How to choose the right statistical software?—a method increasing the post-purchase satisfaction

Roberto Cavaliere

ADALTA snc, Italy

Correspondence to: Roberto Cavaliere. Corso Umberto I, 50, Mercato San Severino, SA 84085, Italy. Email: r.cavaliere@adalta.it.

Abstract: Nowadays, we live in the "data era" where the use of statistical or data analysis software is inevitable, in any research field. This means that the choice of the right software tool or platform is a strategic issue for a research department. Nevertheless, in many cases decision makers do not pay the right attention to a comprehensive and appropriate evaluation of what the market offers. Indeed, the choice still depends on few factors like, for instance, researcher's personal inclination, e.g., which software have been used at the university or is already known. This is not wrong in principle, but in some cases it's not enough at all and might lead to a "dead end" situation, typically after months or years of investments already done on the wrong software. This article, far from being a full and complete guide to statistical software evaluation, aims to illustrate some key points of the decision process and introduce an extended range of factors which can help to undertake the right choice, at least in potential. There is not enough literature about that topic, most of the time underestimated, both in the traditional literature and even in the so called "gray literature", even if some documents or short pages can be found online. Anyhow, it seems there is not a common and known standpoint about the process of software evaluation from the final user perspective. We suggests a multi-factor analysis leading to an evaluation matrix tool, to be intended as a flexible and customizable tool, aimed to provide a clearer picture of the software alternatives available, not in abstract but related to the researcher's own context and needs. This method is a result of about twenty years of experience of the author in the field of evaluating and using technical-computing software and partially arises from a research made about such topics as part of a project funded by European Commission under the Lifelong Learning Programme 2011.

Keywords: Software evaluation; benchmarking; software buying decision process

Submitted Oct 19, 2015. Accepted for publication Nov 03, 2015.
doi: 10.3978/j.issn.2072-1439.2015.11.57
View this article at: http://dx.doi.org/10.3978/j.issn.2072-1439.2015.11.57

Introduction

Before going into technical aspect of the article, a preamble is necessary. The traditional literature about "decision process" is quite huge and mainly refers to the "decision making" subject in psychology [1]. Another part (smaller than the first one) of the traditional literature is about the "buyer decision process" [2], inside of which a very small documentation (mainly gray literature) is found about "software buyer decision process". Almost nothing is available about "statistical software buyer decision process" (just few articles and comments on blogs). This to say two things: papers, articles, surveys and other references here mentioned are usually coming from the general software buyer decision process and when possible an inference is done to adjust results to our case study about statistical software; there is no a clear method or strategy in this kind of decision process so what is presented here is basically the experience and the results of personal research made by the author in about twenty years of works with mathematical software (programming, training, marketing, evaluation, etc.).

It might appear like a provocation, but the main issue addressed by this work is the lack of attention paid by

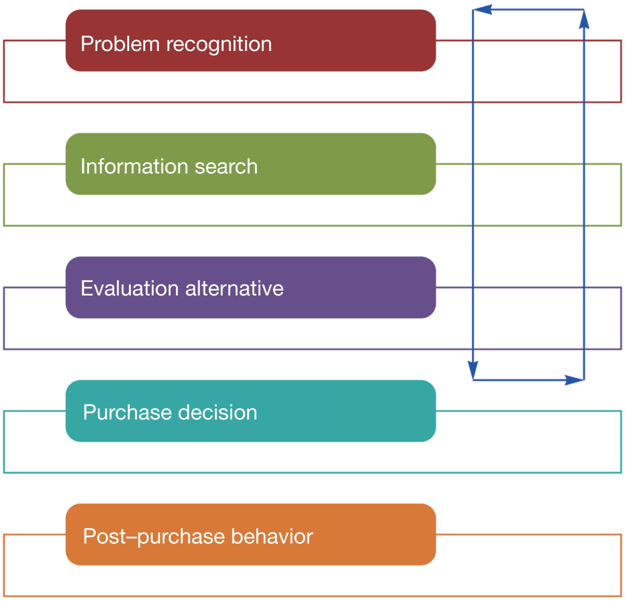

Figure 1 Five stage decision making process that customers use in any purchase.

researchers in the software evaluation process at the beginning of their career or projects. The wide diffusion of general knowledge about computer and the information technology culture has a critical side effect, not recognized by the most part of scientific world, that is the persuasion that almost everybody has, about the easiness of how to evaluate/choose a software to buy. If you need to navigate on internet or write a document it is quite easy to decide which software has to be used, but if you have to study a complex natural phenomena or you need to implement a mathematical model simulating your research topic, the choice is not equally easy. Unfortunately, many times people think it is the same.

Extending the point of view to a wider context, the software evaluation is one of the actions performed inside the more complex "buying decisions process", that in turns belongs to the "decision making process". It's well known the five stage decision making process summarized by the *Figure 1*.

As the thin arrows suggest, the first three stages are sometimes repeated cyclically to refine the process going back to review the problem statements, retrieving additional information from the market and then making further benchmarking to compare different products. Our suggested methodology is based on a review of such three stages in a detailed way.

"*Nobel laureate Herbert A. Simon sees economic decision-making as a vain attempt to be rational. He claims (in 1947 and 1957) that if a complete analysis is to be done, a decision will be immensely complex. He also says that peoples' information processing ability is limited. The assumption of a perfectly rational economic actor is unrealistic. Consumers are influenced by emotional and non-rational considerations making attempts to be rational only partially successful*" [3]. Note that this survey is not focused on a specific software, however if we consider the generalization of the buying decision process model the results can be considered valid, with a small tolerance, even for the specific field of statistical software.

So, why software buyer should act differently from any other buyer?

