

The objective of the following chapter is to introduce the Business Intelligence terminology in order to ease the interpretation of the thesis.

When a business needs to analyse its data in order to profit them and extract information and take advantage of this, business intelligence takes the role. Most of the companies generates data and these data are stored in databases.

A database is an organized collection of data, where data are typically organized in a specific way to ease the queries. In this area, a query is a set of commands that the user types, in a specific database language, in order to get specific information about the information stored in the database. For example, a query can be*: How many male clients are in our database?* Queries are launched from a system responsible to access to the database. These types of program are called Database Management Systems (DBMS). Apart from storing data they can also modify data. We can say that it is the connector between the data storage and the user. Some examples of DBMS are Oracle, My SQL, Microsoft Access. There are a large number of database languages like SQL, QUEL, ISBL, SPL, XQuery...The use of a specific database language depends on the target database.

The most common database language is SQL, and it is used in relational databases. Relational database is a type of database that organizes data into tables, and links them, based on defined relationships. These relationships enables users to retrieve and combine data from one or more tables with a single query.

In the relational model, every row must have a unique identification or primary key on the data. For example a social security account number (SSAN) can be a key that uniquely identifies each row. The relations between tables are done by this primary key field, which uniquely identifies rows. In chapter 6 relational data model is explained in more detail.

Nowadays, there are other ways of storing data besides the relational database model, although it is still the most used. The need for the data to be well-structured, actually has become a substantial burden with extremely large volumes, with result the decline on performance as size gets bigger.. Thus, relational DBMS is generally not thought of as a scalable solution to meet the needs of 'big' data. Other database infrastructure has appeared, they are called NoSQL, which represents a completely different framework of databases. Unlike relational databases that are highly structured, NoSQL databases are unstructured in nature, trading off stringent consistency requirements for speed and agility. Unstructured data may be stored across multiple processing nodes, and often across multiple servers. This distributed architecture allows NoSQL databases to be horizontally scalable; as data continues to explode, just add more hardware to keep up, with no slowdown in performance.

A *Hadoop* cluster is a special type of distributed architecture designed specifically for storing and analysing huge amounts of unstructured data in a distributed computing environment with fast processing. Basically *Hadoop* is a distributed file system (HDFS), which lets storing large amount of data files on a cloud of machines. On top of that distributed file system, *Hadoop* provides an API (Application Programming Interface) for processing all stored data, it is called Map-Reduce. The basic idea of Map-Reduce is that each node processes the data stored on itself, and by this way, data are not transferred over the network and time is not wasted.

Moreover, to access on data in *Hadoop* environment it exists Hive. Hive is a data warehousing package/infrastructure built on top of *Hadoop*. It provides an SQL dialect, called Hive Query Language (HQL) for querying data stored in a *Hadoop* cluster. Hive adds extensions to provide better performance in the context of *Hadoop* and to integrate with custom extensions and even external programs.

After introducing the most common ways to store data, it is time to introduce the tools to analyse business data. Business intelligence tools are a type of application software designed to retrieve, analyse, transform and report data for business intelligence. In order to get data, they

can be connected to data sources or can import data from files. The connection to a data source must be done using an API which translates the queries of the BI tool to queries in the particular database language. Since now, some of these tools were complex and technical users were the responsible to use them. Currently, Self-Service BI tools have been appeared, which have revolutionized the BI world. The main targets of Self-Service BI tools is to ease the analysis of data, do fast analysis and get understandable information beneficial for the business. In order to get easy analysis they generally have intuitive interfaces. To get fast analysis, they load data in RAM (Random Assigned Memory) memory and the tool works on it. By this way, they do not need to access data in the hard disk, which implies spending more time. Finally, they offer a huge variety of graphs, and many ways to report the results attractively.

Data analysis are done by creating new fields, filtering data, visualizing data in graphics or grids,...When a user creates data visualizations like graphics or grids, in many of the Self-Service BI tools, the user can interact with them, easing the discovery of the characteristics of data. User creates graphics in interactive sheets called "dashboards" (there can be many dashboards in the same project). The graphics can be exported as static images, or user can save the whole project to interact with the graphs. Most of Self-Service BI tools, have the option to report the results, creating a PDF file with graphics, text added by the user, links, images, etc. Moreover, some of this tools, offers mobile applications to visualize and interact with dashboards on tablets or mobiles.

Finally, most of these tools, offers the option of Server Editions. This type of editions can be implemented in corporative servers, which usually are more powerful than users systems and several users can profit of this power. Moreover, these editions allows to define and manage security devices with password protection and user permissions. Finally, Server Editions ease the sharing of projects. Additionally to Server Editions, some tools offers the SaaS version (Software as a Service). SaaS business intelligence is a delivery model in which applications are typically deployed outside of a company's firewall at a hosted location and accessed by an end user, which only requires a secure Internet connection and a browser. That is, SaaS BI lets companies use BI tools without having to install a program on a computer, independently of the Operative System requirements, which sometimes might suppose a problem.

There are different type of Self-Service BI users, which are explained in the following **sub-chapter 3.3**. As every assessment must be based on the users' opinion, it is important to know the type of users of Self-Service BI tools.

2.3 BI users

A rigorous evaluation should be done by several users in order to get trustworthy results. In particular, Self-Service BI tools, as data systems, usually have different user profiles and several users of each type should evaluate the tools, from their particular point of view.

There are three different profiles of user in data systems, according to (Inmon, Imhoff, & Susa, 1998).

Farmers: They access to information in an absolutely predictable and repetitive way. We could say that they have their parcel of information and they cultivate and extract profit from this, regularly. They do not access to huge amount of data (because they do not leave the parcel) and they usually ask for aggregated data. These users usually use OLAP (Online Analytical

Processing) tools which are focus on non-informatics users. They are simple and their main objective is the data visualization. As farmers, there are employers, providers and customers to whom the organization offers informational services. Currently, Business Intelligence, which promotes the use of these systems at all levels of the organization, allows business users to use data and information in business processes naturally, without having to leave their applications.

Explorer: At the contrary of farmers, explorers have totally unpredictable and irregular accesses. They spend much time planning and preparing their studies and when they have everything ready, they start to explore a lot of such detailed information as possible. They really do not know exactly what are looking until they find it, and in any case the results are guaranteed. However, sometimes they find something really interesting that clearly improves business. They are also known as power users. Thanks to big data, explorers become data scientist. A data scientist has to be able to extract information from large volumes of data according to a not randomly clear business objective, and then present it in a simple way to the non-expert users in organization. Therefore, it consists in a cross profile with skills in computer, mathematics, statistics, data mining, graphic design, data visualization and usability.

Tourist: Usually they are a group of two or more people. On one side there is a person with an overview of the company that comes up with the possibility of a study on a certain topic. On the other hand, there would be a computer expert, knowing the systems analysis of the company and he is the manager to find out if the study is feasible with the data and available tools. This team will access to data without following any pattern and rarely observe same data twice. Therefore, their requirements cannot be known priori. Tools used by tourist are browsers or search engines (to search both data and metadata) and the result of their work will be projects carried out by *Farmers* or *Explorers*. In short, a tourist is a casual user of the information.

This project aims to develop a method of evaluation that should be applicable taking into consideration the different profiles of tool users. For example, if the tool is used by *Farmers* and *Explorers*, some *Farmer* users should evaluate the tool and the same number of *Explorer* users should evaluate the tool. After that, a mean is done with the results. In this project, as an approximation, an evaluation by one *Explorer* user is done, but it is important to do not interpret the results as concluding results.

3 Methodology

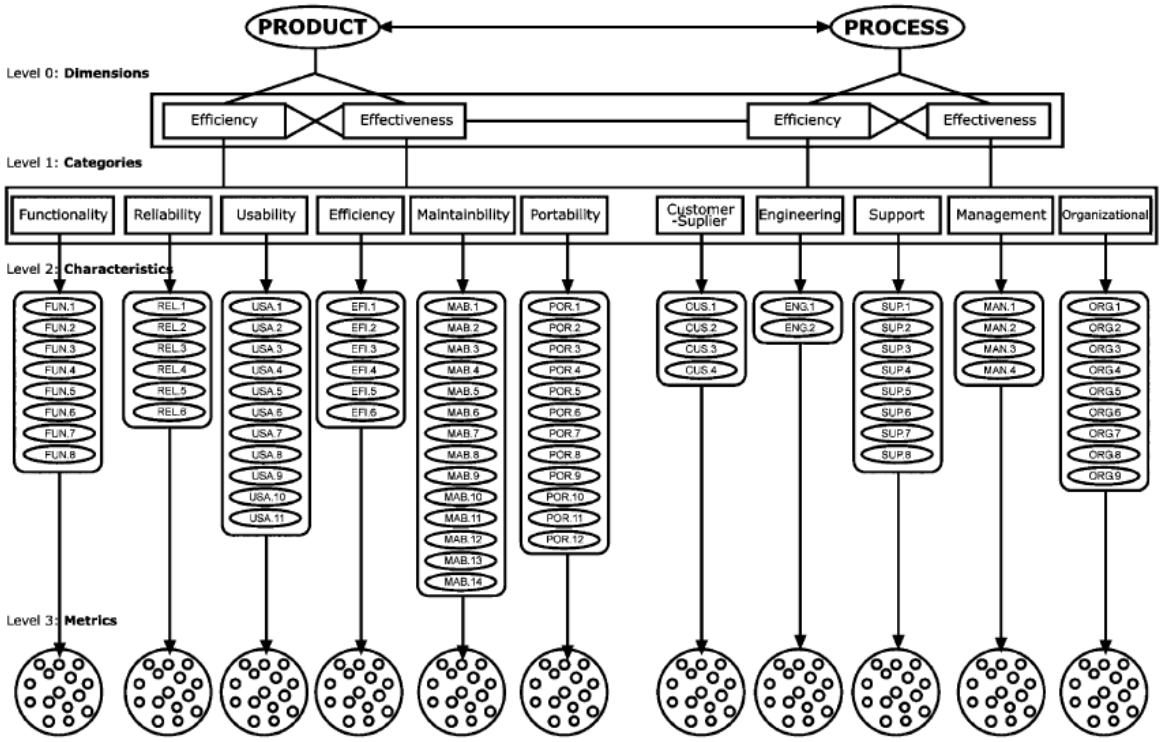


Fig. 1 Diagram of the systemic quality model (SQMO) (Callaos & Callaos, 1996)

3.1 The systemic quality model (SQMO)

The Systemic Quality Model (SQMO) was developed in 2001 by the Universidad Simón Bolívar (Venezuela). Since then, the University adopted the SQMO for software evaluations and provided successfully implementation examples (Mendoza, Pérez, & Grimán, 2005), (Rincon, Alvarez, Perez, & Hernandez, 2005).

Until then, it existed several models to evaluate the product software and other to evaluate the process software, but any one with the capability to evaluate both aspects accurately. As Humprey (1997) said, having different models capable of measuring individually the product quality or the process quality of software does not guarantee the total systemic quality of the software. As seen in **¡Error! No se encuentra el origen de la referencia.** ***¡Error! No se encuentra el origen de la referencia.***, SQMO consists of two sub-models (a *Product* and a *Process* sub-models). The SQMO can use either the *Product* sub-model, the *Process* sub-model, or both. The first sub-model is designed to evaluate the already developed software, while the second is designed to evaluate the development process of the software. The SQMO sub-models have different levels in order to assess software, which they are:

3.1.1 Level 0: dimensions

There are two dimensions for each sub-model. These dimensions are *Efficiency* and *Effectiveness* for the *Product*; and *Efficiency* and *Effectiveness* for the *Process*. *Effectiveness* is

the capability of producing a desired result, while *Efficiency* is the capability to produce a specific outcome effectively with a minimum amount or quantity of waste, expense, or unnecessary effort.

3.1.2 Level 1: categories

There are six elements corresponding to *Product* and five corresponding to *Process*. They are concretized by (Callaos N. C., 1993).

The categories for the *Product* sub-model are the followings:

- *Functionality* is the ability of a software product to provide functions that meet specific and implicit needs when software is used under specific conditions.
- *Reliability* is the capacity of a software product to maintain a specified level of performance when used under specific conditions.
- *Usability* is the capacity of a software product to be attractive, understood, learned and used by the user under certain specific conditions.
- *Efficiency* is the capacity of the software product to perform adequately under specific conditions, depending on the amount of resources used, under stated conditions.
- *Maintainability* is the capacity of the software to be modified. Modifications can include corrections, improvements or adaptations of the software to adjust to changes in the environment, in terms of functional requirements and specifications.
- *Portability* is the capacity of the software product to be transferred from one environment to another.

And the categories for the *Process* sub-model are the followings:

- *Client-Supplier* is made up of processes that have an impact on the client, support the development and transition of the software to the client, and give the correct operation and use of the software product or service.
- *Engineering* consists of processes that directly specify, implement or maintain the software product, its relations to the system and documentation on it.
- *Support* consists of processes that contain practices of a generic nature that can be used by anyone managing any kind of project or process, within a primary life cycle.
- *Management* consists of processes that contain practices of a generic nature that can be used by anyone managing any kind of project or process, within a primary life cycle.
- *Organizational* contains processes that establish the organization's commercial goals and develop process, product and resource good (values) that will help the organization attain the goals set in the project.

3.1.3 Level 2: characteristics

SQMO states that each category has associated a set of characteristics, which define the key points that must be fulfilled in order to guarantee and control de *Product* and /or *Process* quality of software. *Product* characteristics are specified in **Tab. 1** and *Process* characteristics in **¡Error! No se encuentra el origen de la referencia.**. They are defined more accurately in (Mendoza, Pérez, & Grimán, 2005).

Category	Characteristics	
	Product Effectiveness	**Product Efficiency**
Functionality	Fit to purpose Precision Interoperability Security	Correctness Structured Encapsulated Specified
Reliability	Maturity Fault tolerance Recovery	Correctness Structured Encapsulated
Usability	Ease of understanding Ease of learning Graphical Interface Operability Conformity of standards	Complete Consistent Effective Specified Documented Auto-descriptive
Efficiency	Execution performance Resource utilization	Effective No redundant Direct Used
Maintainability	Analysis Capability Ease of changing Stability Testability	Attachment Cohesion Encapsulated Software maturity Structure information Descriptive Correctness Structural Modularity
Portability	Adaptability Installation capability Co-existence Replacement capability	Consistent Parameterized Encapsulated Cohesive Specified Documented Auto-descriptive No redundant Auditing Quality management
	Data Quality -both dimensions-	

Tab. 1 Characteristics for Product sub-model

Category	Characteristics	
	Process Effectiveness	**Process Efficiency**
Customer-Supplier	Acquisition System or Software product Requirements determination	Supply Operation
Engineering	Development	Maintenance of software and systems
Support	Quality assurance Joint review Auditing Solving problems	Documentation Configuration management Verification Validation Joint review Auditing Solving problems
Management	Management Quality management Risk management	Management Project management Quality management Risk management
Organizational	Organizational Alignment Management of change Process improvement Measurement Reuse	Establishment of the process Process evaluation Process improvement HHRR management Infrastructure

Tab. 2 Characteristics for Process sub-model

3.1.4 Level 3: Metrics

Each characteristic has a group of metrics to be evaluated. They are the evaluable attributes of the product and the process and they are not agreed because they vary depending on each study case. Metrics are defined, for our particular case in **sub-chapter 3.4**.

3.2 Algorithm

The algorithm to measure the systematic quality by the SQMO, referenced *in Mendoza, L. E., Pérez, M. A., & Grimán, A. C. (2005)* is the following explained. First of all the Product Software is measured and then the Development Process.

3.2.1 Product software

The first measured category must be always *Functionality*. If the category *Functionality* is satisfied, the evaluation continues with other categories. If the product does not meet the *Functionality* category, the evaluation is ended. It is because the functional category is the most

important in the quality measuring, given that *Functionality* identifies the software capability to fit to purpose for it was built.

After that, a sub-model is adapted depending on the requirements. Two categories from the five remaining must be selected, which should be satisfied by the product and evaluated. The algorithm recommends working with a maximum of three product characteristics (including *Functionality*) , because if more than three product features are selected , some might conflict . In this sense, (Bass, Clements, & Kazzman, 1998) indicates that the satisfaction of quality attributes can have an effect, sometimes positive and sometimes negative , on meeting other quality attributes .

*(The definition of satisfaction can vary depending on the case of use and it is not fixed by the methodology. In **sub-chapter 3.3.2**, this issue is discussed).*

Finally, to measure the quality product of the software there is shown **Tab. 3**, in which there are the quality levels related with the satisfied categories.

Functionality	Second category	Third category	Quality level
Satisfied	*No satisfied*	*No satisfied*	*Basic*
Satisfied	*Satisfied*	*No satisfied*	*Medium*
Satisfied	*No satisfied*	*Satisfied*	*Medium*
Satisfied	*Satisfied*	*Satisfied*	*Advanced*

Tab. 3 Quality levels for the *Product Software*

Once the evaluation of the product software has ended, recalling that only if the quality level is at least *basic*, the *Development Process* evaluation may start.

3.2.2 Development Process

In order to evaluate the *Development Process* there are 4 steps to follow. The algorithm used in the *Development Process* evaluation is fixed, unlike the *Product Software* evaluation. The steps are the followings:

1. Determining the percentage of N/A (Not applying) answers in the questionnaire for each category. If this percentage is greater than 11% appliance of the measuring instrument must be analysed and the algorithm stops. Otherwise, the step two is the next.
2. Determining the percentage of N/K (Not knowing) answers in the questionnaire for each category. If this percentage is greater than 15%, it shows that there exist a high level of ignorance for the activities of the particular category. If the percentage is lower, step three is the next.
3. Determining the satisfaction level for each category . *(The definition of satisfaction can vary depending on the case of use and it is not fixed by the methodology. In **sub-chapter 3.3.2**, this issue is discussed).*
4. Measuring the quality level of the process. The quality levels related with the satisfied categories are:

   ***Basic level*:** It is the minimum required level. Categories *Customer-Supplier* and *Engineering* are satisfied.

***Medium level*:** In addition to the basic level categories satisfied, categories *Support* and *Management* are satisfied.

***Advanced level*:** All categories are satisfied.

Quality Levels			Category satisfied
Advanced	**Medium**	**Basic**	Customer-Supplier
			Engineering
		Support	
		Management	
		Organizational	

Tab. 4 Quality levels for *Development Process*

Finally, there must be a join between the product quality measuring and the process quality measuring, in order to obtain the systematic quality measuring. The systemic quality levels are proposed in **Tab. 5**.

Product quality level	Process quality level	Systemic quality level
Basic	-	*Null*
Basic	*Basic*	*Basic*
Medium	-	*Null*
Medium	*Basic*	*Basic*
Advanced	-	*Null*
Advanced	*Basic*	*Medium*
Basic	*Medium*	*Basic*
Medium	*Medium*	*Medium*
Advanced	*Medium*	*Medium*
Basic	*Advanced*	*Medium*
Medium	*Advanced*	*Medium*
Advanced	*Advanced*	*Advanced*

Tab. 5 Systemic quality levels

This method of measurement is responsible for maintaining a balance between the sub-models (when they are both included in the model).

3.3 Adoption of the systemic quality model (SQMO)

SQMO was adopted as reference because it is a complete work influenced by many other models. First of all, it respects the concept of Total Quality Systemic from (Callaos & Callaos, Designing with a systemic total quality, 1996). It also considers the balance between the *Process* and *Produc*t sub-models proposed by (Humphrey, 1997). These sub-models are based on the *Product* and *Process* Quality models from (Ortega, Pérez, & Rojas, 2000) and (Pérez, Rojas, Mendoza, & Grimán, 2001), respectively. Moreover, the product quality categories are based on the work of (Dromey, 1996) and the international standard ISO/IEC 9126 (JTC 1/SC 7, 1991). And the process categories are extracted from the international standard ISO/IEC 15504 (ISO IEC/TR 15504-2, 1998).

Some authors as Kitcheman (Kitchenman, 1996) have pointed out that when characteristics are complex can be divided into a set of some simpler and a new level, sub-characteristics, can be created. In that particular case, sub-characteristics have been considered in order to gain clarity.

In order to adapt the SQMO to each particular case, there must be decided which sub-model is considered (*Product*, *Process* or both), which dimension (*Efficiency* or/and *Effectiveness*), which sub-characteristics for each characteristic and which respective metrics should be evaluated.

In the current evaluation, only the *Product* sub-model of SQMO was considered. The *Process* sub-model is excluded, because our intent is to evaluate the fully already developed tools as a future tool used for the BI workforce. For this reason, *Process* sub-model is not considered. Moreover, only *Effectiveness* dimension was considered because the special attention was focused on the evaluation of the software quality features observed on its execution. But, if anyone ever considers appropriately to include the sub-model *Process* or the *Efficiency* dimension, due to his owns interests, there exists the option to do so, by following the steps explained above.

Hence, **Fig. 2** reflects the adapted model used in the current evaluation for BI tools.

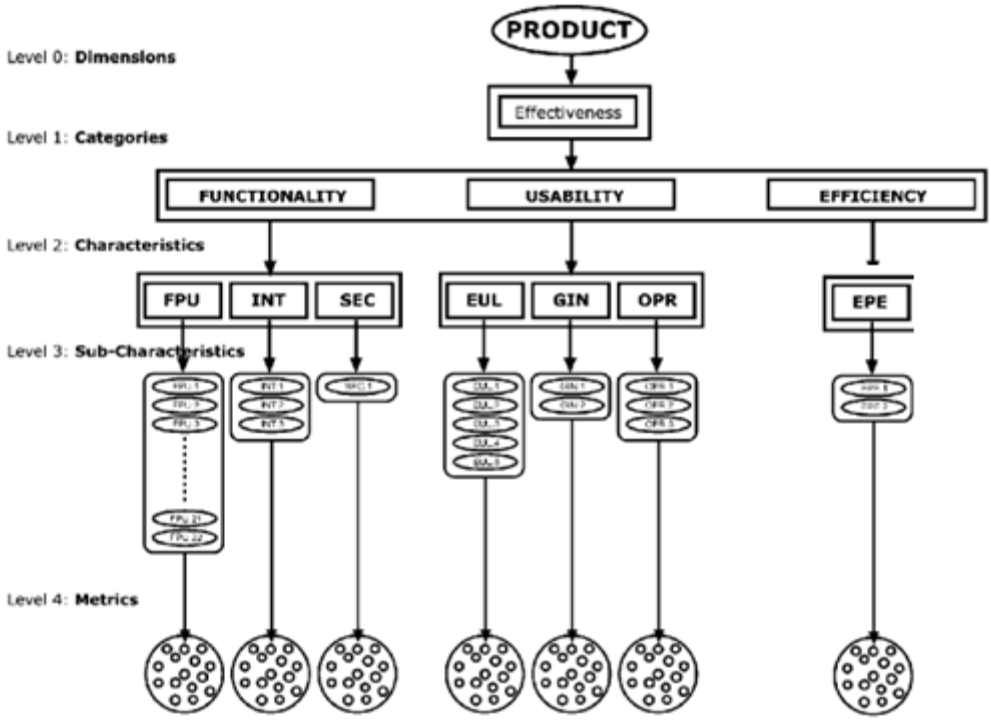


Fig. 2 Diagram of the adapted Systemic Quality Model (Rincon, Alvarez, Perez, & Hernandez, 2005)

Besides the *Functionality* category, we chose the *Usability*, because this type of tools are focused on non-technical users and the difficulty of the product must be minimum. Moreover, it must be an attractive product because the success of the tool depends on the user's satisfaction. Finally, the *Efficiency* category was chosen because the processor type, the hard disk space and the minimum RAM required, are factors that play an important role in making the deployment

of the tool a successful one. Self-Service BI tools are popular thanks to its "working in memory". Then it is important to evaluate the minimum amount of memory required.

3.3.1 Scales of measurement

In the current evaluation, all the evaluated metrics are ordinal variables because they have more than two categories and they can be ordered or ranked. Recalling, that metrics are explained, in detail, in the next **sub-chapter 4.9**. There are different types of scale measurement depending on the metric.

- **Type A of scale measurement:**

The main part of metrics are measured by the following scale. With this scale, metrics are measured from 0 to 4, as follows:

0: The application does not have the feature.

1: The application matches the feature poorly or it does not matches strictly the feature but it can get similar results.

2: The application has the feature and matches the expectations, although it needs an extra corporative complement. This mark should be also assigned, when the feature implies a manual job (e.g. typing code, click a bottom) and the metric is requiring an automatic job.

3: The application has the feature and matches the expectations successfully without a complement.

4: The application has the feature and moreover, present advantages behind others.

Even so, other metrics need to be measured in a special way.

- **Sub-type A.1** of scale measurement is assigned to binary metrics:

We assign 0 value if the application does not have the feature, and 4 value if the application has it. We chose these values in order to be consistent with the rest of the measurement scales.

- **Sub-type A.2** of scale measurement is assigned when the metric is measurable:

We assign 4 to the application with a better result and lower score to the others. As there are 4 applications, the scale is from 4 to 1. Although, if some applications had the same value for the metric, the same score has to be assigned to them. To clarify the current scale measurement, we present an example of the metric *Compilation speed* **(**which will be presented in **sub-chapter 3.4**). The *Compilation Speed* is measured with a scale from 1 to 4. We assign 1 value to the tools which requires more time to compile, and 4 to the tool which requires the shorter time.

Additionally, the official SQMO method involves a balance between all the characteristics because they have the same level of importance. But, sometimes, the user wants to give more importance to certain characteristics depending on his owns interests, and for that, we provided the following alternative, also very used as a variant of SQMO.

This alternative consists on assigning weights to the metrics. Therefore the importance level of the metrics varies. In the current project, the weights were assigned by Carlos Barahona, an expert user from INDRA. He remarked, that weights must depend on the requirements of each project. However, he tried to assign weights generalizing and based on his own experience managing projects. Recalling, that if the methodology is implemented for another use case, they can be modified. The used weights scale are the following:

0: Not applicable to Organization.

1: Possible usage feature or wish list item.

2: Desired feature.

3: Required or must have feature.

Finally, final scores for sub-characteristics, are computed using the weights assigned to the metrics. The final score of a sub-characteristic corresponds to the following formula:

$$score_{sub-characteristic\ i} = \frac{\sum_1^n v_j * w_j}{\sum_j^n w_j}$$

Where, $\boldsymbol{v_j}$ *is the value for the score assigned to metric* $\boldsymbol{j}$*, while* $\boldsymbol{w_j}$ *is the weight for the corresponding metric. And* $\boldsymbol{n}$ *corresponds to the number of metrics in the sub-characteristic* $\boldsymbol{i}$ *.*

This adaption of the methodology, is applied when the importance level of the metrics is not the same in all metrics. By this way, we got a score for each sub-characteristic, considering the weights of metrics. **Tab. 6** shows the weights assigned to each metric, in that case of use. Recall, that metrics are defined in the next **sub-chapter 3.4**.

METRIC	WEIGHT
Direct connection to data sources	2
BigData sources	1
Apache Hadoop	1
Microsoft Access	2
Excel files	3
From an excel file, import all sheets at the same time	2
Cross-tabs	2
Plain text	3

Connecting to different data sources at the same time	2
Easy integration of many data sources	2
Visualizing data before the loading	2
Determining data format	2
Determining data type	2
Allowing column filtering before the loading	2
Allowing row filtering before the loading	2
Automatic measures creation	3
Allow renaming datasets	2
Allow renaming fields	3
Data cleansing	2
Data model is done automatically	2
The done data model is the correct one	2
Data model can be visualized	3
Alerting about circular references	3
Skiping with circular references	3
A same table can be used several times	2
Creating new measures based on previous measures	3
Creating new measures based on dimensions	3
Variety of functions	3
Descriptive statistics	2
Preduction functions	2
R connection	2
Geographic information	2
Time hierarchy	3
Creating sets of data	2
Filtering data by expression	3
Filtering data by dimension	3
Visual Perspective Linking	2
No Null data specifications	2
Considering nulls	3

Variety of graphs	3
Modify graphs	3
No limitations to display large amounts of data	2
Data refresh	2
Dashboards Exportation	3
Templates	2
Free design	2
Reports Exportation	3
Templates	2
Free design	2
Languages displayed	2
Operating Systems	2
SaaS/Web	1
Mobile	2
Using the project by third parts	2
Exportation in txt	2
Exportation in CSV	2
Exportation in HTML	2
Exportation in Excel file	3
Password protection	3
Permissions	3
Average learning time	3
Consistency between icons in the toolbars and their actions	3
Displaying right click menus	3
Ease of understanding the terminology	3
User guide quality	2
User guide adquisition	2
On-line documentation	2
Availability of tailor-made training courses	2
Phone technical support	2
On-line support	2

Availability of consulting services	2
Free formation	2
Community	2
Editing elements by double-clicking	2
Dragging and dropping elements	2
Editing the screen layout	2
Automatic update	2
Compilation Speed	2
CPU(processor type)	2
Minimum RAM	2
Hard disk space required	2
Additional software requirements	2

Tab. 6 Weights of metrics

3.3.2 The concept of satisfaction

As it is said in **sub-chapters 3.2.1** and **3.2.2**, the term satisfaction can vary depending on the case of use. In fact, the evaluator can assign a limit, for example, 50%, and sentence that a feature is satisfied if its score is higher than the 50% ,of its maximum score in the measuring scale. For example, as our metric measuring scale is from 0 to 4, a metric is satisfied if its score is higher than 2. But, the evaluator can also sentence the limit to 3, and by this way, a metric is satisfied if its score is higher than 3.

Usually, assessments are done to determine which tools are better than others, supposing that all the evaluated tools satisfy the main part of the features. When the evaluator is looking for a distinction between tools, these type of limits can be useful.

This concept is applicable to our units of measurement, which are metrics, sub-characteristics, characteristics and categories.

Once, metrics are evaluated with their respective scales of measurement (**A**, **A.1**, **A.2**), the methodology used to determine the satisfaction score is as follows:

- Metrics scores are normalized with a percentage.
- A metric is satisfied if its percentage score is higher or equal than the fixed limit.
- Sub-characteristics are measured by the amount of metrics satisfied (**satisfaction score**). Then, a particular sub-characteristic is satisfied if the amount of satisfied metrics is higher or equal than its fixed limit. As weights were added, the satisfaction score become

$$score_{sub-characteristic\ i} = \frac{\sum_1^n v_j * w_j}{\sum_j^n w_j}$$

Where, $v_j = \begin{cases} 1, if\ the\ metric\ \boldsymbol{j}\ is\ satisfied, \\ 0, if\ the\ metric\ \boldsymbol{j}\ is\ not\ satisfied \end{cases}$

While $\boldsymbol{w_j}$ *is the weight for the corresponding metric* ***j****. And* ***n*** *corresponds to the number of metrics in the sub-characteristic* ***i*** .

- Characteristics are measured by the amount of satisfied sub-characteristics (**satisfaction score**). Then, a particular characteristic is satisfied if the amount of satisfied sub-characteristics is higher or equal than its fixed limit.
- Categories are measured by the amount of satisfied characteristics (**satisfaction score**). Then, a particular category is satisfied if the amount of satisfied characteristics is higher or equal than its fixed limit.

  In the current evaluation, we decided to use the following limits, in order to get distinctions between tools:

Limit for metric	50%
Limit for sub-characteristic	50%
Limit for Characteristic	75%
Limit for Category	75%

Tab. 7 Satisfaction limits

The evaluator can decide to modify the levels, in the case that any distinction exist between tools or to be more restrictive or unrestrictive.

3.4 Sub-characteristics and metrics for Self-Service BI tools evaluation

In a evaluation, the most important step is to decide which characteristics must be evaluating. According to the SQMO schema showed in **Fig. 3**, these characteristics were already agreed, but we had to establish the metrics related to each characteristic. By the experience in BI department and after working with these type of tools, we felt able to decide which particular topics must be checked from Self-Service BI software. For each of the three evaluable characteristics, there are listed the sub-characteristics and their respective metrics.

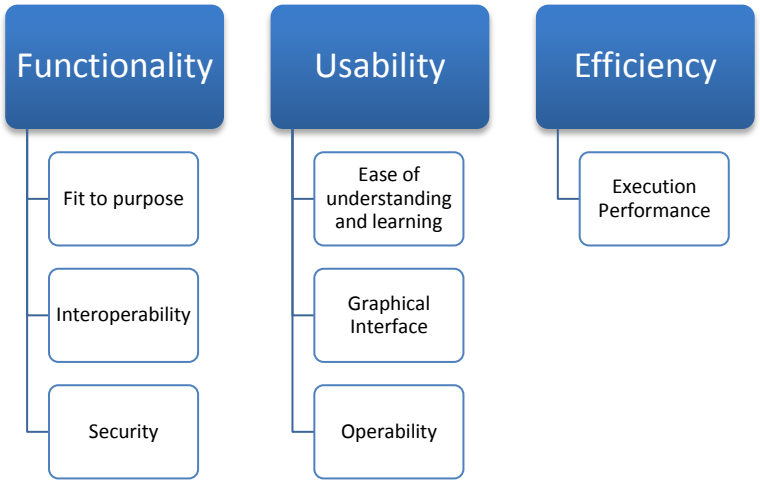


Fig. 3 Characteristic schema for each category, according to (Mendoza, Pérez, & Grimán, 2005)

3.4.1 Functionality category[1]

1 ***Fit to purpose*** **characteristic:** This characteristic includes different metrics classified in 5 sub-characteristics: Data loading, Data model, Field relations, Analysis, Dashboards and Reporting.

i ***Data loading*****:** This sub-characteristic includes various metrics to evaluate the loading process.

1 ***Direct connection to data source*** *(FFI1)****:*** It measures the possibility of a direct connection to data sources. There are some applications which integrates connector drivers (e.g. ODBC, JDBC...) compatible with some databases, and user does not need to install it in order to connect the application to the data source.

2 ***Big Data sources*** *(FFI2)****:*** It measures the capability to connect to any Big Data source different from Hadoop.

3 ***Apache Hadoop****(FFI3)****:*** It refers to the ability to connect to Hadoop infrastructure. This technology is used to manage large volumes of structured or non-structured data allowing fast access to data. Hadoop simply becomes one more data source and it is the most common way of storing big data, nowadays.

4 ***Microsoft Access*** *(FFI4)****:*** It evaluates the capability to connect to Microsoft Access database.

5 ***Excel files*** *(FFI5)****:*** It evaluates the capability to load data from Excel files.

6 ***From an Excel file, load data from all sheets at the same time*** *(FFI6)****:*** It evaluates the capability to load data from all sheets at the same time. In some applications, user must do the same data loading process for each one of the sheets, while other tools lets user to choose which sheets he wishes to load, and import them at the same time.

7 **Cross-tabs** *(FFI7)***:** It measures the capability of loading data from cross-tabs in Excel files. Usually, applications need cross-tabs in a specific format and some of them have an excel complement to normalize the cross-tabs before importing it.

8 ***Plain text*** *(FFI8)****:*** It evaluates the capability of loading data from plain text files (.txt,.inf, .80, .dat, .tmp, .prv, .hlp, .htm., etc.).

9 ***Connecting to different data source at the same time*** *(FFI9)****:*** It evaluates the capability to connect the application to several data sources at the same time, in order to do cross analysis between data from them.

10 ***Easy integration of many data sources*** *(FFI10)****:*** It evaluates how easy is for the user integrate many data source in the data analysis.

11 ***Showing data before the data loading*** *(FFI11)****:*** It evaluates the capability to show data before the data loading. Showing data can be useful for the user to understand how data are before load them.

12 ***Determining data format*** *(FFI12)****:*** It evaluates the capability to show data formats *(integer, double, date, string...)* of the fields before the data loading. Some applications assign formats to fields automatically while some others lets the user to assign them before the loading. Determining data formats before the loading is the best choice but in some applications it can be done after the loading, and it is equal evaluated.

13 ***Determining data type*** *(FFI13)****:*** It evaluates the capability to show data types *(dimension, measure)* of the fields before the data loading. Some applications assign

[1] Each metric has a corresponding code, which is used to identify them. In questionnaires from **Annex 2**, they are also identified by this code, like in the detailed evaluations from **Annex 3**.

types to fields automatically, while some others lets the user to assign them before the loading. Depending on the applications terminology, data types can be attribute or dimension and measure. Determining data types before the loading is the best choice but in some applications it can be done after the loading, and it is equal evaluated.

14 ***Allowing column filtering before the loading*** *(FFI14)**:*** It evaluates the capability to load only the columns that user wants.

15 ***Allowing row filtering before the loading****(FFI5)**:*** It evaluates the capability to filter registers before loading them. Sometimes, user does not want to analyse the whole dataset, and data filtering can be useful before load them.

16 ***Automatic measures creation*** *(FFI16)**:*** The ability of the tool to automatically create some measures, possibly useful, from the already loaded data.

17 ***Allow renaming datasets*** *(FFI17)**:*** It evaluates the capability to assign a name to datasets which should be loaded in the application.

18 ***Allow renaming fields*** *(FFI18)**:*** It evaluates the capability to rename fields. It can be useful when user has not named the fields in the database by himself, and prefers to rename them with more appropriate names to the analysis. Renaming fields before the loading is the best choice but in some applications it can be done after the loading, and it is equal evaluated.

19 ***Data cleansing*** *(FFI19)**:*** It evaluates the capability of the applications to allow user to clean data. For example, drop registers with null values or substitute particular values.

ii **Data model:** This sub-characteristic includes various sub-metrics to evaluate the modelling process for each tool. In order to clarify some of the proposed metrics, the database 20141220_Initial_test is used with examples.

1 ***Data model is done automatically*** *(FFD1)**:*** It refers to the capability of the applications to relate automatically tables. Some applications relate two tables if they have fields with same name and structure, therefore, these applications model data automatically.

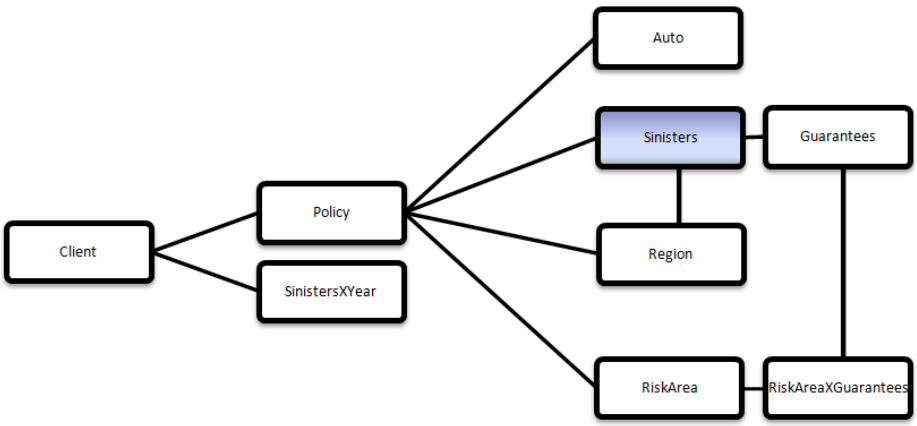


Fig. 4 The correct data model for 20141220_Initial_test data

2 ***The done data model is the correct one*** *(FFD2)**:*** This metric evaluates the capability of applications to get relations between tables by as the user wants it. In

our particular case of 20141220_Initial_test data, the model is showed in **Fig. 4**. If user builds the data model manually, getting the desired model should be easy. While, if the model is done automatically it can be more difficult depending on if the automatic model is the right one, or if there exist the possibility to modify the model by the user.

3 ***Data model can be visualized*** *(FFD3)**:*** This metric evaluates if a tool lets seeing the data model during the analysis. Visualizing the model during the analysis lets the user to check at all time the relations between fields.

iii **Field relations:** This sub-characteristic includes several metrics related to the connections between fields when the data source is relational. In order to clarify some of the proposed metrics, the database 20141220_Initial_test is used with examples.

1 ***Alerting about circular references*** *(FFF1)**:*** A circular reference exists when there are, at least, 3 tables related between them. In the particular case of the 20141220_Initial_test data, the schema in **Fig. 5** synthesizes the concept. For example, user can desire to visualize **Tab. 8**, it means particular policies and the regions where the policies have had an accident. *Policy table* is related to *Region table* by the field *Code*, which refers to the code identification for the region where the policy is registered. *Region table* have other fields, additionally to the *Code*, as the name of the region. On the other hand, *Sinisters table* is also related to *Region table* by the field *Code*, which refers to the code identification for the region where accidents occur.

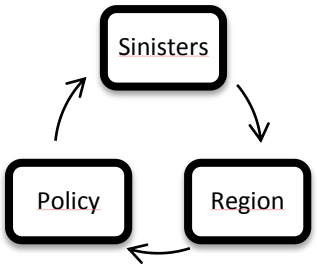


Fig. 5 Circular Reference

Policy_id	Code of the Region	Region

Tab. 8 Circular reference

In that particular case, some applications could show non-correct values for *Region* because they could doubt about which way take in order to reach the *Region table*. If it passes by *Policy table* then it shows regions where the policy is registered, but if it passes by the *Sinister table*, it shows regions where accidents ocure. This metric evaluates the capability of a tool to realize about a circular reference and alert the user about it.

2 ***Skipping circular references*** *(FFF2)**:*** This sub-characteristic evaluates the capability of the software to omit circular references.

3 ***A same table can be used several times*** *(FFF3)**:*** It evaluates the capability of the application to use a table directly related to more than one table. For example, if there is a table with coordinates, it can be related with more than one table, for example, with two tables where in the first one table there is a place of birth and in the second one there is a place of death. Some tools allows to load just once the table and use it as many times as the user wants. Others tools, require to load the table as many time as relations it will have.

iv **Analysis:** This sub-characteristic includes several metrics about the capabilities of the analysis.

1 ***Creating new measures based on previous measures*** *(FFA1)**:*** All the applications analyzed in that project must be able to create a measure based on already loaded measures. This sub-characteristic evaluates how easy is to build new measures based on loaded measures.

2 ***The creation of new measure based on dimensions*** *(FFA2)**:*** This sub-characteristic evaluates how easy is to build new measures based on loaded dimensions.

3 ***Variety of functions*** *(FFA3)**:*** It measures the diversity of functions offered by the application to build a new field. Applications can offer functions related with statistics, economics, mathematics and also with strings and logic functions.

4 ***Descriptive statistics*** *(FFA4)**:*** It refers to the possibility to analyze data statistically from a descriptive point of view. All the applications analyzed in that project are able to do descriptive statistics. Therefore this metric evaluates the complexity of the descriptive statistic allowed in each program.

5 ***Predictive Statistics*** *(FFA5)**:*** It measures the ability of getting indicators by predictive functions. It is not a common feature in Self-Service BI tools and because of that, the presence of few predictive methods will be positively evaluated.

6 ***R Connection*** *(FFA6)**:*** It evaluates the capability of applications to connect to R in order to get advanced analytical functions.

7 ***Geographic Information*** *(FFA7)**:*** This sub-characteristic measures the capability of display data in maps.

8 ***Time hierarchy*** *(FFA8)**:*** It evaluates the capability of the application to create: *time intelligence*. It consist in, from a particular date, create other fields like month, quarter or year. This set of fields are grouped in a hierarchy. Particularly, a time hierarchy. This metric evaluates the capability of the tool to create directly time hierarchies.

9 ***Creating sets of data*** *(FFA9)**:*** It evaluates the capability of a tool to create sets of data. During the analysis, the user can be interested in analyse in more depth a set of registers. Some tools lets to save these datasets and work with them.

10 ***Filtering data by an expression*** *(FFA10)**:*** It evaluates the capability of a tool to filter data during the analysis by expression values.

11 ***Filtering data by a dimension*** *(FFA11)**:*** It evaluates the capability of a tool to filter data during the analysis by dimension values.

12 ***Visual Perspective Linking****(FFA12)**:*** It evaluates the capability to link multiple images, so a selection on one image shows related and relevant data in others images.

13 ***No null data specifications*** *(FFA13)**:*** This metric evaluates if the applications have any requisite to the null values, for example that null values must be noted as NULL, or just with an space… or by contrary, that the user can define how are the null values represented in the data-source.

14 ***Considering nulls*** *(FFA14)**:*** This metric measures if applications consider null values as another value. Considering null as other value might be useful, because the user can visualize the behaviour of null data and then detect a pattern for them. This metric also evaluates if null values are skipped from a calculated expression.

15 ***Variety of graphs*** *(FFA15)**:*** It measures the diversity of graphs offered by the application.

16 ***Modifying graphs*** *(FFA16)**:*** It measures the capability to modify the default setting of graphs. For example, if there is the possibility to change levels of a legend, change colours, change the shapes of markers... It is an important characteristic because sometimes it is the key to understand a data pattern.

17 ***Huge amount of data*** *(FFA17)**:*** It measures the capability to display huge amount of data. Particularly, it measures the capability to display datasets without any data problem because of its size.

18 ***Data refresh*** *(FFA18)**:*** It measures the capability to update data automatically. For example, if data are modified in the original file, some applications update automatically the data while in others the user must do it, manually.

v **Dashboard:** This sub-characteristic includes several metrics to measure the capabilities of a tool relating to dashboards.

1 ***Dashboards exportation*** *(FFD1)**:*** It evaluates the capability of the tool to export the dashboard in order to share with other people to visualize and interact with the results.

2 ***Templates*** *(FFD2)**:*** It evaluates the capability to fix a schema dashboard or access to templates in order to use it several times with different types of data. It is a useful feature to homogenize projects.

3 ***Free design*** *(FFD3)**:*** It measures the ability to let the user to build dashboards with total freedom. Some tools have limited options to building dashboards, while others lets the user to insert text, format it, inserting images…

vi **Reporting:** This sub-characteristic includes several metrics to measure reporting capabilities of a tool.

1 ***Reports Exportation*** *(FFR1)**:*** It evaluates the diversity of formats to export reports. Some formats are excel spread sheet, PDF files, HTML files, Flash file, the own tool format...

2 ***Templates*** *(FFR2)**:*** It evaluates the capability to fix a schema report or access to templates in order to use it several times with different data. It is a useful feature to gain consistency when the user builds the same type of report periodically.

3 ***Free design*** *(FFR3)**:*** It measures the ability to let the user to build reports with total freedom. Some tools have limited options to building dashboards, while others lets the user to insert text, format it, insert images…

2 **Interoperability characteristic**: This characteristic includes several sub-characteristics in order to evaluate the capability of an application to work with other organizations and systems.

i **Languages:** This sub-characteristic is composed by a metric, which evaluates the variety of languages displayable in the tool.

1 ***Languages displayed*** *(FIL1)**:*** It evaluates the variety of displayed languages offered by the tool. In particular, it evaluates if the tool can be displayed in more than two languages or not.

2 **Portability:** This sub-characteristic is composed by three metrics, which evaluates the ability of a tool to be executed in different environments.

3 ***Operating systems*** *(FIP1)**:*** This metric measures the variety of different operating systems compatible with the tool. In particular, it evaluates if the tool can work, at least, in two different operating systems.

4 ***SaaS/Web*** *(FIP2)**:*** The acronym SaaS means Software as a Service. This metric evaluates if a tool offers access to projects via web browser for hosting their own deployments in the cloud.

5 ***Mobile*** *(FIP3)**:*** It evaluates the possibility to have reports and dashboards available in the mobile device via a mobile app.

ii ***Use project by third parts:*** This sub-characteristic is composed by a unique metric and it measures the capability of sharing and modifying projects by other people.

1 ***Using the project by third parts*** *(FIU1)**:*** It evaluates the capability to share projects and modify them by other users.

iii **Data exchange:** This sub-characteristic is composed by metrics, which evaluate the data exportation when they have already been manipulated in the tool.

1 ***Exportation in .txt*** *(FID1)**:*** It evaluates the capability of a tool to export data .txt.

2 ***Exportation in CSV*** *(FID2)**:*** It evaluates the capability of a tool to export data in CSV format.

3 ***Exportation in HTML*** *(FID3)**:*** It evaluates the capability of a tool to export data in HTML format.

4 ***Exportation in Excel file*** *(FID4)**:*** It evaluates the capability of a tool to export data in Excel files.

3 **Security characteristic:** This characteristic is composed by a unique sub-characteristic, which groups metrics about the security process.

i **Security devices:** This sub-characteristic is composed by two metrics related with the protection of data.

1 ***Password protection*** *(FSS1)**:*** It evaluates the capability to protect projects with password.

2 ***Permissions*** *(FSS2)**:*** It evaluates the capability to assign different permissions to different users.

3.4.2 Usability category

1 **Ease of understanding and learning characteristic:** This characteristic includes different sub-characteristics.

i **Learning time**: This sub-characteristic includes only one metric.

1 ***Average learning time*** *(UEL1)**:*** This metric measures the time spent by the user in learning the functionality of the tool.

ii **Browsing facilities**: This sub-characteristic evaluates how the user can browse inside the tool.

1 ***Consistency between icons in the toolbars and their actions*** *(UEB1)**:*** This metric measures the capability of the tool to be consistent with its icons.

2 ***Displaying right click menus*** *(UEB2)**:*** This metric measures if the tool offers a displaying menu by right clicking.

iii **Terminology**: This sub-characteristic evaluates if the terminology is consistent with the global business intelligence terminology.

1 ***Ease of understanding the terminology*** *(UET1)**:*** This metric measures how easy is for the user to understand the terminology.

iv **Help and documentation:** This sub-characteristic is composed by metrics, which measures the help offered by the tool to a user when he has doubts about the functionality or management of the tool.

1 ***User guide quality*** *(UEH1)**:*** This metric evaluates if the user guide is understandable. Highlighting that Self-Service tools are also offered for non-technical users.

2 ***User guide acquisition*** *(UEH2)**:*** This metrics measures the process to get to the user manual. For example, if it is free, if it is difficult to find in the web….

3 ***On-line help*** *(UEH3)**:*** It measures the offering of on-line help.

v **Support and training:** This sub-characteristic measures the quality and variety of the support offered by the tool.

1 ***Availability of tailor-made training courses*** *(UES1)**:*** It measures if the tool offers training courses adapted to organizations, and it is positively measured if the course can be done in the organization.

2 ***Phone technical support*** *(UES2)**:*** It measures if the tool offers a phone for technical support and the timetable of it.

3 ***On-line support*** *(UES3)**:*** It measures if the tool offers on-line support, and if it is in life or not.

4 ***Availability of consulting services*** *(UES4)*: It measures if the company offers consulting services.

5 ***Free formation*** *(UES5)*: It evaluates if the platform offers free formation for users.

6 ***Community*** *(UES6)**:*** It evaluates if there exist a community to ask for doubts or to share knowledge with other users.

2 **Graphical interface characteristic:** This characteristic evaluates the graphical interface of the tool.

i **Windows and mouse interface:** This sub-characteristic evaluates the windows interface and the mouse functions.

1 ***Editing elements by double-clicking*** *(UGW1)**:*** It measures if the tool offers editing elements by double-clicking.

2 ***Dragging and dropping elements*** *(UGW2)**:*** It measures the capability of the tool in dragging and dropping elements.

ii **Display:** This sub-characteristic refers to a unique metric about the capability of editing the screen layout.

1 ***Editing the screen layout*** *(UGD1)**:*** It measures the capability of a tool to edit the screen layout.

3 **Operability characteristic:** This characteristic evaluates the ability of the tool to keep the system and the tool in reliable functioning conditions.

i **Versatility:** This sub-characteristic evaluates the versatility of the tool.

1 ***Automatic update*** *(UOV1)**:*** It measures if the tool is automatically updated when new versions appears.

3.4.3 Efficiency category:

1. **Execution performance characteristic**: This characteristic is composed by sub-characteristics, which evaluates the execution performance of the tool.

ii **Compilation speed:** This sub-characteristic measures the compilation speed, how fast the software build a particular chart.

1 ***Compilation speed*** *(EEO1)**:*** It measures the compilation speed. It is a very subjective measure because it depends on the machine where it is installed.

iii **Resource utilization:** This sub-characteristic evaluates the extra hardware and software requirements.

1 ***CPU (processor type)*** *(EER1)**:*** This metric evaluates if the tool can be installed as much to x86 processors suc has to x64 processors.

2 ***Minimum RAM*** *(EER2)**:*** It measures the RAM needed in the way that a maximum punctuation means it requires low memory while the minimum punctuation means it needs many memory.

3 ***Hard disk space required*** *(EER3)**:*** It measures hard disk space needed in the way that a maximum punctuation means it requires low space while the minimum punctuation means it needs many memory.

iv ***Software requirements:*** This sub-characteristic is composed by a unique metric, which measures if adding software is required to execute the tool.

1 ***Additional software requirements*** *(EES1):* this metric evaluates if adding software is required to execute the tool.

4 Software selection for the evaluation

Prior to an evaluation there must be a selection of software, hence some aspects should be considered.

Firstly, the area of application and use of the software should be pre-established. The selection of software depends on this aspect because not every software is appropriate for every area. If the area of application is pre-established, the selected software will be according with it.

Secondly, a new level of depth should be considered with more specifications about the tool functionality. It should consider the features that make the tool useful for what we want to do.

And finally, there is the identification of the required attributes based on the particular aims of the organization who will use the tool. Some of these attributes must be mandatory and others must be non-mandatory. Mandatory attributes are those that must be met by the selected software, while non-mandatory are those that will be evaluated, that are the metrics. Therefore, this aspect takes an important role in the selection and also in the evaluation.

4.1 Algorithm

In order to select the software, the first step was to decide which tools could be evaluated with this model. Nowadays, there are many applications in the market related with Business Intelligence. And because of that, deciding which applications should be included in an evaluation is a laborious task. In this stage we were inspired by the methodology for selecting software proposed by Le Blanc. In the first place, a long list of BI tools was elaborated. Next step was to reduce this to a medium list containing popular tools which accomplish critical capabilities for business intelligence and analytics. And finally, a short list provided with particular aims of the organization, was built.

The particular area of application is Business Intelligence and there are many platforms specialized in this area in the market. Therefore we focus on which have been mentioned in the report Magic Quadrant for Business Intelligence and Analytics Platforms (February 2015) from Gartner. Gartner is an information technology research and advisory company, which presents every year different market research reports on IT products. Magic Quadrant is an annual report that reflects the innovations and changes that are driving the BI market and shows the relative position of each competitor in the business analytics space. They consider all tools in the market, and if these tools met the inclusion criteria they are included in the evaluation. In this first step, we used Gartner as a data source of all Business Intelligence and Analytics platforms in the market. Each year it edits an updates reports and also their inclusion criteria changes depending on how the market changes, so it is a reference company to have knowledge of BI tools. By this way, all the tools mentioned in the Magic Quadrant report of February 2015 (although Gartner, finally, have not evaluated them) composed our long list of 63 different platforms, which is the following:

1) Adaptive Insights
2) Advizor Solutions
3) AFS Technologies
4) Alteryx
5) Antivia
6) Arcplan
7) Automated Insgihts
8) BeyondCore
9) Birst
10) Bitam
11) Board International
12) Centrifuge Systems
13) Chartio
14) ClearStory Data
15) DataHero
16) Datameer
17) DataRPM
18) Datawatch
19) Decisyon
20) Dimensional Insight
21) Domo
22) Dundas Data Visualization
23) Eligotech
24) eQ Technologic
25) FICO
26) GoodData
27) IBM Cognos
28) iDashboards
29) Incorta
30) InetSoft
31) Infor
32) Information Builder
33) Jedox
34) Kofax(Altosoft)
35) L-3
36) LavaStorm Analytics
37) Logi Analytics
38) Microsoft BI
39) MicroStrategy.
40) Open Text (Actuate)
41) Oracle
42) Palantir Technologies
43) Panorama
44) Pentaho
45) Platfora
46) Prognoz
47) Pyramid Analytics
48) Qlik
49) Salesforce
50) Salient Management Company
51) SAP
52) SAS (SAS Business Analytics)
53) Sisense
54) Splunk
55) Strategy Comapnio
56) SynerScope
57) Tableau
58) Targit
59) ThoughtSpot
60) Tibco Software
61) Yellowfin
62) Zoomdata
63) Zucche

To build the medium list we followed also the steps of Gartner, in the Magic Quadrant report, where they choose the platforms to be evaluated if they satisfied 13 technique features and 3 non-technique.

The 13 technique features were, by Gartner, the critical capabilities that every Business Intelligence and Analytics platform must satisfy. And they were classified in three categories: *Enable*, *Produce* and *Consume*.

Enable:

- Functionality and Modelling: Combination of different sources and the creation of analytic models such as user-defined measures, sets, groups and hierarchies. Advanced capabilities include semantic auto discovery, intelligent joins, intelligent profiling, hierarchy generation, data lineage and data blending on varied data sources, including multi structured data.
- Internal Platform Integration: A common look and feel, install, query engine, shared metadata, promo ability across all platform components.
- BI Platform Administration: Capabilities that enable securing and administering users, scaling the platform, optimizing performance and ensuring high availability and disaster recovery.
- Metadata Management: Tools for enabling users to leverage the same systems-of-record semantic model and metadata. They should provide a robust and centralized way for administrators to search, capture, store, reuse and publish metadata objects, such as dimensions, hierarchies, measures, performance metrics/KPIs, and report layout objects.
- Cloud Deployment: Platform as a service and analytic application as a service capabilities for building, deploying and managing analytics in the cloud.
- Development and Integration: The platform should provide a set of programmatic and visual tools and a development workbench for building reports, dashboards, queries and analysis.

Produce:

- Free-Form Interactive Exploration: Enables the exploration of data via the manipulation of chart images, with the colour, brightness, size, shape and motion of visual objects representing aspects of the dataset being analysed.
- Analytic Dashboards and Content: The ability to create highly interactive dashboards and content with visual exploration and embedded advanced and geospatial analytics to be consumed by others.
- IT-Developed Reporting and Dashboards: Provides the ability to create highly formatted, print-ready and interactive reports, with or without parameters. This includes the ability to publish multi object, linked reports and parameters with intuitive and interactive displays.
- Traditional Styles of Analysis: Ad hoc query enables users to ask their own questions of the data, without relying on IT to create a report. In particular, the tools must have a reusable semantic layer to enable users to navigate available data sources, predefined metrics, hierarchies and so on.

Consume:

- Mobile: Enables organizations to develop and deliver content to mobile devices in a publishing and/or interactive mode.
- Collaboration and Social Integration: Enables users to share and discuss information, analysis, analytic content and decisions via discussion threads, chat, annotations and storytelling.
- Embedded BI: Capabilities for creating and modifying analytic content, visualizations and applications, and embedding them into a business process and/or an application or portal.

Moreover, platforms had met other non-technical criteria:

- Generating at least $20 million in total BI-related software license revenue annually, or at least $17 million in total BI-related software license revenue annually, *plus* 15% year-over-year in new license growth.
- In the case of vendors that also supply transactional applications, show that its BI platform is used routinely by organizations that do not use its transactional applications.
- Had a minimum of 35 customer survey responses from companies that use the vendor's BI platform in productions.

With this added non-technical features, they guaranty that at least 35 companies use each one of the tools. Moreover, they guaranty that companies, which are growing year-over-year, use these tools. That's why INDRA is interested in these particular tools because of their popularity and, as a consultant, they want to be up-to-date on this area.

And the medium list obtained is:

1) Alteryx
2) Birst
3) Board International
4) Datawatch
5) GoodData
6) IBM Cognos
7) Information Builder
8) Logi Analytics
9) Microsoft BI
10) MicroStrategy. (MicroStrategy Visual Insight)
11) Open Text (Actuate)
12) Oracle
13) Panorama
14) Pentaho
15) Prognoz
16) Pyramid Analytics
17) Qlik (QlikView)
18) Salient Management Company
19) SAP (SAP Lumira)
20) SAS (SAS Business Analytics)
21) Tableau
22) Targit
23) Tibco Software
24) Yellowfin

Finally, to build the short list we focus on the particular aims of the organizational unit.
The particular tools that we wanted to evaluate are Self-Service BI tools and it means that the business user should be able to analyze the information he wants and build his owns reports. In traditional tools, user asked to a technical team for the information he needed and he ordered how information had be displayed and the technical team prepared data and built the ordered reports. Against that, Self-Service tools are being imposed on others because the working methodology is changing from being driven by the business model to being driven by the data model.

The main features of Self-Service tools, according to INDRA S.A are:

- *Ease of use:* These tools are designed to be used by non-technical people. It means that users do not need to spend much time in learning how the tool works before doing a basic analysis.
- Ability to incorporate data sources, both corporative data base (Oracle, SAP,etc) as local (Basically excels) and also external data base (Twitter,etc).
- *'Intelligence' to interpret correctly data models.* As they are auto-service tools and they face to many type of data model, without a previous modeling by a technical team, the interpretation of the model from the tool must be the correct one. If it is not the correct one, it can be misleading. How easy is to discover that the data model is wrong and how easy is to arrange the data model, are also important points to consider.
- *Analysis functions:* Besides the typical pie and bar graphs, they must incorporate other tools in order to get advanced analysis (integration in R, statistic routines…) always remembering the easy use.
- *Possible integration with corporative systems and efficiency:* Usually, the user will work with huge volume of data and therefore the analysis cannot be in a local PC. Tools should have the option of a central server which access to data and process them. Big companies, as INDRA S.A clients, needs security when the server is incorporated to the corporative environment. And then, the role of an administrator to manage the user's access is key for big companies.
- *Support:* In the case, that an open source tool was included in the larger list, it will not be considered in the medium list because open source cannot offer an instantly customer support. In open source there are communities of users who can help others in their problems, and for INDRA as a big company, and as a company who offers they workers as a service, the customer support is very important and must be fast.

Therefore, these five (5) features characterize the particular aims of the organization for the tools to be evaluated with the adapted SQMO. And then, from the medium list, the short list includes only tools, which, by our point of view, satisfy the mentioned features, and they are:

1) MicroStrategy Visual Insight from MicroStrategy platform
2) Panorama
3) Pyramid Analytics
4) QlikView tool from Qlik platform
5) SAP Lumira from SAP platform
6) SAS Business Analytics from SAS platform
7) Tableau

8) Tibco Jaspersfot from Tibco Software Platform

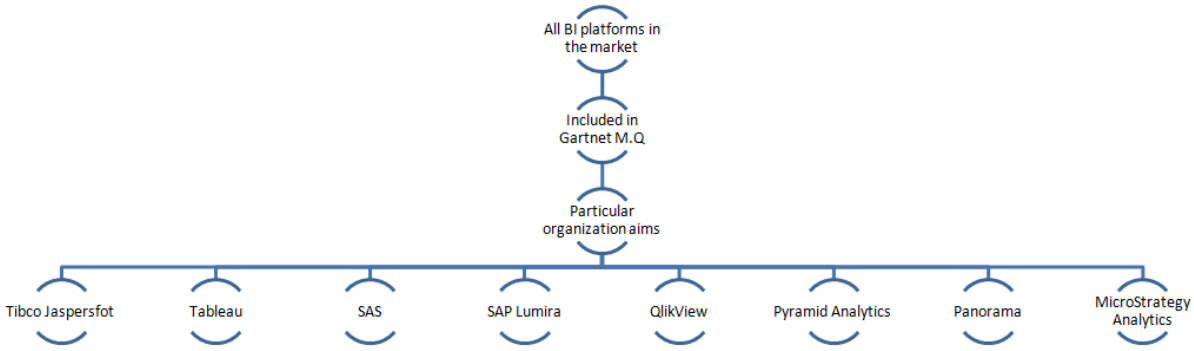


Fig. 6 Schema for the selection process

4.2 The 4 evaluated software

In this project we evaluate four (4) tools from the short list. The evaluated tools are those that, according to the vision of INDRA (INDRA has the major Business Intelligence unit in Spain), have more projection. As a cause of the amount of clients/projects implementing them or because clients show interest in these applications, the final four tools are: QlikView, Tableau, MicroStrategy Analytics and SAP Lumira.

QlikView was designed in 1993 to generate business insight by accessing information from standard database applications and displaying their data associatively. Moreover, it already runs entirely in memory, as a pioneer. And 7 years later, QlikView was focused on the BI market. Because it was the more mature tool in the market running in memory associative search engine, it was a very interesting tool for clients, and it was evaluated.

MicroStrategy, as a global BI platform, was the most implemented tool among INDRA‘s clients. And because of that, its BI tool, **MicroStrategy Analytics,** was evaluated.

Tableau was the tool, among all Self-Service BI tools, which was mentioned by more clients. This tool was created with the objective of giving more emphasis to the visual data analysis. Because of its popularity among clients, it was evaluated.

SAP Lumira was chosen because of the huge number of SAP implementations in management systems. SAP had been incorporated recently in the Data discovery with SAP Lumira but its success in management systems and its huge number of implementations maked it a natural competitor to consider. As many clients had implemented SAP in their management system, they, surely, would opt to implement SAP Lumira thinking in a better integration with their system and an easier architecture because of a unique provider.

5 Data

In order to use and evaluate the applications, we needed a set of data and we decided to simulate it. The data set was simulated using the tool R and it was constructed replying an car insurance company database and using a relational structure.

5.1 Relational data model

A relational database is based on the relational model developed by E.F. Codd. In such models data are organized into tables related one to each other by at least one common field.

The main important properties of relational data model are that:

- Data are presented as a collection of relations between tables.
- Each relation is defined by one or more column (field) in common between tables.
- Columns are attributes that belong to the entity modelled by the table (ex. In a client table, you could have name, gender, birthday, etc.).
- Each row (also called *tuple*) represents a single entity (ex. In a client table, John Smith, Male, 30/11/1975, would represent one client entity).
- Every table has a set of attributes that taken together as a *key*, uniquely identifies each entity (e.g.: in a client table, "ClientID" would uniquely identify each client – no two clients would have the same clientID).

Certain fields are designated as keys, which means that searches for specific values of that field will uniquely identify each entity. There are many types of keys, however, quite possibly the two most important are the primary key and the foreign key. The primary key is what uniquely identifies each entity within a table. The foreign key is a primary key of one table, that is also present into another table. Where fields in two different tables take values from the same set, a join operation can be performed to select related records in the two tables by matching values in those fields. Usually, but not always, the fields will have the same name in both tables. Ultimately, the use of foreign keys is the heart of the relational database model. This linkage that the foreign key provides, is what allows to link data together. In the relational data model, there are two important rules that help to ensure data integrity. They are:

- Every *tuple* is unique. This means that for every record in a table there is something that uniquely identifies it from any other *tuple*, the primary key.
- Table names in the database must be unique and attribute names in tables must be unique. No two different tables can have the same name in a data model. Attributes (columns) cannot have the same name in a table. You can have two different tables that have similar attribute names.