Going back to software evaluation, and thus purchase, again emotional and non-rational considerations are the most influencing factors. For instance, if the researcher has a previous experience with a statistical software, even if related to a totally different mathematical problem, he/she will be strongly convinced that the tool used in the past is still the right choice even for the new problem. This decision is more due to the trust that he/she has in its own knowledge about the software rather than based on a technical analysis of what the software is actually able to do and, on the other hand, it is based on the fear that more time and efforts will be required to learn a new software. In addition to this natural inclination, when dealing with software there are some extra factors influencing the behavior. Indeed, unlike other kind of goods (product or services) it is not so evident that software has a life cycle, needs to be maintained, updated and repaired, almost like a car, a bike or a house. Most of us think that being the software intangible, it doesn't need to be considered like any other material good. This means that the evaluation is frequently done considering a simplified schema: what is the technical problem to be solved (small or direct problem recognition, not a comprehensive needs analysis), find a couple of software that has a technical feature matching that need (information search), if one of the selected software is already known stop and buy (no evaluation, just purchase decision) otherwise collect some additional technical information (evaluation alternatives) then decide (purchase decision) (*Figure 2*).

From a survey run by Capterra [3] arise some date confirming such consideration: "1/3 of respondents did not demo any product" and "22% only considered 1 option", only "14% considered more than 3 options".

Apart from technical features, that with all likelihood remain at the first position in the list of critical factors, there are many other elements (latent influencing factors) which can contribute to the success of the software selection, in terms of satisfaction of the user about what he/she can potentially do with the statistical software purchased and what he/she really does in the next future (post-purchase behavior). For instance, elements like software usability, technical support, training courses/material, community of users, and many others, may influence the actual exploitation of the purchased software.

However, from the buyer's behavior point of view, another important aspect is registered in many circumstances: the high rate of influence of stakeholders, like other people using the same software or experts like programmer, blogger or scientists. Indeed. In many researchers it arises that the most influencing factor is the peer recommendation or "word of mouth". The following table shows the result of a survey where respondents were required to self-evaluate their adopter type (*Table 1*).

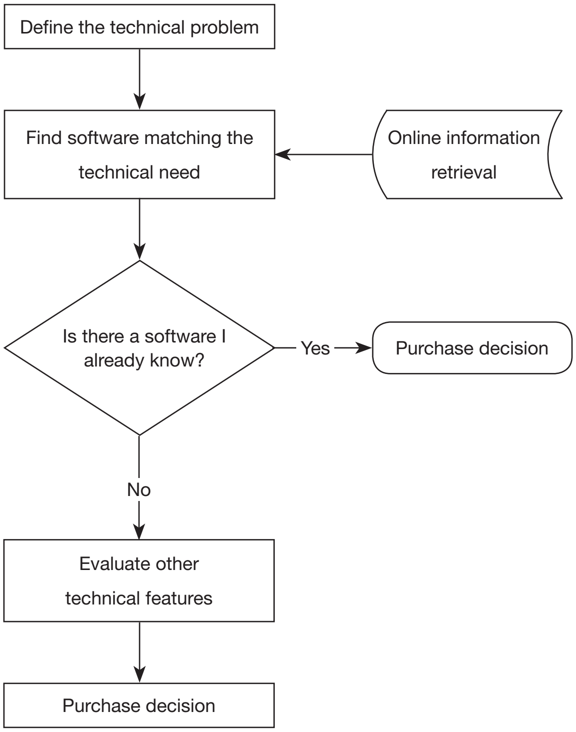
Define the technical problem
Find software matching the technical need
Online information retrieval
Is there a software I already know?
Yes
Purchase decision
No
Evaluate other technical features
Purchase decision

Figure 2 Simplified purchase decision model for software.

As shown in the table, about the 61% adopts caution and prefers to be supported by other stakeholders. In the next paragraphs we will turn back to this important aspect, highlighting which factors are related to this behavior.

Finally, another key point should be kept in mind during software evaluation, in particular when considering general purpose software like statistical or mathematical one. The most important investment we do buying a software is not the money we pay it, but the time we spend to learn and use it. As mentioned before, for some aspects, especially from the buying decision process point of view, the purchase of software is similar to the purchase of any other material good (cars, bikes or houses), but on the other hand it is totally different. Buying a new car doesn't mean learn to drive once again or invest more time to travel or use the car. Buying a new statistical software may require re-training or completely new programming paradigm or technics, new skills, and so on. These are all aspects that should be considered and evaluated during the stage three of the buying decision process. And, these aspects are those impacting the emotional and non-rational behaviors.

The general context: software tools for statistical data analysis

According to the context that we are faced with, there could be dozens of key factors to be considered. Indeed, computer

Table 1 Innovativeness in regards purchase of software upgrade for most important software

Adopter type when purchasing the software upgrade for most important software	Frequency	%
Enthusiastic, innovator, first to buy, risk taker, evaluator	5	3.3%
Visionary, early adopter, leader in your industry, others ask your opinion	39	25.5%
Pragmatic, cautious, not the first nor the last to buy, ask others what to buy	94	61.4%
Conservative, late adopter, not confident, follower, doubtful	13	8.5%
Skeptical, last to purchase, have to be absolutely certain	2	1.3%
Total	153	100.0%

Source: D. Roberts, A. Cater-Steel, M. Toleman "Factors Influencing the Decisions of SMEs to Purchase Software Package Upgrades", 17th Australasian Conference on Information Systems— 6–8 Dec 2006, Adelaide.

Table 2 Relation between number of people involved in the buyer process and time needed

People involved	Time needed (months)
1	1.8
2	2.2
3	2.7
4 to 6	4.4
7 to 9	5.1
10 or more	7.2

and software are fundamental devices in almost any modern working context, so purchase of software may be referred to many different situation. It also means that we have to consider questions like: who takes final decision about software purchase in the company? Is there a company policy about which kind of software or license schema to buy? Is there any other IT infrastructure managing digital information that has to be connected with the statistical software? (for instance the database management systems for large repository of medical data); is there a fixed budget or the price is not an issue?

It's easy to understand that all these questions lead to many possible paths and each one will require the right attention. Just to show an example, from the aforementioned survey arise that in the 40% of software purchase, the CEO/President is directly involved in the decision process and in 55% the IT personnel is involved, and then in the 60% of the cases there are more than three people involved. Moreover, it is also known that the number of people involved in the decision process significantly influence the time needed to achieve a decision (*Table 2*).

For that, here we consider a simplified situation where one researcher autonomously decided to use a statistical software for medical research, thus he/she is directly involved in the evaluation process. In other words, we don't consider more than one people participating to that process and we don't consider the company profile or policy about software equipment purchase.

Doing a deeper analysis about the specificity of statistical software, we have to say that from a user point of view, such a kind of software can be considered more a programming language [4] than an application software [5]. There are, indeed, many steps during the statistical data analysis process that require technical skills. Basically we can consider a four stage process:

(I) Data import, management, recording, organization—put data into the model, optimizing the fruition, save them, make available for the analysis according to what kind of statistical model is needed.

(II) Data analysis—the core statistical calculations, many algorithms are already available but many times a border-line research might require exploration of new method or algorithms.

(III) Data representation, reporting, graphics—numerical data are usually well represented by many kind of charts and graphics, so the visualization component is fundamental.

(IV) Deployment—how to share results and achievements, documentation and reports according to standards.

The following picture briefly describes these general stages of a data analysis workflow with a few other details (*Figure 3*).

This means that anyone who intends to make medical statistical analysis should have two additional skills: strong theoretical background in statistics and software programming practice. Indeed, as shown in the figure, if we consider the whole workflow we can see that while statistics background is needed only for the "data analysis core phases", all other steps like the import, management and storage of data, the production of reports or graphics, the delivery of final results may require additional programming capability. More important, all these factors have also a meaningful role during the evaluation process, when the decider has to check some features which may have negative impact on the potential development of new algorithms and functionalities from the statistical point of view, and have impact on the learning curve and actual usability, from the programming point of view.

Classification of key factors

In the previous paragraphs we introduced many hints letting us to perceive how complex is the buying decision process when dealing with a new (statistical) software. If we want to go even deeper, there are too many factors influencing such process so there is the need to make a clear schema and try to simplify the picture. Indeed, what is really interesting is to suggest a possible approach to the evaluation process that simplify the final decision and not to make it more complex that what it is in realty.

To do that, we start from a classification of factors according to the following characteristics:

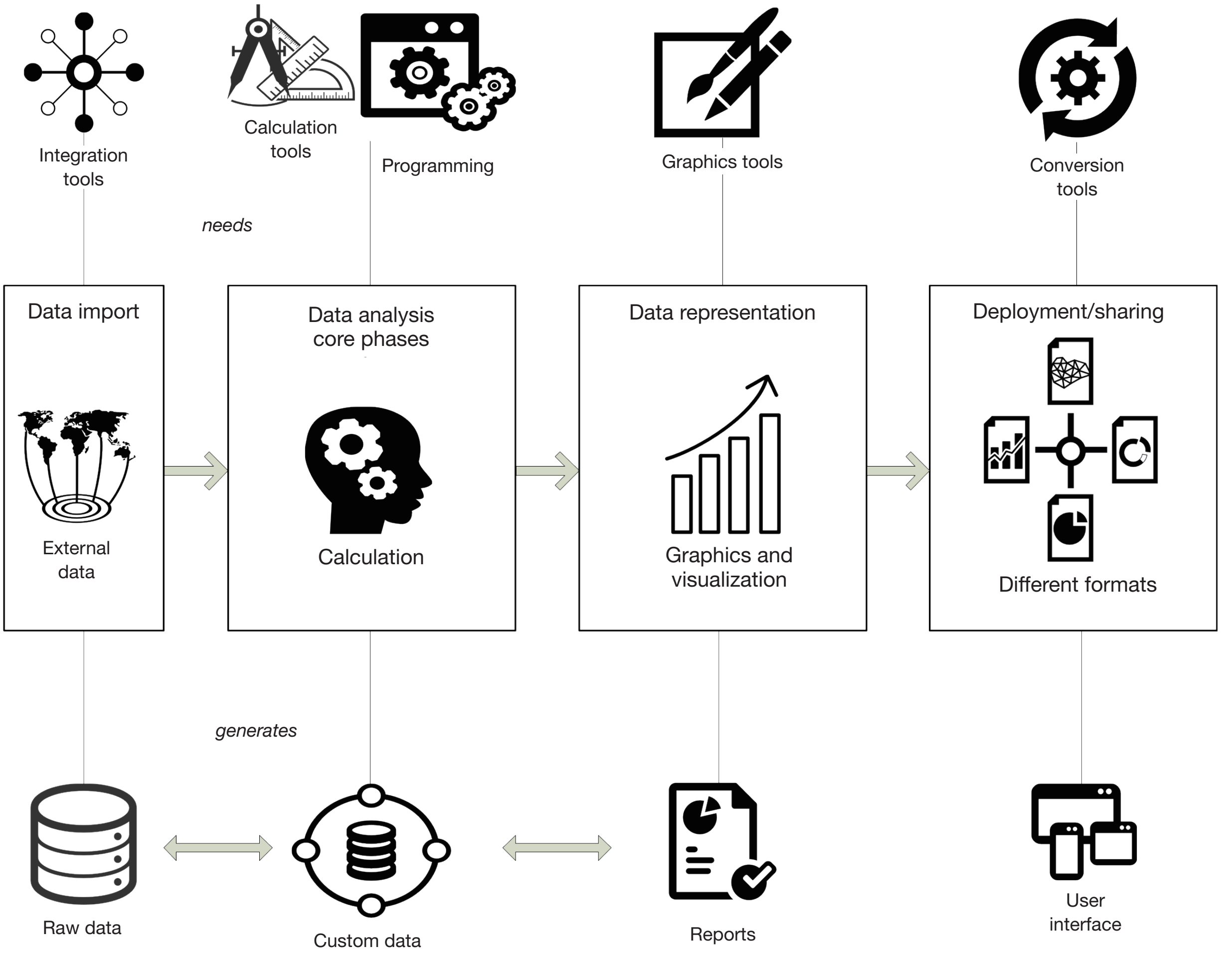

Figure 3 The general workflow of a statistical data analysis process.