Among relational data model, there are different types of models depending on its structure. Particularly, we used a **snow flake** schema which is a type of relational data model composed by two types of tables: *Fact tables* and *Dimension tables.*

- A *Fact table* contains information of particular events. By this way, they typically have three types of columns: those that identifies the tuple (primary key), those that contains the facts, and those that are a foreign key to dimension tables.
- A *Dimension table* contains extensive information about the values that are used in the fact tables. By this way, it contains information about all the possible values of a particular field, used in the fact table.

For example, a *Fact table* contains information about occurred accidents. And it has fields *ClientID*, *Date*, *Region* and *Guarantee*. Then, a dimension table can be one with information about the client. With fields *ClientID*, *Gender* and *Birthday*. And another *Dimension table* can be one with information of the region, with fields *Region* and *Population*.

A **snow flake** schema contains a *Fact table* and multiple dimension tables, which can be connected to other *Dimension tables*. The last ones, are known as parents. A parent in a relational data model, represents a *Dimension tables* which is connected to another *Dimension tables*. It means, that a parent dimension table offers information about other *Dimension table*.

When the model is only composed by a *Fact table* and several *Dimension tables* without parents, it is called star schema. Which is a simpler snow flake schema.

Finally, there are two types of relationships between tables, in relational data models:

- A **1:n** relationships is typically modelled using a simple foreign key. Where, one column in table A references a similar column in table B, typically the primary key. Since the primary key uniquely identifies exactly one row, this row can be referenced by many rows in table B, but each row in table B can only reference one row in table A. The name of this type of relationships means '1 to many'.

- A **n:m** (or n:n) relationship occurs when each row in table A can reference many rows in table B, and each row in table B can reference many rows in table A. The name of this type of relationships means 'many-to-many';

Once, the reader has been introduced in the relational data modelling, it is time to present the database created in this project: **20141220_Initial_test**.

5.2 20141220_Initial_test

The created database, used in order to evaluate the applications, is called **20141220_Initial_test**. It is composed by nine (9) tables forming a relational database, and particularly a snow flake schema.

In **Fig. 7**, it is showed the relational data model structure, where two tables are related by a common field (*foreign key*), which appears in both tables and which is showed in the figure, next to the type of relationship. The *Fact table* is called *Sinisters*, and the *Dimension tables* are *Client*, *Policy*, *Auto*, *Region*, *SinistersXYear*, *RiskArea*, *Guarantees* and *GuaranteesXRiskArea*.

We decided to propose a relational database in order to realize how the evaluated Self-Service BI tools managed the relations between tables. Some of these tools built automatically the data model (the relations between tables), that is that the user loads tables and the tool, by itself, relates tables. Hence, we wanted to know if this automatic modelling worked well or not.

Moreover, we wanted to evaluate if applications were capable to understand both types of relationships. In fact, the most common relationship is 1:n, and we were almost certain that applications support them. But we doubted about the supporting of n:m relationships. In fact, there were one of the evaluated tools, *SAP Lumira*, which could not relate two tables by a n:m relationship.

Additionally, our model has a particularity. There are two circular references in *Region* and *Guarantees* fields. A circular reference exists when there are, at least, 3 tables related between them. For example, *Region table* has information about the regions and in fact it has the name of the all regions and its population. *Policy table* is connected to *Region table*, by the field *Code*. This field corresponds to the code of the region where the policy is registered. On the other hand, *Sinisters table* is also connected to *Region table*, by the field *Code*. But, this time, it corresponds to the code of the region where the accident had happened. Both relations have different meaning, but they are related to the same table. We added these circular references in order to know how the Self-Service applications managed them.

Skipping circular references can be done easily, by duplicating tables. In fact, we have loaded two tables identically equal to *Region table*, one is related to *Policy table* and the other one to *Sinisters table*. But, this action implies the use of more memory and it is not a recommended.

The fact that we decided to simulate a car's insurance company database is due to it is a common case of use in consultancy, like INDRA. Moreover, we were lucky to know an actuarial expert who offered us some information about the car's insurance area.

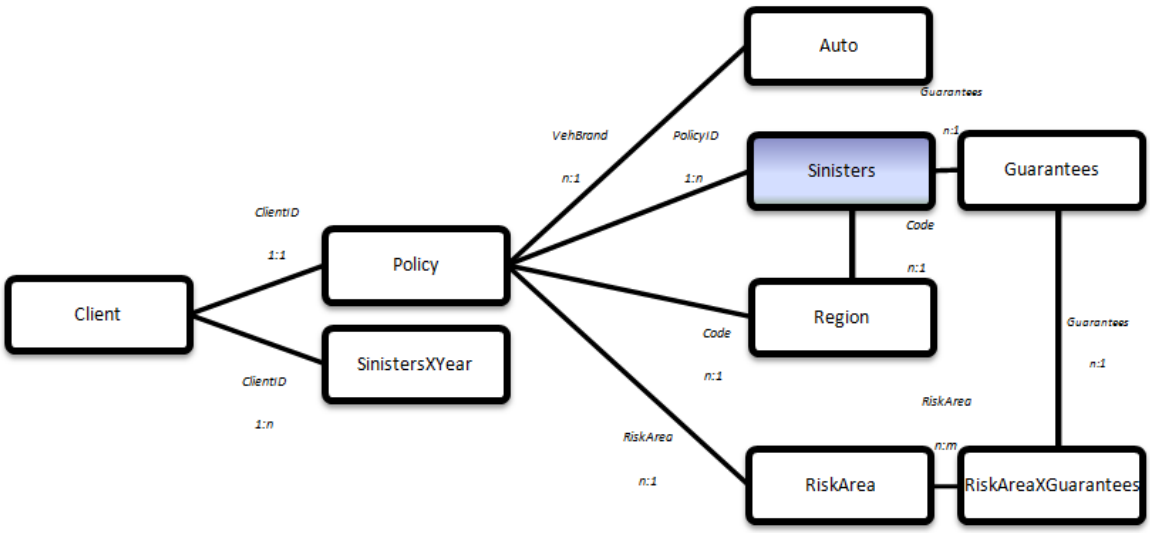


Fig. 7 Data model for 20141220_Initial_test

The main point of the thesis was not to do an accurate analysis of data. For this reason, the simulation was just a way of getting data and they cannot be considered as real data, because the process to get them is just a roughly approximation.

We obtained some data from two existing datasets of R. In order to not have to invent all data, although some fields were invented by ourselves because they were not in the existing datasets. Some data were extracted from the *CASdatasets* package of R. It is composed by several actuarial datasets (originally for the 'Computational Actuarial Science' book). Particularly, we extracted some data from *freMPL6* and *freMTPL2freq* datasets.

Moreover, in order to evaluate the analysis capabilities of the applications, data were simulated forcing patterns. In particular, geographical and stationary patterns were imposed. In **20141220_Initial_test**, the amount of occurred car accidents in a region is proportional to the amount of population in it. But, in the months of July, August and September, in the region of Granada we force to have more accidents. Additionally, some people is forced to have more probability to have accidents than the main part of the population. They are woman with ages between 40 and 45, man with ages between 50 and 65, young people under 24 and beginners.

The following chapter explains, in more detail, each one the tables composing the database **20141220_Initial_test**. And the scripts of the R code is available in **Annex 1**.

5.3 Tables

Data from 20141220_Initial_test come from a mix of two datasets (*freMPL6* and *freMTPL2freq*) provided in *CASdatasets* R package, and additional data were invented by us. We choose fields from *freMPL6* and *freMTPL2freq*, that were interesting for us, and not all fields were added in our database.

In one hand *freMPL6* comes from a private motor French insurer. It includes client's information of 26.000 policies in the year 2004. On the other hand, *freMTPL2freq* includes information about 413.169 policies (observed on an unknown year), like the vehicle power, the vehicle age, the value 'bonus-malus', the vehicle brand, the region where the policy is registered and its population density.

We wanted to have data for 10 years and more fields different from the ones offered from *freMPL6* and *freMTPL2freq.* For this reason, we invented some fields and also extrapolated data to 10 years.

In the following sub-chapters, detailed information about the creation of the tables is introduced. Reader can access to the R scripts, which are in **Annex 1**.

Although it could seem that tables are just tables, the creation and transformation of original data has become a laborious task. Hence, we included the transformations carried out, yet the reader might skip them.

5.3.1 Client table

Client table corresponded to a *Dimension table*, including clients information. It was composed by the following fields: *ClientID*, *Gender*, *MariStat*,*CSP*, *DrivBeg* and *LicBeg*. Fields are described in Tab. 9

Numeric fields			
Field name	**Description**	**Min**	**Max**
ClientID	Client identification	1	26.000
Date fields			
Field name	**Description**	**Min**	**Max**
LicBeg	Date of license	1941-09-05	2009-01-11
DrivBeg	Birthday	1916-01-25	1991-01-06
Categorical fields			
Field name	**Description**	**Values**	
Gender	Gender	'Male', 'Female'	
MariStat		'Others', 'Alone'	
CSP[2]	Social category	CSP2, CSP3, CSP7,... (33 different values)	

Tab. 9 Client table description

All data for the corresponding fields were extracted from the *freMPL6* dataset, with the exception of *ClientID*, which was generated.

We created a discrete vector called *ClientID* with values from 1 to N=26.000. They were the client's identifications. We chose N=26.000 because the dataset *freMPL6* had 26.000 registers of clients, becoming *ClientID* the primary key of the table because of its uniqueness.

The dataset also provides two fields related to the driving license age and the driver age. But driving license age was measured in months and driver age in years. Then, we did the necessary calculations to get the values as dates. The results are the date when the license was gotten and the birthday of driver, which are called *LicBeg and DrivBeg*, respectively.

Finally, the resulting *Client table*, which is showed in **Fig. 8**, has 26.000 registers and 6 columns:

ClientID	Gender	MariStat	CSP	DrivBeg	LicBeg
1	Male	Other	CSP50	12/01/1965	25/09/1983
2	Male	Other	CSP50	12/01/1965	26/08/1983
3	Male	Other	CSP50	09/01/1979	16/10/1996
4	Female	Other	CSP55	14/01/1959	07/02/1981
5	Male	Other	CSP60	15/01/1954	05/01/1976
6	Male	Other	CSP60	15/01/1954	07/10/1975
7	Male	Other	CSP48	15/01/1953	19/02/1973
8	Male	Other	CSP48	15/01/1953	20/01/1973
9	Female	Other	CSP50	14/01/1958	26/09/1977
10	Male	Other	CSP55	15/01/1954	19/04/1979
11	Male	Other	CSP55	15/01/1954	20/11/1978

Fig. 8 Client table

[2] [2] The classification of socio-professional categories (CSP) was conceived by INSEE in 1954. The objective was to categorize individuals according to their professional situation, taking account of several criteria: their profession, economic activity, qualification, hierarchical position and status.

5.3.2 Auto table

Auto table corresponds to a dimension table , including vehicle information and it is composed by the following fields: *VehBrand*, *VehPow* and *VehType.* Fields are described in **Tab. 10**.

Categoric fields		
Field name	**Description**	**Values**
VehBrand	Vehicle brand	‘1’,’2’,’3’,’4’,’5’,’6’,’10’,’11’,’12’,’13’,’14’
VehPow	Vehicle Power	‘6’,’7’,’8’,’9’
VehType	Vehicle type	‘familiar’, ‘compact’, ‘sport’, ‘terrain’

Tab. 10Auto table description

Values for *VehBrand* and *VehPow* were extracted from f*reMTPL2freq* dataset.
Data for *VehBrand* field were extracted selecting 26.000 random values of the vehicle brand field from *freMTPL2freq*. But, only the distinct values are stored in *VehBrand* of the current table . In fact, the 26.000 registers for the vehicle brand are stored in the field *VehBrand* from *Policy table*. And by this way, *Auto table* and *Policy table* will be related by the common field *VehBrand*.
We can say, that *Auto table* is a parent table of *Policy table*, because extends the information about the vehicle brand.
Moreover, *VehPow* is the field with the corresponding powers for each one of the *VehBrand* values. At the last point, *VehType* values were assigned according to our opinion.
Finally, the resulting *Auto table* **Fig. 9** has 11 registers and 3 columns.

VehBrand	VehPow	VehType
1	6	compact
2	6	compact
3	6	familiar
4	6	compact
5	6	terrain
6	6	familiar
10	9	sport
11	9	sport
12	7	terrain
13	8	sport
14	7	sport

Fig. 9 Auto table

5.3.3 Region table

Region table corresponds to a dimension table, including regions information and it is composed by the following fields: *Code*, *Region* and *Population*. Fields are described in **Tab. 11**.

Numeric fields			
Field name	**Description**	**Min**	**Max**
Population	Population	78.476	6.489.680
Categorical fields			
Field name	**Description**	**Values**	

Code	Region identification	'1', '2', '3',..., '52'
Region	Region name	'Alava', 'Albacete', 'Alicante', ... (52 different values).

Tab. 11 Region table description

There had been supposed that regions were the 52 Spanish provinces. Each value of de *Code* field represented a region and *Population* was the population of each region extracted from (wikipedia).
Finally, the resulting *Region table* **Fig. 10** had 52 registers and 3 columns:

Code	Region	Population
1	Alava	319227
2	Albacete	402318
3	Alicante	1934127
4	Almeria	702819
5	Asturias	1081487
6	Avila	172704
7	Badajoz	693921
8	Barcelona	5529099
9	Burgos	375657
10	Caceres	415446

Fig. 10 Region table

5.3.4 RiskArea table

RiskArea table corresponds to a dimension table, including information about the type of insurance (risk area) and it is composed by the following fields: *RiskArea* and *RiskAreadesc.* They are described in **Tab. 12**.

Categorical fields		
Field name	**Description**	**Values**
RiskArea	Risk area identification	'1', '2', '3','4','5'
RiskAreadessc	Risk area description	'gold', 'silver', 'master', 'plus' and 'regular'.

Tab. 12 RiskArea table description

Values for *RiskArea* were extracted from *freMPL6* dataset but they were modified in order to reduce diversity.

We select the values for our 26.000 clients from the risk area field of *freMPL6*. But, only distinct values are stored in *RiskArea* field from the current table. In fact, the 26.000 registers for the risk area are stored in the field *RiskArea* from the *Policy table*. And by this way, *RsikArea table* and *Policy table* will be related by the common field *RiskArea*.

We can say, that *RiskArea table* is a parent table of *Policy table*, because extends the information about the risk area.

At the last point, a name was assigned to each risk area category by ourselves in the *RiskAreadesc* field.

Finally, the resulting *Risk Area table* **Fig. 11** has 5 registers and 2 columns.

RiskArea	RiskAreadesc
1	gold
2	silver
3	master
4	plus
5	regular

Fig. 11 RiskArea table

5.3.5 Guarantees table

Guarantees table corresponds to a dimension table, including information about guarantees offered for the insurance company and it is composed by the following fields: *Guarantees* and *Base.* They are described in **Tab. 13**.

Numeric fields			
Field name	**Description**	**Min**	**Max**
Base	The cost which is responsible the company	25	3.000
Categorical fields			
Field name	**Description**	**Values**	
Guarantees	Guarantees	‘windows’, ‘travelling’, ‘driver insurance’, ‘claims’, ‘fire’, ‘theft’, ‘total loss’, ‘health assistance’	

Tab. 13 Guarantees table description

Both fields were created by ourselves.

Finally, the resulting *Guarantees table* **Fig. 12** has 8 registers and 2 columns.

Guarantees	Base
windows	50
travelling	500
driver insurance	100
claims	25
fire	1000
theft	2000
total loss	3000
health assistance	300

Fig. 12 Guarantees table

5.3.6 RiskAreaXGuarantees table

RiskAreaXGuarantees table corresponds to a dimension table**,** which provides the relation between *RiskArea table* and *Guarantees table*. It shows which guarantees are offered by each risk area. And it is composed by the following fields: *RiskArea* and *Guarantees*. They are described in **Tab. 14**.

Categorical fields		
Field name	**Description**	**Values**
RiskArea	Risk area identification	'1', '2', '3','4','5'
Guarantees	Guarantees	'windows', 'travelling', 'driver insurance', 'claims', 'fire', 'theft', 'total loss', 'health assistance'

Tab. 14 RiskXGuarantees table description

We created the *RiskArea field* repeating the value of each risk area as many times as guarantees it offers. And *Guarantees* is the field with the respective guarantees offered by each risk area. Recall that it has a composed primary key because any of the fields is capable to identify uniquely the registers, but both together form a composed primary key.

Finally, the *resulting RiskXGuarantees table* showed in **Tab. 14**, has 29 registers (risk area 1 offers 8 guarantees, risk area 2 offers 7 guarantees, risk area 3 offers 6 guarantees, risk area 4offers 5 guarantees and risk area 5 offers 3 guarantees) and 2 columns.

RiskArea_guarantees	Guarantees
1	windows
1	travelling
1	driver insurance
1	claims
1	fire
1	theft
1	total loss
1	health assistance
2	windows
2	travelling
2	driver insurance
2	claims
2	fire
2	theft
2	health assistance
3	windows
3	driver insurance
3	claims
3	fire
3	theft
3	health assistance

Fig. 13 RiskAreaXGuarantees table

5.3.7 Policy table

Policy table corresponds to a dimension table, which includes information about the policies of the 26.000 clients and it is composed by the following fields: *PolicyID*, *ClientID*, *RecordBeg*, *RecordEnd*, *VehBeg*, *VehBrand*, *BonusMalus*, *RiskArea* and *Code*. Fields are described in **Tab. 15**.

Numeric fields			
Field name	**Description**	**Min**	**Max**
PolicyID	Policy identification	1	26.000
ClientID	Client identification	1	26.000
Date fields			
Field name	**Description**	**Min**	**Max**
RecordBeg	Policy starts date	2000-01-01	2010-12-31
RecordEnd	Policy ends date	2000-01-01	2010-12-31
VehBeg	Date when vehicle was build	1911-01-26	2011-01-01
Categorical fields			
Field name	**Description**	**Values**	
VehicleBrand	Vehicle Brand	‘1’,’2’,’3’,’4’,’5’,’6’,’10’,’11’,’12’,’13’,’14’	
BonusMalus	Bonus/Malus	50, 51, 51,... (75 different values)	
RiskArea	Risk Area included in the policy	‘1’,’2’,’3’,’4’,’5’	
Code	Region identification	‘1’, ‘2’, ‘3’,... (52 different values)	

Tab. 15Policy table description

The field *PolicyID* was created with values from 1 to 26.000. In that particular case, this field is equal to the *ClientID* because we were supposing that a client only had a policy in order to ease the analysis. The field *ClientID* let the join between *Policy table* and *Client table*.

In order to create the fields *RecordBeg* we took 26.000 random dates between ‘2000-01-01’ and ‘2010-12-31’, remembering that we wished data for 10 years. And in order to create *RecordEnd* field we have supposed that the probability to leave the policy is 0.3. It means, that with a probability of 0.3 we assigned the NULL date ‘9999-01-01’ to *RecordEnd* components. For the filled components we assigned randomly a date between one year after the *RecordBeg* and ‘2010-12-31’.

VehBeg field is composed by data extracted from the vehicle age field from *freMTPL2freq*, for the 26.000 randomly selected vehicles in *Auto table*. The values were previously modified because, in *freMTPL2freq*, the vehicle age was measured in years and we preferred to have the date when the car was building.

VehBrand is the field mentioned in the *Auto table* paragraph, with 26.000 values corresponding to the vehicle brand of the selected vehicles. And *RiskArea* is the field mentioned in the *RiskArea table* paragraph. The field *BonusMalus* corresponds to a risk indicator of the policy. Its data were extracted from the BonusMalus field of *freMPL6* dataset, for our 26.000 clients. Finally, the field *Code* refers to the code of the region where the policy is registered. It is created by assigning randomly numbers from 1 to 52 (each number refers to a region) depending on the population of each region.

Finally, the resulting *Policy table* **Fig. 14** has 26.000 registers and 8 columns.

PolicyID	ClientID	RecordBeg	RecordEnd	VehBeg	VehBrand	BonusMalus	RiskArea	Code
1	1	12/11/2006	01/01/9999	04/01/1998	1	50	4	50
2	2	05/02/2008	01/01/9999	05/01/1996	1	50	4	21
3	3	27/02/2000	01/01/9999	08/01/1984	2	68	5	29
4	4	21/09/2000	22/04/2010	03/01/2004	13	50	3	29
5	5	28/01/2004	01/01/9999	03/01/2004	2	50	5	40
6	6	29/03/2002	01/01/9999	01/01/2011	2	50	5	8
7	7	25/04/2000	01/01/9999	03/01/2003	13	50	3	29
8	8	15/05/2002	20/09/2010	04/01/1998	3	50	3	8
9	9	01/12/2003	13/05/2009	03/01/2001	10	50	5	40
10	10	17/11/2010	01/01/9999	01/01/2010	12	50	4	39
11	11	27/08/2001	01/01/9999	02/01/2005	2	50	4	8

Fig. 14 Policy table

5.3.8 SinistersXYears

SinistersXYears table corresponds to a fact table**,** which shows how many accidents are registered in each policy along the years from 2000 to 2010. This is a cross-tab **¡Error! No se encuentra el origen de la referencia.** and because of that the structure of that is more special than others. It is composed by the following fields: *PolicyID, Sinisters, 2000, 2001,...2010*. They are described in **Tab. 16**.

Qualifier field	
Field name	**Description**
ClientID	Policy identification
Attribute field	
Field name	**Description**
2000	Year
2001	Year
2002	Year
...	...
2010	Year
Data field	
Field name	**Description**
Sinisters	Amount of sinisters

Tab. 16 SinistersXYears table description

In order to assign the amount of accidents per year to each policy, we assigned a probability of 0,2 to have an accident in a year, to every policy. Depending on the characteristics of the client the probability could be increase.

- If client is younger than 24, the probability increases in “0.1”.
- If license is less than 12 months old, the probability increases “0.2”.
- If client is between 50 and 65 years old and is a Male the probability increases in “0.2”.
- If client it is between 40 and 45 years old and is a woman the probability increases in “0.2”.

After one accident, the probability of accident decreases on "0.1". And we supposed that a policy could not have more than 3 accidents in the same year.

The resulting *SinistersXYear table* Fig. 15 SinistersXYear table**Fig. 15** has 26.000 registers and 12 columns. The first column corresponds to the *ClientID* field and each one of the others corresponds to a year from 2000 to 2010.

ClientID	2000	2001	2002	2003
1	0	0	0	1
2	2	0	0	0
3	1	0	1	0
4	1	0	1	0

Fig. 15 SinistersXYear table

5.3.9 Sinisters table

This table corresponds to a fact table **¡Error! No se encuentra el origen de la referencia.,** including information about the accidents and it is composed by the following fields: *PolicyID*, *RiskArea*, *Guarantees*, *Sinisterdate* and *Code.* They are described in

Numeric fields			
Field name	**Description**	**Min**	**Max**
PolicyID	Policy identification	1	26.000
Date fields			
Field name	**Description**	**Min**	**Max**
Sinisterdate	Policy starts date	2000-01-01	2010-12-31
Categoric fields			
Field name	**Description**	**Values**	
RiskArea	Risk Area included in the policy	'1','2','3','4','5'	
Guarantees	Guarantees	'windows', 'travelling', 'driver insurance', 'claims', 'fire', 'theft', 'total loss', 'health assistance'	
Code	Region identification	'1', '2', '3',... (52 different values)	

Tab. 17 Sinisters table description

This table is based on *SinistersXYears table*, because *SinistersXYears table* fixes the amount of sinister for each policy.

Data for *Sinisterdate* are random dates with the year fixed for the *SinistersXYears table*. If a policy does not have any accident along the 11 years, it also appears in the table but with '9999-01-01' as *Sinisterdate*. It is not usual, to add a policy without any accident in that type of tables, but it is useful to analyse how the applications manage null values.

To create *Code* field, random numbers (referring to the code of the region) from 1 to 52 were assigned, but imposing that having a accident in the same region where the policy is registered,

is most probable (probability of "0.7") than in another region (probability of "0.3/51"). Moreover, we also imposed that in the particular region 'Granada', the probability of accident increases in summer for people who are not from the region or environs ('Jaen', 'Cordoba', 'Albacete', 'Malaga', 'Almeria') and decrease for people who are from the region or environs.

The *Guarantees* field has random guarantees as components depending on the risk area contracted by the policy.

At the last, *RiskArea* is a field with the same components as *RiskArea* from *Policy table* but repeating each value as many times as accidents have the particular policy. Finally, the resulting *Sinisters table* has 196.235 registers and 5 columns:

PolicyID	RiskArea	Guarantees	Sinisterdate	Code
1	4	theft	30/09/2003	29
1	4	windows	16/06/2006	9
1	4	windows	09/03/2008	50
1	4	driver insurance	20/09/2008	29
1	4	driver insurance	05/09/2009	29
1	4	health assistance	03/08/2010	29
2	4	windows	22/09/2000	29
2	4	health assistance	30/12/2000	21
2	4	windows	27/01/2008	21

Fig. 16 Sinisters table

The 9 datasets were exported in an Excel files, and from excel file they were loaded to the corresponding applications.

6 Evaluation Results

Once the metrics are chosen, weights are assigned to each metric, applications are selected and data are available, it is time to carry out the evaluation. The current evaluation was done only by me as an *Explorer* user. But, as it is said in **sub-chapter 3.3**, an evaluation should be done by several users, representing all the different types of users. In this project, it could not be possible, but in order to get concluding results, it should be done.

In order to store the scores, an excel sheet with the 82 metrics was build. It is where the evaluator have to fill the cells with the score for each of the metrics. The sheet was build considering the weights and the satisfaction score, established **in sub-chapters 3.3.1** and **3.3.2**, respectively.

The sheet was replicated identically assigning a sheet to each application. Therefore, a total of 4 excel sheets were filled by the evaluator. Reader can visualize them in **Annex 2**, although the whole file will be attached to the thesis.

Additionally to the sheets, user has to work with the particular Self-Service BI applications in order to evaluate them, and they must be available to him.

Mostly all Self-Service applications offer distinct editions. They usually have a Server Edition and a Personal Edition (also known as Desktop Editions). A Server edition is focused on companies. They offer the connection of several users to a central server which access to data and process them with much power than a local computer. Moreover, projects and data can be easily shared between users connected to the same server. On the other hand, Personal Editions, are single user editions, usually free trials, with the same operational characteristics, except for the connection to a server. And consequently, they can have limitations in the projects sharing. Moreover, the connection of different users to a server, usually implies the option of security devices, assigning permissions and passwords to data or projects. These options are not offered by Personal Editions.

In this project, the evaluated applications are **Server Editions**, although, in order to evaluate their operation, we used **Personal editions**, which can be installed in local PC and they are offered in their respective corporative webs, by free. Particularly, the used editions were:

QlikView View Personal Edition is the free trial for QlikView. With QlikView Personal Edition, user cannot open projects done by another Personal Edition's user and does not have security devices. As we said, QlikView Personal Edition cannot be connected to a server with other QlikView users. On the other hand, it can be connected to the same type of databases than QlikView.

MicroStrategy Analytics Desktop is the free edition for MicroStrategy Analytics Enterprise. MicroStrategy Analytics Desktop does not have any problem using projects done by other free edition's users. Moreover, it can be connected to the same type of databases than the MicroStrategy Analytics. But, it cannot be connected to a server with other users and it has not security devices.

SAP Lumira Desktop Standard Edtion used in the project is a 30-day free trial. This edition, is the personal edition of SAP Lumira Server. It can be connected to the same type of databases than the server one, and it can also open projects done by other users. SAP Lumira Desktop cannot be connected to a server with other users and it has not security devices.

Tableau Desktop used in this project is a 14-day free trial. This edition, is the personal edition of Tableau Server. It can be connected to the same type of databases than the server one, and it can also open projects done by other users. Although, it cannot be connected to a Server with other users and it has not security devices.

Once time, the user has the evaluation sheets and he has already used the corresponding applications with the database **20141220_Initial_test**, he has to score the metrics. Scoring the metrics is the key step in order to get results about each of the applications in each of the three categories: *Functionality*, *Usability* and *Efficiency*. Remember, that the results could not be considered as concluding, because more user opinions should be considered. If there are more evaluators, the four (4) sheets, the four (4) applications and the database **20141220_Initial_test** must be offered to each of them.

It must be consider, that the database used in the evaluation is in a excel file, and therefore we don't have the experience to connect tools to databases. Therefore the information about connecting to databases provided here is extracted from external sources(user guides or corporative webs) and our experience do not prove it.

6.1 Results

Once time every metric has been evaluated it is the time to get the results of the assessment. In the case than more than one user, is being implied in the evaluation of the metrics, we recommend to calculate a mean score for each metric.

On the other hand, one of the basis of the methodology of (Mendoza, Pérez, & Grimán, 2005) is that if *Functionality* category is not satisfied, the evaluation is aborted and other categories are not evaluated. Because of that, the analysis starts with the satisfaction score of *Functionality* category.

In the current evaluation, using the satisfaction limits mentioned in **Tab. 7** from **sub-chapter 3.3.1**, the obtained satisfaction scores for *Functionality* are showed in **Fig. 17**.

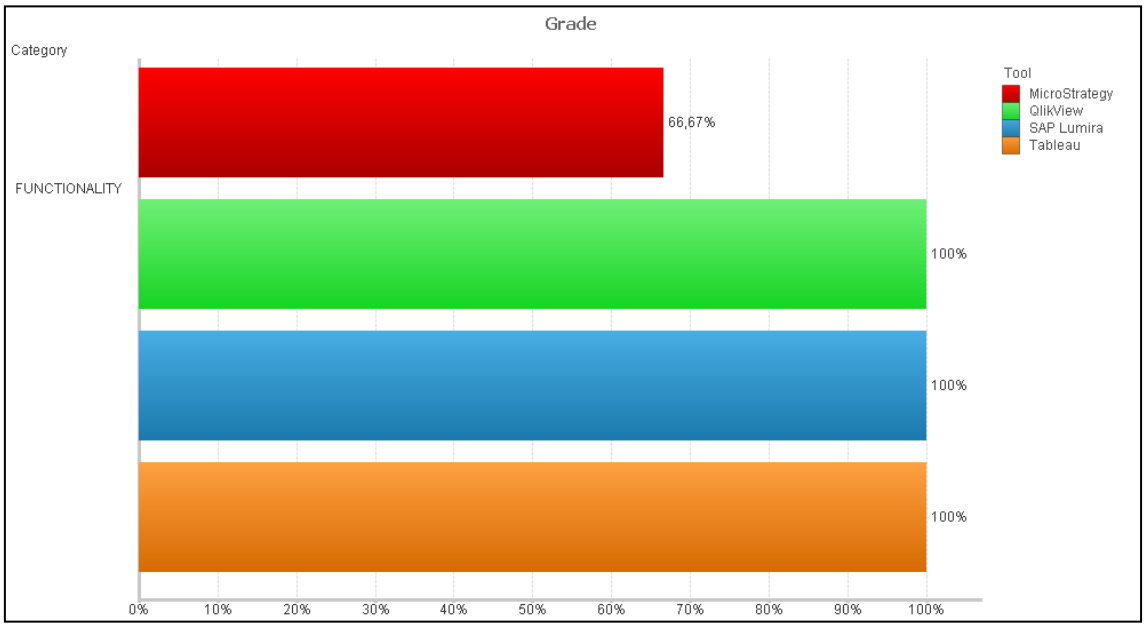


Fig. 17 Results for Functionality category

In the adaption of the methodology in **sub-chapter 3.3.1**, we sentence that a category is satisfied if the 75% of their characteristics are satisfied. And, applying that, MicroStrategy Analytics did not satisfy the *Functionality* category because it only satisfies the 66,67% of the functional characteristics. Then, the evaluation of MicroStrategy is aborted. On the other hand, the other three (3) tools satisfy the *Functionallity* category, because they satisfy the 100% of the corresponding characteristics.

In order to know the reason why MicroStrategy does not satisfy the *Functionality* category, a deeper level helped us to know what are the scores for each functional characteristic. Functional characteristics are *Fit to Purpose*, *Interoperability* and *Security*, and **Fig. 18** shows their respective

satisfaction score. We could see that the characteristic *Fit to purpose* is not satisfied because only the 66,67% of its sub-characteristics are satisfied. Particularly, the sub-characteristics non-satisfied are *Field Relations* and *Reporting*.

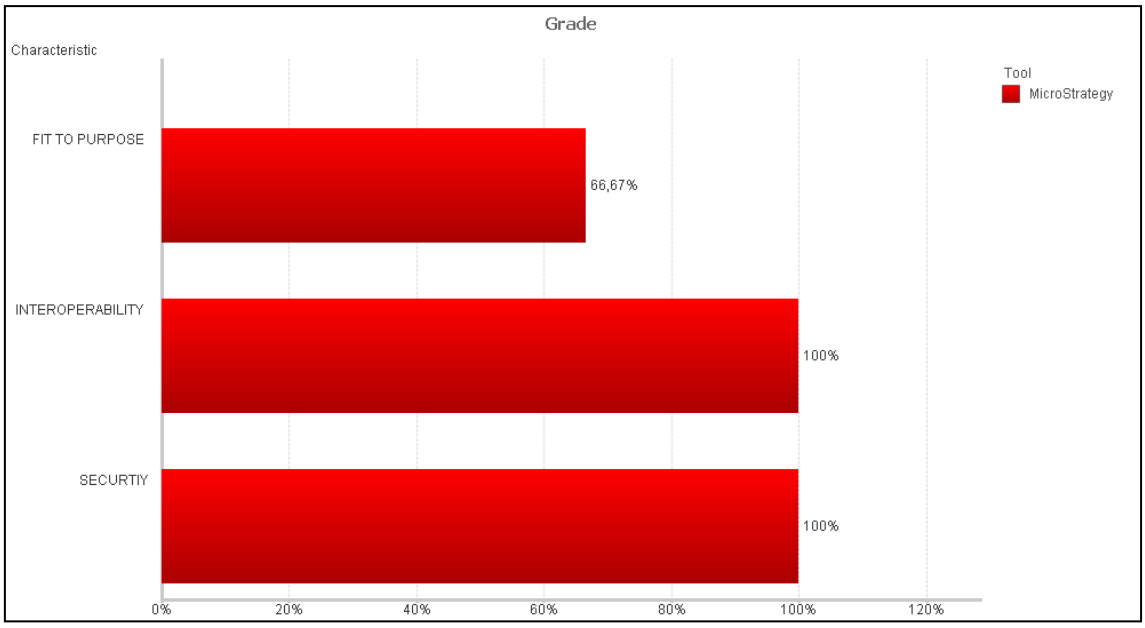


Fig. 18 Functinality characteristics results, for MicroStrategy Analytics

MicroStrategy Analytics does not satisfy the sub-characteristic *Fields relations* because it is not capable to alert about the presence of circular references *(FFF1)*, and in fact, it does not skip them *(FFF2)*. Moreover, it cannot directly relate a table to more than one table *(FFF3)*.
On the other hand, *Reporting* sub-characteristic, are not satisfied because MicroStrategy Analytics does not have an option to build reports *(FFR1)*, *(FFR2)*, *(FFR3)*.

Then, MicroStrategy evaluation is aborted and the evaluation followes for the other three (3) tools. The other three tools satisfy, additionally to the *Functionality*, the *Usability* category. Moreover, QlikView and Tableau satisfy also the *Efficieny* category, but SAP Lumira does not. **Fig. 19** shows the satisfaction score in each category.

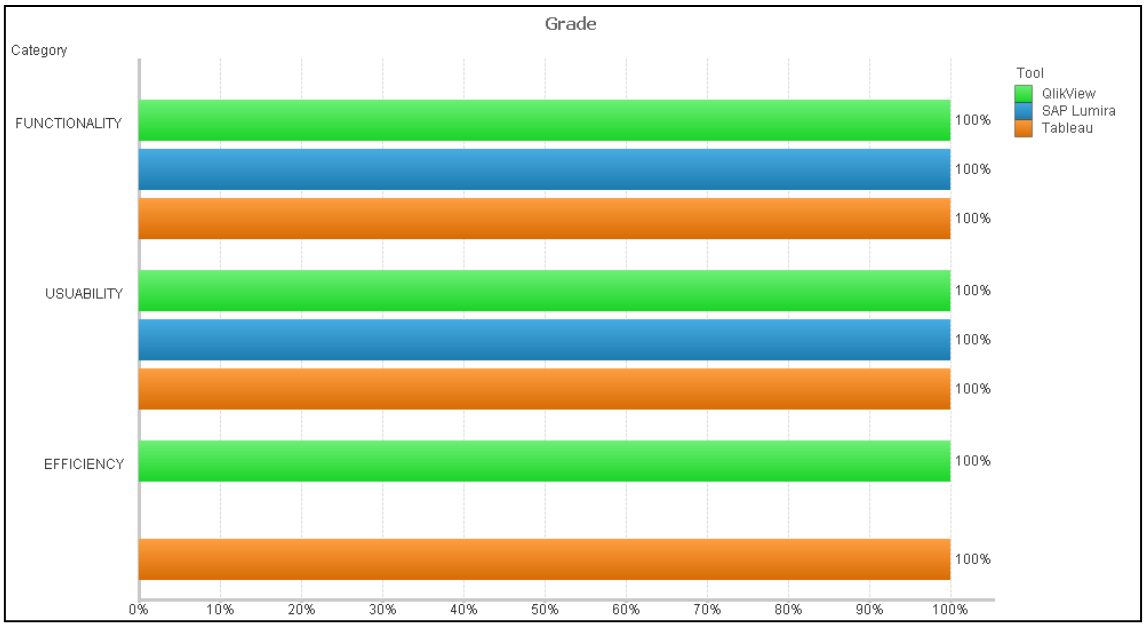


Fig. 19 Category results

SAP Lumira does not satisfy the *Efficiency* category. In fact, it does not satisfy the unique characteristic for *Efficiency*, which is *Execution Performance*, as it is shown in **Fig. 20**.

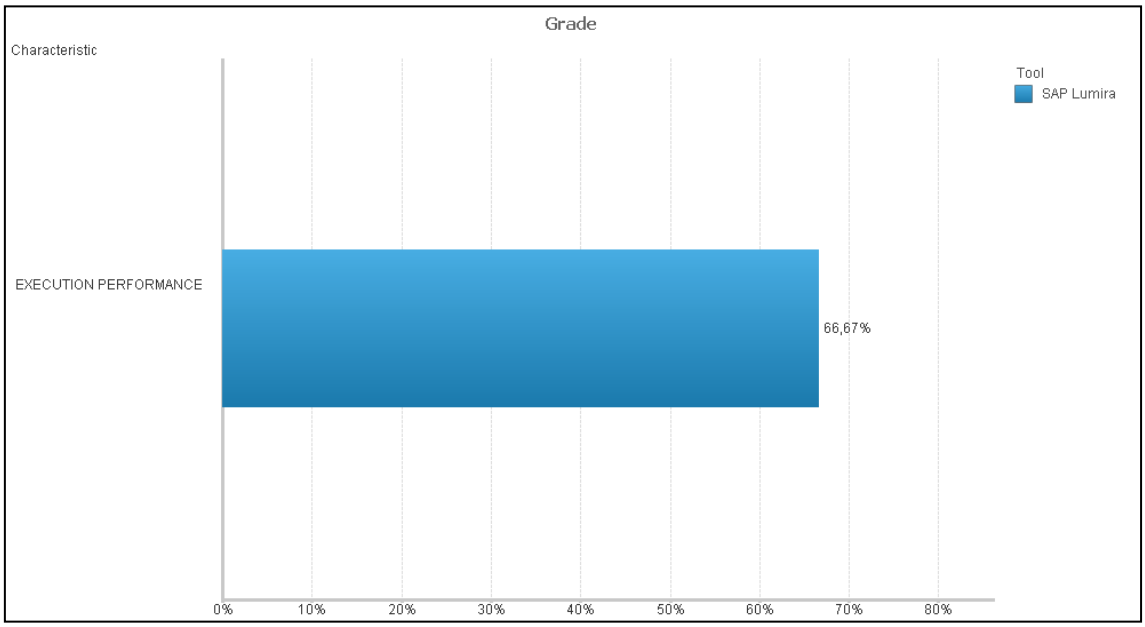


Fig. 20 Efficiency characteristics results, for SAP Lumira

This characteristic has a satisfaction score of 66,67%, lower than the fixed limit 75% and because of that it is considered as not satisfied. Only the 66,67% of the *Execution Performance* sub-characteristics are satisfied. In particular, **Fig. 21** shows the satisfaction scores for the corresponding sub-characteristics.

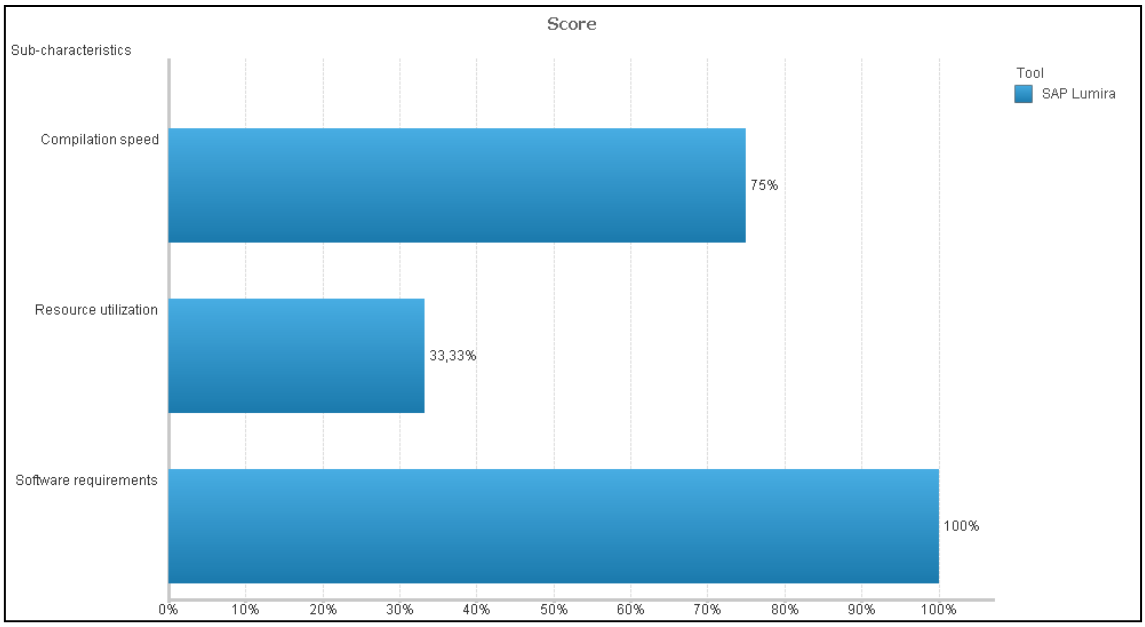


Fig. 21 Execution Performance sub-characteristics results, for SAP Lumira

Resources Utilization is not satisfied with a 33,33% of satisfaction score because it is the tool which requires more hard disk space *(EER3)* and additionally, SAP Lumira cannot be installed in processors of type x32 bits *(EER1)*.

Finally, according to table **Tab. 3** defined in **sub-chapter 3.2.1**, the product quality levels of QlickView, SAP Lumira and Tableau are:

Tool	Functionality	Usability	Efficiency	Quality level
Qlik View	*Satisfied*	*Satisfied*	*Satisfied*	*Advanced*
SAP Lumira	*Satisfied*	*Satisfied*	*No satisfied*	*Medium*
Tableau	*Satisfied*	*Satisfied*	*Satisfied*	*Advanced*

Fig. 22 Quality levels depending on satisfied categories. The particular case

Then, QlikView and Tableau offeres an advanced quality level while SAP Lumira has a medium quality level.

In order to get differences between QlikView and SAP Lumira, the fixed levels for satisfaction are increased, being more restrictive. Particularly, we use the following levels:

Limit for metric	50%
Limit for sub-characteristic	50%
Limit for Characteristic	80%
Limit for Category	75%

Tab. 18 Satisfaction Limits, for a second evaluation

By this way, a characteristic becomes satisfied if only the 80% of its sub-characteristics are satisfied. And as it can be seen in **Fig. 23**, only Tableau satisfy the *Functionality* category,

unlike QlikView, which does not, because only the 66,67% of its functional characteristics are satisfied.

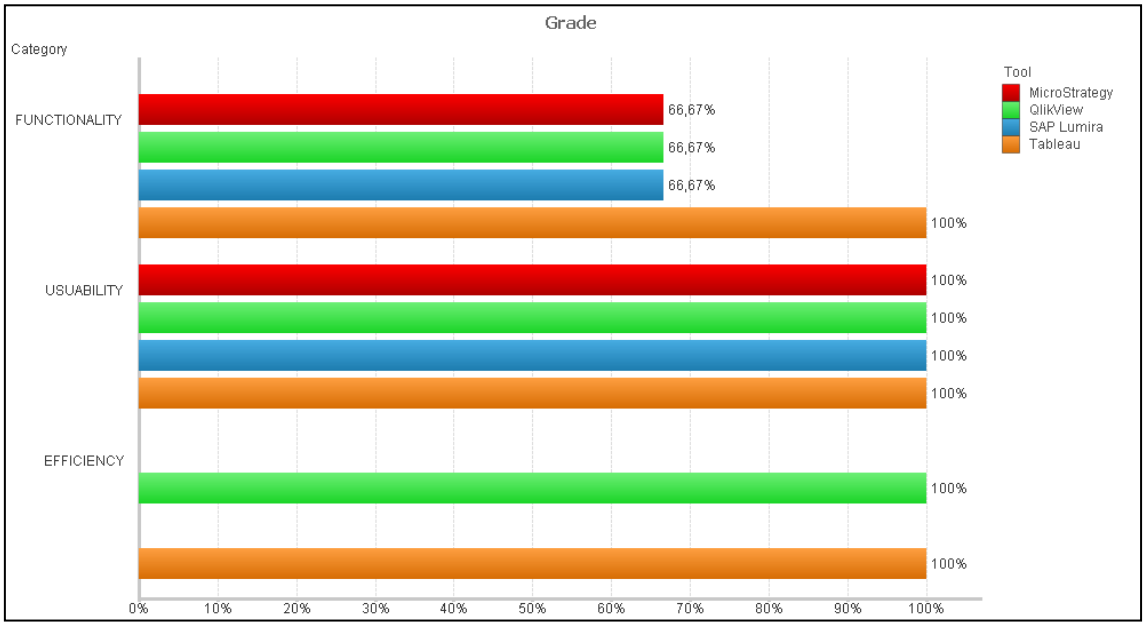


Fig. 23 Category results, for a second evaluation

In fact, as it its seen in **Fig. 24**, QlikView does not satisfy the *Functionality* category, because the *Interoperability* characteristic is not satisfied, in fact it has a score of 75%, meaning that only the 75% of the *Interoperability* sub-characteristics are satisfied.

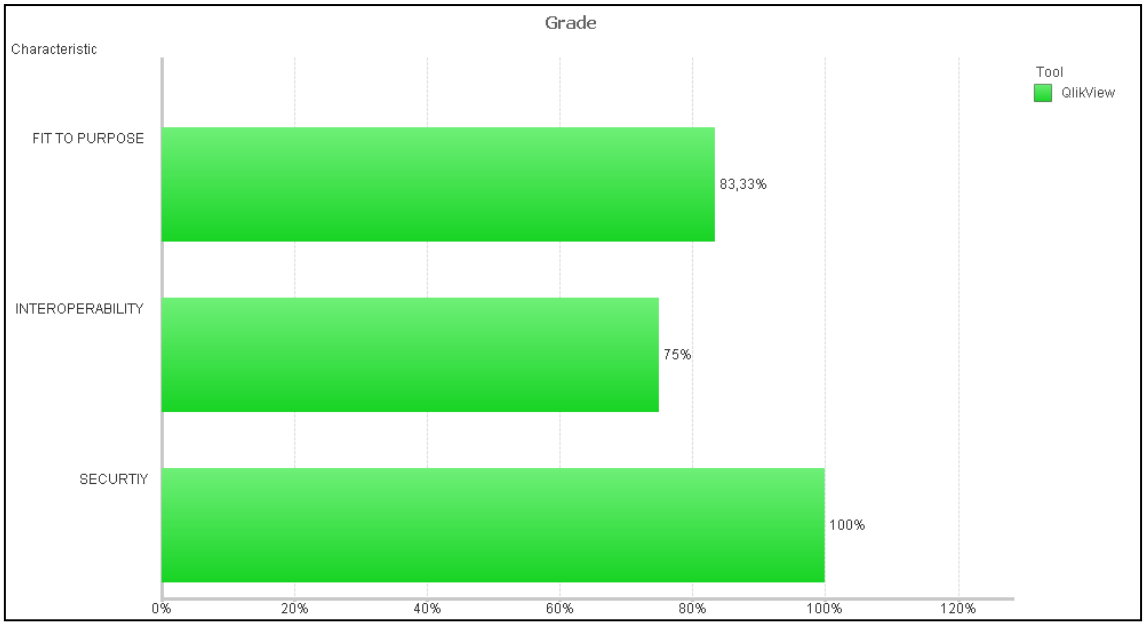


Fig. 24 Characteristic results, for a second evaluation

It is because, the *Portability* sub-characteristic is not satisfied, as a consequence of QlikView only works on Windows operating systems *(F1P1)* and it does not offer an available SaaS (Software as a service) edition *(FIP2)*.

Then, Tableau reached an advanced quality level. And it can be considered the most appropriate tool for the established requirements. Although, recalling that the results cannot be interpreted as concluding because only a user has done the evaluation.

There is detailed information about the evaluation in **Annex 3**.

7 Conclusions

This project had the purpose of building an assessment of Self-Service BI tools, and evaluate, in particular, 4 tools: MicroStrategy Analytics, QlikView, SAP Lumira and Tableau.

In order to build the assessment, an existing quality model was taken as a reference. It was the Systemic Quality Model (SQMO) developed by the Universidad Simón Bolívar (Venezuela). We adapted this model to our particular aims and then we established the metrics, with the help of a BI expert. To evaluate all the metrics established by the expert, we had to learn many new concepts of the BI area. We came from a different area of the BI world, and we had to learn the basics of all this concepts, and we also learned that many of this concepts are grouped in different professional specializations. Such as the complexity of the architecture behind a database, in that case, a relational database. Which, we learned during the creation of the database with R. Also, we learned different ways of storing data and its characteristics. Moreover, we learned the interests of companies in such kind of products. In fact, there were more metrics related to the ease of a exploratory analysis than related to the power of predictions of the tools.

Once, we were capable of evaluating the tools, we carried out the study. We learned that there can be a huge structure behind an evaluation and building it can be hard task, because we had to know about the area of BI and moreover we had to know how to measure them, additional to all the work done to get the database. While we were deciding how to measure the metrics, we realize that the cutoff of satisfaction it might be subjective. That is why we did the evaluation with two different satisfaction limits.

The first one established that a characteristic was satisfied if the 75% of its sub-characteristics were satisfied. While, the second one established that it was satisfied if the 80% of its sub-characteristics were satisfied. In both cases, the rest of the satisfaction limits kept constant. In the first scenario, we observed that Tableau and QlikView got an *advanced quality level*, unlike SAP Lumira, which got a *medium quality level*, and MicroStrategy, which was rejected according to the rules established by SQMO. And, in order to get differences between QlikView and Tableau, we did the second evaluation being more restrictive in the satisfaction limit. And the results were that Tableau got an *advanced quality level* and QlikView was rejected according to the rules established by SQMO.

As a further analysis, it would be adequate to run this evaluation by more users of different profiles *Farmers*, *Explorers* and *Tourist* in order to get proper results. And it would be advisable to try different satisfaction levels depending on the final purpose of the company.

8 Bibliography

Bass, L., Clements, P., & Kazzman, R. (1998). *Software Architecture in Practice (SEI Series in Software Engineering).* Massachussetts: Addison-Wesley Longman, Inc.

Callaos, N. C. (1993). Intention, action and design. *The 5th International Conversation on Comprehensive Systems Design* (pp. 36-47). Monterey, California: Asilomar.

Callaos, N., & Callaos, B. (1993). Intention, action and design. *The 5th International Conversation on Comprehensive Systems Design* (pp. 36-47). Monterey, California: Asilomar.

Callaos, N., & Callaos, B. (1996). Design with a systematic total quality. (pp. ISAS'96, 15-23). Proceeding of the International conference on information Systems Analysis and Syntesis.

Callaos, N., & Callaos, B. (1996). Designing with a systemic total quality. *Proceeding of the INternational Conference on Information Systems Analysis and Synthesis, ISAS'96*, 15-23.

Dromey, R. (1996). Cornering to Chimera. In *IEE Software* (Vol. 13, pp. 33-43).

Humphrey, W. (1997). *Introduction to the Personal Software Process.* Massachusetts: Addison Wesley Longman, Inc.

Inmon, W., Imhoff, C., & Susa, R. (1998). *Corporate Information Factory.* EE.UU: John Wiley & Sons, Inc.

Kitchenman, B. (1996). *Evaluating software engineering methods and tools, Part 5, Principles of Feature Analysis.* University of Keele, Department of Computer Science, England.

Mendoza, L. E., Pérez, M. A., & Grimán, A. C. (2005). Prototipo de Model Sistémico de Calidad (MOSCA) del Software. *Computación y Sistemas, 8*(3), 196-217.

Ortega, M., Pérez, M., & Rojas, T. (2000). A Model for Software Product Quality with a Systemic Focus. *4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems, Analysis and Synthesis ISAS 2000*, (pp. 395-401). Orlando, Florida.

Pérez, M., Rojas, T., Mendoza, L., & Grimán, A. (2001). Systemic Quality for System Development Process: Case Study. *Seventh Americas Conference on Information Systems-AMCIS 2001*, (pp. 1297-1304). Boston, Massachusetts.

Rincon, G., Alvarez, M., Perez, M., & Hernandez, S. (2005). A discrete-event simulation and continuous software evaluation on a systemic quality model: An oil industry case. *Elsevier, Information & Management*(42), 1051-1066.

Sallam, R. L., Hostmann, B., Schlegel, K., Tapadinhas, J., Parenteau, J., & Oestreich, T. W. (2015). Magic Quadrant for Business Intelligence and Analytics Platforms.

wikipedia. (n.d.). *Provincias y ciudades autònomas de España*. Retrieved from : http://es.wikipedia.org/wiki/Anexo:Provincias_y_ciudades_aut%C3%B3nomas_de_Espa%C3%B1a

9 Figures index

10 Tables index

Annex 1 : Scripts for 20141220_Initial_test database

During the simulation process, some variables were called different than in the database. It Because of the relationships (1:n or n:m) , foreign key fields were build two times with different lengths, one time for each table from where they pertained. And to keep the consistency in the R code, they had to be considered as different fields, and for this reason they were called different. Although, once the database was created, the names were changed in order to build relations between tables by a common field. **Tab. 18** shows the fields which take a different name during the simulation.

Variable Name in Simulation	Table	Data base field
VehBrand_N	Policy	VehBrand
RiskArea_N	Policy	RiskArea
Code_O	Policy	Code
V1,…,V11	SinistersXYear	2000, …, 2010
PolicyID_S	Sinisters	PolicyID
RiskArea_s	Sinisters	RiskArea
Guarantees_s	Sinisters	Guarantees
Code_S	Sinisters	Code

```
##  CLIENT TABLE  ##

#LIBRARIES

######################

library(xts)

library(zoo)

library(CASdatasets)

data(freMPL6)

#PARAMETERS

N<-26000

#FIELDS

ClientID<-c(1:N)

Gender<-freMPL6$Gender[ClientID]

MariStat<-freMPL6$MariStat[ClientID]

CSP<-freMPL6$SocioCateg[ClientID]

actual<-as.Date('2011-01-01')
```

```
LicAge<-freMPL6$LicAge[ClientID]

 LicBeg<-actual-(LicAge*30)

 DrivAge<-freMPL6$DrivAge[ClientID]

 DrivBeg<-actual-(DrivAge*365)

#DATAFRAME

df_Client<-data.frame(ClientID, Gender, MariStat, CSP, LicBeg, DrivBeg)

View(df_Client)

#EXPORTATION

library(foreign)

write.table(df_Client, "C:/Users/jorcajo/Desktop/Tables/Client.txt", sep="\t", row.names=F)


##   AUTO TABLE   ##
#LIBRARIES
library(xts)
library(zoo)
library(CASdatasets)
data(freMTPL2freq)
#FIELDS
 set.seed(12342)
autos<-sample(1:dim(freMTPL2freq)[1], N, replace=FALSE)
VehBrand_N<- freMTPL2freq$VehBrand[autos]
VehBrand<-levels(factor(VehBrand_N))
 ########################
#Cluster power in few categories depending on the VehBrand. For each VehBrand, I do the
power mean:
   #and select the means as the new levels for VehPow. From freMTPL2freq.
 #> levels(factor(freMTPL2freq$VehPow))
   #[1] "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
 #> levels(factor(freMTPL2freq$VehBrand))
   #[1] "1"  "2"  "3"  "4"  "5"  "6"  "10" "11" "12" "13" "14"
 ########################
 l<-length(levels(factor(freMTPL2freq$VehBrand)))
 VehPow<-NULL
 for (i in 1:l){
   pow<-
freMTPL2freq$VehPow[which(freMTPL2freq$VehBrand==levels(factor(freMTPL2freq$VehBran
d))[i])]
   VehPow[i]<-round(mean(pow),0)
 }
 ########################
```

```
 #> VehPow
 #[1] "6" "6" "6" "6" "6" "6" "9" "9" "7" "8" "7"
 ##########################
 VehType<-c("compact", "compact", "familiar", "compact", "terrain", "familiar",
        "sport", "sport", "terrain", "sport", "sport")
#DATAFRAME
df_Auto<-data.frame(VehBrand, VehPow, VehType)
View(df_Auto)
#EXPORTATION
library(foreign)
write.table(df_Auto, "C:/Users/jorcajo/Desktop/Tables/Auto.txt", sep="\t", row.names=F)

## REGION TABLE ##
#FIELDS
Code<-c(1:52)
Region<-c("Alava", "Albacete", "Alicante", "Almeria", "Asturias", "Avila", "Badajoz",
"Barcelona",
         "Burgos", "Caceres", "Cadiz", "Cantabria", "Castellon", "Ciudad Real", "Cordoba",
         "La Coruna", "Cuenca", "Gerona", "Granada", "Guadalajara", "Guipuzcoa", "Huelva",
"Huesca",
         "Islas Baleares", "Jaen", "Leon", "Lerida", "Lugo", "Madrid", "Malaga", "Murcia",
"Navarra",
         "Orense", "Palencia", "Palmas", "Pontevedra", "La Rioja", "Salamanca", "Segovia",
"Sevilla",
         "Soria", "Tarragona", "Teruel", "Tenerife", "Toledo", "Valencia", "Valladolid",
"Vizacaia", "Zamora",
         "Zaragoza", "Melilla", "Ceuta")
Population<-c(319227, 402318, 1934127, 702819, 1081487, 172704, 693921, 5529099,
375657, 415446,
        1243519, 593121, 604344, 530175, 805857, 1147124, 219138, 756810, 924550,
256461, 709607,
        521968, 228361, 1113114, 670600, 529799, 442308, 351350, 6489680, 1625827,
1470069, 642051, 333257, 171668,1096980,
        963511, 322955, 352986, 164169, 1928962, 95223, 811401, 144607,1029789, 707242,
        2578719,534874, 1155772, 191612, 973325, 78476, 82376)
#DATAFRAME
df_Region<-data.frame(Code, Region, Population)
View(df_Region)
#EXPORTATION
library(foreign)
write.table(df_Region, "C:/Users/jorcajo/Desktop/Tables/Region.txt", sep="\t", row.names=F)
```

```
##  RISKAREA TABLE ##
#FIELDS
RiskArea_N<-freMPL6$RiskArea[ClientID]
#Clusters RiskArea in 5 group depending on their frequency
#> sort(table(freMPL6$RiskArea))
#1   13   12    2    3    4    8    5   11    9    6   10    7
#13   25   57  345  471  895 1236 1535 3022 3828 3864 4620 6089
##############################
for(i in 1:N){
 if(RiskArea_N[i]==1 || RiskArea_N[i]==13 || RiskArea_N[i]==12 ){
  RiskArea_N[i]<-1
 }
 if(RiskArea_N[i]==2 || RiskArea_N[i]==3 || RiskArea_N[i]==4 ){
  RiskArea_N[i]<-2
 }
 if(RiskArea_N[i]==8 || RiskArea_N[i]==5 || RiskArea_N[i]==11 ){
  RiskArea_N[i]<-3
 }
 if(RiskArea_N[i]==9 || RiskArea_N[i]==6){
  RiskArea_N[i]<-4
 }
 if(RiskArea_N[i]==10 || RiskArea_N[i]==7 ){
  RiskArea_N[i]<-5
 }
}
RiskArea<-levels(factor(RiskArea_N))
 #RISKAREADESC
RiskAreadesc<-c("gold", "silver", "master", "plus", "regular")
#DATAFRAME
df_RiskArea<-data.frame(RiskArea, RiskAreadesc)
View(df_RiskArea)
#EXPORTATION
library(foreign)
write.table(df_RiskArea, "C:/Users/jorcajo/Desktop/Tables/RiskArea.txt", sep="\t",
row.names=F)


## RISKAREAXGUARANTEES ##
#FIELDS
RiskArea_guarantees<-c(rep(1, 8), rep(2, 7), rep(3, 6), rep(4, 5), rep(5, 3))
g1<-c("windows", "travelling", "driver insurance", "claims", "fire", "theft", "total loss", " health
assistance")
g2<-c("windows", "travelling", "driver insurance", "claims", "fire", "theft", " health assistance")
g3<-c("windows", "driver insurance", "claims", "fire", "theft", " health assistance")
```

```
g4<-c("windows", "driver insurance", "fire", "theft", " health assistance")
g5<-c("windows", "driver insurance", " health assistance")
Guarantees<-c(g1, g2, g3, g4, g5)
#DATAFRAME
df_RiskAreaXGuarantees<-data.frame(RiskArea_guarantees, Guarantees)
View(df_RiskAreaXGuarantees)
#EXPORTATION
library(foreign)
write.table(df_RiskAreaXGuarantees,
"C:/Users/jorcajo/Desktop/Tables/RiskAreaXGuarantees.txt", sep="\t", row.names=F)


##   POLICY TABLE  ##
#FIELDS
PolicyID<-c(1:N)
prob<-(Population/sum(Population))
#probabilities of each region depending on its population
set.seed(12342)
Code_O<-sample(c(1:52), N, replace=TRUE, prob=prob)
set.seed(12342)
inici <- as.Date('2000-1-1')
fi <- as.Date('2010-12-31')
dates <- as.Date(inici:fi, origin='1970-1-1')
 RecordBeg<-sample(dates, N, replace=T)
RecordEnd<-rep(0, N)
RecordEnd<-as.Date(RecordEnd, origin='1970-1-1')
#Supposing that the 70% of the clients keep
#their policy during the following 10 years
  for(i in 1:N){
    x<-sample(c(0,1), 1, replace=TRUE, prob=c(0.7, 0.3))  #x=0 means that there is no an end
date for the policy.
    if(x==0){
      RecordEnd[i]<-"9999-01-01"
    }
    else{                          #Supposing that policies keep, one year as minimum, in the
company.
      inici<-RecordBeg[i]+365
      dates <- as.Date(inici:fi, origin='1970-1-1')
      RecordEnd[i]<-sample(dates, 1)
    }
  }
  VehAge<-freMTPL2freq$VehAge[autos]
  actual<-as.Date('2011-01-01')
  VehBeg<-actual-(VehAge*365)
BonusMalus<-freMPL6$BonusMalus[ClientID]
```

```
#DATAFRAME
df_Policy<-data.frame(PolicyID, ClientID, RecordBeg, RecordEnd, VehBeg,  VehBrand_N,
BonusMalus, RiskArea_N, Code_O)
View(df_Policy)
#RiskArea_N is created in RiskArea script.
#EXPORTATION
library(foreign)
write.table(df_Policy, "C:/Users/jorcajo/Desktop/Tables/Policy.txt", sep="\t", row.names=F)



##   SINISTERS TABLE AND SINISTERXYEAR TABLE  ##
#LIBRARIES
library("lubridate", lib.loc="C:/Program Files/R/R-3.1.1/library")
inici <- as.Date('2000-1-1')
 fi <- as.Date('2010-12-31')
 dates <- as.Date(inici:fi, origin='1970-1-1')
 #The vector dates is modified in order to delete all the 29th February to prevent errors.
  dates_sinisters<-dates
  for(i in 1:length(dates_sinisters)){
    if(day(dates_sinisters[i])==29 && month(dates_sinisters[i])==02){
      dates_sinisters[i]<-dates_sinisters[i]-1
    }
  }

#PROBABILITIES OF SINISTER
#every client has a probability of 0.2 to have a sinister, as minimum.
p<-rep(0.2, N)
#Some characteristics make this probability increases
for ( i in 1:N){
if(DrivAge[i]<24){
     p[i]<-p[i]+0.1
 }
if(LicAge[i]<12){
     p[i]<-p[i]+0.2
 }
if(DrivAge[i]>50 && DrivAge[i]<65 && Gender[i]=="Male"){
     p[i]<-p[i]+0.2
 }
if(DrivAge[i]>40 && DrivAge[i]<45 && Gender[i]=="Female"){
     p[i]<-p[i]+0.2
 }
}
#SINISTERDATE FIELD AND SINISTERXYEAR MATRIX         #It spend 20 minutes
```

```
Sinisterdate<-as.Date('1970-1-1')
SinistersXYear<-matrix(data=0, nrow=N, ncol=11)
set.seed(12342)
for( i in 1:N){
   nsinisters<-rep(0,11)
   for( j in 1:11){
    prob<-p[i]
    s<-NULL
    for(k in 1:3){                          #A maximum of 3 accidents per year.
     s[k]<-sample(c(0,1), 1, replace=TRUE, prob=c((1-prob), prob))
     if (s[k]!=0) {
      prob<-(prob-0.1)                      #After a happening a sinister, the probability to
     }                                      #have a sinister decreases.
    }
   }
    if(sum(s)!=0){
     Sinisterdate_2<-sample(dates_sinisters, sum(s), replace=FALSE)
     year(Sinisterdate_2)<-2000+j-1
     Sinisterdate<-c(Sinisterdate, Sinisterdate_2)
    }
    nsinisters[j]<-sum(s)                   #total number of sinisters for policy i in year j
   }
   SinistersXYear[i, ]<-nsinisters
   if(sum(nsinisters)==0){
    Sinisterdate<-c(Sinisterdate, as.Date('9999-01-01'))

}
}
Sinisterdate<-Sinisterdate[2:length(Sinisterdate)]      #Delete the first value of Sinisterdate
#DATAFRAME SINISTERSXYEAR
df_SinistersXYear<-data.frame(ClientID, SinistersXYear)
View(df_SinistersXYear)
#EXPORTATION SINISTERSXYEAR
write.table(df_SinistersXYear, "C:/Users/jorcajo/Desktop/Tables/SinistersXYear.txt", sep="\t",
row.names=F)


#OTHER FIELDS
#Code_S
l<-length(Sinisterdate)
Code_S<-rep(0, l)
Code_O_sinisters<-rep(0, l)
probs<-rep((1/52), 52)
total<-rep(0,N)
for (i in 1:N){
```

```
   total[i]<-sum(SinistersXYear[i, ])
 }
#Code_O_SINISTERS                          #Code_O_sinisters is needed to create Code_S
 j<-1
for( i in 1:N){
   if(total[i]!=0){
     for(k in 0:(total[i]-1)){
       Code_O_sinisters[j+k]<-Code_O[i]
     }
     j<-j+k+1
   }
  if(total[i]==0){
     Code_O_sinisters[j]<-Code_O[i]
     j<-j+1
  }
}
probs<-NULL                                #Code_S is equal to Code_O with a probability of 0.7
set.seed(12342)
for( i in 1:l){
   for( j in 1:52){
     if(Code_O_sinisters[i]==Code[j]){
       probs[1:(j-1)]<-(0.3/51)
       probs[j]<-0.7
       probs[(j+1):52]<-(0.3/51)
       probs<-probs[1:52]
       Code_S[i]<-sample(Code, 1, prob=probs)
     }
   }
}
PolicyID_s<-rep(0,l)
RiskArea_s<-rep(0, l)
Guarantees_s<-rep(0, l)
set.seed(12342)
j<-1
for( i in 1:N){
   if(total[i]!=0){
     for(k in 0:(total[i]-1)){
       PolicyID_s[j+k]<-PolicyID[i]
       RiskArea_s[j+k]<-RiskArea_N[i]
       if(Code_O_sinisters[j+k]!=25 && Code_O_sinisters[j+k]!=19 && Code_O_sinisters[j+k]!=15
          && Code_O_sinisters[j+k]!=2 && Code_O_sinisters[j+k]!=30 &&
Code_O_sinisters[j+k]!=4
          && month(Sinisterdate[j+k])%in%c(7,8,9)){
         p<-probs-0.002
         p[19]<-0.104                                    #19 is the position for "Granada"
```

```
      Code_S[j+k]<-sample(Code, 1, replace=TRUE, prob=p)
     }
     if (RiskArea_N[i]==1){
      Guarantees_s[j+k]<-sample(g1,1)
     }
     if(RiskArea_N[i]==2){
      Guarantees_s[j+k]<-sample(g2, 1)
     }
     if(RiskArea_N[i]==3){
     Guarantees_s[j+k]<-sample(g3, 1)
     }
     if(RiskArea_N[i]==4){
      Guarantees_s[j+k]<-sample(g4, 1)
     }
     if(RiskArea_N[i]==5){
      Guarantees_s[j+k]<-sample(g5, 1)
     }
   }
    j<-j+k+1
 }
  if(total[i]==0){
   PolicyID_s[j]<-PolicyID[i]
   RiskArea_s[j]<-RiskArea_N[i]
   Guarantees_s[j]<-""
   Code_S[j]<-""
   j<-j+1
  }
}
#length(Guarantees_s)
#length(PolicyID_s)
#length(RiskArea_s)
sinisterID<-rep(0, length(sinisterdate2))
 min<-fi
 for (j in 1:216856){
  for(i in 1:216856){
   if(sinisterdate2[i]<=min){
    min<-sinisterdate2[i]
   }
  }
  sinisterID[i]<-j
 }
#DATAFRAME
df_Sinisters<-data.frame(PolicyID_s, RiskArea_s, Guarantees_s, Sinisterdate, Code_S)
View(df_Sinisters)
#EXPORTATION
```

```
library(foreign)
write.table(df_Sinisters, "C:/Users/jorcajo/Desktop/Tables/Sinisters.txt", sep="\t",
row.names=F)
```