- Nature of the factor:
  - Some factors are like a Boolean variables, they can assume the value yes or no; we call them attributes or objective factor.
  - Others are more like optional arguments, that means they may have multiple or subjective interpretation because the actual value depends on the buyer feeling, perception, skills or other features.
- Origin of the factor:
  - Endogenous factors are strictly related to the software itself, being feature of the product, for instance the language, the technical quality or built-in functionalities, the integrability with other software and so on. These are typically strictly related to technical requirements.
  - Exogenous are related to any element influencing the software evaluation but being originated externally to the software itself, for instance the producer company, developers, user community, blogger, influencer and other stakeholders. These are typically related to non-technical requirements of the context surrounding the software, where stakeholders play important roles.

Following such classification, we can define a matrix helping us to collocate each factor and then to better evaluate them. This figure shows an example of that matrix of factors (*Figure 4*).

It has to be said that this classification can be subject to some particular interpretations according to specific cases, that mean for some circumstances a factor can be moved from one box to another. Therefor the matrix of factors is a starting point of the evaluation process and not a rigid or unmodifiable schema.

	Endogenous	Exogenous
Subjective	Usability Productivity Cognitive accessibility Learning curve Programming or analytical approach Scalability Performance enhancements	Training courses General users community Other users in the same scientific field Blogger, influencer or stakeholders Consultant, expert availability
Objective	Functional requirements: mathematical functions, probability distributions, statistical functions, import/export capabilities, debugging tools, etc. Non-functional requirements: portability, interoperability, etc. Language of the interface and documentation Open-source, free or commercial paradigm	Books, articles and manuals Service or product Local reseller Prize/Upgrade policy Versioning License schema/policy Bug fixing policy

Figure 4 An example of matrix of factor classification.

The objective and endogenous factors are quite easy to discover and evaluate, because are both functional and non-functional requirements of the software and thus typically the most evident information provided by the makers, normally in the main web pages of the product. The characteristic of these requirements is that they cannot be interpreted or changed and we can just accept or refuse them. For that they do not represent a critical point of the decision process. Just to make some examples, if we need some mathematical model or statistical methods and/or particular tests we look for software having such functional requirements and if there aren't we don't have to spend more time on the valuation, the software doesn't fit our main need. Again, if we found a software that is in English and all documentation is written in English and we don't know that language, we have a clear idea of what will be the main difficult by adopting such software, and this is a fact more than an interpretable matter. Or, if we are looking for a software that has to be integrated into some other IT infrastructures or software in our laboratory, and a statistical software we have found on the market doesn't have such level of interoperability we cannot change or interpret this lack of feature in any way, we can just evaluate how critical it is for our topics and then decide if accept or not the feature. For that the first box in the bottom-left corner of the matrix can be also named "Yes/No factors".

The objective and exogenous factors, on the contrary, are not directly related to the software product itself, rather are linked to elements external to it. Typically here we have non-functional requirements. For instance, can be considered in this category all those factors related to the policy adopted by the producer company like the price, the license schema, the existing of a national/local reseller that can help in the evaluation process, the versioning policy, etc. Even if these elements are typically fixed (for instance the price or the license schema) sometimes they can be ponderable or eventually even negotiated with the producer because are more policy than technical feature (unmodifiable just for one user). For instance, in particular cases an unusual kind of installation can be agreed by the producer/seller if there are motivations that we disclose to them. Or, to make another example, we can agree to buy a service like a software hiring for some years, including upgrades and then after that period definitively decide if to keep the software or give back to the producer. All that to say that some features can also be evaluated jointly with the producer or the local reseller in order to better fit our particular needs. This happens in particular for commercial products, being the free or open-source software less tied up to factors like license, price and upgrades. Due to this relatively small degree of freedom, the bottom-right corner of the matrix can be named "Ponderable/Negotiable factors".

The subjective and endogenous factors are mainly related to non-functional requirements and are not easy to evaluate because are those properties interpretable on the base of our previous knowledge and/or skills, knowledge and feeling about main topics of statistical software, that are as aforementioned principally the theoretical statistics and programming knowledge. Indeed, features like usability, configuration management, platform compatibility and/or requirements, interoperability, portability, reliability, affordability, efficiency, extensibility, response time and many other are not evaluable on a fixed example, rather they require more than one case study or complete example related to what we really need to do with the software to be tested and measured. Moreover, most of such requirements also depend on the level of our skills and competences. For instance, the performance and the efficiency of a statistical software may depend on many factors, like the programming paradigm we adopt when developing our algorithms or how well we know the software. Many times these requirements cannot be evaluated before purchasing the software and for that they represent the factors that have the highest impact on the post-purchase behavior and negative feeling. The following results arise from the survey [3]:

Top three reasons for wanting to purchase new software:

(I) Previous software we were using for the same purpose was out of date;

(II) Needed/ wanted to increase worker efficiency/ productivity;

(III) Needed/ wanted to reduce costs by using software to optimize operations/processes.