Annex 2 : Questionnaires

This annex attaches the questionnaires filled by the unique user implied in the evaluation, me. There 4 questionnaires, one for each evaluated tool. They shows the scores, according to the scale of measurement established in **sub-chapter 3.3.1**, for each metric. Moreover, according to the satisfaction limits established in **subchapter 6.1** for the second evaluation, the satisfaction score is showed. Recall, that the whole excel file will be attached with this thesis.

- The column **M.S** refers to the scale of measurement established for each metric.
- The column **WEIGHTS** refers to the weights established for each metric in **sub-chapter 3.3.1**.
- The **COMPENSED VALUE** refers to the product of the weight and the metric's score.
- The **NORMALIZED VALUE** is the satisfaction score for the metric. While, for sub-characteristics/characteristics/categories they are called simply **TOTAL.**
- The columns called **INDICATOR** take values 1 or 0, depending on if the Metric/Sub-characteristic/Characteristic/Category is satisfied according to the satisfaction limit established in **subchapter 6.1.**

CATEGORY	CHARACTERISTIC	SUBCHARACTERISTIC-DESC	METRIC-CODE	METRIC	M.S	VALUE	WEIGHT	COMPENSED VALUE	NORMAL. VALUE	INDICATOR_METRIC	TOTAL SUB-CHARACTERISTIC	INDICATOR_SUB-CHARAC.	TOTAL CHARACTERISTIC	INDICATOR_CHARAC	TOTAL CATEGORY	INDICATOR_CATEG.
FUNCTIONALITY	FIT TO PURPOSE	Data loading	FFI1	Direct connection to data sources	A	3	2	6	75,00%	1						
			FFI2	BigData sources	A	3	1	3	75,00%	1						
			FFI3	Apache Hadoop	A	3	1	3	75,00%	1						
			FFI4	Microsoft Access	A	3	2	6	75,00%	1						
			FFI5	Excel files	A	3	3	9	75,00%	1						
			FFI6	From an excel file, import all sheets at the same time	A	0	2	0	0,00%	0						
			FFI7	Cross-tabs	A	1	2	2	25,00%	0						
			FFI8	Plain text	A	3	3	9	75,00%	1						
			FFI9	Connecting to different data sources at the same time	A	3	2	6	75,00%	1						
			FFI10	Easy integration of many data sources	A	3	2	6	75,00%	1						
			FFI11	Visualizing data before the loading	A	3	2	6	75,00%	1						
			FFI12	Determining data format	A	3	2	6	75,00%	1						
			FFI13	Determining data type	A	3	2	6	75,00%	1						
			FFI14	Allowing column filtering before the loading	A	3	2	6	75,00%	1						
			FFI15	Allowing row filtering before the loading	A	0	2	0	0,00%	0						
			FFI16	Automatic measures creation	A	3	3	9	75,00%	1						
			FFI17	Allow renaming datasets	A	3	2	6	75,00%	1						
			FFI18	Allow renaming fields	A	3	3	9	75,00%	1						
			FFI19	Data cleansing	A	2	2	4	50,00%	1	**85,00%**	**1**				
		Data model	FFD1	Data model is done automatically	A	4	2	8	100,00%	1						
			FFD2	The done data model is the correct one	A	3	2	6	75,00%	1						
			FFD3	Data model can be visualized	A	1	3	3	25,00%	0	**57,14%**	**1**				
		Field relations	FFF1	Alerting about circular references	A	0	3	0	0,00%	0						
			FFF2	Skiping with circular references	A	0	3	0	0,00%	0						
			FFF3	A same table can be used several times	A	0	2	0	0,00%	0	**0,00%**	**0**				
		Analysis	FFA1	Creating new measures based on previous measures	A	3	3	9	75,00%	1						
			FFA2	Creating new measures based on dimensions	A	2	3	6	50,00%	1						
			FFA3	Variety of functions	A	4	3	12	100,00%	1						
			FFA4	Descriptive statistics	A	3	2	6	75,00%	1						
			FFA5	Preduction functions	A	2	2	4	50,00%	1						
			FFA6	R connection	A	3	2	6	75,00%	1						
			FFA7	Geographic information	A	3	2	6	75,00%	1						
			FFA8	Time hierarchy	A	3	3	9	75,00%	1						
			FFA9	Creating sets of data	A	3	2	6	75,00%	1						
			FFA10	Filtering data by expression	A	3	3	9	75,00%	1						
			FFA11	Filtering data by dimension	A	3	3	9	75,00%	1						
			FFA12	Visual Perspective Linking	A	3	2	6	75,00%	1						
			FFA13	No Null data specifications	A.1	0	2	0	0,00%	0						
			FFA14	Considering nulls	A	1	3	3	25,00%	0						
			FFA15	Variety of graphs	A	3	3	9	75,00%	1						
			FFA16	Modify graphs	A	3	3	9	75,00%	1						
			FFA17	No limitations to display large amounts of data	A.1	3	2	6	75,00%	1						
			FFA18	Data refresh	A	2	2	4	50,00%	1	**88,89%**	**1**				
		Dashboards	FFD1	Dashboards Exportation	A	3	3	9	75,00%	1						
			FFD2	Templates	A	0	2	0	0,00%	0	**71,43%**	**1**	**66,67%**	**0**	**66,67%**	**0**

			FFD3	Free design	A	3	2	6	75,00%	1						
		Reporting	FFR1	Reports Exportation	A	1	3	3	25,00%	0						
			FFR2	Templates	A	0	2	0	0,00%	0						
			FFR3	Free design	A	1	2	2	25,00%	0	**0,00%**	**0**				
	INTEROPERABILITY	Languages	FIL1	Languages displayed	A.1	4	2	8	100,00%	1	**100,00%**	**1**				
		Portability	FIP1	Operating Systems	A.1	0	2	0	0,00%	0						
			FIP2	SaaS/Web	A	3	1	3	75,00%	1						
			FIP3	Mobile	A	2	2	4	50,00%	1	**60,00%**	**1**				
		Using the project by third parts	FIU1	Using the project by third parts	A	3	2	6	75,00%	1	**75,00%**	**1**				
		Data exchage	FID1	Exportation in txt	A	3	2	6	75,00%	1						
			FID2	Exportation in CSV	A	3	2	6	75,00%	1						
			FID3	Exportation in HTML	A	0	2	0	0,00%	0						
			FID4	Exportation in Excel file	A	3	3	9	75,00%	1	**77,78%**	**1**	**100,00%**	**1**		
	SECURTIY	Security devices	FSS1	Password protection	A	3	3	9	75,00%	1						
			FSS2	Permissions	A	3	3	9	75,00%	1	**100,00%**	**1**	**100,00%**	**1**		
USUABILITY	EASE OF UNDERSTANDING AND LEARNING	Learning time	UEL1	Average learning time	A.2	2	3	6	50,00%	1	**50,00%**	**1**				
		Browsing facilities	UEB1	Consistency between icons in the toolbars and their actions	A	3	3	9	75,00%	1						
			UEB2	Displaying right click menus	A	0	3	0	0,00%	0	**50,00%**	**1**				
		Terminology	UET1	Ease of understanding the terminology	A	3	3	9	75,00%	1	**75,00%**	**1**				
		Help and documentation	UEH1	User guide quality	A.2	4	2	8	100,00%	1						
			UEH2	User guide adquisition	A	3	2	6	75,00%	1						
			UEH3	On-line documentation	A	3	2	6	75,00%	1	**100,00%**	**1**				
		Support training	UES1	Availability of tailor-made training courses	A	3	2	6	75,00%	1						
			UES2	Phone technical support	A	2	2	4	50,00%	1						
			UES3	On-line support	A	2	2	4	50,00%	1						
			UES4	Availability of consulting services	A	3	2	6	75,00%	1						
			UES5	Free formation	A	4	2	8	100,00%	1						
			UES6	Community	A	3	2	6	75,00%	1	**100,00%**	**1**	**100,00%**	**1**		
	GRAPHICAL INTERFACE CHARACTERISTIC	Windows and mouse interface	UGW1	Editing elements by double-clicking	A	0	2	0	0,00%	0						
			UGW2	Dragging and dropping elements	A	3	2	6	75,00%	1	**50,00%**	**1**				
		Display	UGD1	Editing the screen layout	A	3	2	6	75,00%	1	**75,00%**	**1**	**100,00%**	**1**		
	OPERABILITY	Versatility	UOV1	Automatic update	A	3	2	6	75,00%	1	**75,00%**	**1**	**100,00%**	**1**	**100,00%**	**1**
EFFICIENCY	EXECUTION PERFORMANCE	Compilation speed	EEC1	Compilation Speed	A.2	1	2	2	25,00%	0	**25,00%**	**0**				
		Resource utilization	EER1	CPU(processor type)	A.1	4	2	8	100,00%	1						
			EER2	Minimum RAM	A.2	3	2	6	75,00%	1						
			EER3	Hard disk space required	A.2	2	2	4	50,00%	1	**100,00%**	**1**				
		Software requirements	EES1	Additional software requirements	A	4	2	8	100,00%	1	**100,00%**	**1**	**66,67%**	**0**	**0,00%**	**0**

QlikView

CATEGORY	CHARACTERISTIC	SUBCHARACTERISTIC-DESC	METRIC-CODE	METRIC	M.S	VALUE	WEIGHT	COMPENSED VALUE	NORMAL.VALUE	INDICATOR_METRIC	TOTAL SUB-CHARACTERISTIC	INDICATOR_SUB-CHARAC.	TOTAL CHARACTERISTIC	INDICATOR_CHARAC	TOTAL CATEGORY	INDICATOR_CATEG.
FUNCTIONALITY	FIT TO PURPOSE	Data loading	FFI1	Direct connection to data sources	A	2	2	4	50,00%	1						
			FFI2	BigData sources	A	2	1	2	50,00%	1						
			FFI3	Apache Hadoop	A	2	1	2	50,00%	1						
			FFI4	Microsoft Access	A	2	2	4	50,00%	1						
			FFI5	Excel files	A	3	3	9	75,00%	1						
			FFI6	From an excel file, import all sheets at the same time	A	1	2	2	25,00%	0						
			FFI7	Cross-tabs	A	4	2	8	100,00%	1						
			FFI8	Plain text	A	3	3	9	75,00%	1						
			FFI9	Connecting to different data sources at the same time	A	3	2	6	75,00%	1						
			FFI10	Easy integration of many data sources	A	3	2	6	75,00%	1						
			FFI11	Visualizing data before the loading	A	1	2	2	25,00%	0						
			FFI12	Determining data format	A	3	2	6	75,00%	1						
			FFI13	Determining data type	A	4	2	8	100,00%	1						
			FFI14	Allowing column filtering before the loading	A	3	2	6	75,00%	1						
			FFI15	Allowing row filtering before the loading	A	2	2	4	50,00%	1						
			FFI16	Automatic measures creation	A	0	3	0	0,00%	0						
			FFI17	Allow renaming datasets	A	3	2	6	75,00%	1						
			FFI18	Allow renaming fields	A	3	3	9	75,00%	1						
			FFI19	Data cleansing	A	2	2	4	50,00%	1	**82,50%**	**1**				
		Data model	FFD1	Data model is done automatically	A	3	2	6	75,00%	1						
			FFD2	The done data model is the correct one	A	2	2	4	50,00%	1						
			FFD3	Data model can be visualized	A	4	3	12	100,00%	1	**100,00%**	**1**				
		Field relations	FFF1	Alerting about circular references	A	3	3	9	75,00%	1						
			FFF2	Skiping with circular references	A	3	3	9	75,00%	1						
			FFF3	A same table can be used several times	A	0	2	0	0,00%	0	**75,00%**	**1**				
		Analysis	FFA1	Creating new measures based on previous measures	A	3	3	9	75,00%	1						
			FFA2	Creating new measures based on dimensions	A	3	3	9	75,00%	1						
			FFA3	Variety of functions	A	3	3	9	75,00%	1						
			FFA4	Descriptive statistics	A	3	2	6	75,00%	1						
			FFA5	Preduction functions	A	0	2	0	0,00%	0						
			FFA6	R connection	A	3	2	6	75,00%	1						
			FFA7	Geographic information	A	2	2	4	50,00%	1						
			FFA8	Time hierarchy	A	2	3	6	50,00%	1						
			FFA9	Creating sets of data	A	1	2	2	25,00%	0						
			FFA10	Filtering data by expression	A	1	3	3	25,00%	0						
			FFA11	Filtering data by dimension	A	1	3	3	25,00%	0						
			FFA12	Visual Perspective Linking	A	4	2	8	100,00%	1						
			FFA13	No Null data specifications	A.1	0	2	0	0,00%	0						
			FFA14	Considering nulls	A	4	3	12	100,00%	1						
			FFA15	Variety of graphs	A	3	3	9	75,00%	1						
			FFA16	Modify graphs	A	4	3	12	100,00%	1	**73,33%**	**1**	**83,33%**	**1**	**66,67%**	**0**

			FFA17	No limitations to display large amounts of data	A.1	4	2	8	100,00%	1						
			FFA18	Data refresh	A	2	2	4	50,00%	1						
		Dashboards	FFD1	Dashboards Exportation	A	3	3	9	75,00%	1						
			FFD2	Templates	A	0	2	0	0,00%	0						
			FFD3	Free design	A	4	2	8	100,00%	1	**71,43%**	**1**				
		Reporting	FFR1	Reports Exportation	A	3	3	9	75,00%	1						
			FFR2	Templates	A	0	2	0	0,00%	0						
			FFR3	Free design	A	1	2	2	25,00%	0	**42,86%**	**0**				
	INTEROPERABILITY	Languages	FIL1	Languages displayed	A.1	4	2	8	100,00%	1	**100,00%**	**1**				
		Portability	FIP1	Operating Systems	A.1	0	2	0	0,00%	0						
			FIP2	SaaS/Web	A	1	1	1	25,00%	0						
			FIP3	Mobile	A	3	2	6	75,00%	1	**40,00%**	**0**				
		Using the project by third parts	FIU1	Using the project by third parts	A	3	2	6	75,00%	1	**75,00%**	**1**				
		Data exchage	FID1	Exportation in txt	A	3	2	6	75,00%	1						
			FID2	Exportation in CSV	A	3	2	6	75,00%	1						
			FID3	Exportation in HTML	A	3	2	6	75,00%	1						
			FID4	Exportation in Excel file	A	3	3	9	75,00%	1	**100,00%**	**1**	**75,00%**	**0**		
	SECURTIY	Security devices	FSS1	Password protection	A	3	3	9	75,00%	1						
			FSS2	Permissions	A	3	3	9	75,00%	1	**100,00%**	**1**	**100,00%**	**1**		
USUABILITY	EASE OF UNDERSTANDING AND LEARNING	Learning time	UEL1	Average learning time	A.2	4	3	12	100,00%	1	**100,00%**	**1**				
		Browsing facilities	UEB1	Consistency between icons in the toolbars and their actions	A	3	3	9	75,00%	1						
			UEB2	Displaying right click menus	A	3	3	9	75,00%	1	**100,00%**	**1**				
		Terminology	UET1	Ease of understanding the terminology	A	4	3	12	100,00%	1	**100,00%**	**1**				
		Help and documentation	UEH1	User guide quality	A.2	3	2	6	75,00%	1						
			UEH2	User guide adquisition	A	3	2	6	75,00%	1						
			UEH3	On-line documentation	A	3	2	6	75,00%	1	**100,00%**	**1**				
		Support training	UES1	Availability of tailor-made training courses	A	3	2	6	75,00%	1						
			UES2	Phone technical support	A	3	2	6	75,00%	1						
			UES3	On-line support	A	3	2	6	75,00%	1						
			UES4	Availability of consulting services	A	3	2	6	75,00%	1						
			UES5	Free formation	A	3	2	6	75,00%	1						
			UES6	Community	A	3	2	6	75,00%	1	**100,00%**	**1**	**100,00%**	**1**		
	GRAPHICAL INTERFACE CHARACTERISTIC	Windows and mouse interface	UGW1	Editing elements by double-clicking	A	0	2	0	0,00%	0						
			UGW2	Dragging and dropping elements	A	3	2	6	75,00%	1	**50,00%**	**1**				
		Display	UGD1	Editing the screen layout	A	3	2	6	75,00%	1	**75,00%**	**1**	**100,00%**	**1**		
	OPERABILITY	Versatility	UOV1	Automatic update	A	2	2	4	50,00%	1	**50,00%**	**1**	**100,00%**	**1**	**100,00%**	**1**
EFFICIENCY	EXECUTION PERFORMANCE	Compilation speed	EEC1	Compilation Speed	A.2	4	2	8	100,00%	1	**100,00%**	**1**				
		Resource utilization	EER1	CPU(processor type)	A.1	4	2	8	100,00%	1						
			EER2	Minimum RAM	A.2	3	2	6	75,00%	1						
			EER3	Hard disk space required	A.2	4	2	8	100,00%	1	**100,00%**	**1**				
		Software requirements	EES1	Additional software requirements	A	4	2	8	100,00%	1	**100,00%**	**1**	**100,00%**	**1**	**100,00%**	**1**

SAP Lumira

CATEGORY	CHARACTERISTIC	SUBCHARACTERISTIC-DESC	METRIC-CODE	METRIC	M.S	VALUE	WEIGHT	COMPENSED VALUE	NORMAL. VALUE	INDICATOR_METRIC	TOTAL SUB-CHARACTERISTIC	INDICATOR_SUB-CHARAC.	TOTAL CHARACTERISTIC	INDICATOR_CHARAC	TOTAL CATEGORY	INDICATOR_CATEG.
FUNCTIONALITY	FIT TO PURPOSE	Data loading	FFI1	Direct connection to data sources	A	2	2	4	50,00%	1						
			FFI2	BigData sources	A	2	1	2	50,00%	1						
			FFI3	Apache Hadoop	A	2	1	2	50,00%	1						
			FFI4	Microsoft Access	A	2	2	4	50,00%	1						
			FFI5	Excel files	A	3	3	9	75,00%	1						
			FFI6	From an excel file, import all sheets at the same time	A	3	2	6	75,00%	1						
			FFI7	Cross-tabs	A	4	2	8	100,00%	1						
			FFI8	Plain text	A	3	3	9	75,00%	1						
			FFI9	Connecting to different data sources at the same time	A	3	2	6	75,00%	1						
			FFI10	Easy integration of many data sources	A	3	2	6	75,00%	1						
			FFI11	Visualizing data before the loading	A	3	2	6	75,00%	1						
			FFI12	Determining data format	A	3	2	6	75,00%	1						
			FFI13	Determining data type	A	3	2	6	75,00%	1						
			FFI14	Allowing column filtering before the loading	A	3	2	6	75,00%	1						
			FFI15	Allowing row filtering before the loading	A	3	2	6	75,00%	1						
			FFI16	Automatic measures creation	A	3	3	9	75,00%	1						
			FFI17	Allow renaming datasets	A	3	2	6	75,00%	1						
			FFI18	Allow renaming fields	A	3	3	9	75,00%	1						
			FFI19	Data cleansing	A	3	2	6	75,00%	1	**100,00%**	**1**				
		Data model	FFD1	Data model is done automatically	A	0	2	0	0,00%	0						
			FFD2	The done data model is the correct one	A	1	2	2	25,00%	0						
			FFD3	Data model can be visualized	A	3	3	9	75,00%	1	**42,86%**	**0**				
		Field relations	FFF1	Alerting about circular references	A	4	3	12	100,00%	1						
			FFF2	Skiping with circular references	A	4	3	12	100,00%	1						
			FFF3	A same table can be used several times	A	3	2	6	75,00%	1	**100,00%**	**1**				
		Analysis	FFA1	Creating new measures based on previous measures	A	3	3	9	75,00%	1						
			FFA2	Creating new measures based on dimensions	A	3	3	9	75,00%	1						
			FFA3	Variety of functions	A	3	3	9	75,00%	1						
			FFA4	Descriptive statistics	A	3	2	6	75,00%	1						
			FFA5	Preduction functions	A	3	2	6	75,00%	1						
			FFA6	R connection	A	2	2	4	50,00%	1						
			FFA7	Geographic information	A	3	2	6	75,00%	1						
			FFA8	Time hierarchy	A	3	3	9	75,00%	1						
			FFA9	Creating sets of data	A	0	2	0	0,00%	0						
			FFA10	Filtering data by expression	A	2	3	6	50,00%	1						
			FFA11	Filtering data by dimension	A	1	3	3	25,00%	0						
			FFA12	Visual Perspective Linking	A	0	2	0	0,00%	0						
			FFA13	No Null data specifications	A.1	0	2	0	0,00%	0						
			FFA14	Considering nulls	A	4	3	12	100,00%	1						
			FFA15	Variety of graphs	A	4	3	12	100,00%	1						
			FFA16	Modify graphs	A	1	3	3	25,00%	0	**68,89%**	**1**	**83,33%**	**1**	**66,67%**	**0**

			FFA17	No limitations to display large amounts of data	A.1	0	2	0	0,00%	0						
			FFA18	Data refresh	A	2	2	4	50,00%	1						
		Dashboards	FFD1	Dashboards Exportation	A	3	3	9	75,00%	1						
			FFD2	Templates	A	0	2	0	0,00%	0						
			FFD3	Free design	A	3	2	6	75,00%	1	**71,43%**	**1**				
		Reporting	FFR1	Reports Exportation	A	3	3	9	75,00%	1						
			FFR2	Templates	A	3	2	6	75,00%	1						
			FFR3	Free design	A	3	2	6	75,00%	1	**100,00%**	**1**				
	INTEROPERABILITY	Languages	FIL1	Languages displayed	A.1	4	2	8	100,00%	1	**100,00%**	**1**				
		Portability	FIP1	Operating Systems	A.1	0	2	0	0,00%	0						
			FIP2	SaaS/Web	A	3	1	3	75,00%	1						
			FIP3	Mobile	A	1	2	2	25,00%	0	**20,00%**	**0**				
		Using the project by third parts	FIU1	Using the project by third parts	A	3	2	6	75,00%	1	**75,00%**	**1**				
		Data exchage	FID1	Exportation in txt	A	0	2	0	0,00%	0						
			FID2	Exportation in CSV	A	3	2	6	75,00%	1						
			FID3	Exportation in HTML	A	0	2	0	0,00%	0						
			FID4	Exportation in Excel file	A	3	3	9	75,00%	1	**55,56%**	**1**	**75,00%**	**0**		
	SECURTIY	Security devices	FSS1	Password protection	A	3	3	9	75,00%	1						
			FSS2	Permissions	A	3	3	9	75,00%	1	**100,00%**	**1**	**100,00%**	**1**		
USUABILITY	EASE OF UNDERSTANDING AND LEARNING	Learning time	UEL1	Average learning time	A.2	3	3	9	75,00%	1	**75,00%**	**1**				
		Browsing facilities	UEB1	Consistency between icons in the toolbars and their actions	A	4	3	12	100,00%	1						
			UEB2	Displaying right click menus	A	0	3	0	0,00%	0	**50,00%**	**1**				
		Terminology	UET1	Ease of understanding the terminology	A	4	3	12	100,00%	1	**100,00%**	**1**				
		Help and documentation	UEH1	User guide quality	A.2	3	2	6	75,00%	1						
			UEH2	User guide adquisition	A	3	2	6	75,00%	1						
			UEH3	On-line documentation	A	3	2	6	75,00%	1	**100,00%**	**1**				
		Support training	UES1	Availability of tailor-made training courses	A	0	2	0	0,00%	0						
			UES2	Phone technical support	A	3	2	6	75,00%	1						
			UES3	On-line support	A	3	2	6	75,00%	1						
			UES4	Availability of consulting services	A	0	2	0	0,00%	0						
			UES5	Free formation	A	3	2	6	75,00%	1						
			UES6	Community	A	3	2	6	75,00%	1	**66,67%**	**1**	**100,00%**	**1**		
	GRAPHICAL INTERFACE CHARACTERISTIC	Windows and mouse interface	UGW1	Editing elements by double-clicking	A	0	2	0	0,00%	0						
			UGW2	Dragging and dropping elements	A	3	2	6	75,00%	1	**50,00%**	**1**				
		Display	UGD1	Editing the screen layout	A	3	2	6	75,00%	1	**75,00%**	**1**	**100,00%**	**1**		
	OPERABILITY	Versatility	UOV1	Automatic update	A	3	2	6	75,00%	1	**75,00%**	**1**	**100,00%**	**1**	**100,00%**	**1**
EFFICIENCY	EXECUTION PERFORMANCE	Compilation speed	EEC1	Compilation Speed	A.2	3	2	6	75,00%	1	**75,00%**	**1**				
		Resource utilization	EER1	CPU(processor type)	A.1	0	2	0	0,00%	0						
			EER2	Minimum RAM	A.2	3	2	6	75,00%	1						
			EER3	Hard disk space required	A.2	1	2	2	25,00%	0	**33,33%**	**0**				
		Software requirements	EES1	Additional software requirements	A	4	2	8	100,00%	1	**100,00%**	**1**	**66,67%**	**0**	**0,00%**	**0**

Tableau

CATEGORY	CHARACTERISTIC	SUBCHARACTERISTIC-DESC	METRIC-CODE	METRIC	M.S	VALUE	WEIGHT	COMPENSED VALUE	NORMAL. VALUE	INDICATOR_METRIC	TOTAL SUB-CHARACTERISTIC	INDICATOR_SUB-CHARAC.	TOTAL CHARACTERISTIC	INDICATOR_CHARAC	TOTAL CATEGORY	INDICATOR_CATEG.
FUNCTIONALITY	FIT TO PURPOSE	Data loading	FFI1	Direct connection to data sources	A	3	2	6	75,00%	1						
			FFI2	BigData sources	A	3	1	3	75,00%	1						
			FFI3	Apache Hadoop	A	3	1	3	75,00%	1						
			FFI4	Microsoft Access	A	3	2	6	75,00%	1						
			FFI5	Excel files	A	3	3	9	75,00%	1						
			FFI6	From an excel file, import all sheets at the same time	A	3	2	6	75,00%	1						
			FFI7	Cross-tabs	A	2	2	4	50,00%	1						
			FFI8	Plain text	A	3	3	9	75,00%	1						
			FFI9	Connecting to different data sources at the same time	A	3	2	6	75,00%	1						
			FFI10	Easy integration of many data sources	A	3	2	6	75,00%	1						
			FFI11	Visualizing data before the loading	A	3	2	6	75,00%	1						
			FFI12	Determining data format	A	3	2	6	75,00%	1						
			FFI13	Determining data type	A	3	2	6	75,00%	1						
			FFI14	Allowing column filtering before the loading	A	3	2	6	75,00%	1						
			FFI15	Allowing row filtering before the loading	A	0	2	0	0,00%	0						
			FFI16	Automatic measures creation	A	3	3	9	75,00%	1						
			FFI17	Allow renaming datasets	A	3	2	6	75,00%	1						
			FFI18	Allow renaming fields	A	3	3	9	75,00%	1						
			FFI19	Data cleansing	A	0	2	0	0,00%	0	**90,00%**	**1**				
		Data model	FFD1	Data model is done automatically	A	4	2	8	100,00%	1						
			FFD2	The done data model is the correct one	A	3	2	6	75,00%	1						
			FFD3	Data model can be visualized	A	3	3	9	75,00%	1	**100,00%**	**1**				
		Field relations	FFF1	Alerting about circular references	A	3	3	9	75,00%	1						
			FFF2	Skiping with circular references	A	0	3	0	0,00%	0						
			FFF3	A same table can be used several times	A	3	2	6	75,00%	1	**62,50%**	**1**				
		Analysis	FFA1	Creating new measures based on previous measures	A	3	3	9	75,00%	1						
			FFA2	Creating new measures based on dimensions	A	3	3	9	75,00%	1						
			FFA3	Variety of functions	A	3	3	9	75,00%	1						
			FFA4	Descriptive statistics	A	3	2	6	75,00%	1						
			FFA5	Preduction functions	A	3	2	6	75,00%	1						
			FFA6	R connection	A	3	2	6	75,00%	1						
			FFA7	Geographic information	A	3	2	6	75,00%	1						
			FFA8	Time hierarchy	A	3	3	9	75,00%	1						
			FFA9	Creating sets of data	A	3	2		0,00%	0						
			FFA10	Filtering data by expression	A	3	3	9	75,00%	1						
			FFA11	Filtering data by dimension	A	3	2	6	75,00%	1						
			FFA12	Visual Perspective Linking	A	0	2	0	0,00%	0						
			FFA13	No Null data specifications	A.1	0	2	0	0,00%	0						
			FFA14	Considering nulls	A	0	3	0	0,00%	0						
			FFA15	Variety of graphs	A	3	3	9	75,00%	1						
			FFA16	Modify graphs	A	4	3	12	100,00%	1	**79,55%**	**1**	**100,00%**	**1**	**100,00%**	**1**

			FFA17	No limitations to display large amounts of data	A.1	3	2	6	75,00%	1						
			FFA18	Data refresh	A	4	2	8	100,00%	1						
		Dashboards	FFD1	Dashboards Exportation	A	3	3	9	75,00%	1						
			FFD2	Templates	A	0	2	0	0,00%	0						
			FFD3	Free design	A	3	2	6	75,00%	1	**71,43%**	**1**				
		Reporting	FFR1	Reports Exportation	A	3	3	9	75,00%	1						
			FFR2	Templates	A	0	2	0	0,00%	0						
			FFR3	Free design	A	3	2	6	75,00%	1	**71,43%**	**1**				
	INTEROPERABILITY	Languages	FIL1	Languages displayed	A.1	4	2	8	100,00%	1	**100,00%**	**1**				
		Portability	FIP1	Operating Systems	A.1	4	2	8	100,00%	1						
			FIP2	SaaS/Web	A	3	1	3	75,00%	1						
			FIP3	Mobile	A	1	2	2	25,00%	0	**60,00%**	**1**				
		Using the project by third parts	FIU1	Using the project by third parts	A	3	2	6	75,00%	1	**75,00%**	**1**				
		Data exchage	FID1	Exportation in txt	A	3	2	6	75,00%	1						
			FID2	Exportation in CSV	A	3	2	6	75,00%	1						
			FID3	Exportation in HTML	A	0	2	0	0,00%	0						
			FID4	Exportation in Excel file	A	3	3	9	75,00%	1	**77,78%**	**1**	**100,00%**	**1**		
	SECURTIY	Security devices	FSS1	Password protection	A	3	3	9	75,00%	1						
			FSS2	Permissions	A	3	3	9	75,00%	1	**100,00%**	**1**	**100,00%**	**1**		
USUABILITY	EASE OF UNDERSTANDING AND LEARNING	Learning time	UEL1	Average learning time	A.2	1	3	3	25,00%	0	**25,00%**	**0**				
		Browsing facilities	UEB1	Consistency between icons in the toolbars and their actions	A	3	3	9	75,00%	1						
			UEB2	Displaying right click menus	A	3	3	9	75,00%	1	**100,00%**	**1**				
		Terminology	UET1	Ease of understanding the terminology	A	3	3	9	75,00%	1	**75,00%**	**1**				
		Help and documentation	UEH1	User guide quality	A.2	3	2	6	75,00%	1						
			UEH2	User guide adquisition	A	3	2	6	75,00%	1						
			UEH3	On-line documentation	A	3	2	6	75,00%	1	**100,00%**	**1**				
		Support training	UES1	Availability of tailor-made training courses	A	3	2	6	75,00%	1						
			UES2	Phone technical support	A	1	2	2	25,00%	0						
			UES3	On-line support	A	3	2	6	75,00%	1						
			UES4	Availability of consulting services	A	3	2	6	75,00%	1						
			UES5	Free formation	A	4	2	8	100,00%	1						
			UES6	Community	A	3	2	6	75,00%	1	**83,33%**	**1**	**80,00%**	**1**		
	GRAPHICAL INTERFACE CHARACTERISTIC	Windows and mouse interface	UGW1	Editing elements by double-clicking	A	3	2	6	75,00%	1						
			UGW2	Dragging and dropping elements	A	3	2	6	75,00%	1	**100,00%**	**1**				
		Display	UGD1	Editing the screen layout	A	3	2	6	75,00%	1	**75,00%**	**1**	**100,00%**	**1**		
	OPERABILITY	Versatility	UOV1	Automatic update	A	2	2	4	50,00%	1	**50,00%**	**1**	**100,00%**	**1**	**100,00%**	**1**
EFFICIENCY	EXECUTION PERFORMANCE	Compilation speed	EEC1	Compilation Speed	A.2	2	2	4	50,00%	1	**50,00%**	**1**				
		Resource utilization	EER1	CPU(processor type)	A.1	4	2	8	100,00%	1						
			EER2	Minimum RAM	A.2	4	2	8	100,00%	1						
			EER3	Hard disk space required	A.2	3	2	6	75,00%	1	**100,00%**	**1**				
		Software requirements	EES1	Additional software requirements	A	4	2	8	100,00%	1	**100,00%**	**1**	**100,00%**	**1**	**100,00%**	**1**

Annex 3:QlickView evaluation

This chapter explains how does QlikView meet (or not) the metrics evaluated in this project. Metrics are grouped in sub-characteristics, and metric's codes appear in the text when they are mentioned.