Most difficult parts of the software selection process were:

(I) Getting a clear picture of how well each possible software option could meet specific needs;

(II) Being able to make comparisons between software companies/vendors;

(IV) Absorbing and understanding the information available about different software solutions.

For these reasons, the top-left corner of the matrix can be named "Transitory/Improvable" factors.

The subjective and exogenous factors are even more complex to be evaluated because are intangible resources and there is not a metric to measure them. However, in the use of any software the phenomena of user community is very important and in many cases may represent the only way to get support in difficult situation. Nowadays, many software producers prefer to sponsor and support the creation of community of users and maintain their social life through forums, wiki, conferences or specific technological platform rather than invest in their own staff as technical support. Indeed, few engineers could have more troubles in answering to many questions arising from beginners or even advanced users, whereas a community or thousands of experts is able to find solution to almost any problem due the heterogeneous knowledge coming from the high number of different profiles of participating professionals. In some cases, especially for open-source and free software, the existence of community, and if possible community in our own country (for language issues) and/or in our own scientific field, are really fundamental if we want to avoid to be stuck against a problem with the code or the strange behavior of the system. Just to make an example, many producers don't like to make public the list of bugs so it means that we can have a problem with our code and we don't know that the problem is not our code but is due to some built-in function not working properly. In that case the collaboration of other users is very essential for the solution.

A note has to be spent about such factors. Indeed, the existence of user communities or other kind of social aggregations around the statistical software we are evaluating, can assume a double value, as features to be evaluated because represent an important feature of the software, as above mentioned, and, at the same time, as source of information during and for the evaluation process, as peer recommendation or word of mouth. For that, the top-right corner of the matrix can be named "Word of mouth factors".

The classification of factors so far illustrated has also another advantage: it allows to define a sequence of factors to evaluate with an increasing complexity. Indeed, it should be easy to understand that the factors easy to discover and evaluate are those strictly related to the software and then those related to features external to the software but still clearly identifiable (objective ones). For that, it is strongly suggested to start from the Yes/No category because if a software doesn't has the primary requisites we are looking for we discard it without to further check other factors. Many times the last factors, those named Word of mouth, are still under evaluation after we purchased the software and started to use it, so can be considered the latest in the evaluation process. The following figure shows the suggested sequence (*Figure 5*).

Well, a first achievement has reached, now we have a tool useful to cluster factors according to some criteria and

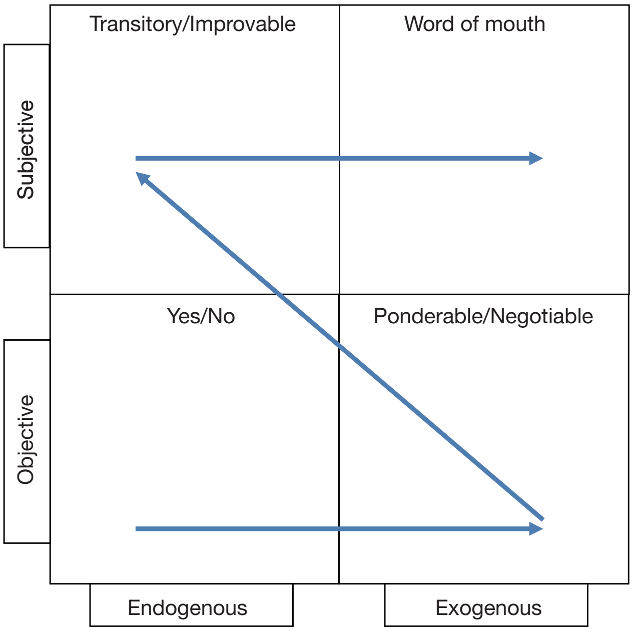

Figure 5 How it changes the complexity of the evaluation according to where the factor is located into the matrix.

to approach the increasing complexity of the evaluation in a step-by-step way. Now, we have to do a similar work for the evaluation process.

The evaluation process (or how to choose the right statistical software)

Even if in many situations the money investment is considered the central point of whole story, we strongly believe that the most important aspect is that, once we buy a statistical software and then start to learn and adopt it, we are doing an investment in terms of knowledge and skills and in terms of efficiency and affordability of results we get, that is much more important than money we pay the software license. So, the evaluation process have to be considered carefully and without to undervalue any critical factor.

"The selection of the most appropriate statistical program alternative involves multiple objectives or/and criteria and hierarchy process" [6]. For that we have to invest time and attention during it. After all, the evaluation process is nothing more than an optimization problem with constraints: "In mathematics, computer science and operations research, mathematical optimization (alternatively, optimization or mathematical programming) is the selection of a best element (with regard to some criteria) from some set of available alternatives [7]. Unfortunately, the objective function is not actually a real function of real variables, and the constraints are not easily convertible into equations. Thus, emotional and non-rational considerations have a great impact on that process.

In the previous paragraphs we spoken about the general problem of the evaluation process and then we presented a way to create a schematic representation of the huge cloud of factors we should consider before to buy and adopt a statistical software. Now, it is time to introduce the process of analysis of such factors. Even in that case we suggest a linear and conceptually easy approach that can be supported by the use of tables to create a clear picture of the scenario we are faced with, during the decision process.

First of all we introduce a variation to the buying decision process of five stages shown in *Figure 1*. In particular we split the first three stages up into smaller steps:

(I) Problem recognition (define goals and domain of variables):
   (i) Define the technical scenario (objective function):
      (a) Identification of the main workflow of statistical analysis: as suggested by the schema in *Figure 3* this should include the overall process and not just the statistical calculation moment. Additionally, the IT context where the research is embedded should be considered because it might address additional technical requirements like the interoperability with other software, the mid and long term potential development of the statistical analysis in the research field, and so on.
      (b) Identification of functional requirements: accordingly to the previous analysis, all those technical requirements critical for the appropriate and full exploitation of the statistical software, have to be decided.
   (ii) Define the context scenario (constraints):
      (a) Knowledge: level of knowledge, skills and competencies about the appropriate use of statistical software in the research strands where it has to be applied;
      (b) Money: budget for the software and related assets (training, books, counselling);

(c) Time: how long the "decision process" can be (when it's needed to start to use the software) and the "time to market" (when it is needed or estimated to get first results from the statistical research).