Data loading:

User can extract data from files (table files, data files and web files) *(FFI8)* or can connect to databases by ODBC (Open Database Conectivity) and OLEDB (Object Linking and Embedding Database). Examples of connections are Oracle, Microsoft Access *(FFI4)* or Microsoft SQL Server. Moreover, it can also be connected to BigData sources like Teradata*(FFI2)*. QlikView has not integrated connectors (ODBC or OLEDB) to database *(FFI1)*, but QVSource, which is a Qlik's partner, offers a variety of API connectors for QlikView. These API connectors allow the connection to different social and business APIs without requiring any technical knowledge. Some examples are Twitter, Facebook, Google Analytics, Google Docs/Calendar, Mashape...Additionally, QVSource also offers developing connectors for other sources that may not have an ODBC driver, such as NoSQL type databases as MongoDB or Hadoop *(FFI3)*.

The loading data is done by the Editor Script, where user has to write a specific code in SQL-like language. There also exist the option to click on tabs and the code is written on the Editor Script by the machine, and finally, user just executes the code in order to load data.

Additionally, user can load data, easily, from spreadsheets (e.g Excel files) by an assistance (Wizard Assistance *(FFI5)*. Unfortunately, if user wants to import data from more than one sheet in the same file, he must repeat the same process as many time as there are sheets. But, if user knows SQL code, it is advisable to type code in the Editor Script instead of repeat the same browsing through menus process many times *(FFI6)*.
Usually, in excel data sources there are cross-tabs and QlikView has the option to import them from excel files. During the loading, user indicates if a table is a cross tab and he can establish the parameters of the cross tab and change their names *(FFI7)*.

On the other hand, connecting to multiple data sources is possible, just repeating the same process for each different connection *(FFI9)*. QlikView can combine data from many different data sources with high performance, regardless of how these data sources work on their own. Tables from wherever data source will be charged in the memory of QlikView as simply datasets. Therefore, the integration of many datasources become the integration of different datasets *(FFI10)*.

With the Editor Script, user can clean and prepare data for the loading. For example, user can create new calculated fields, rename fields *(FF117)*, filter data and columns *(FFI14) (FFI15)*, assign name to the dataset *(FFI18)* ... As the loading data is based on the written code, user can also insert data manually. User can do almost all these functions typing code or by menus, interchangeably,because the Editor Script also offers menus to do almost all functions. But, for example, filtering can not be done by menus and user must to type code to get this.

QlikView does not assign a data type to fields (dimension or measure), only when fields are displayed in charts, they take the names of dimensions or expressions, otherwise they are called just fields *(FFI12)*. On the other hand, the data format can be agreed, by user, before loading the data. On the top of the script there are the default settings for the data format and user can

modify them. For example, the following sentence can be written on the top of the script: **SET** *DateFormat*='DD/MM/YYYY'; It means, that every data with the following format: DD/MM/YYYY will be interpreted as a date by QlickView. Therefore, data formats can be changed by user typing the corresponding code during the loading *(FFI13)*.
Whether user loads data from files, data are showed before the loading. However, when data are loaded from data bases they are not showed *(FFI11)*. Finally, QlikView does not create automatically any measure from fields as other tools do *(FFI16)*.

With knowledge of SQL language user has more flexibility and gain more speed during the processes, but not knowing SQL language is not an impediment to use Qlikview. However, there exist an option "Syntax check" that marks the code that is not right and it can be useful to learn and improve SQL language.

Data model:

QlikView creates automatically the data model taken as reference the names of the fields. In fact, two tables are related if there exist two fields, one in each one, with same name(case sensitive) and matched values *(FFD1)*. Therefore, during the loading process is important to pay attention to field names in order to get the right model. Automatic modeling is an advantage because it saves time, but sometimes user can visualize the model and realize that it is not the desired. In that case, user has the alternative of modifying the loading script in order to get the desired model *(FFD2)*. The visualization of the data model is key to understand the relations between fields and with QlikView user can visualize the data model every time *(FFD3)*.

Field relations:

Unlikely, because of the automatic relation by name, there can appears circular references and QlikView doesn't support them. A circular loop appears when '*there are two ways to get the same field by two different tables*'. As a response of the circular reference, QlikView alerts about that *(FFF1)* and disconnects one of the tables, in fact it disconnect the biggest one, in order to display data*(FFF2)*. User can realize the disconecction, visualizing the data model. Circular references can be repaired duplicating tables, but with QlikView, it implies to load the table one time more in memory *(FFF3)*.

Analysis:

Once time data are loaded, user can create calculated fields, to display in different objects as list boxes, statistics boxes, multi boxes, table boxes and charts... User can also create new fields in the Editor Script after the loading, executing only the part of the script corresponding to the creation of the new field. These fields are considered as loaded fields. Morover, in sheet objects user can create calculated expressions and/or dimensions from loaded fields but they can only be used in the respective sheet object. It means, they are not considered fields *(FFA1) (FFA2)*.

QlikView offers a variety of functions to create new fields/expressions from all type of imported fields and they are classified in: Aggregation, Color, Conditional, Counter functions, Date and Time, Exponential and Logarithmic, Financial, Formatting, General Numeric, Inter-

record, Logical, Mapping, Mathematical constants and Parameter Free Functions, None, Null, Number interpretation, Range, Ranking, String, System and Trigonometric and Hyperbolic *(FFA3)*. With these functions a descripitive analysis can be done *(FFA4)*, but unfortunately it does not offer predictive functions *(FFA5)*. Anyway, QlikView can be connected to R project, which is an open source programming language and software environment for statistical computing and graphics. R has its own language and for this reason user who wants to use its functions must know it. The integration of R in QlikView is not very popular yet, and for this reason, there is not much information on the net and either on the website of QlikView *(FFA6)*.

QlikView is famous because of its Visual Perspective Linking. When a value or several values (in a field) are selected, QlikView makes a split second association showing only values (in other fields) associated with the current selection. Simultaneously, sheet objects (holding one or several general expressions), are calculated to show the result of the current selection. For example, there exist interaction between charts when user select some values in a chart automatically another chart will only show values associated with the selection. This fact eases discovery relations between fields and it is key in data discovery science *(FFA12)*.

On the other hand, creating new data sets is useful to analyze directly particular samples in the same workbook, but QlikView has not the option to do create them. Similarly, QlikView analizes particular datasets using its visual perspective linking *(FFA9)*. Due to the same reason, QlikView have not got filters. User filters data using the interaction between sheet objects *(FFA10)(FFA11)*. Moreover, QlikView also offers the option to lock sheet objects in order to not being modified due the interaction.

QlikView offers a corporative complement, GeoQlik, which is a GIS component for Geo-Business Intelligence within QlikView. It offers normalized GIS formats as ShapeFile, PostGIS, Oracle Spatial, Oracle Locator, Esri spatial databases, virtual globes Google Maps, OpenStreetMap, all kinds of geometries, rasters and Web services and also .csv files with the coordinates. It gives much power to QlikView in Geo-Business Intelligence, but it is a component and it is not integrated in QlikView versions *(FFA7)*.

On the other hand, user can create expressions or fields based on time functions. For example, user can use the year function as Year(sinister_date), when sinisterdate has the corresponding date format. Some tools create some of the fields Year, Month and Quarter of a date, automatically based on date. But it is not the case of QlikView, in which user must create them by himself *(FFA8)*.

QlikView distinguish between nulls and empty spaces. When data comes from a database and there are nulls, they traspass automatically to QlikView as nulls. But when data comes from files, white spaces are considered missing values and not nulls. Calculation are made although some operands or function parameters are null or missing values. In charts, missing values are considered as other values, while null values are special and user can decide to show them in a chart or not. User can transform missing values to null values by functions in QlikView Editor *(FFA14)*. Then, the only requisite to treat null values is that when they are coming from files, they must be a white space *(FFA13)*.

In order to display data, QlikView offers a variety of objects, they are: List box, Statistic box, Multibox, Table box, Chart, Input Box, Current Selections Box, Button, Text Objects, Line/Arrow objects, Slider/Calendar objects, Bookmark object, Search object, Container and Custom object. And particularly, the charts are: Bar chart, Line chart, Radar chart, Gauge chart, Mekko chart, Scatter chart, Grid chart, Pie chart, Funnel chart and Block chart. It is not the tool which offers more distinc charts, but it has a good selection *(FFA15)*. Charts offer a variety of customizing settings (Dimension limits, Sort, Style, Presentation, Axes, Colors, Number format,

Font, Layout and Caption) and user can modify them whenever he wants. It is the tool of the evaluated which offers more flexibility in the design of charts *(FFA16)*.

Although, it is not the tool with a more variety of different graphs, thanks to the graphs settings, user can get similar graphs to graphs done with other tools. For example, QlikView does not offer a Heat map but it offers a Block Chart. The difference between them, is that in a Heat map two expressions can be displayed in addition to a dimension. While with Block chart only one expression and one dimension can be displayed. Heat maps can relates the color of the blocks on a expression and the size of the blocks to another expression. By this way, two expressions can be showed in a block chart, and user can realize if there exist any relation between them or not. Basically he can visualize if there exist any pattern related with both expressions. The following image is an example of a heat map where blocks represents the values of the field Region_sinister, the size of blocks is based on the population of each region, and the color of the blocks is based on the number of sinisters happened there.

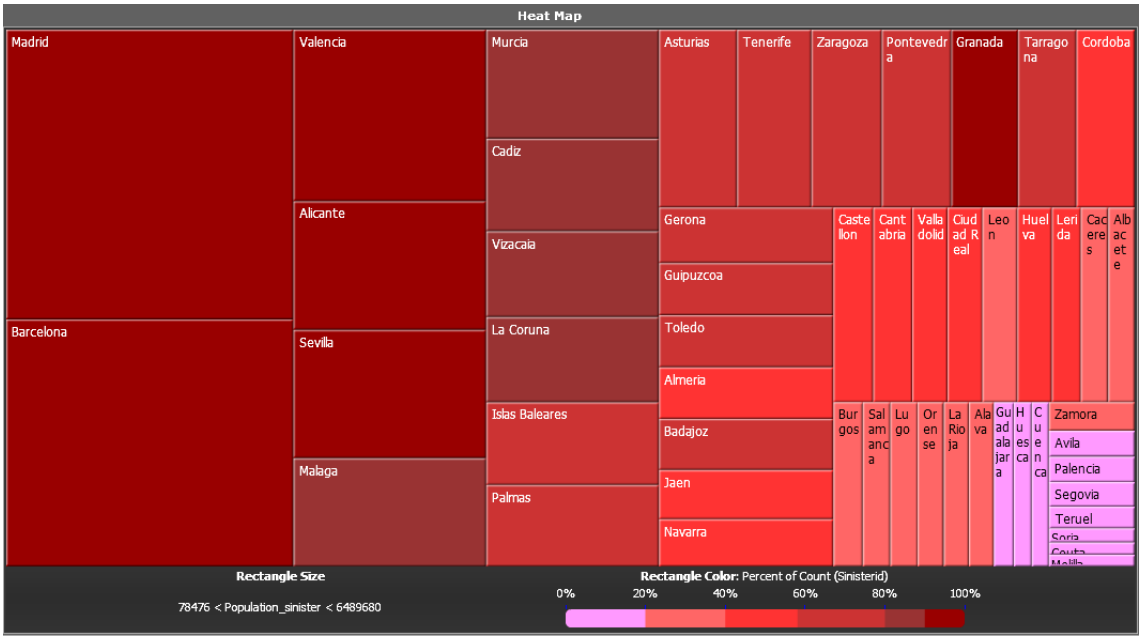


Fig. 25 Heat Map chart built in MicroStrategy

With QlikView's block chart, color cannot be represented by an expression. Only the size can represent an expression and in this particular example the expression is the population:

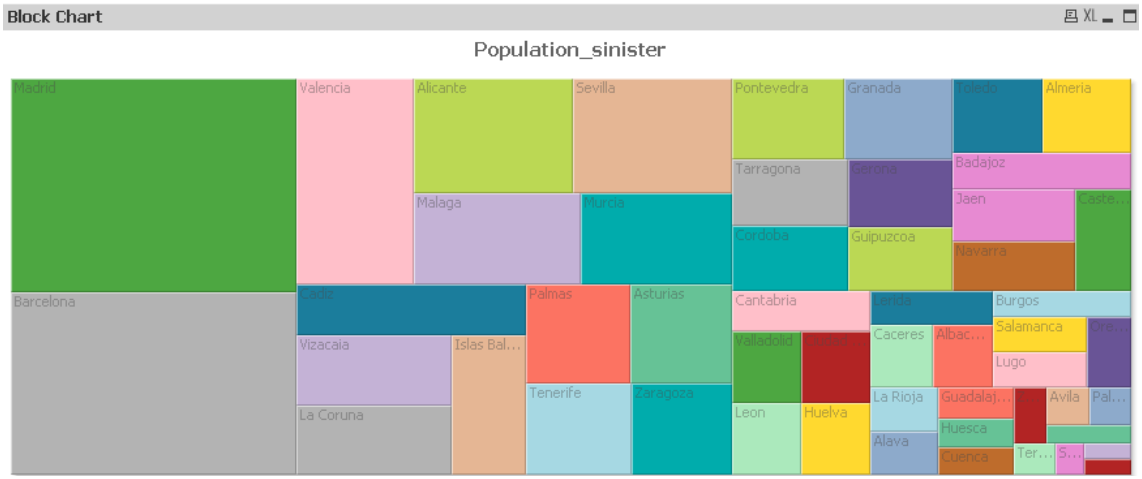


Fig. 26 Block chart built in QlikView

Even so, there exist an alternative to relate color with expressions. Each built expression in a chart has a backgroud color and user can set it to fix a color pattern for the values of the expression; by default this expression is empty. It requires type code and in my opinion it is difficult because there are not many information on the userguide about that. However, the option to set the background color is very useful and interesting, although implementing that can be not trivial. The following function is an exemple of how user can set the color of an expression. In particular, colors are set depending on the fractiles 0.2, 0.40, 0.60, 0.80 and 0.90 of the amount of sinisters in each region. It corresponds to the Background color for the population expression, which is the expression visualized in the chart.

For example:

```
If([NumericCount (SinisterID)]<=1574.8, rgb(255,204, 204),
If([NumericCount (SinisterID)]>=1574.8 and [NumericCount (SinisterID)]<2082.4,
rgb(255,153,152),
If([NumericCount (SinisterID)]>=2082.4 and [NumericCount (SinisterID)]<2630,
rgb(255,102,102),
If([NumericCount (SinisterID)]>=2630 and [NumericCount (SinisterID)]<3532.6,
rgb(255,51,51),
If([NumericCount (SinisterID)]>=3532.6 and [NumericCount (SinisterID)]<7134.4,
rgb(255,0,0),
If([NumericCount (SinisterID)]>=7134.4, rgb(204,0,0)
))))))
```

And the result is:

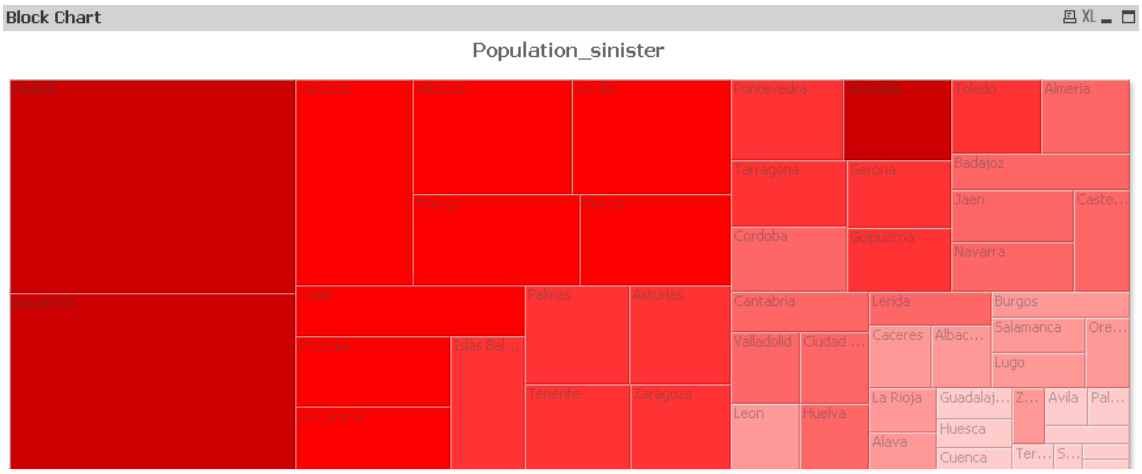


Fig. 27 Block Chart with background color assigned to an expression built in QlikView

In this chart, the size is based on the population and the color is based on the amount of sinisters. But as the amount of sinisters is not an expression of the chart there is not a legend to relate the color with the amount of sinisters.

Dynamic Updates allows adding new or modify existing data in the in-memory data model of a QlikView application without re-running loading script. These dynamic updates must be written as macros in the Edit Module. There is no any bottom to do that automatically. If there is not many data and the update is punctual, user can reload the script from the Editor Script In order to get updated charts. But, if there are many data and the updates are usual is recommended to write a macro and the update will be automatic *(FFA18)*.

Finally, QlikView Desktop has not present any problem displaying big amount of data, with the 20141220_Initial_test database *(FFA17)*.

Dashboards:

QlikView offers plain freedom to design dashboards. In QlikView, dashboards are composed by sheets. QlikView offers sheet objects of type: List box, Statistic box, Multibox, Table box, Chart, Input Box, Current Selections Box, Button, Text Objects, Line/Arrow objects, Slider/Calendar objects, Bookmark object, Searchobject, Container and Custom object. User can also change the background color of the sheet *(FFD3)*. There are many examples in the net and user can take them as reference, but there are not templates included in QlikView. Althought, user can save settings for particular charts in order to use its settings, more times *(FFD2)*.

User can save sheets, individually, in PDF file and print them. Moreover, images can also be copied to clipboard or saved. On the other hand, user can export sheets directly to email, word file and Excel files… using the macros in the Editor Module. Anyway, creating macros is not an easy process. The whole Project cannot be shared by the Personal Edition but they can be shared using the Server edition *(FFD1)*.

Reporting:

There exist the option to create a static report from a dashboard, adding object sheets, images and text, but the options to design reports are poor comparing with other tools. There exist a corporative complement called NPrinting that is a report generation and it seems to have more options in the report designing. But as, QlikView offers options to do reports (although they are

poor), the integrated reporting is the evaluated feature *(FFR3)*. Reports can be printed or saved as XPS file *(FFR1)*. And there are not templates to do that *(FFR2)*.

Languages:

The interface language can change between: Chinese, Dutch, English, French, German, Italian, Japanese, Korean, Polish, Portuguese, Russian, Spanish, Swedish and Turkish depending on the user demands. Moreover, it offers the option to use a different language for the Help assistance *(FIL1)*.

Portability:

QlikView Personal Edition and QlikView Server are compatible with Windows XP, Windows Vista, Windows Server 2003 and Windows Server 2008, for 32-bit and 64-bit. There is not the option to install it in Mac *(FIP1)*. QlikView does not directly provide a SaaS service offering. However many QlikView partners offer QlikView-built applications as a SaaS offering to their own customers *(FIP2)*. QlikView on Mobile, is provided free as part of QlikView Server *(FIP3)*.

Using he Project by third parts:

With the QlikView Personal Edition, user can only open files that he created himself. On the other hand, with the licenced Server edition it does not occur. In fact, documents created by QlikView Server can be shared with other users and they can, even, open projects created by Personal Editions *(FIU1)*. Moreover, they can be published and accessed in mobile platforms, by users with permissions.

Data exchange:

Data from graphs can be copied to clipboard and pasted in several type of files, for example: Comma Delimited (.*csv, *.txt), Semicolon Delimited(*.skv, *.txt), Tab delimited (*.tab, *.txt) *(FID1)(FID2)*, Hypertext (*.html, *.htm) *(FID3)*, XML (*.xml), Excel (*.xls) *(FID4)*. There is a direct option to export data directly in an Excel sheet.

Security devices:

With QlikView Personal Edition there is not the possibility to assign security devices to projects or data. However, with the Server Edition assigning permissions and passwords to users is posible by the Editor Script *(FSS1) (FSS2)*. QlikView Personal Edition does not offer the option to assign permissions because it is a personal edition and documents are not shared to other users. Moreover there not exist the possibilty to require a pasword.

Learning time:

In few minutes user is able to create a chart following the user guide. Its intuitive interface let the user to fastly adapt to it. Being able to use some of the advanced features only comes with time and experience. In my opinion, it is one of the tools with a fast learning *(UEL1)*.

Browsing facilites:

QlikView has many icons in toolbars, but they are intuitive *(UEB1)*. Right menu can be displayed when the cursor is on the sheet or on a sheet object. Then, a menu appears in order to change properties from the sheet and the sheet object, respectively *(UEB2)*.

Terminology:

QlikView has an easy terminology. For example, all fields are considered as fields except when user is building a sheet object. For example, if user is building a bar chart, one of the fields takes the name of dimension and the other field is called expression. The terminology is understable for everybody and it does not require depth knowledge in Data Analysis*(UET1)*.

Help and documentation:

Userguide, which is for free in the web *(UEH2)*, is very detailed for a basic learning. Advanced topics must be searched in the net *(UEH3)*, for example in QlikView's community, where users can ask questions and an expert user will answer them *(UEH1)*. Moreover, there is a option in QlikView web, where user can ask questions to a group of experts in the nearest sales office and it is free.

Support training:

It is important to remark that QlikView personal edition does not have any type of support offered by QlikView. Servers users has the option to assist to courses delivered in-person at Public classrooms or privately where an instructor-led offers support *(UES1)*. Moreover, phone technical support is avaible for Server users from Monday to Friday and between 8:00a.m and 5:00pm *(UES2)*. Online support is also offered by server users asking the nearest sales office *(UES3)*. Additionally, QlikView also offers consulting services classified in three categories: Implementation Services , Application Services and Business Services *(UES4)*. Moreover, QlikView proposes some online free courses and also free tutorials and papers of particular features for every type of users *(UES5)*. Finally, there is also a Community where every users can ask for questions or look for questions already answered y experts *(UES6)*.

Windows and mouse interface:

User cannot edit elements by double-clicking *(UGW1)*, but dragging and dropping method can be used in QlikView. For example user can drag fields to the dimension layout and then drop them when he is building a chart *(UGW2)*.

Display:

The screen layout can be editted, dropping or adding icons in toolbars *(UGD1)*.

Versatility:

Version updated must be done manually from the web of QlikView, login with a client user and a password *(UOV1)*.

Compilation speed:

By our experience with QlikView Desktop, it has a fast compilation speed. QlikView differs from other tools in its compilation speed, additionallt to the *Visual Perspective Linking (EEC1)*.

Resources utilization:

For Qlikview Personal Editon it is recommended to use Intel Core duo or higher for the 32-bit version, and Intel Core 2 Duo or higer for 64-bit version. But it is not a requisite. Fort he 32-bit versions the minimim RAM is 1GB, but depending on the volum of data more memory can be required. Fort he 64-bit version, the reccomended is 2 GB. The Hard Disk space required is 200MB for the 32 bit and 250MB for the 64-bit versión. Fort he QlikView Server Edition it is recommended to use Intel Core Duo ir higher for the 32-bit and Multi-core for the 64-bit. For the 32-bit, 1GB of RAM memory is required (much memory can be required depending on the data volum), while for the 64-bit the mínimum of RAM memory is 4B *(EER1) (EER2) (EER3)*.

Software requirements:

QlikView Server requires a browser, and it can be Internet Explorer 7 or Firefox 3 *(EES1)*.

Annex 4: SAP Lumira evaluation

This chapter explains how does SAP Lumira meet (or not) the metrics evaluated in this project. Metrics are grouped in sub-characteristics, and metric's codes appear in the text when they are mentioned.

Data loading:

With SAP Lumira, user can load an Excel worksheet *(FFI5)* and a text file (*.csv, *.txt, *.log, *.pm, *.tsv) *(FFI8)* as a dataset and can also copy data from clipboard. As a part of the enormous platform SAP, Lumira can obviously connect directly to SAP HANA and can also load data downloaded by SAP HANA without requiring a connection. SAP Lumira can be also connected to SAP Business Objects Universe and to SAP Business Warehouse to load datasets. Finally, it also can run freehand SQL on a database to load datasets. Some examples are Microsoft Access *(FFI4)*, Apache Hive *(FFI3)*, Cloudera, IBM DB2, IBM Netezza, Microsoft SQL Server, Oracle, Sybase, Mongodb, Teradata… Then, SAP Lumira can be connected to many different types of Big Data sources *(FFI2)*. In order to connect a database to SAP Lumira, the requirement is to be provided by a JDBC connector to access to data from foreign applications because the connectors are not integrated in the application*(FFI1)*. As the way to download data is by SQL code, user can create a query with its requisites for example loading only columns that he wants or adding filters... it means user can clean data before loading them. When user is loading data from Excel files, there is the option to append all sheets and then all worksheets in the workbook are added to the dataset, appending the common columns and adding as new columns the different ones*(FFI6).* With excel files user can also choose which columns and which rows he wants to import. There is also the option to load cross tables, just selecting the option *(FFI7)*.

The connection to many data sources at same time is possible and easy, just repeating the same process to acquire for each different database *(FFI9)*. The integration works well, because a graph must come from a unique dataset. Then, datasets are always separated and user just have to select in which dataset he wants to work. Moreover, user can also merge datasets from different data sources, and work with the resulting dataset. Then, the integration of many data sources is good thanks to the merge option *(FFI10)*.

Before the data loading, the application displays a preview of data and user can filter data and columns *(FFI11) (FFI14) (FFI15)*. Fields can not be renamed (they can be renamed after the loading) *(FFI18)* althouht datasets can be renamed *(FFI17)*.

User cannot know the data type (dimension, measure) , assigned to fields by Lumira, until after data is loaded, and then he can change them *(FFI13)*. And in order to prevent specified columns from being proposed as Measures when data is acquired, the applications uses the enrichment_suggestions.<versionsnumber>.txt file to define which columns should not be proposed as Measures. User can also prevent objects from being considered as Time objects or Geographical objects. And the enrichment will be processed if in the Preferences settings, user choose automatic detection of enrichment. SAP Lumira creates automatically aggregated measures from every numeric field *(FFI16)*.

After the data loading user can view and prepare data, editing and cleaning it, converting data to another data format (integer, biginteger ,double, string, date, Boolean) *(FFI12)*, duplicate columns, convert cases, replace values, fill string values with a prefix or suffix, creating geography and time hierarchies, creating a measure from a column or a dimension, adding a dataset, merging (joining) datasets...This data cleansed is easy to do because it is done by menus *(FFI19).*

Data model:

The data model is not done automatically like other applications *(FFD1)*, in fact there not exists a data model. Lumira offers an alternative option called Combine and it let user to merge datasets combining data from two datasets using JOIN. The columns for the second dataset are matched based on compatibility with a key column. The matched columns are proposed with the probability of the match. Combine option let user to use a particular dataset in several merges with different merging fields and loading it in memory just one time. By this way, a dataset can be connected to two different other datasets, avoiding circular references because user impose what are the pair of dataset merged and by which field (primary key). It is possible to build inner joins or left outer joins, user can decide it. To build a merge, the foreign key must be a primary key of the right table.. Moreover, some datasets don't have the foreign key a a primary key, and then it is not possible to merge it, therefore it is not possible to relate this table with other ones *(FFD2)*. User can always visualize the loaded datasets, and therefore the merged datasets to know the relations between fields. If the data model is complex, append data by this way can be a laborious task.On the other hand, It can be a good alternative for users who are not familiarized with data models and prefer a big table with all the columns, but in business intelligence area data models application is well extended. Additionally, a same table can be joined more than one time with different relations *(FFD3)*.

Field relations:

Because of the joins are done manually, and user can determine which is the foreign key, circular references does not exist *(FFF1) (FFF2) ((FFF1).*

Analysis:

User can create new measures to enrich the dataset at any time either directly from a measure or dimension or using the formula language to create a calculated measure *(FFA1) (FFA2)*. When using Connection to SAP HANA data source, it is not possible to create a measure from a numeric or string dimension. Measures need to be created in the SAP HANA View, before being acquired automatically in the application.

By default, all numeric dimensions are also created automatically as aggregated measures. The Aggregated measures are the typical ones: SUM, MIN, MAX, Count(Distinct), Count (All), Average. If user wants to transform a dimension to a mesure without aggregation, then he must select NONE in the menu of Aggregations.

Moreover, user can create calculated measures and calculated dimensions by the formula editor script. There, two fields can be combined to create a new one. User can apply functions from a predefined set of numeric, date and text functions. Using also If...Then...Else clauses, called logical functions and a calendar picker for date parameters. SAP Lumira is not the application with more functions, but they are enough to do descriptive statistics *(FFA3) (FFA4)*.