(iii) Identify a gross list of evaluation factors: at this stage it should be quite clear the whole list of factors, however, if needed, the list can be refined in any moment.

It's obvious that the first stage, when we have to define not only the technical problem for which we require the statistical software but also the "boundary conditions" that apply to the software purchase, we have to decide which evaluation factors have to be included in the process. The factors we include are related to the problem, the value they assume in our evaluation depends on the software products we find on the marketplace.

(II) Information search:

(i) Try to understand which kind of statistical software does fit your technical problem at the best.

(ii) Search information about different products/brands for the category of statistical software you need; in this step it is useful to consider the Yes/No factors.

(iii) Make a skimming of all software in the list and decide to fix your attention on three or max five different software (a few is better); at some level could be useful a raw check of the main Ponderable/Negotiable factors.

(iv) Search additional information for each candidate software, in particular collect source of information for each one (web sites, forums, local/national reseller contacts, user communities and so on).

The information retrieval is focused on three main communication channels: the software producer or their local reseller (web site, brochure, ad hoc material required via email, phone calls); the gray literature, online forums, user community, blogs, expert's web site, etc.; colleagues in our research network, having similar problem and that already adopted a statistical software (sometimes this is the most influential input we get).

(III) Evaluation alternatives:

(i) Refine the list of all factors considering the features of candidate software and try to highlight the most important once, those matching your actual needs.

(ii) Analyze carefully each factor individually for each candidate (for instance using the factor evaluation matrix shown below). From that analysis, in all likelihood, some candidates will be rejected (in most cases the final decision is made among two software).

(iii) Make a benchmark, rating all remained candidate in comparison each to others (for instance by using the second matrix).

One of the most critical step of the above list is (III) (ii) the analysis of each factors. Indeed, according to the *Figure 5* moving toward subjective and exogenous factors it increases the evaluation complexity. During this step it is very important to record (and keep track during the whole evaluation period) negative impacts of factors against our real needs. Indeed, what does influence our final decision, more than any other things, are negative impacts we register during the evaluation process.

Apparently the above process might appear too much long and bothering but you should consider that you are deciding about how you will spend the next months or even years working on your statistical scientific research, so the affordability and the efficiency of your choice is fundamental for yourself and the quality of your results. Moreover, in the case you will need to make the evaluation process once again in the next future, for instance some years later just because the software available on markets are updated or because you made a wrong choice and the selected software doesn't fit your long-term needs, the schematic approach and the use of matrix and table will help you to recover part of the previous work.

For that, we suggest to use an evaluation matrix or table that can help to declare and then evaluate and re-evaluate as many time as needed, the impact of each single factor. The table is like the following, but it has to be considered that anyone can add or modify the elements in the table according to specific context or different conditions about factors (*Table 3*).

Once that the tables related to the first software is filled in with elements for each factor, we can go ahead doing the same with the second software and so on. Finally, we can summarize the evaluation using the following matrix of scores (*Table 4*).

Table 3 List of notes about an evaluation factor

Evaluation factor	Name or description
Note	Short note about the how the candidate software matches the factor, could be a yes/no or a short comment
Impact	What impact could have on your specific scenario. This is important in particular for negative impact because it helps to keep track of what can represent a cause of post-purchase frustration
Remedy	Is there anything you can do to reduce negative impact? It could be something related to your knowledge or skill or a combination of other factors applying to your specific context
Alternative	Is there any other software with a lower impact? This is important because can help during the final benchmark step
Added value	Is there any added value (not yet included in other evaluation factor). Normally is something related to your personal experience that can add a value to a factor
Rating [1-5]	A personal global rate according to perception about factor applied to that software. 5/Excellent, 4/Good, 3/Fair, 2/Poor, 1/Bad

Table 4 Example of final score's matrix for evaluation of three software A, B and C

Software *vs*. evaluation factor	Evaluation factor 1	Evaluation factor 2	…	Evaluation factor k
Software A	scoreA1	scoreA2	…	scoreAk
Software B	scoreB1	scoreB2	…	scoreBk
Software C	scoreC1	scoreC2	…	scoreCk

The source of information

Looking around on internet, it's easy to discover that there are few more than 50 statistical software available on the marketplace that worth to have attention in the case we need a statistical software for scientific data analysis (not just descriptive statistics but even complex statistics model analysis or mathematical statistics). This means that a serious evaluation is not so easy and therefore a key role is played by channels and method we use to gather facts and opinions about all candidate software. We believe that a dedicated note is required about the source of information, being very wide and heterogeneous the range of sources we can find, especially on Internet. Of course, our intention is not to make a list of sites like a bookmark, rather we want to suggest a short set of criteria helping us to evaluate which sites and for what can be used during such process. Once again, a classification is possible and useful in order to simplify the search and exploitation of such resources. Considering the online resources we can identify the following type category of web sites and related common pros and cons (based on general consideration, that means not necessary valid for all products and all sources of information):

- Web sites of makers (in case of commercial software): software are widely presented and illustrated on the web site of the company developing and selling them. Normally, the most valuable and affordable source of information for many aspects, not for all. Many of such makers, for instance, make available the download of full functional demo, or full documentation and sometimes white papers and articles showing specific scope and use of their software. However, one of the side effects of such sources is the phenomena we call "Windows 95 preview" that is looking at the maker's web site it seems that the software is the best, the most powerful, without a bug and easy to learn and use. Then, sometimes the realty is quite far from that magic scenario. A short list of pros and cons (based on the general experience).
  - ○ Pros:
    - ▪ Appropriate and valid information about price and license schema.
    - ▪ Complete source of information for functional requirement (sometimes partially complete for non-functional, because being subjective requirements are not always expressed by makers).
    - ▪ Updated information and guarantee to be online until the product is still developed and sold (obsolescence of information, especially for those related to software that evolves quite frequently, is one of the most dangerous aspect of web sites).

- Generally available information related to service and learning resources like articles, books and courses.
- Cons:
  - No benchmark with other software (in rare case there are some comparison shown by means of list of features).
  - No wrong behavior or bugs; very rarely commercial software makers expose these details about their code because they believe this could influence the new buyer. However, for many users this is an appreciated value, because these information can represent a valid alternative to hours and hours of frustrating attempts to make working a snippet of code that doesn't behaves correctly due to a built-in malfunctioning.
  - Not at all or partial information about internal implementation notes. These kind of information may be useful during the buying evaluation process for deep analysis about particular methods and statistical technics used by built-in functions and macro of the software.
  - Well-hidden negative aspects like the missing or still partially developed functional requirements (to avoid an easy comparison with competitors).
  - Biased information especially for positive aspects (over evaluated) and negative aspects (belittled or hidden).
  - Normally just in English (sometimes with partial translation into Spanish, French or Dutch).
- Web site of developers (open-source software): for open-source software it's obvious to have a web site where all participants to the developers community can contribute with information and material, so this is still a valid source of information during the buying process but has some meaningful differences compared with commercial software web sites, we try to list in the following points.
  - Pros:
    - Source code, installer, demo or full products, new or old versions.
    - Much more code and technical information; many times they also provide a list of missing functions or future plans of developments and bug's list.
    - Implementation notes, sometimes very useful for deep analysis of technical functionalities of statistical functions.
    - Updated information and guarantee to be online until the product is still developed (but here, generally speaking, there is no guarantee about mid-long term developing plans due to the open source paradigm).
    - Being a "community product" many times local community creates translations of official documentation and web sites, so a better probability to find localized web sites with useful information for the evaluation process.
  - Cons:
    - Quality of information: redundancy, missing and sometimes not fully affordability of information due to a lack of unifying marketing policy; not all open source developers' communities have a good coordination and common marketing strategy (sometimes they don't like to declare to have a marketing strategy at all, anyhow they propose a product on the market and what they present and how they offer is evaluated under the marketing point of view).
    - Missing information about applications, specific uses and case study; open software have the features that there are much more personal web sites or sub-community web sites (like those in local languages), but the "official" web site is much more technical than informative. On the other hand there is not an explicit investment in marketing. This kind of information is found but in other kind of web sites (listed below).
- Web site of individual expert or consulting companies: another important piece of information during the buying process can be collected reading experience and documents shared by those users having much more practice with the software.
  - Pros:
    - Information are normally objective and impartial, being the experts/companies external to the software selling business. However, sometimes they offer third-party applications developed with the software and/or counseling and training, so in that cases they can be interested in the software promotion.
    - They provide a wide range of examples of applications of the software, being experts/companies involved in different research fields, for that the information we get is suitable to see the software in action in many different

situation.
    - Many times they also provide benchmarks and performance tests, with criticisms about points of weakness and limits of the software.
  - Cons:
    - Information can be sometimes partial, being a personal opinion about software and its applicability in specific fields.
    - They do not provide a general overview of the software, rather case study focused on specific topics.
- Web site of community of users: almost any software in the world has its own users community acting by means of a forum or wiki portal or a general web site. These are important not only as source of information during the evaluation process but also when we use the software, because they provide great support for many situations. As for the evaluation, here we can contact someone and post direct questions or we can participate to beginners threads so to collet additional impressions and opinions about the candidate software. Though, we have to pay attention because forums and similar sites are frequented by any kind of user, many times very inexperienced and too aggressive in the expression of opinions (both in positive or negative against the software).
- Web sites of scientific magazines or publisher: in some cases we can also find scientific magazines dedicating attention to the statistical software in general or applied to a specific subject. In that case we can read articles, books or any other documentation written by scientists and researchers, from all around the world, using the statistical software for a particular research topic. These are very useful and high level user experiences or case stories and they can definitively drive our decision, influencing the perception we get about how the software can solve real life cases. If we found an authoritative magazine dealing with statistical application in our research field, most of the times we will decide the same software they suggest.

In the bibliography there is a list of many sites useful for a first check of general and detailed information about many of the most known and adopted statistical software.

Conclusions

We are convinced that the decision process related to the purchase of a statistical software is a process strongly depending on the context where we have to undertake the decision. Constraints like the budget, the surrounding IT infrastructure, the level of programming knowledge and skills of end user, the extent to which the statistical analysis has to be addressed, and many other context-based elements are much more than factors, they open to different scenario and require more than a reflection. The evaluation of functional requirement is an important moment and can be even easy (a good example of how different software compare is found here https://en.wikipedia.org/wiki/Comparison_of_statistical_packages and here https://sites.google.com/a/nyu.edu/statistical-software-guide/summary), but it's not enough and many times can drive to a wrong choice.

Being the context too relevant for the final decision, we think it is not appropriate to propose a fixed list of factors to be used for "every season".

Moreover, the current scenario of statistical software, as said before with few more than 50 different software, is characterized by a group of leading tools—less than ten software, and a big group of small tools with minor features but very useful and suitable for many common statistical analyses. So, even if we want to reserve our attention to the "top ten" of statistical software, we find that many of them have similar features and an "a priori" comparison cannot be done. The only thing we can do is to measure the matching of each one with our particular needs and context setting.