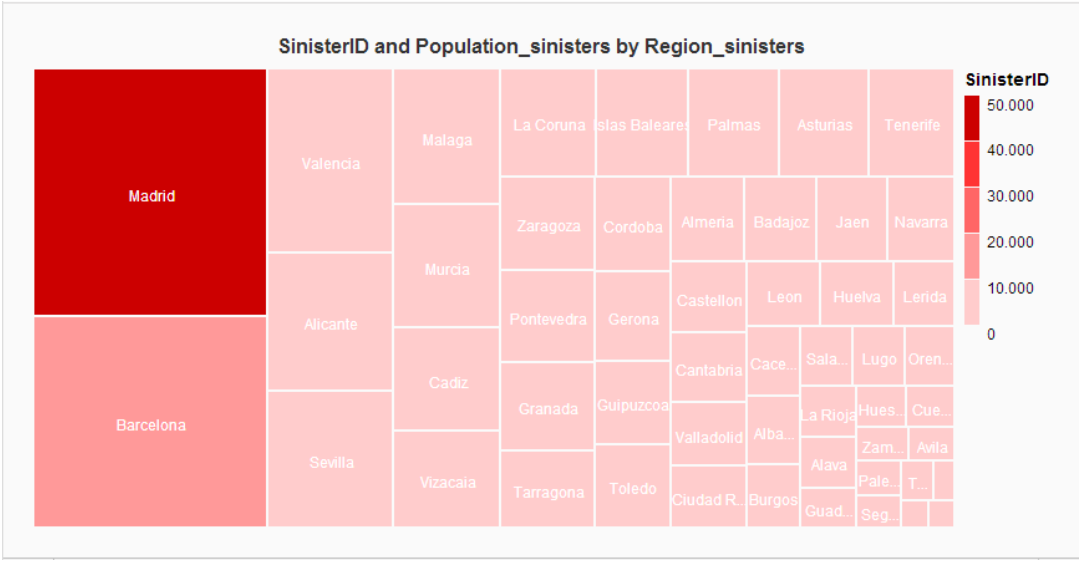
SAP Lumira offers also predictive calculations. By choosing the down arrow on a measure displayed with a date range user is able to choose a Forecast or Linear Regression Predictive Calculation type to add to the visualization along with specifying how many periods forward user wanted to predict *(FFA5)*. Moreover, there is also another snap-in product for SAP Lumira called SAP Predictive Analysis which can adds predictive features into SAP Lumira. SAP Predictive Analysis is far more robust with regards to predictive features than the offered in SAP Lumira and moreover SAP Predictive Analysis supports the use of predictive algorithms from open source R, unlike SAP Lumira *(FFA6)*.

SAP Lumira offers a a plethora of data visualization types. Moreover, in the user guide they are very well classified depending on the type of analysis the user want to do. It is one of the keys of SAP Lumira, the variety of graphs *(FFA15)*.

- Comparison: Column Chart, Column Chart with 2 Y-Axes, 3D Column Chart, Radar Chart, Area Chart, Tag Cloud, Heat Map, Table.
- Percentage: Pie chart, Donut chart, Pie with depth chart, Stacked Column Chart, Tree, Funnel Chart.
- Correlation: Scatter plot, Scatter Matrix Chart, Bubble Chart, Network Chart, Numeric Point, Tree.
- Trend: Line chart, Line Chart with 2 Y-Axes, Combined Line chart, Combined Line chart with 2 Y-Axes, Waterfall Chart, Box Plot, Parallel Coordinates Chart.
- Geographical: Geo Bubble Chart, Geo Choropleth Chart, Geo Pie Chart, Geo Map. But, to create a Geo Map, user must have an Esri ARcGis Online Account.

There exist Trellis to display a particular chart for each value of an additional dimension. For example, if user creates a bar chart that compares revenue by region, and then he adds country to the trellis, multiple charts will appear. Each cart will display the revenue by region for one country. In other tools is also possible but is not as comfortable to do.

User can always change the color of a legend, but when the measure showed in the legend is continuous, there is no the possibility to change the rank limits for each color. Then, although there is much variety between data, user can not realize it. In fact, the data rank is always divided in 5 equal portions and for each portion a color is assigned. User can change this color but not the rank limits. An example is the following heat map, in where it is not possible to realize about a pattern, which in other tools can be visualized:



It can be considered an important con when user is doing an analysis.

SAP Lumira is one of the tools with more distinct graphs to display data. But they can not be very customized by user *(FFA16)*.

There exist an icon to refresh data. If there is any change in the data source, user can refreshes data by it, because it is not done automatically *(FFA18)* .

User can create a time hierarchy (Year level, Quarter level, Month level and Day level) from a date field just in one click. Also geographical and personalized hierarchies can be built in order to be ploted in maps *(FFA7) (FFA8).*

User can filter data in a visualization filtering by any dimension *(FFA11)*, selecting data points in a chart to filter or exclude them, and displaying only the top or the bottom ranking data points for dimension or measure. But, for example, there is no the possibility to filter by a specific requirement of a measure by a formula or for a specific value of a dimension *(FFA10)*. Other tools offer much better filtering capabilities. User can interact with a chart filtering data by the legend or selecting data points. But a filter done in a chart does not interact with other charts. It means, the interactions is in a chart and not between charts *(FFA12)*. And moreover, there is not exist the possibility to create data sets *(FFA9)*.

Null data are considered, by default, as a one more value of the field. Null values does not intervene in calculated measures. User can choose if the null value is represented in a visualization or not *(FFA14)*. There is not any problem with null data. When data is loaded from a file, null data must be represented as an empty space *(FFA13)*.

Moreover, usually SAP Lumira shows the advice: “There are too many data points to visualize (maximum is 10.000) Filter the data to reduce the number of data points”. It is a big con when user is displaying many data. Although, it is a default parameter and user can increase it, also considering increasing the virtual memory allocated to SAP Lumira. However, there exists a option to solve the problem, but in my opinion the default maximum number of data points is much low for this type of tool. It totally pales in comparison to Tableau that can render 60 million points *(FFA17)*.

Dashboards:

Dashboards can be exported and share with other users *(FFD1)*. There are not templates integrated in the application *(FFD2)*. Moreover, user can design himself dashboards with totally freedom*(FFD3)*.

Reporting:

Reports can be exported to SAP Lumira Cloud, SAP Lumira Server and SAP Business Objects BI platform. Moreover, they cam be exported to pdf file *(FFR1)*. There are templates that fix only the distribution of the charts but user has totally free to customize them *(FFR2)*. There are different options to report the results. Boards pages where user can add interactive charts, filter and input controls, Infographic where user can add chart visual properties, pictograms, shapes, images and text and Report pages where user can add interactive charts, sections and input controls. Input controls, let user filter data in a comfortable way. When user is building the report, and he has choose a board or a report page, he can select a dimension, and in the board there will appear a input control with all the values of the selected dimension and user can filter by them *(FFR3)*.

Languages:

SAP Lumira is displayed in several languages: Deutsch, English, Spanish, French, Hungarian, Polish, Portuguese, Japanese and simplified Chinese *(FFL1)*.

Portability:

SAP Lumira only works in Windows 7/8 with 64 or 32-bit, Windows Server 2008 with 64 bit and Windows 2012 operative systems *(FIP1)*. SAP Lumira Cloud is the cloud computing SaaS of SAP and it is free. But SAP Lumira Cloud has some limitations, for example, predictive charts are not supported *(FIP2)*. SAP Lumira Cloud can be also accessed from any mobile device supporting HTML5, but there is not any particular mobile application for SAP Lumira. There exist the free mobile application SAP Business Object Explorer connected to the SAP Business Object Explorer that can visualize documents built in firstly in SAP Lumira and then published in SAP Business Object Explorer *(FIP2)*.

Data exchange and Using the project by third parts:

User can share data, charts and the full project. Datasets can be exported to .csv or Microsoft Excel file and published to SAP HANA *(FID1) (FID4)*. Data cannot be exported in .txt *(FID2)* files or HTML files *(FID4)*. Charts can be shared via email and printing or saving it in PDF, but they can't be saved in Excel files. Moreover, user can also publish projects, including data and charts to an SAP StreamWork activity to share with the community, also publishing in SAP Lumira Cloud, SAP Lumira Server to share with colleagues and publishing in SAP business Objects Intelligence platform.

Security devices:

In SAP Lumira Desktop Standard Edition password is not requested to open a locally saved document (Note: In SAP Lumira Desktop Standard Edition only the creator of a document can open it again). If the document is in SAP Lumira Cloud, a user name and a password is required. By the way, it would be useful the possibility to require a password in the Desktop Standard Edition. Moreover, with SAP Lumira Server, an administrator can assign passwords *(FSS1)* to projects and permissions to users *(FSS2)*.

Ease of understanding and learning:

It is the second tool evaluated in this thesis, which requires a shorter average learning time(UEL1). Addtionally, SAP Lumira is very intuitive and its icons and tabs are very consistency an let user easily learn the management. It is the tool with the easiest interface *(UEB1)*. Right cliclking does not display any menu *(UEB2)*.

Its terminology is agree with the BI terminology and it is not complicated *(UET1)*.

There are many tutorials of different topics in web of SAP Lumira, not only a user guide. There are tutorials of more advanced actions. All are understandable and actions are explained step by step. There are also video tutorials in the web *(UES1) (UES2)(UES3)*.

In the web of SAP Lumira, there is no information about tailor-made training-courses in the web. SAP Lumira offers a phone technical support but its schedule is not showed. But, user can also ask questions in the web to an expertise. This option is available at full time *(UES1) (UES2) (UES3)*.
SAP Lumira also offers in its web consulting solutions, classified in Sales Management with Sales Team Performance and Sales Quota & Comission, Marketing with Campaign Analysis and Segment Analysis, Financial Plannig with Financial Statement Analysis and Corporate Project Monitoring and finally Human Resources with SuccessFAcots Compensation Analysis *(UES4)*.

It offers free webinars and free video and interactive tutorials in the web of SAP Lumira as free formation. Morover, user can register to events by free *(UES5)*. And there is also a coumminity *(UES6)*.

Graphical interface:

User cannot edit elements by double-clicking *(UGW1)*, but dragging and dropping method can be used *(UGW2)*. Screen layout can be modified by the user *(UGD1)*.

Operability:

SAP Lumira can check automatically dayly, weekly, monthly for updates connecting by itself to SAP Public Portal or SAP Pulic Support *(UOV1)*.

Execution performance:

Its compliation speed can be considered fast. Server Edition requires 3.7 GB of disk space *(EER3)* and 4 GB of RAM *(EER2)* .And it can only be installed in x64 bit processor *(EER1)*.

Annex 5: MicroStrategy Analytics evaluation

Data loading:

Data can be imported from a file (*.xls, *.xlsx, *.txt , *.csv) *(FFI8) (FFI5)* of 200MB as maximum, from a database or using a database query. For *.xls and *.xlsx files, multiple worksheets can be included in the file, but only one worksheet can be uploaded at a time *(FFI6)*. And crosstabs can be also uploaded but they must be in a particular format *(FFI7)*.

Moreover, data can be also loaded by a connection to databases. In order to load data from a database the connection is done by ODBC. Some data sources can be directly connected to Analyitcs because it already has several particular ODBC integrated (FFI1). Some examples are Amazon EMR Cloud, Apache Hadoop *(FFI3)*, Cloudera CDH, Cloudera Impala, Greenplum, Hortonworks HDP, IBM DB2, IBM Netezza, Microsoft Acces, Microsoft SQL Database, Microsoft SQL Server, Oracle, PostfreSQL, Web services data sources…*(FFI4)* Other data sources can be connected but an ODBC previously installed is needed to establish the connection between SAP Lumira and the database. Some examples are Mongo, My sql Community and Enterprise, SAP CDBMS, SAP HANA, Teradata , Aster Google BigQuery, HP Vertica,…

An intuitive visual interface makes easy to import data by dragging and dropping tables, selecting columns *(FFI14)*, rename fields and datasets *(FFI17) (FFI18)* and specifying filter conditions *(FFI15)*. Moreover, data is visualized before being loaded *(FFI11)*. By default, when user selects data to import, MicroStrategy automatically generates the SQL query that is required to select the data from the database. Alternatively, there is the option to load data by Freeform script. With Freeform script user writes his owns database queries to retrieve data from a relational database. For example, user can load data from a database using SQL, from third-party web services using XQuery, from Hadoop using HiveQL… always using a previous connection to the data source. User can then customize how the data is imported by changing the SQL query displayed in the Editor panel. User can use joins, expressions, aggregations, and filters to define the data that he wants to load. Therefore, Freeform script loading, let user to do a data cleansing before load data *(FFI19)*. Moreover, user can also import a dashboard and data that an Analytics Desktop or MicroStrategy Analytics Express user has shared with him.

During the loading process, user can define a data column type as attribute or metric but only during the importing process because in the dashboard user cannot modify it *(FFI13)*. When it is defined as attribute, user can decide to assign a geo role to the attribute. Moreover, user can also choose the data format between: number, text, date, date time, big decimal, email, html tag, phone number, symbol and url *(FFI12)*.

If the column's data type is Date, Time, or DateTime, Analytics automatically generates additional time-related information based on the contents of the data column. For example, if the column is assigned the Date data type, Analytics Desktop can automatically generate separate attributes for year and month information just with a click. User can also assign geo-role or shape key to the data column to enable data to be displayed on a map-based visualization. When a data column has been assigned a geo role, Analytics can also

automatically generate separated attributes containing higher level of geographical data. If the data column contains city data (that MicroStrategy Analytics recognize), the tool automatically generates the State attribute, which contains the state each city is located in. Moreover, for each dataset, the measure Row_Count is automatically created. It gives information about how many rows the dataset has. During the loading user can also choose not to import a column of data and rename columns while data is visualized. In contrast, user cannot filter the rows. Finally, user loads the datasets renaming it *(FFI16)*.

Analytics Desktop allows user to combine data from different sources in order to analyze relations between them *(FFI9)*. The integration of different data sources is easy because fields are grouped in datasources, and user can distinguish them *(FFI10)*.

Data Model:

MicroStrategy gives users a powerful option, to model their data, or not to model. By default, when user imports a new dataset directly into a dashboard that contains at least one dataset, the new dataset is automatically linked to attributes that already exist in the dashboard. MicroStrategy attempts to link attributes that share the same name. User can also manually link attributes that are shared across multiple existing datasets *(FFD1)* in order to get the desired data model *(FFD2)*. An attribute that is linked across multiple datasets is displayed with a link icon and is displayed as one attribute when added to a visualization. MicroStrategy Anlytics does not offer the option to visualize the data model, and because of that this icons are very important*(FFD3)*. User can unlink attributes that are already linked. Unlinked attributes with the same name are treated as two separate attributes when displayed in a visualization. Manually linking attributes allows user to link attributes across multiple existing datasets. The attributes that user link must be the same data type. User can link an attribute to attributes in one or more datasets.

Circular references:

Although the link is done manually, there can be circular references and MicroStrategy does not advice about them *(FFF1)*. In fact, it works but results are not the correct ones *(FFF2)*. The unique way to solve circular references are to load one time more the same table and link manually, or renaming the fields. It is because Analytics does not let to use the same table with different connections but by the same field *(FFF3)*. The unique way to realize that there are circular references is knowing well the data, because MicroStrategy Anlytics does not offer the option to visualize the data model.

Analysis:

During the data analysis, user can create new metrics (called derived metrics) based on attributes and metrics that have already been loaded to a dashboard *(FFA1)*. A derived metric performs a calculation on-the-fly with the data available on dashboard, without re-executing the dashboard against the data source. Derived metrics are saved and displayed only in the specific

dashboard in which they are created. MicroStrategy Analytics offers a plethora of functions to create new metrics.

- User can create easily a new metric based on an arithmetic calculation (+,-,x,÷) from two metrics already in the dataset. -User can also create a new metric by aggregation functions as create a new metric to calculate a running total, moving total, assigning a numeric rank to each value in a metric, displaying metric values as percentages of a cumulative total, combining the values of two or more metrics...
- User can also create a metric based on a dimension *(FFA2)* with the functions: Average, Count, Maximum, Minimum, Standard Deviation, Sum and Variance. If the attribute is not numeric it seems to don't have sense. But if user wants to use a dimension to create a new measure, for example: He wants to create a new metric with value 1 if gender is "Male" and 0 if the gender is "Female" , he cannot do it directly, because MicroStrategy formulas only support metrics as arguments. Then, the solution is to create firstly a metric based on Gender by Average, Maximum, Minimum, Standard Deviation, Sum and Variance. These functions have only transform from attribute to metric, but the resulting values are the same: "Male" and "Female". Once time, the new metric is created with exactly same values than the attribute, user can use it in logical formulas to create other new metric.
- User can also create a new derived metric based on a MicroStrategy function. The categories of functions offered are: Basi functions, Data Mining functions, Date and Time, Financial, Internal, Math, Null/Zero, OLAP, Rank and NTile, Statistical and String.
- User can create a derived metric from scratch and use conditional calculations provide conditional analysis by combining data into different groups based on the value of one or more metrics in a dashboard.

Once time a derived metric is created it cannot be edited. If user wants to edit it, he must to remove it and created another one derived metric. Summarizing, there are many functions *(FFA3)* in order to let the user to do a well descriptive analysis *(FFA4)*.

Moreover, user can perform statistical analysis in Analytics using R *(FFA6)*. Analytics supports the deployment of R analytics from the R statistical environment as derived metrics and once an R analytic is deployed to Analytics as a derived metric, the statistical analysis can be added to and analyzed on visualizations. The connection with R is not very well documented and it is not very popular. The integrated Data mining functions in MicroStrategy requires a connection to R *(FFA5)*. The R scripts of the functions are already built and user only needs connect to R to execute the code.

As stated above, when data have type Time, Date or DateTime Analytics Desktop create derived attributes with additional time-related information. Similar things happen when an attribute has assigned a geo role. To sum up, when a field is containing time or geographic information it is considered different *(FFA7) (FFA8)*.

When user has a very large set of data in a dashboard, it can be easier to work with that data grouping by in into logical subsets, and viewing only one of the subsets at a time. Adding an attribute to the Page-by panel, user can click an attribute elements to use to group data. When user group data in a dashboard, the grouping is applied to all visualizations on the current layout tab. Each layout in a dashboard is grouped separately, without affecting the contents of the other layouts in the dashboard *(FFA9)*. Moreover, filter can be added in visualizations in order to display only filtered data. And filters can be on dimensions and also on metrics

(FFA10)(FFA11). On the other hand, when user is creating multiple visualizations in a same dashboard, he can filter data, selecting values in one visualization (the source) to automatically update the data displayed in another visualization (the target). In the settings menu of the source visualization user can choose the option to use as filter and can choose in which target visualizations the filter should be applied. Then, MicroStrategy allows the interaction between graphs but it is not automatic *(FFA12)*.

There are some assumptions about NULL data in files from where data is loaded. For Excel, .csv and text files, user should leave cells of data empty to represent NULL values, rather than using the text NULL *(FFA13)*. On the other hand, on measures, null data are not considered. In fact, they are omitted for new calculated measures, but there exist functions to create a new calculate measure related with null data of a measure. For example, Null function determines how many nulls are displayed. In contrast, null data for dimensions are considered as a one more value, but when this attribute is also in other dataset, Analytical Engine only use the non-null form value from the other dataset instead of the null form *(FFA14)*.

MicroStrategy Analytics offers many types of visualizations *(FFA15)*. There are graph visualization where user can choose between lines, bar, area, scatter, bubble, grid and pie graph. Moreover there is also the option to add double axis. There are also, grid visualizations, network visualization, heat map visualizations, map visualization, density map visualizations, and map with areas visualization. Particularly, for a map with areas visualization user must provide an attribute whose values include the names of each area in the map's base map. The base map is an ESRI map that contains the shape of each area that can be displayed in the visualization. The base maps available in MicroStrategy Analytics includes maps from continents, countries of the world, United States counties, United States regions, United States state abbreviations , United States state names, United States ZIP codes and World administrative divisions. Finally, MicroStrategy Analytics is not the tool with a wider variety of functions, but is the tool in which graphs can be more customized by user. Moreover, they can be totally modify by the user *(FFA16)*.

There has not been limitation to show many data dots, during the testing *(FFA17)*.

There is an option to refresh data, but it is not done automatically, user must order that clicking on an icon. And user can choose between overwrite existing data, update existing data as well as add new data and keep existing data and add new data *(FFA18)*.

Dashboard:

User can export the whole MicroStrategy file to keep the interaction in visualizations. When user exports a dashboard, the entire dashboard, including visualizations, filters, and so on, as well as the associated dataset, are exported. And then a other user can import it and works on it *(FFD1)*. Unfortunately, there are not templates integrated in dashboards *(FFD2)*. And user has totally freedom to build them, adding text, images, links... *(FFD3)*.

Reports:

There is not the option to build reports with MicroStrategy Analytics. Reports are considered dashboards saved them as pdf *(FFR1)*. Then, there are not templates for specific reports *(FFR2)*. And free design is the offered in dashboards *(FFR3)*.

Languages:

MicroStrategy Analytics Desktop can be displayed in more than two languages. They are Chinese, Danish, Dutch, English, French, German, Italian, Japanese, Korean, Polish, Portuguese, Russian, Spanish and Swedish *(FIL1)*.

Portability:

MicroStrategy Analytics is a solution only for Windows operating systems *(FIP1)*. The compatible versions are Windows Vista Business Edition SP2 (on x86 or x64), Windows Vista Enterprise Edition SP2 (on x86 or x64), Windows 7 Professional Edition SP1 (on x86 or x64), Windows 8 all editions (on x64), Windows Server 2008 Standard Edition R2 SP1 and SP2 (on x64),, Windows Server 2008 Enterprise Edition R2 SP1 and SP2 (on x64), and Windows Server 2012 Standard (on x64).

MicroStrategy platform offers MicroStrategy Analytics Express which is a cloud-based self-service visual analytics *(FIP2)*. It is free for a year and it let user to be able to establish an account, invite colleagues to connect, analyze and share their data insight and do it all at no charge. With Express, user can easily access and explore data in Analytics using interactive visualizations. Analytics Express and Analytics Desktop are not connected, then if you want to have a project done by Desktop in the cloud you should load the project to Express manually.

MicroStrategy also offers a custom free mobile app, which is built by MicroStrategy professionals in two weeks and it is for either the iPad, iPhone or Android. Then, there are not a application available for users, they must ask it to the corporation *(FIP3)*.

Using the Project by third parts:

Analytics allows user to create MicroStrategy files (.mstr) to share dashboards with other Analytics Desktop, Analytics Express and MicroStrategy Analytics Enterprise users *(FIU1)*.

Data Exchange:

Analytics Desktop also allows user to easily and rapidly export data from particular visualizations to Excel and CSV files. *(FID1) (FID4)(FID2)(FID3)*.

Security:

With Desktop edition there are not security devices. But, with the server one, MicroStrategy offers a innovation it is called Usher. Usher is an enterprise-grade mobile identity platform designed to provide security for every business process and application across an enterprise. It replaces traditional forms of enterprise identity such as IDs, passwords, and tokens, with mobile identity badges securely delivered on a smart phone. For example, to log in the Express SaaS user must introduce a user name and password or access by Usher app *(FSS1)*. Additionally, in the server edition an administrator can assign permissions to users *(FSS2)*.

Browsing facilities:

There are not many icons in toolbars, and it facilitates the use of the tool by the user *(UEB1)* .
Right click menus are not displayed. Elements in the page have a tab where user can click on and a menu is displayed *(UEB2)*.

Terminology:

Terminology is not difficult and it is in the standard, but it is not the tool with easiest terminology *(UET1)*.

Help and documentation:

The user manual is extremely detailed and it has many concepts definitions. These definitions is what differentiates the MicroStrategy Analytics user manual from others *(UEH1)*. The user guide is available in the net *(UEH2)*, next to other many papers with specific information *(UEH3)*.

Support and training:

There are courses offered on line or in particular education centers. and many of them can be customized with user data and delivered onsite *(UES1)*. About the support, users can hire different types of support depending on their requirements. All of them have phone and online support 24x/x365 *(UES2)* and also online supports *(UES3)*. Moreover, MicroStrategy also offers consulting services about technology implementations, technology management, data management, presentation and delivery and advisor services in particular solutions as education solutions, healthcare solutions, financial services,…*(UES4)*.

In the net user can sign in the Learning Portal, where he can access to many free courses and certified test about basic and advanced features *(UES5)*.

There is also a community where user can discover technical documents, latest release notes and product manuals. Moreover user can also ask questions, find solutions, and share insights about MicroStrategy implementation. *(UES6)*.

Windows and mouse interface:

Elements cannot be edited by double clicking *(UGW1)*. But, elements can be moved by drag and drop*(UGW2)*.

Display:

The screen layout can be modified by user, changing the background color, adding bar tools... *(UGD1)*.

Versatility:

By default, Analytics automatically checks for updates, downloads the update file, and then notifies you to install an update when it is available. User can also prevent Analytics from automatically checking for updates *(UOV1)*.

Compilation Speed:

The Desktop Edition, which is the used in this evaluation, compiled slower than the other tools*(EEC1)*.

Resources utilization:

Server Edition requires 4GB memory RAM *(EER2)* and 1GB of Hard disk space required*(EER3)*. Analytics can be installed in processors of x86 and x64 bits *(EER1)*.

Additionally, it requires a web browser and an Adobe Reader and Flash Player. Web Browser *(EES1)*.

Annex 6: Tableau evaluation

Data loading:

Tableau supports a wide variety of data source types. It is compatible with files of type Tableau Data Extract, Microsoft Access files, Microsoft Excel files, Text files and importing files from Tableau workbook. Moreover, Tableau can also be connected directly to some types of databases as Tableau Server, Microsoft Access, Actian VectorWise, Amazon Redshift, Aster Database, Cloudera Hadoop, DataStax Enterprise, EXA Solutions, Firebird, Google Analytics, Google Bigquery, Hortonworks Hadoop Hive, HP Vertica, IBM BigInsights, IBM DB2, IBM Netezza, MapR Hadoop Hive and MarkLogic *(FFI1)(FFI2)(FFI3)*. Other databases with ODBC connection are compatible with Tableau.

On the other hand Tableau can be also connected to Microsoft Access 2003 or later *(FFI4)*, Microsoft Excel 2007 or later *(FFI5)*, Microsoft Windows Azufre Marketplace DataMarket, OData, Tableau Data Extract and text files *(FFI8)*.

Connecting to an Excel file, all sheets from the file are loaded in Tableau's memory at same time. Moreover, user can also load crosstabs from Excel files. Tableau needs crosstabs in a specific format, and to get this, Tableau offers an Excel complement to normalize cross tabs in order to load the crosstab to Tableau *(FFI6) (FFI7)*.

The connection to different data source at same time is possible and the integration of them is easy because the relation between datasets is done automatically if the names are equal or it can be done manually by the user. Then, if there is a link field that connect two datasets, a chart can be build with fields from different datasets *(FFI9)(FFI10)*.

During the loading, user can visualize some rows of the dataset *(FFI11)*. User can also filter columns but he cannot filter rows*(FFI14)(FFI15)*. User can also decide the data format of the fields between number (decimal), number (integer), Date and time, Date, String and can also assign a Geographic Role *(FFI12)*. Data type is not determined in the data loading process *(FFI13)*. Once time data is loaded is when user can see the data type (dimension or measure) automatically assigned to fields and he can change it.

Moreover, Tableau creates an automatic field for each loaded dataset, in fact it is a measure which counts the number of registers of each dataset *(FFI16)*.

Tableau has not the option to clean data before load them. User can only rename the fields and the datasets *(FFI19)*.
Additionally, Tableau let rename fields and datasets during the loading, in particular during the creation of the data model *(FF17)(FF18)*.

Data model:

Data model can be done automatically relating fields with same name. But user can also do that manually and decide by which fields are the relations done, do not minding if they are not named equal. Moreover, user can also choose the type of join (iner, left and right) *(FFD1)*. With this option,

the final data model is tailored to user and then is easy to get the correct data model*(FFD2)*. Moreover, user can visualize the data model at every time *(FFD3)*.

Fields relations:
Although, if user does the model by himself is more difficult to find circular references, sometimes there can appear and then Tableau stops to display and remark it until user repairs the circular reference*(FFF1)(FFF2)*.
A same table can be used more than one time although it is loaded only one time. It is useful when a table with coordinates is related to a dataset containing regions where people are born and it is also related to another dataset containing regions where people are dead. Some tools needs to load coordinates data in two distinct tables one for each relation. However, Tableau can load only one time the table, and then use it in many relations *(FFF3)*.

Analysis:

During the analysis user can create new measures from dimensions and measures already loaded *(FFA1)(FFA2)*. There are many functions to create new measures although it is not the tool which offers more functions, but they are enough for a typical descriptive analysis *(FFA3)(FFA4)*. There are not many integrated data mining functions. User can forecast quantitative time-series data using exponential smoothing models integrated Tableau. With exponential smoothing, recent observations are given relatively more weight than older observations. These models capture the evolving trend or seasonality of your data and extrapolate them into the future. Forecasting is fully automatic, yet configurable. *(FFA5)*
Even so, Tableau can be connected to R Project and from there user can create a data mining analysis for Tableau's data and visualize the results in Tableau *(FFA6)*.

Tableau automatically assign geographic roles to fields with common location names such as State, Country, and so on. User can manually assign geographic roles to fields that don't use common names and weren't automatically detected. When user assign a field geographic role it is marked with a globe icon. It means that Tableau has automatically geo-coded the information in that field and associated each value with a latitude and longitude which user can use to geo-code his data. If Tableau does not recognize a geographic role user can create new geographic roles and assign them to the geographic fields in data. *(FFA7).*Time hierarchies are also created automatically *(FFA8)*.

A particularity of Tableau is that user can create sub-datasets and analyze only them(FFA9). Then user can create charts only for particular datasets. Moreover, user can also add typical filters to a visualization and filter displayed data by dimensions and measures*(FFA10)(FFA11)*. Tableau offers interaction in a visualization, but interaction between visualizations it is not possible *(FFA12)*.

There are some assumptions about NULL data in files from where data is loaded. For Excel, csv and text files, user should leave cells of data empty to represent NULL values, rather than using the text NULL. When data comes from a database and there are nulls, they pass automatically to Tableau as nulls *(FFA13)*.

Tableau don not consider Nulls values. In fact, if there are null values in a field it can not be displayed in a visualization and the tool alarms about the type of problem. The solutions then is filter data to display all the non-values *(FFA14)*.
There are many graphs to display data and they offer many possible modifications to tailor it to user's requirements *(FFA15)(FFA16)*.

Tableau does not present any problem to display huge amounts of data during the testing*(FFA17)*.

In tableau, by default, the data refresh is done automatically although user can deselect this option*(FFA18)*.

Dashboards:

In dashboards, user has many options to design them. He can change the format of lines, rows, columns, text, alignment, shading, borders, lines, …*(FFD1)*
User can share dashboards publishing in Tableau Public (a free service that lets the user publish interactive data to the web) or Tableau Server. Users of Personal Edition can only publish them in Tableau public while Professional Edition users can publish to Tableau Server. In both cases, user can embed passwords and authentification to visualize it. User can also save the packaged workbook to use it for other persons using Tableau. The packaged workbook contains the workbook with a copy of any local file data sources and background images.
From individual views user can export the image in the clipboard or save it or data in a Excel file or Access database. It is a convenient way to share work with coworkers who do not have access to Tableau. User can also print individual images or the whole worksheets to PDF choosing what elements of the dashboard user wants to include in the printed PDF.
User can save a workbook as the role of bookmark to preserve the data connection, formatting, calculated fields and groups, essentially provides a template from which to create future workbooks. *(FFD2)*
Unfortunately, there are not template in order to reuse them several times *(FFD3)*.

Reporting:
Reports are done printing on PDF the work done in worksheets. Before user prints, there are several options that he can set to specify how the worksheet will look when it is printed. User can configure the Page Setup settings and print the file with the settings applied. Some settings that user can do are specifying the margins, centering, print scaling, the layout, legends… *(FFR1)(FFR2)(FFR3).* With Tableau Server, they can be shared among coworkers.

Languages:
English, French, German, Spanish, Brazilian Portuguese, Japanese, Korean and Simplified Chinese *(FIL1)*.

Portability:
Tableau can be installed in Microsoft Windows and also Mac systems *(FIP1)*.

Tableau offers the product Tableau online which is a hosted version of Tableau server, with no setup required. User can publish dashboards with Tableau Desktop and share them with colleagues, partners or customers. It Works in web browser or mobile device(dashboards are automatically optimized for mobile tablets without any programming. Only authorized users will be able to interact with data and dashboards published there *(FIP2)*.
With the Server Edition, user can get a mobile application. *(FIP3)*

Using the project by third parts:
With Server Edition there is no problem to share projects with others. Moreover, with the Desktop Edition have not problems about that. *(FIU1)*

Data exportation:
From Professional Edition data can be published in Tableau server. While with the Personal Edition user can publish on Tableau Public. User can select any portion of a data view to export into Microsoft Excel or Access database. *(FID1) (FID2)(FID3) (FID4)*

Security:
In Tableau public and Server there can be permissions specified to regulate Access to the workbook. *(FSS1)(FFS2)*

Learning time:
It is not the most easy tool. The automatic aggregation of all measures, by dimensions can confuses the user. *(UEL1)*

Browsing facilities:
There are many icons, and the same action can be done by more than one way. It can be confused *(UEB1)*. Right-clicking displays a menu with Tableau actions *(UEB2)*.