Thus, we focused on a methodology enabling us to classify facts and information about each software and attribute to them our personal and subjective score, according to our idea of the context where we need to use the software. We suggested a schematic approach to each of the step in the decision process, hopefully preserving the evaluator from wrong decision. The time needed to follow such process is surely bigger than what normally one would guess for a statistical software, but the investment we are doing adopting a new software for research data analysis is too strategic to be spent in the wrong direction. More the time we spend during the evaluation less is the probability we will need to replace the software with a different one in the next future.

Acknowledgements

None.

Footnote

Conflicts of Interest: The author has no conflicts of interest to declare.

Bibliography

[1] Available online: https://en.wikipedia.org/wiki/Decision-making

[2] Available online: https://en.wikipedia.org/wiki/Buyer_decision_process

[3] Available online: http://www.capterra.com/software-buying-trends-2013

[4] Available online: https://en.wikipedia.org/wiki/Programming_language

[5] Available online: https://en.wikipedia.org/wiki/Application_software

[6] Kaygisiz Ertuğ Z, Girginer N. A multi criteria approach for statistical software selection in education. Hacettepe Universitesi Egitim Fakultesi Dergisi-Hacettepe University Journal of Education 2014;29:129-143.

[7] Available online: https://en.wikipedia.org/wiki/Mathematical_optimization

[8] Ozgur C, Kleckner M, Li Y. Selection of statistical software for solving big data problems. SAGE Open 2015:1-12.

[9] White Paper from HyperOffice. SELECTING SOFTWARE A Systematic Approach to Buying Software. Jan 2008. Available online: http://www.hyperoffice.com/files/pdf/selectingsoftware.pdf

[10] Roberts D, Cater-Steel A, Toleman M. Factors influencing the decisions of SMEs to purchase software package upgrades. 17th Australasian Conference on Information Systems—6-8 Dec 2006, Adelaide.

[11] Loebbecke C, Weiss T, Powell P, et al. Drivers of B2B software purchase decisions. In: Respício A, et al. editors. Bridging the socio-technical gap in decision support systems. IOS Press, 2010.

[12] Weisberg S. Lost Opportunities: Why We Need a Variety of Statistical Languages. Journal of Statistical Software 2005;13. Available online: file:///C:/Users/smt/Desktop/v13i01.pdf

[13] Muenchen RA. The Popularity of Data Analysis Software. Available online: http://r4stats.com/articles/popularity/

[14] Available online: http://www.jstatsoft.org/

[15] Available online: https://blog.minitab.com/blog/understanding-statistics/choosing-statistical-software-four-questions-you-should-ask (Choosing Statistical Software: Four Questions You Should Ask).

[16] Available online: https://blog.minitab.com/blog/understanding-statistics/what-statistical-software-should-you-choose-three-more-critical-questions (What Statistical Software Should You Choose: Three More Critical Questions).

[17] Available online: http://www.theanalysisfactor.com/choosing-statistical-software/ (SPSS, SAS, R, Stata, JMP? Choosing a Statistical Software Package or Two).

[18] Available online: https://sites.google.com/a/nyu.edu/statistical-software-guide/summary (Which Statistical Software to use?).

[19] Available online: http://www.statistical-software.net/ (Statistical Software Guide).

[20] Available online: http://www.capterra.com/statistical-analysis-software/ (Top Statistical Analysis Software Products).

[21] Available online: https://en.wikipedia.org/wiki/Comparison_of_statistical_packages (Comparison of statistical packages).

[22] Available online: http://brenocon.com/blog/2009/02/comparison-of-data-analysis-packages-r-matlab-scipy-excel-sas-spss-stata/ (Comparison of data analysis packages: R, Matlab, SciPy, Excel, SAS, SPSS, Stata).

[23] Available online: http://www.wolfram.com/solutions/industry/statistics/ (The Wolfram Solutions for Statistics).

[24] Available online: http://www.idealware.org/articles/purchasing_major_systems.php (The Perfect Fit: A Guide to Evaluating and Purchasing Major Software Systems).

[25] Available online: http://www.icpsr.umich.edu/icpsrweb/ICPSR/studies/1243/version/1 (Statistical Software Benchmarks (ICPSR 1243)).

[26] Available online: http://www.businessknowhow.com/manage/software.htm (10 Questions to Ask Before Buying Software).

[27] Available online: http://r4stats.com/articles/popularity/ (The Popularity of Data Analysis Software).

[28] Available online: http://influitive.com/blog/peer-recommendations-happier-with-b2b-software-purchases/ (Peer Recommendations Make Buyers 5X Happier With B2B Software Purchases).

[29] Available online: http://www.softwareadvice.com/buyerview/software-selection-smbs-report-2014/ (The Best Software Selection Tactics for SMBs BuyerView | 2014) .

[30] Available online: http://blog.market8.net/b2b-web-design-and-inbound-marketing-blog/Detailing-the-

Buying-Process-for-B2B-Awareness-Evaluation-and-Decision (Detailing the Buying Process for B2B: Awareness, Evaluation and Decision).

[31] Available online: https://insightpool.com/peer-to-peer-recommendations-still-valuable/ (Software Advice Survey: Are peer-to-peer recommendations still valuable?).

[32] Available online: http://www.prostatservices.com/statistical-consulting/articles-of-interest/a-review-of-the-top-five-statistical-software-systems (A Review of the Top Five Statistical Software Systems).

[33] Available online: http://www.amstat.org/careers/statisticalsoftware.cfm (Statistical Software).

[34] Available online: http://www.ats.ucla.edu/stat/stat_pkg.htm (Purchasing and Updating Statistical Software Packages).

Cite this article as: Cavaliere R. How to choose the right statistical software?—a method increasing the post-purchase satisfaction. J Thorac Dis 2015;7(12):E585-E598. doi: 10.3978/j.issn.2072-1439.2015.11.57