Terminology:
The terminology keeps on standard *(UET1)*.

Help and documentation:
The user manual is available in the web, there is many information and it can be understand by non-expert users. *(UEH1)(UEH2)(UEH3)*

Support training:
There is the option of accessing to tailor-made training courses *(UES1)*. Additionally, there are many online course, video tutorials in the web and integrated in Tableau application. It is very good in free formation*(UES5)*.
There is the option to get support by call but if user has an special support account *(UES2)(UES3)*. There is many information to support user in the web and user can create a case with a particular question, and with a couple of days an expert will answer it. Moreover, there is a community that can also solve problems. Additionally, Tableau also offers consulting services *(UES4)*. There are also a community where users can share their doubts*(UES6)*.

Windows and mouse interface:
Elements can be edited by double-clicking. *(UGW1)*. Moreover, Tableau Works with dragging and dropping elements *(UGW2)*.

Display:
User can set the screen layout *(UGD1)*.

Versatility:
The upgrades are not automatic, and user must go to the web to upgrade the version *(UOV1)*.

Compilation Speed:
During the test, it is not the tool which is faster, but it runs acceptably *(EEC1)*.

Resources Utilization:
Tableau can be installed in processors x86 and also in x64*(EER1)*. The minimum required RAM are 2GB *(EER2)* and the Hard disk space required is 750MB *(EER3)*

Software requirements:
Non extra softwares are required, excepto f a browser, an adobe reader and a Flash Player.*(EES1)*

Annex 7: Reporting examples

In this chapter there are some charts created by the 4 evaluated Self-Service BI applications. These charts are showed in order to introduce their functionality to the reader.

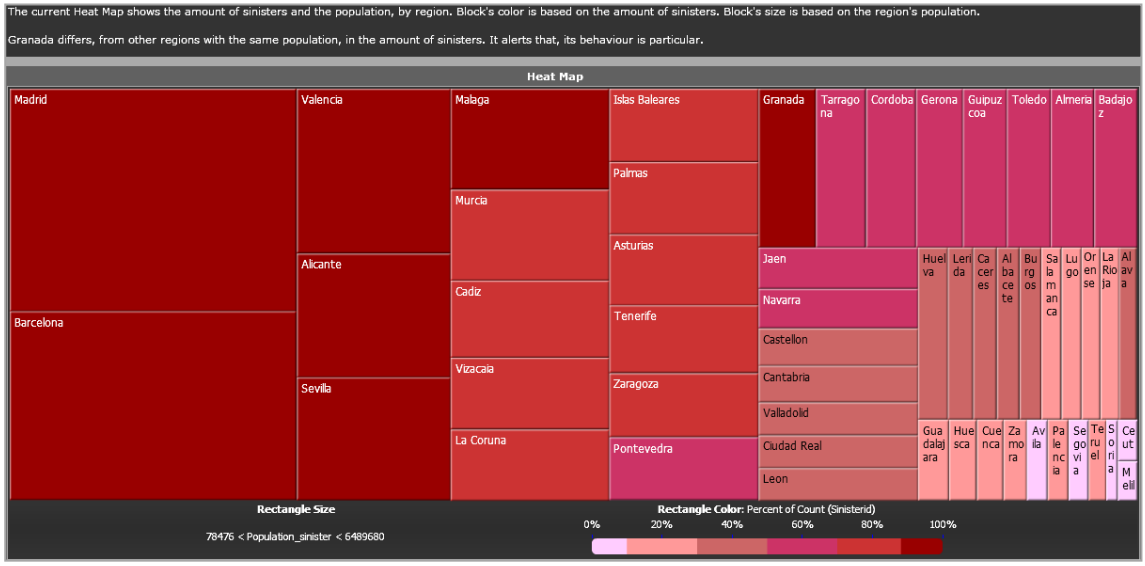


Fig. 28 Example of a Heat Map build by MicroStrategy Analyitics

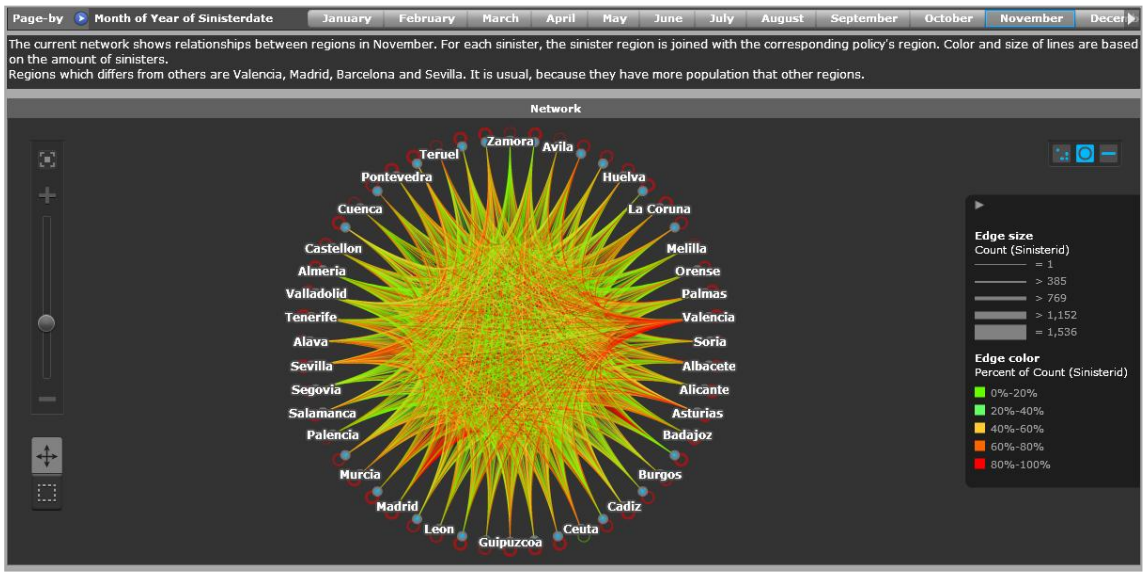


Fig. 29 Example of a Network chart built by MicroStrategy Analyitics

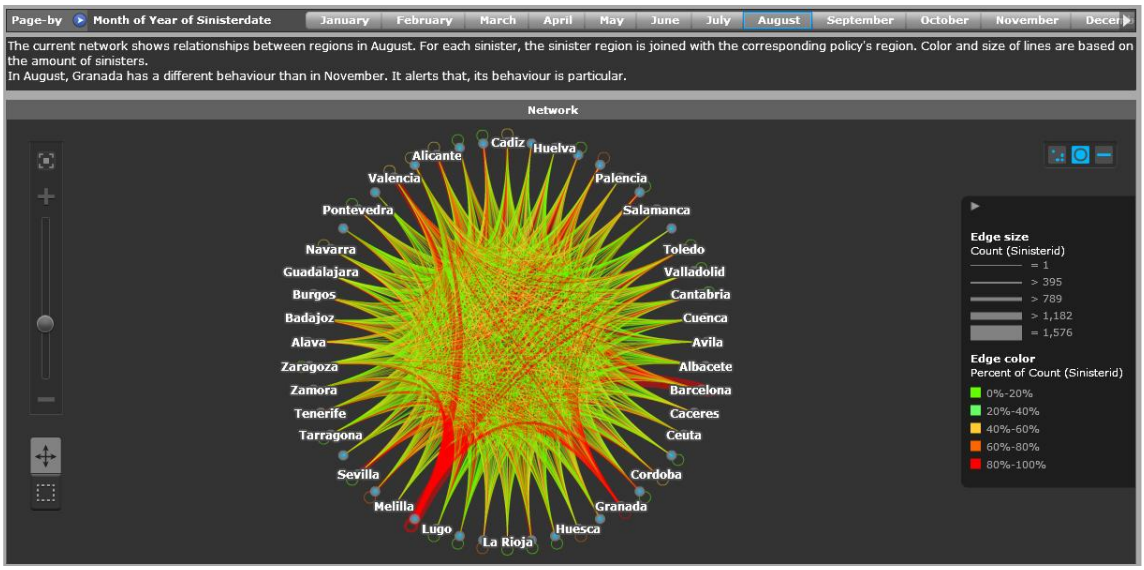


Fig. 30 Example of a Network chart built by MicroStrategy Analyitics

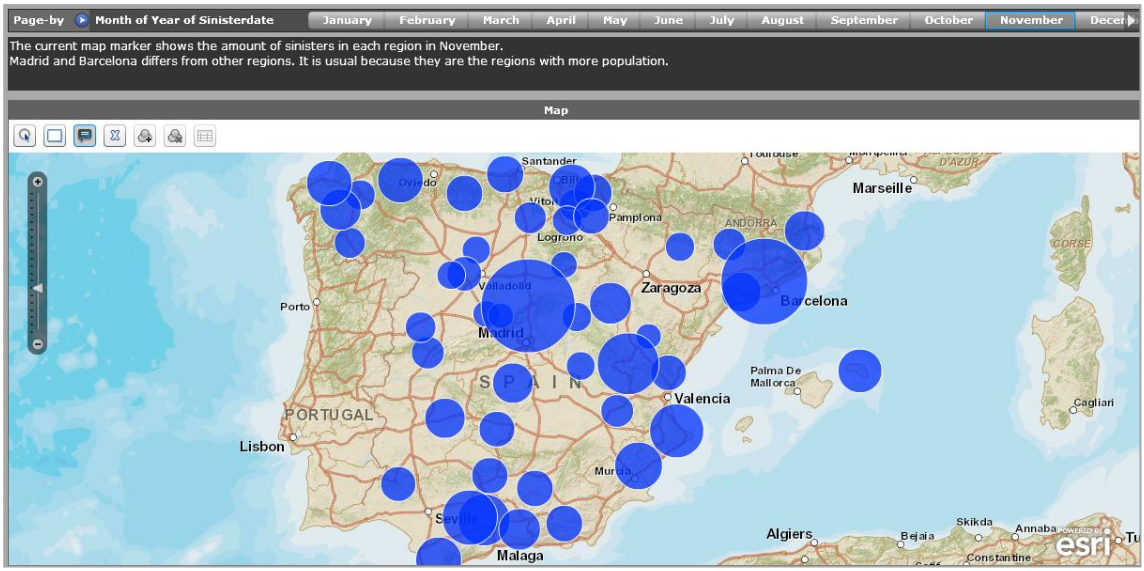


Fig. 31 Example of a Map chart built by MicroStrategy Analyitics

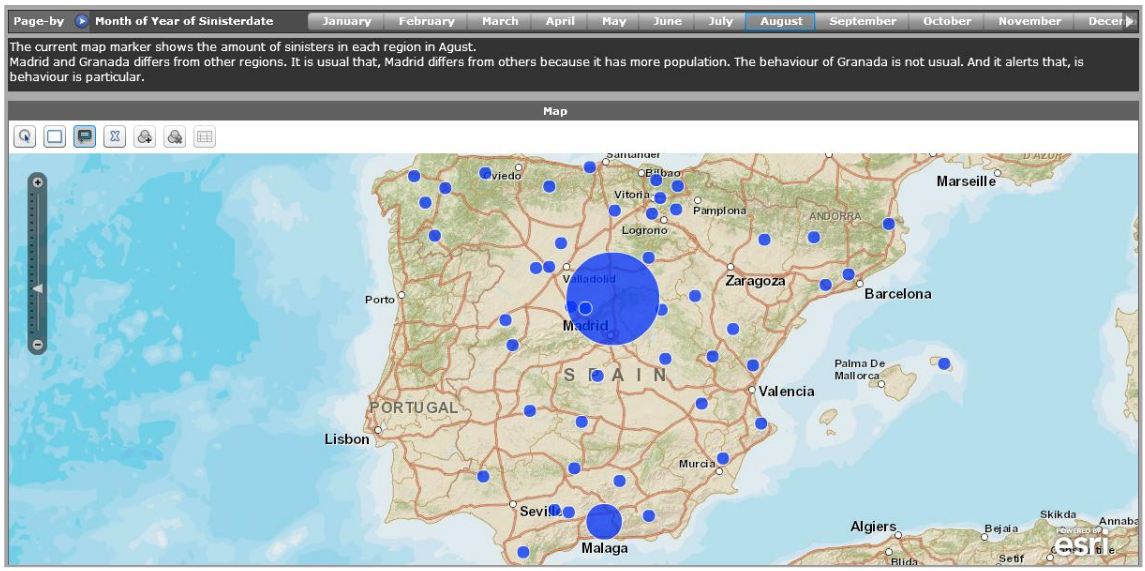


Fig. 32 Example of a Map chart built by MicroStrategy Analyitics

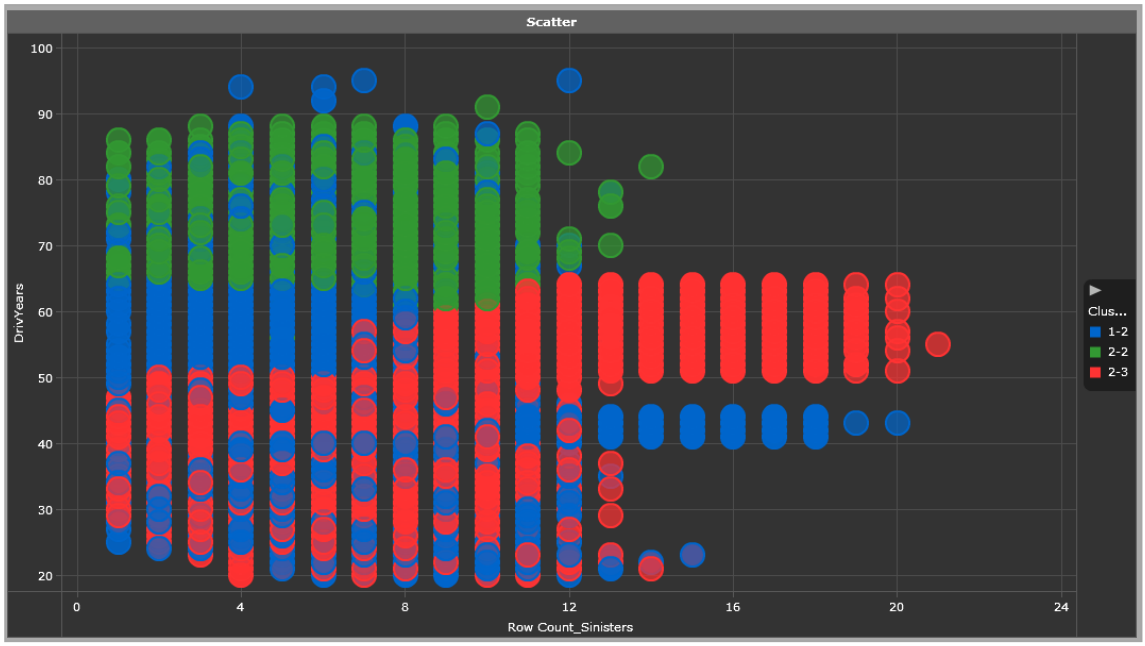


Fig. 33 Example of a k-means classification plot, done by the previous connection of MicroStrategy Analytics to R

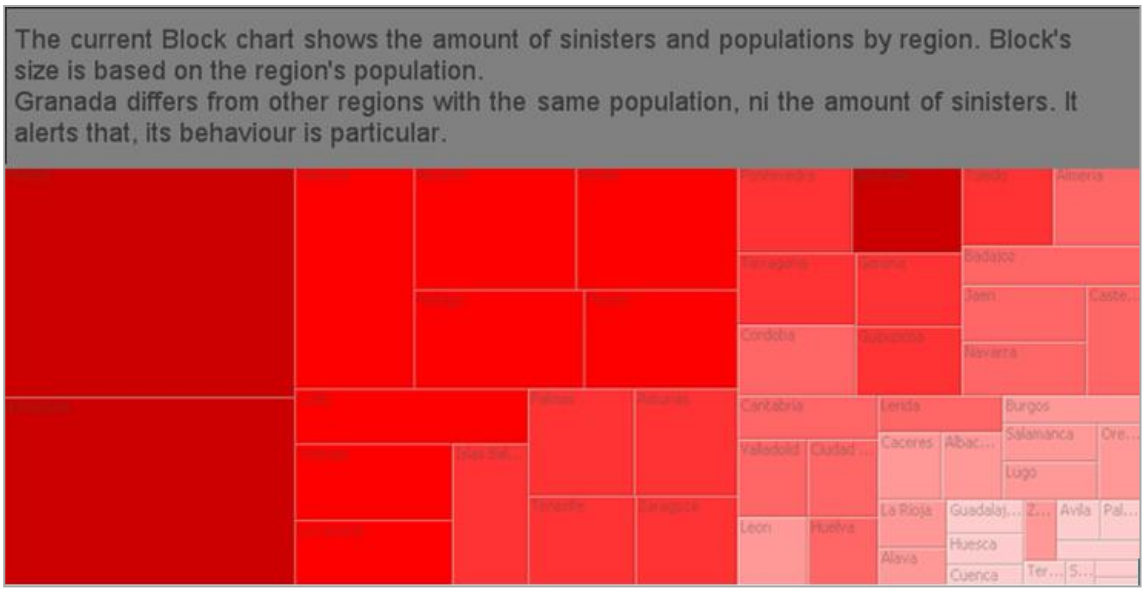


Fig. 34 Example of a Heat Map built by QlikView

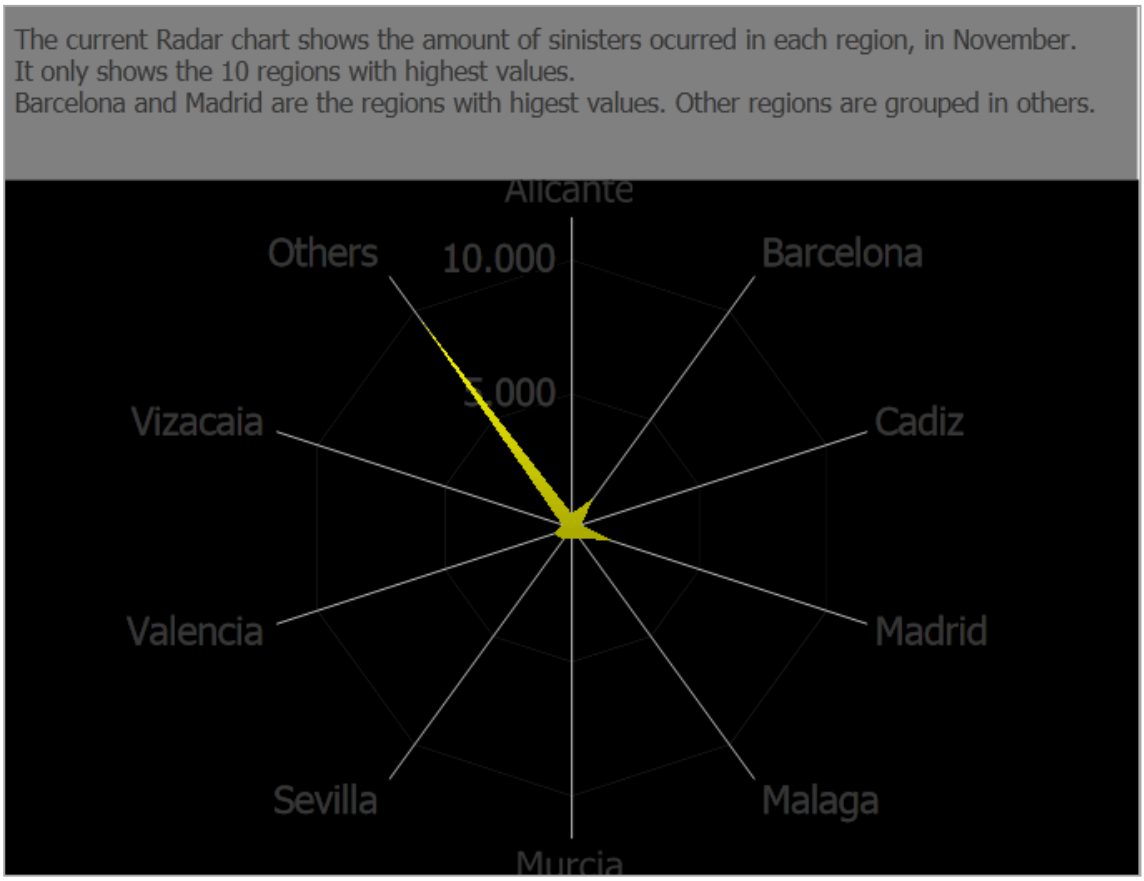


Fig. 35 Example of a Radar Map built by QlikView

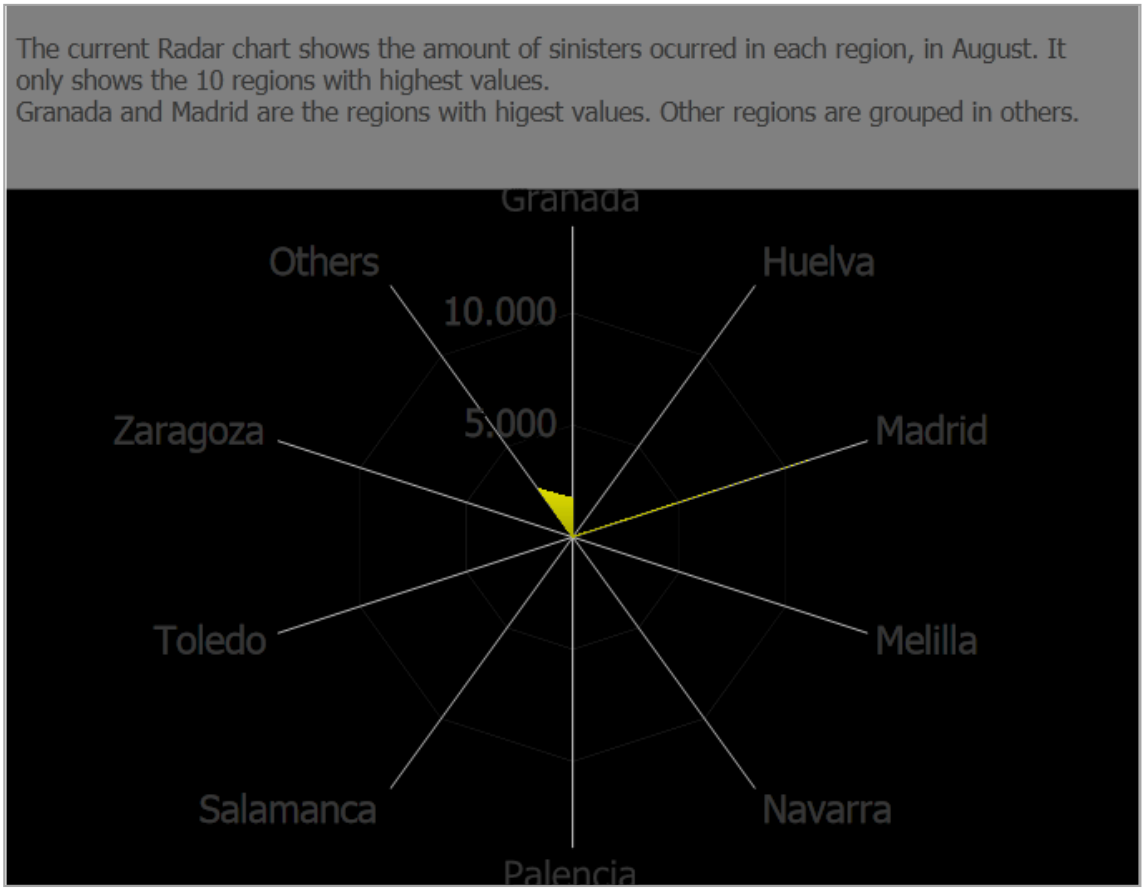


Fig. 36 Example of a Radar Map built by QlikView

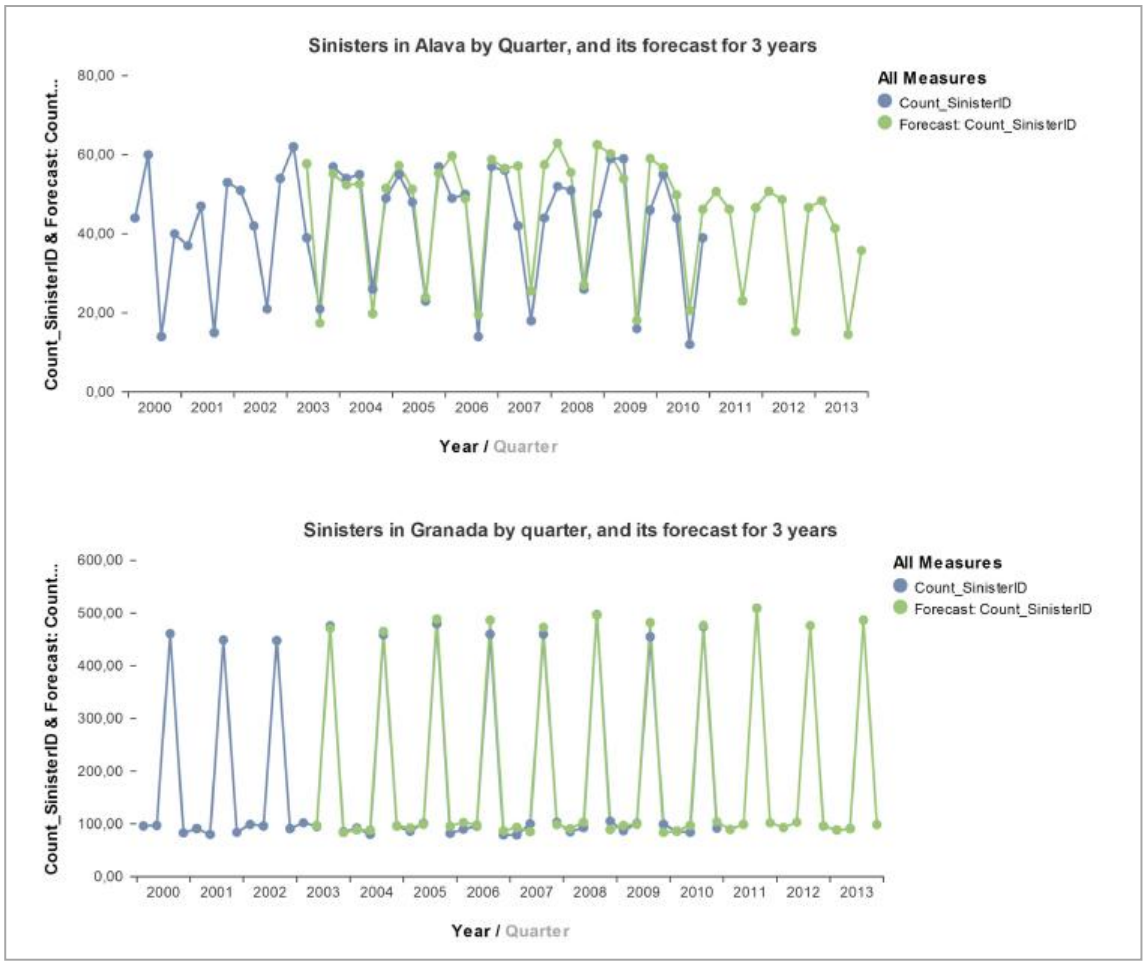


Fig. 37 Example of a forecasting ,built by SAP Lumira

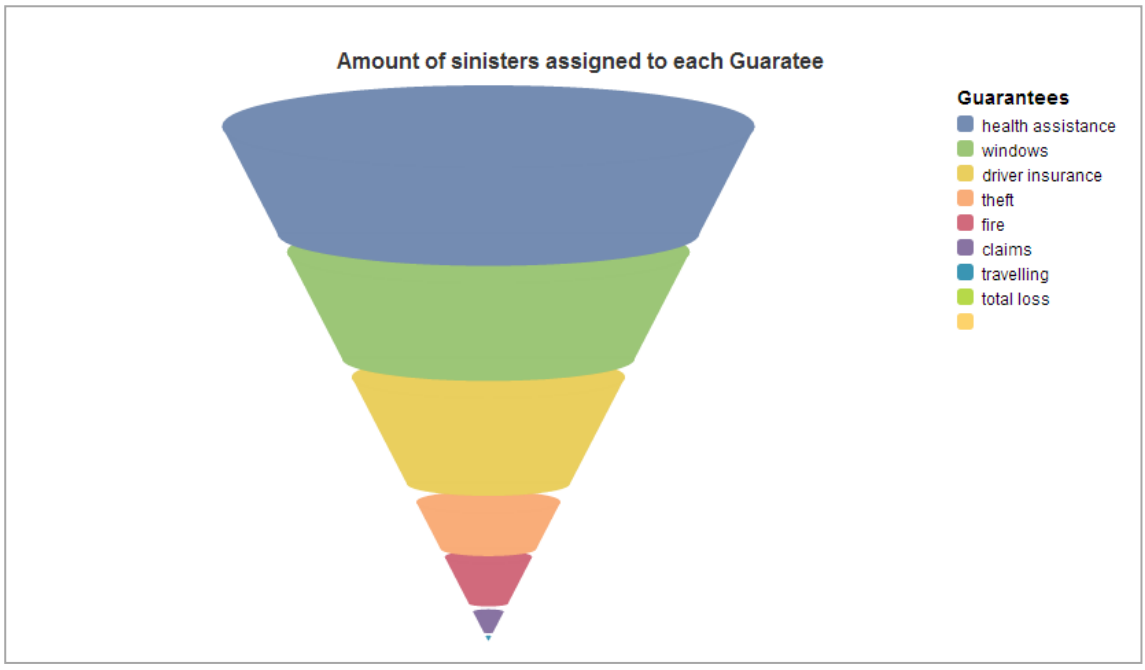


Fig. 38 Example of a Funnel map, built by SAP Lumira

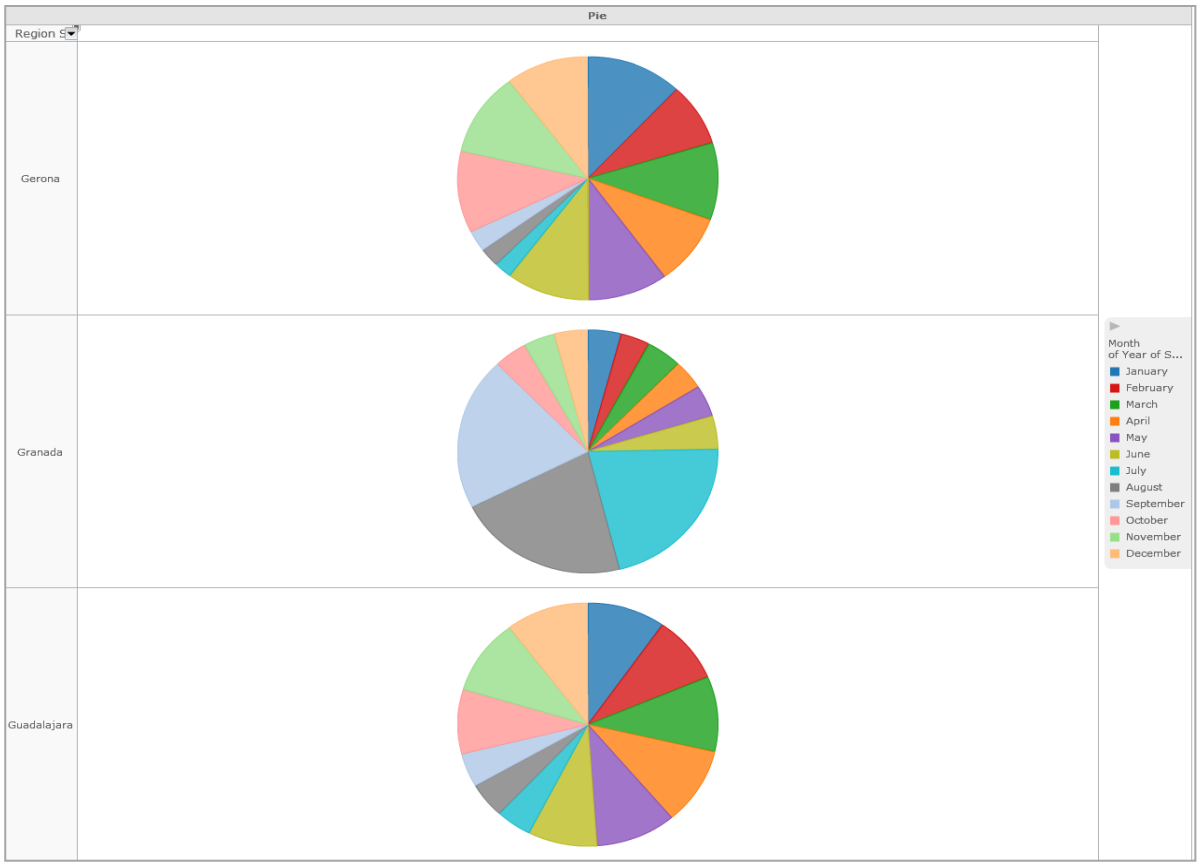


Fig. 39 Pie charts, built by Tableau